

Oracle® Cloud

Administering Visual Builder Studio



Release 26.01.0

G49427-01

January 2026

ORACLE®

Oracle Cloud Administering Visual Builder Studio, Release 26.01.0

G49427-01

Copyright © 2020, 2026, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

About This Content

1 The Basics

What Is Oracle Visual Builder Studio?	1
What's an Extension?	2
Build Enterprise Apps in Minutes	3
Use VB Studio to Manage Your Development Cycle	4
How Do I Set Up VB Studio?	5
Key Concepts, Components, and Terms	5
Access VB Studio from the Oracle Cloud Home Page	7

2 Get Started with Setting Up VB Studio

3 Set Up VB Studio to Extend Oracle Cloud Applications

Work With Multiple VB Studio Instances	1
Essential Setup Tasks	5
Create a Simple Extension	6
What Project Artifacts Are Created?	10
Manage Your Simple Extension	12
What's Next?	14
Set Up VB Studio Users	14
How Do I Assign the Roles?	16
Configure Oracle Cloud Applications Custom Roles	20
Optional Administrator Tasks	21
Set Merge Restrictions on the main Branch	22
Use CI/CD Pipelines for Deployment	22
Deploy to Production Through CI/CD Pipelines	24
Deploy to Other DEV and TEST Instances Through CI/CD Pipelines	40
Manually Create a Project for Extensions	49
Access VB Studio from Oracle Cloud Applications	50
Delete an Extension	51

Delete an Extension Manually	51
Configure a Job to Delete an Extension	53

4 Set Up VB Studio for Developing Visual Applications

Set Up VB Studio for Visual Applications	1
Get Access to Visual Builder Instances	2
Get the Required IDCS Roles	3
Set Up the VB Studio Project for Development	3
Create and Set up a Project for Development	4
Create and Set Up a Project Using the Visual Application Template	8
Create a Project Using the Visual Application Template	9
Configure the Deployment Job	13
Run the Pipeline Manually	14
View the Deployed Visual Application	14
Use CI/CD Pipelines for Deployment	15
Add Users to the Project	16
Optional Configuration	18
Configure the Packaging Job	18
Set Merge Restrictions on the main Branch	19
Develop Your Application	20
Set Up Additional Deployment Instances	20
Set Up the Project to Deploy for Production	21
Create a Production Branch	21
Create and Configure Production Build Jobs	22
Create a Production Packaging Build Job	23
Create a Production Deployment Build Job	23
Configure a Production Job's Privacy Setting	25
Create and Configure a Pipeline	25
Run the Pipeline	26
Undeploy a Visual Application	26
Undeploy a Visual Application Manually	27
Configure a Build Job to Undeploy a Visual Application	27

5 Set Up VB Studio for CI/CD

Step 1: Decide whether to use the free VM build executor	1
OCI Resources in VB Studio	2
VB Studio's Free VM Build Executor	4
Configure VB Studio to Use the Free VM Executor for Build Jobs and Pipelines	5
Add the Free VM Build Executor Manually	6
Connect to Your OCI Paid Account	7

Get the Right Roles	8
Create a VB Studio Instance in Your Oracle Cloud Account	8
Step 2: Decide whether to run VM build executors or Docker executors	12
What are Build Executors and Build Templates?	12
VM Build Executors	13
Docker Build Executors and Docker Images	18
Step 3: Set up VB Studio to run VM build executors	21
Create and Manage Build Executor Templates	21
Add and Manage VM Build Executors	23
What Determines How Many VMs I Can Create?	30
Connect to Your OCI Account and Add VM Build Executors	30
Run VM Build Executors in Virtual Cloud Network	32
Use VB Studio's Default VCN	32
Run VM Build Executors in Another Compartment's VCN and Subnets	34
Software for Build Executor Templates	41
Step 4: Set up VB Studio to run builds on Docker executors	43
Add Your First Docker Deployment VM	44
Migrate to Docker	47
Create and Manage Docker Images	49
Add and Manage Docker Deployment VMs	51
Add Docker Deployment VMs	51
Manage Docker VMs	52
Manage Docker VM Resources	53
Reset Docker and Move to VM Build Executors	55
Step 5: Set Up Users and Groups	55

6 Manage the VB Studio Instance

Change the CI/CD Compartment	1
Authorize VB Studio to Access Visual Builder and Integration Instances	2
Switch to the Built-In Free Account	3
Change Your OCI Account	3
Validate the OCI Connection	4
Manage VB Studio Instance Tags	5
Write IAM Policies that Control Access to VB Studio	5
Create Rules that Use Service Event Notifications	7
Move the VB Studio Instance	7
Terminate the VB Studio Instance	8

7 Manage the Organization

Update the Organization's Display Name and Description	1
--------------------------------------------------------	---

Set the Organization's Default Wiki Markup Language	1
Restrict Project Creation to Organization Administrators	1
Restrict Sharing a Project to Organization Administrators Only	2
Limit the Number of Personal Access Tokens a User Can Create	2
Assign a Project's Ownership to Another User	5
Manage Your Organization's Projects	5
Manage Your Organization's Groups	8

8 Manage Your Component Exchange

What is a Component Exchange?	1
Use a Project as the VB Studio Component Exchange	1
Use an External Server as the VB Studio Component Exchange	3
Share the Component Exchange's URL	3
Workflow for Web Components	4

9 Troubleshooting

Resolve Access Issues After Switching Users to Federated IDCS	1
Unable to See the "Edit Page in Visual Builder Studio" Option in an Oracle Cloud Application	2
How Do I Resolve a "You Are Not a Member of This Organization" Error?	2
How Do I Access OCI Identity and Access Management From the Oracle Cloud Console?	2
Resolve Access Issues After Allow-Listing Your VB Studio IP Address	3
Resolve the "Loading Completed with Errors" Message	4

Part I Appendices

A Migrate Oracle Cloud Applications Configurations

Before You Configure CSM Build Jobs and Pipelines	A-1
Create and Configure Build Jobs to Migrate Your Oracle Cloud Application's Configurations	A-3
Configure a Job to Export Configuration	A-4
Configure a Job to Import Configuration	A-5
Configure a Job to Cancel the Configuration Migration	A-7
Configure a Job to Apply and Save the Configuration Migration	A-8
Configure a Job to Restore Configuration Migration	A-9
Create and Configure the CSM Pipeline	A-10
Create and Configure Build Jobs to Deploy the App Extension to the Sandbox	A-14
Create the Production Sandbox Packaging Job	A-14
Create the Production Sandbox Deployment Build Job	A-16
Create and Configure the Production Sandbox Pipeline	A-19

B Public IP Addresses for VB Studio

About This Content

This guide describes how to set up Visual Builder Studio to develop extensions and visual applications, and to manage large software development projects that use Oracle Cloud Infrastructure (OCI) cloud services.

Audience

This document is intended for Oracle Visual Builder Studio administrators who administer and set up the service.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Resources

For more information, see these Oracle resources:

- Oracle Public Cloud
<http://cloud.oracle.com>
- Manage Your Development Process in *Using Visual Builder Studio*
- Create Visual Applications in VB Studio in *Building Responsive Applications with Visual Builder Studio*
- What Can You Do with Oracle Visual Builder Studio? in *Extending Oracle Cloud Applications with Visual Builder Studio*

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

The Basics

Learn about Oracle Visual Builder Studio, its projects, components, roles, and how to access the service.

See Oracle Cloud Terminology in *Getting Started with Oracle Cloud* for definitions of terms found in this and other documents in the Oracle Cloud library.

What Is Oracle Visual Builder Studio?

Oracle Visual Builder Studio (VB Studio) is a robust application development platform that helps your team effectively plan and manage your work throughout all stages of the app dev lifecycle: design, build, test, and deploy. It makes it easy for your entire team to develop the artifacts they need, including:

- Oracle Cloud Applications developers, who need to create *extensions* to customize their Apps—like Human Capital Management, Supply Chain and Manufacturing, and so on—to meet their business needs.
- Low-code developers, who want to create *responsive applications* using a visual designer, then deploy them to their own version of Visual Builder.
- Experienced programmers, who want to modify the source code for applications created by others, or to develop bespoke apps using the web programming language of their choice.

Regardless of what type of developer you are—or if you're focused on managing the work that others are creating—everyone can take advantage of these key VB Studio features:

- A rich visual designer integrated with source control (Git) so that developers can manage changes, apply version control best practices, and collaborate with their teammates to develop applications
- The ability to build and display different flavors of the UI to meet the needs of discrete users of certain Oracle Cloud Applications (those built with VB Studio and Oracle JavaScript Extension Toolkit (Oracle JET)), also within a Git framework
- Built-in repositories for hosting code in Git and for hosting binaries, such as Maven dependencies
- A continuous integration service so you can automate your build and test systems
- A continuous delivery service that tightly integrates with Oracle Cloud Applications
- Agile boards and an issue tracking system for tracking sprints, tasks, defects, and features

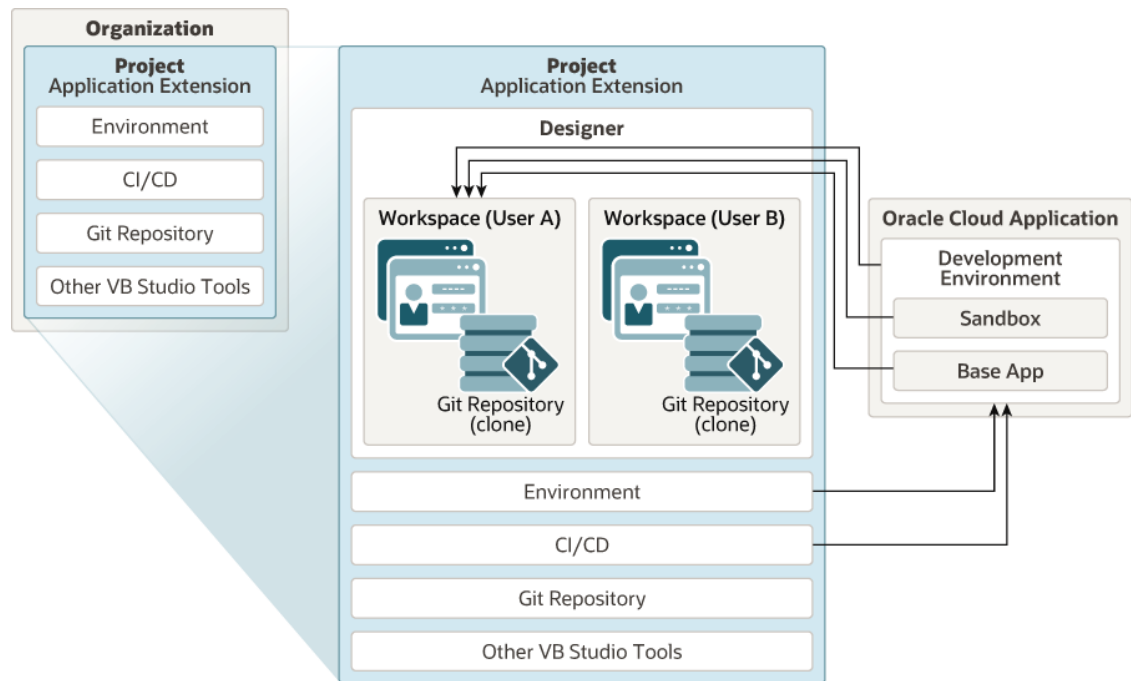
VB Studio enables developers to easily deploy their applications to their preferred target, whether it's a staging or production instance of Oracle Cloud Applications, an Oracle Cloud Infrastructure (OCI) service instance, or a Visual Builder instance.

Note

[Public Cloud Regions](#) lists the commercial, EU Sovereign, and US/UK Government regions in which Visual Builder Studio is currently available.

What's an Extension?

The changes you make to your Oracle Cloud Application in VB Studio are stored in an artifact called an *extension*. Physically, the source files associated with the extension are stored in a Git repository.



Here are the key components of the VB Studio ecosystem as it pertains to extensions:

- Within a single Oracle Cloud Application environment family—which comprises all your development, test, and production instances—you have multiple instances of VB Studio, all of which are tied together into a single *organization*. Within your organization, you will likely belong to one or more *projects*, each of which is devoted to a discrete software effort. For example, you might have a project for building extensions for Oracle Cloud Applications, and another project for building a bespoke application your own department will use. A project brings together all the tools you need to create those artifacts, such as a Git repository for storing your source code, a pipeline to provide continuous integration and delivery, an issue tracking system, team wikis, and more.
- When it comes to extensions, it's easiest to use one project for all the extension work you do within an Oracle Cloud Applications environment family. Within that project, you can create a separate Git repository to store the source files for each pillar—one repo for HCM, another for CRM, and so on. If your team is too large to make that feasible, however, or if you simply want to isolate the extension work for each pillar, you can keep things completely separate by creating a unique project for each Oracle Cloud Application.
- When you work on an extension, you do so within the editors provided in the VB Studio Designer. Depending on which aspect of the page you're customizing, VB Studio invokes the proper editor to provide the experience you need.
- All of your work in VB Studio is done in the context of a *workspace*, a completely private area where you can work on your extension.

Build Enterprise Apps in Minutes

VB Studio makes it easy for you to build custom interfaces that meet your unique needs. No coding, no setup, and no IT resources required. It provides cloud-based visual tools to help you rapidly create and host engaging applications.

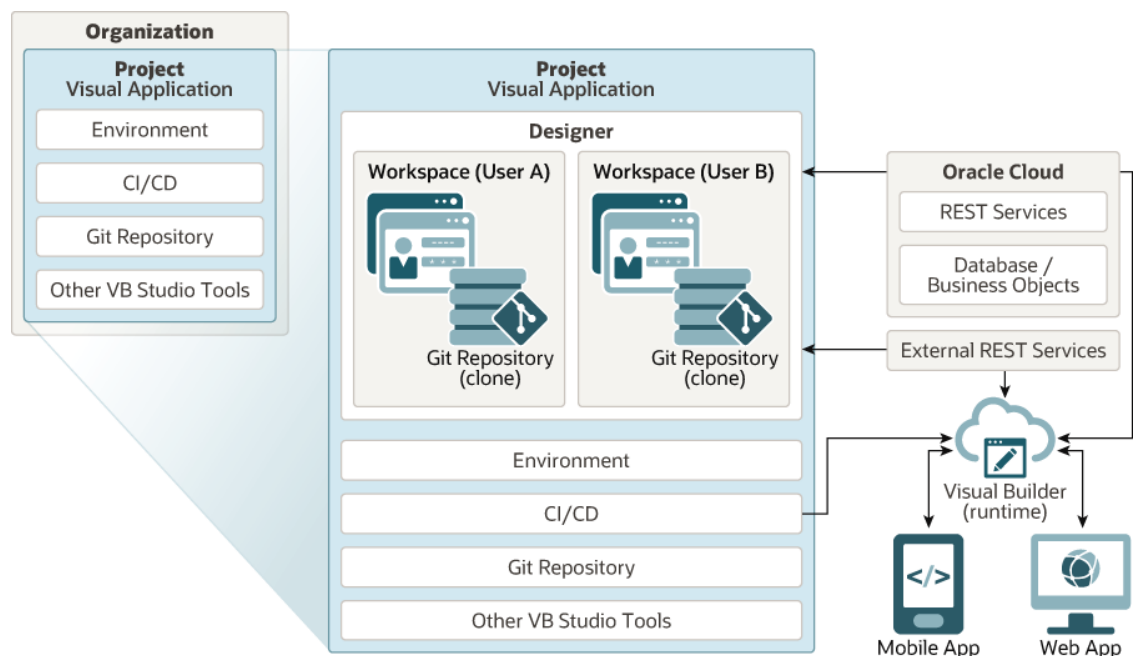
With VB Studio, you can:

- Use the what-you-see-is-what-you-get (WYSIWYG) designer to visually create your app: simply drag and drop UI components onto your app's pages, customize their behavior, and arrange them however you want. For more complex needs, you can extend functionality using standard JavaScript, HTML5, and CSS.
- Connect to data through REST-based services: you can create your own business objects that implement your app's business logic and store its data, pick data objects exposed by an integrated catalog of Oracle REST services (such as Oracle Cloud Applications and Oracle Integration), or pull data from any external REST service with just a few clicks.
- Deploy your application at the click of a button and make it available to users. This means you can publish your app instantly and access it on any device, anytime. You can even deploy it as a Progressive Web App (PWA)—combining the on-device mobile experience with a web app's ease of distribution—so it can be installed on user devices to leverage capabilities such as camera access and location information.

VB Studio also provides:

- Integration with Git repositories so you can share application resources with team members
- Infrastructure for securing access to your application, its data, and the Oracle Cloud services it consumes.

All of this becomes available to you within a *project*, which is the starting point for building a visual application in VB Studio. Here are the key components of the VB Studio ecosystem as it pertains to visual applications:

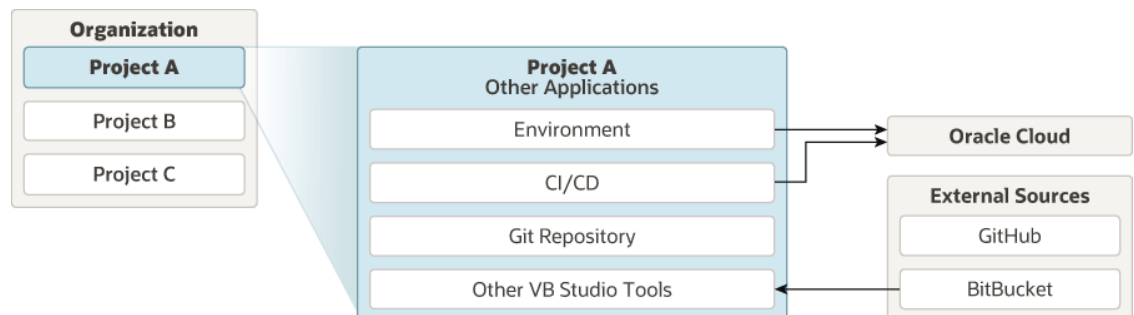


- Within a single VB Studio instance, you and your team members who use that instance are considered an *organization*. Within your organization, you will likely belong to one or more *projects*, each of which is devoted to a discrete software effort. For example, you might have a project for building a new Financial app, and a different project for creating an HR application. A project brings together all the tools you need to create those artifacts, such as a Git repository for storing your source code, a pipeline to provide continuous integration and delivery, an issue tracking system, team wikis, and more.
- All of your work in VB Studio is done in the context of a *workspace*, a completely private area where you can work on your visual application. Your work is stored in your own clone of the project's Git repository, and is not visible to others until you push it to the project's Git repo, choose to share it with others for testing, or publish it. A workspace also includes a pointer to the development or test environment where you plan to deploy your app, which must be a separate Oracle Visual Builder instance (also known as an environment). You can deploy your app manually, or wire it up to a pipeline to do it automatically, such as when a developer on your project publishes their changes.
- Your workspace is your ticket into the *Designer*, a declarative environment that helps you build visual applications. Specifically, you get a page editor, a Components palette, and a Structure view, for designing the layout of a page. Within this environment, you have access to a rich set of UI components, including those based on Redwood, the Oracle standard for user experience. You can use these components—all based on Oracle JET—to develop apps that provide the same look and feel as apps delivered from Oracle.
- The data for your visual applications can come from a database (in the form of business objects), Oracle REST services, or external REST services.

If you previously developed visual applications in Visual Builder and now work with VB Studio, you'll see that some things have changed. For details about tasks that you now do differently in VB Studio, see [For Visual Builder Users](#).

Use VB Studio to Manage Your Development Cycle

The VB Studio components shown in the diagram are used to manage the development process for applications other than visual applications or application extensions:



Though many users will rely on Visual Builder Studio to help them create visual applications or create extensions for Oracle Cloud Applications, you can use VB Studio purely as a tool to help you manage your development processes:

- Within a single Visual Builder Studio instance, you and your team members who use that instance are considered an organization. Within your organization, you will likely belong to one or more projects, each of which is devoted to a discrete software effort. A project brings together all the tools you need to create those artifacts, such as a Git repository for storing your source code, a pipeline to provide continuous integration and delivery to the Oracle Cloud, an issue tracking system, team wikis, and more.
- Although VB Studio provides a Git repository for you, you can choose to use external repos, such as GitHub and BitBucket.

How Do I Set Up VB Studio?

Using VB Studio's hosted development platform, you and your team can develop new visual applications or customize certain Oracle Cloud Applications by extending them. These extensions are then managed just like any other VB Studio artifact. You can also develop software applications using any language you choose.

Before your team can use VB Studio, you, as the administrator, need to set it up, configure it, add your team members to your identity domain in Oracle Cloud Infrastructure (OCI) Identity and Access Management, and assign them proper roles.

Note

Only one identity domain is supported per service instance.

Most administrators will follow this path to set up VB Studio:

First, everyone should read [Get Started with Setting Up VB Studio](#) to see if you need to create your own VB Studio instance. If you're using VB Studio solely to configure and extend Oracle Cloud Applications, for example, you can skip this step because Oracle will provision VB Studio instances for you.

After that, your next steps depend on what you want to do:

1. If you want to set things up so you and your users can extend (customize) Oracle Cloud Applications, you can go directly to [Set Up VB Studio to Extend Oracle Cloud Applications](#).
2. If you plan on creating visual applications hosted on an instance of Visual Builder, go to the [Set Up VB Studio for Developing Visual Applications](#) chapter.
3. If you want to manage a large software development project that uses OCI cloud services and doesn't involve either Oracle Cloud Applications or visual applications, work through [Set Up VB Studio for CI/CD](#).

The other chapters in this guide are intended for use on an as-needed basis.

Key Concepts, Components, and Terms

Before you use VB Studio, it helps to become familiar with these key concepts, components, and terms. (If you have general questions about Oracle Cloud features and services, see Oracle Cloud Terminology in *Getting Started with Oracle Cloud*.)

Term	Description
Oracle Cloud Infrastructure (OCI)	Oracle Cloud Infrastructure is a set of cloud services that enable you to build and run a wide range of applications and services in a highly available hosted environment. Oracle Cloud Infrastructure offers high-performance compute capabilities (as physical hardware instances) and storage capacity in a flexible overlay virtual network that is securely accessible from your on-premise network.
Oracle Cloud Infrastructure Compute (OCI Compute)	Service that hosts virtual machines (VMs) on Oracle Cloud with all the necessary storage and networking resources. VB Studio uses the VMs to run project builds.
OCI Compute VM instance	A virtual machine that runs on top of physical bare metal hardware. To learn more about a Compute instance, see Overview of the Compute Service .

Term	Description
OCI Compute VM shape	A shape is a template that determines the number of CPUs, amount of memory, and other resources allocated to a newly created VM Compute instance. To find more about shapes, see VM Shapes .
OCI region	OCI is hosted in regions and availability domains. A region is a localized geographic area, and an availability domain is one or more data centers located within a region. To learn more about regions and availability domains, see Regions and Availability Domains .
OCI Virtual Cloud Network (VCN) and subnets	A VCN is a software-defined network that you set up in Oracle Cloud Infrastructure data centers in a particular region. To find out more about VCNs and subnets, see VCNs and Subnets and Overview of Networking .
Oracle Cloud Infrastructure Object Storage (OCI Object Storage)	Oracle Cloud service that hosts containers on Oracle Cloud to store project data. VB Studio uses the buckets to archive build artifacts and Maven artifacts, and export project data.
Oracle Cloud Applications	Oracle Cloud Applications are a set of modular Cloud-ready enterprise applications. To learn more, see https://www.oracle.com/applications/ .
Application UI (App UI)	An App UI is an application that includes a user interface component in the form of Visual Builder pages and flows. Some App UIs are created by Oracle—like some Oracle Cloud Applications—but you can build your own App UIs and deploy them as peers alongside Oracle's App UIs in your Oracle Cloud Applications ecosystem.
Visual Builder	A Visual Builder instance that provides the server for delivering pages in web applications, and services your apps might use to access data, including the database used to store data and the proxy server for managing connections to REST services.
Visual application	A responsive application developed using VB Studio's browser-based development environment. You deploy a visual application to a Visual Builder instance.
Project	A project is a collection of VB Studio features. You can use a project to host source code files, track issues, collaborate on code, build, and deploy your applications. A project can host multiple Git repositories. Each Git repository can have multiple branches and hundreds of code files. You can create a merge request for each branch of the Git repository and ask reviewers to review the code. You can create and configure multiple build jobs to generate different project artifacts that you can deploy to Oracle Cloud or your on-premise web server.
Organization	The top-most entity in the project structure of VB Studio. Think of an organization as the umbrella for all the projects in a given identity domain.
VB Studio Designer	VB Studio's browser-based development environment.
Git repository	A Source Code Management (SCM) and distributed version control tool to host source code files.
Maven repository	A hosted binary repository to store build artifacts, library files, and dependencies for Maven applications.
Issue tracker	A built-in issue management system to create and track tasks, defects, and features.
Environment	Defines the target Oracle Cloud Applications, Visual Builder, Oracle Cloud SaaS, or Oracle Cloud Infrastructure service instance as a single entity. You'll define an environment to deploy an application to a service instance or to get information from a service instance.


Term	Description
Merge request and code review	A method to merge a Git repository branch with another branch. Before merging the branches, team members can review differences between files of both branches and provide their feedback.
Wiki	Built-in wiki system to help your team author and manage wiki pages.
Build system	A built-in system to define and automate builds of your applications.
Continuous Integration (CI) and Continuous Delivery (CD)	Continuous integration is a set of practices that allow development teams to implement small code changes and push the code to version control repositories, such as Git, frequently. Continuous delivery is a practice that enable developers to produce software in short cycles.
VM build executor	An OCI Compute VM instance dedicated to run VB Studio builds. Only one build can run on a VM build executor at a time. To learn more, see VM Build Executors .
Build executor template	A template that defines the operating system and the software installed on a VM build executor. To learn more, see Build Executor Templates .
Docker image	Defines the operating system and software packages your organization needs to run builds on a Docker executor. A Docker image can be either imported from an external Docker registry or created from a build executor template.
Docker deployment VM	An OCI VM Compute instance dedicated to run builds of jobs defined in VB Studio projects using Docker executors.
Docker executor	When a job is created, a Docker image is associated as a build template with the job. Then, when the job's build is triggered, a Docker executor is created using the Docker image and deployed on any Deployment VM to run the build. Unlike a VM executor, a Docker executor is not directly associated with a specific VM.
Job (or build job)	A configuration that defines your application's builds. You can create a job to perform various actions, such as package artifacts, run shell commands, run unit test scripts, and deploy application artifacts.
Build	The result from a job's run.
Pipeline	A path or a chain of builds. A pipeline helps you run continuous integration jobs and reduce network traffic.
Oracle Java Cloud Service	Oracle Cloud service to deploy web applications to a public Oracle WebLogic Server domain on Oracle Cloud.

Access VB Studio from the Oracle Cloud Home Page

If you've signed out of the Oracle Cloud or VB Studio, follow these steps to sign back in.

1. In a web browser, go to <https://cloud.oracle.com>.
2. On the Sign-In page, in **Cloud Account Name**, enter your Oracle Cloud account or tenancy name and click **Next**.
3. On the Single Sign-On (SSO) panel, if required, select your identity domain and click **Continue**.
4. Enter your Oracle Cloud account credentials and click **Sign In**.

The Oracle Cloud Console, also called the OCI console, opens. If you've recently created your Oracle Cloud account, wait for some time to see your services in the Oracle Cloud Console.

5. In the upper-left corner, click **Navigation Menu** .
6. Select **Developer Services** and then select **Visual Builder Studio**.

7. On the Visual Builder Studio instances page, click the VB Studio instance's name.
8. On the instance details page, click **Service console**.

The VB Studio **Organization** page opens, which displays all the projects you're a member of, as well as your favorite projects, the projects you own, and all the shared projects in your organization.

Organization administrators can see additional tabs where they can set up OCI connections and build executors, and configure the organization's component exchange, groups, and properties. The Activities tab tracks changes made at the organization level, such as deleted projects or projects that the organization administrator assigned themselves as owner.

Here's an example of the Organization page with some projects:

The screenshot shows the Visual Builder Studio Organization page. On the left is a sidebar with navigation options: Organization, Project Home, Workspaces, Git, Merge Requests, Maven, NPM, Docker, Builds, and Releases. The main content area is titled 'Visual Builder Studio' and 'Organization'. It features a search bar and tabs for Member, Favorites, Owner, Shared, and All. A summary bar shows 698 Projects and 20.64 GB Disk Usage. Below this is a table of projects with columns for Name, Template, Favorite, Status, Updated, Members, and Disk Usage. The table lists four projects: Customer Experience, Demo, Employee Manager, and CX Express Mode.

Name	Template	Favorite	Status	Updated	Members	Disk Usage
Customer Experience		☆	● Active	Feb 12, 2025	YH	11.56 MB
Demo	Based on: Visual Applica...	☆	● Active	on Thu	YH	23.05 MB
Employee Manager	Based on: Visual Applica...	☆	● Active	Feb 3, 2025	YH	20.57 MB
CX Express Mode CX Pillar Testing Express Mode	Based on: Application Ex...	☆	● Active	Jan 27, 2025	YH	22.48 MB

2

Get Started with Setting Up VB Studio

To get started, consider what you want to do with VB Studio. Your answer determines whether or not you'll need to create your own VB Studio instance.

Use Case	Description	Where to Get More Details
1	<p>You won't need to create your own VB Studio instance if you want to do any of these things:</p> <ul style="list-style-type: none">• Configure Oracle Cloud Applications by modifying existing application pages• Extend Oracle Cloud Applications by creating a bespoke <i>App UI</i> and hosting it alongside your Oracle Cloud Applications• Extend Oracle Cloud Applications by creating a bespoke <i>visual application</i> and hosting it on a Visual Builder instance—when your developers have access to Oracle Cloud Applications* <p>— *If you don't have access to Oracle Cloud Applications, see the next row.</p> <p>For the above use cases, don't create a VB Studio instance; Oracle provisions you with the instances you need.</p>	Set Up VB Studio to Extend Oracle Cloud Applications
2	<p>You will need to create your own VB Studio instance, however, if you want to do these things:</p> <ul style="list-style-type: none">• Use VB Studio as a CI/CD tool to manage the development operations of a large software implementation project• Extend Oracle Cloud Applications by creating a bespoke <i>visual application</i> and hosting it on a Visual Builder instance—when your developers don't have access to Oracle Cloud Applications	Create a VB Studio Instance in Your Oracle Cloud Account

Once you've got your VB Studio instance set up, you can move on with the rest of the chapters in this guide:

- To use VB Studio to customize certain Oracle Cloud Applications through extensions, see [Set Up VB Studio to Extend Oracle Cloud Applications](#).
- If you want to develop responsive applications and deploy them to Visual Builder instances, see [Set Up VB Studio for Developing Visual Applications](#).
- To manage a large software development project that uses OCI cloud services and doesn't involve either Oracle Cloud Applications or visual applications, see [Set Up VB Studio for CI/CD](#).

3

Set Up VB Studio to Extend Oracle Cloud Applications

This chapter tells you how to set up Oracle Visual Builder Studio (VB Studio) so your users can work with Oracle Cloud Applications in an extension.

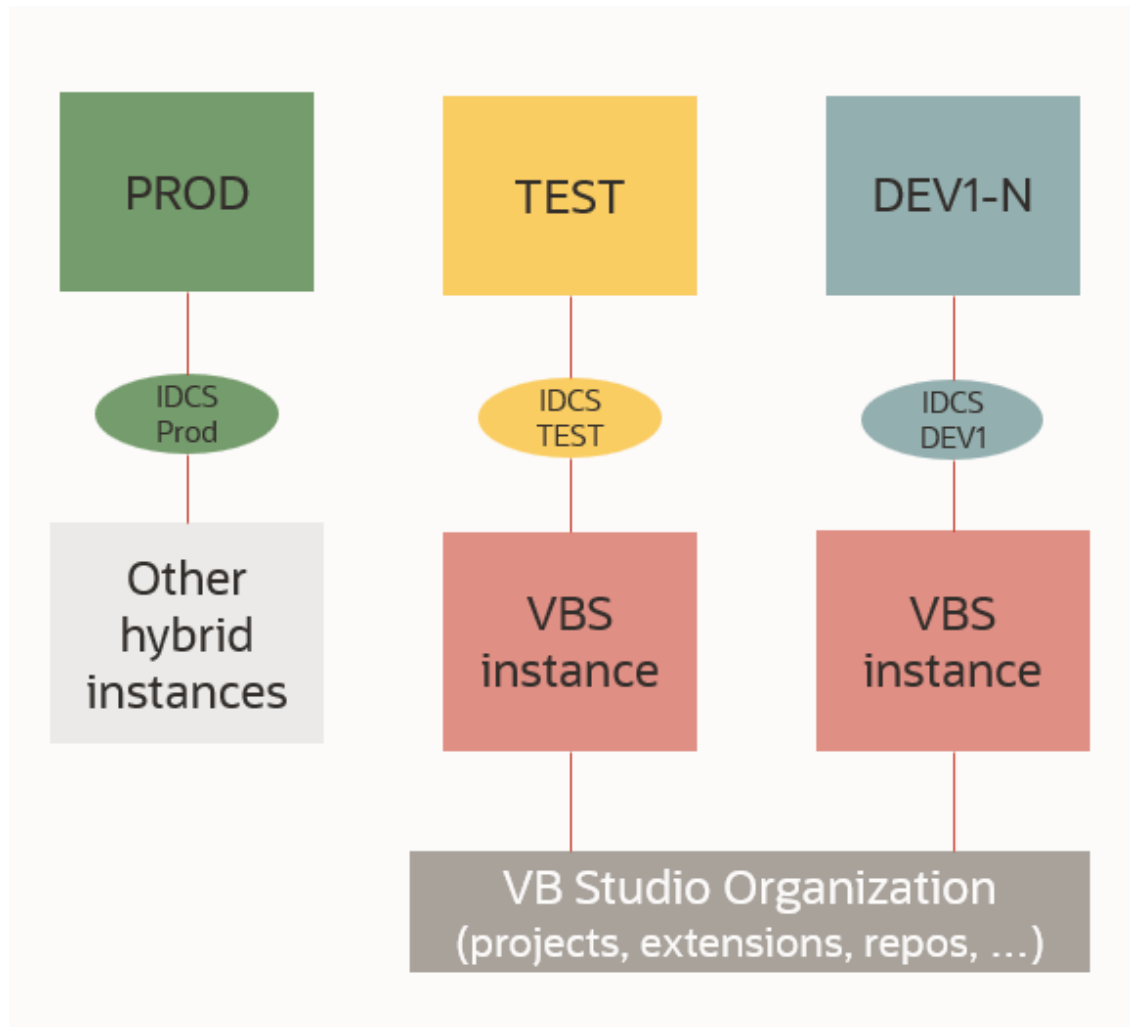
An *extension* is what your team members use to deliver new capabilities into Oracle Cloud Applications. In most cases, an extension contains the work a user has done to customize an app to meet specific business needs. See [What Is an Extension?](#) to learn more.

While this chapter explains everything you need to set up VB Studio for use by others, there are only a handful of activities that **MUST** be done before a user tries to use VB Studio. Those activities are described in [Essential Setup Tasks](#). After you have completed these steps, users need only click the **Edit Page in Visual Builder Studio** button in their Oracle Cloud Application, and VB Studio will guide them through the necessary steps so they can begin their work. Of course, there are a few other activities that you may want to perform after completing the essential tasks. Those activities are described in [Optional Administrator Tasks](#), but they are purely optional.

Work With Multiple VB Studio Instances

Before you begin working through the essential setup tasks, it's important to understand how your Oracle Cloud Application and VB Studio instances are organized.

Oracle is in the process of provisioning multiple instances of Visual Builder Studio for each Oracle Cloud Applications customer. New customers will see this configuration as soon as they are provisioned with Oracle Cloud Applications; existing customers are being migrated to the new landscape over the next several months. In this new configuration, every TEST and DEV instance in your Oracle Cloud Application environment family receives its own instance of VB Studio, which in turn are tied together by a single VB Studio organization:



By sharing a common organization, users working in VB Studio associated with a TEST instance, for example, are able to access the same projects, repositories, and extensions as users working in a completely different instance—say a DEV instance tied to a different VB Studio instance, or another TEST instance. They can see each other's changes, review and approve each other's merge requests, and collaborate on the same wikis, all as though they were in the same VB Studio instance.

Note

If you don't yet have multiple VB Studio instances and would like to, file a service request with Oracle Support.

Here are some important considerations to keep in mind when using multiple VB Studio instances:

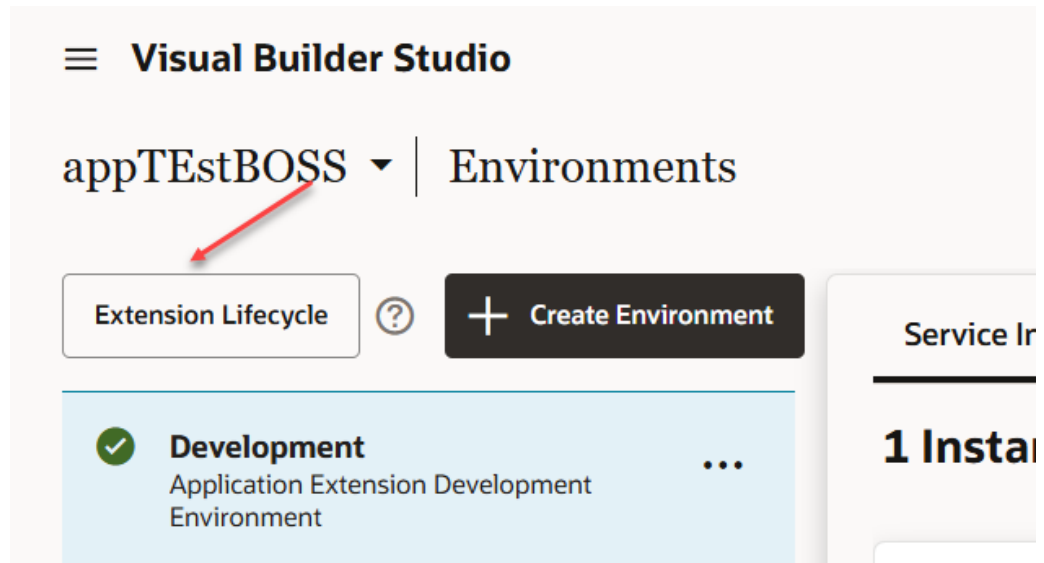
- You can use different VB Studio instances for different purposes. For example, you might maintain one or more extensions using VB Studio in your TEST instance, while simultaneously evaluating new functionality in an upcoming Oracle Cloud Applications version using a different VB Studio (likely one associated with a DEV instance running the newer version).

- Within your Oracle Cloud Applications environment, your identity is different on each VB Studio instance you're working with. For example, in the development instance DEV1, your user name might be known as JoeSmith@<Oracle Cloud Applications-instance-name>-DEV1 to VB Studio; JoeSmith@<Oracle Cloud Applications-instance-name>-DEV2 on the DEV2 instance, and so on. This means that your administrator may need to assign roles for you multiple times, one for each user name/identity. It also means that to work on the same project from both Dev1 and Dev2, **both** of your user names/identities must be added to that project.

Note

Because workspaces operate on a per-user basis, you'll likely have more than one workspace, one for each user name/identity. If you're going to work on the same set of files from more than one instance (that is, from more than one workspace), it's important to publish (or merge) your changes from one workspace before switching to another. This helps to avoid merge conflicts and ensures that you have the very latest changes before you resume your work.

- Having multiple VB Studio instances gives you flexibility in case the instance where you normally work is unavailable for some reason. For example, suppose you usually work in Project A on the DEV1 instance, but DEV1 is down for maintenance. As long as your identity on DEV2 is also a member of Project A—and assuming that you've pushed your latest changes to Git—you can access the project through the VB Studio associated with DEV2 and pick up with your work from there. (See [Set Up VB Studio Users](#) for more about user identities.) Note that both DEV1 and DEV2 must be running the same release of Oracle Cloud Applications in order for this scenario to work properly.
- You must develop your extensions **only** in the VB Studio instance associated with the Oracle Cloud Applications instance you're working in. That is, if you use DEV1 to build your extensions, make sure you're using the VB Studio associated with DEV1, and not, say, the VB Studio instance associated with TEST, or a VB Studio instance in a completely different Oracle Cloud Applications environment family. Of course, you can **deploy** extensions to any instance in any environment family, as long as you have a VB Studio environment defined to support that.
- To deploy an extension to a PROD instance (or to delete an extension when it's no longer needed), you should always use the Manage Lifecycle Extension page. The only reason not to do so is if some of the extension's work has been isolated in discrete branches, which means you'll need to employ CI/CD; in this case, see [Deploy to Production Through CI/CD Pipelines](#).
- If you are developing an extension in one instance and want to test it in another:
 - Define a new environment. The easiest way is to go to the Environments tab in the VB Studio left navigator, then click **Extension Lifecycle**:



If you use the **+ Create Environment** button instead, be sure to select **OAuth 2.0** under Authorization Type:

Regardless of the method you choose, enter the base URL, the name by which you want to refer to the instance, and the credentials of a user authorized to deploy to the instance. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication. Once the instance is connected, these credentials are discarded; they're needed only to set up the initial OAuth-based connection.

- Back in the Designer, press the **Publish** button to commit the changes you made in the extension to the Git repository in VB Studio.
- Use the Manage Lifecycle Extension page to deploy the extension to the environment you just created, as well as to keep tabs on all of your deployments.

Note

If you can't see the deployments for a given instance, it's probably because the instance wasn't created using OAuth. To solve this issue, create a new environment that points to the instance, specifying OAuth 2.0 as the authentication method. (If you create the environment from the Manage Extension Lifecycle page, OAuth is used automatically.)

- As shown in the image above, each non-Prod instance in an Oracle Cloud App environment family is associated with its own VB Studio instance, which in turn is tied to a single organization. If you have the `DEVELOPER_ADMINISTRATOR` VB Studio role for any of these VB Studio instances, you're considered a VB Studio organization administrator. In that capacity, any changes you make to the organization through **any** of the VB Studio instances are reflected on **all** the other instances in the environment family as well.
- Similarly, if you have the OCI policies that allow you to administer VB Studio instances from the Oracle Cloud Console, any change you make through the console—say, enabling CI/CD for a single VB Studio instance—impacts the entire organization. That is, enabling CI/CD for one instance enables it for all instances in the environment family.

Want to create visual applications with one of the VB Studio instances provisioned alongside your Oracle Cloud Applications account? You can certainly do so, but you'll need an instance of Visual Builder to deploy them to. In addition, because each instance of VB Studio is provisioned in the same identity stripe as its associated Oracle Cloud Applications instance (TEST, DEV1, DEVn, and so on), every visual app developer will need Oracle Cloud Application credentials to sign on to VB Studio. Also, should your Oracle Cloud Application instance be down for maintenance or some other reason, the associated VB Studio instance won't be available either, which means visual app development will be halted.

Alternatively, you can use the Oracle Cloud Console to create your own instance of VB Studio to develop your visual apps, if you have the ability to do so, although you will still need a separate instance of Visual Builder to deploy to.

Essential Setup Tasks

Before you can start setting up or using VB Studio, there are a few tasks that must be performed first:

1. Ensure that IDCS user sync is enabled on the IDCS domain associated with **any** instance where you plan to use VB Studio for developing extensions.
Developers who plan to use VB Studio must be created and managed in your Oracle Cloud Application instance. As long as IDCS user sync is enabled, developers created in Oracle Cloud Applications will be synced to IDCS where the required VB Studio roles can then be assigned—either automatically or manually.
2. Make sure you have one of these predefined standard job roles in your development and production instances:
 - Application Administrator (ORA_FND_APPLICATION_ADMINISTRATOR_JOB)
 - Application Developer (ORA_FND_APPLICATION_DEVELOPER_JOB)
 - Human Capital Management Application Administrator (ORA_HRC_HUMAN_CAPITAL_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB)
 - Sales Administrator (ORA_ZBS_SALES_ADMINISTRATOR_JOB)

- Customer Relationship Management Application Administrator (ORA_ZCA_CUSTOMER_RELATIONSHIP_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB)

You'll need these roles to set up VB Studio and your Oracle Cloud Applications instances, and to deploy an extension to an Oracle Cloud Applications environment.

Note that, if you're using custom roles, you'll want to be sure that your custom roles have the privileges required for working in VB Studio. See [Configure Oracle Cloud Applications Custom Roles](#).

3. If IDCS role synchronization is enabled (recommended), the Oracle Cloud Applications roles listed in steps 2 and 3 are sufficient and you should have access to VB Studio after 30 minutes or so. If role synchronization is disabled, you'll need to use the IDCS console and explicitly grant VB Studio access by assigning yourself the DEVELOPER_USER or DEVELOPER_ADMIN VB Studio role in your instance.

See [Assign VB Studio Roles in OCI Identity and Access Management](#). Without these role assignments, the first user (you or one of your users) who begins to create a simple extension will see the error, "You are not a member of this Organization."

If you can't do this yourself, please consult with the infrastructure administrator at your site who has the necessary IDCS identity administrator privileges.

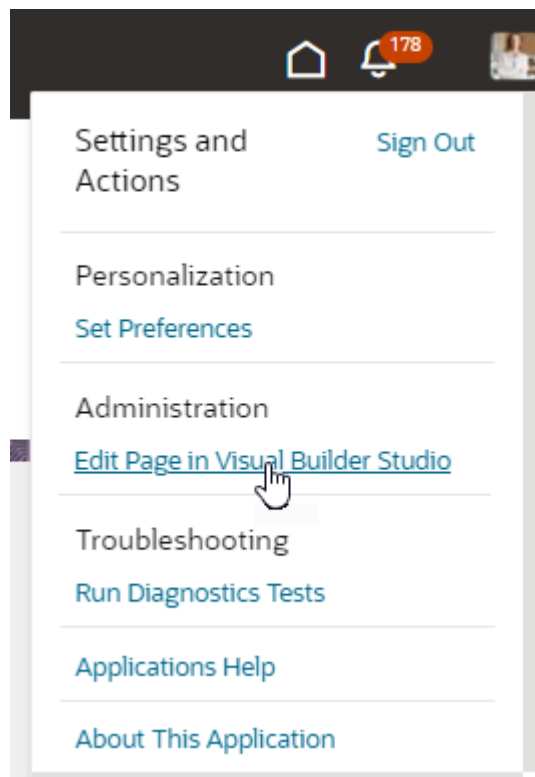
4. If your Oracle Cloud Applications instances restrict ingress IP access, you must add your VB Studio infrastructure IP address to the list of allowed IP addresses in your Web Application Firewall (WAF) and/or Location Based Access Control (LBAC) configuration. If you have done this, but are experiencing access problems, see [Resolve Access Issues After Allow-Listing Your VB Studio IP Address](#).

Create a Simple Extension

The easiest way to set up VB Studio so that others can configure Oracle Cloud Applications pages is to create a simple extension yourself. This lets you ensure everything is working properly.

Before you begin, decide how you want to set up the *projects* for your team:

- In general, it's easiest to use a single VB Studio project for all your extension work, with a different Git repository for each *pillar* (like HCM, SCM, and so on). Using a single project keeps everything together in one place, limits the number of permissions each user requires, and generally makes sense if your team is on the smaller side. This is how VB Studio does things by default, so if this works for your team you can go right to step 1 and get started.
 - If, on the other hand, you have users that need to be isolated from each other's work—say, different consultant organizations doing implementation work for different pillars, or a larger team that uses a common Oracle Cloud Application environment family to develop, test, and host extensions from multiple pillars—you may opt to separate your work by creating one project for each pillar. If this direction makes the most sense for you, go to your Organization page in VB Studio and start creating projects, using Create a Project to guide you. When you're done, proceed to step 1 below.
1. Navigate to a page in your Oracle Cloud Application.
 2. Click your user name at the top-right corner to open the **Settings and Actions** menu, then select **Edit Page in Visual Builder Studio**:



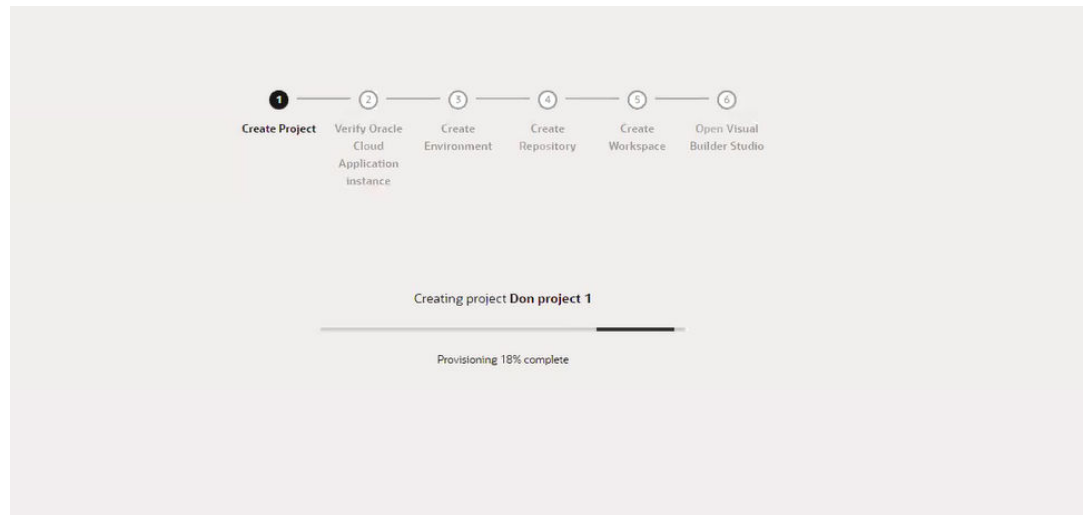
3. In the New App Extension Project dialog, enter a unique name for the project, as well as some team members or groups you think might use this project to extend this Oracle Cloud Application:

A screenshot of the 'New App Extension Project' dialog box. The title bar says 'New App Extension Project' with a close button (X). The main text reads: 'To edit this page, you need to be a member of a Visual Builder Studio project. You're not currently a member of any projects, so enter a name, click Create, and we'll do the rest!'. Below this is a text input field labeled 'Project Name' with a 'Required' label to its right. Further down, it says: 'If you think others may want to work on this Oracle Cloud Application, go ahead and add them as members of your project too.' Below this is another text input field labeled 'Add Members'. At the bottom right, there are two buttons: 'Cancel' and 'Create'.

Note

After you've completed this procedure, make sure that the team members you've added have the proper roles, by following the steps in [Set Up VB Studio Users](#).

VB Studio shows you a status dialog so you can follow along as each asset is created:



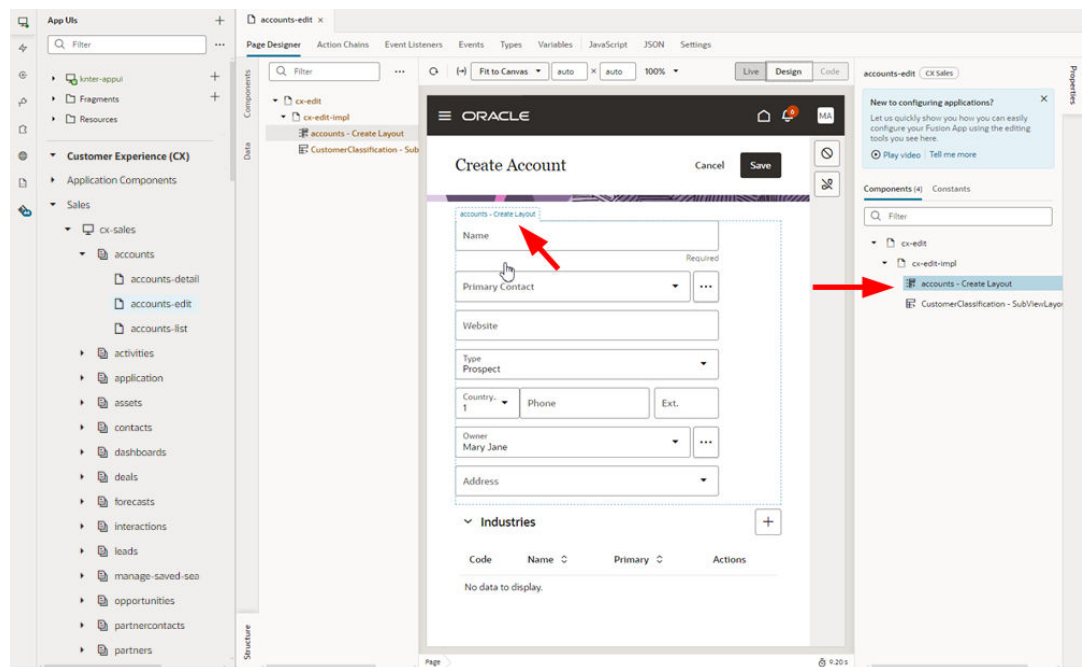
If you're curious about these assets, you can find out more about them in [What Project Artifacts Are Created?](#).

By default, VB Studio creates a private project that's *discoverable*, which means that users you haven't added explicitly to the project will be allowed to choose the project when they click **Edit Page in Visual Builder Studio** from the associated Oracle Cloud Application. Remember, it's best to keep all work for a given Oracle Cloud Application within the same project, so VB Studio will place a "Recommended" badge next to the project's name to encourage users to select it.

Note

The default workspace name generated when you jump over to VB Studio typically uses the pillar of the page you're trying to extend, in the format `Workspace PILLAR`, for example, `Workspace HCM`. If that name already exists, a number is added to the name and incremented as needed, for example, `Workspace HCM 1`, `Workspace HCM 2`, and so on.

4. VB Studio opens the page you were just viewing, in the Designer. When you hover over an extendable component on the canvas, it is outlined in blue, like this:



5. Make a small change to the page, just for testing purposes. You can drag a Heading component onto the page, for example, or add or remove a field on a dynamic form. VB Studio implicitly saves your changes as you work.
6. In the Designer's header, click **Publish**:



7. Since this is just a test, accept the defaults in the Publish dialog and click **Publish**.


By default, your extension's sources are merged to the default branch (`main`) of the workspace's Git repository, then built and deployed immediately to the Oracle Cloud Applications instance within your DEV environment.

Note


If you prefer, you can change the default and instead deploy your changes via a continuous integration and delivery (CI/CD) pipeline. CI/CD pipelines give you the option of deploying an extension to multiple environments as well as automating some lifecycle operations. For more information, see [Use CI/CD Pipelines for Deployment](#).

8. Now click the link in the Publish dialog to see your published changes:

Publish




- ✓ git commit demoBranch
- ✓ git fetch
- ✓ git merge main to demoBranch
- ✓ git merge demoBranch to main
- ✓ git push
- ✓ Deploying extension

 We've deployed your extension **HCM Extension** to <https://myoraclecloudappsinstance.example-test.com:443/fscmUI/redwood>

Branches can't be reused after they're published, so we've created a new one for you: **demoBranch-123**

You can manage your extension's lifecycle on the [Manage Extension Lifecycle page](#).

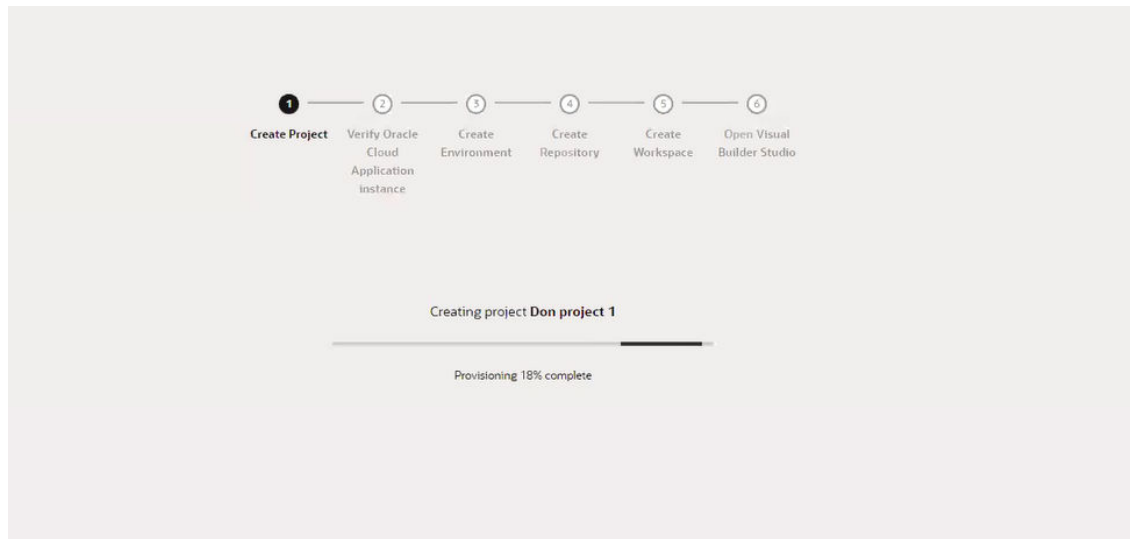


Note

Make sure you copy and paste the deployment URL to your clipboard before you click **Close**. You won't have access to the URL after the Publish dialog is closed.

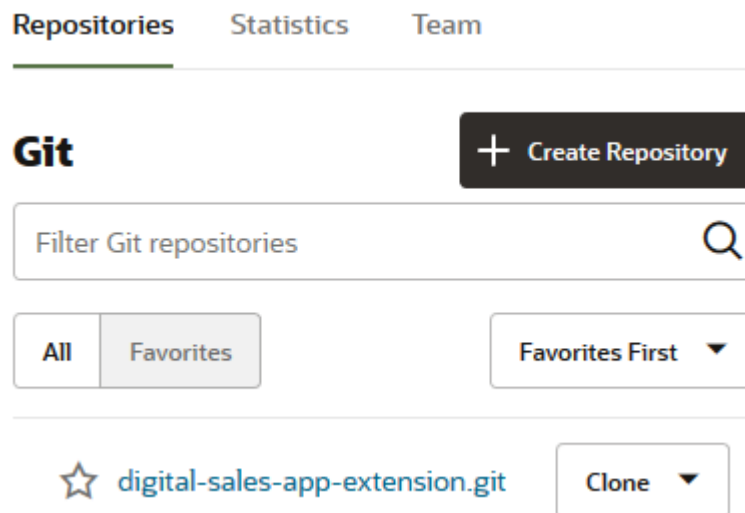
What Project Artifacts Are Created?

When you create a new project, several other VB Studio artifacts are also created for you. VB Studio shows you a status diagram so you can follow along as each artifact is generated:

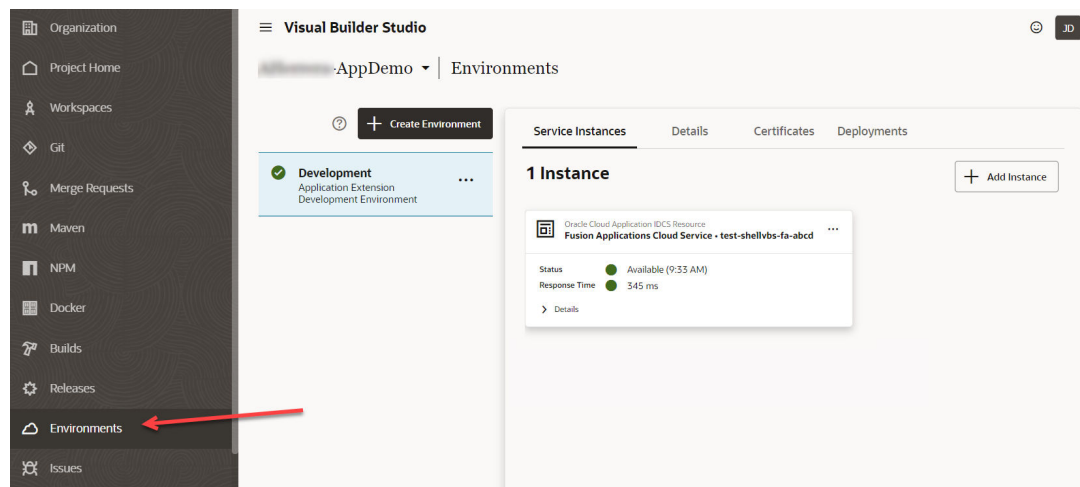


These artifacts include:

- A Git repository, which will contain the source code for all the extensions based on this project.
To see the Git repository's files, go to the **Project Home** page, click the **Repositories** tab, then click the Git repository name:



- A **Development** environment, pointing to the development instance where the Oracle Cloud Application you plan to customize is running. (A VB Studio *environment* can include Oracle Cloud Applications, Visual Builder, or Integration service instances, so you can treat them as a single entity.) To see the definition for the Development environment, click **Environments** in the VB Studio navigator:



- A workspace, which is a private area where you work on your extensions. Each user has a unique workspace, although users can collaborate on the same extension by cloning each other's workspace (which also clones the project's Git repository).
- Build jobs that package and deploy the extension's artifact to Oracle Cloud Application's development instance. The package job, named **extensionName-Package**, generates the extension's artifact file. The deploy job, **extensionName-Deploy**, deploys the extension's artifact file to the Oracle Cloud Application instance defined in the Development environment. (If you [create a project manually](#), these jobs are named **Application-Extension-Package** and **Application-Extension-Deploy**.) To run the package and deploy build jobs in a sequence, the **extensionName-Package and Deploy** pipeline is also created. (If you create a project manually, this pipeline is named **Application Extension - Package and Deploy**.)

Note

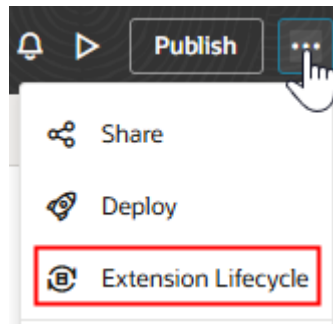
These build jobs and pipelines are automatically created for you, but whether they are enabled or disabled depends on your extension's **Enable CI/CD pipeline** setting.

- A VM executor, but only if this is the first project in this VB Studio's instance and this instance did not already have any VM executors. (Builds in a build system run in *build executors*. The operating system and software packages required to run builds on build executors are defined in *build templates*.) The VM executor uses the **System Default OL7 for Visual Builder** executor template. You can use this VM executor to run build jobs that reference the **System Default OL7 for Visual Builder** template in the current project and other projects as well.

Manage Your Simple Extension

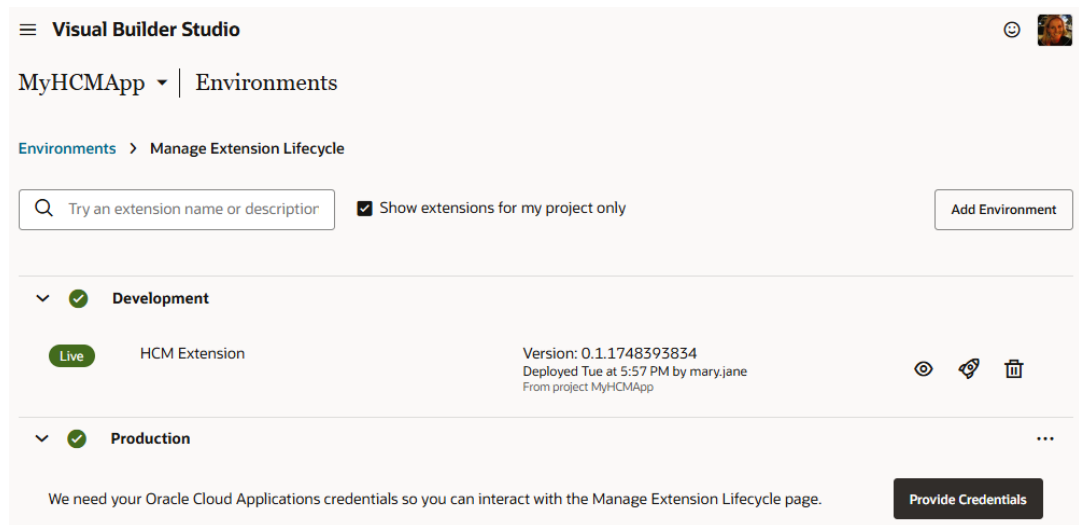
Now that you've created and published your simple extension, let's look at how you can promote it to other environments, and how you can delete it once your setup testing is complete. To do both tasks, you'll use the Manage Extension Lifecycle page.

1. From the menu in the Designer's header, click **Extension Lifecycle**.



You should now see the Manage Extension Lifecycle page, which lists all the environments that are available for you to deploy to.

At this point in your setup testing, you'll probably see only your DEV environment. You can create additional environments (for example, for your TEST and PROD instances) from the Environment's page. See [Add an Oracle Cloud Application Instance to an Environment](#) and [Add the Oracle Cloud Application's Production Instance to an Environment](#). Or, you can add a new environment right from this page (see step 2).



Note

If you see a **Provide Credentials** button next to an environment, you must first sign in before you can work with that environment on this page.


2. If you'd like to deploy to an Oracle Cloud Applications instance but it's not listed on this page, you can add it.

- a. Click **Add Environment**.
- b. In the Add Environment panel, enter basic environment details, such as the environment's base URL, environment name, and user name and password for the target Oracle Cloud Applications instance.

When you add an environment from this page, the authentication method for its Oracle Cloud Applications instance is automatically set to OAuth.

- c. Click **Add**.

You also have the option to add an environment from the Environment's page. See [Add the Oracle Cloud Application's Production Instance to an Environment](#).

3. Once you've got an environment set up, you can now try deploying your simple extension to it:
 - a. Find your simple extension and click .
 - b. Choose the environment that points to the instance you want to deploy to.
 - c. Click **Deploy**.
4. Note that you can **delete** the credentials that let you work with an instance from this page. Why might you do this? Maybe you need to authenticate with the Manage Extension Lifecycle page as a *different* Oracle Cloud Applications user—perhaps someone who has the privileges to deploy and delete extensions. Or, maybe you just need to start over again with a different login.

To delete existing credentials, click the three dots at the end of an environment's row and then click **Delete Extension Lifecycle Credentials > Delete**. The three dots display for an environment as long as the authentication method for its Oracle Cloud Applications instance is OAuth.

Once you delete the existing credentials, you can click **Provide Credentials** to authenticate with this page as a different Oracle Cloud Applications user.

5. As this was just a simple test and not a real extension, you may want to remove it. To

delete the extension, click .

Deleting an extension restores your application to its original state.

For more information, see [Manage Your Published Extensions](#) in *Extending Oracle Cloud Applications with Visual Builder Studio*.

What's Next?

At this point, you have several options:

- Set up the VB Studio users [you added in step 1](#), if you haven't already. (This must be done before your users can access VB Studio.) See [Set Up VB Studio Users](#).
- Take advantage of some the optional settings you can apply. See [Optional Administrator Tasks](#).
- Change some of the project's attributes, like the markup language used to format wiki pages and comments. See [What Else Can You Do with Your Project?](#)

Set Up VB Studio Users

To extend Oracle Cloud Applications applications with VB Studio, you need to assign roles to your users in both Oracle Cloud Applications and IDCS.

Note that in VB Studio, a single user is assigned a unique name to reflect each Oracle Cloud Applications instance where the user is active. For example, UserA in the development instance DEV1 might be known as `UserA@<Oracle Cloud Applications-instance-name>-DEV1` to VB Studio, UserB might be `UserB@<Oracle Cloud Applications-instance-name>-DEV2`, and so on. This means that you may need to assign roles for the same user multiple times, one for each user name/identity.

There are two setup procedures you can use to assign your Oracle Cloud Applications users the suggested roles, depending on whether IDCS role sync has been turned off or not.

- If you're a customer that has *not* disabled role sync, you can follow the instructions in [Assign Oracle Cloud Applications Roles](#). After you assign roles to your users, Oracle Cloud Applications will synchronize those users and their roles with IDCS. There is no need for you to manually assign the VB Studio IDCS roles to your users.
- If you're a customer that *has* disabled role sync between your Oracle Cloud Applications development instance that's associated with VB Studio and its corresponding identity domain, after you follow the instructions in [Assign Oracle Cloud Applications Roles](#), you'll need to log into the IDCS console and follow the instructions in [Assign VB Studio Roles in OCI Identity and Access Management](#) for assigning VB Studio access.

How do you know if role sync has been disabled? There are two ways to find out:

- You can reach out to your infrastructure and identity administrator, who has identity administrator privileges in the identity domain, and have them review this configuration.
- You can wait 30 minutes after you assign the Oracle Cloud Applications roles. If you continue to receive a `You are not a member of this organization` error when attempting to create a simple extension, then role sync isn't enabled (or not working), and you'll either need to re-enable it or assign VB Studio roles directly by using the IDCS console and the instructions in [Assign VB Studio Roles in OCI Identity and Access Management](#).

As an administrator, you must have certain privileges to assign the necessary roles:

- In Oracle Cloud Applications, you must have one of these privileges: cloud administrator, service administrator, or application administrator. Everyone who assigns Oracle Cloud Applications roles to their users needs one of these privileges.
- In IDCS, you must have either an Identity Domain Administrator or User Administrator role. You only need these roles if IDCS role sync has been turned off and you need to assign the VB Studio IDCS roles manually.

This table shows the VB Studio IDCS roles that are mapped to users with Oracle Cloud Application roles:

Oracle Cloud Applications users with these roles:	Need to be mapped to this VB Studio IDCS role:
Application Administrator (ORA_FND_APPLICATION_ADMINISTRATOR_JOB) Human Capital Management Application Administrator (ORA_HRC_HUMAN_CAPITAL_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB) Sales Administrator (ORA_ZBS_SALES_ADMINISTRATOR_JOB) Customer Relationship Management Application Administrator (ORA_ZCA_CUSTOMER_RELATIONSHIP_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB) Synchronization Enabled Administrator Identity (ORA_SYNC_ENABLED_ADMINISTRATOR_ABSTRACT)	VB Studio administrator (DEVELOPER_ADMINISTRATOR)
Application Developer (ORA_FND_APPLICATION_DEVELOPER_JOB) Synchronization Enabled Developer Identity (ORA_SYNC_ENABLED_DEVELOPER_ABSTRACT)	VB Studio user (DEVELOPER_USER)

How Do I Assign the Roles?

If IDCS role synchronization hasn't been disabled for your Oracle Cloud Applications environment, you can follow the instructions in [Assign Oracle Cloud Applications Roles](#), wait a half hour while the roles are synchronized with the IDCS system, and then your users should be able to use VB Studio without any further involvement on your part.

If role synchronization has been disabled, role assignment is a two-step process:

1. Create an Oracle Cloud Applications user, then assign a role to that user using the Oracle Cloud Applications Security Console. See [Assign Oracle Cloud Applications Roles](#).
2. In IDCS, manually assign a VB Studio role (DEVELOPER_ADMINISTRATOR or DEVELOPER_USER) to the Oracle Cloud Applications user. See [Assign VB Studio Roles in OCI Identity and Access Management](#).

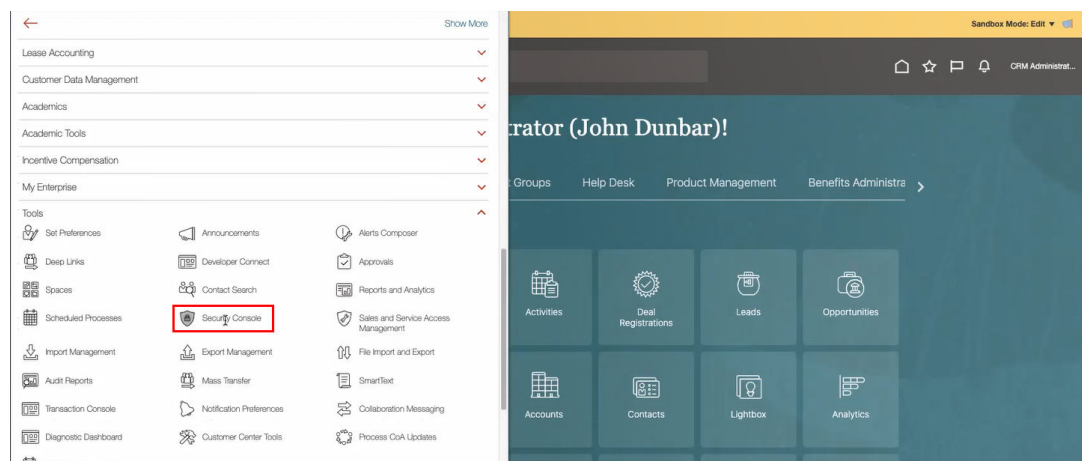
Assign Oracle Cloud Applications Roles

Before beginning the process, review some background information:

- Standard roles (listed in step 7 below) are predefined. Their privileges are automatically updated as necessary, such as when new features or services are added.
- Custom roles are created as substitutes for standard roles, allowing only specific privileges. These privileges are assigned in the Oracle Cloud Applications Security Console.
If you're using custom roles, you'll want to be sure that your custom roles have the privileges and permissions required for working in VB Studio. See [Configure Oracle Cloud Applications Custom Roles](#).

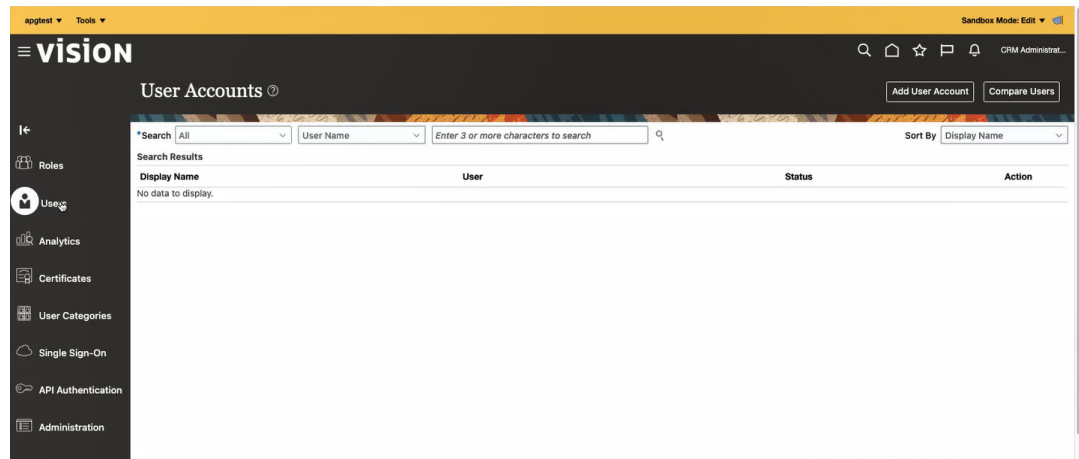
To assign an Oracle Cloud Applications role to a new or existing Oracle Cloud Applications user:

1. Sign in to Oracle Cloud Applications as an administrator with access to the Security Console.
2. Click **Navigation** ≡ menu to open the navigation menu and, under **Tools**, click **Security Console**.



3. Click **Users** in the navigation pane.

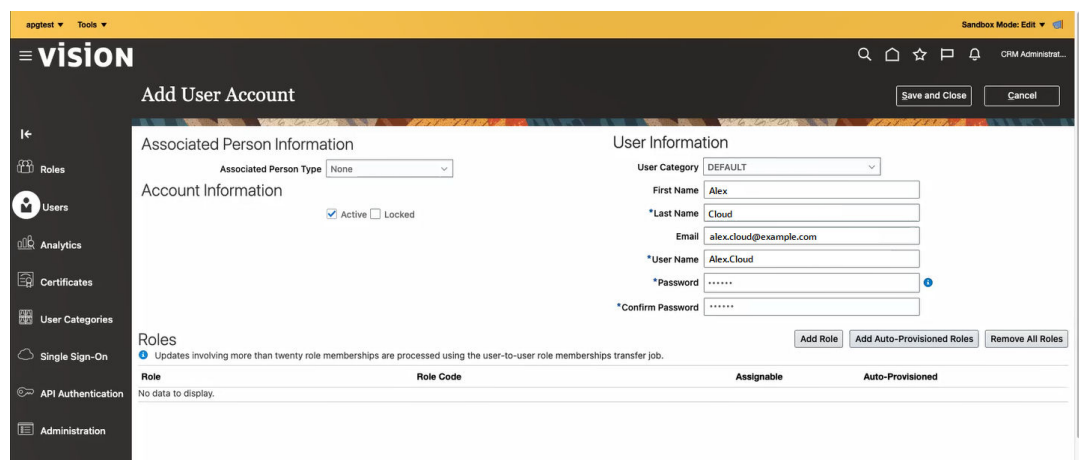
The User Accounts page is displayed.



4. To create a new user, click **Add User Account** and follow the prompts. To assign a role to an *existing* user, use the Search field to find the user you want, then skip to step 6.

The Add User Account page is displayed.

5. Fill out the User Information fields (**First Name**, **Last Name**, **Email**, **Password**, **Confirm Password**).



The **User Name** field has been filled in for you, using the first and last names you entered separated by a period.

6. Click **Add Role**.

The Add Role Membership from Role page is displayed.

7. Select one of the standard Oracle Cloud Applications roles:
 - Application Administrator (ORA_FND_APPLICATION_ADMINISTRATOR_JOB)
 - Application Developer (ORA_FND_APPLICATION_DEVELOPER_JOB)
 - Human Capital Management Application Administrator (ORA_HRC_HUMAN_CAPITAL_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB)
 - Sales Administrator (ORA_ZBS_SALES_ADMINISTRATOR_JOB)

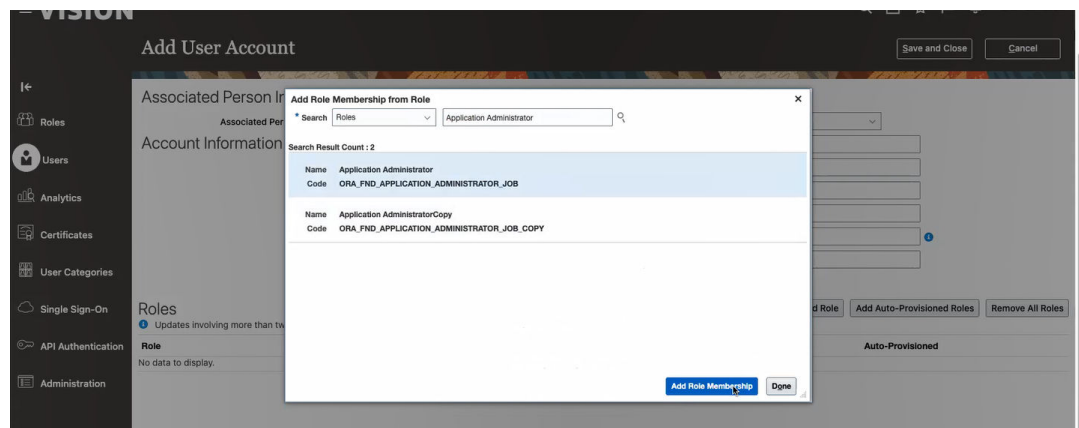
- Customer Relationship Management Application Administrator (ORA_ZCA_CUSTOMER_RELATIONSHIP_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB)

Note

To see the list of assignable roles when extending Oracle Cloud Applications, users need to be assigned a role with the `PER_REST_SERVICE_ACCESS_USERS_AND_ROLES_LOVS_PRIV` privilege, such as `ORA_PER_EMPLOYEE_ABSTRACT` or `ORA_HRC_HUMAN_CAPITAL_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB`.

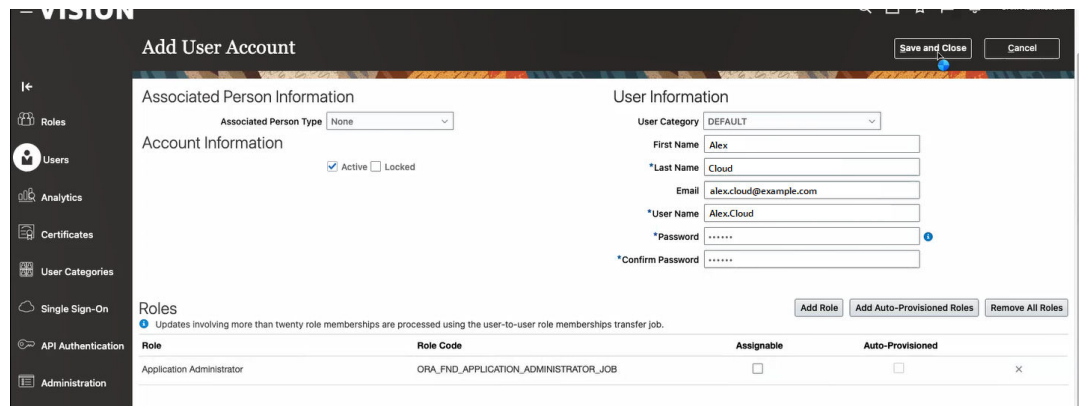
Instead of standard roles, you can also select custom roles if you've created them.

- After you select the role to assign to the user, click **Add Role Membership**.



- Click **Done**.

The Add User Account page is displayed, showing the new user with an assigned role.



- Click **Save and Close**.

From the time role membership was granted via the Security Console, it will take approximately 30 minutes for a user to gain access. If, after 30 minutes, a user is still seeing a

You are not a member of this organization warning message when attempting to create a simple extension, it's very likely that IDCS role sync has been disabled on your environment. In this case, you need assign one of the VB Studio roles to the Oracle Cloud Applications user. See [Assign VB Studio Roles in OCI Identity and Access Management](#).

Assign VB Studio Roles in OCI Identity and Access Management


If you just created a new user in Oracle Cloud Applications, it will take at least 30 minutes for the user profile to show up in OCI Identity and Access Management so that you can assign a VB Studio user role to it.

Once you've waited at least 30 minutes, you can assign VB Studio roles. Here's how:

1. In a web browser, go to <https://cloud.oracle.com>.
To view the list of supported browsers for the Oracle Cloud Console, also called the OCI console, see [Supported Browsers](#).

Note

If you have a direct link to OCI Identity and Access Management (formerly called the IDCS Console), use that instead, make sure you select **Domains**, and then skip to step 7.

2. In **Cloud Account Name**, enter your Oracle Cloud account or tenancy name and click **Next**.
3. On the Single Sign-On (SSO) panel, if required, select the **OracleIdentityCloudService** identity domain and click **Continue**.
This is the identity domain associated with your Oracle Cloud Applications instance.
4. Enter your Oracle Cloud account credentials and click **Sign In**.
If you've recently created your Oracle Cloud account, wait for some time to see your services in the Oracle Cloud Console.
5. In the upper-left corner, click **Navigation Menu** .
6. Select **Identity & Security** and then, under Identity, select **Domains**.
7. On the Domains page, next to **Applied filters**, make sure the root compartment is selected.
8. Click the **OracleIdentityCloudService** identity domain, then click the **Oracle cloud services** tab.
9. Search for and select the service that begins with either **DevServiceAppAUTO_** or **VisualBuilderStudioInstance_**:
 - Select **DevServiceAppAUTO_** for older VB Studio instances that were migrated from Oracle Gen 1 Cloud to Oracle Gen 2 Cloud Infrastructure, also known as Oracle Cloud Infrastructure (OCI).
 - Select **VisualBuilderStudioInstance_** for newly created Gen 2 (OCI) VB Studio instances.

10. Click **Application roles**.

A user must be assigned one of these two roles to access VB Studio, either directly or indirectly via a group.

This VB Studio role...	Enables a user to:
DEVELOPER_ADMINISTRATOR	Set up VB Studio, manage all projects, manage VM executors and executor templates, view the Recent Activities feed, and update the organization details. The user with this role is also called the <i>Organization Administrator</i> . Assign this role to users who can administer VB Studio.
DEVELOPER_USER	Create and access VB Studio projects. All non-admin users of VB Studio must be assigned this role. Note that this role doesn't allow the user to update the organization details and view the Recent Activities feed.

- For each VB Studio role that you want to assign, click the three dots at the end of the row and then click **Manage users**.
- The Manage user assignments page lists the users already assigned to the selected VB Studio role. Click **Assign users** to add a new user to this role.
- On the Assign users page, search for and select the user you'd like to grant this VB Studio role to, then click **Assign**.
- Click **Close**.
- Repeat these steps to assign VB Studio roles to additional users.

Configure Oracle Cloud Applications Custom Roles

Oracle delivers a set of predefined Oracle Cloud Applications *standard* roles that already have the required VB Studio-related privileges. If you use your own *custom* roles, you'll need to manually add those specific privileges so that assigned users can work with extensions.

You create custom roles in the Security Console. Generally, the most efficient approach is to copy a standard role and then edit it to add or remove privileges. If your custom role will have only a few privileges, you can create it from scratch. See [Copy Job Role and Abstract Role](#) and [Create Job Role and Abstract Role from Scratch](#) in *Oracle Fusion Cloud Applications Securing Applications*.

Note

The privileges listed below are required for custom roles assigned to VB Studio users, but don't—by themselves—provide access to VB Studio. For access to VB Studio, see [How Do I Assign the Roles?](#)

Required Privileges

Your custom roles must include these privileges:

- View Administration Link (FND_VIEW_ADMIN_LINK_PRIV) - this privilege allows users to see the **Edit Page in Visual Builder Studio** link on Oracle Cloud Applications pages, if enabled by Oracle. Users click this link to jump over to VB Studio.
- Administer Sandbox (FND_ADMINISTER_SANDBOX_PRIV) - this privilege allows users to deploy and manage the extension lifecycle from VB Studio.

The best practice is to copy and then edit one of these standard roles:

- Application Administrator (ORA_FND_APPLICATION_ADMINISTRATOR_JOB)
- Application Developer (ORA_FND_APPLICATION_DEVELOPER_JOB)
- Human Capital Management Application Administrator (ORA_HRC_HUMAN_CAPITAL_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB)
- Sales Administrator (ORA_ZBS_SALES_ADMINISTRATOR_JOB)
- Customer Relationship Management Application Administrator (ORA_ZCA_CUSTOMER_RELATIONSHIP_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB)

If you copy one of the preceding roles, the resulting custom role will automatically include both the View Administration Link and Administer Sandbox privileges.

In addition to making sure your custom roles have the right privileges, you'll also need to give VB Studio access to your users. See [How Do I Assign the Roles?](#)

Optional Administrator Tasks


Once you've completed the essential tasks and set up one or more projects, there are a few other things you may want to do to tailor operations so they better suit your organization's needs:

- [Set merge restrictions](#) on the `main` branch for the project's Git repository;
- Consider the benefits of using [continuous integration and delivery \(CI/CD\) pipelines](#) to deploy your extensions.
If required, create and configure the build jobs and pipelines for each Oracle Cloud Applications instance you want to deploy to:
 - [Deploy to Production Through CI/CD Pipelines](#)
 - [Deploy to Other DEV and TEST Instances Through CI/CD Pipelines](#)

Set Merge Restrictions on the `main` Branch

By default, the `main` branch is accessible to all project users. To control who can push commits to `main`, you may want to set restrictions on it and allow merges only after they are approved. This may be especially important if you have users operating in Express mode.

Be sure to repeat this process for each project you created in [Create a Simple Extension](#).

1. In the left navigator, click **Project Administration** .
2. Click **Branches**.
3. In **Repository** and **Branches**, select the Git repository and the `main` branch.
4. Select the **Requires Review** option.
5. In **Default Reviewers**, enter and select the users.

A default reviewer is a project member who is automatically added as a reviewer when a merge request is created on the branch.

6. From the **Approvals** drop-down list, select the minimum number of reviewers who must approve the review branch of a merge request, where the selected branch is the target branch
7. (Optional) To allow a review branch to be merged to the selected branch only if the last build of the linked job in Merge Request is successful, select the **Require successful build** check box.
To use this option, link a build job to a merge request.
8. (Optional) To reset the approval status of reviewers if change is pushed to a branch after they have approved the merge request, select the **Reapproval needed when branch is updated** check box.
9. (Optional) To ensure changes pushed to the target branch match the contents of the review branch, select the **Changes pushed to target branch must match review content** check box.
10. (Optional) In **Merge Request Exempt Users**, specify users who can bypass the branch restrictions and merge the review branch of a merge request outside VB Studio or without required approvals.

This is useful if you want to allow some users to merge the review branch irrespective of review conditions being met.

11. Click **Save**.

Use CI/CD Pipelines for Deployment

VB Studio gives you the option to use CI/CD pipelines to deploy an extension to your Oracle Cloud Applications environments, whether it's to your DEV or PROD instance.

- An extension is initially set up so that you can quickly deploy your changes to its target Oracle Cloud Applications instance. By default, when you click **Publish** in the Designer, your extension is merged to your workspace's `main` branch and then deployed *directly* to the environment associated with your workspace.

You can deploy an extension to additional instances, also with a click of a button, from the Manage Lifecycle Extension page.

- CI/CD pipelines, meanwhile, let *you* decide how an extension is deployed. When clicking **Publish**, for example, maybe you'd like to deploy that extension to multiple instances at once—not just to a DEV instance, but perhaps to a TEST instance as well. And, maybe you'd also like to automate some lifecycle operations (for example, you can deploy dependent artifacts to the target environment in parallel or delete older build artifacts to clean up the environment).


You can set this up by enabling a CI/CD pipeline for the target branch you want to merge to, and configuring that pipeline to suit your organization's needs.

Note

Use a combination of whichever deployment options work best. For example, in your organization, maybe it's most efficient to publish extensions directly using the **Publish** button, and to use the Manage Extension Lifecycle page for additional deployments, as needed. At the same time, you can leverage the use of CI/CD pipelines to automate certain lifecycle operations tasks.

Here's a brief rundown of a few key differences between the two publishing options:

Publishing Aspect	Publish Directly	Publish via CI/CD Pipelines
Deployment flexibility	The key benefits of publishing an extension <i>directly</i> to Oracle Cloud Applications instances—using both the Designer's Publish button as well as the Manage Extension Lifecycle page—are speed and simplicity. The deployment process is straightforward, but not flexible.	Flexibility is the main reason to use CI/CD pipelines. When clicking Publish , maybe you'd like your extension to be deployed not just to a DEV instance, but to other instances as well. You can also do things like configure a pipeline to download archived artifacts or discard old builds and artifacts, for example, or run builds on a specified schedule.
Speed	Publishing an extension directly to an Oracle Cloud Applications instance is fast. Deployment happens immediately, so end users can quickly view the changes.	When publishing an extension using a CI/CD pipeline, the deployment process isn't immediate. You might need to wait for the entire process to complete before others can see the changes.
Deploying to multiple instances	When working with multiple Oracle Cloud Applications instances, you must first publish your extension to your DEV instance. To deploy that extension to other instances, you can then use the Manage Extension Lifecycle page. (Deployments from this page reuse the build artifact that was previously packaged and published to your DEV instance.)	You can configure CI/CD pipelines to deploy changes to any instance, not just to your DEV instance. You can also configure a pipeline to deploy to multiple instances simultaneously.
Merge requests	Since deployment is immediate, you can't include merge requests as part of the publishing process. To get changes reviewed, therefore, merge requests must be created <i>before</i> clicking Publish .	When publishing an extension using a CI/CD pipeline, you do have the option to build in merge requests.

Publishing Aspect	Publish Directly	Publish via CI/CD Pipelines
Approval workflow	When publishing directly to an Oracle Cloud Applications instance, there aren't any built-in approval workflow capabilities. If you have access to an environment, you can deploy to it.	With CI/CD pipelines, you can add an approval item that requires one or more authorized users to manually approve a step before executing the rest of its run. For example, a pipeline can automatically deploy an extension to your DEV and TEST instances, but require a manager's approval before deploying to your PROD instance.
Logs and build history	Publishing directly to an Oracle Cloud Applications instance doesn't provide a record of build details, log reports, and build history.	If you've used a CI/CD pipeline to deploy an extension, prior logs and build history are available for review.
Support for rolling back extensions to a previous version	If an extension was published directly to an Oracle Cloud Applications instance, that extension can't be redeployed at a later time.	If an extension was deployed using a pipeline and its build artifacts were archived, it can be redeployed at a later time, if needed.
Delete capability	If you need to delete an extension, the best practice is to click  on the Manage Extension Lifecycle page.	You can configure a job to delete an extension and include that job in a pipeline, but deleting an extension using a job doesn't delete the whole extension, only a specific version. It's cleaner and therefore a better practice to use the Manage Extension Lifecycle page to delete extensions.

If you decide you'd like to use a pipeline to deploy your extension to the Oracle Cloud Applications instance (typically a DEV instance) associated with your workspace, you'll need to change the **Enable CI/CD pipeline** setting in the Designer; this setting is disabled by default for your workspace's `main` branch. See [Enable or Disable the CI/CD Pipeline for Publishing in Extending Oracle Cloud Applications with Visual Builder Studio](#). You can then configure the pipeline as needed.

For subsequent deployments to additional Oracle Cloud Applications instances, you should use the Manage Lifecycle Extension page. You do have the option, however, to create and configure build jobs and pipelines for each additional instance you want to deploy to, as needed:

- [Deploy to Other DEV and TEST Instances Through CI/CD Pipelines](#)
- [Deploy to Production Through CI/CD Pipelines](#)

Deploy to Production Through CI/CD Pipelines

By default, a project deploys extensions to the Development environment, which is the Oracle Cloud Application instance you were using when you clicked **Edit Page in Visual Builder Studio**. Follow these steps to set up a CI/CD pipeline to deploy to the PROD instance, so users can deploy to it after their development and test cycles are complete.

 **Tip**

If you don't need a CI/CD pipeline to deploy your extension, skip this section and use the Manage Extension Lifecycle page to deploy to PROD. From this single page, you can add a new environment for your PROD instance and deploy extensions to that instance without the need to create build jobs and pipelines.

The Oracle Cloud Applications PROD instance resides in a different identity domain from the DEV and TEST instances, which means you'll need the URL for the PROD instance, as well as user credentials that VB Studio can use to deploy to this instance. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.

To set up a project to deploy to PROD:

1. [Add the PROD instance to an environment](#)
2. [Create a production branch](#)
3. [Create and configure production build jobs](#)
4. [Configure the pipeline so that the packaging and deployment jobs run in succession](#)
5. [Run the production pipeline.](#)

Add the Oracle Cloud Application's Production Instance to an Environment

Only one Oracle Cloud Applications instance can be added to an environment. As the Development environment already points to the TEST instance (or DEV, if you asked Oracle to re-associate it), you must create a new environment for the PROD instance.

To create a new environment for the PROD instance:

1. In the VB Studio navigator, click **Environments**.
2. Click **+ Create Environment**.
3. In the Service Instances tab, click **+ Add Instance**.
4. In the Add Service Instance dialog box, select **Oracle Cloud Applications**.
5. Under Add Instance Using, select **Application Credentials**.
6. In **Base URL**, enter the Oracle Cloud Application's base URL.
7. In **Instance Name**, if required, update the instance's display name. The name will be displayed in the **Service Instances** tab.
8. In **Username** and **Password**, enter the credentials of a user who can access this Oracle Cloud Applications instance.
These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.
9. Under Authorization Type, select **OAuth 2.0** (default) and click **Add**.

 **Note**

OAuth is the recommended authorization type. Use Basic authentication only if you run into issues with setting up an OAuth connection.

10. If you chose **OAuth 2.0** as the Authorization Type, you may be prompted for authorization to acquire OAuth tokens for accessing the target instance. Click **Authorize** when prompted, then sign in with credentials to access the Oracle Cloud Applications instance. Note that the username that shows in the environment Details section isn't the VB Studio user, but the user for the connected instance whose credentials were used to create the connection's refresh token. The details don't show the VB Studio username, which isn't relevant for a 3-legged OAuth Environment connection.

OAuth tokens (access and refresh) are cycled during regular use. A refresh token is used to obtain an access token whenever a user accesses the target instance. This refresh token is typically valid for seven days. (The token expiration time is set in the IDCS resource app and may be different based on your security requirements.) If the user authenticates with the target instance within the seven-day period, the active refresh token generates a new access token and a new refresh token. This cycle continues indefinitely as long as the refresh token stays valid. If the refresh token expires during extended periods of inactivity (say, when you're away on vacation), click **Actions ***** and select **Renew OAuth Access**.

 **Note**



Service administrators can control the OAuth access or refresh token's expiration in the IDCS resource app, specifically *Fusion Applications Cloud Service* under Oracle Cloud Services in your identity domain's resource app. See [Edit High-Level Information for Oracle Applications](#).

If the newly added instance stays in the **Unknown** status for some time, it usually indicates that the IDCS Application provisioning may have failed. VB Studio added the Oracle Cloud Applications instance, but can't access it. In this case, click **Actions ***** and select **Remove** to remove the Oracle Cloud Applications instance from the environment, and then click **Add** to add it again.

Once you're done, the Recent Activities feed on the **Project Home** page lists the newly created environment along with the service instance you added.

Create a Production Branch

Follow your organization's guidelines to create a branch and protect it from unverified changes. To protect the branch, you can set merge restrictions, make the branch private, and restrict who can push commits to it, or freeze it.

1. In the left navigator, click **Git** .
2. Click the **Refs** view and then click **Branches** .
3. From the **Repositories** drop-down list, select the repository.
4. Click **+ Create Branch**.
5. In the New Branch dialog box, in **Name**, enter the branch name. From the **Base** drop-down list, select the `main` branch as the base branch.
6. Click **Create**.

After creating the production branch:

- Any changes pushed to the `main` branch aren't automatically available in the production branch.

You must create a merge request or manually push the changes to the production branch. Or, you can set this branch as the target branch when you publish, and create and enable a CI/CD pipeline for the branch. See [Enable or Disable the CI/CD Pipeline for Publishing](#).

- If you want to set merge restrictions on the production branch, see [Set Review and Merge Restrictions on a Repository Branch](#). To freeze the branch or make it private, or set other restrictions, see [Protect a Branch](#).

Create and Configure Production Build Jobs

To deploy extensions to your Oracle Cloud Application's PROD instance, you can either use the Manage Extension Lifecycle page or set up a CI/CD pipeline. If you want to use a pipeline, you'll need to set up some packaging and deployment jobs.

1. Migrate the configurations to the production Oracle Cloud Application instance. See [Overview of Configuration Life Cycle](#) and [Overview of Migration](#) in *Configuring and Extending Applications* for instructions.
2. Create a build job that packages the extension. See [Create the Production Packaging Build Job](#) for instructions.
3. Create a build job that deploys the extension to the production instance. See [Create the Production Deployment Build Job](#) for instructions.
4. (Optional) Restrict who can see or edit the production build jobs or run their builds. See [Configure Job Protection Settings](#) for instructions.
5. Configure the pipelines to run the packaging and deployment jobs in succession. See [Create and Configure the Production Pipeline](#) for instructions.
6. Run the production pipeline to package the extension and deploy it to the production instance. See [Run Production Pipelines](#) for instructions.


Before You Configure Build Jobs and Pipelines


Here are some things you need to know before you configure and run build jobs and pipelines:

- Make sure that the source and target instances are of the same release, with the same standard and one-off patches applied to both environments.
- If you configured the development packaging job to overwrite the application's version defined in `visual-application.json`, get the new version. You'll configure the production's packaging job to use the same version.
- VB Studio can create and enable the build jobs and pipeline for you, instead. In the extension's Settings editor, under Building and Publishing, select the production branch and then click **Create CI/CD Pipeline**. You can then modify the newly created build jobs as needed. For example, in the deployment job, the deployment target is always the environment associated with the workspace, so you'll need to change the target instance to your production instance.

Create the Production Packaging Build Job

The packaging job generates an extension artifact that's ready to deploy to the production instance.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.

4. In **Description**, enter the job's description.
5. In **Template**, select the **System Default OL7 for Visual Builder** template.
6. Click **Create**.
7. Click **Configure** .
8. Click the **Git** tab.
9. From the **Add Git** list, select **Git**.
10. In **Repository**, select the Git repository. In **Branch or Tag**, select the production branch.
11. Click the **Steps** tab.
12. From **Add Step**, select **Application Extension**, and then select **Package**.
13. By default, the packaging step minifies the application's source code before running the build. If you don't want to minify the source files, deselect the **Optimize extension** check box.

Minification is a process to remove the unnecessary characters (such as blank spaces, new lines, and comments) from the source code and reduce the size of the files, making the transfer of files consume less bandwidth and storage.


 **Note**

When you deselect the **Optimize extension** check box, this warning is displayed: **Optimization not selected. Packaging without optimization can cause performance problems, so avoid doing so unless you're debugging or troubleshooting.**

14. (Optional) If you want to change the artifact file's name, in **Build Artifact**, enter the new name. The best practice is to keep the default, `extension.vx`, but you can change it.



If you change it, you must also use the new name in the **Files to archive** field (step 18) as well as in the deployment job's **Build Artifact** field. See [Create the Production Deployment Build Job](#).
15. (Optional) If you configured the development packaging job to overwrite the extension's default version defined in the `visual-application.json` file, specify the same version in **Extension Version**.
16. Click the **After Build** tab in the Job Configuration window.
17. From **Add After Build Action**, select **Artifact Archiver**.
18. In **Files to archive**, enter the build artifact name.

You can use wild characters. For example, `*.vx`. Make sure that you include the path for the Application Extension build artifact.

If you entered a value in the **Target Directory** field in **Copy Artifacts** for the deployment job, it is considered to be a subdirectory of the workspace and should be part of the artifact's path in the **Build Artifact** field.
19. If you want to discard the build's old artifacts, click **Settings** . In the **General** tab, select the **Discard Old Builds** check box and specify the discard options.
20. Click **Save**.

Create the Production Deployment Build Job

The deployment job deploys the extension's artifact that was generated in the packaging job to the Oracle Cloud Application's production instance. Before you create the job, make sure you have credentials that VB Studio can use to access the Oracle Cloud Application's PROD instance.

1. In the left navigator, click **Builds** .
 2. In the **Jobs** tab, click **+ Create Job**.
 3. In the New Job dialog box, in **Name**, enter a unique name.
 4. In **Description**, enter the job's description.
 5. In **Template**, select the **System Default OL7 for Visual Builder** template.
 6. Click **Create**.
 7. Click **Configure** .
 8. Click the **Before Build** tab.
 9. From **Add Before Build Action**, select **Copy Artifacts**.
 10. In **From Job**, select the packaging job that generated the extension's artifact.
 11. In **Which Build**, select one of the following:
 - Last successful build (default)
 - Last keep forever build
 - Upstream build in this pipeline instance
 - Specified by permalink
 - Specific build
 - Specified by build parameter
- Depending on what you select, you may be prompted to select which permalink, build, or build parameter you want to be used.
12. Leave other fields with their default or empty values.
 13. Click the **Steps** tab.
 14. From **Add Step**, select **Application Extension**, and then select **Deploy**.

This image shows the Deploy build job page that's been partially filled in.

Application Extension Deployment ⓘ

Target Instance *
Fusion Applications Cloud Service

Authorization

☒ Use OAuth ☐ Use Basic

ⓘ Authorization is required. Authorize

Build Artifact *
extension.vx

15. In **Target Instance**, select the environment with the Oracle Cloud Application's production instance.
16. In the Authorization section, specify the authorization type to run this build step. With **Use OAuth** selected by default, you'll see the *Authorization is required* message, indicating that this build step needs a one-time authorization to handle OAuth requests to your environment's Oracle Cloud Applications instance. Click **Authorize** and enter credentials to access your Oracle Cloud Applications instance; you can also run the job manually and enter the credentials when prompted.

Either way, it is recommended that you authorize your OAuth connection during initial configuration. If you skip this step, you won't be able to publish your changes from the Designer and will need to complete the required authorization before attempting to deploy changes.

Once authorized, the *Authorization has been provided* message shows.

ⓘ **Note**

OAuth is the recommended authorization type. Use Basic authentication only if you run into issues with setting up an OAuth connection. To use Basic authentication, select **Use Basic**, then enter the credentials of a user who can access the Oracle Cloud Applications instance in **Username** and **Password**. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.

OAuth tokens (access and refresh) are cycled during regular use. A refresh token is used to obtain an access token whenever a user accesses the target instance. This refresh token is typically valid for seven days. (The token expiration time is set in the IDCS resource app and may be different based on your security requirements.) If the user authenticates with the target instance within the seven-day period, the active refresh token generates a new access token and a new refresh token. This cycle continues indefinitely as long as the refresh token stays valid. If the refresh token expires during extended periods of inactivity (say, when you're away on vacation), click **Renew Authorization** (or run the job manually, so you're prompted to authorize any expired OAuth tokens).

The **Build Artifact** field should show the same artifact name that was used in the packaging build step. Confirm this value, especially if the packaging job used an artifact name other than the default, `extension.vx`.

17. Click **Save**.

Note

If you develop an extension on, say, 24D in your Test environment, then want to deploy the extension to your 24C Prod environment, you'll have to wait until your Prod instance has been upgraded to 24D before you can deploy successfully. In most cases, there shouldn't be more than a two week gap between pod upgrades.


Configure Job Protection Settings

To restrict access, the project owner can mark a job as private. Users that don't have access can see the build job in the **Jobs Overview** page, but they can't see the **Job Details** page or view the build's details; nor can they see or edit the job configuration, or delete/enable/disable the build job. In addition, the project owner can use a glob pattern that is defined in a rule to protect any job whose name matches the specified pattern.

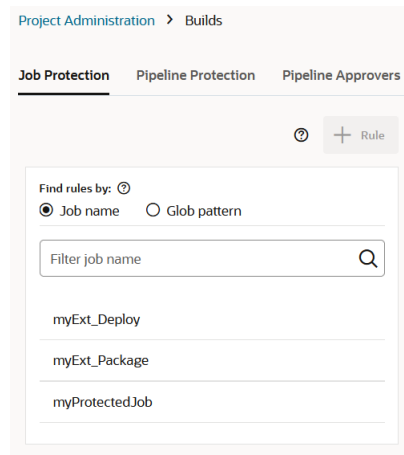
Note


Before you apply any protections to a job, you should consider the following:

- A protection rule defined with a glob pattern will not overrule a job protection defined by using a name (no glob pattern or rule).
- A protection that is applied to a single job will override a protection applied by using a rule (defined by a glob pattern).
- When two rules are combined, the protection is determined by the most restrictive rule. You need to look at the events in the Activities feed and examine the notifications, which provide the information explaining the restrictions when one rule overrides another.
- A job will not be created if the user that is creating the job wouldn't be able to access their own job. The same principle is true for renaming jobs.

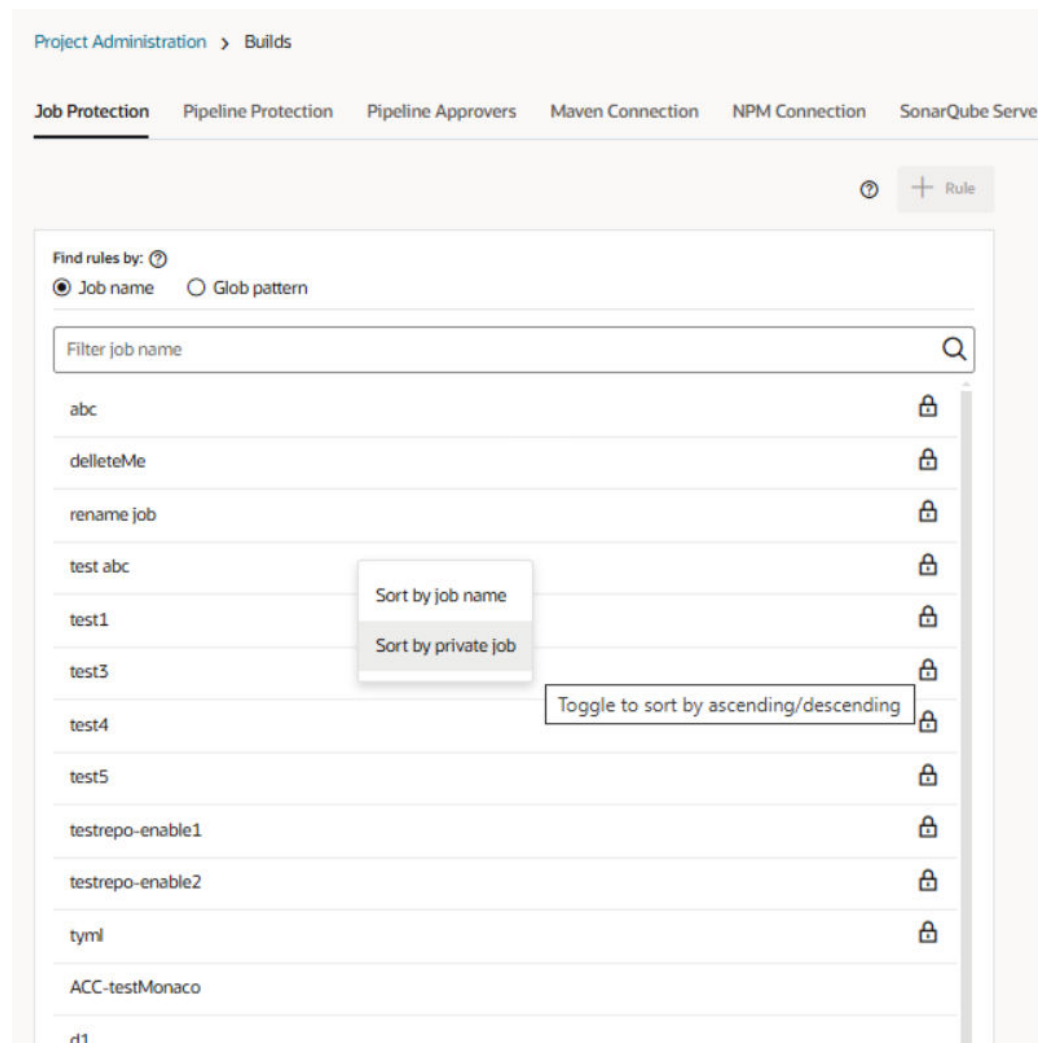
1. In the left navigator, click **Project Administration** .
2. Select the **Builds** tile.
3. Select the **Job Protection** tab.


The Job Protection page is displayed.



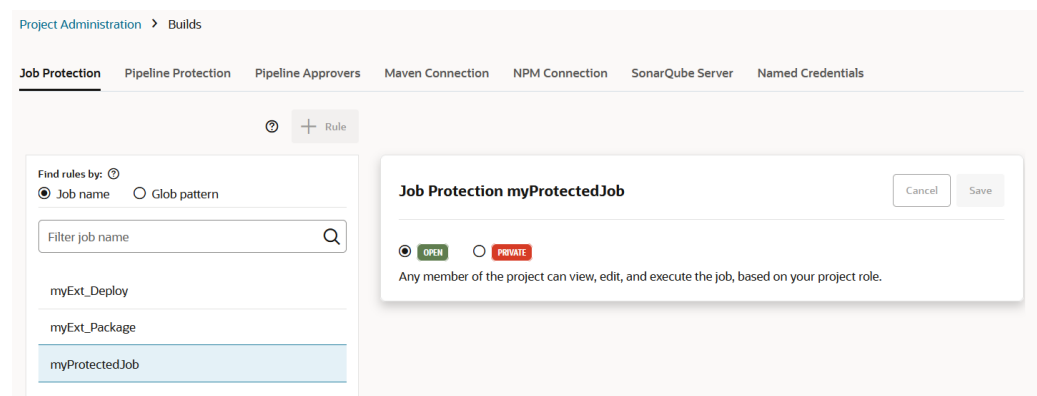
4. In the Find rules by panel, located above the jobs/rules list, select one of these radio buttons:
 - Select **Job name** to choose a job from the list.
If your project has many jobs, you may have difficulty finding the specific job you want to protect. Use the **Filter job**  bar to quickly locate the job to which you want to add the restricted settings.

Right-click the jobs list to display additional sorting options. Click **Sort by job name** or **Sort by private job** to sort the list accordingly. Re-click the sorting option to toggle between ascending and descending order.



If a job has a lock icon , it has already been protected. A protected job's restrictions can still be modified, removed, or the list of authorized users and groups can still be changed.

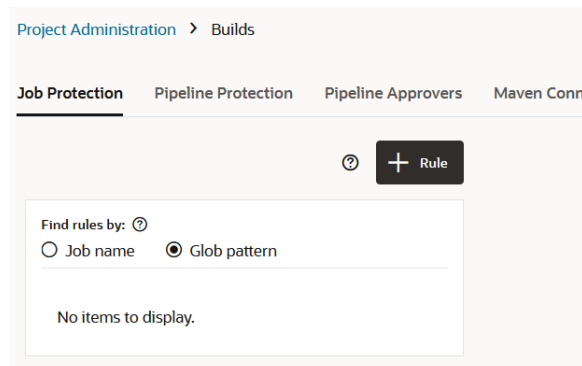
The **Job Protection** dialog box is displayed.



When a job is not directly protected but is protected by a rule instead, an informational message like the following one will show the rules, <ExampleRegex05> in this case, that protect the specific job:

This job is protected by the following glob pattern rules matching this job name: <ExampleRegex05>

- Select **Glob pattern** to specify a string that is matched against the job name. This is what you'd see if no rules have been defined yet.



The glob syntax can be used to specify pattern-matching behavior. These wildcard characters can be used in glob patterns: *, **, ?, [], {}, and \.

Either select an existing protection rule from the list or click **+ Rule** to display the **New Protection Rule** dialog and create a new one.

The Protection Rule dialog box is displayed.

Here we've entered a name (Test Rule) and a glob pattern (test*) and we're about to press **Create** to create a new job protection rule.

5. Select the **PRIVATE** check box.

This is what you see after selecting the Private option for a job.

Job Protection myProtectedJob Cancel Save

☐ OPEN
 ☒ PRIVATE

☐ Allow any member of the project to manually start this private job
☐ Allow commits and triggers to start this private job

Only authorized users and groups will be able to view the "Job Details" page or edit the job.
 Only authorized users and groups will be able to manually run the job.
 If the job is triggered in a pipeline by an unauthorized user or group, it will not be initiated.
 If the job is triggered by SCM or timer it will not be initiated.

Authorized Users/Groups *

With just this option selected, only authorized users and groups will be able to view the Job Details page, edit the job, or manually run it. If the job is triggered in a pipeline by an unauthorized user or group, or if it is triggered by SCM or a timer, it will not be initiated.

This is what you see after selecting the Private option for a protection rule.

Protection Rule Test Rule Cancel Save

☐ OPEN
 ☒ PRIVATE

Provide glob pattern * ?
test*

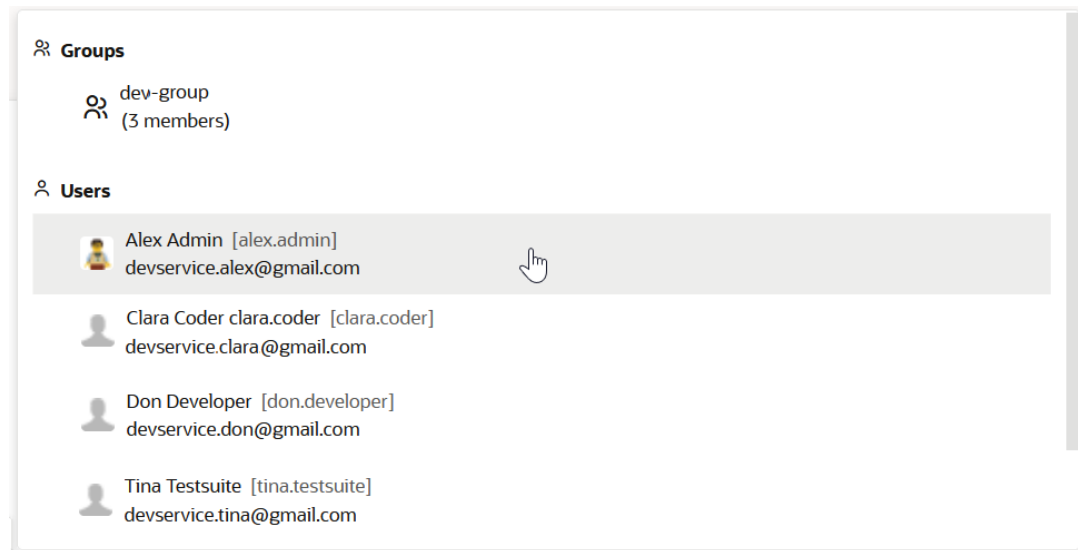
☐ Allow any member of the project to manually start any job matching this glob pattern
☐ Allow commits and triggers to start any job matching this glob pattern

Only authorized users and groups will be able to view the "Job Details" page or edit the job matching this glob pattern.
 Only authorized users and groups will be able to manually run the job matching this glob pattern.
 If the job is triggered in a pipeline by an unauthorized user or group, it will not be initiated.
 If the job is triggered by SCM or timer it will not be initiated.

Authorized Users/Groups *

- Click in the **Authorized Users/Groups** field to display a dialog that lists the project's **Groups** and **Users** you can select from.

Under Users, you can see a flattened list of all users that are members of the group(s) as well as ones that were added individually. For example, the dev-group members (Clara Coder, Don Developer, and Tina Testsuite) appear in the Users list, along with Alex Admin, who was added individually. From the list, select one or more groups and/or users. Don't forget to add yourself.



This is what you would see for the myProtectedJob job after selecting Alex Admin as an authorized user.



This is what you'd see for the Test Rule protection rule after selecting Alex Admin as an authorized user.

Protection Rule Test Rule

Cancel Save

☐ OPEN
 ☒ PRIVATE

Provide glob pattern * ?
 test*

☐ Allow any member of the project to manually start any job matching this glob pattern
☐ Allow commits and triggers to start any job matching this glob pattern

Only authorized users and groups will be able to view the "Job Details" page or edit the job matching this glob pattern.
 Only authorized users and groups will be able to manually run the job matching this glob pattern.
 If the job is triggered in a pipeline by an unauthorized user or group, it will not be initiated.
 If the job is triggered by SCM or timer it will not be initiated.

Authorized Users/Groups *
 Alex Admin [alex.admin] X

7. Select the checkboxes to allow project members to manually start private jobs and/or allow commits and triggers to automatically start private jobs:
- Select the **Allow any member of the project to manually start this private job** checkbox to allow any project member, not just authorized users, to manually start the job.

This what you'd see after selecting the **Allow any member of the project to manually start this private job** check box for the myProtectedJob job.

Job Protection myProtectedJob

Cancel Save

☐ OPEN
 ☒ PRIVATE

☒ Allow any member of the project to manually start this private job
☒ Allow commits and triggers to start this private job

▲ SCM commits and triggers will automatically run the job

Only authorized users and groups will be able to view the "Job Details" page or edit the job.
 Any member of the project will be able to manually start the job.

Authorized Users/Groups *
 Alex Admin [alex.admin] X

Notice that when you select the first checkbox, VB Studio automatically selects the second checkbox, which allows commits and triggers to start the private job, and grays it out. With this setting, only authorized users and groups can view the Job Details page or edit the job, but any project member can start and run the job. In addition, SCM commits or triggers will also automatically start and run the job.

This what you'd see after selecting the **Allow any member of the project to manually start this private job** check box for the Test Rule protection rule.

Protection Rule Test Rule

CancelSave

☐ OPEN ☒ PRIVATE

Provide glob pattern * ?

test*

☒ Allow any member of the project to manually start any job matching this glob pattern

☒ Allow commits and triggers to start any job matching this glob pattern

▲ SCM commits and triggers will automatically run the job

Only authorized users and groups will be able to view the "Job Details" page or edit the job matching this glob pattern.
Any member of the project will be able to manually start the job matching this glob pattern.

Authorized Users/Groups *

Alex Admin [alex.admin] ×

- Select just the **Allow commits and triggers to start this private job** checkbox if you want SCM commits and triggers to be able to automatically run this job.

Job Protection myProtectedJob

CancelSave

☐ OPEN ☒ PRIVATE

☐ Allow any member of the project to manually start this private job

☒ Allow commits and triggers to start this private job

▲ SCM commits and triggers will automatically run the job

Only authorized users and groups will be able to view the "Job Details" page or edit the job.
Only authorized users and groups will be able to manually run the job.
If the job is triggered in a pipeline by an unauthorized user or group, it will not be initiated.

Authorized Users/Groups *

Alex Admin [alex.admin] ×

With just this checkbox selected, periodic triggers will run any job or pipeline, including private jobs set to allow commits and triggers to start the private job. However, if a pipeline includes a private job with this option selected and a non-authorized user attempts to run the pipeline manually, the private job won't run but periodic triggers and SCM commits will.

Leave the checkbox unselected if you don't want the job to be started when it is triggered by an SCM commit or timer.

Note**Best Practice:**

If you use the checkbox to enable the protected build to be triggered by an SCM commit, you need to protect the branch that the build job is tied to. If you don't do this, anyone can trigger the protected build by making a commit to trigger it.



This is what you would see if you selected the **Allow commits and triggers to start any job matching this glob pattern** for the Test Rule.

8. Click **Save**.

The activity stream displays all changes to a job's protection status, like changing the job protection from public to private, or private to public, or changing a private job to allow commits and triggers.

You can see if a job is private from several places in the VB Studio user interface.


- In the jobs list found on the **Job Protection** tab on the **Project Administration** page's **Builds** tile, to the right of each protected job's name.
- In the **Private** column on the **Builds** page's **Jobs** tab.
- In the jobs shown in the **Pipelines** tab on the **Builds** page.

A private job is indicated by a **Lock**  icon. A private job you can run and edit is indicated with an **Unlock**  icon. A private job you can run but not edit is indicated with a **Lock-edit** icon.

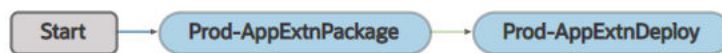
An unauthorized user can't run a private build job manually, through a pipeline, or by using an SCM/periodic trigger.

Create and Configure the Production Pipeline

To ensure the deployment job runs automatically after the packaging job, create a pipeline and set the dependency.

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. Click **+ Create Pipeline**.
4. In the Create Pipeline dialog box, in **Name** and **Description**, enter a unique name and description.
5. Click **Create**.
6. On the Pipeline Configuration page, right-click the **Start** node and select **Add New Start** jobs.
7. Click in the **Select new on success job(s)** field, select the packaging job and click **Save**.
8. Right-click the packaging job and select **Add**, then **Add New On Success Jobs**.
9. Click in the **Select new on success job(s)** field, select the deployment job, and click **Save**.

Here's an example of the final pipeline:



10. Click **Save**.

Run Production Pipelines

Before you run production build jobs or a production pipeline, make sure that all code changes have been pushed to the production branch and there are no open merge requests.

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. In the production pipeline's row, click the **Actions** *** menu and select **Run Pipeline**.

After a successful build, you can see the deployed application on the Manage Extension Lifecycle page. You should also notify the Oracle Cloud Applications administrator that the extension has been pushed to the Oracle Cloud Application's production instance.

Deploy to Other DEV and TEST Instances Through CI/CD Pipelines

If you've followed this chapter to this point, you've configured VB Studio to deploy extensions to your primary Oracle Cloud Applications Development environment. This section explains how to set up other Oracle Cloud Application instances, such as DEV and TEST, where your team members may want to test their extensions before deploying to PROD.

This involves creating an environment for each instance you want to deploy to, as well as configuring jobs and pipelines to package up and deploy extensions to those target instances.

✓ Tip

If you don't need a CI/CD pipeline to deploy your extension, skip this section and use the Manage Extension Lifecycle page to deploy to additional instances. From this single page, you can add new environments for your DEV and TEST instances and deploy extensions to those instances without the need to create build jobs and pipelines.

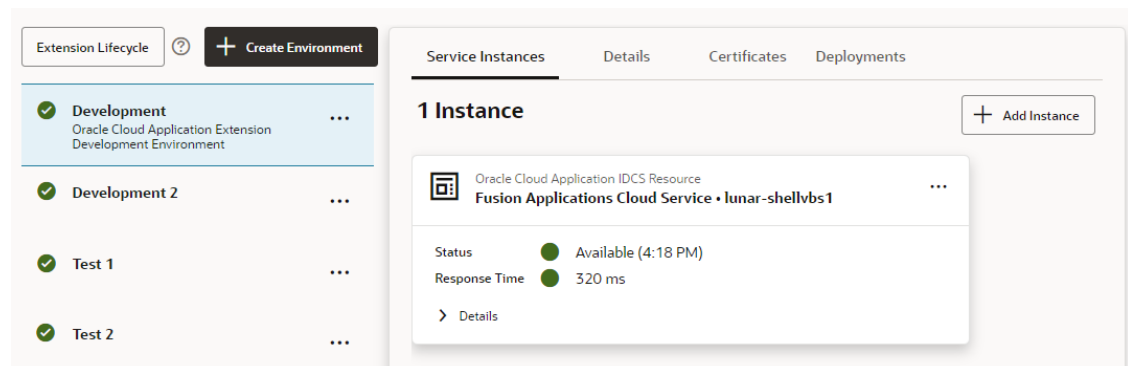
Before you proceed, make sure that the Oracle Cloud Applications instances are up and running.

Add an Oracle Cloud Application Instance to an Environment

To deploy an extension to an Oracle Cloud Applications instance, you must create a VB Studio environment and add the instance to it. You can add only one Oracle Cloud Applications instance to each environment.

Each Oracle Cloud Applications DEV and TEST instance has its own identity domain. This means you'll need the target instance's URL—as well as a user's credentials that can access the instance—to set up the environment. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.

Here's what your Environments page might look like after you've added three new environments, one for another DEV instance and two for some TEST instances:



To create a new VB Studio environment:

1. In the VB Studio navigation menu, click **Environments**.
2. Click **+ Create Environment**. In **Environment Name** and **Description**, enter a unique name and description, then click **Create**.
3. In the Service Instances tab, click **+ Add Instance**.
4. In the Add Service Instance dialog box, select **Oracle Cloud Applications**.
5. Under Add Instance Using, select **Application Credentials**.
6. In **Base URL**, enter the Oracle Cloud Application's base URL.
7. In **Instance Name**, if required, update the instance's display name. The name will be displayed in the Service Instances tab.
8. In **Username** and **Password**, enter the credentials of a user who can access the Oracle Cloud Applications instance.

These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.

9. Under Authorization Type, select **OAuth 2.0** (default) and click **Add**.

Note

OAuth is the recommended authorization type. Use Basic authentication only if you run into issues with setting up an OAuth connection.

10. If you chose OAuth 2.0 as the Authorization Type, you may be prompted for authorization to acquire OAuth tokens for accessing the target instance. Click **Authorize** when prompted, then sign in with credentials to access the Oracle Cloud Applications instance. Note that the username that is shown under the Details section of the connected Oracle Cloud Applications endpoint shown in the **Environments** page's **Instances** tab is the user that is logged in to the Oracle Cloud Applications instance and who creates the Refresh Token that the connection is using. The details don't show the VB Studio username, which isn't relevant for a 3-legged OAuth Environment connection.

OAuth tokens (access and refresh) are cycled during regular use. A refresh token is used to obtain an access token whenever a user accesses the target instance. This refresh token is typically valid for seven days. (The token expiration time is set in the IDCS resource app and may be different based on your security requirements.) If the user authenticates with the target instance within the seven-day period, the active refresh token generates a new access token and a new refresh token. This cycle continues indefinitely as long as the refresh token stays valid. If the refresh token expires during extended periods of inactivity (say, when you're away on vacation), click **Actions ***** and select **Renew OAuth Access**.

Note

Service administrators can control the OAuth access or refresh token's expiration in the IDCS resource app, specifically *Fusion Applications Cloud Service* under Oracle Cloud Services in your identity domain's resource app. See [Edit High-Level Information for Oracle Applications](#).

If the newly added instance stays in the **Unknown** status for some time, it usually indicates that the IDCS Application provisioning may have failed. VB Studio added the Oracle Cloud Applications instance, but can't access it. In this case, click **Actions ***** and select **Remove** to remove the Oracle Cloud Applications instance from the environment, and then click **Add** to add it again.

Once you're done, the Recent Activities feed on the Project Home page lists the newly created environment along with the service instance you added.

Create and Configure Deployment Build Jobs and Pipelines

To deploy extensions to other Oracle Cloud Applications instances, you can use the Manage Extension Lifecycle page. If you'd rather use a CI/CD pipeline for deployment, you'll need to set up or create a packaging job, one or more deploy jobs, and a pipeline. Specifically:

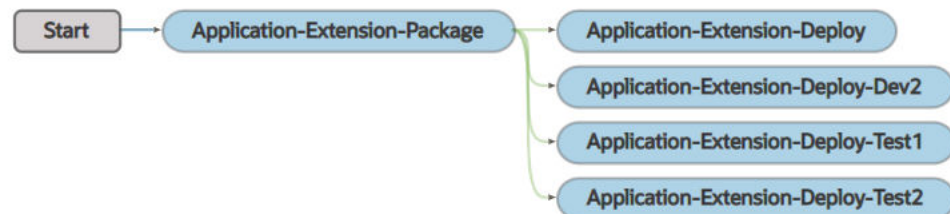
- A packaging job packages the extension and creates a build artifact. Instead of creating new packaging jobs, you can use the `extensionName-Package` job that was created with the project. (If you created this project manually, the package job is called **Application-Extension-Package**.) This job packages the extension and creates a build artifact from

the same branch your team members used to deploy to the primary Oracle Cloud Applications Development instance. Remember, this packaging job is configured to trigger a build on every SCM commit.

- Create one deploy job for each Oracle Cloud Applications instance you plan to deploy to. You'll also need a user's credentials who can deploy to each Oracle Cloud Applications instance. To create the job, see [Create a Deployment Build Job](#).
- After you've created deployment jobs, create pipelines to run the package job and deployment jobs in sequence. You can create either of these pipelines:
 - A pipeline to deploy the extension to the primary and other Oracle Cloud Applications instances.
Instead of creating a new pipeline, configure the existing `extensionName-Package and Deploy` pipeline to deploy to other instances as well. (If you created this project manually, this pipeline's name is **Application-Extension Package and Deploy**.) See [Configure the Default Pipeline to Deploy to Other Oracle Cloud Applications Instances](#). Remember, the default pipeline runs automatically when a code change is pushed to the `main` branch. After you've configured the pipeline, it deploys the extension to all development and test instances on every SCM commit to the `main` branch.

After creating this pipeline, you'll have one pipeline in your project that deploys to all DEV and TEST instances.

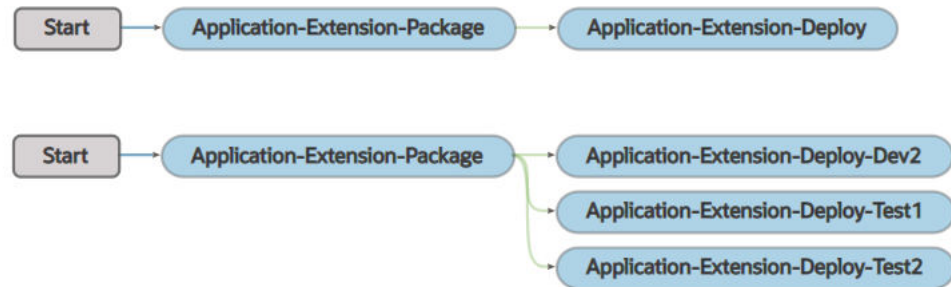
Here's an example of what this pipeline could look like:



- A pipeline to deploy the extension to other DEV and TEST Oracle Cloud Applications instances manually. See [Create and Configure a Pipeline to Deploy to Other Oracle Cloud Applications Instances](#).
Create this pipeline if you don't want to deploy the extension to other instances on every SCM commit, but only after your team members have validated the extension on the primary development instance.


After creating this pipeline, you'll have two pipelines in the project: 1) Your primary pipeline, which deploys to the primary development instance, and 2) this pipeline, which deploys to other DEV and TEST instances.

Here's an example of what these pipelines could look like:



Create a Deployment Build Job

The deployment job deploys the extension's artifact that was generated in the default packaging job to the Oracle Cloud Application's instance. Before you create the job, make sure you have credentials that VB Studio can use to access the instance.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. Select the **Copy From Existing** check box.
6. From the **Copy From** drop-down list, select the deployment job that you'd like to copy.
7. In **Template**, make sure that the **System Default OL7 for Visual Builder** template is selected.
8. Click **Create**.
9. From the Job Configuration page, click the **Steps** tab.
The Application Extension Deployment section is displayed.
10. In **Target Instance**, select the environment with the target Oracle Cloud Applications instance.
11. In the Authorization section, specify the authorization type to run this build step. With **Use OAuth** selected by default, you'll see the *Authorization is required* message, indicating that this build step needs a one-time authorization to handle OAuth requests to your environment's Oracle Cloud Applications instance. Click **Authorize** and enter credentials to access your Oracle Cloud Applications instance; you can also run the job manually and enter the credentials when prompted.

Either way, it is recommended that you authorize your OAuth connection during initial configuration. If you skip this step, you won't be able to publish your changes from the Designer and will need to complete the required authorization before attempting to deploy changes.

Once authorized, the *Authorization has been provided* message shows.

Note


OAuth is the recommended authorization type. Use Basic authentication only if you run into issues with setting up an OAuth connection. To use Basic authentication, select **Use Basic**, then enter the credentials of a user who can access the Oracle Cloud Applications instance in **Username** and **Password**. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.

OAuth tokens (access and refresh) are cycled during regular use. A refresh token is used to obtain an access token whenever a user accesses the target instance. This refresh token is typically valid for seven days. (The token expiration time is set in the IDCS resource app and may be different based on your security requirements.) If the user authenticates with the target instance within the seven-day period, the active refresh token generates a new access token and a new refresh token. This cycle continues indefinitely as long as the refresh token stays valid. If the refresh token expires during extended periods of inactivity (say, when you're away on vacation), click **Renew Authorization** (or run the job manually, so you're prompted to authorize any expired OAuth tokens).

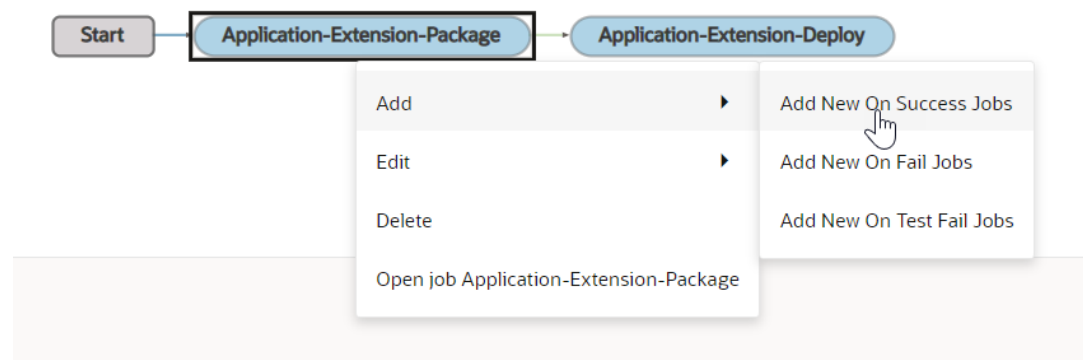
12. The **Build Artifact** field should show the same artifact name that was used in the packaging build step. Confirm this value, especially if the packaging job used an artifact name other than the default, `extension.vv`.
13. Click **Save**.

Configure the Default Pipeline to Deploy to Other Oracle Cloud Applications Instances

If you want to deploy to other DEV and TEST instances along with your primary Oracle Cloud Applications instance, you can configure the default pipeline to deploy to other instances as well.

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. In the `extensionName--` **Package and Deploy** pipeline row, click the **Actions** menu and select **Configure Pipeline**. (If you created the project manually, the pipeline name is **Application Extension - Package and Deploy**.)
4. Right-click the `extensionName--Package` job and select **Add**, then **Add New On Success Jobs**.

Here's an example:



- Click in the **Select new on success job(s)** field, select the jobs that you want to add from the list, and click **Save**.

Add New On Success Job(s)

×

Select new on success job(s)

Application-Extension-Deploy-Dev2 ×

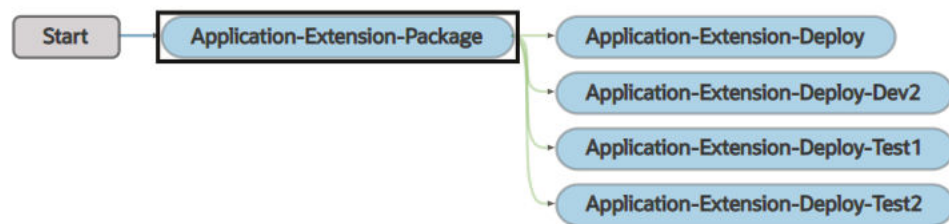
Application-Extension-Deploy-Test1 ×

Application-Extension-Deploy-Test2 ×

Cancel


Save

Here's an example of the final diagram:



- Click **Save**.

Create and Configure a Pipeline to Deploy to Other Oracle Cloud Applications Instances

- In the left navigator, click **Builds** .
- Click the **Pipelines** tab.
- Click **+ Create Pipeline**.
- In the Create Pipeline dialog box, in **Name** and **Description**, enter a unique name and description.
- Deselect the **Auto start when pipeline jobs are built externally** check box.
- Click **Create**.
- On the Pipeline Configuration page, right-click the **Start** node and select **Add New Start** jobs.
- Click in the **Select new on success job(s)** field, select `extensionName-Package`, and click **Save**.

Add New Job(s) to Start
×

Select new job(s)

Application-Extension-Package ×

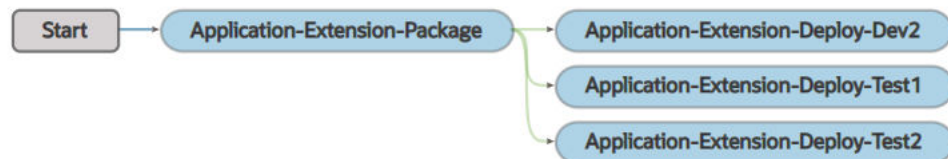
Choose new job(s) that will start when the pipeline is run.

Cancel

Save

9. Right-click the `extensionName-Package` job and select **Add**, then **Add New On Success Jobs**.
10. Click in the **Select new on success job(s)** field, select the jobs that you want to add from the list, and click **Save**.

Here's an example of a finished pipeline:



11. Click **Save**.

Deploy Changes from a Different Branch

At some point, you'll likely want to create a separate branch to host your new feature development work. You can package those changes from your new branch and deploy them to an Oracle Cloud Applications instance for testing.

To deploy to the Oracle Cloud Applications instance in the environment associated with your workspace, you can select the new branch when you publish your changes. Or, if your changes already exist in the new branch, you can use the **Deploy** option in the Designer. See *Deploy Changes From the Remote Branch* in *Extending Oracle Cloud Applications with Visual Builder Studio*.



To deploy to a different Oracle Cloud Applications instance, such as TEST or PROD, you'll need to set up some build jobs and a CI/CD pipeline.

 **Tip**

VB Studio can create the build jobs and pipeline for you, if you prefer. In the extension's Settings editor, under Building and Publishing, select the new branch, and then create and enable a CI/CD pipeline for the branch. (If you're deploying to a different instance that's **not** the one associated with your workspace, you'll need to change the target environment in the deployment job, after it's created.) See Establish Extension-Level Settings in *Extending Oracle Cloud Applications with Visual Builder Studio*.

Create a Packaging Job for the New Branch

Create a packaging job following the same steps to create a [production packaging build job](#), but when configuring the job, select the new branch instead of the default `main` branch:

1. In the left navigator, click **Builds** .
2. Create a job using the **System Default OL7 for Visual Builder** template.
3. Click **Configure** .
4. Click the **Git** tab.
5. From the **Add Git** list, select **Git**.
6. In **Repository**, select the Git repository. In **Branch or Tag**, select the new branch.

Finish configuring the packaging job as usual. See [Create the Production Packaging Build Job](#).

Create a Deployment Job

Create a deployment job following the same steps to create a [production deployment build job](#), but when configuring the job, select the packaging job that you just created.



Create and Configure a Pipeline for the New Branch

After you've created the packaging and deployment jobs for the new branch, create a pipeline to run the packaging job and deployment jobs in sequence. See [Create and Configure a Pipeline to Deploy to Other Oracle Cloud Applications Instances](#).


Run the Pipeline

The test build pipeline runs automatically when a commit is pushed to the Git repository's test branch specified in the packaging job.

If you want to run the pipeline manually:

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. For the pipeline you want to run, click **Build** .

To monitor the pipeline, click the pipeline's name on the **Builds** page. After a successful build, you can see the deployed application on the Manage Extension Lifecycle page.

If you want to run a job's build manually, open the job's details page and click **Build Now**. You can monitor its build on the job's details page. To view the latest build log of a job, open the **Builds** page, click the job's name, and then click **Build Log** .

Manually Create a Project for Extensions

The easiest way to create a project for extensions is through the **Edit Page in Visual Builder Studio** link in an Oracle Cloud Application, because that workflow automates some provisioning steps. However, if you don't want to configure an existing Oracle Cloud Application page, but instead would like to create your own application that lives as a peer alongside Oracle applications, you can create a project manually.

To create an extension project manually:

1. On the Organization page, click **+ Create**.
2. On the New Project wizard's **Project Details** page, enter a name and optional description.
3. In **Security**, select the project's privacy setting:

- a. Select **Private** to restrict access to project members only.

The **Discoverable** checkbox, selected by default for private projects, allows organization members who aren't org admins or project members to see basic information, such as name and owner contact information, about your private project. Private projects that aren't discoverable won't be exposed to non-members.

Note

If you want this project to be visible to users when they use the **Edit Page in Visual Builder Studio** option to make changes to their Oracle Cloud Applications page in VB Studio, mark it as Discoverable. This gives users an opportunity to select a project that's already associated with the page they want to modify (or they can create a new project, if they prefer).

- b. Select **Shared** to make the project code, wiki docs, tasks, and builds available to anyone inside your organization.
4. In **Preferred Language**, specify the language for the email notifications your project users will receive.

(You can change the language of the *user interface* in your user preferences.)

5. Click **Next**.
6. On the **Project Template** page, select **Empty Project**, then click **Next**.
7. On the **Project Properties** page, in **Wiki Markup**, the organization's wiki markup language is displayed by default. You can change it for this project, if needed.

Project users use markup to format content in wiki pages, as well as in issue and merge request comments. See [Working with Markup Languages in VB Studio](#).

8. Click **Next**.
9. On the **Project Team members** page:
 - a. Click **Add Members** and select Oracle Cloud Applications users or groups to add to the project, from the list displayed, if you know they might work in this project.
 - b. Select the membership (Project Owner, Developer Full Access, Developer Limited Access, or Contributor) that the members you're adding will have in the project. See [What Are Project Memberships?](#) for more information about each membership.
 - c. Click **Add**.

- d. Repeat substeps a, b, and c for different users and groups with various membership types, if needed.

10. Click **Finish**.


After the project is provisioned, the **Project Home** page opens.

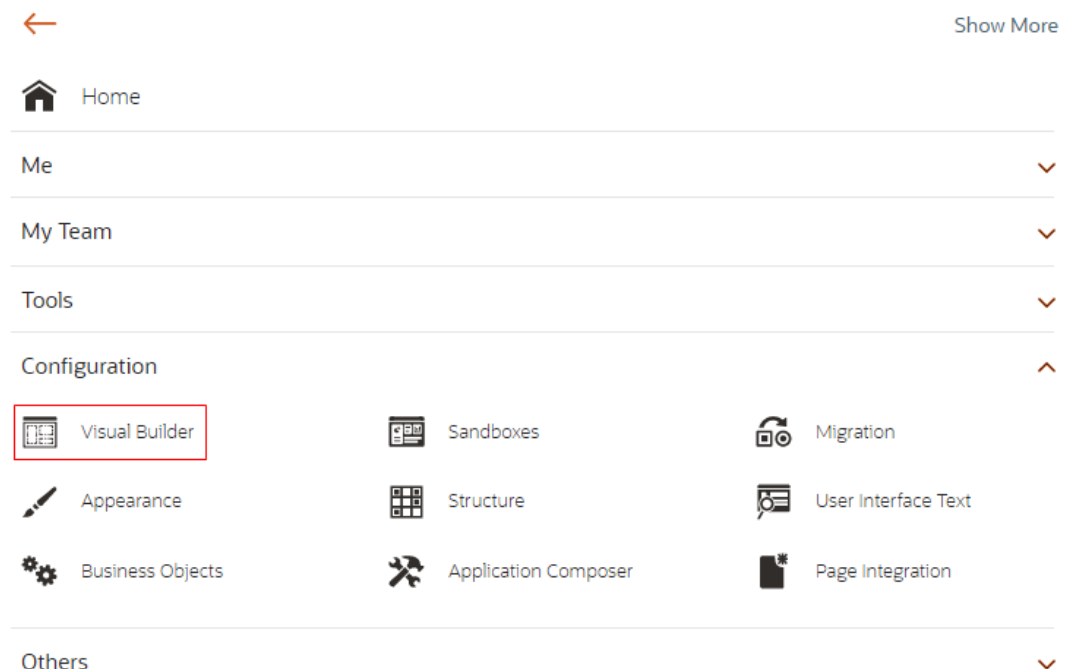
When you create a project through the **Edit Page in Visual Builder Studio** link in an Oracle Cloud Application, several other VB Studio artifacts are also created for you. If you manually create a project, however, you'll need to do these steps yourself:

1. Define a new environment.
 - To add an Oracle Cloud Applications instance in the same identity domain to an environment, select **Oracle Cloud Applications** under Instance Type and **Identity Domain** under Add Instance Using, then select an instance.
 - To add an Oracle Cloud Application from a different identity domain, select **Oracle Cloud Applications** under Instance Type and **Application Credentials** under Add Instance Using. You'll need the base Oracle Cloud Application's URL and the credentials of a user who can access the instance. See [Add the Oracle Cloud Application's Production Instance to an Environment](#).
2. [Create a workspace for your extension](#). On the Workspaces tab, select **New**, then **New Application Extension**.

Access VB Studio from Oracle Cloud Applications

While the **Edit Page in Visual Builder Studio** link in an Oracle Cloud Application is the easiest way to access VB Studio, there is an alternative approach:

1. In your Oracle Cloud Applications instance, click **Navigator** .
2. In the navigation menu, under **Configuration**, select **Visual Builder**.



The VB Studio **Organization** page opens, which displays all the projects you're a member of, as well as your favorite projects, the projects you own, and all the shared projects in your organization.

Delete an Extension

Delete unnecessary extensions to restore your application to its original state and clean up your environment.

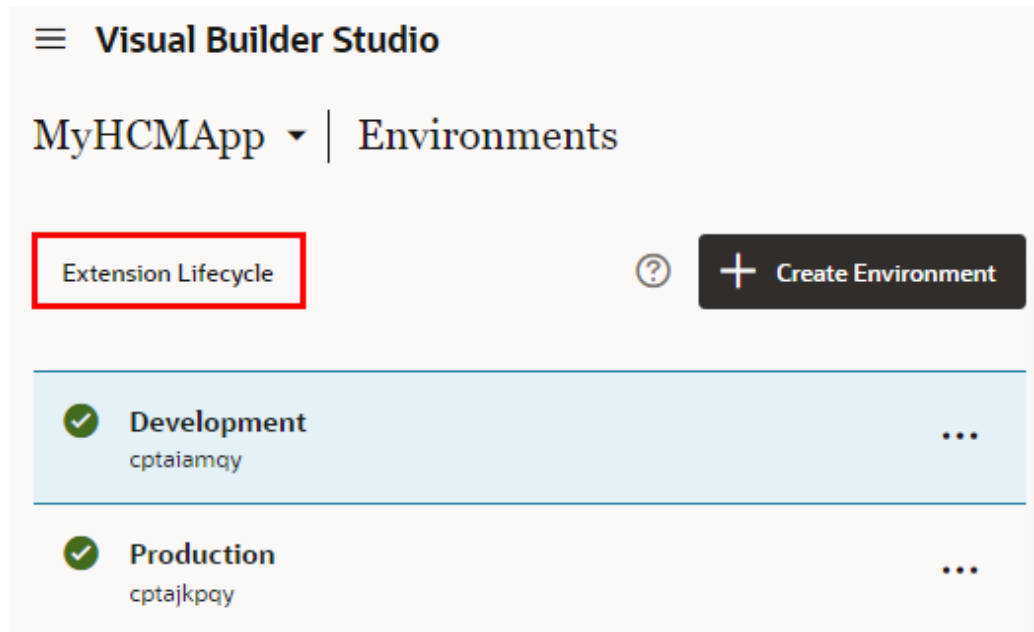
You have two options to delete an extension:

- The best practice is to use the Manage Lifecycle Extension page to delete an extension. When you delete an extension from this page, all versions of that extension are deleted.
- If you need to delete a specific extension *version*, you must [configure a build job](#) to delete it.

Delete an Extension Manually

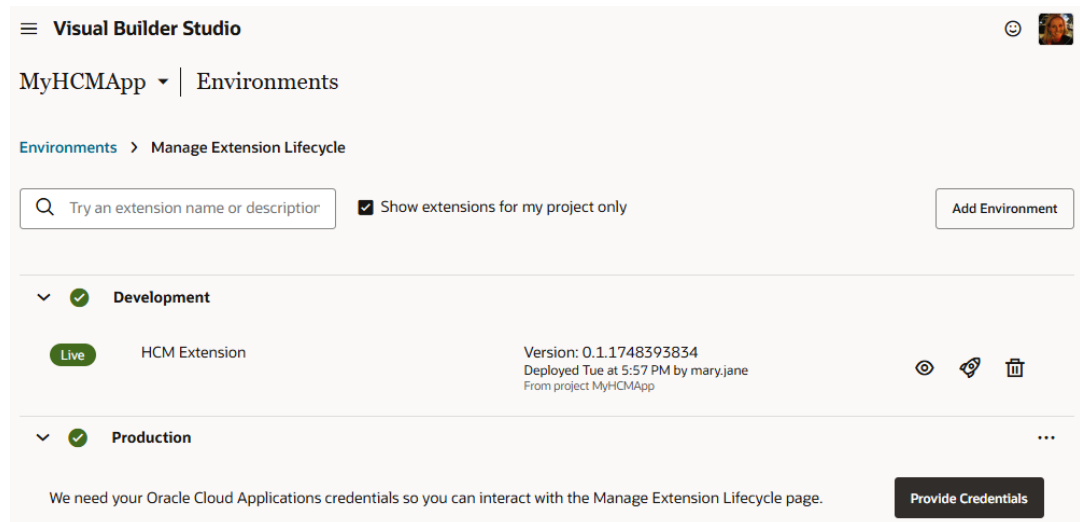
Use the Manage Extension Lifecycle page to delete extensions deployed from any project and to any Oracle Cloud Applications instance.

1. Access the Manage Extension Lifecycle page:
 - From the VB Studio left navigator, click **Environments**, then click **Extension Lifecycle**. You won't see this option if you haven't yet added an Oracle Cloud Applications instance to your environment.



- Or, from the menu in the Designer's header, click **Extension Lifecycle**.

You should now see the Manage Extension Lifecycle page, which lists all the environments that are available.



You can work with an environment on this page as long as the authentication method for its Oracle Cloud Applications instance is OAuth. If there's an environment listed whose authentication method isn't OAuth, then you'll first need to add a new environment with the same base URL, but using OAuth as the authentication method, before you can work with it.

Note

If you see a **Provide Credentials** button next to an environment, you must first sign in before you can delete any extensions deployed to that instance.

Expand each environment to view a list of the extensions already deployed. Use the **Show extensions for my project only** checkbox to toggle between viewing extensions deployed from the current project only, or from all projects.

Each entry includes the extension name plus additional details, such as extension version, deployment date, who deployed the extension, and project name.

2. To delete an extension from an Oracle Cloud Applications instance, find the extension that



you want to delete and click

All extension versions are deleted. If the extension was originally deployed via a CI/CD pipeline, the extension's build jobs and their artifacts remain; this is helpful in case you ever need to redeploy an earlier extension version.

Tip

Before deleting an extension from your PROD instance, delete the extension from the DEV instance (or TEST instance, if applicable) and make sure there aren't any adverse effects.

Configure a Job to Delete an Extension


If you want to delete a specific extension *version* that's deployed to an Oracle Cloud Applications instance, you must configure a build job and run it.

You can also use the Manage Extension Lifecycle page to delete extensions, but keep in mind that deleting an extension from that page deletes the *entire* extension.

Note that there can be only one *active* extension version at a time. The recommended best practice is to avoid deleting the active extension version because, if you delete it, the extension will become inactive as if it wasn't installed at all. You can, however, delete inactive versions without impacting the currently active deployed extension.

Before you configure and run the job, delete the extension from the DEV instance (or TEST instance, if applicable) and make sure there aren't any adverse effects. For example, let's assume you have an attribute that's hidden in both the extension's business object and the user interface. After you delete the extension, the user interface shows the attribute that's still hidden in the business object. This may cause an error.

To configure the job, make sure you have valid administrator credentials for the Oracle Cloud Applications instance where the extension is deployed.



1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select **System Default OL7 for Visual Builder**.
6. Click **Create**.

The Job Configuration page opens.

7. Click the **Steps** tab.
8. From **Add Step**, select **Application Extension**, and then select **Delete**.

This image shows the Application Extension Delete build job page that's partially filled in.



Application Extension Delete ?  

Instance *
Fusion Applications Cloud Service (Env-IDCS) ▼

Authorization

☒ Use OAuth ☐ Use Basic

 Authorization is required. 

Extension ID *
site_AdminAppExt240401

Version *
1.0.0

9. In **Instance**, select the Oracle Cloud Applications instance where the application is deployed.
10. In the Authorization section, specify the authorization type to run this build step. With **Use OAuth** selected by default, you'll see the `Authorization is required` message, indicating that this build step needs a one-time authorization to handle OAuth requests to your environment's Oracle Cloud Applications instance. Click **Authorize** and enter credentials to access your Oracle Cloud Applications instance; you can also run the job manually and enter the credentials when prompted.

Once authorized, the `Authorization has been provided` message shows.

Note

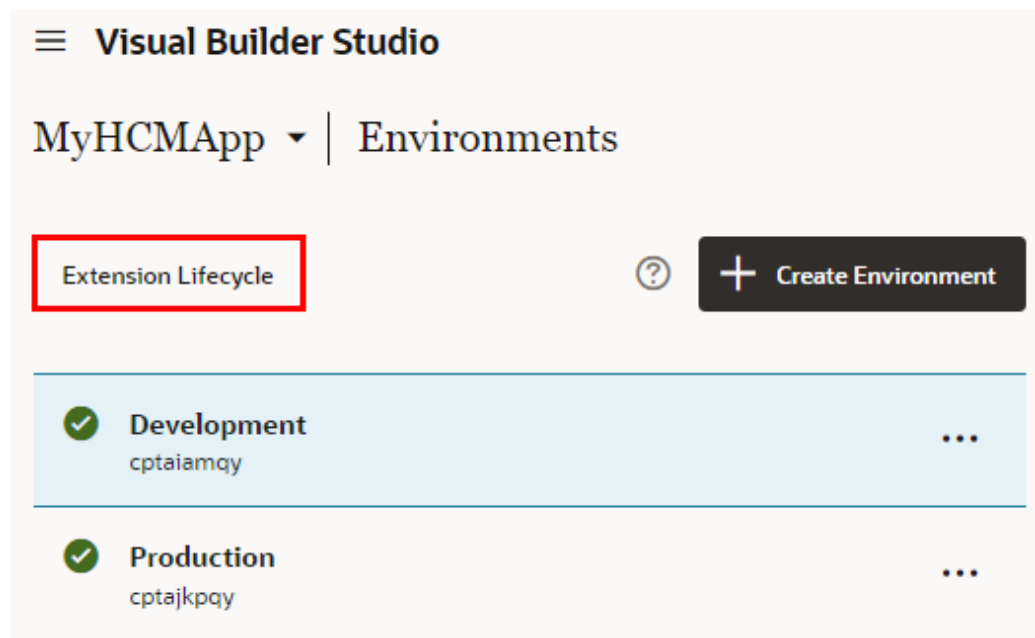
OAuth is the recommended authorization type. Use Basic authentication only if you run into issues with setting up an OAuth connection. To use Basic authentication, select **Use Basic**, then in **Username** and **Password**, enter the credentials of a user who is not only an Oracle Cloud Applications user, but one who can access the Oracle Cloud Application's instance and undeploy from it. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.


OAuth tokens (access and refresh) are cycled during regular use. A refresh token is used to obtain an access token whenever a user accesses the target instance. This refresh token is typically valid for seven days. (The token expiration time is set in the IDCS resource app and may be different based on your security requirements.) If the user authenticates with the target instance within the seven-day period, the active refresh token generates a new access token and a new refresh token. This cycle continues indefinitely as long as the refresh token stays valid. If the refresh token expires during extended periods of inactivity (say, when you're away on vacation), click **Renew Authorization** (or run the job manually, so you're prompted to authorize any expired OAuth tokens).

11. In **Extension ID** and **Version**, enter the extension ID and version of the extension you want to delete.

You can find these items in the extension's details on the Manage Extension Lifecycle page.

- a. In the left navigator, click **Environments**, then click **Extension Lifecycle**.



- b. Find the extension you want to delete, and select  to obtain the extension ID and version.
12. Click **Save**.
13. To run a build, click **Build Now**.

4

Set Up VB Studio for Developing Visual Applications

This chapter tells you how to set up Oracle Visual Builder Studio (VB Studio) so that your users can create *visual applications*.

A visual application is a responsive application developed using VB Studio's browser-based development environment. You deploy a visual application to a separate Visual Builder instance. The Visual Builder instance must be version 19.4.3.1, or later.

Assuming you have already created a VB Studio instance, the setup process for visual applications involves three phases:

- [Set Up VB Studio for Visual Applications](#)
- [Set Up the VB Studio Project for Development](#)
- [Set Up the Project to Deploy for Production](#)

Set Up VB Studio for Visual Applications

Here's a summary of how to set up VB Studio for developing visual applications:

To perform this action:	See this:	Why do I need to perform this action?
1. Get access to Visual Builder instances, and make sure VB Studio is authorized to access those instances.	Get Access to Visual Builder Instances Authorize VB Studio to Access Visual Builder and Integration Instances	To deploy an app, your users need credentials that allow them to access the Visual Builder instance (deployment target). If an instance isn't available for them, you must create it. Further, to make the Visual Builder instance available for selection as your environment's deployment target, VB Studio must be authorized to access that instance.
2. Get the required IDCS roles assigned to you.	Get the Required IDCS Roles	To create and set up the VB Studio instance, you must be assigned some specific IDCS roles.
3. (Optional) Configure VB Studio to run build jobs and pipelines.	Configure VB Studio to Use the Free VM Executor for Build Jobs and Pipelines	To run build jobs and pipelines, you must configure VB Studio to connect to an OCI paid account or the built-in free account.

Note

If you're developing a visual application using an instance of VB Studio that was provisioned in your Oracle Cloud Applications account, follow the procedures in [Create and Set up a Project for Development](#).

Get Access to Visual Builder Instances

You can deploy a visual application to a separate Visual Builder instance—you cannot deploy to Visual Builder Studio itself.

If you don't already have an instance to deploy to:

1. [Create a Visual Builder Service Instance](#).
To create an instance, you'll need the IDCS VISUALBUILDER_ENTITLEMENT_ADMINISTRATOR service role.
2. [Add an IDCS User to Your Cloud Account](#) and assign the user a Visual Builder role to connect and deploy applications.

You'll also need to configure the instance once it's created. If you're a Visual Builder runtime administrator, sign in to the Visual Builder runtime to complete these tasks, as described in *Administering Oracle Visual Builder, Gen 2*:

- [Access Instance Settings](#)
- [Configure Security Options for Applications](#)
- [Set Page Messages for Access Denied Errors](#)
- [Allow Other Domains Access to Services](#)
- [Switch to Your Own Oracle DB Instance](#)
- [Reset an Expired Password or ATP Wallet for Your Oracle DB Instance](#)
- [Manage Self-signed Certificates](#)
- Configure Support for a Custom Domain

It's important to keep these things in mind before you deploy a visual application to a Visual Builder instance:

- The Visual Builder instance must be version 19.4.3.1 or later.
- To ensure that business objects work properly, your Visual Builder administrator must manually add the VB Studio hostname to the allow list for each Visual Builder instance. For more information, see [Add an Allowed Origins Domain](#).
- If you need to create a Visual Builder instance to deploy to, create it in the same Oracle Cloud account and identity domain as the VB Studio instance you are developing the visual application with. When you need to deploy your app for testing or to go live, you'll need to create instances for those environments using working credentials in different identity domains. See [Create a VB Studio Instance in Your Oracle Cloud Account](#).
- Follow your organization's guidelines to create and set up Visual Builder instances for deployment targets. Your guidelines may suggest that you create instances for different deployment environments, such as development, integration, test, pre-stage, stage, pre-production, and production. You can create these instances in different identity domains or in a common identity domain.

This documentation assumes that you have created and set up Visual Builder instances you can deploy to in your development and production environments, where each instance is in a different identity domain.

To deploy your visual application to Visual Builder development and production instances, you'll need credentials to access them. You'll be prompted for these credentials when you set up the instances or when you perform a manual deployment to those instances.

Note

Make sure your VB Studio instance is authorized to access Visual Builder instances connected to your OCI account. Without this authorization, your users won't be able to create environments with a Visual Builder instance to deploy their visual applications. See [Authorize VB Studio to Access Visual Builder and Integration Instances](#) on the steps to complete this authorization.

Get the Required IDCS Roles

Make sure that you have the necessary identity domain roles you'll need to set up VB Studio on an OCI tenancy:

You must be assigned this role:	To ...
Identity Domain Administrator or User Administrator	Add users and assign IDCS roles.
DCS_INSTANCE_ENTITLEMENT_ADMINISTRATOR OR DEVELOPER_ADMINISTRATOR	Create the VB Studio instance. Set up VB Studio. After you're assigned the role, you're considered VB Studio's <i>Organization Administrator</i> .
OCI_Administrator (OCI Administrator)	Set up OCI compartments and buckets, which are required to set up the VB Studio build system.

Set Up the VB Studio Project for Development

After setting up VB Studio, follow these steps to create and set up a project for developing your visual applications.

To perform this action:	See this:	Why do I need to perform this action?
1. Create a project for visual applications and set up the project for development.	Create and Set up a Project for Development	To develop a visual application, you must create a VB Studio project and set up the project for your developers.
2. Add users to the project.	Add Users to the Project	To allow your team members to access the visual application project, invite them to join the project.
3. (Optional) Configure advanced settings.	Optional Configuration	These settings are optional, depending on your configuration.
4. Developers can now create visual applications.	Develop Your Application	This is the main development stage for your visual app.

Create and Set up a Project for Development

A project gathers all the resources you need for developing software—in this case, a project for visual applications.

After you've [set up VB Studio to develop visual applications](#), follow the steps in this table to create an empty project, add an environment with a Visual Builder instance, configure the deployment job, and add users to the project. When your VB Studio instance is in a different identity domain than your Visual Builder instance, you'll additionally need to authorize the connection to your Visual Builder instance.

✓ Tip

Instead of creating an empty project and setting it up manually (as described here), you can use the Visual Application template to quickly create a project with everything you need to develop visual applications. Note though that this option is available only when your VB Studio instance is in the same identity domain as your Visual Builder instance. See [Create and Set Up a Project Using the Visual Application Template](#).



Action	How to
1. Create an empty VB Studio project.	<ol style="list-style-type: none"> 1. Sign in to VB Studio. See Access VB Studio from the Oracle Cloud Home Page. 2. On the Organization page, click + Create. 3. On the Project Details page, enter a unique name and description for the project. 4. In Security, select the project's privacy setting. A private project is accessible to invited users only. Users who aren't invited can't access it or make changes to it. You can invite users after creating the project. A shared project is accessible to all users of the organization. Any user can view the source code, create or update issues, edit wiki pages, and interact with project builds. However, only invited users can make updates to the source code in Git repositories, create and run build jobs, and perform deployment operations. 5. In Preferred Language, specify the language for the email notifications your project users will receive. You can change the language in which the user interface appears in your user preferences. 6. Click Next. 7. On the Template page, select the Empty Project template, then click Next. 8. On the Properties page, select the wiki markup type, then click Finish. <p>After the project is provisioned, the Project Home page opens where you can see a summary of the project's provisioning activities. Review the activities feed for any errors.</p>

Action	How to
--------	--------

2. In the project, create an environment with your Visual Builder instance.	
-----------------------------------------------------------------------------	--

Note


Before you try to create an environment, make sure your VB Studio instance is authorized to access the Visual Builder instances that are connected to your OCI account. Without this authorization, those instances won't be available for selection in your environment as a deployment target. See [Authorize VB Studio to Access Visual Builder and Integration Instances](#).


1. In the navigation menu, click **Environments** , then click **+ Create Environment**.
2. In **Environment Name** and **Description**, enter a unique name and description, and click **Create**.
3. In the environment's **Service Instances** tab, click **Add Instance**.
4. Locate and add your Visual Builder instance:
 - a. Make sure Instance Type is set to **Visual Builder** and Add Instance Using is set to **Instance List**. This is the preferred approach.
 - b. Click  next to **OCI Details**, select the region and compartment where your Visual Builder instance is, and click **Update**.
 - c. Select your Visual Builder instance from the selected compartment's instance list and click **Add**.

If you don't see the instance you want to add (because it's connected to a different OCI account), select **Visual Builder Credentials** under Add Instance Using. In **Base URL**, enter the Visual Builder instance's URL; update the **Instance Name** if needed; enter the credentials of a user who can access the Visual Builder instance in **Username** and **Password**; and click **Add**. Note though that using this approach won't allow you to set up OAuth for this environment in the deployment build job, used by the Publish action in the Designer (depending on [setup](#)).

5. If your target instance uses a different identity domain than your VB Studio instance, OAuth tokens (via three-legged OAuth flows) are used to secure programmatic access to the instance. Before any OAuth tokens can be created, you must provide authorization to acquire OAuth tokens. Click **Authorize** when prompted, then sign in with credentials to connect and deploy to the target instance.

Once you're done, the Recent Activities feed on the Project Home page lists the newly created environment along with the Visual Builder instance you added.

Action	How to
3. When users are ready, they can create a workspace in the project.	<ol style="list-style-type: none">1. In the navigation menu, click Workspaces .2. Click New and select New Visual Application.3. In Workspace Name, add a name for your workspace.4. In Development Environment, select the environment you created in the previous step.5. In Visual Application Template, don't change the default (Default VB Application).6. In Git Repository, select Create new repository, and enter a repository name and working branch name. <p>When a visual application is created via a new workspace, it's configured to deploy <i>directly</i> to the development Visual Builder instance, without the use of a CI/CD pipeline. If needed, you can instead enable the CI/CD pipeline for the target branch you're merging to.</p>

Action	How to
4. (Optional) The user that created the visual application will need to configure their deployment job.	<p>If you're not using CI/CD pipelines for deployment, skip this step.</p> <ol style="list-style-type: none"> 1. In your Visual Builder Studio navigation menu, click Builds . 2. In the Jobs tab, click the deployment job. When you create a workspace, the packaging and deploy jobs are automatically created and take the name of your repo. (They will be disabled, however, until you enable them.) 3. Click Configure, then click the Steps tab. 4. In Target Instance, make sure that the environment you created in Step 2 is selected. 5. In the Authorization section, specify the authorization type to run this build step. With Use OAuth selected by default, you'll see the <code>Authorization is required</code> message, indicating that this build step needs a one-time authorization to handle OAuth requests to your environment's Visual Builder instance. Click Authorize and enter credentials to access your Visual Builder instance; you can also run the job manually and enter the credentials when prompted. Either way, it is recommended that you authorize your OAuth connection during initial configuration. If you skip this step, you won't be able to publish your changes from the Designer and will need to complete the required authorization before attempting to deploy changes. Once authorized, the <code>Authorization has been provided</code> message shows. <div data-bbox="727 1054 841 1087" data-label="Section-Header"> <p>Note</p> </div> <div data-bbox="766 1106 1448 1251" data-label="Text"> <p>OAuth is the recommended authorization type. Use Basic authentication only if you run into issues with setting up an OAuth connection. To use Basic authentication, select Use Basic, then enter the credentials of a user who can access the Visual Builder instance in Username and Password.</p> </div> <p>OAuth tokens (access and refresh) are cycled during regular use. A refresh token is used to obtain an access token whenever a user accesses the target instance. This refresh token is typically valid for seven days. (The token expiration time is set in the IDCS resource app and may be different based on your security requirements.) If the user authenticates with the target instance within the seven-day period, the active refresh token generates a new access token and a new refresh token. This cycle continues indefinitely as long as the refresh token stays valid. If the refresh token expires during extended periods of inactivity (say, when you're away on vacation), click Renew Authorization (or run the job manually, so you're prompted to authorize any expired OAuth tokens).</p> <ol style="list-style-type: none"> 6. (Optional) Choose Use custom file names and add custom file name details. 7. (Optional) To overwrite the application's default version, specify the new version in Application Version. Leave it empty to use the version defined in the application's <code>visual-application.json</code> file. Don't deselect the Include the application version in the URL checkbox.

Action	How to
	<ol style="list-style-type: none"> 8. (Optional) In Application Profile, specify the development application profile. Leave it empty to use the application's default profile. Using application profiles, you can define different combinations of servers and security settings for each of your environments, and use them when deploying the application to an environment. To learn more, see What Are Application Profiles? 9. (Optional) To use the existing application's database, in Data Management, select Keep existing environment data. To use a clean database for the application, in Data Management, select Use clean database. 10. Click Save.
5. Add users to the project.	To allow your team members to access the visual application project, invite them to join the project. See Add Users to the Project .

Create and Set Up a Project Using the Visual Application Template

When your VB Studio instance is in the same identity domain as your Visual Builder instance, you can create a project based on the Visual Application template, then make a few configuration settings in the project so that it's ready for developing visual applications.

Here's a summary of how to create and set up a VB Studio project for development starting with the Visual Application template:

To perform this action:

1. Create a project.

See this:

[Create a Project Using the Visual Application Template](#)

Note

Before you try to create a project, make sure your VB Studio instance is authorized to access Visual Builder instances that are connected to your OCI account. Without this authorization, these instances won't be available for selection as a deployment target in your project. See [Authorize VB Studio to Access Visual Builder and Integration Instances](#).

To perform this action:	See this:
2. (Optional) Configure the deployment job to add credentials for connecting to the target development instance.	Configure the Deployment Job When you create a project using the Visual Application template, a workspace is created that's configured to deploy your visual application to the development Visual Builder instance via a <i>CI/CD pipeline</i> . If you prefer, you can disable the CI/CD pipeline and instead deploy <i>directly</i> to the development Visual Builder instance. If you disable the CI/CD pipeline for deployment, skip this step.
3. (Optional) Verify your credentials by running the development pipeline.	Run the Pipeline Manually If you disable the CI/CD pipeline for deployment, skip this step.
4. View the deployed visual applications.	View the Deployed Visual Application
5. Add team members to the project.	Add Users to the Project

Create a Project Using the Visual Application Template

If you're developing a visual application using a VB Studio instance that's in the same identity domain as your Visual Builder instance, you can quickly get started by basing your project on the Visual Application project template.

1. Sign in to VB Studio. See [Access VB Studio from the Oracle Cloud Home Page](#).
2. On the Organization page, click **+ Create**.
3. On the **Project Details** page of the New Project wizard, enter a unique name and description for the project.
4. In **Security**, select the project's privacy:
 - a. Select **Private** to restrict access to project members only.
 Select the **Discoverable** checkbox to allow organization members that aren't org admins or project members to see basic information, such as name and owner contact information, about your private project. Private projects that aren't discoverable won't be exposed to non-members.
 - b. Select **Shared** to make the project code, wiki docs, tasks, and builds available to anyone inside your organization.
5. In **Preferred Language**, specify the language for the email notifications your project users will receive.
 You can change the language in which the user interface appears in your user preferences.
6. Click **Next**.
7. On the **Project Template** page, select the **Visual Application** project template, and then click **Next**.
8. On the **Project Properties** page:
 - a. In **Git Repository Name**, change the Git repository's default name, if required.
 - b. In **Region** and **Compartment**, select the OCI region and compartment where your Visual Builder instance is.
 - c. In **Development VB Instance**, if not already selected, select the Visual Builder development instance.

If you don't see the Visual Builder instance you need, go back and revise your selections for the OCI region and compartment.

- d. In **Visual Application Template**, select a visual application template available on the selected development instance.

By default, VB Studio uses the **Default VB Application** template. To select another template, click **Change Template**, select the template and click **Use Selected**.

- e. (Optional) In **Workspace Name**, if required, change your private workspace's name. By default, it is `Workspace1`.

A workspace contains all the artifacts that you need to develop visual applications, including a clone of this project's Git repository—and the branch—containing the source files. To learn more about workspaces, see *Create a Workspace in Building Responsive Applications with Visual Builder Studio*.

- f. (Optional) In **Working Branch Name**, if required, change the workspace's working branch name. By default, it is `branch1`.

When the project is provisioned, the Git repository's `main` branch contains your application's files. While creating the workspace, VB Studio creates a copy of the `main` branch, renames it with your specified name and uses it as the workspace's working branch.

- g. Click **Next**.

9. On the **Team** page:

- a. Click **Add Members** and select users or groups to add to the project, from the list displayed.

- b. Select the membership (Project Owner, Developer Full Access, Developer Limited Access, or Contributor) that the members you're adding will have in the project.

See *What Are Project Memberships?* for more information about each membership.

- c. Click **Add**.

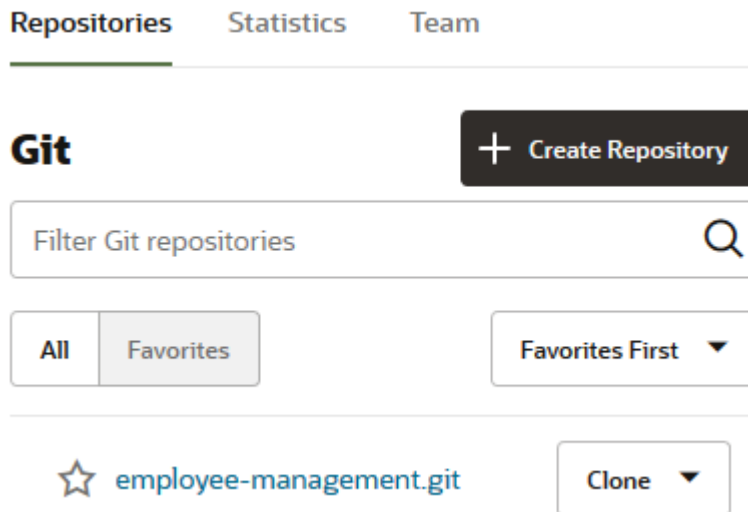
- d. Repeat substeps a, b, and c for different users and groups with various membership types, if needed.

10. Click **Finish**.

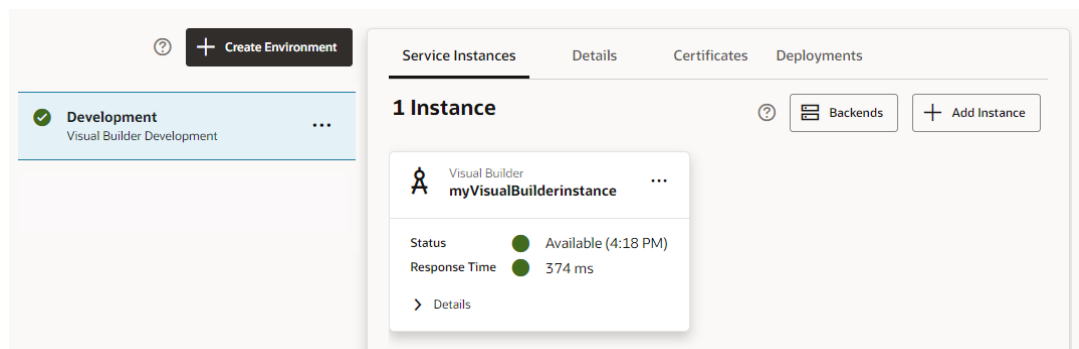
After the project is provisioned, the **Project Home** page opens where you can see a summary of the project's provisioning activities; default environment; default workspace; and Git, Maven, and NPM repositories. Review the activities feed and the Environments box for any errors.

When you create a project using the Visual Application template, these artifacts are created for you:

- A Git repository, which contains the visual application's source code. To see the Git repository's files, go to the **Project Home** page, click the **Repositories** tab, then click the Git repository name:



- A **Development** environment pointing to the Visual Builder development instance. In the left navigator, click **Environments** to see the **Development** environment:



- Build jobs that package and deploy the visual application's artifact to the Visual Builder development instance. By default, the **Visual-Application-Package** and **Visual-Application-Deploy** jobs are created for you. The **Visual-Application-Package** job generates the visual application's artifact file. The **Visual-Application-Deploy** job deploys the visual application's artifact file to the Visual Builder development instance.

In the left navigator, click **Builds** and then click the **Jobs** tab to see the build jobs:

Jobs (16)

Job Queue (0)

Pipelines (10)

Pipeline Queue (0)

Auto Refresh

+

Create Job

View Job Statistics

All Jobs

Successful Jobs

Failed Jobs

Test Failed Jobs

YAML Jobs

Status

Private

Weather

Job

Last Successful

Last Unsuccessful

Duration

Actions

Prod-Visual-Application-Package

N/A

N/A

N/A

Prod-Visual-Application-Deploy


N/A

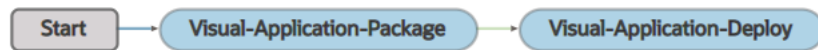
N/A

N/A

To run builds of the package and deploy jobs, you must first allocate VM build executors and make the appropriate deployment configurations. Without the appropriate configuration or VM executors, the builds won't run.

- A pipeline to run the build jobs in a sequence.


In the left navigator, click **Builds**  and then click the **Pipelines** tab. In the pipeline's row, click the **Actions** *** menu and select **View Layout**.




If you prefer to bypass the CI/CD pipeline and instead deploy directly to your Visual Builder instance, change the [CI/CD pipeline setting](#).

- A private workspace to edit the visual application in the VB Studio Designer.

In the left navigator, click **Workspaces**  to see the workspace:

Search	Q	Mine	Others	New ▼	Import	Clone From Git
 A workspace connects you to everything you need—the right Git repository, the right environment and, if required, the right sandbox. It's the foundation for your app. Tell me more about workspaces						
Workspace	Repository	Current Branch	Current Sandbox	Environment	Last Opened	Size
Workspace1	employee-management.git	branch1		Development	3 minutes ago	9.7 KB
Visual Application						

- By default, the project uses the organization's default markup language. Your project's users use the markup language to format wiki pages and comments. If required, you can change the project's markup language from the **Project Administration** page. See [Change a Project's Wiki Markup Language](#).
- A VM executor is created if this project is VB Studio's first project and no build VM executors had existed when you created the project. The VM executor uses the **System Default OL7 for Visual Builder** build executor template. You can use this VM executor to run build jobs that reference the **System Default OL7 for Visual Builder** template in the current project and other projects as well. See [VM Build Executors](#) for more information about VM executors and build executor templates.

In the left navigator, click **Organization**  and then click the **VM Build Executors** tab to see the VM executor.

Overview

1 VM

None of your 1 VM build executors are currently running.

None of your VMs are in an error state.

Search

Sleep Timeout

Update Selected

+

Create Free VM

Template

Region

Shape

VCN OCID

Details

Status

Actions

System Default OL7 for Visual Builder



us-ashburn-1

VM.Standard.E2.2

Pending

Configure the Deployment Job

The deployment job deploys the visual application's build artifact to your Visual Builder development instance. In the deployment job, specify the application's version and profile, and the credentials required to connect and deploy the build artifact to your Visual Builder development instance.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click the deployment job.
3. Click **Configure**.
4. Click **Configure** .
5. Click the **Steps** tab.
6. In the Authorization section, specify the authorization type to run this build step. With **Use OAuth** selected by default, you'll see the `Authorization is required` message, indicating that this build step needs a one-time authorization to handle OAuth requests to your environment's Visual Builder instance. Click **Authorize** and enter credentials to access your Visual Builder instance; you can also run the job manually and enter the credentials when prompted.

Either way, it is recommended that you authorize your OAuth connection during initial configuration. If you skip this step, you won't be able to publish your changes from the Designer and will need to complete the required authorization before attempting to deploy changes.

Once authorized, the `Authorization has been provided` message shows.

Note

OAuth is the recommended authorization type. Use Basic authentication only if you run into issues with setting up an OAuth connection. To use Basic authentication, select **Use Basic**, then enter the credentials of a user who can access the Visual Builder instance in **Username** and **Password**.

OAuth tokens (access and refresh) are cycled during regular use. A refresh token is used to obtain an access token whenever a user accesses the target instance. This refresh token is typically valid for seven days. (The token expiration time is set in the IDCS resource app and may be different based on your security requirements.) If the user authenticates with the target instance within the seven-day period, the active refresh token generates a new access token and a new refresh token. This cycle continues indefinitely as long as the refresh token stays valid. If the refresh token expires during extended periods of inactivity (say, when you're away on vacation), click **Renew Authorization** (or run the job manually, so you're prompted to authorize any expired OAuth tokens).

7. (Optional) To overwrite the application's default version, specify the new version in **Application Version**. Leave it empty to use the version defined in the application's `visual-application.json` file.

Don't deselect the **Include the application version in the URL** check box.

8. (Optional) In **Application Profile**, specify the development application profile. Leave it empty to use the application's default profile.

Your visual application accesses data from different servers for REST services and may need different security settings for different environments, such as development and production. Using application profiles, you can define different combinations of servers and security settings for each of your environments, and use them when deploying the application to an environment. This simplifies management of the visual application as you move through development to production. To learn more, see [About Application Profiles](#).

9. (Optional) To use the existing application's database, in **Data Management**, select **Keep existing environment data**.


To use a clean database for the application, in **Data Management**, select **Use clean database**.

10. Click **Save**.

Run the Pipeline Manually

The development build pipeline runs automatically when a commit is pushed to the Git repository's branch specified in the packaging job.

If you want to run the pipeline manually:


1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. In development pipeline's row, click the **Actions** *** menu and select **Run Pipeline**.

To monitor the pipeline and see each job's status, click the pipeline's name. To see a job's build log, click the job's name and click **View Log**.

If you want to run a job's build manually, open the job's details page and click **Build Now**. You can monitor its build on the job's details page.

View the Deployed Visual Application

After the deployment job has successfully run, you can view the deployed applications in the **Deployments** tab of the **Environments** page.

1. In the left navigator, click **Environments** .
2. Select the Visual Builder environment.
3. Click the **Deployments** tab.
4. If not enabled, click the **Visual Applications** toggle button.
5. If the Visual Builder instance is from a different identity domain, provide its access credentials.
6. Expand the app's name to see the deployed app's link.

The **Deployments** tab displays the applications you've deployed from the current project. It doesn't show applications deployed by other users of the project, or applications deployed from other projects. Here's an example:

Service Instances


Details

Certificates





Deployments

Visual Applications

Application Extensions

 This table shows deployments only for the current project.

X

Git Repository	Application URL Root	Service (Target)	Schema	Status	Version	Updated	Actions
▼ myvisualapp	myvisualapp	VisualBuilder-Dev	SP15594091... <div></div>	Deployed	0.1	Yesterday at 9:35 AM	...
<div> webapp</div>							
▶ myvisualapp	myvisualapp	VisualBuilder-Dev	SP15594091... <div></div>	Deployed	1.1	Yesterday at 11:52 AM	...
▶ myvisualapp	myvisualapp	VisualBuilder-Dev	SP15594091... <div></div>	Shared	vbshare_1	Mon at 3:22 PM	...

If you want to undeploy your deployed visual application, you can do so manually or through a job configuration. See [Undeploy a Visual Application](#).

Use CI/CD Pipelines for Deployment

Visual applications can be deployed to your development Visual Builder instance either directly or via continuous integration and delivery (CI/CD) pipelines.

- Direct deployment means that, when you click **Publish** in the Designer, your visual application is merged to a target branch in the remote repo, such as the `main` branch, and then deployed *directly* to the environment associated with your workspace.

This is the default deployment method for new workspaces you create.

- CI/CD pipelines, meanwhile, provide flexibility—for example, you can configure a pipeline to deploy dependent artifacts to the target environment in parallel, run builds on a specified schedule, or discard old builds and artifacts.

You can set this up by enabling a CI/CD pipeline for the target branch you want to merge to, and configuring that pipeline to suit your organization's needs.

📘 Note






Use a combination of whichever deployment options work best. For example, in your organization, maybe it's most efficient to deploy a visual application directly to your development Visual Builder instance using the **Publish** button. At the same time, you can leverage the use of CI/CD pipelines for deployments to other Visual Builder instances, as well as to automate certain lifecycle operations tasks.

Here's a brief rundown of a few key differences between the two publishing options:

Publishing Aspect	Publish Directly	Publish via CI/CD Pipelines
Deployment flexibility	The key benefits of publishing a visual application <i>directly</i> to a Visual Builder instance using the Designer's Publish button are speed and simplicity. The deployment process is straightforward, but not flexible.	Flexibility is the main reason to use CI/CD pipelines. When clicking Publish , maybe you'd like your visual application to be deployed not just to a DEV instance, but to other instances as well. You can also do things like configure a pipeline to download archived artifacts or discard old builds and artifacts, for example, or run builds on a specified schedule.
Speed	Publishing a visual application directly to a Visual Builder instance is fast. Deployment happens immediately, so end users can quickly view the changes.	When publishing a visual application using a CI/CD pipeline, the deployment process isn't immediate. You might need to wait for the entire process to complete before others can see the changes.
Deploying to multiple instances	You can deploy your visual application to your DEV instance directly, without using a CI/CD pipeline. To deploy that visual application to other instances, however, you must use CI/CD pipelines.	You can configure CI/CD pipelines to deploy changes to any instance, not just to your DEV instance. You can also configure a pipeline to deploy to multiple instances simultaneously.
Merge requests	Since deployment is immediate, you can't include merge requests as part of the publishing process. To get changes reviewed, therefore, merge requests must be created <i>before</i> clicking Publish .	When publishing a visual application using a CI/CD pipeline, you do have the option to build in merge requests.
Approval workflow	When publishing directly to a Visual Builder instance, there aren't any built-in approval workflow capabilities. If you have access to an environment, you can deploy to it.	With CI/CD pipelines, you can add an approval item that requires one or more authorized users to manually approve a step before executing the rest of its run. For example, a pipeline can automatically deploy a visual application to your DEV and TEST instances, but require a manager's approval before deploying to your PROD instance.
Logs and build history	Publishing directly to a Visual Builder instance doesn't provide a record of build details, log reports, and build history.	If you've used a CI/CD pipeline to deploy a visual application, prior logs and build history are available for review.
Support for rolling back visual applications to a previous version	If a visual application was published directly to a Visual Builder instance, that visual application can't be redeployed at a later time.	If a visual application was deployed using a pipeline and its build artifacts were archived, it can be redeployed at a later time, if needed.

Add Users to the Project

You must explicitly add users before they can work within a project, as explained in this table:

If you want to:	Do this:
Add a user to the project	<ol style="list-style-type: none"> 1. In the left navigator, click Project Home . 2. Click the Team tab. 3. Click + Create Member. 4. Click the Username drop-down list. 5. Under Users, select the user. If you can't find a particular user, enter the user's name or username in the search box. As you begin typing, users matching the search term are displayed. 6. From the membership option types, select the user's membership. 7. Click Add.
Add a group to a project	<ol style="list-style-type: none"> 1. In the left navigator, click Project Home . 2. Click the Team tab. 3. Click + Create Member. 4. Click the Username drop-down list. 5. Under Groups, select the group. 6. From the membership option types, select the membership you want to assign to the group's members. 7. Click Add.
Add multiple users or groups to the project	<ol style="list-style-type: none"> 1. In the left navigator, click Project Home . 2. Click the Team tab. 3. Click + Create Member. 4. Click the Username drop-down list. 5. From the drop-down list, select a user or a group. Click Username again to select another user or group. If you can't find a particular user, enter the user's name or username in the search box. As you begin typing, users matching the search term are displayed. 6. From the membership option types, select the user's membership. 7. Click Add.
Change a user's or a group's project membership	To change a user's or a group's project membership, click the Change Membership icon  . From the dropdown, select a new project membership (Contributor, Developer, Developer Limited, or Project Owner).
Remove a user or a group from the project	<p>Before removing a user, change the ownership of any assigned issues and merge requests to another user.</p> <p>For the user or the group you to remove, click Remove .</p>

Optional Configuration



After setting up the project, you can follow these optional steps to configure some advanced settings in your project. You can perform these steps at any time during your development cycle.

To perform this action:	See this:
Configure the packaging job to change the visual application's archive file names. By default, the packaging job generates two archive files: <code>sources.zip</code> and <code>built-assets.zip</code> . If you want to specify custom file names, configure the packaging job.	Step 7 in Configure the Packaging Job
Protect the Git repository's <code>main</code> branch for unapproved code updates. By default, a branch is accessible to all project users and anyone can make changes to its files. To restrict changes and push commits to the <code>main</code> branch, set restrictions on it to allow branch merges only after they are approved.	Set Merge Restrictions on the <code>main</code> Branch

Configure the Packaging Job

If you've set up your visual application to use CI/CD pipelines for deployment, the packaging job generates two archive files from your visual application's source files: a source archive file that contains the visual application's source files and a build artifact archive file. You have the option to specify custom file names for these archive files.


If you don't want to change the visual application's default archive file names, there's no need to configure the packaging job. However, if you do want to specify custom names, follow these steps:

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click the packaging job.
3. Click **Configure**.
4. Click **Configure** .
5. Click the **Steps** tab and navigate to the **Visual Application Packaging** section.
6. By default, the build jobs minifies the application's source code before running the build. If you don't want to minify the source files, deselect the **Optimize application** check box.

Minification is a process to remove the unnecessary characters (such as blank spaces, new lines, and comments) from the source code and reduce the size of the files, making the transfer of files consume less bandwidth and storage.
7. If you want to change the default names of the archive files, in **File names**, select **Use custom file names**.


In **Sources**, specify the visual application source archive file's name and path. In **Build Artifact**, specify the build artifact archive file's name and path.
You'll need both archive files to deploy the visual application.
8. Click **Save**.

If you change the default file names of archive files, then remember to configure the deployment job to use the same file names.

1. In the **Jobs** tab, click the deployment job.
2. Click **Configure**.
3. Click **Configure** .
4. Click the **Steps** tab and navigate to the **Visual Application Deployment** section.
5. In **File names**, select **Use custom file names**.
6. In **Sources and Build Artifact**, enter the same file names (with path) you specified in the packaging job.
7. Click **Save**.

Set Merge Restrictions on the `main` Branch

By default, the `main` branch is accessible to all project users and anyone can make changes to its files. To restrict changes and who can push commits to it, you may want to set restrictions on it to allow branch merges only after they are approved.

1. In the left navigator, click **Project Administration** .
2. Click **Branches**.
3. In **Repository** and **Branches**, select the Git repository and the `main` branch.
4. Select the **Requires Review** option.
5. In **Default Reviewers**, enter and select the users.

A default reviewer is a project member who is automatically added as a reviewer when a merge request is created on the branch.
6. From the **Approvals** drop-down list, select the minimum number of reviewers who must approve the review branch of a merge request, where the selected branch is the target branch
7. (Optional) To allow a review branch to be merged to the selected branch only if the last build of the linked job in Merge Request is successful, select the **Require successful build** check box.

To use this option, link a build job to a merge request.
8. (Optional) If you want to reset the approval status of reviewers if change is pushed to a branch after they have approved the merge request, select the **Reapproval needed when branch is updated** check box.
9. (Optional) To ensure changes pushed to the target branch match the contents of the review branch, select the **Changes pushed to target branch must match review content** check box.
10. (Optional) In **Merge Request Exempt Users**, specify users who can bypass the branch restrictions and merge the review branch of a merge request outside VB Studio or without required approvals.

This is useful if you want to allow some users to merge the review branch irrespective of review conditions being met.
11. Click **Save**.


Develop Your Application

Now that you've set up VB Studio and created a visual application project and added users to it, guide your organization members who develop visual applications to these documents to learn more about VB Studio.

- Build Enterprise Apps in Minutes in *Building Responsive Applications with Visual Builder Studio* describes how to create, edit, and publish visual applications.
- Get Started in *Using Visual Builder Studio* is a good resource, as it explains how to manage a project, create and manage issues and Agile boards, review source code with merge requests, and more.

Set Up Additional Deployment Instances

If you want to deploy a visual application to additional Visual Builder instances (for example, a production or QA instance), you must create a separate VB Studio environment and add the instance to it. You can add only one Visual Builder instance to an environment.

1. In the left navigator, click **Environments** .
2. Click **+ Create Environment**.
In **Environment Name** and **Description**, enter a unique name and description, and click **Create**.
3. In the Service Instances tab, click **Add Instance**.
4. Locate and add your Visual Builder instance:
 - With Instance Type set to **Visual Builder** and Add Instance Using set to **Instance List** (preferred), select your Visual Builder instance and click **Add**.
 - If you don't see the instance you want to add (because it's connected to a different OCI account), select the Authentication method as **Visual Builder Credentials**. In **Base URL**, enter the Visual Builder instance's URL; update the **Instance Name** if you want; enter the credentials of a user who can access the Visual Builder instance in **Username** and **Password**; and click **Add**.

If your target instance is in a different identity domain than your VB Studio instance, OAuth tokens (via three-legged OAuth flows) are used to secure programmatic access to the instance. Before any OAuth tokens can be created, you must provide authorization to acquire OAuth tokens. Click **Authorize** when prompted, then sign in with credentials to connect and deploy to the instance.

After you add a Visual Builder instance to an environment, the **Service Instances** tab displays a tile that shows basic information, such as status and response time, about the instance. Click **> Details** to see additional information (Visual Builder release version, Instance URL, etc.) or click the Home Page link to visit its home page URL.

If the newly added instance stays in the **Unknown** status for some time, it usually indicates that IDCS Application provisioning may have failed. VB Studio added the Visual Builder instance but can't access it. In such a case, click **Actions ***** and select **Remove** to delete the Visual Builder instance from the environment, and then click **Add** to add it again.

Note

OAuth tokens (access and refresh) are cycled during regular use. A refresh token is used to obtain an access token whenever a user accesses the target instance. This refresh token is typically valid for seven days. (The token expiration time is set in the IDCS resource app and may be different based on your security requirements.) If the user authenticates with the target instance within the seven-day period, the active refresh token generates a new access token and a new refresh token. This cycle continues indefinitely as long as the refresh token stays valid. If the refresh token expires during extended periods of inactivity (say, when you're away on vacation), click **Actions ***** and select **Renew OAuth Access**.

Once you're done, the Recent Activities feed on the **Project Home** page lists the newly created environment along with the Visual Builder instance you added.

Set Up the Project to Deploy for Production

After your development and test cycles are complete, you may want to configure the project to build and deploy visual applications to the Visual Builder production instance.

Before you proceed, contact the Visual Builder administrator and make sure that the Visual Builder production instance is properly configured and running. For example, make sure that the production instance's security options are configured and the instance points to the correct database.

Here's a summary of how to set up the VB Studio project for deployment:

To perform this action:	See this:
1. In the VB Studio project, create an environment for the Visual Builder production instance. The Visual Builder production instance usually resides in another identity domain.	Set Up Additional Deployment Instances
2. Create a production branch from the <code>main</code> branch. Use this branch to push your application to production.	Create a Production Branch
3. Create the production packaging and deployment build jobs, and set up the pipeline.	Create and Configure Production Build Jobs
4. (Optional) Restrict users who can edit the production jobs or run their builds.	Configure Job Protection Settings

Before you run production jobs, make sure that all code changes have been pushed to the production branch and there are no open merge requests.

Create a Production Branch

Follow your organization's guidelines to create a branch and protect it from unverified changes. To protect the branch, you can set merge restrictions, make the branch private and restrict who can push commits to it, or freeze it.

1. In the left navigator, click **Git** .
2. Click the **Refs** view and then click **Branches** .
3. From the **Repositories** drop-down list, select the repository.

4. Click **+ Create Branch**.
5. In the New Branch dialog box, in **Name**, enter the branch name. From the **Base** drop-down list, select the `main` branch as the base branch.
6. Click **Create**.

After creating the production branch, if you want to set merge restrictions on it, see [Set Review and Merge Restrictions on a Repository Branch](#). To freeze the branch or make it private, or set other restrictions, see [Protect a Branch](#).

Create and Configure Production Build Jobs

You need to set up some packaging and deployment jobs before you can deploy visual applications to your Visual Builder production instance. This topic explains how to do that.

✓ Tip



If you prefer, VB Studio can create and enable the build jobs and pipeline for you. In the visual application's Settings editor, under Building and Publishing, select the production branch and then click **Create CI/CD Pipeline**. You can then modify the newly created build jobs as needed (see the guidance below for additional configuration steps). For example, in the deployment job, the deployment target is always the environment associated with the workspace, so you'll need to change the target instance to your production instance.

Before you configure production build jobs, make a note of these:


- In the production packaging job, use the Git repository's production branch to generate production artifacts.
- In the development packaging job, if you changed the default file names of archive artifact files, get the new names and their paths. You'll need them when you configure the production packaging job.
- If you configured the development packaging job to overwrite the application's version defined in `visual-application.json`, get the new version. You'll need it when you configure the production packaging job.
- While configuring deployment build jobs, you specify whether to include the application's version in its URL. A visual application without the version in its URL is called a Live application. Usually, you deploy a live application to a production instance.
- If you deploy a version of a visual application that's never been deployed, VB Studio overwrites the last deployed version with the new version. VB Studio doesn't undeploy the previously deployed version from the production instance. It continues to remain on the target instance, but is inaccessible.
- If you want to redeploy a live application or a previously deployed version, undeploy it first, else the deploy build fails. To undeploy a previously deployed visual application version, [configure a undeploy build job and run it](#). You can't undeploy it manually from the **Environments** page.
- If you're deploying an app to a custom URL, you must configure the deploy job to deploy to the environment hosting the custom URL. You can only deploy the app using basic auth authentication, so you'll need to confirm that the credentials required to connect and deploy to the environment are correct.
- If you've created an application profile for production, get its name. You'll need it when you configure the production deployment job.

Create a Production Packaging Build Job

The production packaging job generates a visual application artifact that's ready to deploy.


1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select the **System Default OL7 for Visual Builder** template.
6. Click **Create**.
7. Click **Configure** .
8. Click the **Git** tab.
9. From the **Add Git** list, select **Git**.
10. In **Repository**, select the Git repository. In **Branch or Tag**, select the production branch.
11. Click the **Steps** tab.
12. From **Add Step**, select **Visual Application**, and then select **Package**.
13. By default, the build jobs minifies the application's source code before running the build. If you don't want to minify the source files, deselect the **Optimize application** check box.


Minification is a process to remove the unnecessary characters (such as blank spaces, new lines, and comments) from the source code and reduce the size of the files, making the transfer of files consume less bandwidth and storage.
14. If you want to change the default names of the archive files, in **File names**, select **Use custom file names**.

In **Sources**, specify the visual application source archive file's name and path. In **Build Artifact**, specify the build artifact archive file's name and path.
You'll need both archive files to deploy the visual application.
15. Click the **After Build** tab.
16. From **Add After Build Action**, select **Artifact Archiver**.
17. In **Files to archive**, enter the build artifact name. You can also use wild characters. For example, *.zip.
18. If you want to discard old artifacts of the build, click **Settings** . In the **General** tab, select the **Discard Old Builds** check box and specify the discard options.
19. Click **Save**.

Create a Production Deployment Build Job

The production deployment job deploys the visual application's artifact that was generated in the production packaging job to the Visual Builder production instance. Before you create the job, get the access credentials of a user who can connect and deploy to the Visual Builder production instance.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.

3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select the **System Default OL7 for Visual Builder** template.
6. Click **Create**.
7. Click **Configure** .
8. Click the **Before Build** tab.
9. From **Add Before Build Action**, select **Copy Artifacts**.
10. In **From Job**, select the production packaging job that generated the visual application's artifact.
11. In **Which Build**, select **Last Successful Build**.
12. Leave other fields with their default or empty values.
13. Click the **Steps** tab.
14. From **Add Step**, select **Visual Application**, and then select **Deploy**.
15. In **Target Instance**, select the Visual Builder production instance.
16. In the Authorization section, specify the authorization type to run this build step. With **Use OAuth** selected by default, you'll see the *Authorization is required* message, indicating that this build step needs a one-time authorization to handle OAuth requests to your environment's Visual Builder instance. Click **Authorize** and enter credentials to access your Visual Builder instance; you can also run the job manually and enter the credentials when prompted.

Either way, it is recommended that you authorize your OAuth connection during initial configuration. If you skip this step, you won't be able to publish your changes from the Designer and will need to complete the required authorization before attempting to deploy changes.

Once authorized, the *Authorization has been provided* message shows.

Note

OAuth is the recommended authorization type. Use Basic authentication only if you run into issues with setting up an OAuth connection. To use Basic authentication, select **Use Basic**, then enter the credentials of a user who can access the Visual Builder instance in **Username** and **Password**.

OAuth tokens (access and refresh) are cycled during regular use. A refresh token is used to obtain an access token whenever a user accesses the target instance. This refresh token is typically valid for seven days. (The token expiration time is set in the IDCS resource app and may be different based on your security requirements.) If the user authenticates with the target instance within the seven-day period, the active refresh token generates a new access token and a new refresh token. This cycle continues indefinitely as long as the refresh token stays valid. If the refresh token expires during extended periods of inactivity (say, when you're away on vacation), click **Renew Authorization** (or run the job manually, so you're prompted to authorize any expired OAuth tokens).

17. In the production packaging job, if you changed the default file names of archive artifact files, then in **File names**, select **Use custom file names**.

In **Sources** and **Build Artifact**, enter the same file names (with path) you specified in the packaging job.

If you didn't change the file names, use the default option.

18. (Optional) If you configured the development deployment job to overwrite the application's default version, specify the same version in **Application Version**. Leave it empty to use the version defined in the application's `visual-application.json` file.

If the version is already deployed, undeploy it first.

19. Deselect the **Include the application version in the URL** check box.

When deploying to a production instance, don't include the application's version in the deployed application's URL.

20. (Optional) In **Application Profile**, specify the production application profile. Leave it empty to use the application's default profile.

Your visual application accesses data from different servers for REST services and may need different security settings for different environments, such as development and production. Using application profiles, you can define different combinations of servers and security settings for each of your environments, and use them when deploying the application to an environment. This simplifies management of the visual application as you move through development to production. To learn more, see *About Application Profiles*.


21. To use a clean database for the application, in **Data Management**, select **Use clean database**.

22. Click **Save**.

Configure a Production Job's Privacy Setting


Mark a job as private to restrict who can see or edit a job's configuration, or run its build.

A private job must be run manually. It won't run if a non-authorized user tries to run the job directly, through an SCM/periodic trigger or a pipeline.

1. In the left navigator, click **Project Administration** .
2. Click **Builds**.
3. Click the **Job Protection** tab.
4. From the jobs list, select a production job.
5. Select the **Private** option.
6. In **Authorized Users**, add yourself.


Then, to add other users, select them individually or, under **Groups**, select a group (or groups) of users.

7. Repeat the steps 4-6 for the deployment production job.
8. Click **Save**.

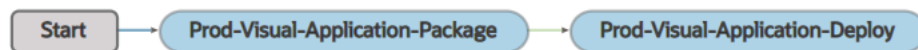
A private job shows a **Lock**  icon in the jobs list on the right side of the Job Protection page, in the **Jobs** tab of the **Builds** page, and in the pipelines.

Create and Configure a Pipeline

To ensure the production deployment job runs automatically after the production packaging job, create a pipeline and set the dependency.

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. Click **+ Create Pipeline**.
4. In the Create Pipeline dialog box, in **Name** and **Description**, enter a unique name and description.
5. Click **Create**.
6. On the Pipeline Configuration page, right-click the **Start** node and select **Add New Start** jobs.
7. Click in the **Select new on success job(s)** field, select the packaging job and click **Save**.
8. Right-click the packaging job and select **Add**, then **Add New On Success Jobs**.
9. Click in the **Select new on success job(s)** field, select the deployment job, and click **Save**.

Here's an example of the finalized pipeline:



10. Click **Save**.

Run the Pipeline

When you're ready to deploy the visual application to the production instance, run the production pipeline.

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. In development pipeline's row, click the **Actions** *** menu and select **Run Pipeline**.

After a successful build, you'll find the deployed application's link in the **Deployments** tab of the **Environments** page.

To view the latest build log of a job, open the **Builds** page, click the job's name, and then click **Build Log**.

Undeploy a Visual Application

If you want to undeploy a visual application that's deployed to Visual Builder, you can do so manually or through a job configuration.


You can undeploy a visual application manually if it is deployed to a Visual Builder instance that's in the same identity domain as VB Studio. If the visual application is deployed to a Visual Builder instance in a different identity domain than VB Studio (such as your production instance) or the Visual Builder instance was added to an environment through credentials, you should configure a build job to undeploy it.

Undeploy a Visual Application Manually

You can undeploy a visual application that's deployed to your current identity domain's Visual Builder manually from the **Deployments** tab of its environment.



① Note

If your visual application is deployed to an identity domain that's different than your VB Studio instance, you'll need to follow the instructions in [Configure a Build Job to Undeploy a Visual Application](#) to undeploy the application.

1. In the left navigator, click **Environments** .
2. Select the environment where the visual application is deployed.
3. Click the **Deployments** tab.
4. Expand the application.
5. For the visual application to undeploy, click **Actions** *** and select **Undeploy**.
6. If the visual application is a live application (without the version number in the URL), select the **Yes, I'm sure** check box.
7. Click **Undeploy**.

Configure a Build Job to Undeploy a Visual Application

To undeploy a visual application deployed to the production instance, or another identity domain's Visual Builder instance, configure a build job. Before you configure the job, get the access credentials of a user who can access the Visual Builder instance where the visual application is deployed.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select the **System Default OL7 for Visual Builder** template.
6. Click **Create**.
7. Click **Configure** .
8. Click the **Steps** tab.
9. From **Add Step**, select **Visual Application**, and then select **Undeploy**.
10. In **Instance**, select the Visual Builder instance where the application is deployed.
11. In the Authorization section, specify the authorization type to run this build step. With **Use OAuth** selected by default, you'll see the *Authorization is required* message, indicating that this build step needs a one-time authorization to handle OAuth requests to your environment's Visual Builder instance. Click **Authorize** and enter credentials to access your Visual Builder instance; you can also run the job manually and enter the credentials when prompted.

Either way, it is recommended that you authorize your OAuth connection during initial configuration. If you skip this step, you will need to complete the required authorization before attempting to undeploy changes again.

Once authorized, the `Authorization has been provided` message shows.

Note

OAuth is the recommended authorization type. Use Basic authentication only if you run into issues with setting up an OAuth connection. To use Basic authentication, select **Use Basic**, then enter the credentials of a user who can access the Visual Builder instance in **Username** and **Password**.

OAuth tokens (access and refresh) are cycled during regular use. A refresh token is used to obtain an access token whenever a user accesses the target instance. This refresh token is typically valid for seven days. (The token expiration time is set in the IDCS resource app and may be different based on your security requirements.) If the user authenticates with the target instance within the seven-day period, the active refresh token generates a new access token and a new refresh token. This cycle continues indefinitely as long as the refresh token stays valid. If the refresh token expires during extended periods of inactivity (say, when you're away on vacation), click **Renew Authorization** (or run the job manually, so you're prompted to authorize any expired OAuth tokens).

12. In **Application URL Root** and **Application Version**, enter the visual application's root URL and its version.

You can find the application's root URL and its version from the **Deployments** tab of the environment where the visual application is deployed. Here's an example:

Service Instances


Details


Certificates





Deployments

Visual Applications

Application Extensions

 This table shows deployments only for the current project.



Git Repository	Application URL Root	Service (Target)	Schema	Status	Version	Updated	Actions
▼ myvisualapp	myvisualapp	VisualBuilder-Dev	SP15594091... <div></div>	Deployed	0.1	Yesterday at 9:35 AM	...
<div> webapp</div>							
▶ myvisualapp	myvisualapp	VisualBuilder-Dev	SP15594091... <div></div>	Deployed	1.1	Yesterday at 11:52 AM	...
▶ myvisualapp	myvisualapp	VisualBuilder-Dev	SP15594091... <div></div>	Shared	vbshare_1	Mon at 3:22 PM	...

13. Click **Save**.
14. To run a build, click **Build Now**.

After the build is successful, open the build's log. You'll see a similar message:

```
[2020-01-01 12:00:00] === Begin VB App Ops Undeploy Application===
[2020-01-01 12:00:00] Application employee-manager-visual-web-app/0.1 has
been undeployed
[2020-01-01 12:00:00] === End VB App Ops Undeploy Application ===
Executor log size 364 B (364)
[2020-01-01 12:00:02]
```

```
[2020-01-01 12:00:02] Build completed.  
[2020-01-01 12:00:02] Status: DONE Result: SUCCESSFUL Duration: 7.0 sec
```

5

Set Up VB Studio for CI/CD

If you want to use VB Studio to manage a large software development project that uses Oracle Cloud Infrastructure (OCI) cloud services, and doesn't involve either Oracle Cloud Applications or visual applications, read this chapter to familiarize yourself with VB Studio's build system.

VB Studio's build system enables your organization's developers to create jobs that build, package, test, and deploy applications and extensions. In VB Studio, builds run either in VM build executors (VM executors, for short) or in Docker executors. Build executor templates define the operating system and software packages for VM executors, and Docker images define the operating system and software packages for Docker executors.

To build their applications, your organization's developers define *build jobs* in their VB Studio projects. A *job* is a configuration that defines the location where the application's source files are found, the commands needed to run the job and test scripts, the instructions for generating and packaging artifacts, and the location where those artifacts will be deployed.

To run builds, your organization's developers need some Compute instances with operating systems and software packages installed. For example, to build Node.js applications, they'll need Node.js installed on an OCI VM Compute instance, which they'll use to run their CI/CD builds.

Before you start configuring CI/CD for VB Studio, you need to make several decisions that will guide you through the configuration process:

- [Step 1: Decide whether to use the free VM build executor](#)
This step is required. See [VB Studio's Free VM Build Executor](#) to understand what this build executor provides and whether it meets the needs of your organization's members. If it doesn't, you'll need to configure VB Studio to connect to your own OCI paid account.
- [Step 2: Decide whether to run VM build executors or Docker executors](#)
This step is required. In VB Studio, builds run either in VM build executors or in Docker executors. If you're not sure which type you need, see [What are Build Executors and Build Templates?](#).
- [Step 3: Set up VB Studio to run VM build executors](#)
This step is optional. If you decided to go with VM Build executors in Step 2, use this step. If you decided to use Docker executors, go to Step 4. You must use one or the other.
- [Step 4: Set up VB Studio to run builds on Docker executors](#)
This step is optional. If you decided to go with Docker executors in Step 2, use this step. If you decided to use VM build executors, go to Step 3. You must use one or the other.
- [Step 5: Set Up Users and Groups](#)
This step is required. If you want to add groups instead of adding users individually, see [Manage Your Organization's Groups](#).

Step 1: Decide whether to use the free VM build executor

Required step: Decide whether to use the free build executor or an OCI paid account before you create the VB Studio instance.

To package and deploy the applications you configure, VB Studio uses *continuous integration and continuous delivery* (CI/CD) build jobs. To run those builds, VB Studio can use either the free build executor or your own OCI paid account.

To help you decide, [OCI Resources in VB Studio](#) provides a comprehensive comparison, but here's a quick primer with links to the required setup for each option:

Option	Description	Where to Get More Details
Free VM build executor	VB Studio comes preconfigured with one OCI Compute VM to run builds.	See VB Studio's Free VM Build Executor to learn more about the capabilities of the free VM build executor.
Your OCI paid account	Use your own OCI paid account so you can add more build executors to reduce the wait time for your organization's members, create custom build executor templates, or use advanced features (such as use your own VCN or use a different VM shape). In addition, your build artifacts are stored in OCI Object Storage buckets, and you have the ability to increase storage space when needed.	See Create a VB Studio Instance in Your Oracle Cloud Account . (You can also connect it later , as well.)

If you later change your mind, you can always switch between the two, but you *might* need to manually migrate some data. Data migration shouldn't be a concern, however, if you're using your OCI account solely to run CI/CD builds. See [Change Your OCI Account](#).

Note

If you're using the free build executor to deploy *visual applications*, you must manually authorize the VB Studio instance to access Visual Builder instances connected to your OCI account. Without this authorization, your users won't be able to create environments with a Visual Builder instance to deploy their visual applications. See [Authorize VB Studio to Access Visual Builder and Integration Instances](#) for the steps to complete this authorization.

OCI Resources in VB Studio

VB Studio uses your OCI resources to run builds that package and deploy your application, and to store build artifacts. As part of your VB Studio setup, you should decide if the built-in free account is sufficient, or if you want VB Studio to use your OCI paid tenancy to create Compute instances and Storage objects for CI/CD build jobs.

Review the differences between both options to decide if you need the additional resources that your OCI paid account provides:

- **Built-in free account:** The VB Studio instance comes preconfigured with a built-in free account that offers [one OCI Compute VM](#) to run builds. (Available in all regions in the OC1 realm and will be made available in select other realms, such as OC2, OC3, and OC4, in the future.)
- **OCI paid account:** If you're subscribed to OCI, you can use your own account's OCI Compute VMs to run builds. To use your own OCI Compute VMs, configure VB Studio to [connect to your OCI account](#).

This table summarizes the differences between the built-in free account and your OCI paid account:

Features	Built-In Free Account	Your OCI Paid Account	What it means
Number of build executors	1	Unlimited	<p>A VM executor can run only one build at a time. The more VM executors you have available, the more builds your organization's members can run at the same time. If members trigger multiple builds while VM executors are already running, they must wait until a build running on the VM executor is complete.</p> <p>If VB Studio is connected to your OCI account, you can add multiple VM executors to reduce the build wait time for your organization's members.</p>
Storage for build artifacts	50 GB	Unlimited	<p>When your organization's members run builds, VB Studio saves the generated build artifacts in its configured storage space. If you're using the built-in free account, the build artifacts are stored in a limited storage space. If you run out of storage space, builds that generate artifacts will fail. If this happens, you must either remove some artifacts from the storage, or switch to an OCI account. You can't increase the storage space allocated to your built-in free account.</p> <p>If VB Studio is connected to your OCI account, the artifacts are stored in OCI Object Storage buckets. You can always increase the storage space to store more artifacts.</p>

Features	Built-In Free Account	Your OCI Paid Account	What it means
VM shape	Fixed shape	Custom shapes, including shapes offered by OCI	The built-in free account offers a fixed shape OCI Compute VM. If VB Studio is connected to your OCI account, you can choose any VM shape offered by OCI. To learn more about shapes, see VM Shapes .
VCN	VB Studio's default VCN	VB Studio's default VCN or your custom VCN	If VB Studio is connected to your OCI account, you can add VM executors in a custom VCN to access Oracle Cloud services that are running in the same VCN.
Wait time	5 minutes (fixed)	Custom wait time	After running a build, a VM executor waits for some time to run any queued builds. If no builds run in the wait time period, the VM executor stops. If a build triggers after a VM executor has stopped, the VM executor takes some time to start before it could run the build. If VB Studio is connected to your OCI account, you can customize the wait time for your VM executors.
Maven and NPM repositories	Not available	Available	If VB Studio is connected to your OCI account, your organization's members can use their projects' hosted Maven and NPM repositories to upload custom dependencies and binary files.

VB Studio's Free VM Build Executor

The free VM build executor is available in the VB Studio's built-in free account and runs on VB Studio's default VCN.

Note

The free VM build executor feature is available only for Commercial OC1 realm regions.

The more VM executors you have available, the more builds your organization's members can run at the same time. As the built-in free account offers only one VM executor, your VB Studio organization members can run only one build at a time. If members trigger multiple builds while a build is already running, they must wait until the build running on the VM executor is complete. The VM executor then runs the queued builds on a first-come-first served basis.

The free VM build executor template, called **System Default OL7 for Visual Builder**, includes the necessary software to run extension and visual application builds. You can't change the default executor template's configuration. This table lists the software defined in **System Default OL7 for Visual Builder**:

Software	Version
Oracle Linux operating system	8.x
Oracle Java SE	17.0.17
Ant	1.10.5 or later
C++ Compiler (cpp/gcc)	8.5.0 or later
Firefox	ESR 140.5.0 or later
Git	2.43.7 or later
Grunt	1.6.1 or later
Grunt-Cli	1.5.0 or later
Jq	1.6 or later
JUnit 4	4.12 or later
Maven	3.9.9
Node.js	22.6.0
NPM	10.8.2 or later
Python2 (including Virtualenv)	2.7.18 or later
Ruby	2.5.9p229 or later
Xvfb	1.20.11 or later


If you want to add more executor templates to reduce the wait time for your organization's members, to create custom executor templates with different software configurations, or to use advanced features (such as using your own VCN or using a different VM shape), you should configure VB Studio to connect to your own OCI paid account.

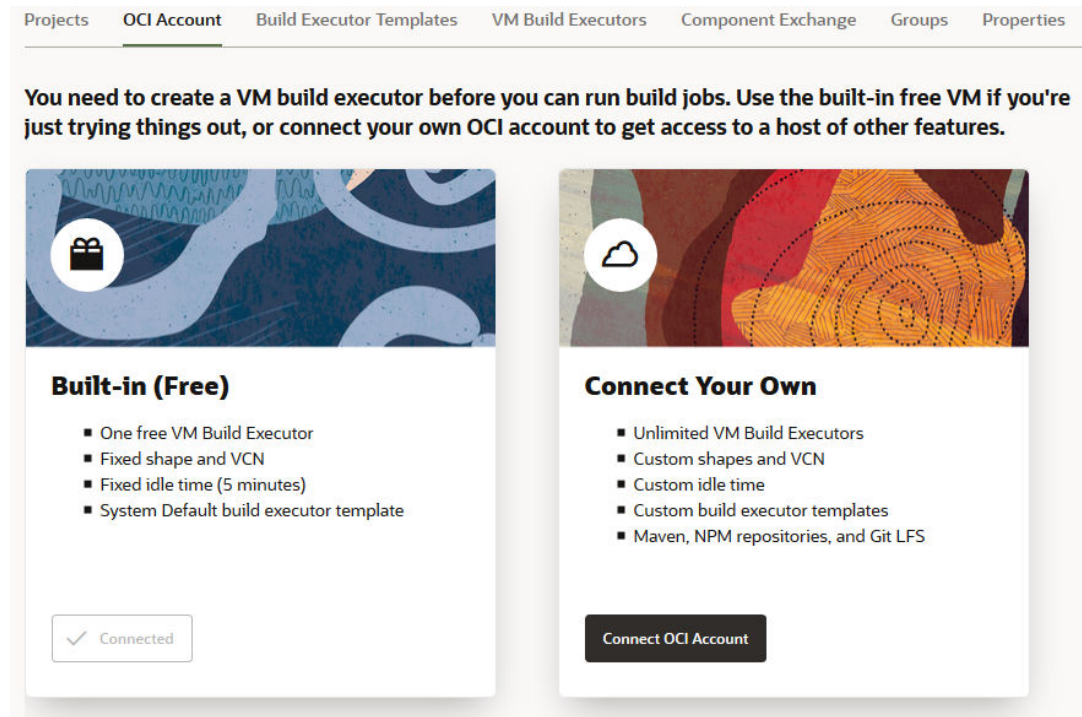
Configure VB Studio to Use the Free VM Executor for Build Jobs and Pipelines

In a VB Studio visual application project, you use build jobs to compile the source code, package the visual application, and deploy it to a Visual Builder instance. The builds and pipelines run on build executors, also called VM build executors. These VM executors are Oracle Cloud Infrastructure (OCI) VM Compute instances dedicated to run VB Studio builds.

In some Oracle Cloud regions and data centers, VB Studio is available preconfigured with a built-in free account, which provides one free VM executor that you can use to run build jobs that package and deploy your extensions. However, there are some limitations (see [VB Studio's Free VM Build Executor](#)) associated with the free VM build executor, so you may want to connect to your own OCI paid account instead, if you have one. See [OCI Resources in VB Studio](#) for more a more comprehensive comparison between the built-in free account and your OCI paid account.

To find out whether your VB Studio instance is connected to the built-in free account, follow these steps:

1. In the left navigator, click **Organization** .
2. Click the **OCI Account** tab.
You should see a similar page:




Depending on your VB Studio's data center, you may or may not see the **Built-in (Free)** option.

What do you see?	What you need to do:
I see the Built-in (Free) option	If you're trying out extensions, no additional configuration is required. Go ahead and create your extension project . VB Studio creates the free VM build executor when you create your first project.
I see the Built-in (Free) option, but want to run builds without any limitations	Configure VB Studio to connect to your OCI paid account and add VM executors . If you're new to OCI, see Welcome to Oracle Cloud Infrastructure .
I don't see the Built-in (Free) option, but have access to an OCI paid account	The built-in free account isn't available in your data center. You should configure VB Studio to connect to your OCI paid account and add VM executors .

Before you create a project, note that in a VB Studio instance that has no projects and no VM executors, the first VM executor is created for you when you create the first project. If the project isn't the initial one, the VM executor must be created manually.

Add the Free VM Build Executor Manually

If you're using VB Studio's built-in free account, the free VM build executor is added when you create your first project. Here's how to add the VM executor manually if you've deleted it or it wasn't added when you created the project:

1. In the left navigator, click **Organization** .

- Click the **VM Build Executors** tab.

If you see a VM executor of the **System Default OL7 for Visual Builder** template, the free VM build executor is available in your VB Studio instance. Ignore the remaining steps.

Overview		1 VM	None of your 1 VM build executors are currently running.		None of your VMs are in an error state.	
Search		Sleep Timeout		Update Selected		Create Free VM
Template	Region	Shape	VCN OCID	Details	Status	Actions
<input type="checkbox"/> System Default OL7 for Visual Builder	us-ashburn-1	VM.Standard.E2.2			Pending	...

If you don't see the VM executor, jump to the next step.

- Click **Create Free VM Build Executor**.

Projects
OCI Account
Build Executor Templates
VM Build Executors
Component Exchange
Groups
Properties

You need to create a VM build executor before you can run build jobs. Use the built-in free VM if you're just trying things out, or connect your own OCI account to get access to a host of other features.

Free VM Build Executor

You're currently using the built-in free VM, which allows you to create one VM build executor with these attributes:

- Fixed shape and VCN
- Fixed idle time (5 minutes)
- System Default Build Executor Template

Create Free VM Build Executor

Custom VM Build Executor

Connect your OCI account to take advantage of:

- Unlimited VM Build Executors
- Custom shapes and VCN
- Custom idle time
- Custom Build Executor Templates
- Maven, NPM repositories, and Git LFS

Connect OCI Account

Connect to Your OCI Paid Account

If you've determined that you require the additional resources you get with your own OCI paid account, as opposed to the built-in free account, you'll need to connect it to your VB Studio instance.

- Before you do anything, make sure you have the required roles. See [Get the Right Roles](#).
- You can then configure VB Studio to connect to your own OCI paid account.
 - If you haven't created a VB Studio instance yet, you can create the instance and connect it to your OCI paid account at the same time. See [Create a VB Studio Instance in Your Oracle Cloud Account](#).

- If your VB Studio instance already exists, you can switch OCI accounts. See [Change Your OCI Account](#).

If you aren't authorized to create and manage OCI resources, ask someone who can, like your tenancy administrator.

If you ever need to switch OCI compartments or accounts, refer to the topics in [Manage the VB Studio Instance](#).

Get the Right Roles

If you're the OCI account subscriber, you already have the required roles to set up the VB Studio instance. If you're not the subscriber, make sure that you've been assigned the correct roles in OCI Identity and Access Management.

You must be assigned this role:	To ...
Cloud Account Administrator or Tenancy Administrator	Set up Oracle Cloud Infrastructure (OCI) compartments, policies, dynamic groups, and permission to create and manage the VB Studio instance. If you're not an account or tenancy administrator, contact the administrator to add you to the OCI_ADMINS IDCS group.
Identity Domain Administrator or User Administrator	Add users and assign IDCS roles. If you're not an IDCS domain or user administrator, make sure that your OCI group is assigned a policy like this: Allow group MyGroup to inspect identity-providers in tenancy

Create a VB Studio Instance in Your Oracle Cloud Account

Create your own VB Studio instance if you're using VB Studio to manage a large software development project that uses Oracle Cloud Infrastructure (OCI) cloud services, and *doesn't* involve the extension of Oracle Cloud Applications.

Note

If you're using VB Studio just to configure and extend Oracle Cloud Applications, skip this. Instead, you'll use the VB Studio instances that Oracle provisions for you. You can also skip this if you're going to create visual applications using one of the VB Studio instances provisioned with your Oracle Cloud Applications account. See [Work With Multiple VB Studio Instances](#).

Note that when you sign up for Oracle Cloud Infrastructure, a set of *service limits* is configured for your tenancy (account), which is the quota set for a given resource. The VB Studio service limit provides **one VB Studio instance per region that your tenancy spans**. If your tenancy spans multiple regions, each region can have one VB Studio instance. This is on top of any VB Studio instance that Oracle provisions for you as an Oracle Cloud Applications customer.


1. In a web browser, go to <https://cloud.oracle.com>.

To view the list of supported browsers for the Oracle Cloud Console, also called the OCI console, see [Supported Browsers](#).

2. In **Cloud Account Name**, enter your Oracle Cloud account or tenancy name and click **Next**.
3. On the Single Sign-On (SSO) panel, if required, select your identity domain and click **Continue**.


Note

You'll be creating a VB Studio instance in the identity domain you're logged into; there is no mechanism to request a different domain. As long as you have the proper user credentials, you should be able to create the instance in any domain you wish. All users who wish to access the new VB Studio instance must be in its same domain.

4. Enter your Oracle Cloud account credentials and click **Sign In**.
If you've recently created your Oracle Cloud account, wait for some time to see your services in the Oracle Cloud Console.
5. In the upper-left corner, click **Navigation Menu** .
6. Select **Developer Services** and then select **Visual Builder Studio**.
7. On the Visual Builder Studio instances page, click **Create Visual Builder Studio**.

Visual Builder Studio instances

Only one Visual Builder Studio instance can be created, and it must be created in the root compartment.

 Search and Filter

Applied filters **Compartment** **(root)**

Create Visual Builder Studio

The Create Visual Builder Studio instance page is displayed.

8. In **Instance name**, enter the VB Studio's instance name.
9. The **Compartment** field is read-only and always displays the root compartment.
10. (Optional) Add one or more tags to this resource.

Default tags, which help you identify the instance, always get created.

If you aren't sure whether to add tags, skip this option. You can always [add tags](#) later. For more information about tagging resources, see [Resource Tags](#).

- a. In the Tags section, click **Add tag**.
- b. Select the **Tag namespace**, or select **None** to apply a free-form tag.
- c. Select or enter the **Tag key**.
- d. In **Tag value**, select or enter a value.

e. To apply another tag, click **Add tag**.

11. Click **Next**.

12. On the CI/CD setup page, the **Yes, I authorize this** checkbox is selected by default.

This authorizes VB Studio to create and use Compute and Storage instances, and automatically creates a policy that authorizes VB Studio to perform the necessary operations in the selected compartment for the CI/CD resources. It also creates a policy authorizing VB Studio to access Oracle Visual Builder and Oracle Integration instances connected to your OCI paid account. For this automatic policy creation to succeed, the user who creates the instance must also have permissions to manage policies in the root compartment. See [Get the Right Roles](#).

Typically, tenancy administrators have these permissions, so the easiest way to create a VB Studio instance is to have a tenancy administrator do it. However, if that isn't feasible, it's also possible to create the VB Studio instance without the CI/CD authorization and then [set it up later](#).

Note

If you want to use the built-in free account's free VM build executor, **deselect** the authorization checkbox. See [VB Studio's Free VM Build Executor](#) for information about the resources provided with the free VM build executor. Make sure you [manually authorize this VB Studio instance to access Visual Builder and Integration instances](#) connected to your OCI account, so you can add these instances to an environment later on.

13. If you keep the default "selected" setting of the **Yes, I authorize this** checkbox, in **Compartment**, select the compartment to host VB Studio's Compute and Storage instances.

The required policies will be applied to the selected compartment.

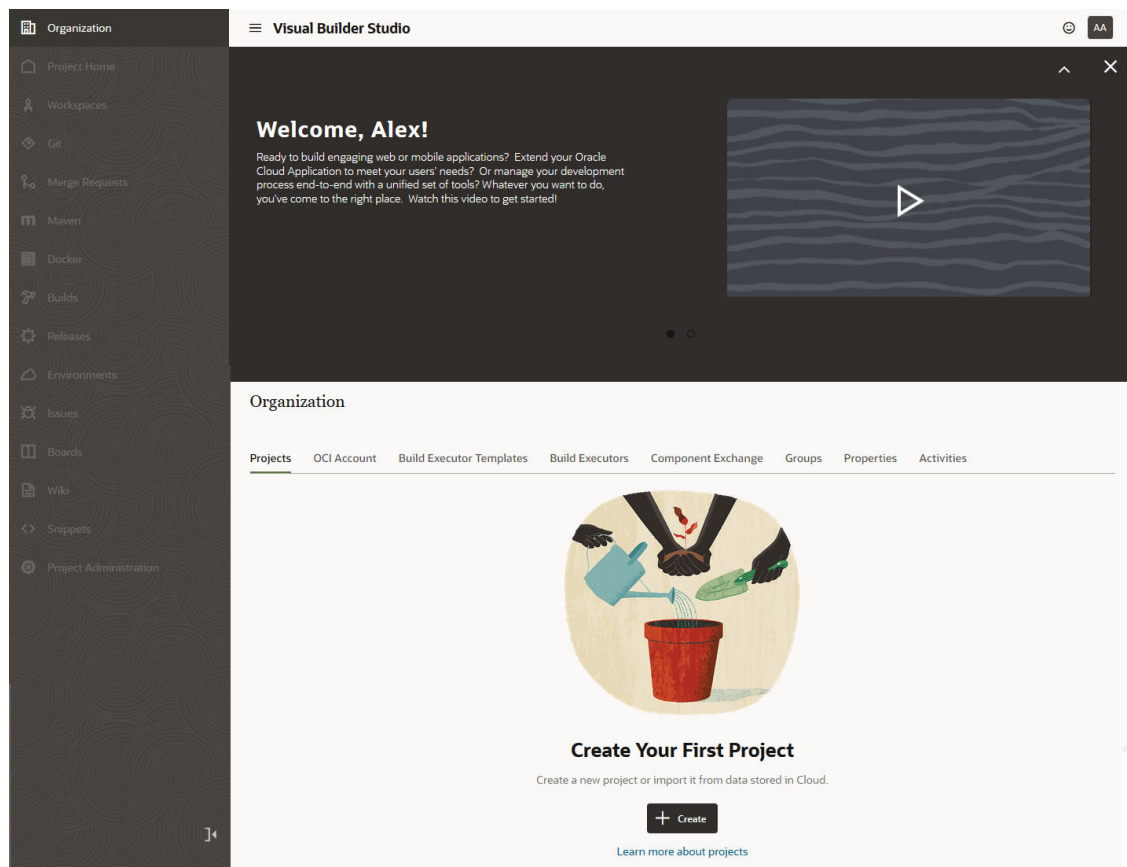
14. (Optional) Click **Save as stack** to create a stack in Resource Manager.

The stack can be used to install, configure, and manage the resource through the Resource Manager service. See [Creating a Stack from a Resource Creation Page](#).

15. Click **Create Visual Builder Studio**.

Once the service instance is created, click the instance's link to view its detail page. From here, you can click **Service console** to open VB Studio's Organization page.

Here's an example:



What are the next steps?

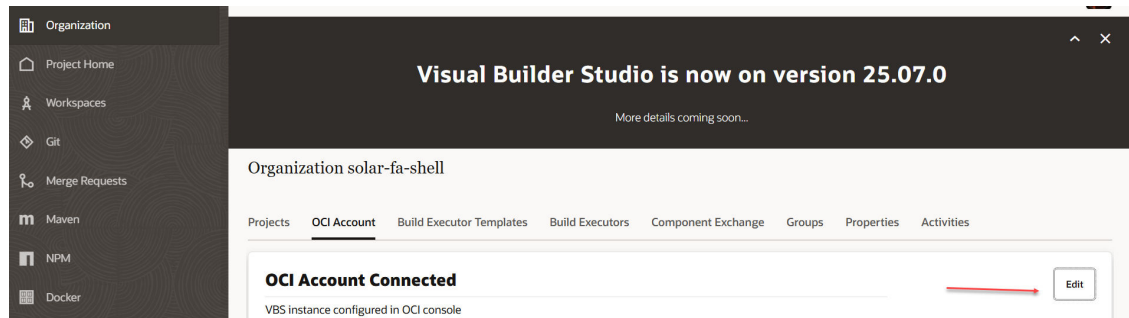
After you create the instance:

1. You're assigned the **DEVELOPER_ADMINISTRATOR** IDCS role, which enables you to set up VB Studio, manage the VM build executors, and update the organization details. After you're assigned the role, you become a VB Studio *Organization Administrator*.
2. You'll receive an email with the subject **Verify your Oracle Visual Builder Studio email**. Open the email and click the URL link in the email body to verify your email. If you don't do this, you won't receive important email notifications.
3. After verifying the email address, you'll receive another email with the subject **Welcome to Oracle Visual Builder Studio**. This email contains the VB Studio URL that you can bookmark.
4. Set up VB Studio's build system for CI/CD, which enables your organization's developers to create jobs to build, package, test, and deploy their apps. See [Set Up VB Studio for CI/CD](#).
5. As part of that CI/CD setup, you must provide access for additional VB Studio users. See [Step 5: Set Up Users and Groups](#) to add individual users or [Manage Your Organization's Groups](#) to add groups of users.
6. If you created a VB Studio instance so that your users can develop visual applications and deploy them to Visual Builder runtime instances, see [Set Up VB Studio for Developing Visual Applications](#).

Note

If your users are using the free VM account to develop and deploy visual applications, make sure the VB Studio instance is authorized to access Visual Builder instances connected to your OCI account. Without this authorization, they won't be able to create environments with a Visual Builder instance to deploy their visual applications. See [Authorize VB Studio to Access Visual Builder and Integration Instances](#) for the steps to complete this authorization.

If you need to make any changes to your OCI account, you can use the **Edit** button on the Organization page's OCI Account tab to get to the Oracle Cloud Console:



Step 2: Decide whether to run VM build executors or Docker executors

Required step: In VB Studio, builds run either in VM *build executors* (VM executors, for short) or in Docker executors. Before your developers create jobs and run builds, you must decide whether to run your organization's builds on VM executors or on Docker executors, and then create the required resources.

If you already know what you want to do, skip ahead to either [Step 3: Set up VB Studio to run VM build executors](#) or [Step 4: Set up VB Studio to run builds on Docker executors](#).

If you're not sure which one to use, see [VM Build Executors](#) or [Docker Build Executors and Docker Images](#).

Note

Regardless of which executor type you pick, you must configure VB Studio to connect to your OCI paid account if you want to add more than one VM executor, or if you want to create or manage Docker images.

What are Build Executors and Build Templates?

Builds in a build system run in *build executors*. The operating system and software packages required to run builds on build executors are defined in *build templates*. In VB Studio, builds run either in VM build executors (VM executors, for short) or in Docker executors.

VM Build Executors

VM build executors are OCI VM Compute instances dedicated to run builds of *jobs*, which your organization's members define in VB Studio projects.

A VM executor is always associated with a build executor template. When your organization's members create jobs, they simply associate the appropriate executor template with the job. When the job's build triggers, the VM executor that's associated with the executor template starts automatically. Oracle charges you only when a VM executor is active, runs a build, or is preparing itself to run a build.

This table describes the different states of a VM executor:

State	What does it mean?	Does it cost?
Pending	After you add a VM executor, it is in this state until it runs a build. When a VM executor starts from this state, it takes some time to install the operating system and software packages.	No
Starting	The VM executor is starting. If the VM executor starts from the Pending state, VB Studio installs the operating system and software packages on the VM executor's assigned boot volume. This takes time. If the VM executor starts from the Stopped state, VB Studio uses software packages and the operating system from the previous run's saved boot volume. VB Studio periodically checks all executor templates for updates. If an executor template is found with new updates, VB Studio deletes the preserved boot volume of all stopped VM executors that reference the executor template and changes their status from Stopped to Pending .	Yes
Available	Operating system and software packages are installed, and the VM executor is ready to run a build.	Yes
In Use	The VM executor is running a build. After the running build is complete, the VM executor returns to the Available state.	Yes

State	What does it mean?	Does it cost?
Stopping	The VM executor is shutting down. Before shutting the VM executor down, VB Studio saves the operating system and software packages to the VM executor's assigned boot volume.	Yes
Stopped	The VM executor has shut down.	No
Error	There's a hardware or a software issue on the VM executor. Check the VM executor's log to find more about the cause.	No
Destroying	The VM executor is being deleted.	No
Error Unrecoverable	This state is most likely caused when a customer changes the Compute account's OCI access so that control of the VM executor OCI resources are blocked. This state can also occur if the build executor is in an Error status and VB Studio is not able to remove all used OCI resources or if there is a temporary network glitch during the removal process. VB Studio tries once a day to clear OCI resources used by the VM executor in the Error Unrecoverable state. To manually clear these resources, you can also use the Try to Reset action.	No

Some key points to remember about VM executors:

- After creating a VB Studio instance, VB Studio creates a VM executor when you create your first project (assuming you don't already have one). The VM executor is associated with the **System Default OL7 for Visual Builder** executor template in the connected OCI account.
- When you add a VM executor manually, you must specify the executor template, choose an OCI region from the connected OCI account's subscribed regions, specify the OCI Compute VM's shape, and select the VCN (optional).
- Your OCI account may have some Compute instance limits set. When you add a VM executor, VB Studio looks into the specified OCI region's Availability Domains, finds available OCPUs with the specified shape, calculates the number of Compute instances, and displays the number of Compute VM instances you can add from your OCI account's set limit.
Here's an example of the VB Studio's Add VM Build Executors dialog box that displays the number of VM executors you can add with the VM.Standard.1.1 shape:

Add VM Build Executor

Quantity
3

Build Executor Template
System Default OL7 for Visual Builder

Region
us-ashburn-1

Shape
VM.Standard1.1 (OCPU: 1, Memory: 7 GB)

Volume Size (GB)
100

Reserved Public IP (Disabled because Quantity > 1)

☒ Keep updated with the latest OS image

VMs available for current settings: 111

VCN Selection

Choose Custom if your build VM needs access to an Oracle resource (like a database) that's either behind a private network or not available through the public internet. The Custom option lets you configure the VM to run in the same VCN as your private resources.

☒ Default

☐ Custom

Cancel

Add

- When you add a VM executor, it is added in the **Pending** state and doesn't cost you anything. You can add more VM executors than the number of available Compute VM instances.
Remember, VB Studio creates a Compute VM instance when a VM executor starts, not when you add it.
- You can add standard and legacy VM shapes with these series:
 - VM.Standard1
 - VM.Standard2
 - VM.Standard.E2
 - VM.Standard3.Flex
 - VM.Standard.E3.Flex
 - VM.Standard.E4.Flex
 - VM.Standard.E5.Flex
 - VM.Standard.B1
 - VM.Standard.Intel.Generic
 - VM.Standard.x86.Generic

- VM.Standard.AMD.Generic

For more details about the above shapes, see [Standard shapes](#) and [Legacy shapes](#).

- A VM executor can run one build at a time.
- When a job's build runs, if VB Studio finds multiple VM executors allocated for the job's executor template, it runs the build on any one of them. You can't choose or specify a particular VM executor to use for the build.
- If you expect your organization's members to run parallel builds of jobs that refer to a common executor template, add multiple VM executors for that executor template. If you're not sure, you can start with one VM executor and add more VM executors later.
- While adding multiple VM executors that reference a common executor template:
 - Add all VM executors in the same VCN. If you add VM executors with a common executor template in different VCNs (such as some VM executors in the default VCN and other VM executors in a custom VCN), your builds might behave unpredictably.
 - Add all VM executors with the same shape. If you add VM executors with different shapes (such as some VM executors of the VM.Standard1.1 shape and some of the VM.Standard2.8 shape), your builds may run slow or fast depending on the VM executor it runs on.
- After a build is complete, a VM executor continues to be in the **Available** state and waits for some time for any queued builds. This wait time is called sleep timeout. If no builds run on the VM executors in this duration, VB Studio automatically stops the VM executors.
- The more VM executors you have running at a specific time, the higher the cost. To minimize the higher cost, configure the sleep timeout to stop inactive VM executors after some time. The sleep timeout setting applies to all your organization's VM executors.
- If all VM executors are in an **Error** state for a given build executor template, the VM executor will attempt to rerun the build after successive delays of 2 minutes, 4 minutes, and 8 minutes. This is to allow enough time for the VM executors in the **Error** state to be deprovisioned, reprovisioned, and transition to the **Pending** state.

Build Executor Templates

A build executor template defines the operating system and the software packages your organization's members need to run builds on VM executors.

VB Studio offers some executor templates out-of-the-box, but you can create more templates if the default templates don't meet your requirements. See [Create and Manage Build Executor Templates](#).

Some key points to remember about executor templates:

- Each executor template contains an Oracle Linux operating system, Java, and some required software packages. If required, you can install more software packages available in the [VB Studio Software Catalog](#).
- You can't add a software package or a version that's not available in the VB Studio Software Catalog.
- In the Software Catalog, some software packages have multiple versions. You can add only one version to the executor template. However, you can add multiple versions of Java to an executor template.
- Executor templates don't consume storage space or increase your cost. You can create as many executor templates as you require.

- Some software packages, such as Fn and Oracle Jet, require other software packages in the same executor template. When you add these software packages, VB Studio prompts you to add the dependent software packages.
- Don't add unnecessary software packages to an executor template. The more software packages you add to an executor template, the more storage space its VM executors consume when they run builds.

What Happens When a Build Runs in a Build Executor?

When VB Studio runs a build, it follows a set order to select a VM executor. You can't choose or specify a particular VM executor to use for the build. If no VM build executors are found, VB Studio fails the build.

When multiple VM executors are found, VB Studio selects a VM executor to run the build in this order:

- If a VM executor is in the **Available** state, VB Studio runs a build on it.
- If no VM executors are in the **Available** state, VB Studio starts a **Stopped** VM executor, installs the operating system and the executor template's software packages from the saved boot volume, and then runs the build on it.
- If no VM executors are in the **Available** or the **Stopped** state, VB Studio starts a **Pending** VM executor, installs the operating system and the executor template's software packages, and then runs the build on it.
- If all VM executors are running builds, VB Studio waits for a VM executor to complete its build, and then runs a build on it.

Note

A VM executor in the **Stopped** or **Pending** state will take several minutes to start because VB Studio installs the operating system and the executor template's software packages before running the build.

The VM executor should start subsequent builds more quickly. You can adjust the sleep timeout to avoid start-up delays.

The steps for a build running in a build executor include:

1. VB Studio checks the job's build executor template and then finds a VM executor allocated to it.
2. VB Studio checks the job's configuration and runs the commands in the specified order.
3. After the build is complete, VB Studio copies any generated artifacts to the configured OCI Object Storage bucket.
4. The VM executor waits for some time for any queued builds. If no builds run during the wait time period, the VM executor stops.
Before stopping the VM executor, VB Studio saves the operating system and software packages to the VM executor's assigned boot volume.

Docker Build Executors and Docker Images

A *Docker image* defines the operating system and software packages your organization's members need to run builds on a *Docker executor*. You can either import a Docker image from an external Docker registry, such as DockerHub, or create it from a build executor template.

Unlike a VM executor, a Docker executor is not directly associated with any specific VM. When your organization's members create jobs, they simply associate a Docker image as a build template with the job. When the job's build triggers, VB Studio runs the build on any *Docker deployment VM*.

Some key points to remember about Docker images:

- When you create an image from a build executor template, VB Studio creates a new Docker image by installing the software configured in that template. When you create an image from a registry, VB Studio pulls the image from the registry, adds a build agent, and creates a new image.
- You can specify the maximum number of executors that can be created from the image.
- You can create as many Docker images as you require, as long as the Management VM has enough space.

Docker Deployment VMs and Docker Executors

Docker executors run in a Docker deployment VM. Like a VM build executor, a Docker deployment VM is an OCI VM Compute instance dedicated to run builds.

To run builds in Docker executors, you add Docker deployment VMs. A Docker executor can run only one build at a time, but a Docker deployment VM can run multiple builds simultaneously, depending on how many Docker executors are configured for the deployment VM. The maximum number of Docker executors that can be run by a given Docker deployment VM is configurable between 1 and 10.

When a job's build is triggered, VB Studio starts a Docker deployment VM if it is in a stopped state, creates a Docker executor in it, and builds the job using the executor.

Remember, a Docker deployment VM costs you only when it's in a running state and actively building jobs. When the Docker deployment VM is idle and not building jobs, it'll be put to sleep. You can configure the number of minutes the deployment VM remains idle before it is put to sleep.

This table describes the Docker deployment VM states:

State	What does it mean?	Does it cost?
Pending	After you add a Docker deployment VM, it is in this state until the management VM finishes building, and then it will start automatically.	No

State	What does it mean?	Does it cost?
Starting	<p>The Docker deployment VM is starting.</p> <p>If the Docker deployment VM starts from the Pending state, VB Studio installs the operating system and software packages on the Docker deployment VM's assigned boot volume. This takes time.</p> <p>If the Docker deployment VM starts from the Stopped state, VB Studio uses software packages and the operating system from the previous run's saved boot volume.</p>	Yes
Ready	<p>The operating system and Docker agent are installed and started.</p> <p>The VM is ready to deploy Docker executors.</p>	Yes
Stopping	<p>The Docker deployment VM is shutting down.</p> <p>Before shutting the Docker deployment VM down, VB Studio saves the operating system and software packages to the Docker deployment VM's assigned boot volume.</p>	Yes
Stopped	<p>The Docker deployment VM has shut down.</p>	No
Error	<p>There's a hardware or a software issue on the Docker deployment VM. Check the Docker deployment VM's log to find more about the cause.</p>	No

Some key points to remember about Docker deployment VMs and Docker executors:

- Once created, the Docker Management VM is always running and costs you 24-7. To keep your costs lower, if you are planning to run fewer builds with fewer images, select a smaller shape with fewer OCPUs.
- To run builds in Docker executors, you add Docker deployment VMs and specify the number of Docker executors it can run, select the OCI region from the connected OCI account's subscribed regions, the OCI Compute VM's shape, and optionally the VCN.
- Your OCI account may have some Compute instance limits set. When you add a Docker deployment VM, VB Studio looks into the specified OCI region's Availability Domains, finds available OCPUs with the specified shape, calculates the number of Compute instances, and displays the number of Compute VM instances you can add from your OCI account's set limit.
- When you add your first Docker deployment VM, VB Studio adds one more VM, called the Docker Management VM, to manage Docker images.
- The Docker Management VM is responsible for creating Docker images and Docker executors and for deploying images to Docker executors.

- When you create your first Docker deployment VM, the Docker Management VM is also created with the same shape.
- VB Studio provides a default shape selection for the management VM. This default is only a recommendation. Some things to consider:
 - You could still choose a lower-end configuration for your specific purposes, but the default is what VB Studio determines to be the optimal powered VM for the management VM. Although a low-end VM (such as Standard 1.1 with 1 CPU and 8 GB of memory) could be used for Docker Image management in scenarios that have few Executor templates and create just a few Docker images, it is better to use the default recommendation. If you try to select a lower-powered VM than what is recommended, you'll see a warning.
 - For scenarios that have several Executor templates and will create several Docker images, using a high-end VM (such as VM.Standard.E3.Flex with 4 OCPU and 32 GB RAM) instead is recommended. Using a low-end VM in demanding scenarios will severely hamper overall performance.
- If a Docker deployment VM is not in use, it will eventually time out and enter the Stopped state, which has no associated cost to you.
- The higher the number of Docker deployment VMs running Docker executors at any specific time, the higher the associated cost. To minimize the higher cost, you should configure the sleep timeout to stop inactive Docker deployment VMs after some period of time. The default timeout is 300 minutes. The sleep timeout setting will be applied to all your organization's Docker deployment VMs.

What Happens When a Build Runs on a Docker Executor?

1. VB Studio checks the Docker image that corresponds with the template configured in the job and finds whether a build of the image has run before.
If this is the Docker image's first build, it finds a Docker deployment VM to run the image in an executor, and then it downloads the image from the Docker Management VM if the image doesn't already exist in the VM.

If one or more Docker deployment VMs are found, VB Studio selects a Docker deployment VM to run the build in this order:
 - a. If a Docker deployment VM is in the **Ready** state and has at least one Docker executor that's not running, VB Studio runs the build on it. A Docker deployment VM is full if VB Studio has already deployed the maximum number of executors on that VM.
 - b. If no Docker deployment VMs are in the **Ready** state, VB Studio starts a **Stopped** Docker deployment VM and runs the build on one of its Docker executors.
 - c. If all Docker executors of all Docker deployment VMs are running builds, VB Studio waits for a Docker executor to complete its build, and then runs a build on it.
 If this is not the Docker image's first build, VB Studio finds the Docker deployment VM that ran the image's Docker executor earlier and uses it to run the build again, unless the Docker deployment VM is full. If all Docker executors of the Docker deployment VM are busy running builds, VB Studio selects another Docker deployment VM to run the build in the above described order.
2. VB Studio checks the job's configuration and runs the commands in the specified order.
3. After the build is complete, VB Studio copies any generated artifacts to the configured OCI Object Storage bucket.
4. The Docker deployment VM waits for some time for any queued builds. If no builds run during the wait time period, the Docker deployment VM stops.

Before stopping the Docker deployment VM, VB Studio saves the Docker image to the Docker deployment VM's assigned boot volume.

Step 3: Set up VB Studio to run VM build executors

Optional step: If you decided to go with VM build executors in Step 2, use this step.

If you decided to use Docker executors, go to [Step 4: Set up VB Studio to run builds on Docker executors](#). You must use one or the other.

To run VM build executors, you'll need to create build executor templates and then add VM executors.

Create and Manage Build Executor Templates

You can create and manage build executor templates from the Organization Administration page's **Build Executor Templates** tab.

Note


You can't create or manage executor templates if VB Studio is connected to the built-in free account. If you want to use software that isn't available in the default executor templates, configure VB Studio to connect to your OCI paid account. See [Change Your OCI Account](#).

VB Studio offers these executor templates out-of-the-box for your organization to use. You can't modify or delete these executor templates.





Build executor template	Description
System Default OL7 for Visual Builder	Use this template to package and deploy extensions and visual applications. The template defines these software packages on Oracle Linux 8: <ul style="list-style-type: none">• Node.js 22• Default software packages
System Default OL8	Use this template for build jobs that use the default software packages on Oracle Linux 8.
System Default OL7	Use this template for build jobs that use the default software packages on Oracle Linux 7.


If none of these meet your requirements, you can create as many executor templates as you need. Executor templates don't consume storage space or increase your cost.

You can create and manage executor templates from the **Build Executor Templates** tab on the Organization page:

1. In the left navigator, click **Organization** .
2. Click the **Build Executor Templates** tab.

This table describes the actions you can perform to create and manage executor templates.

Action	How To
Create a custom executor template	<ol style="list-style-type: none"> 1. Click + Create Template. 2. In the New Build Executor Template dialog box, enter a name and description of the executor template. In Platform, select the operating system to run on the VM executor. 3. Click Create. 4. Click Configure Software. 5. If necessary, in Filter, enter the search term and click Search . To see the latest version of the software package, select the Show latest versions only check box. This is helpful if multiple versions of the same software are available in the catalog. 6. In the software catalog, select the software package tile's check box to add it. The tile is highlighted in green and the software package adds to the Selected Software list on the right. To remove a selected tile, deselect the check box or click Remove this software  under Selected Software. You can't remove the required software packages that are highlighted in grey. If a software package depends on another software package or if there's a conflict between software packages, a message under Selected Software informs you about it. 7. Click Done.
Configure an executor template's software	<ol style="list-style-type: none"> 1. From the executor template list on the left, select the executor template and click Configure Software. 2. In the software catalog, to search a software package, enter the search term in Filter and click Search . To see the latest version of the software package, select the Show latest versions only check box. This is helpful if multiple versions of the same software are available in the catalog. To add a software, select the software package tile's check box. The tile is highlighted in green and the software package adds to the Selected Software list on the right. To remove a software package, deselect its check box or click Remove this software  under Selected Software. You can't remove the required software packages that are highlighted in grey. If a software package depends on another software package or if there's a conflict between software packages, a message under Selected Software informs you about it. 3. Click Done.
Edit an executor template's name or description	<ol style="list-style-type: none"> 1. From the executor template list on the left, select the template you want to modify and click the Edit button on the far right side of the panel. 2. In the Edit Build Executor Template dialog box, update the name or description. 3. Click Save.

Action	How To
Delete an executor template	<p>When you delete an executor template, its VM executors are deleted too.</p> <ol style="list-style-type: none">1. From the executor template list on the left, select the template you want to delete.2. Either click the Delete  icon next to it in the list or click the Delete button to the right in the template's panel.3. In the Delete Build Executor Template dialog box, click Yes to confirm.

Add and Manage VM Build Executors

A VM executor can run one build at a time. When you add a VM build executor, you allocate an OCI VM Compute instance to run VB Studio builds. If you expect your organization's members to run builds in parallel jobs that refer to a common executor template, you should add multiple VM executors for that executor template. Note that the more VM executors you have running at a specific time, the higher the cost. To minimize the higher cost, use the Sleep Timeout setting to automatically shut down inactive VM executors.

Note

You can't add or manage VM executors if VB Studio is connected to the built-in free account. If you want to add more than one VM executor, configure VB Studio to connect to your OCI paid account.

Here's an example of multiple VM executors that use a common executor template.

Visual Builder Studio

Organization

Projects OCI Account Build Executor Templates **Build Executors** Component Exchange Groups Properties Activities

105 VMs **Migrate to Docker** **Auto Refresh**

Sleep Timeout **Update Selected** **+ Create VM**

<input type="checkbox"/>	Id	Template	Region	Shape	Details	Status	Actions
<input type="checkbox"/>	82201	chisriva_Devsecops	us-ashburn-1	VM.Standard1.1		Stopped	
<input type="checkbox"/>	88221	DCS QA Docker 17 Linux 7	us-ashburn-1	VM.Standard1.1		Stopped	
<input type="checkbox"/>	88241	DCS QA Docker 17 Linux 7	us-ashburn-1	VM.Standard1.1		Stopped	
<input type="checkbox"/>	80884	DCS QA Docker 19 Linux 7	us-ashburn-1	VM.Standard1.1		Stopped	
<input type="checkbox"/>	83782	DCS QA Docker 19 Linux 7	us-ashburn-1	VM.Standard1.1		Pending	

You add and manage VM executors from the **Build Executors** tab on the **Organization** page:

1. In the left navigator, click **Organization** .
2. Click the **Build Executors** tab.

This table describes the actions you can perform to manage VM executors.

Action	How To
Add the free VM build executor in the built-in free account	<p>You can add and use the free VM build executor only if your VB Studio instance is connected to the built-in free account.</p> <p>The free VM build executor will be automatically created upon creation of the first project in the org. You can also add it by clicking Create Free VM Build Executor on the Build Executors tab.</p> <p>VB Studio creates the free VM build executor that uses the System Default OL7 for Visual Builder template. You can't change the VM executor's template.</p> <p>The new VM executor is in the Pending state until you manually start it or trigger a job's build that references the associated executor template.</p>
Find VM executors of an executor template	In the search box, enter the executor template's name.
Sort VM executors	Click the arrow icon in the column's header to sort VM executors. For example, to sort VM executors by their state, click the arrow icon in the Status column's header.

Action	How To
Add a VM executor in the VB Studio's default VCN	<ol style="list-style-type: none"> 1. Find VM executors of the executor template you want to use. If there are no VM executors of the executor template, jump to the next step. If you find one or more VM executors that use the executor template, get each VM executor's VCN name. If the VCN's name isn't <code>vbs-executor-vcn</code>, click *** and select Delete to delete the VM executor. You should not add VM executors to different VCNs that use the same executor template. 2. Click + Create VM. 3. In the Add VM Build Executor dialog box, in Quantity, specify the number of VM executors you want to create. To minimize build execution delays, set the number to the number of jobs that you expect to run in parallel using the template you'll specify in Build Executor Template. If you're not sure, start with one VM executor. You can add more VM executors later, based on your actual usage. 4. In Build Executor Template, select the executor template. 5. In Region, select the region that's closer to you geographically. The drop-down list displays regions your OCI account is subscribed to. 6. In Shape, select the VM executor's shape. For example:

Add VM Build Executor

Quantity: 3

Build Executor Template: System Default OL7 for Visual Builder

Region: us-ashburn-1

Shape: VM.Standard1.1 (OCPU: 1, Memory: 7 GB)

Volume Size (GB): 100

Reserved Public IP (Disabled because Quantity > 1)

☒ Keep updated with the latest OS image

VMs available for current settings: 111 ?

VCN Selection
Choose Custom if your build VM needs access to an Oracle resource (like a database) that's either behind a private network or not available through the public internet. The Custom option lets you configure the VM to run in the same VCN as your private resources.

☒ Default
☐ Custom

Cancel Add

Click the question mark icon to see how the number of available VMs was calculated for the current settings. Here's an example of what you might see:

Action**How To**

VMs available for current settings: 100 ⓘ

Availability Domain	Max by OCPUs	Max by Volume	Max by Memory	Max VMs this AD
ZQib:US-ASHBURN-AD-1	25	31	N/A	25
ZQib:US-ASHBURN-AD-2	41	159	N/A	41
ZQib:US-ASHBURN-AD-3	34	284	N/A	34
Total				100

The display shows the realtime statistics that were used to make this calculation, using your settings. As you change your settings, new statistics are computed and displayed. See [What Determines How Many VMs I Can Create?](#) for more information about how the calculations are made.

If the required number of Compute VM instances aren't available, choose another shape.

7. Enter the **Volume Size** (between 50 GB and 500 GB) you want to use for *each* build this VM executor builds. The default is 100 MB. Larger volumes give more space per VM. While this is great for heavy workloads, the drawback is you won't be able to create as many VMs.
8. Optionally, select **Keep updated with the latest OS image**. If you select this option, VB Studio checks nightly for new OS versions from OCI (usually, OCI releases about one new OS version each month) and automatically updates any affected VMs. This keeps your VM executors up to date without waiting for the quarterly updates.

Note

If a build is already running when the update is to start, VB Studio lets that build finish first, then starts the update. If a build is scheduled to start during the update, VB Studio runs the update first and then starts the build right after.

Note that these updates might temporarily delay any scheduled builds, but they'll start as soon as the update is complete.

9. Select the **Reserved Public IP** that you want to add to the VM executor. You can add one Reserved Public IP to each VM executor. An allow listing for your VM executor is possible only if you specify an OCI account by assigning a Reserved Public IP to a build executor. The Reserved Public IP must be defined in the compartment of the specified OCI account.
10. In **VCN Selection**, select **Default**, and then click **Add**.

The new VM executor is in the Pending state until you manually start it or trigger a job's build that references the associated executor template.

Action	How To
Add a VM executor in another compartment's VCN	<p>Before you add a VM executor, make sure you've added a public subnet in the VCN. See Create and Configure a Public Subnet in a VCN.</p> <ol style="list-style-type: none"> 1. Find VM executors of the executor template you want to use. If there are no VM executors of the executor template, jump to the next step. If you find one or more VM executors that use the executor template, get each VM executor's VCN name. If the VCN's name is different from your VCN, click *** and select Delete to delete the VM executor. You should not add VM executors to different VCNs that use the same executor template. 2. Click + Create VM. 3. In the Add VM Build Executor dialog box, in Quantity, specify the number of VM executors you want to allocate. 4. In Build Executor Template, select the executor template. 5. In Region, specify the VM executor's region. 6. In Shape, select the VM executor's shape. For example:

Add VM Build Executor

Quantity: 3

Build Executor Template: System Default OL7 for Visual Builder

Region: us-ashburn-1

Shape: VM.Standard1.1 (OCPUs: 1, Memory: 7 GB)

Volume Size (GB): 100

Reserved Public IP (Disabled because Quantity > 1)

☒ Keep updated with the latest OS image

VMs available for current settings: 111 ?

VCN Selection
Choose Custom if your build VM needs access to an Oracle resource (like a database) that's either behind a private network or not available through the public internet. The Custom option lets you configure the VM to run in the same VCN as your private resources.

☒ Default
☐ Custom

Cancel Add

Notice the message below the **Reserved Public IP (Optional)** field. Click the question mark icon (?) to see how the number of available VMs was calculated for the current settings. Here's an example of what you might see:

Action**How To**

VMs available for current settings: 100 ⓘ

Availability Domain	Max by OCPUs	Max by Volume	Max by Memory	Max VMs this AD
ZQib:US-ASHBURN-AD-1	25	31	N/A	25
ZQib:US-ASHBURN-AD-2	41	159	N/A	41
ZQib:US-ASHBURN-AD-3	34	284	N/A	34
Total				100

The display shows the realtime statistics that were used to make this calculation, using your settings. As you change your settings, new statistics are computed and displayed. See [What Determines How Many VMs I Can Create?](#) for more information about how the calculations are made.


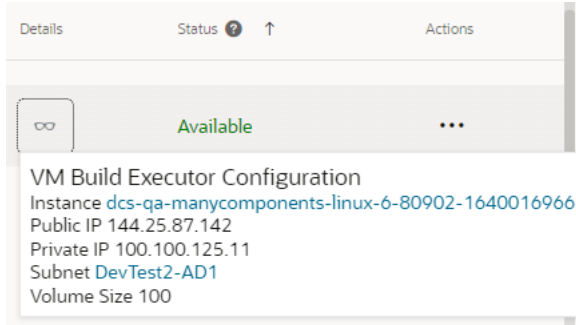
If the required number of Compute VM instances aren't available, choose another shape.

7. Enter the **Volume Size** (between 50 GB and 500 GB) you want to use for *each* build this VM executor builds. The default is 100 MB. Larger volumes give more space per VM. While this is great for heavy workloads, the drawback is you won't be able to create as many VMs.
8. Optionally, select **Keep updated with the latest OS image**. If you select this option, VB Studio checks nightly for new OS versions from OCI (usually, OCI releases about one new OS version each month) and automatically updates any affected VMs. This keeps your VM executors up to date without waiting for the quarterly updates.

Note

If a build is already running when the update is to start, VB Studio lets that build finish first, then starts the update. If a build is scheduled to start during the update, VB Studio runs the update first and then starts the build right after.

9. Select the **Reserved Public IP** that you want to add to the VM executor. You can add one Reserved Public IP to each VM executor. An allow listing for your VM executor is possible only if you specify an OCI account by assigning a Reserved Public IP to a build executor. The Reserved Public IP must be defined in the compartment of the specified OCI account.
10. In **VCN Selection**, select **Custom**.
11. In **VCN Compartment**, select the compartment. If you're an Oracle Cloud OS Management Service (OSMS) user, don't select the OSMS compartment or a compartment with an OSMS policy.
12. In **VCN**, select the VCN.
13. In **Subnets Compartments**, select the compartments where your public subnets are. By default, it adds your VCN's compartment. If required, you can add more compartments.
14. In **Subnets**, select a subnet. The list shows public subnets only. You can add multiple public subnets. If VB Studio can't create a VM executor on the first subnet you've added, it tries to create it on the second subnet, and so on.
15. Click **Validate Network Setup**.

Action	How To
	<p>16. After successful validation, click Add.</p> <p>The new VM executor is in the Pending state until you manually start it or trigger a job's build that references the associated executor template.</p>
Get the name and IP addresses of a VM executor's VCN subnet.	<ol style="list-style-type: none"> In the VM executor's Details column, click Show machine details . Note the VCN subnet's name.
	 <p>The VM executor's subnet name is displayed only if it's in the Available state. If the VM executor is in the Stopped or the Pending state, click *** and select Start to start the VM executor.</p>
View a VM executor's log	<p>The VM executor's log has entries for all events along with information about when the events occurred, the type of event, and event details.</p> <ol style="list-style-type: none"> Select the VM executor, click Actions *** and select Display Log. In the Provisioning Log window, review the log. To download the log file to your computer, click Download Log.
Start or stop a VM executor manually	<p>When a build of a job triggers, its VM executor starts automatically if it was in the stopped state. It takes some time to start a VM executor, and the user must wait for a VM executor to start before the job's build runs on it. Similarly, a VM executor stops automatically if no builds run on it during the sleep timeout period.</p> <p>At times, you may want to manually start a VM executor before triggering a job's build or stop it to free resources immediately. To start or stop a VM executor, click Actions *** and select Start or Stop.</p> <p>To start or stop multiple VM executors, select their check boxes, click Update Selected and select Start selected VMs or Stop selected VMs.</p>
Delete a VM executor	<p>To delete a VM executor, click Actions *** and select Delete.</p> <p>If the Delete action doesn't delete the VM executor, you can force a delete using the Force Delete action.</p> <p>To delete multiple VM executors, select their check boxes, click Update Selected and select Delete selected VMs.</p>

Action	How To
Change the sleep timeout of all VM executors	<p>Set the sleepout time to stop inactive VM executors automatically. By default, it is 30 minutes.</p> <p>The higher the sleep timeout value, higher will be your cost as inactive VM executors continue to be in the Available state for a longer time. If you don't expect your organization's users to run builds frequently, specify a lower sleep timeout.</p> <p>When a VM executor starts from the Stopped state, it installs the operating system and all software packages of the executor template. This takes time and your organization's users must wait for the build to start until the VM executor is in the Available state. If you expect your organization's users to run builds frequently, specify a higher sleep timeout value.</p> <ol style="list-style-type: none"> 1. Click Sleep Timeout. 2. In the Sleep Timeout dialog box, change the timeout duration to define how long should a VM executor be in the Available state if no builds run on it. 3. Click Save.
Reset the VM executor	<p>Use the Try to Reset action if the build executor is stuck in the Error Recoverable status. The Try to Reset action will try to remove all used OCI resources of the VM executor.</p> <p>To reset a VM executor, click Actions *** and select Try to reset.</p>

After adding VM executors, you and your organization's members can configure jobs to use executor templates.

What Determines How Many VMs I Can Create?

There are three factors that determine how many VMs you can create:

- **OCPU** - The maximum number of VMs varies according to the shape selected
- **Memory** - The maximum number of VMs is based on the Flex shape selected and the Allocated Memory setting. (This restriction is used for Flex shapes only.)
- **Volume** - The maximum number of VMs is based on the Volume Size setting.

After you specify a volume size (in GB), the **Add VM Build Executor** dialog displays real time numbers for OCPU and Memory, just below the line that tells you how many VMs are available for you to use. These numbers come directly from the OCI Console.


You might notice that **Quantity** isn't on the list. The quantity of VMs that you actually create has no impact on the limits. The limits represent the maximum numbers of what you could *use* but VB Studio doesn't restrict what you can *create*. This is actually beneficial, since availability constantly changes. Instead, the runtime simply limits what you can use in real time.

Connect to Your OCI Account and Add VM Build Executors

Configure VB Studio to connect to your own OCI paid account if you need more VM build executors to reduce the wait time for your organization's members, if you want to create custom build executor templates, or if you want to use advanced features for VM executors (such as use your own VCN or use a different VM shape).

1. Configure your VB Studio instance to connect to your OCI paid account.

If you don't have authorization to create and manage OCI resources, ask someone who can create the resources.

- a. If you haven't created a VB Studio instance yet, you can create the instance and connect it to your OCI account at the same time. See [Create a VB Studio Instance in Your Oracle Cloud Account](#).
- b. If your VB Studio instance already exists and you now want to connect it to your own OCI account:
 - i. In the Oracle Cloud Console, in the upper-left corner, click **Navigation Menu** .
 - ii. Select **Developer Services** and then select **Visual Builder Studio**.
 - iii. On the Visual Builder Studio instances page, click the VB Studio instance's name.
 - iv. On the instance details page, click **Actions**, then click **Edit**.
 - v. Select the **Yes, I authorize this** checkbox.

Edit Visual Builder Studio

General information

Compartment
[redacted] (root)

Name
VBS-dev-instance


CI/CD setup

Visual Builder Studio requires authorization to create Compute instances and Storage buckets in your Oracle Cloud account. We use the Compute instances to execute your CI/CD pipelines, and the Storage buckets to store build artifacts, as well as to enable the binary repository features in your projects (Maven, NPM, and Git LFS). If you have a pay-as-you-go Cloud account, you will incur charges only when you execute a pipeline or consume storage, and only after your overall usage exceeds the Oracle Cloud Free Tier limits. Do you authorize Oracle to use these resources?

☒ Yes, I authorize this

Compartment
DemoCompartment

Select the compartment where you want us to create the Compute and Storage instances

- vi. Select the compartment to host your VB Studio resources.
 - vii. Click **Save changes**.
2. To see your new OCI connection in VB Studio:
 - a. In the left navigator, click **Organization** .
 - b. Click the **OCI Account** tab.

To create custom build executor templates, see [Create and Manage Build Executor Templates](#). Remember to add Node.js 20 or 22 to the executor template. Node.js 20 and 22 are the supported versions for packaging extensions.

To add more VM executors:

1. Click the **VM Build Executors** tab.
2. Click **+ Create VM**.
3. In the Add VM Build Executor dialog box, in **Quantity**, specify the number of VM executors you want to allocate.

To minimize build execution delays, set the number of VM executors to the number of jobs that you expect to run in parallel using that template. If the OCI VM quota is available, that number of VM executors will be added to the **Virtual Machines** tab.

If you're not sure about the number of VM executors you'll need, start with one VM executor and then add more as required. Note that the more VM executors you have running at a specific time, the higher the cost of OCPUs. To minimize the higher cost, use the Sleep Timeout setting on the **VM Build Executors** tab to automatically shut down inactive VM executors. You can always return to the **VM Build Executors** tab to remove or add VM executors, based on your actual usage.

4. In **Build Executor Template**, select **System Default OL7 for Visual Builder** or an executor template with Node.js 20 or 22 software.
5. In **Region**, **Shape**, and **VCN Selection** specify the VM's region, shape, and Virtual Cloud Network (VCN).
 - A region is a localized geographic area where the data centers are located. Remember, a VM executor is a VM on OCI Compute. Choose the region where your VB Studio account is or the one that's closest to you geographically.
 - A shape is a template that determines the number of CPUs, amount of memory, and other resources allocated to the created instance. Choose a shape of your preference.
 - A VCN is a software-defined network that you set up in the Oracle Cloud Infrastructure data centers in a particular region. By default, builds run in your VB Studio's compartment. To run builds in your own VCN, select **Custom** and specify its details.

To learn more about regions and shapes, see [Regions and Availability Domains](#) and [Compute Shapes](#). To find more about VCNs, see [VCNs and Subnets](#).

6. Click **Add**.

Run VM Build Executors in Virtual Cloud Network

A Virtual Cloud Network (VCN) is a software-defined network that you set up in the OCI data centers of a region. A subnet is a subdivision of a VCN. A VM build executor always runs in a VCN's public subnet.

You can choose to run a VM executor in the VB Studio compartment's VCN or in another compartment's VCN. Run the VM executor in the VB Studio compartment's VCN if you don't have a VCN or want to use the default option without any additional configuration. Run the VM executor in another compartment's VCN if you want the VM executor to access Oracle Cloud services running in that compartment's VCN.

Note

You can't run VM executors in a VCN if VB Studio is connected to the built-in free account. To use another compartment's VCN, configure VB Studio to connect to an OCI paid account.

Use VB Studio's Default VCN

VB Studio compartment's VCN, also called the default VCN, is automatically created for you when the first VM build executor that uses it starts.

VB Studio's default VCN is called `vbs-executor-vcn` and resides in the VB Studio's compartment. When a VM executor that uses the default VCN starts, VB Studio checks the OCI compartment for the default VCN. If it doesn't exist, VB Studio creates a VCN called `vbs-executor-vcn` with CIDR block `10.0.0.0/16` and public subnets in all availability domains. If the VCN exists, VB Studio uses it to run your VM executors.

When VB Studio creates the default VCN, it also creates these components and adds them to the VCN:

- An Internet Gateway
- A Route Table that uses the Internet Gateway as the routing rule
- Security Rule Ingress rules that allow TCP traffic on:
 - Destination port 22 (SSH), 9003 (Executor agent debug), 9005 (VM agent debug), 9082 (Executor agent), and 9085 (VM agent), and 8095 (Docker Agent), and 9001-9010 from source 0.0.0.0/0 and any source port.
 - Destination port 443 from sources in the 10.0.0.0/16 IP range.
- A Security Rule that allows Egress to any destination from any protocol.
- Three public subnets, one for each availability domain. Their CIDR is set to 10.0.0.0/24, 10.0.1.0/24, and 10.0.2.0/24.

Here's an example of the default VCN:

The screenshot shows the 'Virtual Cloud Networks' page in the Oracle Cloud console. The selected VCN is 'vbs-executor-vcn', which is in an 'Available' state. The 'VCN Information' section provides the following details:

Property	Value
Compartment	BuildExecutorCompartment
Created	Feb 6, 2020, 19:07 UTC
IPv4 CIDR Block	10.0.0.0/16
IPv6 Prefix	—
OCID	ocid1.vcn.oc1... Copy
DNS Resolver	...-vcn
Default Route Table	Default Route Table for vbs-executor-vcn
DNS Domain Name	DNS isn't enabled for this VCN

Here's an example of the public subnets that are created:

The screenshot shows the 'Subnets' page for the 'vbs-executor-vcn' VCN. It displays a list of three public subnets, each associated with a different availability domain (AD-1, AD-2, AD-3). The subnets are all in an 'Available' state.

Name	State	IPv4 CIDR Block	IPv6 Prefixes	Subnet Access	Created
vbs-executor-subnet-vliq:US-ASHBURN-AD-3	Available	10.0.2.0/24	—	Public (vliq:US-ASHBURN-AD-3)	Feb 6, 2020, 19:07 UTC
vbs-executor-subnet-vliq:US-ASHBURN-AD-2	Available	10.0.1.0/24	—	Public (vliq:US-ASHBURN-AD-2)	Feb 6, 2020, 19:07 UTC
vbs-executor-subnet-vliq:US-ASHBURN-AD-1	Available	10.0.0.0/24	—	Public (vliq:US-ASHBURN-AD-1)	Feb 6, 2020, 19:07 UTC

As soon as the default VCN is available, you have full control over it and can modify it. You can add private subnets for your private services, add more public subnets or delete the existing subnets, modify security lists, and add or remove other components.

Note

- When a VM executor runs on the default VCN, it runs on any of its available public subnets. You can't specify which subnet it should run on.
- If you plan to remove some public subnets of the default VCN, make sure that at least one public subnet is available in the VCN. If there are no public subnets, VM executors in the default VCN won't run and your builds will fail.
- The default VCN is created once and continues to stay until it is deleted manually.
- If your organization's members configure jobs that access Oracle Cloud services in the private or public subnets of the VCN, ask them to configure their jobs to access the services using private IPs or Fully Qualified Domain Name (FQDN).

Run VM Build Executors in Another Compartment's VCN and Subnets

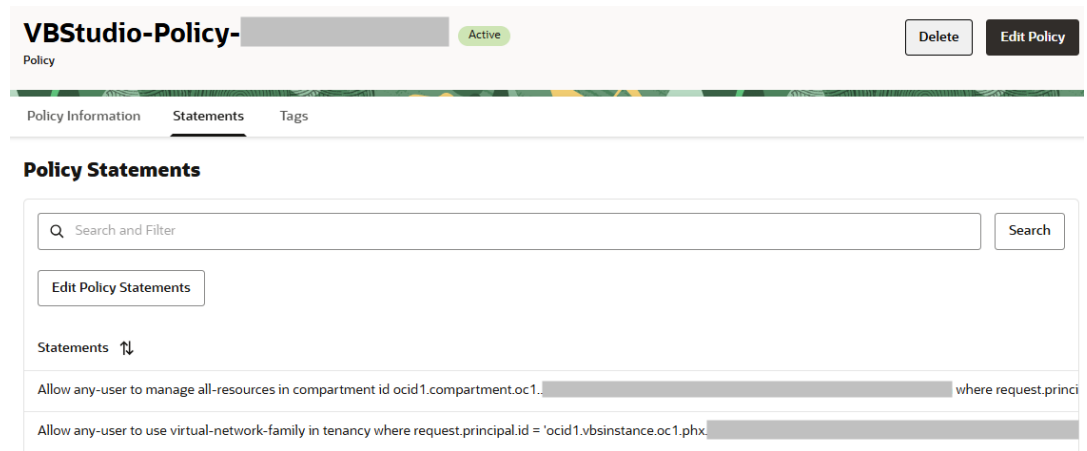
To allow a VM build executor access your Oracle Cloud services in a compartment's VCN, you should configure the VM executors to run in the same VCN. This allows the VM executor to access Oracle Cloud services easily without any complex networking configuration.

Before you configure the VCN, make a note of these:

- A VM executor always runs in a public subnet.
- In the VCN, you must create a public subnet or configure an existing public subnet to allow inbound access from and outbound access to VB Studio. See [Create and Configure a Public Subnet in a VCN](#).
- Make sure that the public subnet is regional.
- Instead of modifying an existing security list's security rules, create a new security list for the public subnet.
For the public subnet, create a security list and add ingress rules from source CIDR 0.0.0.0/0 for VB Studio ports 22 (SSH), 9082 (Executor Agent), and 9085 (VM Agent). This is required to allow VB Studio access the VM executors in the VCN.
- For the subnet's compartment, assign the `use virtual-network-family` OCI policy to the user whose OCID you specified when you set up the OCI connection in VB Studio. This is required for networking permissions and builds to run in the VCN's subnet. This statement assigns the policy to the user's group:

```
allow group <group-name> to use virtual-network-family in compartment <subnet-compartment-name>
```

Here's an example of the `use virtual-network-family` policy:



- Make sure that the VCN has a route table with a rule that allows Internet access.
- To allow the VM executor to access the VCN's private subnet's services and resources, configure the private subnet's security rules to allow incoming traffic from the public subnet used by the VM executor.
- While adding a VM executor, you can specify multiple public subnets. If VB Studio can't create the VM executor on the first specified public subnet, it tries to create it in the second subnet, and so on.
- After configuring a VM executor to run in another compartment's VCN, ask your organization's members to configure their build jobs to use the private IP addresses or the Fully Qualified Domain Name (FQDN) of services that are running in the VCN. Tell them not to use public IP addresses, because when VM executors are in the same VCN as the service, public IP addresses will route the traffic outside the VCN, causing builds to fail.


This table describes what you need to do if you have a VCN.

If ...	Then :
You have a VCN without a public subnet	<ol style="list-style-type: none"> 1. Create and Configure a Public Subnet in a VCN. You'll also configure the subnet's security list to allow inbound access from and outbound access to VB Studio. 2. To allow a VM executor access a service running in the private subnet, configure the private subnet's security list. See Allow VM Build Executors to Access a Private Subnet's Resources. 3. Add and Manage VM Build Executors in the VCN.

If ...	Then :
You have a VCN with a public subnet	<ol style="list-style-type: none"> 1. In your VCN, create a security list with ingress rules and egress rules as described in steps 7-12 in Create and Configure a Public Subnet in a VCN. 2. Open your public subnet's details page and add the security list. See step 15 in Create and Configure a Public Subnet in a VCN. 3. To allow a VM executor access to services running in the private subnet, configure the private subnet's security list. See Allow VM Build Executors to Access a Private Subnet's Resources. 4. Add and Manage VM Build Executors in the VCN.
You don't have a VCN and want to create one	<ol style="list-style-type: none"> 1. Use the VCN wizard to create a VCN with a public subnet and an internet gateway. See Virtual Networking Quickstart. 2. Create and Configure a Public Subnet in a VCN. You'll also configure the subnet's security list to allow inbound access from and outbound access to VB Studio. 3. To allow a VM executor access to services running in the private subnet, configure the private subnet's security list. See Allow VM Build Executors to Access a Private Subnet's Resources. 4. Add and Manage VM Build Executors in the VCN.

Create and Configure a Public Subnet in a VCN

Before you can run VMs in another compartment's VCN, you must first create a public subnet in your VCN with security rules that allow inbound access from and outbound access to VB Studio.

1. Sign in to Oracle Cloud Console.
2. In the upper-left corner, click **Navigation Menu** .
3. Select **Networking** and then select **Virtual cloud networks**.
4. On the Virtual Cloud Networks page, next to **Applied filters**, select the compartment.
5. From the VCNs list, click the VCN's name.
6. On the VCN details page, click the **Security** tab.
7. Under **Security Lists**, click **Create Security List**.
8. On the Create Security List page, in **Name**, enter the security list name.
9. In **Create in Compartment**, ensure that the correct compartment is selected.
10. In **Allow Rules for Ingress**, click **+ Another Ingress Rule** and follow these steps:

- a. In **Source Type**, select **CIDR**.
- b. In **Source CIDR**, enter `0.0.0.0/0`.
- c. In **Destination Port Range**, enter `9082`.
- d. (Optional) In **Description**, add a description.

Here's an example:

Create Security List

Name
Public Subnet Security List for VB Studio Build VMs

Create In Compartment
VBStudioCompartment

Allow Rules for Ingress

Ingress Rule 1 ×

Allows TCP traffic for ports: all

Stateless

☐

To enable bidirectional traffic flow, make sure a complementary rule in the opposite direction exists. Learn about [stateful and stateless rules](#).

Source Type
CIDR

Source CIDR
0.0.0.0/0

IP Protocol
TCP

Source Port Range
All

Destination Port Range
9082

Description
TCP Traffic for ports: 9082

Cancel Create Security List

- e. Click **+ Another Ingress Rule** and repeat steps 10a through 10d to add ports **9085** and **22**.
- f. (Only if you are using Docker executors) Click **+ Another Ingress Rule** and repeat steps 10a through 10d to add ports **8095** and **9001-9010** to `0.0.0.0/0` and add port **443** to your VCN CIDR (for example `10.0.0.0/16`).
11. In **Allow Rules for Egress**, click **+ Another Egress Rule** and follow these steps:
 - a. In **Destination Type**, select **CIDR**.
 - b. In **Destination CIDR**, enter `0.0.0.0/0`.
 - c. In **IP Protocol**, select **All Protocols**.
 - d. (Optional) In **Description**, add a description.
12. Click **Create Security List**.

After creating the security list, click its name and then click the **Security rules** tab to verify the ingress and egress rules you added.

Here's an example of ingress rules:

Default Security List for vbs-executor-vcn Available Actions

Security List

Instance traffic is controlled by firewall rules on each Instance in addition to this Security List

Details **Security rules** Tags

Ingress Rules

Q Search and Filter Search

Add Ingress Rules Actions ⌵

<input type="checkbox"/>	Stateless	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	22		TCP traffic for ports: 22 SSH Remote Login Protocol	...
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	9082		TCP traffic for ports: 9082	...
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	9085		TCP traffic for ports: 9085	...

Here's an example of the egress rule:

Egress Rules

Q Search and Filter Search

Add Egress Rules Actions ⌵

<input type="checkbox"/>	Stateless	Destination	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	Description
<input type="checkbox"/>	No	0.0.0.0/0	All Protocols				All traffic for all ports	...

13. Return to the VCN's details page.

14. Select the **Subnets** tab and follow these steps to create a public subnet:

If you want to edit an existing public subnet, jump to the next step.

a. Click **Create Subnet**.

b. In **Name**, enter the subnet's name.

c. In **Create in Compartment**, select the correct compartment.

d. In **Subnet Type**, make sure that **Regional** is selected.

e. In **IPv4 CIDR Block**, enter the subnet's CIDR block.

Don't set it to 172.17.0.0/16 as it's the default subnet allocated to Docker.

f. Under IPv6 Prefixes, in **Route Table**, select the VCN's route table.

g. In **Subnet Access**, make sure that **Public Subnet** is selected.

h. In **Dhcp Options**, select the VCN's DHCP options.

i. In **Security List**, select the security list you created starting with step 6.

j. Fill in the other fields as required.

k. Click **Create Subnet**.

15. If you want to edit an existing subnet, select the **Subnets** tab and then follow these steps:

a. Click the public subnet's name.

b. Click the **Security** tab and then click **Add Security List**.


c. On the Add Security List page, in **Security List**, select the security list you created starting with step 6.

d. Click **Add Security List**.

That's it. After creating or editing the public subnet, your VM executors can now run in the VCN.

Allow VM Build Executors to Access a Private Subnet's Resources

After adding a public subnet in a VCN, to allow VM build executors access to the resources and services (such as a VM-based Database) running in the VCN's private subnet, configure the private subnet's security rules to allow incoming traffic from the public subnet used by VM executors.

1. Sign in to Oracle Cloud Console.
2. In the upper-left corner, click **Navigation Menu** .
3. Select **Networking** and select **Virtual cloud networks**.
4. On the Virtual Cloud Networks page, click the VCN.
5. On the VCN details page, click the **Security** tab.
6. Under **Security Lists**, click the private subnet's security list.
7. Click the **Security rules** tab and then click **Add Ingress Rules**.
If you want to modify an existing rule, click the Actions icon (three dots), and then select **Edit**.
8. On the Add Ingress Rule page, in **Source Type**, select **CIDR**.
9. In **Source CIDR**, enter the VM executor's public subnet's CIDR range.
10. In **Destination Port Range**, enter the service's port number.
11. (Optional) In **Description**, add a description.
12. Click **Add Ingress Rules**.
13. If required, repeat steps 7-12 for each service's port.

VB Studio Public IP Addresses in OCI Data Centers

If you're running VM executors in another compartment's subnet and don't want to use the CIDR 0.0.0.0/0 range in your ingress and egress rules, specify VB Studio's public IP address in the subnet's ingress and egress rules.

For a list of public IP addresses, see Public IP Addresses for Oracle Visual Builder Studio.

You can also use the following list of IP addresses to restrict outbound access from your build VM executors:

Server type	IP address
Storage	129.150.1.1/32
	129.150.1.2/32
	134.70.80.3/32
	134.70.84.3/32
	130.162.1.2/32
	130.162.1.1/32
Public yum	23.212.73.81/32
Shared services yum	130.35.147.212/32

Server type	IP address
Yum regional mirrors	192.29.194.161/32
	129.149.81.204/32
	129.149.51.216/32
	192.29.166.112/32
	155.248.131.128/32
	129.149.16.241/32
	129.148.209.47/32
	138.1.66.178/32
	192.29.138.86/32
	129.148.132.80/32
	147.154.1.235/32
	192.29.25.15/32
	192.29.113.113/32
	129.149.65.253/32
	192.29.242.73/32
	138.1.16.168/32
	129.149.114.199/32
	192.29.222.78/32
	129.149.98.107/32
	129.149.122.166/32
	192.29.38.186/32
	147.154.123.93/32
	129.149.39.7/32
	129.148.179.162/32
	129.148.160.57/32
	192.29.153.28/32
	129.149.3.154/32
	129.148.144.226/32
	192.29.82.26/32
	192.29.69.154/32
	192.29.181.241/32
NPM (registry.npmjs.org)	104.16.16.35/32
	104.16.17.35/32
	104.16.18.35/32
	104.16.19.35/32
	104.16.20.35/32
	104.16.21.35/32
	104.16.22.35/32
	104.16.23.35/32
	104.16.24.35/32
	104.16.25.35/32
	104.16.26.35/32
	104.16.27.35/32
Node.js (nodejs.org)	104.20.22.46/32
	104.20.23.46/32

Server type	IP address
Docker - OL8 only (download.docker.com)	108.156.91.110/32
	108.156.91.29/32
	108.156.91.41/32
	108.156.91.7/32

Software for Build Executor Templates

VB Studio provides various software packages in the build executors' software catalog. Some software packages are available by default in each executor template.

Platforms

These platforms are available:

- Oracle Linux 8
- Oracle Linux 7 (default)

Default Software Packages

These software packages are available by default in each executor template. You can't edit or remove these software packages from an executor template.

Software	Version in Oracle Linux 8	Version in Oracle Linux 7
Oracle Java SE	17.x	17.x
Ant	1.10.5 or later	1.10.15
C++ Compiler (cpp/gcc)	8.5.0 or later	4.8.5 or later
Git	2.43.7 or later	2.31.1
Git LFS	3.4.1	2.10.0
Jq	1.6 or later	1.5 or later
Maven	3.9.9	3.9.9
Python2 (including Virtualenv)	2.7.18 or later	2.7.5 or later

Software Packages in the Software Catalog

Here's a list of the software available in the VB Studio's software catalog. You can add only one version to an executor template, even if multiple versions of the software are available.

Software	Version in Oracle Linux 8	Version in Oracle Linux 7	Notes
Docker	26.1.1 or later	19.03.11 or later	Note: Docker 19.x on OL7 has been deprecated and will be removed in a future release.
Firefox	ESR 140.5.0 or later	ESR 128.14.0 or later	
Fn	NA	0.6.47	Requires Docker, OCIcli, and Python3 on 3.7 or later.
GraalVM EE for Java 17	23.0.10	23.0.10	For Java 17.0.17, version 23.0.10 is available.
Gradle	8.9	8.9	

Software	Version in Oracle Linux 8	Version in Oracle Linux 7	Notes
Groovy	4.0.26	4.0.26	
Helm	3.9.3	3.9.3	
Java SE	23.0.2	23.0.2	
	21.0.9	21.0.9	
	17.0.17	17.0.17	
	11.0.29	11.0.29	
	1.8.0_471	1.8.0_471	

Note

Java 17.x is automatically included in all new Build Executor templates (instead of Java 1.8.x) and is automatically added to any existing templates.

Templates created before VB Studio 22.10.0 include Java 1.8.x, but they've been updated to also include Java 17.x.

Users can add multiple versions of Java to a build executor template by selecting the Java version to use in a job from the job's configuration page.

JUnit 4	4.12 or later	4.11 or later	
Kubectl	1.24.3	1.24.3	
Node.js grunt <i>pre-install</i>	grunt v1.6.1 or later gruntcli v1.5.0 or later	grunt v1.6.1 or later gruntcli v1.5.0 or later	
Node.js	22.6.0		Node.js 22 requires NPM v10.8.2 or later
	20.17.0	16.17.0	Node.js 20 requires NPM v10.8.2 or later
	18.18.1	14.20.0	Node.js 18 requires NPM v9.8.1 or later
	16.17.0		Node.js 16 requires NPM v8.15.0 or later
	14.20.0		Node.js 14 requires NPM v6.14.17 or later
Node.js Driver for Oracle Database	6.10.0 or later	6.10.0 or later	

Software	Version in Oracle Linux 8	Version in Oracle Linux 7	Notes
Oracle Analytics Client (OAC)	8.7.0.0.0	8.7.0.0.0	After installing Oracle Analytics Client (OAC) tools, you can use UNIX shell commands to run OAC command line utilities. Refer to Using Oracle Analytics Client Tool Utilities on Linux and Generate an .rpd File from JSON/SMML, but ignore the references to downloading and installing the OAC tools locally – instead, you'll Run Unix Shell Commands in VB Studio.
OCIcli	3.71.0 or later	3.62.0 or later	Requires Python3
Oracle Developer Studio 12c 12.5	12.5	12.5	
Oracle Forms Developer	12.2.1.4.0	12.2.1.4.0	
Oracle Instant Client	21.20.0.0.0 or later 19.15.0 or later	21.20.0.0.0 or later 19.15.0 or later	
Oracle JDeveloper Studio	14.1.2.0.0	14.1.2.0.0	
Oracle JET Command-line Interface	19.0.0 or later	19.0.0 or later	Requires Node.js
Oracle SOA Suite 14	14.1.2.0.0	14.1.2.0.0	
Podman	4.9.4 or later	N/A	
<div>  Note Podman will not work if Docker is being used for build executors in your organization. </div>			
Python3	3.10.6 3.9.13 3.8.13 3.7.13	3.10.6 3.9.13 3.8.13 3.7.13	To invoke virtualization for your environment, run this command: <code>python3 -m venv <env_name></code>
Ruby	2.5.9p229 or later	2.0.0p648 or later	
SQLcl	25.4.0.0 or later	25.2.2.0 or later	Requires Java 11 or later
Xvfb	1.20.11 or later	1.20.4 or later	

Step 4: Set up VB Studio to run builds on Docker executors

Optional step: If you decided to go with Docker executors in Step 2, use this step.

If you decided to use VM build executors, go to [Step 3: Set up VB Studio to run VM build executors](#). You must use one or the other.

In VB Studio, you can run builds on Docker executors. You may want to use Docker executors for your execution platform if you have several templates and frequent jobs. Docker requires one management VM to run full-time, as well as one or more deployment VMs that can switch between templates as needed by different builds. Another reason you may want to use Docker executors instead of VM executors is if you want to include custom Docker images as the base for build executor templates.


You can import Docker images from your private or public Docker registry or create them from build executor templates, then deploy these images on VB Studio's Docker deployment VMs.

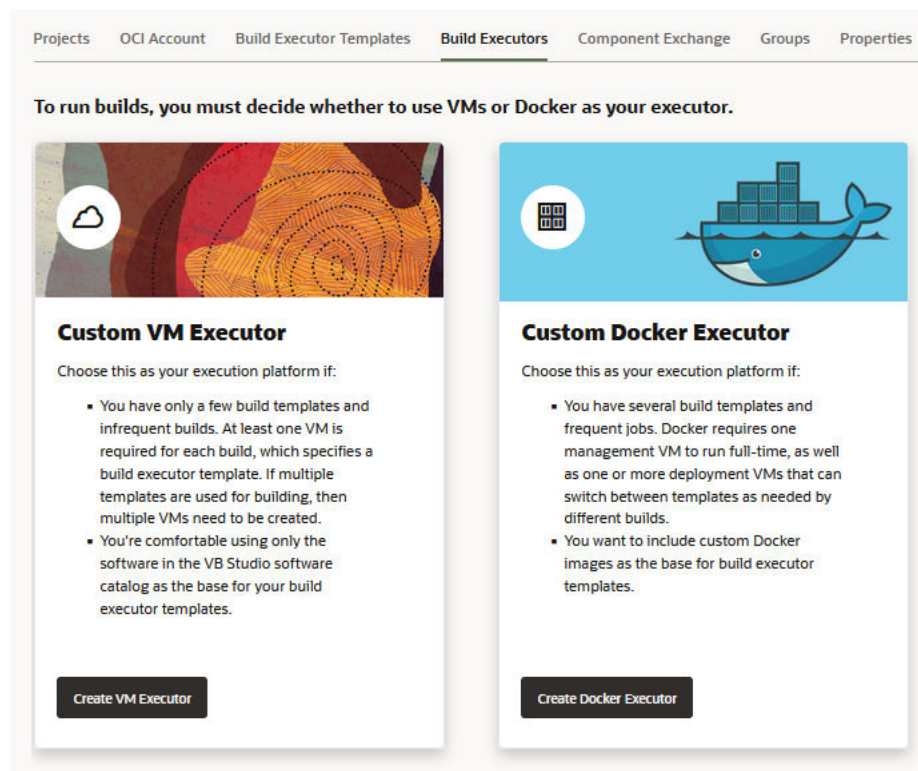
Add Your First Docker Deployment VM

If you haven't configured VB Studio to add VM executors, you can add your first Docker deployment VM from the Organization page's Build Executors tab.

Note

If you have created a VCN for your VM executors, make sure that you have the VCN configured to support Docker executors. For instructions, see [Use VB Studio's Default VCN](#) or [Create and Configure a Public Subnet in a VCN](#) for a custom VCN.

1. Configure VB Studio to connect to your OCI account.
2. In the left navigator, click **Organization** .
3. Click the **Build Executors** tab.
4. In the **Custom Docker Executor** tile, click **Create Docker Executor**.



5. Add details to the Add VMs for Creating Docker Executors dialog:

- a. **Number of VMs:** Specify the number of Docker deployment VMs you want to create.

The number you specify doesn't include the Docker Management VM. For example, if you specify 4 in **Number of VMs**, VB Studio will create five VMs: four Docker deployment VMs and one Docker Management VM.

- b. **Executors created per VM:** Enter the maximum number of Docker executors you want to deploy in each Docker deployment VM.

- c. **Region:** Select the region that's closer to you geographically.

The drop-down list displays regions your OCI account is subscribed to.

- d. **Shape:** Select the Docker executor's shape.

As this is your first Docker deployment VM, the Docker Management VM is also created with the same shape. Choose a shape with at least two OCPUs and 16 GB of RAM.

After selecting the shape, wait for a few seconds. VB Studio calculates the number of Compute VM instances that can be created with the selected shape and displays it in the dialog box. If the required number of Compute VM instances aren't available, choose another shape.

- e. **Volume:** Specify the storage capacity for each Docker deployment VM.
- f. **Number of OCPUs:** If you've selected a Flex shape, specify the number of OCPUs to add in each Docker deployment VM.
- g. **Amount of Memory:** If you've selected a Flex shape, specify each Docker deployment VM's memory.
- h. **Use only for project(s):** (Optional) Select one or more projects to assign to this Docker deployment VM. After the Docker deployment VM is created, it will be

restricted to running builds for the assigned projects. Builds for these projects will only run on this VM, and builds from other projects won't run on this VM.

6. For **VCN Select**, choose **Default** or **Custom**.
 - If you chose **Default**, click **Create**. You are finished with this task.
 - If you chose **Custom**, continue to the next step.
7. Fill out the additional details for the custom VCN:
 - a. **VCN Compartment**: Select the compartment.
If you're an Oracle Cloud OS Management Service (OSMS) user, don't select the OSMS compartment or a compartment with an OSMS policy.
 - b. **VCN**: Select the VCN.
 - c. **Subnets Compartments**: Select the compartments where your public subnets are. By default, it adds your VCN's compartment. If required, you can add more compartments.
 - d. **Subnets**: Select a subnet. The list shows public subnets only.
You can add multiple public subnets. If VB Studio can't create a Docker executor on the first subnet you've added, it tries to create it on the second subnet, and so on.
8. Click **Validate Network Setup**.
9. Click **Create**.

Before you try to use your Docker executors and images, you should check their states by visiting the tabs on the Organization page's **Build Executors** tab:

- The **Docker Images** tab shows the number of Docker images that are in use or are in an error state.

Organization

Projects OCI Account Build Executor Templates **Build Executors** Component Exchange Groups Properties Activities

Docker Images VM Pool Active Executors (0) Auto Refresh ☒

4 Images None of your images are in use. There are 2 images in an error state. ☐ Hide system images Update Selected Create Image

<input type="checkbox"/>	Name ^	Type ^	Size ^	Active Images	Status ^	Actions
<input type="checkbox"/>	DCS QA Req Comp Linux 7	Template	3.49 GB	0 of 4 allowed	Pending	...
<input type="checkbox"/>	DCS QA Req Comp Linux 8	Template	2.08 GB	0 of 4 allowed	Pending	...
<input type="checkbox"/>	System Base OL7	System	1.88 GB		Error	...
<input type="checkbox"/>	System Base OL8	System	2.08 GB		Error	...

- The **VM Pool** tab shows how many of your VMs are running or or are in an error state.
 - If you don't have any deployment VMs, an error message prompts you to define one.
 - If the management VM is down, an error message prompts you to try restarting it, or contact Oracle Support.

Organization

Projects OCI Account Build Executor Templates **Build Executors** Component Exchange Groups Properties Activities

Docker Images **VM Pool** Active Executors (0) Auto Refresh ☒ ↻

3 VMs
 ⚙️ One of your VMs is running.
 ✅ None of your VMs are in an error state.

<input type="checkbox"/>	VM Id ↕	Type ^	Restricted ↕	Details	Active Executors	Disk Usage ⓘ	Status ↕	Actions
<input type="checkbox"/>	961	Management		👁		<div><div></div></div> 10%	Ready	...
<input type="checkbox"/>	962	Deployment		👁	0 of 2 allowed	<div><div></div></div> 0%	Stopped	...
<input type="checkbox"/>	963	Deployment		👁	0 of 2 allowed	<div><div></div></div> 0%	Stopped	...

- The **Active Executors** tab shows how many Docker build executors are currently active.

Organization

Projects OCI Account Build Executor Templates **Build Executors** Component Exchange Groups Properties Activities

Docker Images VM Pool **Active Executors (0)** Auto Refresh ☒ ↻

Job Name ↕	Project Name ↕	Docker Image ↕	VM Id ↕	Started ↕	Status ↕	Actions
No Docker build executors are currently active						

Migrate to Docker



If you've created one or more VM executors and want to use Docker executors to run builds, you can migrate those VM executors to Docker.

You can choose to run builds for your organization on either VM executors or on Docker executors; you can't choose both. If the builds in your organization are already running on VM executors, you can migrate to Docker, and all builds in your organization going forward will use Docker executors.

When you perform a Docker migration, all of your VM executors are destroyed and recreated in Docker as images, separate from deployment VMs. If you want to revert back to VM executors, you will perform a reset that destroys all the Docker images and deployment VMs. You will then have to recreate your VM executors manually, so be sure to capture the details of your current VM executors before performing a migration in case you need to reset and go back.

Note

If you have VM executors that specify different custom VCNs, you *must* capture the details of all of the custom VCNs *before* migrating to Docker. The migration dialog will only let you choose one VCN, and you will need to recreate the deployment VMs with the details of your other custom VCNs.

1. If you're using the free VM build executor, configure VB Studio to connect to your OCI account.
2. In the left navigator, click **Organization** .
3. Click the **Build Executors** tab.
4. Capture details for your current VM executors in case you want to revert back from Docker executors to VM executors.
 - a. To capture ID, template, and shape information, copy the contents of the **Build Executors** table and paste to a text file.
 - b. To capture the contents of the **Details** column, click **Show machine details**  for each VM executor, copy the **VM Build Executor Configuration** details, and paste to a text file.
5. Click **Migrate to Docker**.
6. In the Do you want us to re-create your VM executors as Docker images dialog, enter the required details.
 - a. In **Number of VMs**, specify the number of Docker deployment VMs you want to create.
The number you specify doesn't include the Docker Management VM. For example, if you specify 4 in **Number of VMs**, VB Studio will create five VMs: four Docker deployment VMs and one Docker Management VM.
 - b. In **Executors created per VM**, enter the number of Docker executors you want in each Docker deployment VM.
 - c. In **Region**, select the region that's closer to you geographically.
The drop-down list displays regions your OCI account is subscribed to.
 - d. In **Shape**, select the Docker deployment VM's shape.
As this is your first Docker deployment VM, the Docker Management VM is also created with the same shape. For the image management VM's optimum performance, select a shape with at least 2 OCPUs and 16 GB memory.
Wait for a few seconds. VB Studio calculates the number of Compute VM instances that can be created with the selected shape and displays it in the dialog box. If the required number of Compute VM instances aren't available, choose another shape.
 - e. In **Volume**, specify the storage capacity for each Docker deployment VM.
 - f. If you've selected a Flex shape, then in **Number of OCPUs**, specify the number of OCPUs to add in each Docker deployment VM.
 - g. If you've selected a Flex shape, then in **Amount of Memory**, specify each Docker deployment VM's memory.
7. For **VCN Select**, choose **Default** or **Custom**.
 - If you chose **Default**, click **Migrate**. You are finished with this task.

- If you chose **Custom**, continue to the next step.
- 8. Fill out the additional details for the custom VCN:
 - a. **VCN Compartment**: Select the compartment.
If you're an Oracle Cloud OS Management Service (OSMS) user, don't select the OSMS compartment or a compartment with an OSMS policy.
 - b. **VCN**: Select the VCN.
 - c. **Subnets Compartments**: Select the compartments where your public subnets are. By default, it adds your VCN's compartment. If required, you can add more compartments.
 - d. **Subnets**: Select a subnet. The list shows public subnets only.
You can add multiple public subnets. If VB Studio can't create a Docker executor on the first subnet you've added, it tries to create it on the second subnet, and so on.
- 9. Click **Validate Network Setup**.
- 10. Click **Migrate**.

Create and Manage Docker Images

In VB Studio, you can import a Docker image from an external Docker registry or create a Docker image from a build executor template.

You'll need to configure VB Studio to connect to your OCI paid account before you can proceed. See [Create a VB Studio Instance in Your Oracle Cloud Account](#) or, if you've already created your VB Studio instance, see [Change Your OCI Account](#).


Note

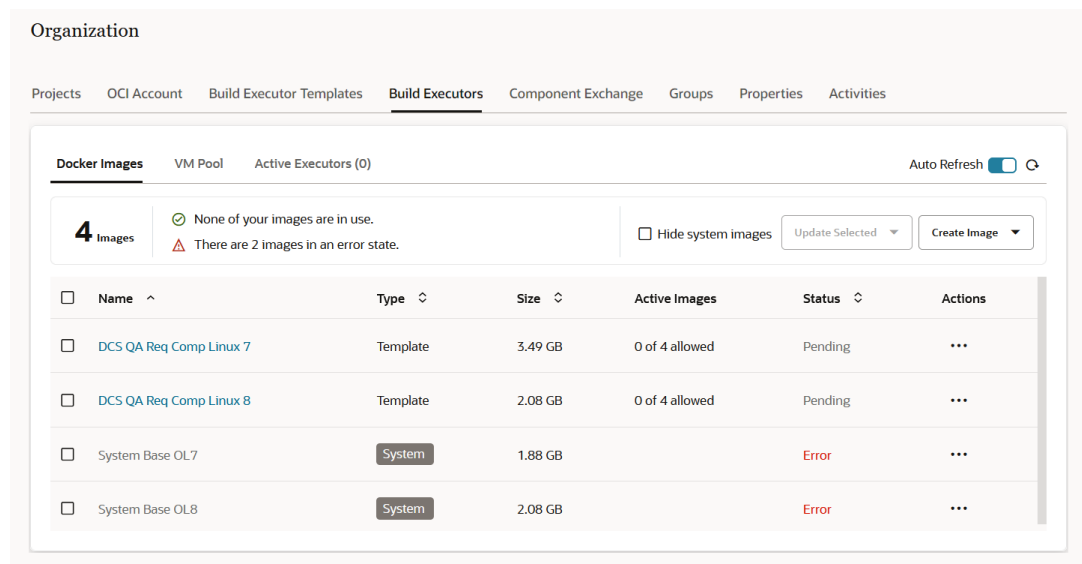
You can't create or manage Docker images if VB Studio is connected to the built-in free account.

VB Studio provides out-of-the-box base Docker images for your organization's users. You can't modify or delete these Docker images, and they can't be used for building your job. These base images are used to create Docker images from Docker build executor templates.

Docker image	Description
System Base OL7	Oracle Linux 7 base image with required software packages
System Base OL8	Oracle Linux 8 base image with required software packages

You can create and manage Docker images from the **Build Executors** tab on the Organization page.

1. In the left navigator, click **Organization** .
2. Click the **Build Executors** tab.
3. Click the **Docker Images** tab.
The tab shows the number of Docker images that are in use or are in an error state.



Notice that the system base images, indicated with the **System** badge, are displayed by default. To hide them, click the **Hide system images** checkbox to the right.

This table describes the actions you can perform to create and manage Docker images.

Action	How To
Import a Docker image from an external Docker registry	<ol style="list-style-type: none"> 1. Click Create Image and select Create Image from Registry. 2. In the Add Custom Docker Image dialog box, in Name, enter the image's name. 3. In Registry Host, enter the Docker registry's host name. 4. In Username and Password, enter the credentials of the users who can access the registry. If it's a public registry, leave the fields empty. 5. In Image Name, enter the Docker image's name to be imported. 6. In Version Tag, specify the Docker image's tag 7. In Max Executors, specify the maximum number of executors to be created from the imported image. 8. Click Add.
Create a Docker image from an executor template	<p>You can create only one Docker image from an executor template.</p> <ol style="list-style-type: none"> 1. Click Create Image and select Create Image from Build Executor Template. 2. From the Build Executor Template drop-down list, select the executor template. 3. In Max Executors, specify the maximum number of executors to be created from the image. 4. Click Save.
Change a Docker image's executor template	<p>Click Action *** and select Edit. In Build Executor Template, change the template and click Save.</p>


Action	How To
Change a Docker image's maximum executors	Click Action *** and select Edit . In Max Executors , change the number of executors and click Save .
Re-create a Docker image	Click Action *** and Recreate Image . In the confirmation dialog box, click OK .
View a Docker image's log	Click Action *** and select Show Log .
Delete a Docker image	Click Action *** and select Delete . In the Delete Docker Image dialog box, click Delete .

Add and Manage Docker Deployment VMs

You can add and manage Docker deployment VMs from the Organization Administration page's Build Executors tab.

Note

You can't add or manage Docker deployment VMs if VB Studio is connected to the built-in free account. If you want to add Docker deployment VMs, you'll need to configure VB Studio to connect to your OCI paid account.

1. In the left navigator, click **Organization** .
2. Click the **Build Executors** tab.
3. Click the **VM Pool** tab and follow one of these procedures:
 - To add Docker deployment VMs after you have added the initial VM, see [Add Docker Deployment VMs](#).
 - To view, start, stop, delete, or view the Docker VM logs, see [Manage Docker VMs](#).
 - To manage disk usage for a Docker VM, see [Manage Docker VM Resources](#).

Add Docker Deployment VMs

When you add a Docker deployment VM, you allocate an OCI VM Compute instance to run your builds in Docker executors.

1. From the **VM Pool** tab, click **+ Add VMs**.
2. Add details to the Creating Docker Executors dialog:
 - a. **Number of VMs:** Specify the number of Docker deployment VMs you want to create.
 - b. **Executors created per VM:** Enter the maximum number of Docker executors you want to deploy in each Docker deployment VM.
 - c. **Region:** Select the same region that you chose for the Management VM in Step 5c of [Add Your First Docker Deployment VM](#).
The drop-down list displays regions your OCI account is subscribed to.
 - d. **Shape:** Select the Docker executor's shape.

Wait for a few seconds. VB Studio calculates the number of Compute VM instances that can be created with the selected shape and displays it in the dialog box. If the required number of Compute VM instances aren't available, choose another shape.

- e. **Volume:** Specify the storage capacity for each Docker deployment VM.
 - f. **Number of OCPUs:** If you've selected a Flex shape, specify the number of OCPUs to add in each Docker deployment VM.
 - g. **Amount of Memory:** If you've selected a Flex shape, specify each Docker deployment VM's memory.
 - h. **Use only for projects:** (Optional) Select one or more projects to assign to this Docker deployment VM. After the Docker deployment VM is created, it will be restricted to running builds for the assigned projects. Builds for these projects will only run on this VM, and builds from other projects won't run on this VM.
 - i. **Use only for images:** (Optional) Select one or more images to assign to this Docker deployment VM. After the Docker deployment VM is created, it will be restricted to running builds using the assigned images.
 - j. **Reserved Public IP:** (Optional) Select a Reserved Public IP to assign to this Docker deployment VM. The Reserved Public IP must be defined in the compartment of the specified OCI account. After the Docker deployment VM is created, the specified Reserved Public IP will always be assigned and used by the Docker deployment VM.
3. For **VCN Select**, choose **Default** or **Custom**.
 - If you chose **Default**, click **Create**. You are finished with this task.
 - If you chose **Custom**, continue to the next step.
 4. Fill out the additional details for the custom VCN:
 - a. **VCN Compartment:** Select the compartment.

If you're an Oracle Cloud OS Management Service (OSMS) user, don't select the OSMS compartment or a compartment with an OSMS policy.
 - b. **VCN:** Select the VCN.
 - c. **Subnets Compartments:** Select the compartments where your public subnets are. By default, it adds your VCN's compartment. If required, you can add more compartments.
 - d. **Subnets:** Select a subnet. The list shows public subnets only.

You can add multiple public subnets. If VB Studio can't create a Docker executor on the first subnet you've added, it tries to create it on the second subnet, and so on.
 5. Click **Validate Network Setup**.
 6. Click **Create**.

Note


After the Docker deployment VM is created, it is listed on the VM Pool tab of the Build Executors page. Restrictions for both projects and images are indicated with a Lock icon in the Restricted column. Click the Lock icon to display details about the assigned Docker deployment VM restrictions.

Manage Docker VMs

You can view, start, stop, or delete a Docker deployment VM, view the VM log from the Organization Administration page's Build Executors tab, or show available resources for the

VM. You can also reset, view the log, or show available resources for the Docker management VM.

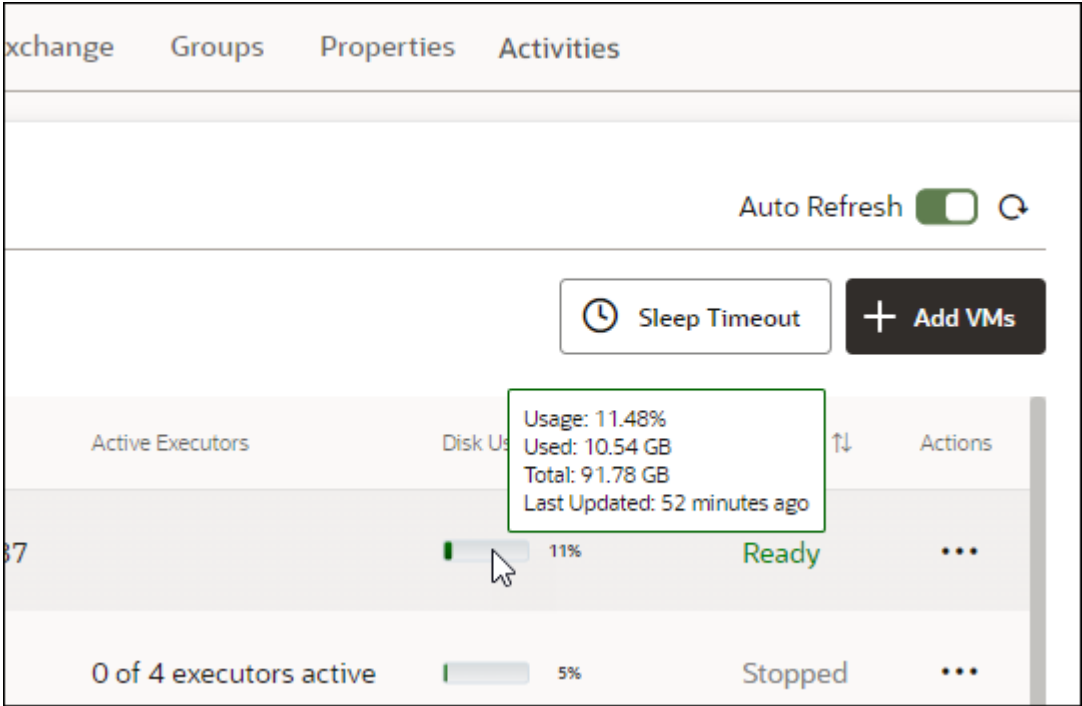
From the **VM Pool** tab, choose one of the actions shown in the following table:

Action	How To
View a Docker deployment VM's or Docker management VM's log	Click Action *** and select Show Log . In the Docker Executor VM Log window, review the log.
Start or stop a Docker deployment VM	Click Action *** and select Start or Stop . To start or stop multiple Docker deployment VMs, select their check boxes, click Update Selected and select Start selected VMs or Stop selected VMs .
Delete a Docker deployment VM	Click Action *** and select Delete . In the confirmation dialog box, click OK . To delete multiple Docker deployment VMs, select their check boxes, click Update Selected and select Delete selected VMs .
Reset a Docker management VM	Click Action *** and select Reset . In the confirmation dialog box, click OK .
View and clean up VM resources	Click Action *** and select Show Resources.. . For more information, see Manage Docker VM Resources .
View the projects associated with a restricted Docker deployment VM.	In the Restricted column, hover over  to view the projects associated with the VM.

Manage Docker VM Resources

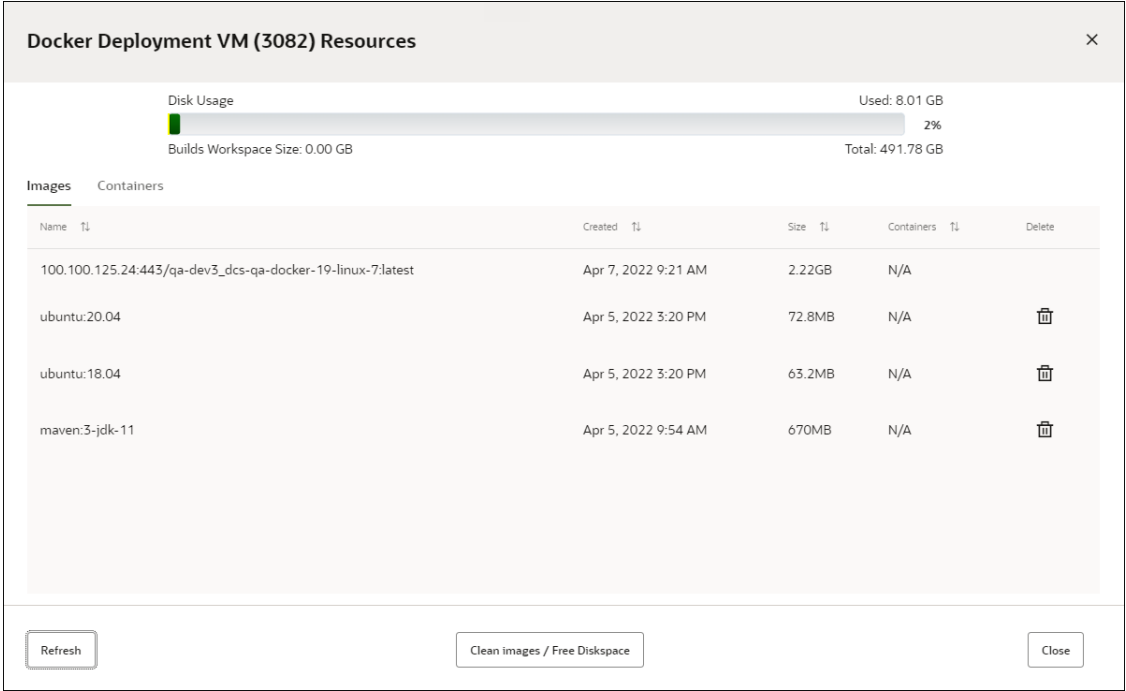
You can view the total disk usage available for a Docker Management VM or Docker Deployment VM and delete images or containers to free up disk space.

For a quick look at the disk usage for a Docker VM, hover over the Disk Usage bar to view usage percentage, GB used, total GB available, and the date that the data was last updated.



The data on this page is only updated every hour or so. For real time usage and additional detail, view the Resources page for the VM.

To view the Resources page for a Docker VM, click ******* and select **Show Resources**.



Note

If you are viewing Docker Management VM resources, you will see the Internal Registry Size instead of the Builds Workspace Size.

From here you can view total disk usage for the VM, as well as usage for individual images and containers.

The following table describes actions you can take to free disk space.


Action	How To
Delete an individual image (for Docker deployment VMs).	In the Images tab, click the trash can next to the image.
Free disk space by removing images that are no longer in use.	In the Images tab, click the Clean Images/Free Diskspace button at the bottom.

Reset Docker and Move to VM Build Executors

If you want to run your organization's builds on VM build executor, you can reset Docker on the Management VM from the Organization page's Build Executors tab.

Note

If you reset Docker, *all of your VMs and images will be destroyed* and your VM executors will need to be created from scratch. Capture the details of your images and deployment VMs before you begin so they can be recreated manually when you switch to VM executors.

1. In the left navigator, click **Organization** .
2. Click the **Build Executors** tab.
3. Click the **VM Pool** tab.
4. For the **Management VM**, click *** and select **Reset**.
5. In the Delete Docker Deployment VM dialog box, click **OK**.

When you've finished with the migration, you will need to create the VM build executors. See [Add and Manage VM Build Executors](#).

Step 5: Set Up Users and Groups

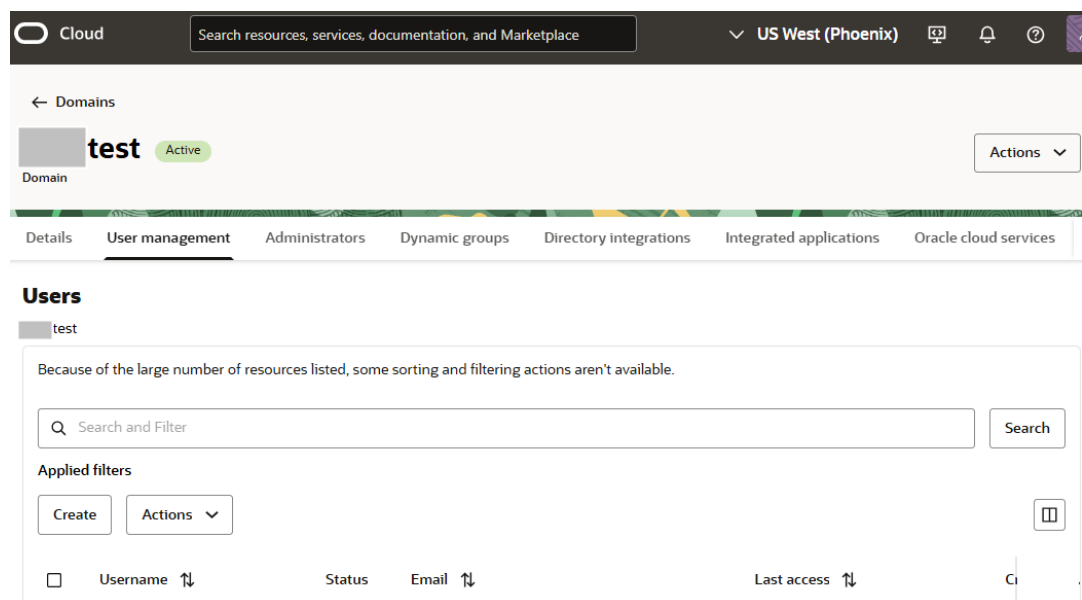
Required step: To add users to VB Studio and its projects, you must first add them to your identity domain in Oracle Cloud Infrastructure (OCI) Identity and Access Management, and then assign the appropriate VB Studio roles.

To federate with your existing identity provider, see [Federating with Identity Providers](#).

First, add users to your identity domain:

1. In a web browser, go to <https://cloud.oracle.com>.

2. In **Cloud Account Name**, enter your Oracle Cloud account or tenancy name and click **Next**.
3. On the Single Sign-On (SSO) panel, if required, select your identity domain and click **Continue**.
4. Enter your Oracle Cloud account credentials and click **Sign In**.
If you've recently created your Oracle Cloud account, wait for some time to see your services in the Oracle Cloud Console.
5. In the upper-left corner, click **Navigation Menu** ☰.
6. Select **Identity & Security** and then, under Identity, select **Domains**.
7. On the Domains page, click the name of the identity domain that you want to work in. You might need to change the compartment, next to **Applied filters**, to find the domain that you want.
8. After clicking the identity domain, click the **User management** tab.



9. Click **Create** and enter the new user's details. See [Creating a User](#).
10. You can optionally select groups to assign to this user. As a best practice, you might prefer to add users to groups, and then assign VB Studio roles to those groups.
To create OCI groups, see [Managing Groups](#).
See also [Manage Your Organization's Groups](#).
11. Click **Create**.

Next, assign the VB Studio roles to your users:

1. From your identity domain's **User management** tab, now click the **Oracle cloud services** tab.
2. Search for and select the service that begins with either **DevServiceAppAUTO_** or **VisualBuilderStudioInstance_**:
 - Select **DevServiceAppAUTO_** for older VB Studio instances that were migrated from Oracle Gen 1 Cloud to Oracle Gen 2 Cloud Infrastructure, also known as Oracle Cloud Infrastructure (OCI).

- Select **VisualBuilderStudioInstance_** for newly created Gen 2 (OCI) VB Studio instances.

3. Click **Application roles**.

A user must be assigned one of these two roles to access VB Studio, either directly or indirectly via a group.

This VB Studio role...	Enables a user to:
DEVELOPER_ADMINISTRATOR	Set up VB Studio, manage all projects, manage VM executors and executor templates, view the Recent Activities feed, and update the organization details. The user with this role is also called the <i>Organization Administrator</i> . Assign this role to users who can administer VB Studio.
DEVELOPER_USER	Create and access VB Studio projects. All non-admin users of VB Studio must be assigned this role. Note that this role doesn't allow the user to update the organization details and view the Recent Activities feed.

4. For each VB Studio role that you want to assign, click the three dots at the end of the row and then click **Manage users**.
You can also click **Manage groups** to assign this role to a group.
5. The Manage user assignments page lists the users already assigned to the selected VB Studio role. Click **Assign users** to add a new user to this role.
6. On the Assign users page, search for and select the user you'd like to grant this VB Studio role to, then click **Assign**.
7. Click **Close**.
8. Repeat these steps to assign VB Studio roles to additional users or groups.

6


Manage the VB Studio Instance

After creating the VB Studio instance, you change the CI/CD compartment and the instance's display name, but can't change the name displayed in the instance's URL.

- [Change the CI/CD Compartment](#)
- [Authorize VB Studio to Access Visual Builder and Integration Instances](#)
- [Switch to the Built-In Free Account](#)
- [Change Your OCI Account](#)
- [Validate the OCI Connection](#)
- [Manage VB Studio Instance Tags](#)
- [Write IAM Policies that Control Access to VB Studio](#)
- [Create Rules that Use Service Event Notifications](#)
- [Move the VB Studio Instance](#)
- [Terminate the VB Studio Instance](#)

Change the CI/CD Compartment

You can switch to another OCI compartment any time. When you switch, your current compartment's executor templates and VM executors migrate to the new compartment, but Object Storage buckets do not migrate. The data in buckets may include build artifacts, Maven artifacts, NPM artifacts, and exported project data.

1. In a web browser, go to <https://cloud.oracle.com>.
2. In **Cloud Account Name**, enter your Oracle Cloud account or tenancy name and click **Next**.
3. On the Single Sign-On (SSO) panel, if required, select your identity domain and click **Continue**.
4. Enter your Oracle Cloud account credentials and click **Sign In**.
5. In the upper-left corner, click **Navigation Menu** .
6. Select **Developer Services** and then select **Visual Builder Studio**.
7. On the Visual Builder Studio instances page, click the VB Studio instance's name.
8. On the instance details page, click **Actions**, then click **Edit**.
9. Under **CI/CD setup**, select the new compartment you want to use for VM executors and storage.
10. Click **Save changes**.

The VB Studio instance is updated to use the specified compartment.


If you switch to the built-in free account, you'll lose access to data in Object Storage, and your current account's executor templates and VM executors. Instead, you'll get access to only one free VM build executor and the default executor template.

Authorize VB Studio to Access Visual Builder and Integration Instances

You may be required to authorize VB Studio to access Oracle Visual Builder and Oracle Integration instances that are connected to your OCI account. This authorization allows users to view Visual Builder and Integration instances linked to the OCI account and create environments with these instances.

So when might you need to do this? Mostly when your VB Studio instance is using the built-in free VM account—but there are several other scenarios when VB Studio isn't authorized to access the Visual Builder and Integration instances linked to your OCI account. Therefore, it's recommended that you always check your VB Studio instance's details to make sure the policy statement that enables this access is correctly defined.

Here's how to check if a VB Studio policy has been defined to include the statements granting read permissions for `visualbuilder-instance` and `integration-instance` resources:

1. Open the Oracle Cloud Console and click **Navigation Menu**  in the upper-left corner.
2. Select **Identity & Security**, then under Identity, select **Policies**.
3. On the Policies page, next to **Applied filters**, make sure the `root` compartment is selected.
4. Locate the policy that was automatically created for your VB Studio instance. The policy name starts with `VBStudio-Policy-`. Click the policy, then click the **Statements** tab.

If a policy exists but is missing one of these statements, update it to include *all* these statements; if a policy does not exist, create one to include *all* these statements in this format:

```
Allow any-user to inspect all-resources in tenancy where
request.principal.id in ('<VBS_instance_OCID>')
Allow any-user to read visualbuilder-instance in tenancy where
request.principal.id in ('<VBS_instance_OCID>')
Allow any-user to read integration-instance in tenancy where
request.principal.id in ('<VBS_instance_OCID>')
```

If you have multiple VB Studio instances (say, because each TEST and DEV instance in your Oracle Cloud Application environment family got its own instance of VB Studio), enter all their OCIDs as a comma-separated list within single quotation marks ('), for example:

```
Allow any-user to inspect all-resources in tenancy where
request.principal.id in ('ocidl.vbsinstance..abc',
'ocidl.vbsinstance..xyz')
Allow any-user to read visualbuilder-instance in tenancy where
request.principal.id in ('ocidl.vbsinstance...abc',
'ocidl.vbsinstance...xyz')
Allow any-user to read integration-instance in tenancy where
request.principal.id in ('ocidl.vbsinstance...abc',
'ocidl.vbsinstance...xyz')
```

Here's how you can retrieve your VB Studio instance's OCID:

- a. In the navigation menu, select **Developer Services** and then **Visual Builder Studio**.

- b. On the Visual Builder Studio instances page, next to **Applied filters**, make sure the root compartment is selected.
- c. Find your VB Studio instance in the table and, at the end of the row, click the three dots and then click **Copy OCID**.


Note

It's important to use the same format as your existing policy. This format depends on when and how your OCI account was configured. So if CI/CD authorization for your OCI account was enabled some time ago, you may see an older format with statements like (for example) `Allow service vbstudio`. In this case, update your policy to include the missing statements in the same format:

```
Allow service vbstudio to inspect all-resources in tenancy
Allow service vbstudio to read visualbuilder-instance in tenancy
Allow service vbstudio to read integration-instance in tenancy
```

Switch to the Built-In Free Account

If you switch to the built-in free account, your current account's build executor templates and build executors will be deleted and you'll have access to only one free VM build executor with fixed software in the default build executor template. You can't add more build executors or edit the executor template's software.

1. In the Oracle Cloud Console, in the upper-left corner, click **Navigation Menu** .
2. Select **Developer Services** and then select **Visual Builder Studio**.
3. On the Visual Builder Studio instances page, click the VB Studio instance's name.
4. On the instance details page, click **Actions**, then click **Edit**.
5. Deselect the **Yes, I authorize this** checkbox.
6. In the **Confirm delete** dialog box, click **Delete resources**.
7. Click **Save changes**.

Change Your OCI Account


If you're using the built-in free account or your own OCI paid account, you can switch to another OCI account any time and use the new account's resources.

If you switch to an OCI paid account:

- Your current OCI account's build executor templates and their metadata migrate to the new OCI account. For example, If you have 10 executor templates in your current OCI account, VB Studio creates the same number of executor templates with the same software in the new OCI account.
- Your current OCI account's VM build executors and their metadata migrate to the new OCI account. For example, if you have 234 VM executors in your current OCI account, VB Studio creates the same number of VM executors of the same shape, VCN, and executor templates in the new OCI account.

- Your current OCI account's Object Storage buckets **do not** migrate to the new account. The data in buckets may include build artifacts, Maven artifacts, NPM artifacts, and exported project data. (To migrate data in Object Storage, see [Export and Import Project Data](#). But, if you were using your original OCI account solely to run CI/CD builds, you can skip this step.)

To switch from the built-in free account to your OCI paid account:

1. In the Oracle Cloud Console, in the upper-left corner, click **Navigation Menu** .
2. Select **Developer Services** and then select **Visual Builder Studio**.
3. On the Visual Builder Studio instances page, click the VB Studio instance's name.
4. On the instance details page, click **Actions**, then click **Edit**.
5. Select the **Yes, I authorize this** checkbox.

Edit Visual Builder Studio

General information

Compartment
 (root)

Name
 VBS-dev-instance

CI/CD setup

Visual Builder Studio requires authorization to create Compute instances and Storage buckets in your Oracle Cloud account. We use the Compute instances to execute your CI/CD pipelines, and the Storage buckets to store build artifacts, as well as to enable the binary repository features in your projects (Maven, NPM, and Git LFS). If you have a pay-as-you-go Cloud account, you will incur charges only when you execute a pipeline or consume storage, and only after your overall usage exceeds the Oracle Cloud Free Tier limits. Do you authorize Oracle to use these resources?

☒ Yes, I authorize this

Compartment
 DemoCompartment


Select the compartment where you want us to create the Compute and Storage instances

6. Select the compartment to host your VB Studio resources.
7. Click **Save changes**.

If you [switch to the built-in free account](#) from an OCI paid account, you'll lose access to your current account's executor templates and VM executors. Instead, you'll get access to only one free VM build executor and the default executor template. You'll also lose access to data in Object Storage, but again, if you were using your original OCI account solely to run CI/CD builds, you don't need to worry about data migration.

Validate the OCI Connection

If you notice any Compute or Storage issues, validate your OCI connection:


1. In the left navigator, click **Organization** .
2. Click the **OCI Account** tab.
3. Click **Validate**.

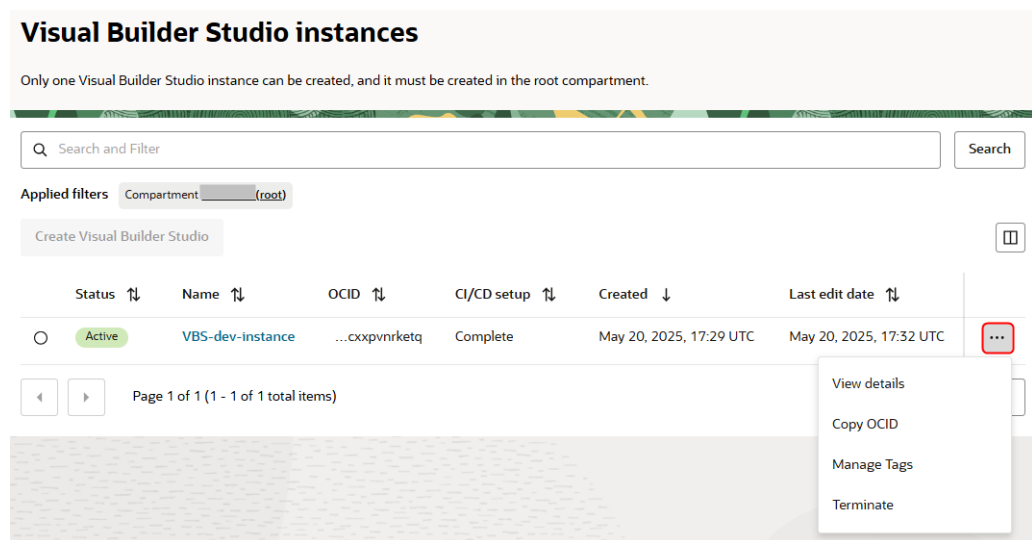
If the validation fails, click **Edit** to edit the connection, or switch to another OCI account. If your OCI account has reached Compute and Storage service limits, contact the OCI administrator to [increase the service limit](#).

Manage VB Studio Instance Tags

OCI tags enable you to tag your OCI resources, such as your VB Studio instance, and help you organize resources based on your business needs. You can manage VB Studio tags from the Visual Builder Studio instances page or directly from an instance's details page.

If you're new to OCI tags, see [Tagging Overview](#).

- To manage tags from the Visual Builder Studio instances page:
 - Open the Oracle Cloud Console and click **Navigation Menu**  in the upper-left corner.
 - In the navigation menu, select **Developer Services** and then **Visual Builder Studio**.
 - For your VB Studio instance on the Visual Builder Studio instances page, click the three dots at the end of the row and then click **Manage Tags**.



- On the Manage Tags page, click **Add tag**, then enter the tag's namespace, key, and value.
You can also edit and delete tags on this page.
 - Click **Save**.
- To manage tags from a VB Studio instance's details page:
 - On the Visual Builder Studio instances page, click the VB Studio instance's name.
 - On the instance details page, select the **Tags** tab.
 - To add a tag, click **Add**, then **Add tag**.
 - To edit or delete a tag, click the three dots at the end of the tag's row and then click **Edit** or **Delete**.

Write IAM Policies that Control Access to VB Studio

Identity and Access Management (IAM) is an Oracle service that helps you manage the permissions that control which OCI resources your users can access. You use IAM to control who is authenticated (signed in) and who is authorized (has permissions) to use resources. In

essence, IAM ensures that the right users have the appropriate access to resources. For VB Studio, that means the vbstudio-instance resource. This topic provides information you can use to write policies that control this access.

Supported Variables

You use variables when you add conditions to a policy. These conditions are evaluated by the OCI authorization service when API requests are authorized.

For the complete list of general variables that are applicable to all requests, including those that can be used in policy statements authorizing VB Studio API requests, see [General Variables for All Requests](#).

Details for Meta-Verb + Resource-Type Combinations

The following table shows the visualbuilder-instance resource type permissions and API operations that are covered by each meta-verb. The level of access is cumulative as you go from inspect to read to use to manage.

Verb	Permissions	APIs Fully Covered
INSPECT	<ul style="list-style-type: none"> VBS_INSTANCE_INSPECT 	<ul style="list-style-type: none"> ListVbsInstances ListWorkRequests
READ	<ul style="list-style-type: none"> Inherits from INSPECT: <ul style="list-style-type: none"> VBS_INSTANCE_INSPECT VBS_INSTANCE_READ 	<ul style="list-style-type: none"> GetVbsInstance GetWorkRequest ListWorkRequestErrors ListWorkRequestLogs
USE	<ul style="list-style-type: none"> Inherits from READ: <ul style="list-style-type: none"> VBS_INSTANCE_INSPECT VBS_INSTANCE_READ VBS_INSTANCE_UPDATE 	<ul style="list-style-type: none"> UpdateVbsInstance
MANAGE	<ul style="list-style-type: none"> Inherits from USE: <ul style="list-style-type: none"> VBS_INSTANCE_INSPECT VBS_INSTANCE_READ VBS_INSTANCE_UPDATE VBS_INSTANCE_CREATE VBS_INSTANCE_DELETE 	<ul style="list-style-type: none"> CreateVbsInstance DeleteVbsInstance

Permissions Required for Each API Operation

These are the permissions that are required for each API operation for instances and work requests.

API Operation	Description	Permissions Required to Use the Operation
ListVbsInstances	List the VB Studio instances	VBS_INSTANCE_INSPECT
GetVbsInstance	Get the VB Studio instance	VBS_INSTANCE_READ
CreateVbsInstance	Create a VB Studio instance	VBS_INSTANCE_CREATE
DeleteVbsInstance	Delete a VB Studio instance	VBS_INSTANCE_DELETE
UpdateVbsInstance	Update a VB Studio instance	VBS_INSTANCE_UPDATE
ListWorkRequests	List a VB Studio instance's work requests	VBS_INSTANCE_INSPECT
GetWorkRequest	Get a VB Studio instance's work request	VBS_INSTANCE_READ
ListWorkRequestErrors	List a VB Studio instance's work request errors	VBS_INSTANCE_READ
ListWorkRequestLogs	List a VB Studio instance's work request logs	VBS_INSTANCE_READ

Sample OCI IAM Policy Statements that Grant These Permissions

This example grants fine-grained permissions to a group called devteam1:

```
Allow devteam1 to
{VBS_INSTANCE_INSPECT,VBS_INSTANCE_READ,VBS_INSTANCE_CREATE,VBS_INSTANCE_UPDATE,VBS_INSTANCE_DELETE,VBS_INSTANCE_MOVE} in tenancy
```

Here's another example that uses the friendlier meta-verbs to grant the same permissions to group devteam1 and a lesser set to another group, devteam2:

```
Allow devteam1 to manage vbstudio-instances in tenancy
```

```
Allow devteam2 to use vbstudio-instances in tenancy
```

Create Rules that Use Service Event Notifications

This step is optional.

Many actions that you perform, such as creating, modifying, or deleting a VB Studio instance, emit event notifications. You can use the OCI console to define *rules* that trigger a specific action when an event occurs and a notification about that event is sent. For example, you might want to define a rule that sends a notification to an administrator (or group of administrators) when a VB Studio instance is created, updated, or deleted.

For more information about events and how to get started using them in rules, see [Overview of Events](#) and [Get Started with Events](#).

Here are the Visual Builder Studio events that you can use in a rule:

Event Name	Event Type
Visual Builder Studio - Create Instance - begin	com.oraclecloud.vbstudioinst.createvbsinstance.begin
Visual Builder Studio - Create Instance - end	com.oraclecloud.vbstudioinst.createvbsinstance.end
Visual Builder Studio - Delete Instance - begin	com.oraclecloud.vbstudioinst.deletevbsinstance.begin
Visual Builder Studio - Delete Instance - end	com.oraclecloud.vbstudioinst.deletevbsinstance.end
Visual Builder Studio - Update Instance - begin	com.oraclecloud.vbstudioinst.updatevbsinstance.begin
Visual Builder Studio - Update Instance - end	com.oraclecloud.vbstudioinst.updatevbsinstance.end

Move the VB Studio Instance

To use VB Studio with a different Oracle Cloud Applications instance, you'll need to file a service request so the VB Studio instance associated with your current TEST or DEV instance can be terminated, then a new one can be created and associated with your preferred instance.

Include these details in the service request:

- Current instance URL for the Oracle Cloud Applications pod that's associated with your VB Studio instance

- Your VB Studio service instance URL
See [How Do I Find the URL of My VB Studio Instance?](#)
- Instance URL for the destination Oracle Cloud Applications pod that you want to move VB Studio to
- Your request for assistance with backing up your existing VB Studio project, if you've used your current VB Studio instance and want to retain project data
- Your approval to terminate the current VB Studio instance

Terminate the VB Studio Instance

When you terminate your VB Studio instance, all projects and their data (such as Git repositories, workspaces, build jobs, and build artifacts) are deleted. OCI resources, such as VM executors, are destroyed and OCI Object Storage buckets are deleted, as well.

1. Open the Oracle Cloud Console and click **Navigation Menu** ☰ in the upper-left corner.
2. In the navigation menu, select **Developer Services** and then **Visual Builder Studio**.
3. For your VB Studio instance on the Visual Builder Studio instances page, click the three dots at the end of the row and then click **Terminate**.

Visual Builder Studio instances

Only one Visual Builder Studio instance can be created, and it must be created in the root compartment.

Search and Filter Search

Applied filters: Compartment (root)

Create Visual Builder Studio ☰

Status	Name	OCID	CI/CD setup	Created	Last edit date	
Active	VBS-dev-instance	...cxpxvnrketq	Complete	May 20, 2025, 17:29 UTC	May 20, 2025, 17:32 UTC	⋮

Page 1 of 1 (1 - 1 of 1 total items)

- View details
- Copy OCID
- Manage Tags
- Terminate

You can also click the instance's name and terminate the instance from the details page.

4. On the Confirm page, enter the VB Studio instance's name and click **Terminate**.
- Wait for some time. OCI takes a few minutes to terminate the VB Studio instance.

7


Manage the Organization

From the Organization page, you can manage projects and update the organization's properties.

- [Update the Organization's Display Name and Description](#)
- [Set the Organization's Default Wiki Markup Language](#)
- [Restrict Project Creation to Organization Administrators](#)
- [Restrict Sharing a Project to Organization Administrators Only](#)
- [Limit the Number of Personal Access Tokens a User Can Create](#)
- [Assign a Project's Ownership to Another User](#)
- [Manage Your Organization's Projects](#)
- [Manage Your Organization's Groups](#)


Update the Organization's Display Name and Description

By default, the organization's display name is `<your_vbstudio_instance_name>-<your_oracle_account_name>`.

1. In the left navigator, click **Organization** .
2. Click the **Properties** tab.
3. In **Name**, edit the organization name.
4. In the Confirm Organization Name Change dialog box, click **Yes**.
5. In **Description**, if necessary, enter or edit the organization's description.


Set the Organization's Default Wiki Markup Language

When you create a project, the project creation wizard displays the organization's default wiki markup language. If you don't change the wiki markup language, the default markup language is used as the project's markup language. You can change the default markup language for new projects, but note that existing projects continue to use their original markup language.

1. In the left navigator, click **Organization** .
2. Click the **Properties** tab.
3. In **Markup Language**, select the default wiki markup language.

Restrict Project Creation to Organization Administrators

By default, all VB Studio users (users with the DEVELOPER_ADMINISTRATOR IDCS role or the DEVELOPER_USER IDCS role) can create a project. As the Organization Administrator, if required, you can restrict project creation to yourself and other organization administrators (users with the DEVELOPER_ADMINISTRATOR IDCS role).


1. In the left navigator, click **Organization** .
2. Click the **Properties** tab.
3. Select the **Limit project creation to organization administrators only** check box.

The **+ Create** button on the All Projects page and in the Switch Project menu is now hidden for users assigned the DEVELOPER_USER IDCS role. To allow them to create projects, deselect the **Limit project creation to organization administrators only** check box.

Restrict Sharing a Project to Organization Administrators Only

By default, all VB Studio users can create shared projects. Users who are project owners can share their private projects and make them accessible to all users of the organization.

As the Organization Administrator, if required, you can restrict the users' ability to create a shared project or share a private project. Once this restriction is set, only you and other organization administrators can create a shared project or share a private project.


1. In the left navigator, click **Organization** .
2. Click the **Properties** tab.
3. Select the **Limit shared project to organization administrators only** check box.

After the check box is selected, the **Shared** option under **Security** in the New Project wizard and the **Project Administration > Properties** page is disabled for users who are assigned the DEVELOPER_USER IDCS role. They can't change a private project's security to shared, but they can change a shared project's security to private.

Limit the Number of Personal Access Tokens a User Can Create

An organization administrator can limit the maximum number of tokens a user in the organization is allowed to create. This action can prevent users from creating excessive numbers of personal access tokens. It also makes it easier for an administrator to maintain existing tokens. The **Organization Properties** page provides an option for controlling the maximum number of tokens each user can create. The default limit is 10 personal access tokens per user, but this limit can be increased to a maximum of 50 tokens per user. The set value applies to all users, including the organization administrator, who sets the limit.

To set a new limit for the number of personal access tokens a user can create:

1. In the left navigator, click **Organization** .
2. Click the **Properties** tab.
3. In the Access Token Controls section, at the bottom of the page, select the **Enforce a per-user limit on the number of personal access tokens** check box.
4. In the **Number of tokens** field, use the selector to set the maximum number of personal access tokens each user in the organization can have.

The default value is 10. The maximum number is 50.

After the check box is selected and a limit set, the **User Preferences**, **Maven Browse**, and **NPM Browse** pages will use the set limit to deal with the situation that arises when a user's token limit has been reached. When the limit is decreased, the new limit applies to new token creation only. It doesn't force users to delete any of their existing tokens. In other words, it doesn't affect the number of already-created tokens but it can prevent them from creating any new tokens until they delete enough of the existing tokens to be under the limit.

To remove the limit, deselect the **Enforce a per-user limit on the number of personal access tokens** check box.

Note

VB Studio automatically deletes tokens 30 days after they expire, but not without giving users plenty of advance notice first:

- An "Expires Soon" status label is displayed next to the token 5 days before the actual expiration date.

Visual Builder Studio

< User Preferences: Alex Admin

Profile | SSH Keys | **Personal Access Tokens** | Notifications | Language | General

Personal Access Tokens

Token Label	Created On	Expires On	Status
NonExpirableToken	Created October 7, 2022		Non-Expirable
AboutToExpire	Created February 13, 2024		Expires Soon
ExpirableToken	Created April 12, 2022		Expirable
AboutToExpire Token 2	Created February 13, 2024		Expires Soon
Token 1 - Expirable	Created February 13, 2024		Expirable
SampleNonExpirableToken-2	Created January 13, 2023		Non-Expirable
tokenAfter2	Created December 28, 2023		Expirable
Token 2 - Non Expirable	Created February 13, 2024		Non-Expirable
createdToken1	Created February 13, 2024		Expirable

AboutToExpire Details:

- Created On: February 13, 2024 3:24 PM +0530
- Expires On: February 18, 2024 5:29 AM +0530
- Last Used: Never
- Token: 7JzR.....TjZ8

- An email notification is sent to the user 5 days before the user's personal token is about to expire.

Personal Access Tokens are about to expire

Oracle Visual Builder Studio <do-not-reply@oracle.com>
To: Alex Admin

Today at 7:19 PM

This is a courtesy email, as access tokens cannot be refreshed prior to their expiration date, nor can they be restored after expiration. We suggest you create a new token now and replace the existing token to avoid disruption.

Personal Access Tokens Details


Token Label	Created On	Expires On	Last Used
AboutToExpire	Tue, Feb 13, 2024 9:54 AM UTC	Sat, Feb 17, 2024 11:59 PM UTC	Never
AboutToExpire Token 2	Tue, Feb 13, 2024 1:22 PM UTC	Sat, Feb 17, 2024 12:00 AM UTC	Tue, Feb 13, 2024 1:32 PM UTC

[Go to Oracle Visual Builder Studio](#)


[About Oracle](#) | [Contact Us](#) | [Legal Notices](#) | [Term of Use](#) | [Your Privacy Rights](#)

Assign a Project's Ownership to Another User


At times, you may want to assign a project's ownership to another user. For example, when a project's owners are removed or have left the organization, you should assign the Project Owner role to another project user.

1. In the left navigator, click **Organization** .
2. Click the **Projects** tab.
3. Click the **Owner** filter button.

If you can't find the project in the filtered list, you aren't assigned the project's Project Owner membership. You should assign the project's ownership to yourself before you assign it to another user.


- a. Click the **All** filter button.
 - b. Locate the project.
 - c. Click the three dots ******* at the end of the row.
 - d. Select **Assign Me as Owner**.
4. Click the project's name to open it.
 5. Click the **Team** tab.
 6. If the user whom you want to assign the ownership is already a member, click the **Change Membership**  icon and select **Project Owner**.

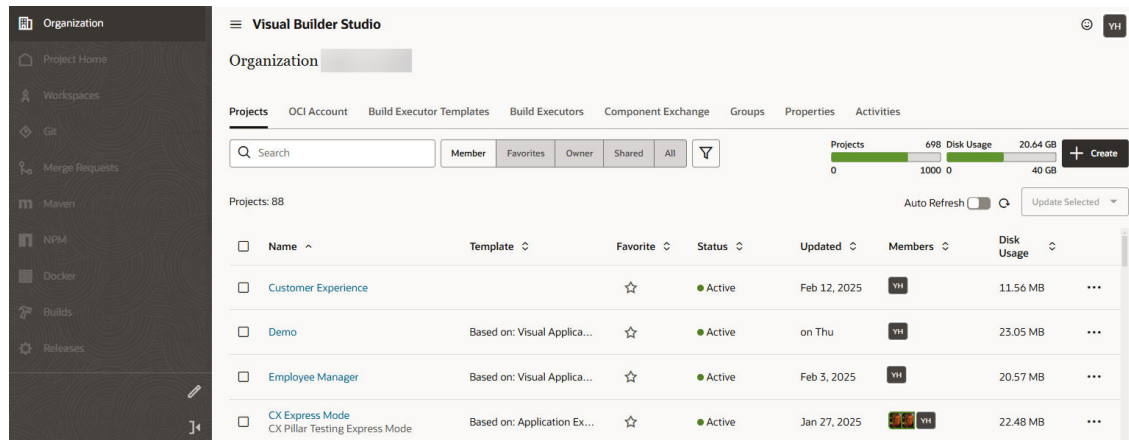
If the user is not a member, add the user to the project.

- a. Click **+ Add Member**.
 - b. Click the **Username** drop-down list and select the user.
 - c. Select the **Project Owner** membership.
 - d. Click **Add**.
7. After adding the user, you can remove yourself from the project. Click **Remove**  next to your name. In the dialog box that opens, click **OK**.



Manage Your Organization's Projects


As the Organization Administrator, you can access and manage all projects. You can create a project, assign a project's Project Owner role to yourself, delete a project, revoke the project's deletion, permanently remove a deleted project, or postpone the project's deletion by a few days.

1. In the left navigator, click **Organization** .
2. Click the **Projects** tab.



Here are the project management actions you can perform:

Action	How To
Create a project	<p>Click + Create. See Create a Project.</p> <p>Newly created projects are recorded in the Recent Activities feed on the Activities tab.</p>
Open a project	<p>In the Name column, click the project name. See Open a Project.</p> <p>You can open a project only if you're a member or an owner. Projects where you aren't a member or an owner don't appear as links. You can't open a project that's in the deleted state.</p>
View a project's team members	<p>The Members column shows the gravatars of the project's team members. A blue circle around a user's gravatar indicates the user is assigned the project's Project Owner role. Click the gravatars to display names and email addresses of all team members.</p>
Search for projects	<p>In the search box above the projects list, enter the search term. The term is matched against your organization's project names and their descriptions, as well as their members' names, usernames and email addresses.</p>
Filter projects	<p>In Member/Owner/Favorite/All, click a filter option to enable it and filter the projects list.</p> <p>Click Owner to view projects where you're a Project Owner. Click the Member button to view projects where you're a team member (any membership including Project Owner). Click Favorite to view your favorite projects. Click All to view organization's all projects.</p> <p>To see more filter options, click Detailed Filter .</p>
Mark a project as a favorite	<p>In the Favorite column, click Favorite  to add it to your favorites list.</p>
Assign a project's Project Owner role to yourself	<p>You can assign a project's Project Owner role to yourself. This is usually helpful in a case when all project's owners are removed or have left the organization. As the Organization Administrator, you can then take the ownership of such projects and then assign the Project Owner role to another project user.</p> <p>For the project whose ownership you want to assign to yourself, click the three dots *** at the end of the row and select Assign Me as Owner.</p> <p>This action is recorded in the Recent Activities feed on the Activities tab.</p>



Action	How To
Delete a project	<p>As the Organization Administrator, you can delete any project.</p> <ol style="list-style-type: none"> For the project you want to delete, click the three dots *** at the end of the row and select Delete. In the Delete Project dialog box, click Delete. <p>The project's State is marked as Deleted - Project will be removed automatically in <i>N</i> days., but the project isn't removed immediately. In the Delete Project dialog box, read the message before you click Delete. Also, check the project's delete message to know when the project will be permanently removed. To immediately remove the project permanently, click *** and select Remove Forever.</p> <p>Deleted projects are recorded in the Recent Activities feed on the Activities tab.</p>
Postpone a deleted project's removal	<p>At times, you may be unsure if you want to revoke a project's deletion that's close to its permanent removal date. For such projects, you can choose to postpone their removal by a few days.</p> <p>For the deleted project you want to postpone, click the three dots *** at the end of the row and select Postpone Removal.</p> <p>The project's permanent removal date is rescheduled and the message is displayed at the top. In Status, check the project's delete message to know when the project will be permanently removed.</p> <p>This action is recorded in the Recent Activities feed on the Activities tab.</p>
Undelete a project	<p>When you undelete a project, you get an option to update the name and description of the project. When the deletion is revoked, the project URLs will change if a different name of the deleted project is specified.</p> <ol style="list-style-type: none"> For the deleted project you want to undelete, click the three dots *** at the end of the row and select Undelete. In the Undelete Project dialog, if necessary, update the project name in Name, and update the description in Description. You would have to change the project name if another project with the same name was created after this project was deleted. Click Undelete. <p>Undeleted projects are recorded in the Recent Activities feed on the Activities tab.</p>
Select multiple projects	<p>You can select multiple projects and perform a common action on them.</p> <ol style="list-style-type: none"> If necessary, filter and sort the projects as desired. Press the Ctrl key or the Shift key and select project rows. To select all projects, click the checkbox to the left of the Name column header. Click Update Selected and select the desired action from the menu. <p>Some options, such as Undelete, are not available in the Update Selected menu.</p>
View usage metrics of projects	<p>Next to the Detailed Filter , review the projects and the disk usage metrics of the organization.</p> <p>For each project, the Disk Usage column shows the project's disk usage. Click the number to open a pop-up with details of usage by each service in the project.</p>







Manage Your Organization's Groups





In VB Studio, you can create groups of your VB Studio organization's users or import existing Oracle Identity Cloud Service (IDCS) groups, and then add these groups to VB Studio projects. When you add a group to a project, you assign a common membership type to all group's users.

After creating a group, you can add more members to it and add the group to projects. You can also import IDCS groups to VB Studio, but can't add or remove users from the imported IDCS group. To do that, sign in to the IDCS Console. Note that the imported IDCS group is synced with IDCS every five minutes.

This table describes the actions you can perform to create and manage groups.

Action	How To
Create a VB Studio group	<ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Click + Create Group. 4. In Type, if not already selected, select the VB Studio tile. 5. In Name and Description (optional), enter the group's name and description. 6. To see all members you can add, click the members list. Users who are assigned the DEVELOPER_ADMINISTRATOR or the DEVELOPER_USER IDCS role are displayed. 7. From the users drop-down list, select users to add to the group. If you can't find a particular user, enter the user's name or username in the search box. As you type, the drop-down list displays users matching the search term. 8. Click Create.
Import an IDCS group	<ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Click + Create Group. 4. In Type, select the IDCS tile. 5. Under the filter box, select the group to import. The drop-down list displays IDCS groups that are assigned either the DEVELOPER_ADMINISTRATOR or the DEVELOPER_USER IDCS role. To select multiple groups, press the Ctrl key or the Shift key, and then select groups. If you can't find a particular group, enter the group's name in the search box. As you type, the drop-down list displays groups matching the search term. 6. Click Create. <p>From VB Studio, you can't add or remove users of an IDCS group. You can do that from the IDCS Console. See Manage Oracle Identity Cloud Service Groups.</p>

Action	How To
See a group's members	<ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate the VB Studio group and click the user gravatars in the Members column.
Add members to an existing VB Studio group	<ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate the VB Studio group and click Add Member to the Group . 4. Click the members drop-down list. All users who are assigned the DEVELOPER_ADMINISTRATOR or the DEVELOPER_USER IDCS role are displayed. 5. Select the user from the drop-down list. If you can't find the user, enter the user's name or username in the search box. As you type, users matching the search term are displayed. 6. Click Add. 7. In the Members tab, verify the added members. 8. Click Close.
Remove members from a VB Studio group	<ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate and double-click the VB Studio group. 4. In the Members tab, select the members to remove. 5. Click Remove Members. 6. Click Remove Members to confirm.
Add a group to a project	<p>To add a group to a project, you must be assigned the project's Project Owner membership.</p> <ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate the VB Studio group and click Add Group to a Project . 4. From the project drop-down list, select the project. 5. From the roles list, select the role you want to assign to the group's members. 6. Click Add. 7. Click Close.

Action	How To
Remove a group from a project	<p>To remove a group from a project, you must be assigned the project's Project Owner membership.</p> <ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate and double-click the VB Studio group. 4. In the Projects tab, select the projects to remove. 5. Click Remove Group from Projects. 6. Click Remove Group from Projects to confirm.
See the projects a group is added to	<ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate the VB Studio group and click the user gravatars in the Projects column. 4. Click Close.
Edit a group's name or description	<ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate and double-click the VB Studio group. 4. Click Actions *** and select Edit. 5. Update the group's name and description 6. Click Save. 7. Click Close.
Delete a group	<ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate the VB Studio group. 4. Click Delete. 5. Click Delete to confirm.

8

Manage Your Component Exchange

If your team develops custom components for visual applications and want the components to be available to all users in the Visual Builder Components tab, you'll need to first set up a component exchange. This chapter tells you how to set up the Component Exchange in Visual Builder.

What is a Component Exchange?

A Component Exchange is a repository of custom components that you use in your visual applications. The custom components include web components, application templates, UI patterns, and actions and are available to all VB Studio users.

To host the custom components, you can define either a VB Studio project or an external server as the Component Exchange. Any VB Studio project can be used as the Component Exchange, but you should create and set up a separate project to host components. If you have multiple VB Studio instances across multiple tenancies, you can specify an external server as the common Component Exchange for all VB Studio instances.

To use a project or a server as the Component Exchange, specify its details on the **Organization** page. When set, the VB Studio Designer automatically shows the Component Exchange's hosted components in the Components tab.

Use a Project as the VB Studio Component Exchange

1. On the Organization page, click **+ Create**.

If you've already created a project to use as Component Exchange, jump to step 10.

2. On the **Details** page, enter a unique name and description for the project.

Example:

New Project

< Back

1

Details

2

Template

3

Properties

Next >

Project Details

Name

Exchange

Description

Project to host custom components

Security

Setting a project as "Shared" makes the project code, wiki docs, tasks, and builds available to anyone inside your organization

☐ Private

☒ Shared

3. In **Security**, select **Shared**.

4. In **Preferred Language**, specify the language for the email notifications your project users will receive.

You can change the language in which the user interface appears in your user preferences.

5. Click **Next**.

6. On the **Template** page, select **Empty Project**, and click **Next**.

7. In **Wiki Markup**, select the project's wiki markup language.


Project's users use the markup language to format wiki pages and comments. If you're not sure which markup language to choose, use the default markup language. You can change the markup language later from the **Project Administration** page. See [Change a Project's Wiki Markup Language](#).

8. Click **Finish**.

After the project is provisioned, you're navigated to the **Project Home** page.

9. Add users to the project and assign them proper memberships. See [Manage Project Users and Groups](#).

Add users who would develop the components and assign them either the **Developer** or **Developer Limited** membership. Assign the **Project Owner** membership to trusted users only. Remember that a user with the **Project Owner** membership can delete the project and Git repositories.

10. In the left navigator, click **Organization** .

11. Click the **Component Exchange** tab.

12. Click **Edit**.

13. Select the **Enable organization-wide component hosting** check box.

14. Select the **In a project** option.


Administering Visual Builder Studio
G49427-01
Copyright © 2020, 2026, Oracle and/or its affiliates.

January 6, 2026
Page 2 of 4

15. In **Select Project**, select the project you created.
16. Click **Save**.

That's it. When your team members upload the components to the Component Exchange, they'll be automatically available in the Designer's Component tab.

Use an External Server as the VB Studio Component Exchange


1. In the left navigator, click **Organization** .
2. Click the **Component Exchange** tab.
3. Select the **Enable organization-wide component hosting** check box.
4. Select the **On a custom server** option.
5. In **Server URL**, enter the server's URL.
6. In **Username** and **Password**, enter the access credentials.
7. Click **Test Connection**.
8. When the connection is successful, click **Save**.

Share the Component Exchange's URL

After defining a project or an external server as the Component Exchange, share its URL with developers who'll upload components.

If you've defined an external server as the Component Exchange, share its URL as-it-is with developers.

If you've defined a project as the Component Exchange, share its URL in the `https://{hostname}/{org_id}/s/{project_id}/compcatalog/0.2.0/` format. These steps tell you how to construct the URL in the required format.

1. Open the Component Exchange project's **Project Home** page.
2. In the **Maven** section of the **Repositories** tab, click the **Project Repository's URLs** drop-down list and then click the HTTPS URL's **Copy**  icon.

Example:

Repositories Statistics Team

Git

+ Create Repository

Filter Git Repositories



All Favorites

Favorites First ▼

☆ exchange.git

Clone ▼

Maven

Project Repository

URLs ▼

NPM

Project Registry

Connect with HTTPS.

<https://myorg-vbs.developer.ocp.oc.com/profile/myorg-vb...>

Connect with Webdav.

<dav:https://myorg-vbs.developer.ocp.oc.com/profile/myor...>

3. Paste the copied URL in a text editor.
4. In the copied URL, remove the `profile/` and `maven/` elements.

For example, if the copied URL is `https://vbs-vbstest.developer.ocp.oraclecloud.com/profile/vbs-vbstest/s/vbs-vbstest_my-project_902/maven/`, then remove the elements marked in bold.

After removing the elements, the URL should be similar to this:

```
https://vbs-vbstest.developer.ocp.oraclecloud.com/vbs-vbstest/s/vbs-vbstest_my-project_902/
```

5. Append `compcatalog/0.2.0` at the end of the URL.
For example: `https://vbs-vbstest.developer.ocp.oraclecloud.com/vbs-vbstest/s/vbs-vbstest_my-project_902/compcatalog/0.2.0`
6. Share the URL with developers who'll upload components to the Component Exchange.

Workflow for Web Components

To find out how to create a web component; test it; use it in your application; build, package, and deploy it to the component exchange, see Typical Workflow for Working with Oracle JET Web Components in *Developing Applications with Oracle JET*.

9

Troubleshooting

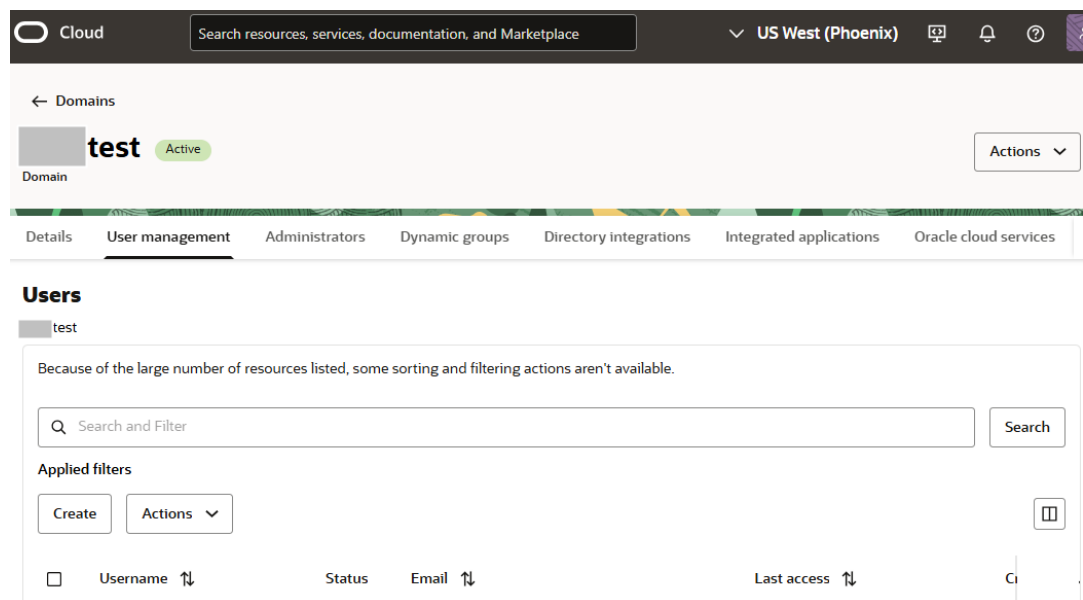
These topics cover some common issues and how to address them.

Resolve Access Issues After Switching Users to Federated IDCS

If your users are federated in OCI Identity and Access Management, previously referred to as IDCS, and they can no longer access Visual Builder Studio, you need to set the primary email address on their user account.

To set the primary email address on the user account:

1. In a web browser, go to <https://cloud.oracle.com>.
2. In the upper-left corner, click **Navigation Menu** ☰.
3. Select **Identity & Security** and then, under Identity, select **Domains**.
4. On the Domains page, click the name of the identity domain that you want to work in. You might need to change the compartment, next to **Applied filters**, to find the domain that you want.
5. After clicking the identity domain, click the **User management** tab.



6. On the Users page, search for and select a user.
7. Click **Edit user**.
8. On the Edit user page, change the email and click **Save changes**.

Unable to See the "Edit Page in Visual Builder Studio" Option in an Oracle Cloud Application

If your users don't see the **Edit Page in Visual Builder Studio** option in their Oracle Cloud Application, it could be because:

- They don't have the right privileges to access VB Studio. See [Set Up VB Studio Users](#) and [Add Users to the Project](#).
- They may not be working in an environment that has an instance of VB Studio associated with it, such as a TEST environment. See [Work With Multiple VB Studio Instances](#).
- Their Oracle Cloud Application hasn't yet adopted Oracle's new Redwood design pattern, so this page isn't extensible using VB Studio. In that case, refer them to *Oracle Applications Cloud: [Configuring and Extending Applications](#)* for instructions on how to customize their Oracle Cloud Applications with App Composer to meet their business needs.

✓ Tip

If their application page URL contains `.../faces/...`, then they can't edit this page using VB Studio.


How Do I Resolve a "You Are Not a Member of This Organization" Error?

If your users see a `You are not a member of this organization` error when trying to access VB Studio from their Oracle Cloud Application, they probably haven't been granted either the required Oracle Cloud Applications roles or VB Studio roles.

See [Set Up VB Studio Users](#).

How Do I Access OCI Identity and Access Management From the Oracle Cloud Console?

If you don't have a direct URL to OCI Identity and Access Management, formerly known as the IDCS console, you can access it from the Oracle Cloud Console instead.

1. In a web browser, go to <https://cloud.oracle.com>.
2. On the Sign-In page, in **Cloud Account Name**, enter your Oracle Cloud account or tenancy name and click **Next**.
3. On the Single Sign-On (SSO) panel, if required, select your identity domain and click **Continue**.
4. Sign in with the user credentials that you received when you set up your account.
The Oracle Cloud Console, also called the OCI console, opens.
5. In the upper-left corner, click **Navigation Menu** .
6. Select **Identity & Security** and then, under Identity, select **Domains**.

7. From here, you can now assign VB Studio roles manually.

To do this, start with step 7 in [Assign VB Studio Roles in OCI Identity and Access Management](#).

Resolve Access Issues After Allow-Listing Your VB Studio IP Address



If you've added your VB Studio infrastructure IP address to the list of allowed IP addresses in your Web Application Firewall (WAF) and/or Location Based Access Control (LBAC) configuration and are now experiencing access issues, try one of these possible solutions:

- If you have LBAC enabled, but you haven't added your VB Studio IP address to the LBAC allow list, your users may receive error messages that say: "We are not able to determine the status of environment <environment name>. We couldn't retrieve the list of sandboxes from <instance name> due to :Forbidden."
- If you see an "Unauthorized" status in the Service Instances tab on VB Studio's Environments page, the issue *could be* related to LBAC or WAF. However, it could also be an issue with your credentials not having access to the FA status endpoints. Only certain endpoints, not all of them, are protected by the LBAC configuration.
- Ask the Oracle Cloud Applications administrator to enable Cross-Origin Resource Sharing (CORS) by adding the VB Studio's root URL to the Oracle Cloud Application's ORA_CORS_ORIGINS profile option. This option is used to specify which domains can talk to each other.

Note

Production to Test (P2T) and other operations that replace the database may cause the profile option value to change. If you perform any operation that replaces the database, you must repeat these steps.

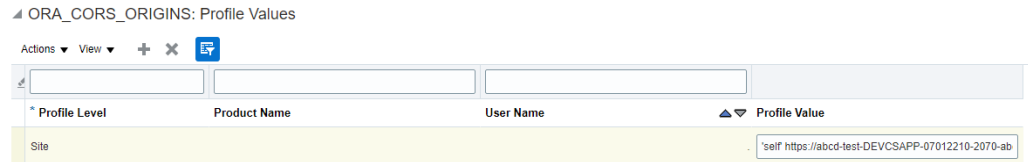
To enable CORS, the Oracle Cloud Applications admin must:

1. Open VB Studio from the Oracle Cloud Applications instance and copy the VB Studio URL from the browser's address bar.
2. In the Oracle Cloud Applications instance, click **Navigator** .
3. Under **Others**, click **Setup and Maintenance**.
4. On the right side of the page, click **Tasks**,  then click **Search**.
5. In the search box, enter `Manage Administrator Profile Values` and click the Search icon.
6. In the search results, click **Manage Administrator Profile Values**.
7. In Profile Options Code, enter `ORA_CORS_ORIGINS`, then click Search.
8. Under `ORA_CORS_ORIGINS: Profile Values`, click **New**.
9. In Profile Level, select **Site**.
10. In Profile Value, enter `'self'` (including the single quotes), followed by VB Studio's root URL. For example: `'self' https://abcd-test-DEVCSAPP-07012210-2070-abcd.developer.ocp.oraclecloud.com`

Note

Don't include port numbers, wild cards, or extraneous characters after the root. For example, if the VB Studio's full URL is `https://abcd-test-DEVCSAPP-07012210-2070-abcd.developer.ocp.oraclecloud.com:443/abcd-test-devcsapp-07012210-2070-abcd`, the root is simply `https://abcd-test-DEVCSAPP-07012210-2070-abcd.developer.ocp.oraclecloud.com`.

Your screen should look something like this:

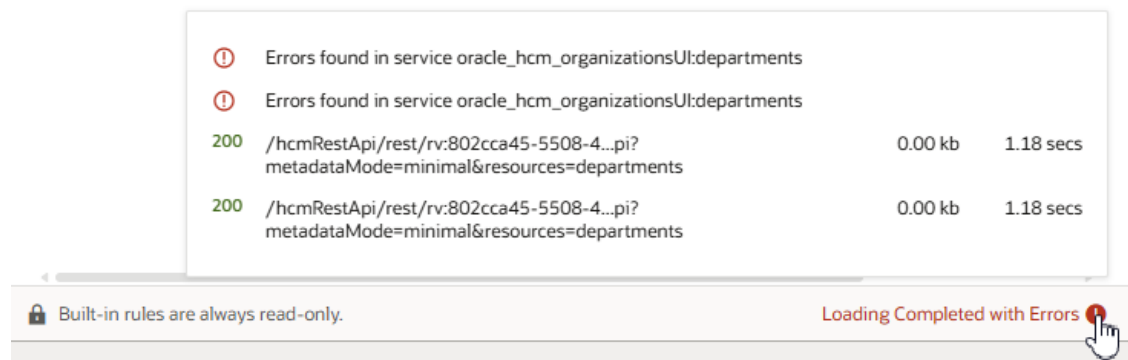


11. Before you finish the CORS setup, you should check the CORS headers to be added. Although all of the other CORS options should be configured by default, in every pod, some of these settings may be missing or may have been erased inadvertently. See [Configure Cross-Origin Resource Sharing](#) for additional information about all the settings that can be configured, including a list of acceptable values for each option.
 12. Click **Save** and **Close**.
- If you are still experiencing problems, contact your Oracle Support representative and file an SR so Oracle can assist you in setting up access to VB Studio.

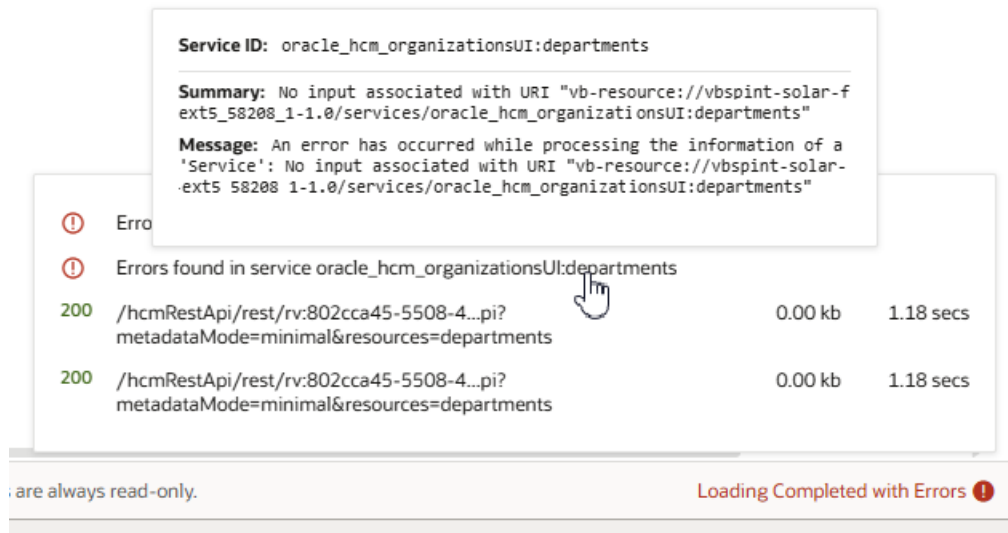
Resolve the "Loading Completed with Errors" Message

When you open a page in your workspace, VB Studio loads all the services that furnish the page's data in the background. If an error is encountered during the loading process, you'll see the message "Loading completed with errors" at the bottom of the page.

If you see this message, click "!" in the status bar to display more detailed information about the services affected:



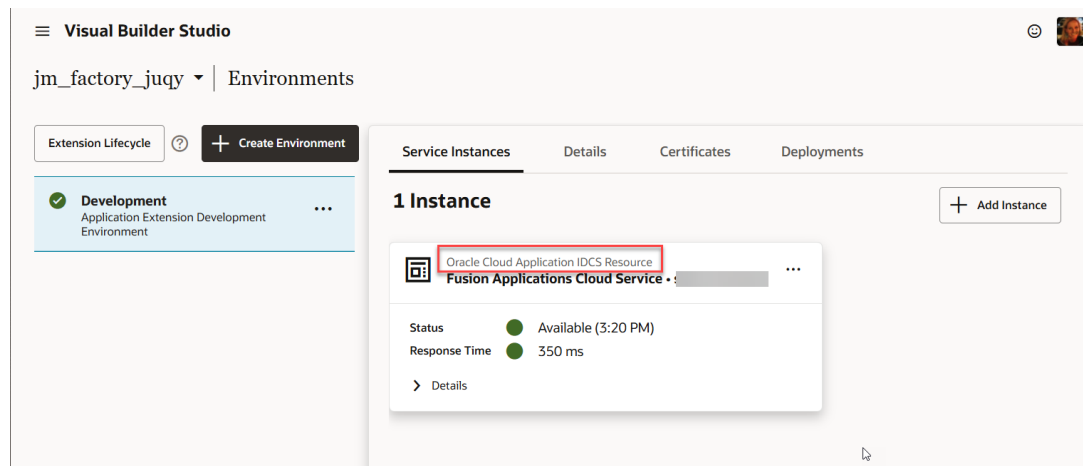
The list might also include information about data fetches that are unusually large, which might cause the editor to load slowly. By hovering over a particular error in the list, you can drill down even further:



It's important to note that even if there were problems retrieving data from one or more services, the Business Rules editor will still show data from the services that are loaded successfully. In other words, your page might be missing regions or fields from some services, while still displaying content from healthy services.

If a service can't be loaded, it's usually because there are issues either in your environment or in CORS. To troubleshoot:

1. Check the environment associated with your workspace. (If you're in the Designer, click the arrow at the top left, then click **Environments** in the left navigator.) Make sure your status indicator is green, and that your Oracle Cloud Applications instance was added as an IDCS resource (as opposed to OAuth or Basic Auth):



If both conditions are true, continue with the next step. If not:

- If your status is anything other than green AND your instance is an IDCS resource, there may be a more complex underlying issue. File an SR for further assistance.
 - If the instance was added with OAuth or Basic Auth, it's best to create a new environment using the Identity Domain option, then switch your workspace's environment to the new one.
2. Make sure there aren't any CORS issues:

- a. Open Chrome Developer Tools, then click the **Network** tab.
 - b. Open your workspace.
 - c. In the Network tab, look for the REST call: GET fscmRestApi/applcoreApi/v1/applicationManifest/faStandalone?appRelativePath=fscmUI/redwood. You might also see an OPTIONS call for the same endpoint, which you can ignore.
 - d. If the call is successful, move to step f. If not, continue with the next step.
 - e. If the call is not successful:
 - i. Click **Console** in Chrome Developer Tools and find the full description of the CORS "Access to fetch" error:
 - If you see: **Access to fetch at '<Oracle Fusion Cloud Applications endpoint>' from origin '<Visual Builder Studio>' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource. If an opaque response serves your needs, set the request's mode to 'no-cors' to fetch the resource with CORS disabled.**, that means the Visual Builder Studio domain has not been added to Oracle Cloud Applications' CORS setting. See the row "Access-Control-Allow-Origin" in [Configure Cross-Origin Resource Sharing](#) for guidance. Also check to make sure there aren't any typos.
 - If you see: **Access to fetch at '<Oracle Fusion Cloud Applications endpoint>' from origin '<Visual Builder Studio>' has been blocked by CORS policy: Request header field <some-specific-header> is not allowed by Access-Control-Allow-Headers in preflight response.**, that means a specific header field hasn't been added to Oracle Fusion Cloud Applications' CORS Allow Headers profile. See the row "Access-Control-Allow-Headers" in [Configure Cross-Origin Resource Sharing](#) for guidance. Also check to make sure there aren't any typos.
 - ii. Make sure there aren't any typos in the CORS profile values, as header fields can't contain spaces. For example, in CORS_ACCESS_CONTROL_ALLOW_HEADERS, **X-HTTP-Method-Override**, and **X-Requested-By** is incorrect, due to the space before and after "and". Instead, you would specify **,X-Requested-By**.
 - f. If there aren't any CORS errors but fscmRestApi/applcoreApi/v1/applicationManifest/faStandalone?appRelativePath=fscmUI/redwood fails with a 401, there may be a problem with your wiring to IDCS. Log an SR for further assistance.
3. If there aren't any problems with fscmRestApi/applcoreApi/v1/applicationManifest/faStandalone?appRelativePath=fscmUI/redwood, but there is a problem with one particular REST describe call, log an SR on the Oracle Cloud Application you're using for further assistance.

Appendices

These appendices provide information about migrating Oracle Cloud Applications configurations, and a list of all IP addresses by region and data center.

Topics:

- [Migrate Oracle Cloud Applications Configurations](#)
- [Public IP Addresses for VB Studio](#)

A

Migrate Oracle Cloud Applications Configurations

Configurations are changes saved in an Oracle Cloud Applications sandbox. A configuration set is an archive file that contains all your configurations across all product families. When you migrate configurations, you export a configuration set from the source Oracle Cloud Applications instance (such as your development or test instance) and import it to a target Oracle Cloud Applications instance (such as your production instance).

If you're new to migration, see [Configuration Life Cycle](#) and [Migration](#) in *Configuring and Extending Applications* to learn about configuration migration.

You can export and import configurations manually from the Oracle Cloud Applications instance's Migration page, or configure Configuration Set Migration (CSM) build jobs and pipelines in VB Studio to automate the export and import process. To learn more about VB Studio build jobs and pipelines, see [What Are Jobs and Builds?](#) and [What is a Pipeline?](#).

This document describes how to configure VB Studio build jobs and pipelines to migrate configurations from the source development instance to a sandbox on the target production instance. This document also describes how to deploy your extension to the production instance's sandbox so you can test it with the migrated configurations.

Before You Configure CSM Build Jobs and Pipelines

Here are some best practices to follow before you configure and run CSM build jobs and pipelines:

General Considerations

- Synchronize the source and target environments to ensure successful configuration migration:
 - Always ensure that the source and target instances are of the same release, with the same standard and one-off patches applied to both instances.
 - In the source instance, register the target instance. See [Overview of Migration](#) in *Configuring and Extending Applications*.
 - Make sure that both the source and target instances are synchronized. To sync them, make application changes to the source instance, test the changes, then migrate them to the target instance.
 - Create all configurations in the source instance, then migrate them to the target instance. Never create configurations directly on the target instance.
 - Test all application changes and configurations that were made in the source instance before starting with the migration process.
 - Ensure that the target instance doesn't have any re-keyed configurations or ones that come from another source. If the source and target instances differ, re-provision the target instance from same common ancestor that provisioned the source instance.

Note

If these guidelines aren't closely adhered to, CSM Migration-related issues and Import/Export-related errors will be encountered. Also, in some instances, your source and target environments will be out of sync and you have to perform a P2T (Production-to-Test) operation to restore order to your environments.

- Always create new and unverified configurations in a sandbox. The migration process will only migrate configurations that have been created in a published sandbox.
 - If you make configuration changes on a sandbox and don't publish to mainline, the configuration changes won't be included in the migration package.
- You should use CSM to migrate lookups and menu configurations, instead of existing Functional Setup Manager (FSM) tasks.
- Functional security artifacts for standard objects are not migrated.
 - If you migrate any functional security associated with roles in the source to a target instance, and the corresponding role doesn't exist in the target instance, an error will occur on import.
 - To avoid these errors, you can selectively create these roles in the target instance when you import your migration set. See [Export and Import of HCM Custom Roles and Security Profiles](#) in *Securing Applications*.
- After you apply configurations, Oracle Cloud Application end users won't see any changes to the user interface until they log out and log back in.

Delta Migration

- A normal migration flow migrates everything. Because this type of migration contains everything, it can be time-consuming. Think of it as an all-or-nothing proposition. A Delta Migration, on the other hand, only migrates what you've changed since the last time you migrated. By only migrating the latest set of changes, you can potentially save considerable execution time during the migration.
- Note that you can't selectively pick and choose which of your latest changes should get migrated. Delta migration takes all of the changes since last time. Yes, you can select which modules you want to migrate, but you have to take all changes within the module.
- To perform a delta migration, use the sandbox feature in both the source and target instances. See [Configuration Life Cycle](#) in *Configuring and Extending Applications*.

Roles Migration

- If you need to review and create roles on the target instance, don't run the migration pipeline. Run the export configurations job manually, create the required roles on the target instance, and then manually run the import configurations job. See [Migrate Your Configurations](#).
- Get the access credentials of an Oracle Cloud Applications user who can connect and deploy to the target instance.

Extensions Using Visual Builder Studio

- In VB Studio, [create a project for extensions](#).
- In your VB Studio project, make sure you have environments that point to both source and target Oracle Cloud Applications instances. See [Add an Oracle Cloud Application's Instance to an Environment](#) in *Administering Visual Builder Studio*.

- If you want to import the configuration set into multiple target instances, create an import job for each target instance.
- When build jobs are running, don't export configurations manually from the source instance or make security changes.
- After you have triggered the migration pipeline, do not cancel it or retry it if it seems slow. Check the build job's log to monitor the migration status.

To deploy your extension to the Oracle Cloud Applications production instance's sandbox, you'll create these jobs and pipelines:

- A packaging job that packages the extension for the Oracle Cloud Applications production instance's sandbox.
- A deployment job that deploys the extension to the Oracle Cloud Applications production instance's sandbox.

Here are some tips that help you speed up your migration process:

- Perform a delta migration first, as it reduces the migration set size and reduces the time to migrate configurations.
To do that, configure your export configurations job to move only the latest and incremental changes.
- To reduce failures, include only the required optional modules in the configuration data set and exclude the modules that have no application changes.
The configuration set migration process involves migrating components, such as CRM, ADF, BI, SOA, as well as workspaces. If your base application has CRM and BI components, for example, but you don't have any BI or Analytics related changes, exclude the BI component from the configuration set when you configure your export configurations job.

Create and Configure Build Jobs to Migrate Your Oracle Cloud Application's Configurations



You need to set up build jobs to export and import configurations. This topic explains how to do that.

To do this:	See this:
Generate the configuration set from the source Oracle Cloud Applications instance and create the configuration set ID. The configuration set ID is required by other CSM build jobs.	Configure a Job to Export Configuration
Import the generated configuration set from source Oracle Cloud Applications instance to the target Oracle Cloud Applications instance	Configure a Job to Import Configuration
Delete the imported configuration set	Configure a Job to Cancel the Configuration Migration
Apply and save the imported configuration set on the target instance	Configure a Job to Apply and Save the Configuration Migration
Restore the target Oracle Cloud Applications instance to the state before you applied the configuration set	Configure a Job to Restore Configuration Migration

After creating jobs, [create and configure a CSM pipeline](#) to run the jobs in a sequence.

Configure a Job to Export Configuration


Configure a job to generate the outbound configuration set from the source Oracle Cloud Applications instance. This job will also generate a unique configuration set ID that is required by other CSM build jobs.

1. Open your VB Studio project.
2. In the left navigator, click **Builds** .
3. In the **Jobs** tab, click **+ Create Job**.
4. In the New Job dialog box, in **Name**, enter a unique name.
5. In **Description**, enter the job's description.
6. In **Template**, select the **System Default OL7 for Visual Builder** template.
7. Click **Create**.
8. (Optional) Click **Configure** .
9. Click the **Steps** tab.
10. From **Add Step**, select **Configuration Set Migration**, and then select **Export Data**.
11. In **Source Instance**, select the environment that points to the Oracle Cloud Application's source instance.
12. (Optional) In **ID Parameter Name**, if necessary, change the parameter name for the configuration set. By default, it's set to `CONFIGURATION_SET_ID`.

When a build runs, the job generates a unique configuration set ID, assigns it to the parameter name defined in **ID Parameter Name**, and then passes the parameter to the downstream jobs when the CSM jobs run in a pipeline. The downstream jobs can access the parameter's value by calling it in the `${<PARAMETER_NAME>}` or `${<PARAMETER_NAME>}` format.

If you change the parameter name, note it down. You'll need it when you configure other CSM jobs.

13. In **CSM/Sandbox Name**, enter a name for the configuration set.
When you import configurations later, the target Oracle Cloud Applications instance will create an internal sandbox with this name.
14. (Optional) In **Description**, enter a description for the configuration set.
15. In **Username** and **Password**, enter the credentials of an IDCS user who is also an Oracle Cloud Applications user.
16. (Optional) Under **Optional Modules**, specify the optional modules you want to export. By default, all modules are included.

To specify a few modules only, deselect the **Include all modules** check box. In the Search Optional Modules dialog box that opens, enter the credentials of an IDCS user who is also an Oracle Cloud Applications user and click **Search Optional Modules** . Select the check boxes of optional modules you want to include and click **OK**.

17. (Optional) To export configurations if the target isn't registered on the source Oracle Cloud Applications instance, select the **Create migration set even if target isn't registered** check box.
18. (Optional) By default, the migration will move the latest application changes. To move all application changes, select the **Move all changes** check box.

19. Click **Save**.

When you run a build of this job, open the build's log to note the configuration set ID. Here's an example:

```
[2020-07-01 08:00:00] Build scheduled. Build started by pipeline CSM
[2020-07-01 08:00:00] Build task id: alb2c123-abc1-12d3-6789-5bc4d76531f2
.
.
.
[2020-07-01 08:01:35] Status: Export Started
[2020-07-01 08:01:35] Configuration Migration Set ID is 1239873457659876
[2020-07-01 08:01:39] IN_PROGRESS...
[2020-07-01 08:30:02] SUCCESS...
[2020-07-01 08:35:10] CREATED...
.
.
.
[2020-07-01 09:30:10] 2020-07-01 08:36:02.990: The outbound migration of CRM
common objects completed successfully.
[2020-07-01 09:35:10] === End CSM Export Execution ===
Executor log size 4.6 KB (4,558)
[2020-07-01 09:35:12]
[2020-07-01 09:35:12] Build completed.
[2020-07-01 09:35:12] Status: DONE Result: SUCCESSFUL Duration: 35 min
```

After a successful build, if you want to view details of the exported migration set or download it, open the source Oracle Cloud Applications instance's Migration page and click the **Outgoing** tab. You don't need to download it if you plan to use the import configuration job to import configurations.

Here's an example of the Migration page:

Migration ⓘ

Incoming
Apply Migration Set

Outgoing
Create Migration Set

Environment Info
Target Instance Set

Outgoing Migration Sets

CSM.Release1.0.1

Your migration set is ready, and available for import in the target environment. [Delete](#) [Content Read Me](#)

Created By alex.cloud Release Version 11.1.13.0.0


Created On 2020-07-07 09:11:53 UTC Modules

Configure a Job to Import Configuration

Configure a job to import the configuration set to the target Oracle Cloud Applications instance.

When a build of this job runs, it requires the unique configuration set ID generated by the export configuration job. If you're not planning to run this job in a pipeline with the export configuration job, you'll need to provide the configuration set ID manually.

1. In the left navigator, click **Builds** .

2. In the **Jobs** tab, click the export configuration job.
3. Click **Configure**.
4. Click the **Steps** tab and note the **Configuration Set ID** field's parameter name.
5. Click **Cancel**.
6. Return to the **Jobs** tab and click **+ Create Job**.
7. In the New Job dialog box, in **Name**, enter a unique name.
8. In **Description**, enter the job's description.
9. In **Template**, select the **System Default OL7 for Visual Builder** template.
10. Click **Create**.
11. (Optional) Click **Configure** .
12. If you plan to run this job manually and not in a pipeline with the export configuration job, add a build parameter to accept the configuration set ID at runtime. Ignore this step if you plan to run the job in a pipeline with the export configuration job.
 - a. Click the **Parameters** tab.
 - b. From **Add Parameter**, select **String Parameter**.
 - c. In **Name**, enter the Configuration Set ID parameter name you noted in Step 4.
If you didn't change the default parameter name, enter `CONFIGURATION_SET_ID`.
 - d. (Optional) In **Description**, enter the parameter's description. For example, you could add this: Enter the configuration ID copied from the CSM Export job's build log.
13. Click the **Steps** tab.
14. From **Add Step**, select **Configuration Set Migration**, and then select **Import Data**.
15. In **Target Instance**, select the environment that points to the Oracle Cloud Application's target instance.
16. In **Configuration Set ID**, enter the Configuration Set ID parameter name you copied in Step 4 in the `$<PARAMETER_NAME>` or `${<PARAMETER_NAME>}` format. By default, it's set to `${CONFIGURATION_SET_ID}`.

When a build of this job runs, the job will get the unique configuration set ID generated by the export configuration job. If no ID is found, the build will fail.
17. In **Username** and **Password**, enter the credentials of an IDCS user who is also an Oracle Cloud Applications user.
18. (Optional) To skip active sandboxes check, select the **Ignore unpublished sandboxes with Application Composer** check box.
19. Click **Save**.

When you import configurations, the target Oracle Cloud Applications instance creates an internal sandbox to test the migrated configurations. The build job gets the sandbox ID from the target Oracle Cloud Applications instance and displays it in the build's log. Here's an example:

```
[2020-07-01 09:36:13] Build scheduled. Build started by pipeline CSM
[Upstream Job: CSM-Release1.0.1 Build: #1]
[2020-07-01 09:36:13] Build task id: d2c1b321-cba8-45e5-2233-2dd3a25678a1
.
.
```

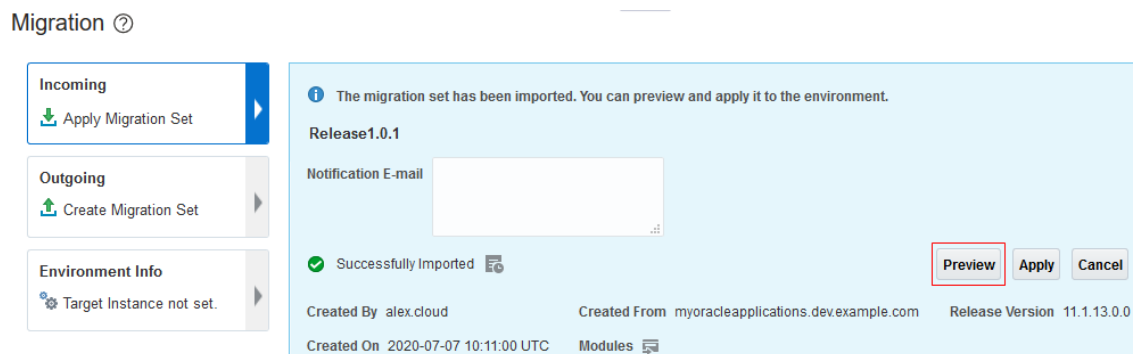
```

.
[2020-07-01 09:36:32] Configuration Set ID = 1239873457659876
[2020-07-01 09:36:49] Status: Import started
.
.
.
[2020-07-01 09:42:32] IMPORT_IN_PROGRESS...
[2020-07-01 09:42:39] CSM Sandbox Id: abc01def2fe345fedcba12345ab01234
[2020-07-01 10:01:46] IMPORT_SUCCESS...
.
.
.
[2020-07-01 10:07:44] === End CSM Import Execution ===
Executor log size 7.6 KB (7,616)
[2020-07-01 10:07:46]
[2020-07-01 10:07:46] Build completed.
[2020-07-01 10:07:46] Status: DONE Result: SUCCESSFUL Duration: 31 min

```

After a successful build, to view details of the imported migration set, open the target Oracle Cloud Applications instance's Migration page and click the **Incoming** tab.

Here's an example:





To test the imported configurations on the newly created sandbox, click **Preview**.

Configure a Job to Cancel the Configuration Migration

If for some reason your import configuration job fails, configure a job to cancel the migration process.

This job requires the unique configuration set ID generated by the export configuration job. If you're not planning to run this job in a pipeline with the export configuration job, you'll need to provide the configuration set ID manually.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click the export configuration job.
3. Click **Configure**.
4. Click the **Steps** tab and note the **Configuration Set ID** field's parameter name.
5. Click **Cancel**.


6. Return to the **Jobs** tab and click **+ Create Job**.
7. In the New Job dialog box, in **Name**, enter a unique name.
8. In **Description**, enter the job's description.
9. In **Template**, select the **System Default OL7 for Visual Builder** template.
10. Click **Create**.
11. (Optional) Click **Configure** .
12. If you plan to run this job manually and not in a pipeline with the export configuration job, add a build parameter to accept the configuration set ID at runtime. Ignore this step if this job runs in a pipeline with the export configuration job.
 - a. Click the **Parameters** tab.
 - b. From **Add Parameter**, select **String Parameter**.
 - c. In **Name**, enter the Configuration Set ID parameter name you copied in Step 4.
If you didn't change the default parameter name, enter `CONFIGURATION_SET_ID`.
 - d. (Optional) In **Description**, enter the parameter's description. For example: Enter the configuration ID copied from the CSM Export job's build log.
13. Click the **Steps** tab.
14. From **Add Step**, select **Configuration Set Migration**, and then select **Cancel Migration**.
15. In **Target Instance**, select the environment that points to the Oracle Cloud Application's target instance.
16. In **Configuration Set ID**, enter the Configuration Set ID parameter name you noted in Step 4 in the `$<PARAMETER_NAME>` or `${<PARAMETER_NAME>}` format. By default, it's set to `{CONFIGURATION_SET_ID}`.


When a build of this job runs, the job gets the unique configuration set ID generated by the export configuration job. If no ID is found, it fails.
17. In **Username** and **Password**, enter the credentials of an IDCS user's credentials who is also an Oracle Cloud Applications user.
18. Click **Save**.

Configure a Job to Apply and Save the Configuration Migration

After your import job is successful and you have manually tested the changes on the target Oracle Cloud Applications instance, configure a migration job to save the configuration set.

This job requires the unique configuration set ID generated by the export configuration job. If you don't know the ID while configuring this job, use a build parameter to accept the configuration set ID at runtime. Don't add this job in the pipeline with the import configuration job, but run this job manually after you've verified all imported configurations on the target Oracle Cloud Applications instance.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click the export configuration job.
3. Click **Configure**.
4. Click the **Steps** tab and note the **Configuration Set ID** field's parameter name.
5. Click **Cancel**.


6. Return to the **Jobs** tab and click **+ Create Job**.
7. In the New Job dialog box, in **Name**, enter a unique name.
8. In **Description**, enter the job's description.
9. In **Template**, select the **System Default OL7 for Visual Builder** template.
10. Click **Create**.
11. (Optional) Click **Configure** .
12. Click the **Parameters** tab.
13. From **Add Parameter**, select **String Parameter**.
14. In **Name**, enter the Configuration Set ID parameter name you copied in Step 4.
If you didn't change the default parameter name, enter `CONFIGURATION_SET_ID`.
15. (Optional) In **Description**, enter the parameter's description. For example: Enter the configuration ID copied from the CSM Export job's build log.
16. Click the **Steps** tab.
17. From **Add Step**, select **Configuration Set Migration**, and then select **Apply Migration**.
18. In **Target Instance**, select the environment that points to the Oracle Cloud Application's target instance.
19. In **Configuration Set ID**, enter the Configuration Set ID parameter name you copied in Step 4 in the `<PARAMETER_NAME>` or `{<PARAMETER_NAME>}` format. By default, it's set to `{CONFIGURATION_SET_ID}`.


When a build of this job runs, the job gets the unique configuration set ID generated by the export configuration job. If no ID is found, it fails.
20. In **Username** and **Password**, enter the credentials of an IDCS user's credentials who is also an Oracle Cloud Applications user.
21. Click **Save**.

Configure a Job to Restore Configuration Migration

Configure a job to restore the target Oracle Cloud Applications instance to the state before the configuration set was applied.

This job requires the unique configuration set ID generated by the export configuration job. If you don't know the ID while configuring this job, use a build parameter to accept the configuration set ID at runtime. Don't add this job in the pipeline with the import configuration job, but run this job manually after you've verified all imported configurations on the target Oracle Cloud Applications instance.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click the export configuration job.
3. Click **Configure**.
4. Click the **Steps** tab and note the **Configuration Set ID** field's parameter name.
5. Click **Cancel**.
6. Return to the **Jobs** tab and click **+ Create Job**.
7. In the New Job dialog box, in **Name**, enter a unique name.
8. In **Description**, enter the job's description.


9. In **Template**, select the **System Default OL7 for Visual Builder** template.
10. Click **Create**.
11. (Optional) Click **Configure** .
12. Click the **Parameters** tab.
13. From **Add Parameter**, select **String Parameter**.
14. In **Name**, enter the Configuration Set ID parameter name you copied in Step 4.
If you didn't change the default parameter name, enter `CONFIGURATION_SET_ID`.
15. (Optional) In **Description**, enter the parameter's description. For example: Enter the configuration ID copied from the CSM Export job's build log.
16. Click the **Steps** tab.
17. From **Add Step**, select **Configuration Set Migration**, and then select **Restore Migration**.
18. In **Target Instance**, select the environment that points to the Oracle Cloud Application's target instance.
19. In **Configuration Set ID**, enter the Configuration Set ID parameter name you copied in Step 4 in the `<PARAMETER_NAME>` or `{<PARAMETER_NAME>}` format. By default, it's set to `{CONFIGURATION_SET_ID}`.

When a build of this job runs, the job gets the unique configuration set ID generated by the export configuration job. If no ID is found, it fails.
20. In **Username** and **Password**, enter the credentials of an IDCS user's credentials who is also an Oracle Cloud Applications user.
21. Click **Save**.

Create and Configure the CSM Pipeline

To ensure the configuration jobs runs automatically in the correct order, create a pipeline and set the dependency.

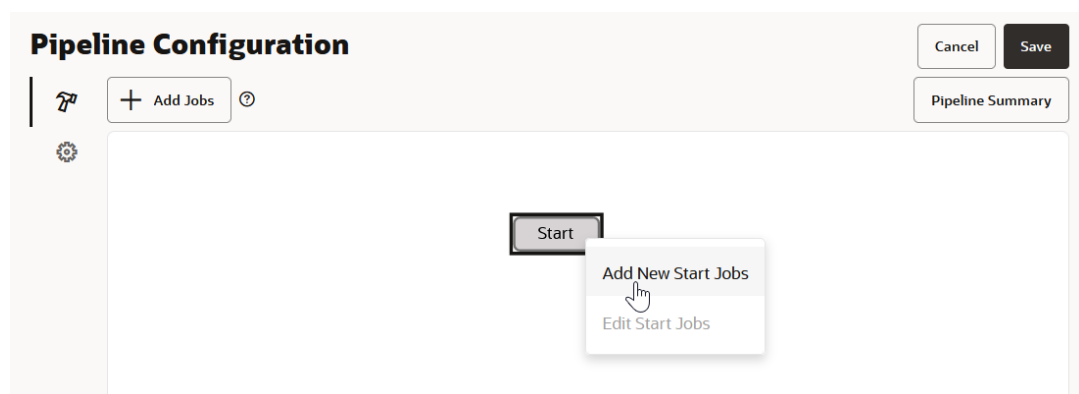
When you define the pipeline, make sure that the import configuration job triggers after the export configuration job is successful. If the import configuration job fails, run the cancel configuration migration job. Don't add the apply configuration job to the pipeline. You should run it manually after verifying your changes on the target Oracle Cloud Applications instance.

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. Click **+ Create Pipeline**.
4. In the Create Pipeline dialog box, in **Name** and **Description**, enter a unique name and description.
5. Click **Create**.

The Pipeline Configuration window is displayed with the Start node in the center.

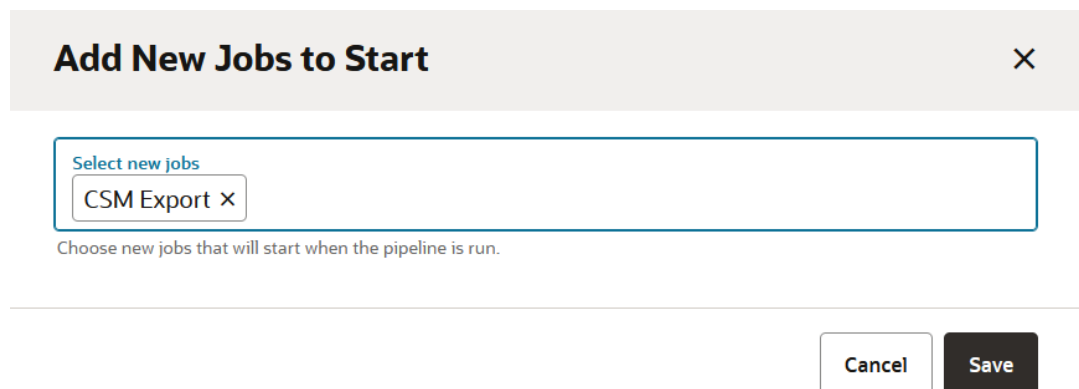


6. Right-click the Start node and select **Add New Start Jobs**.



The **Add New Jobs to Start** dialog box is displayed.

7. Click in the **Select new jobs** area to choose new jobs that will start when the pipeline is run.
- Select one or more jobs from the list displayed.

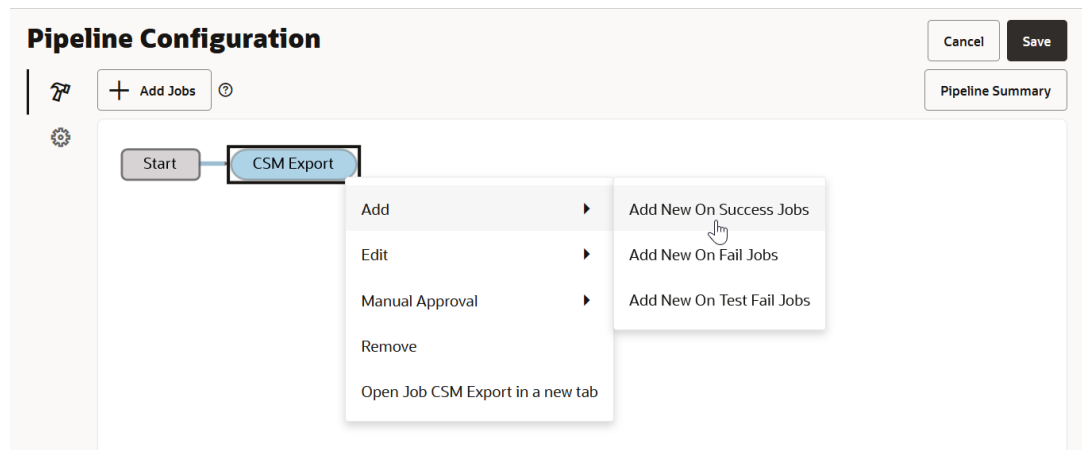


8. Click **Save**.

The Pipeline Configuration window displays the Start node with a blue line that ends in an arrow connecting to the node that represents the job you just added.

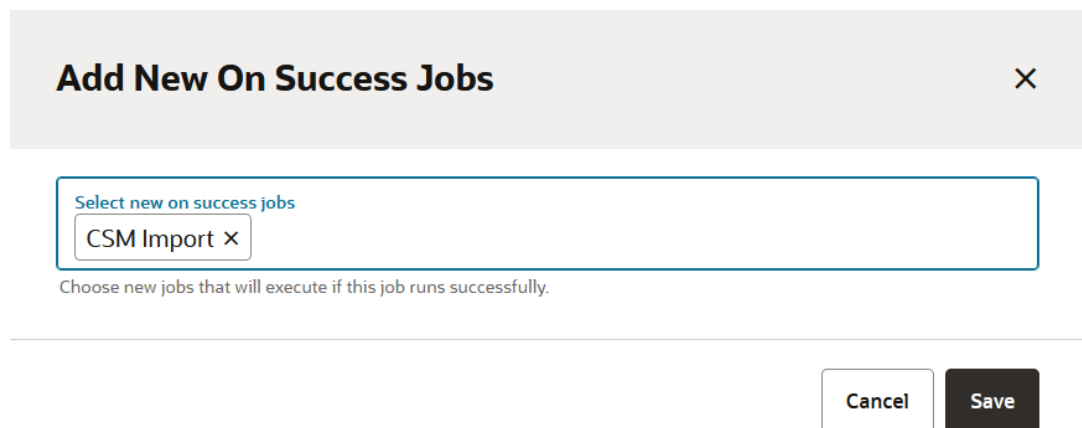


9. Right click the node you just added and select **Add** then select **Add New On Success Jobs**.



The **Add New On Success Jobs** dialog box is displayed.

10. Click in the **Select new on success jobs** area to choose new jobs that will execute if this job runs successfully.
Select one or more jobs from the list displayed.

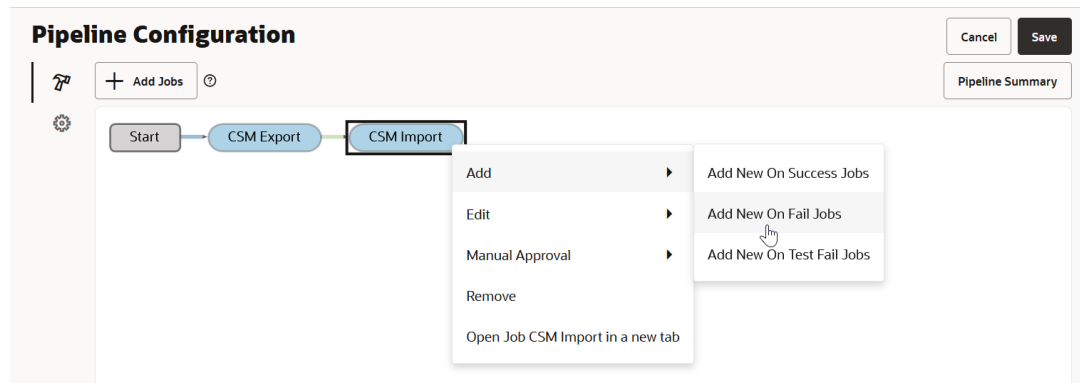


11. Click **Save**.

The Pipeline Configuration windows displays the Start node, the node with the build job you previously selected in step 7, and the node with the build job you just selected, all connected by lines that end in an arrow.



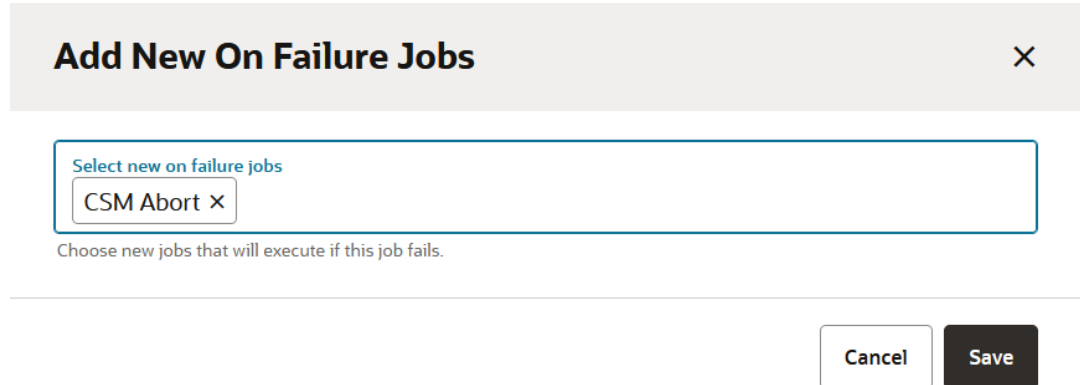
12. Right click the node you just added and select **Add** then select **Add New On Fail Jobs**.



The **Add New On Failure Jobs** dialog box is displayed.

13. Click in the **Select new on failure jobs** area to choose new jobs that will execute if this job's run fails.

Select one or more jobs from the list displayed.



14. Click **Save**.

The Pipeline Configuration windows displays the Start node, the node with the build job you previously selected in step 7, the node with the build job you previously selected in step 10, and the node with the build job you just selected, all connected by lines that end in an arrow.



15. Click the Pipeline Summary button to display a summary screen that provides many details about the pipeline configuration, similar to the example shown here.

Pipeline Configuration Summary

- This pipeline has 3 jobs.
- The following jobs will start when the pipeline is run: CSM Export
- If job CSM Export is successful, jobs CSM Import will build.
- If job CSM Import has failed, jobs CSM Abort will build.
- After job CSM Abort is built, it will not start any additional jobs.

Close

Create and Configure Build Jobs to Deploy the App Extension to the Sandbox



You need to set up packaging and deployment jobs to deploy app extensions to your Oracle Cloud Application's production instance's sandbox. This topic explains how to do that.

To do this:	See this:
Package extension for the Oracle Cloud Application's production sandbox	Create the Production Sandbox Packaging Job
Deploy the packaged extension artifact to the Oracle Cloud Application's production instance's sandbox.	Create the Production Deployment Build Job
Configure the production sandbox pipeline	Create and Configure the Production Sandbox Pipeline

Create the Production Sandbox Packaging Job

The sandbox packaging job generates an extension artifact that's ready to deploy to the Oracle Cloud Applications production instance's sandbox. Configure this job only if you want to deploy the extension to the production instance's sandbox.

To package extension for the sandbox, you'll create a build parameter that accepts the sandbox name when the build runs.

1. In the left navigator, click **Builds** .
2. Click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select the **System Default OL7 for Visual Builder** template.
6. Click **Create**.
7. Click **Configure** .

8. Click the **Git** tab.
9. From the **Add Git** list, select **Git**.
10. In **Repository**, select the Git repository. In **Branch or Tag**, select the production branch.
11. Click the **Parameters** tab.
12. From **Add Parameter**, select **String Parameter**.
13. In **Name**, enter a parameter name. You'll use this name to accept the sandbox's name when this job's build runs.

Here's an example:

The screenshot shows the 'Configure Parameters' dialog in Oracle Visual Builder Studio. The 'Parameters' tab is active. A 'String Parameter' is being configured. The 'Name' field contains 'Sandbox'. The 'Default Value' field is empty. The 'Description' field contains 'Enter the sandbox name'.

14. (Optional) In **Description**, enter the parameter's description. For example: Enter the sandbox name.
15. Click the **Steps** tab.
16. From **Add Step**, select **Application Extension**, and then select **Package**.
17. By default, the packaging step minifies the application's source code before running the build. If you don't want to minify the source files, deselect the **Optimize extension** check box.

Minification is a process to remove the unnecessary characters (such as blank spaces, new lines, and comments) from the source code and reduce the size of the files, making the transfer of files consume less bandwidth and storage.

Note

When you deselect the **Optimize extension** check box, this warning is displayed: **Optimization not selected. Packaging without optimization can cause performance problems, so avoid doing so unless you're debugging or troubleshooting.**

18. (Optional) If you want to change the artifact file's name, in **Build Artifact**, enter the new name. The best practice is to keep the default, `extension.vx`, but you can change it.
If you change it, you must also use the new name in the **Files to archive** field (step 22) as well as in the deployment job's **Build Artifact** field. See [Create the Production Sandbox Deployment Build Job](#).
19. In **Extension Version**, enter the sandbox parameter name you added in step 13. The name must be in the `$<PARAMETER_NAME>` or the `${<PARAMETER_NAME>}` format.

Example:



The screenshot shows the 'Configure Steps' dialog with the 'Steps' tab active. It displays a section for 'Application Extension Packaging' with two input fields: 'Build Artifact' (containing 'extension.vx') and 'Extension Version' (containing '\$Sandbox'). The 'Extension Version' field is highlighted with a red rectangular box.

20. Click the **After Build** tab.
21. From **Add After Build Action**, select **Artifact Archiver**.
22. In **Files to archive**, enter the build artifact's name, `extension.vx`, or the new artifact name if you're not using the default. You can also use wild characters. For example, `*.vx`.
23. If you want to discard the build's old artifacts, click **Settings** . In the **General** tab, select the **Discard Old Builds** check box and specify the discard options.
24. Click **Save**.

Create the Production Sandbox Deployment Build Job

The sandbox deployment job deploys the extension's artifact that was generated in the sandbox packaging job to the Oracle Cloud Applications production instance's sandbox. Before

you create the job, make sure you have credentials that VB Studio can use to access the Oracle Cloud Applications production instance's sandbox.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select the **System Default OL7 for Visual Builder** template.
6. Click **Create**.
7. Click **Configure** .
8. Click the **Before Build** tab.
9. From **Add Before Build Action**, select **Copy Artifacts**.
10. In **From Job**, select the packaging job that generated the extension's artifact.
11. In **Which Build**, select one of the following:
 - Last successful build (default)
 - Last keep forever build
 - Upstream build in this pipeline instance
 - Specified by permalink
 - Specific build
 - Specified by build parameter
12. Depending on what you select, you may be prompted to select which permalink, build, or build parameter you want to be used.
12. Leave other fields with their default or empty values.
13. Click the **Steps** tab.
14. From **Add Step**, select **Application Extension**, and then select **Deploy**.

This image shows the Application Extension Deployment section that's been partially filled in.

Application Extension Deployment ?

Target Instance *
Fusion Applications Cloud Service

Authorization

☒ Use OAuth ☐ Use Basic

i Authorization is required. Authorize

Build Artifact *
extension.vx

15. In **Target Instance**, select the environment with the Oracle Cloud Application's production instance.
16. In the Authorization section, specify the authorization type to run this build step. With **Use OAuth** selected by default, you'll see the *Authorization is required* message, indicating that this build step needs a one-time authorization to handle OAuth requests to your environment's Oracle Cloud Applications instance. Click **Authorize** and enter credentials to access your Oracle Cloud Applications instance; you can also run the job manually and enter the credentials when prompted.

Either way, it is recommended that you authorize your OAuth connection during initial configuration. If you skip this step, you won't be able to publish your changes from the Designer and will need to complete the required authorization before attempting to deploy changes.

Once authorized, the *Authorization has been provided* message shows.

i **Note**


OAuth is the recommended authorization type. Use Basic authentication only if you run into issues with setting up an OAuth connection. To use Basic authentication, select **Use Basic**, then enter the credentials of a user who can access the Oracle Cloud Applications instance in **Username** and **Password**. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.

OAuth tokens (access and refresh) are cycled during regular use. A refresh token is used to obtain an access token whenever a user accesses the target instance. This refresh token is typically valid for seven days. (The token expiration time is set in the IDCS resource app and may be different based on your security requirements.) If the user authenticates with the target instance within the seven-day period, the active refresh token generates a new access token and a new refresh token. This cycle continues indefinitely as long as the refresh token stays valid. If the refresh token expires during extended periods of inactivity (say, when you're away on vacation), click **Renew Authorization** (or run the job manually, so you're prompted to authorize any expired OAuth tokens).

17. The **Build Artifact** field should show the same artifact name that was used in the packaging build step. Confirm this value, especially if the packaging job used an artifact name other than the default, `extension.vv`.
18. Click **Save**.

Create and Configure the Production Sandbox Pipeline

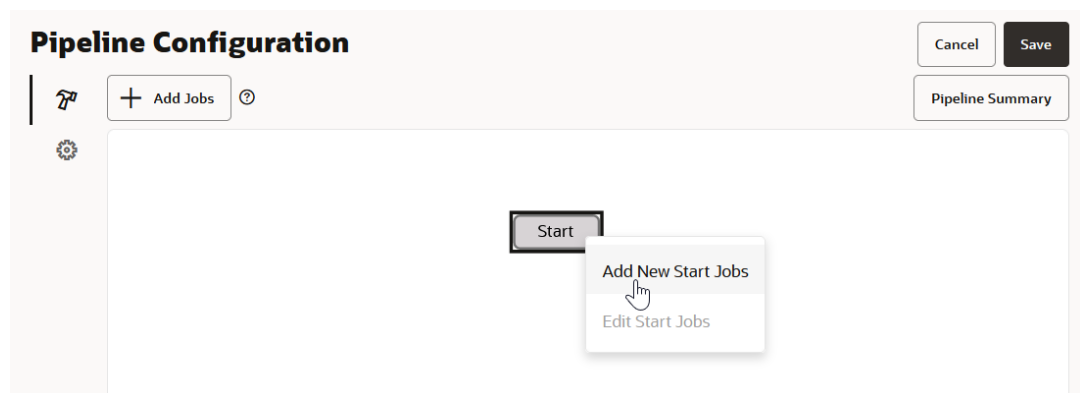
To ensure the package and deploy jobs runs in the correct order, create a pipeline and set the dependency.

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. Click **+ Create Pipeline**.
4. In the **Create Pipeline** dialog box, in **Name** and **Description**, enter a unique name and description.
5. Click **Create**.

The Pipeline Configuration window is displayed with the Start node in the center.



6. Right-click the Start node and select **Add New Start Jobs**.



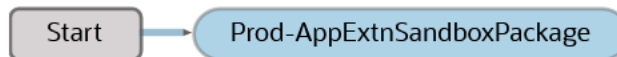
The **Add New Jobs to Start** dialog box is displayed.

7. Click in the **Select new jobs** area to choose new jobs that will start when the pipeline is run.

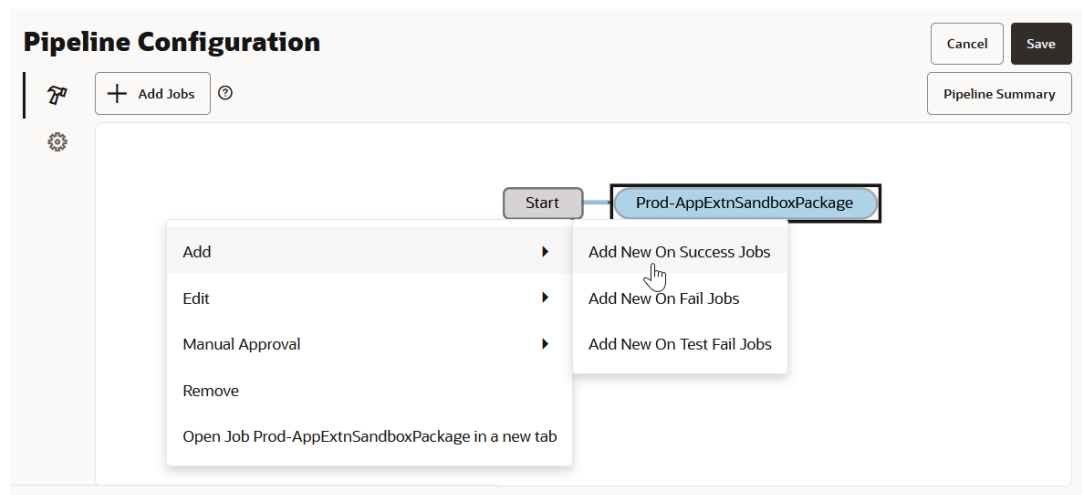
Select one or more jobs from the list displayed.

8. Click **Save**.

The Pipeline Configuration window displays the Start node with a blue line that ends in an arrow connecting to the node that represents the job you just added.



9. Right click the node you just added and select **Add** then select **Add New On Success Jobs**.



The **Add New On Success Jobs** dialog box is displayed.

10. Click in the **Select new on success jobs** area to choose new jobs that will execute if this job runs successfully.

Select one or more jobs from the list displayed.

Add New On Success Jobs ×

Select new on success jobs

Prod-AppExtnDeploy ×

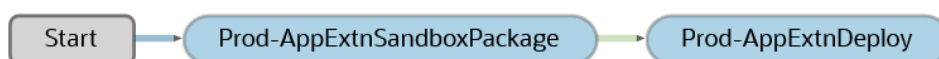
Choose new jobs that will execute if this job runs successfully.

Cancel

Save

11. Click **Save**.

The Pipeline Configuration windows displays the Start node, the node with the build job you previously selected in step 7, and the node with the build job you just selected, all connected by lines that end in an arrow.



12. Click the Pipeline Summary button to display a summary screen that provides many details about the pipeline configuration, similar to the example shown here.

Pipeline Configuration Summary ×

- This pipeline has 2 jobs.
- The following jobs will start when the pipeline is run: Prod-AppExtnSandboxPackage
- If job Prod-AppExtnSandboxPackage is successful, jobs Prod-AppExtnDeploy will build.
- After job Prod-AppExtnDeploy is built, it will not start any additional jobs.

Close

To see even more information about each item in the pipeline's configuration, move the mouse over the node or connection you want to learn more about. For example, in this pipeline configuration, if you hover over the the connection between the Prod-AppExtnSandboxPackage and Prod-AppExtnDeploy nodes, you'll see this information displayed.





Run the Pipelines

Before you deploy your extension to the production instance, run the CSM pipeline to migrate configurations and the production sandbox pipeline to deploy extension to the sandbox where you migrated configurations.

You should run pipelines and jobs in this order:

1. Run the CSM pipeline to migrate configurations.
Before you run the pipeline or the import configurations job, make sure that the source and the target instances match and the target instance does not have any configurations re-keyed or from another source.
2. Run the production sandbox pipeline to deploy the extension to the sandbox where you've migrated configurations.
3. Preview the sandbox and test the extension.
4. If the testing is successful, run the apply configuration job.

Follow these steps to run pipelines in the above order:


1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. On the right side of the CSM pipeline, click **Build** .


Wait for configurations to migrate to the production Oracle Cloud Applications instance's sandbox. Click the pipeline's name to monitor the progress of its build jobs.

4. In the pipeline, if the import configuration job is successful, verify the migrated configurations in the sandbox's preview mode. Open the Oracle Cloud Applications production instance's migration page, click **Incoming** and click **Preview**.

Here's an example:


Migration

Incoming
 Apply Migration Set

Outgoing
 Create Migration Set



Environment Info
Target Instance not set.

Migration

 The migration set has been imported. You can preview and apply it to the environment.


Release1.0.1

Notification E-mail

 Successfully Imported 

Preview **Apply** **Cancel**

Created By alex.cloud Created From myoracleapplications.dev.example.com Release Version 11.1.13.0.0


Created On 2020-07-07 10:11:00 UTC Modules 

To find out on what you can see in the sandbox preview mode, see [Preview Mode in Unified Migration](#) in *Configuring and Extending Applications*.

5. Return to VB Studio's **Build** page.
6. Open the import configuration job's build log and note the CSM Sandbox ID. You'll need the sandbox ID when you deploy the extension to the sandbox.

Here's an example:

```
[2020-07-09 09:31:20] BACKUP_IN_PROGRESS...
[2020-07-09 09:41:31] IMPORT_IN_PROGRESS...
[2020-07-09 09:42:01] BACKUP_DONE...
[2020-07-09 09:42:32] IMPORT_IN_PROGRESS...
[2020-07-09 09:43:02] CSM Sandbox Id: abc01def2fe345fedcba12345ab01234
[2020-07-09 10:26:46] IMPORT_SUCCESS...
[2020-07-09 10:26:46]
```

7. Click the **Pipelines** tab. On the right side of the production sandbox pipeline, click **Build** .
8. In the Configure Parameters dialog, enter the sandbox's name in the `$<SANDBOX_NAME>` format and click **Build Now**.

Here's an example:

Build Pipeline - AppExtn - Deploy to Sandbox ✕

Configure parameters

Sandbox

\$abc01def2fe345fedcba12345ab01234

Enter the sandbox name

Cancel

Build Now

9. After the pipeline is successful, verify the extension along with the migrated configurations in the sandbox's preview mode. Open the Oracle Cloud Applications production instance's migration page, click **Incoming** and click **Preview**.
10. Return to VB Studio's **Build** page.
11. If the imported configurations meet your expectations, save the migration set.

- a. In the **Jobs** tab, open the export configuration job's build log and note the Configuration Migration Set ID.

Here's an example:

```
[2021-01-14 06:18:16] Status: Export Started
[2021-01-14 06:18:16] Configuration Migration Set ID is 1234432100056789
[2021-01-14 06:19:47] IN_PROGRESS...
[2021-01-14 06:42:43] SUCCESS...
[2021-01-14 06:46:19] CREATED...
```

- b. Open the apply configuration job and click **Build Now**.
- c. In the Configure Parameters dialog box, enter the configuration set ID and click **Build Now**.

After saving the configurations on the production instance, if you want to revert to the previously saved instance, run a build of the restore configuration job.

12. Deploy the extension to the production mainline. See [Set Up the Project to Deploy to Production](#) in *Administering Visual Builder Studio*.

After you apply the configurations, Oracle Cloud Application end users won't see any changes to the user interface until they log out and log back in to the App.

B

Public IP Addresses for VB Studio

For VB Studio's list of public IP addresses, see [Public IP Addresses for Oracle Visual Builder Studio](#).