Oracle® Cloud

Administering Oracle Visual Builder in Oracle Integration 3

25.10
F92646-15
November 2025

Oracle Cloud Administering Oracle Visual Builder in Oracle Integration 3, 25.10

F92646-15

Primary Author: Oracle Corporation

Contents

4 Configure Tenant Settings

5 Reference

About This Content

This guide describes tasks for administrators of Visual Builder.

Audience

This guide is intended for administrators who will set up and configure 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

See these Oracle resources:

- Oracle Public Cloud

  `http://cloud.oracle.com`
- About Oracle Visual Builder in *Developing Applications with Oracle Visual Builder in Oracle Integration 3*
- Welcome to Oracle Integration 3 in *Getting Started with Oracle Integration 3*

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.
italic	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

Create a Visual Builder Instance

You use the Oracle Cloud Infrastructure Console to manage Visual Builder instances provisioned in Oracle Integration, for example, to add instances.

> **Note**
>
> If your Visual Builder instance was not provisioned in Oracle Integration, you should refer to Get Started with Oracle Visual Builder in *Administering Oracle Visual Builder* to create and configure Visual Builder instances instead of this guide.

Administer Your Oracle Integration Instances

A Visual Builder instance can be enabled as one of the features in your Oracle Integration subscription. An administrator performs the lifecycle management tasks for Oracle Integration and Visual Builder instances in the Oracle Cloud Infrastructure Console. Depending on your subscription and environment, some options for administering instances might not be available.

The following table describes some Oracle Integration instance management tasks that administrators might need to perform when enabling a Visual Builder instance.

Task	Description
Create an Oracle Integration instance	An administrator can provision new Oracle Integration instances. See Create an Oracle Integration Instance in *Provisioning and Administering Oracle Integration 3*.
View the Oracle Integration instance info	The Oracle Cloud Infrastructure Console provides details on all of your instances. See View Instance Details in *Provisioning and Administering Oracle Integration 3*.
Add users and assign roles	Visual Builder users need to be added to the identity domain of your Oracle Integration instance and assigned at least one of the pre-defined roles that are permitted to access the Visual Builder instance. See Manage Access and Assign Roles in *Provisioning and Administering Oracle Integration 3*.
Start and stop Oracle Integration instances	You can stop instances to free up compute resources used by the instance's virtual machines. See Stop and Start an Oracle Integration Instance in *Provisioning and Administering Oracle Integration 3*.
Enable a Visual Builder instance	You enable a Visual Builder instance from the Oracle Integration Instance Details page. See Enable Visual Builder in Oracle Integration

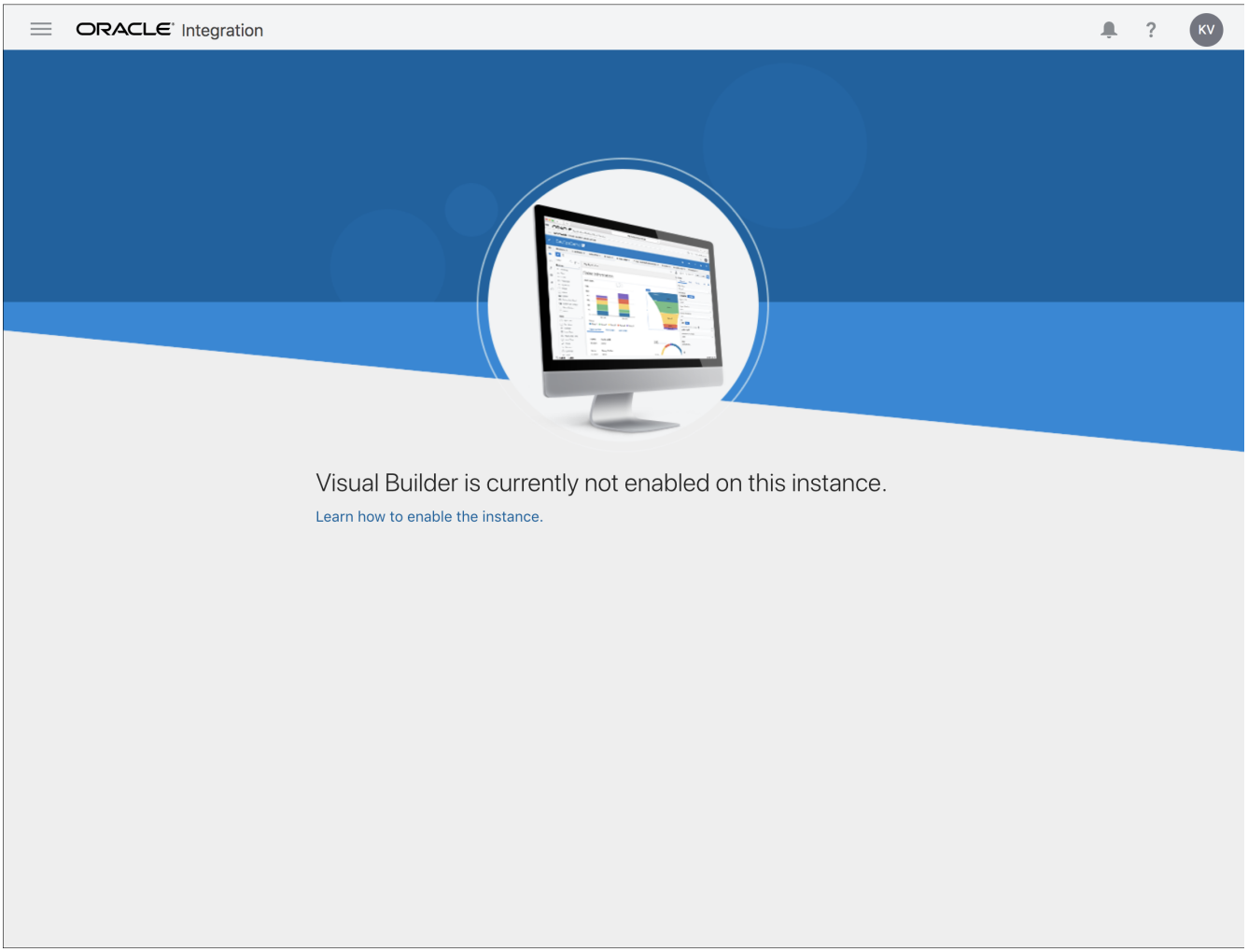
After a Visual Builder instance is enable, administrators can perform lifecycle management tasks for the instance in the Visual Builder instance's details page in the Oracle Cloud Infrastructure Console. See View and Manage the Visual Builder Instance.

Additional global settings for applications created in an instance are set in the Tenant Settings page that administrators can access directly from Visual Builder. See Configure Tenant Settings.

Enable Visual Builder in Oracle Integration

To begin using Visual Builder in Oracle Integration, it must first be enabled for the Oracle Integration instance in the Oracle Cloud Infrastructure Console. Enabling Visual Builder is a one time action and cannot be undone.

If you select Visual Builder from the navigation pane and it's not yet enabled for Oracle Integration, the following message appears:



Important points about enabling Visual Builder:

- Enablement applies to Oracle Integration 3. A Visual Builder link is available in the side navigation pane. When you select the link, instructions for enabling Visual Builder are displayed if it is not yet enabled.

- To enable Visual Builder for an Oracle Integration instance, you must have Oracle Cloud Infrastructure manage access to the instance. See Manage Access and Assign Roles in *Provisioning and Administering Oracle Integration 3*.

  When setting the policies for the VB instance, you must create a policy to allow the OIC administrator group to manage the VB instance. See Set the IAM Policy for Managing the Visual Builder Instance.
- You must enable Visual Builder on each Oracle Integration instance on which you want to use it. Current Oracle Integration 3 users who have used Visual Builder in the past will be automatically enabled.
- The Visual Builder instance is publicly accessible by default, but you can convert it to a private instance inside your VCN. See Convert Your Public Instance to a Private Endpoint and Configure Your Instance as a Private Endpoint.

To enable Visual Builder:

1. Select your instance in the Oracle Cloud Infrastructure Console.

   The Integration Instance Details page is displayed.
2. Click the **Enable** link for Visual Builder on the Integration Instance Information tab.
3. When prompted, click **Enable** to confirm you want to enable Visual Builder.

   If you haven't yet set the required policies for the VB instance, you can use the **OCI Policy** link in the dialog box to open the Policies tab in Oracle Cloud Infrastructure Identity and Access Management (IAM).

   After you click Enable, the Oracle Integration icon turns orange and its status changes to Updating. Enablement can take 15-30 minutes.

   Once complete, the Oracle Integration icon changes back to green with an Active status, and Visual Builder shows as Enabled.
4. After your Visual Builder instance is enabled, you can:
   - View and Manage the Visual Builder Instance
   - Configure Tenant Settings
   - Open the Visual Builder service homepage:
     a. Click **Open console** from the buttons along the top of the Integration Instance Details tab to open the Oracle Integration console.
     b. Select **Visual Builder** in the left navigation pane.
   - Convert Your Public Instance to a Private Endpoint

Set the IAM Policy for Managing the Visual Builder Instance

To enable Visual Builder for an Oracle Integration instance, you need to create a policy that will allow the OIC administrator group to manage the Visual Builder instance.

You'll need to open Oracle Cloud Infrastructure Identity and Access Management (IAM) to create the policy. For details about how to create policies in IAM, see Differences Between Tenancies With and Without Identity Domains and About IAM Policies for Oracle Integration in *Provisioning and Administering Oracle Integration 3*.

To create the policy:

1. Open the navigation menu and click **Identity & Security**. Under **Identity**, click **Policies**.

2. Click **Create Policy**.
3. In the Create Policy window, enter a name and description.
4. In the **Policy Builder**, select **Show manual editor** and enter the required policy statements.

   Syntax:

   ```
   allow group <admin_group> to manage visualbuilder-instance in compartment
   <compartmentId>
   ```

   Example: `allow group domain_admins to manage visualbuilder-instance in compartment oic-dev-comp`

   This policy statement allows the OIC admin group `domain_admins` to `manage` the Visual Builder instance `visualbuilder-instance` in compartment `oic-dev-comp`.
5. Click **Create**.

2

View and Manage the Visual Builder Instance

After enabling the Visual Builder instance, you can view its details and manage it from the OCI Console.

To open the Visual Builder instance in the OCI Console:

1. Open the OCI Console.
2. In the upper-left corner, click **Navigation Menu** ☰.
3. Select **Developer Services** in the menu, then select **Integration** under **Application Integration** in the list of services.
4. If needed, select a compartment in the **Compartment** drop-down list.
5. In the table, click your Oracle Integration 3 instance to open the instance's details page. If no Visual Builder instance has been created, Visual Builder will be marked as " Not Enabled" in the information tab. You can click **Enable** to create an instance.
6. Click **Associated services** under **Resources**.
7. Click the link for the Visual Builder instance in the Associated Services table to open the Visual Builder Instance Details page.

The OCI Console has tools for administering your instance. Some options are not available for Visual Builder instances provisioned in Oracle Integration 3.

To do this ...	Do this:
Add or edit tags	See Manage Visual Builder Tags.
Download activity logs	In the **Activities** section, click **Action** ⋮ and select **Download Logs**.
Download activity error logs	In the **Activities** section, click **Action** ⋮ and select **Download Errors**.
Change the node count	Changing the node count is not supported in Oracle Integration 3. In some cases you might see tools in the OCI Console to perform this task, however, Oracle recommends you do not use them.
Create a custom endpoint	See Create and Configure a Custom Endpoint for Your Instance.
Delete the instance	Once enabled in Oracle Integration 3, you cannot delete a Visual Builder instance. In some cases you might see tools in the OCI Console to perform this task, however, Oracle recommends you do not use them.
Start or stop the instance	Once enabled in Oracle Integration 3, you cannot stop or start a Visual Builder instance. In some cases you might see tools in the OCI Console to perform this task, however, Oracle recommends you do not use them.

To do this ...	Do this:
Move the instance to another compartment	Moving an instance to another compartment is not supported in Oracle Integration 3. In some cases you might see tools in the OCI Console to perform this task, however, Oracle recommends you do not use them.

Some Visual Builder instance management tasks can only be performed in the Oracle Integration 3 instance's details page in the OCI Console. See Administer Your Oracle Integration Instances.

Access Visual Builder from the OCI Console

If you haven't bookmarked the Visual Builder home page in your browser, you can access it from the OCI Console.

1. Open the Visual Builder Instance Details page of the instance.

   The Visual Builder Instance Information tab displays information about your instance, including:

   - Instance creation and last edit date,
   - Instance OCID,
   - Number of nodes,
   - Compartment,
   - NAT gateway IP for the VB service (and VB management, if needed), and
   - VCN OCID for the VB service (and VB management, if needed).

2. Click **Service homepage**.

   If prompted, enter your user credentials and click **Sign In**.

You land on the Visual Builder home page. For quick access, bookmark the home page in your browser.

Edit the Visual Builder Instance

You can edit your Visual Builder instance to change the instance's network access, and to add (or update) a custom endpoint. You cannot rename an instance.

> **Note**
>
> You cannot split a single instance into two parts (for example, into test and development parts). Instead, you must create separate instances for each part.

1. Open the Visual Builder Instance Details page of the instance you want to edit, and then click **Edit**.

   In the Edit Visual Builder Instance panel you can:

   - Configure how the instance is accessible on the network in the Choose network access pane.
     The pane contains options for setting the network access:

- **Default**
  Select this option to allow all networks access to your instance, without restrictions. If you convert a private endpoint to a publicly-accessible instance, after you convert the instance you should confirm that the instance is available publicly, and that the instance can connect to your database.

  If you used a bastion service to access a private endpoint-enabled instance, remove any entries you added in the `/etc/hosts` file.

  When you enable Visual Builder in Oracle Integration 3, the instance is provisioned with this Default option. You can convert the instance to a private endpoint after the instance is provisioned.
- **Secure access from allowed IPs and VCNs only**
  Select this option if you want to limit access to specific IP addresses and VCNs. See Restrict Access to the Instance With an Allowlist.
- **Private endpoint access only**
  Select this option to limit access to a specific VCN. Before converting a publicly accessible instance to a private endpoint, you will need to create and configure the VCN that will contain the private endpoint. See Set the Network Access For a Private Endpoint and Prerequisite Steps for Configuring a Private Endpoint.

> **Note**
>
> The following restrictions apply when converting a publicly-accessible instance to an instance that uses a private endpoint:
>
> * You cannot convert an instance if its node count is greater than one.
> * You cannot convert a VB instance created in an IDCS domain if its home region is different from the currently selected region.
>
> You cannot combine changing an instance's network access with any other instance updates.

- Click **Show Advanced Options** to display the Custom Endpoint pane to add or update a custom endpoint. The custom hostname you want to map to the instance must already be registered on a DNS provider and its SSL certificate stored as a secret in an OCI Vault. See Create and Configure a Custom Endpoint.

2. Click **Save Changes** to update the instance.

Create and Configure a Custom Endpoint

You can map a custom endpoint to a Visual Builder instance and use it to access the instance instead of the original URL generated in the OCI Console.

Let's say you want to open your Visual Builder instance from a custom URL like `https://my-custom-endpoint.example.com/ic/builder` instead of the original URL generated by Oracle (which can look something like `https://<instance-display-name>-<tenancy-name>-<region-code>...oraclecloud.com`). To do this, you create a hostname for your chosen custom domain (for example, `my-custom-endpoint.example.org`), and then create a custom endpoint in your Visual Builder instance that is associated with the hostname. Creating a custom endpoint doesn't affect the original instance URL of your Visual Builder instance. You'll be able to access your instance using the custom endpoint URL as well as the original instance URL.

> **Note**
>
> If you are creating a custom endpoint for a private endpoint-enabled VB instance, and you want to make the custom endpoint public, you will need to use a public load balancer in your tenancy, and create a hostname, listener, and a backend that points to the private endpoint's IP address. This isn't needed if you don't intend to make the endpoint public.

After you have configured a custom endpoint, you can map an app in your instance to the endpoint by selecting the endpoint as the vanity URL in the settings of the visual application containing the app. After setting the app's vanity URL, users can and should open the app directly by entering the vanity URL root (`https://my-custom-endpoint.example.com`) in their browser. For more about using a vanity URL for an app, see Configure Support for a Custom Domain.

These instructions assume you have direct access to a Visual Builder instance and to the OCI Console.

To create and configure a custom endpoint for your Visual Builder instance:

1. Choose a custom hostname for your instance and register it at a DNS provider.
2. Obtain an SSL certificate from a certificate authority (CA) for your hostname.
3. Configure the hostname for your custom endpoint.

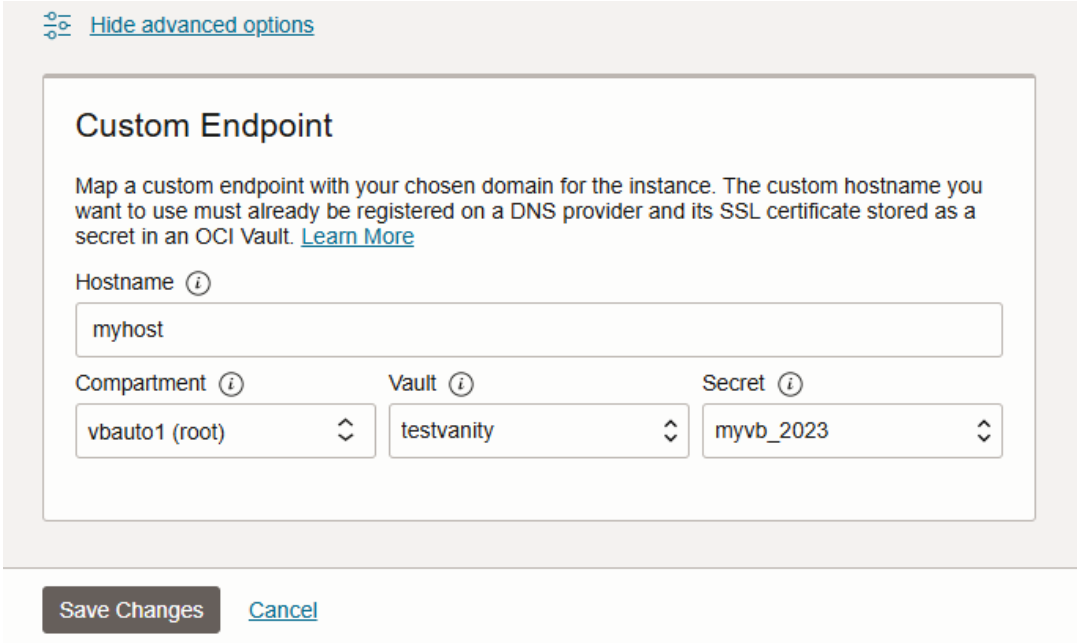
   To create and configure a hostname, do one of the following:

   - Configure a Vault for a Custom Endpoint
   - Configure a Custom URL Using Oracle Web Application Firewall Service V2

   Oracle recommends that you use the Oracle Web Application Firewall service, as this allows you to map your DNS name and upload your associated certificate and private key yourself, and you don't need to update the Visual Builder instance each time a new version of the secret is created.

   > **Note**
   >
   > The options above for creating and configuring a hostname are applicable only for your first (primary) custom endpoint. If your instance already has a custom endpoint and you want to add another, you need to use the command line. Similarly, if your instance already has multiple custom endpoints and you want to edit any of them, you need to do this using the command line. For details on how to do this, see Create and Update Alternate Endpoints.

4. On the Visual Builder Instances page, find the instance you want to work with and open its details page.
5. Select **Edit** to open the Edit visual builder instance panel.
6. Supply the custom endpoint details in the Custom endpoint pane.



Note

If you configured the hostname for the custom endpoint using WAF, you only need to provide the hostname. You do not need to supply the Compartment, Vault, or Secret.

Field	Description
Hostname	Required. Enter the custom hostname chosen for the instance. The custom hostname you want to map to the instance must already be registered on a DNS provider, If the hostname is configured using an OCI vault, its SSL certificate should be stored as a secret in the vault.
Certificate	Required when the hostname is configured using an OCI vault. Provide the location of the hostname's certificate in your OCI tenancy. • **Compartment:** Select the OCI compartment that contains your certificate vault. • **Vault:** Select the vault that contains the hostname's certificate. • **Secret:** Select the secret corresponding to the hostname's certificate.

> **Note**
>
> You can also update or replace a custom endpoint that was previously associated with the instance. You can modify the hostname as well as the certificate details. However, to update the certificate details, you must have access permissions to the vault containing the required certificate. For details, see Update a Secret in a Vault.

7. Finally, update the custom endpoint DNS record to the original instance hostname.

   As a best practice, update the CNAME with the hostname, or update the A record using the public IP address if you want the endpoint to be public. You can obtain the IP address by opening a terminal and using the `dig` command on the VB hostname, for example:

   ```
   dig vb-myinst-vb-adkj3-px.builder.ocp.oraclecloud.com
   ```

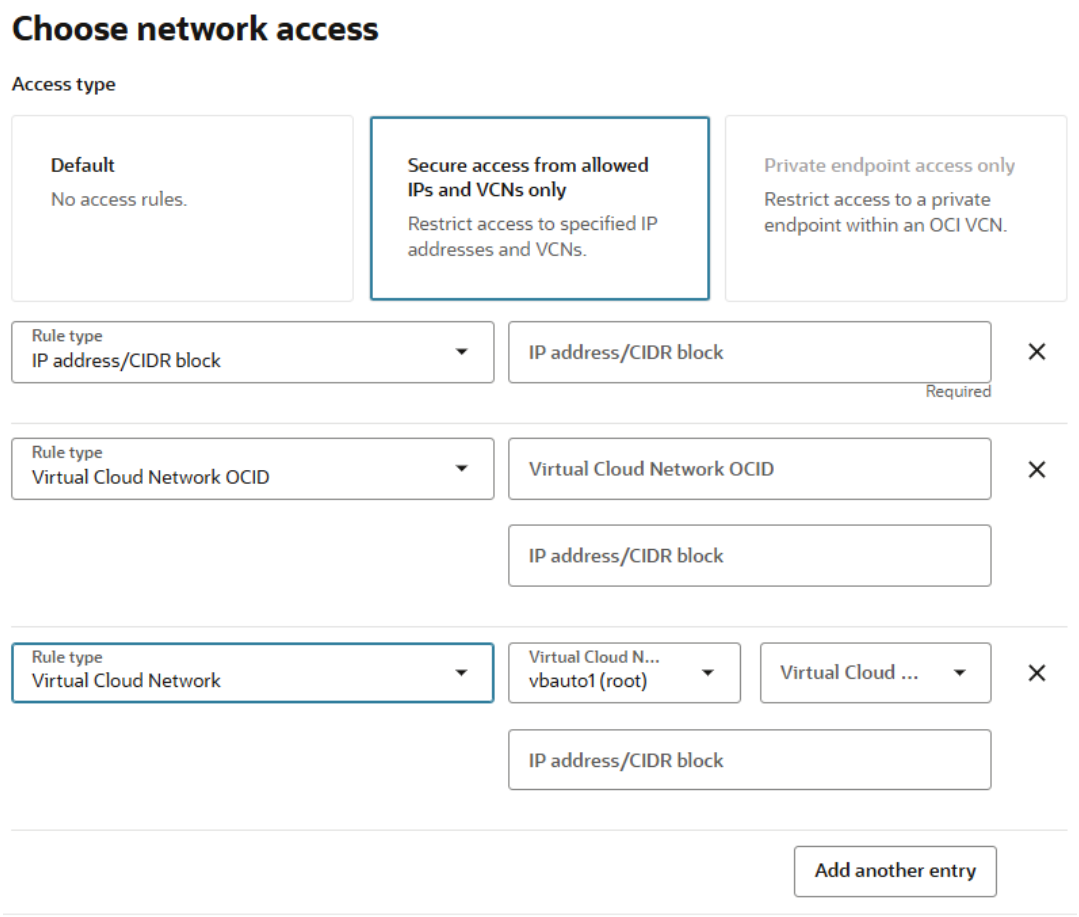
Restrict Access to the Instance With an Allowlist

You can restrict access to your instance by configuring an allowlist when creating an instance, or by editing an existing instance. When the allowlist is enabled, only user Classless Inter-Domain Routing (CIDR) blocks and networks on the list can access the instance.

To add user CIDRs networks to the allow list:

1. Enable allow lists in your instance.
   - If you are creating an instance, select **Secure access from allowed IPs and VCNs only** in the Choose network access pane in the Create Instance dialog.
   - If you are editing an instance without access rules:
     a. Open the Visual Builder Instance Details page for the instance in the OCI Console.
     b. Click **Edit** to open the Edit Visual Builder Instance dialog.
     c. Select **Secure access from allowed IPs and VCNs only** in the Choose network access pane in the dialog.

   If there are no rules listed in the pane, a new empty rule is created when you select the option. There are three types of rules you can define to restrict access. This image shows examples of each rule type:



If you want to disable all allow lists, to allow all networks access to the instance, click **Default** in the Choose network access pane.

2. Select a rule type based on the details you know for the instance, and then enter the details.

   You create a rule for each user/network you want in the allowlist.

   - **IP Address/CIDR Block**. Select this type if you only know the IP address or Classless Inter-Domain Routing (CIDR) block (an IP range) of the instance.
     In the **IP Address/CIDR Block** field, enter the public IP address or CIDR block that is visible on the public internet that you want to grant access.
   - **Virtual Cloud Network**. Select this type if you know the Virtual Cloud Network of the instance and the network route is going through an Oracle Cloud Infrastructure Service Gateway. See Access to Oracle Services: Service Gateway for more information.
     - In the **VCN OCID** field, enter the OCID of the VCN you want to grant access from.
     - Optionally, in the **IP address/CIDR block** field, enter private IP addresses or private CIDR blocks as a comma separated list to allow specific clients in the VCN.
   - **Virtual Cloud Network OCID**. Select this type if you know the Virtual Cloud Network of the instance and the network route is going through an Oracle Cloud Infrastructure Service Gateway. See Access to Oracle Services: Service Gateway for more information.
     - Select the VCN that you want to grant access from. If you do not have the privileges to see the VCNs in your tenancy, this list is empty. In this case, select the **Virtual Cloud Network (VCN) OCID** option to specify the OCID of the VCN.

- Optionally, in the **IP address/CIDR block** field, enter private IP addresses or private CIDR blocks as a comma separated list to allow specific clients in the VCN.

3. Click **Add Another Entry** to create a new rule.
4. Click **x** to remove an entry.

   You can also clear the value in the **IP addresses** or **CIDR blocks** field to remove an entry.

Convert Your Public Instance to a Private Endpoint

If you already have a publicly-accessible VB instance, you can convert it to a private endpoint inside your VCN by editing the instance's network access settings.

Before you can convert an instance to a private endpoint, you will need to complete the steps described in Prerequisite Steps for Configuring a Private Endpoint.

The following restrictions apply when converting a publicly-accessible instance to an instance that uses a private endpoint:

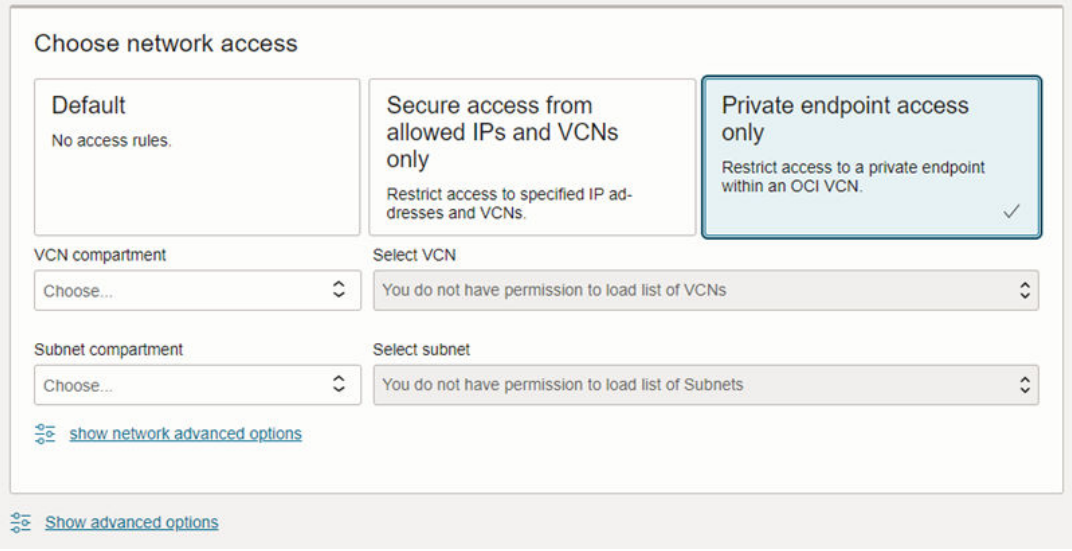
- You cannot convert an instance if its node count is greater than one.
- You cannot convert a VB instance created in an IDCS domain if its home region is different from the currently selected region.

For more details about instances configured as private endpoints, see Private Endpoints Notes.

To convert an instance to a private endpoint:

1. On the Visual Builder Instance Details page, click **Edit** to open the Edit Visual Builder Instance dialog.
2. In the **Choose network access** panel, select **Private endpoint access only**.

   This expands the pane where you configure the VNC details:



> **Note**
>
> You cannot combine changing an instance's network access with any other instance updates.

3. Select a VCN compartment, and a VCN in your compartment.

   See VCNs and Subnets for more information.

4. Select the Subnet compartment, and a private subnet in your compartment.

   See VCNs and Subnets for more information.

5. (Optional) Click **Advanced options** to add network security groups.

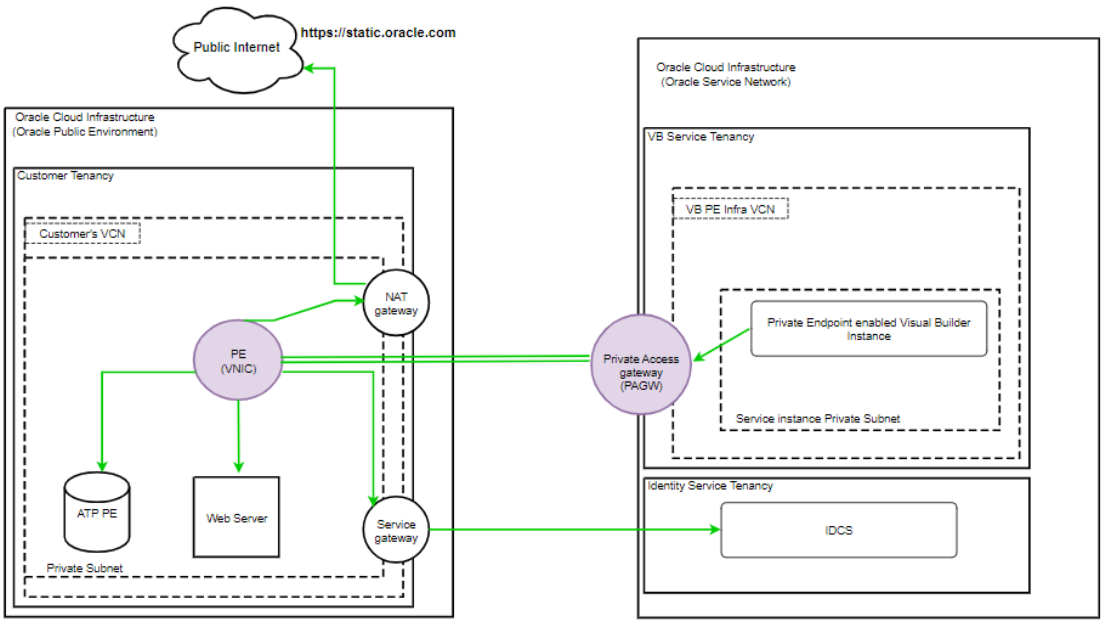
6. Click **Save changes**.

After you have converted the instance, you can update other instance settings, for example, to specify a private IP address or map a custom endpoint to the private endpoint. See Configure Private Endpoint Advanced Network Options and Create and Configure a Custom Endpoint.

Configure Your Instance as a Private Endpoint

Configuring your instance to use a private endpoint inside your Virtual Cloud Network (VCN) in your tenancy instead of a public endpoint allows you to keep all traffic to and from your instance off of the public internet. Specifying the VCN configuration allows traffic only from the VCN you specify, and blocks access to the instance from all public IPs or VCNs.

This diagram shows an example of a network setup for a Visual Builder instance on Oracle Cloud Infrastructure when the instance has a private endpoint enabled.



If you wish, you can allow access to a private endpoint from outside the VCN by using a load balancer in front of the endpoint. This way you can allow public access to the instance, while the instance is within a private VCN where it can access your ATP database.

Prerequisite Steps for Configuring a Private Endpoint

You need to perform some steps before you can configure a private endpoint for a Visual Builder instance.

> **Note**
>
> You can use the Oracle Cloud Infrastructure Resource Manager to help you create the VCN, private subnet and load balancer. See Create Visual Builder Resources Using Oracle Cloud Infrastructure Resource Manager.

Perform the following prerequisite steps before configuring a private endpoint:

- Set required policies for the resources you are working with. See IAM Policies Required to Manage Private Endpoints for more information.
- Create a VCN within the region that will contain your private endpoint instance. See VCNs and Subnets for more information. The VCN and the IDCS of the customer's Identity Domain must be in the same region.
- Configure a private subnet within your VCN configured with default DHCP options. See DNS in Your Virtual Cloud Network for more information.
- Configure your subnet to add a NAT Gateway to allow access from the subnet to the public internet. The minimum requirement is to allow access to the content delivery network (CDN) at `static.oracle.com` on the public internet. The CDN provides resources that are required by the Visual Builder runtime when you stage, publish or use your apps.
- Configure your subnet with a "Service Gateway" to allow connections from the subnet to your Oracle Services (IDCS) instance. For example, you might want to add a Service Gateway to the subnet route table, and set the "Destination" value of the Service Gateway to "All SJC Services In Oracle Services Network". In this case, the subnet security list rules should also allow egress to IDCS using "All SJC Services In Oracle Services Network".
- (Optional) Specify a Network Security Group (NSG) within your VCN. The NSG specifies rules for connections to your instance. See Network Security Groups for more information.

IAM Policies Required to Manage Private Endpoints

In addition to the policies required to provision and manage your instance, some network policies are needed to use private endpoints.

The following table lists the IAM policies required for a cloud user to add a private endpoint. The listed policies are the minimum requirements to add a private endpoint. You can also use a policy rule that is broader. For example, you could set the policy rule like this:

```
Allow group MyGroupName to manage virtual-network-family in compartment
<compartmentName1>
Allow group MyGroupName to manage virtual-network-family in compartment
<compartmentName2>
```

In this policy, `<compartmentName1>` is the compartment where the VCN and subnet exist, and `<compartmentName2>` is the compartment where the Visual Builder instance will be created.

This rule also works because it is a superset that contains all the required policies.

Operation	Required IAM Policies
Configure a private endpoint	`use vcns` for the compartment which the VCN is in `use subnets` for the compartment which the VCN is in `use network-security-groups` for the compartment which the network security group is in `manage private-ips` for the compartment which the VCN is in `manage vnics` for the compartment which the VCN is in `manage vnics` for the compartment in which the visual builder instance is provisioned or is to be provisioned in

Visual Builder relies on the IAM (Identity and Access Management) service to authenticate and authorize cloud users to perform operations that use any of the Oracle Cloud Infrastructure interfaces (the Console, REST API, CLI, SDK, or others).

The IAM service uses **groups**, **compartments** and **policies** to control which cloud users can access which resources. In particular, a policy defines what kind of access a group of users has to a particular kind of resource in a particular compartment. For more information, see Getting Started with Policies.

Create Visual Builder Resources Using Oracle Cloud Infrastructure Resource Manager

You can use the Visual Builder Private Endpoint Quick Start on GitHub and the Oracle Cloud Infrastructure (OCI) Resource Manager to help you create the VCN, private subnet, and load balancer.

The Quick Start on GitHub hosts the zip archive used by the OCI Resource Manager to create the prerequisite infrastructure for a private endpoint-enabled Visual Builder instance. With a single click you can create and deploy the infrastructure for your Visual Builder private endpoint that includes a VCN, private subnet, and load balancer.

To create the infrastructure using OCI Resource Manager:

1. Click this button to open the OCI Resource Manager:

   If the button doesn't work, click this link: Deploy to Oracle Cloud.

   When you click the button, a zip archive for creating the infrastructure is retrieved from the Visual Builder Private Endpoint Quick Start on GitHub, and the OCI Resource Manager opens in your browser.
2. If you aren't already signed in, enter the tenancy and user credentials.
3. Review and accept the terms and conditions.
4. Select the region where you want to deploy the stack.
5. Follow the on-screen prompts and instructions to create the stack.
6. After creating the stack, click Terraform Actions, and select Plan.
7. Wait for the job to be completed, and review the plan.

To make any changes, return to the Stack Details page, click Edit Stack, and make the required changes. Then, run the Plan action again.

8. If no further changes are necessary, return to the Stack Details page, click Terraform Actions, and select Apply.

Set the Network Access For a Private Endpoint

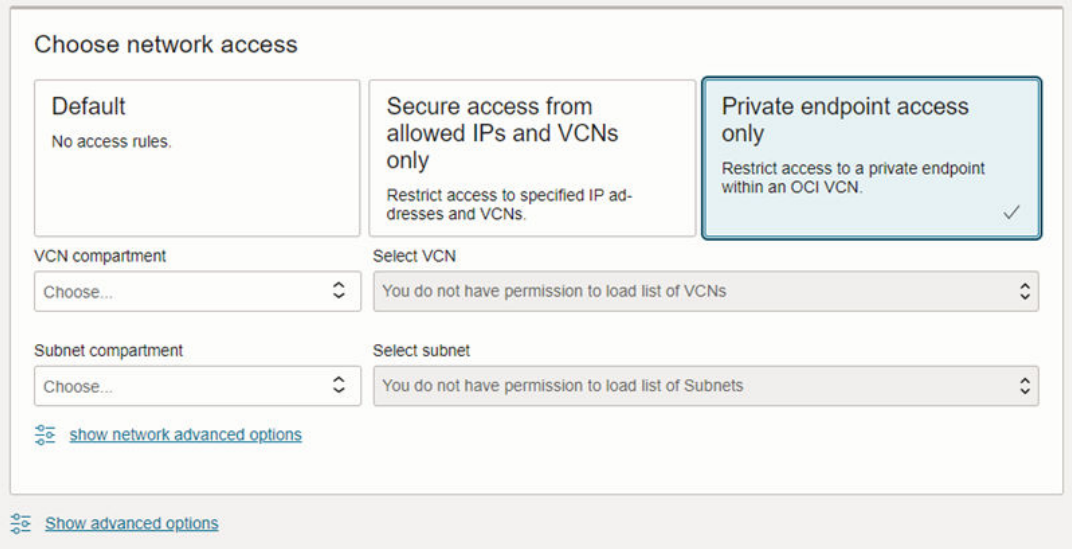
You use the Choose network access panel to configure an instance as a private endpoint. When setting the instance as a private endpoint, you will need to provide details about the VCN where you want the private endpoint.

When configured as a private endpoint, the instance only allows connections from the specified private network (VCN), from peered VCNs, and from on-prem networks connected to your VCN.

These steps assume you are provisioning an instance, or are converting an existing instance to a private endpoint, and you have completed the prerequisite steps:

1. In the **Choose network access** panel, select **Private endpoint access only**.

   This expands the pane where you configure the VNC details:



> **Note**
>
> The following restrictions apply when converting a publicly-accessible instance to an instance that uses a private endpoint:
>
> - You cannot convert an instance if its node count is greater than one.
> - You cannot convert a VB instance created in an IDCS domain if its home region is different from the currently selected region.
>
> You cannot combine changing an instance's network access with any other instance updates.

2. Select a VCN compartment, and a VCN in your compartment.

See VCNs and Subnets for more information.

3. Select the Subnet compartment, and a private subnet in your compartment.

   See VCNs and Subnets for more information.

4. (Optional) Click **Advanced options** to configure advanced options, including adding network security groups, and specifying a private IP address.

   See Configure Private Endpoint Advanced Network Options.

Update the Private Endpoint Details

After an instance is configured as a private endpoint, you can update the instances VCN and subnet details and add network security groups.

1. On the Visual Builder Instance Details page, click **Edit** to open the Edit Visual Builder Instance dialog.
2. Make any changes to the instance's advanced network options.

   You can add the instance to network security groups (NSGs) in the advanced network options. See Configure Private Endpoint Advanced Network Options.

3. (Optional) Make any changes to the VCN compartment and subnet compartment settings.

   > **Note**
   >
   > If you change the subnet, the private endpoint needs to be recreated, and a new IP is assigned to the private endpoint.

   See VCNs and Subnets for more information.

4. Click **Save Changes**.

> **Note**
>
> In some cases you might need to reconfigure your private endpoint. This will delete the private endpoint, and then re-create the endpoint using the same subnet and private endpoint IP from your current settings. To reconfigure a private endpoint:
>
> - Click **More actions** on the Visual Builder Instance Details page, and then select **Reconfigure private endpoint** in the dropdown list. Click **Reconfigure** when asked to confirm.

See Private Endpoints Notes for more information.

Configure Private Endpoint Advanced Network Options

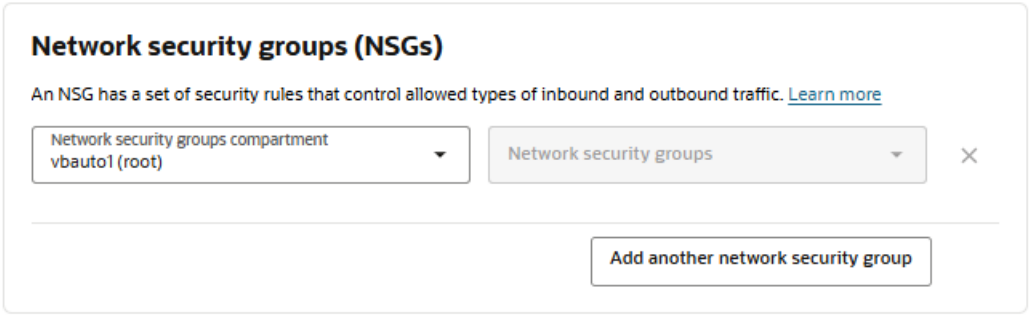
The private endpoint access advanced options allow you to enter a user-specified private IP address and add one or more network security groups.

These steps assume you are provisioning or editing a Visual Builder instance and you are in the **Choose network access** pane.

1. Select **Private endpoint access only**, if not selected.

2. (Optional) Click **Show network advanced options**.

   The advanced network options enable you to provide a private IP address and specify a network security group (NSG).



   a. Optionally enter a **Private IP address**.

      Use this field to enter a custom private IP address. The private IP address you enter must be within the selected subnet's CIDR range.

      If you do not provide a custom private IP address, the IP address is automatically assigned.

   b. Optionally add **Network security groups (NSGs)**.

      If you want more security over connections to the Visual Builder instance, you can define security rules in an NSG; this creates a virtual firewall for your instance.

      - Select a Network Security Group in your compartment to attach the Visual Builder to. If the Network Security Group is in a different compartment, select a different compartment and then select a Network Security Group in that compartment.
      - Click **+ Another Network Security Group** to add another Network Security Group.
      - Click **x** to remove a Network Security Group entry.

      > **Note**
      >
      > Incoming and outgoing connections are limited by the combination of ingress and egress rules defined in NSGs and the Security Lists defined with the VCN. When there are no NSGs, ingress and egress rules defined in the Security Lists for the VCN still apply. See Security Lists for more information on working with Security Lists.

      See Network Security Groups for more information.

Restrict Outbound Traffic Using Network Firewall

You can make outbound traffic from your private endpoint-enabled VB instance more secure by configuring the NAT gateway to ensure that all traffic passing through the gateway is processed by your Network Firewall security rules.

The following are the basic steps for creating a network firewall, and a firewall policy to allow selected URLs to pass through the firewall. For more about using and creating firewalls, see

Learn OCI Network Firewall in Oracle Cloud Infrastructure with Examples and Overview of Creating a Firewall.

> **Note**
>
> To create the firewall policy, you will need to know the reverse connection endpoint (RCE) IP addresses of your Visual Builder private endpoint. You will need to submit a Service Request (SR) to obtain the RCE IPs.

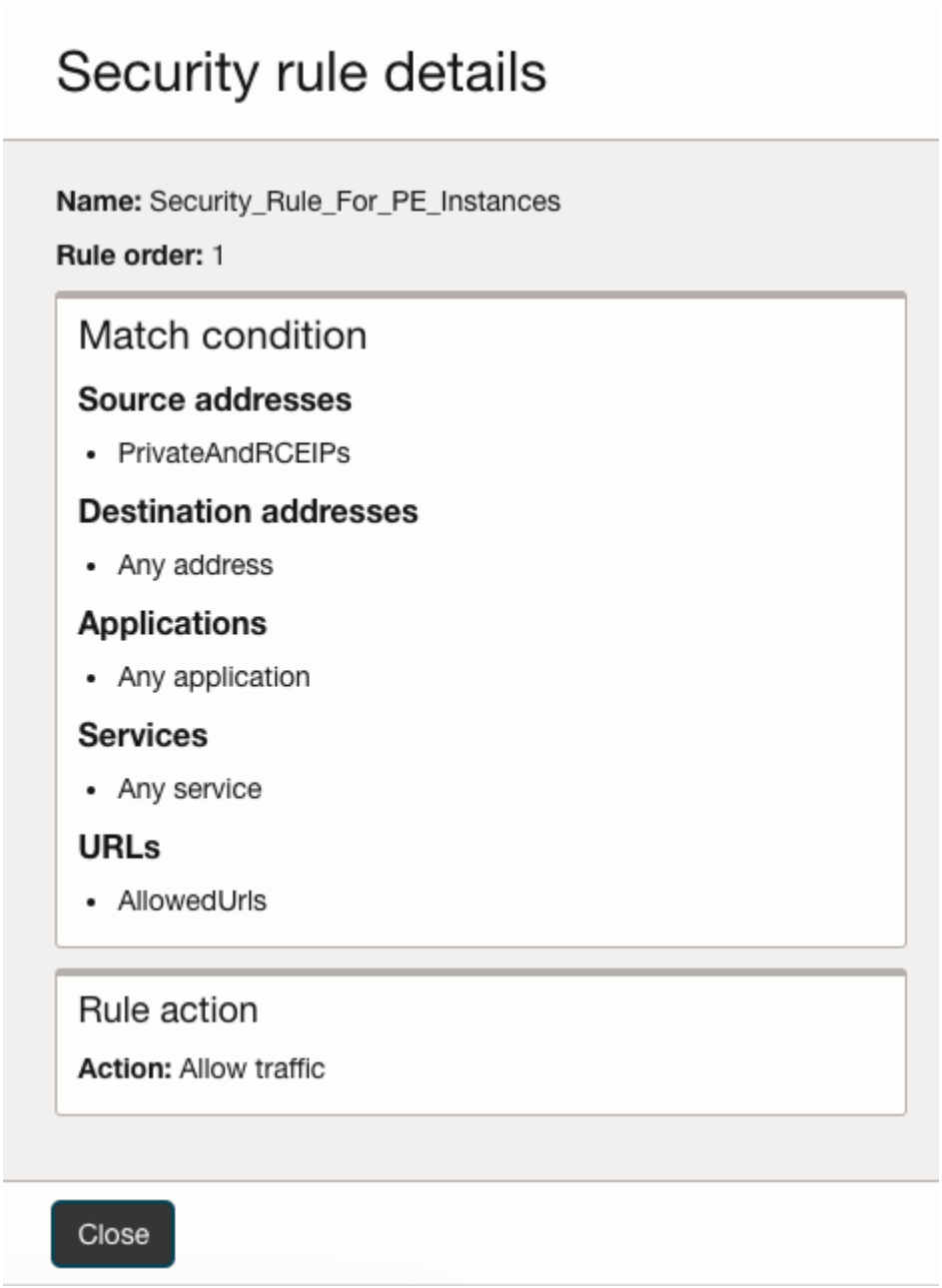
To create a network firewall and policy:

1. Create the network firewall policy.
    a. In the OCI Console navigation menu, click **Identity and Security**, and then select **Network firewall policies**.
    b. Click **Create network firewall policy**, then provide a name and select your compartment in the Create network firewall policy panel. Click **Create network firewall policy**.
    c. Select **Address lists** under Policy resources, then click **Create address list**.
    d. In the Create address list panel, enter the instance's RCE IPs, as well as the private IP address of the VB instance, one on each line. Click **Create address list**.
    e. Select **URL lists** under Policy resources, then click **Create URL list**.
    f. In the Create URL list panel, enter the URLs you want to allow, one on each line. Click **Create URL list**.

       The list must included `static.oracle.com` to allow access to runtime libraries needed during staging and publishing.
    g. Select **Security rules** under Policy resources, then click **Create security rule**.
    h. In the Create security rule panel, enter a name and specify the following security rule details:
        i. In the Source addresses pane, select **Select address lists**, and then select the address list you created.
        ii. In the URLs pane, select **Select URL lists**, and then select the URL list you created.
        iii. In the Rule Action pane, select **Allow Traffic** in the drop-down list.

       Click **Create security rule**.

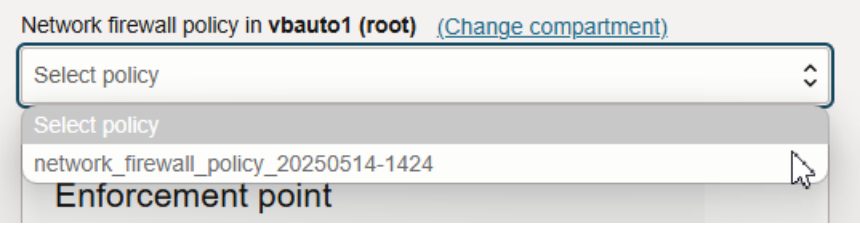
       The details of your security rule might look something like this:



2. Associate the network firewall policy you created with your network firewall.
   a. Select **Network Firewalls**, and then click **Create network firewall**.

      If you already have a firewall, select the firewall you want to use to open the Edit panel.

   b. In the Create network firewall panel (or the Edit panel), select the network firewall policy you created in the drop-down list.



c. Click **Create network firewall** (or **Save changes** if you are editing a firewall).

Access the Instance Locally Using the OCI Bastion Service

You can use the OCI Bastion service to access a private endpoint-enabled Visual Builder instance from your local system.

The Bastion service enables you to create and manage sessions that provide authenticated users temporary access to supported hosts that do not have a public IP address.

1. Create a bastion in the same private subnet as the Visual Builder instance.
   a. In the OCI Console navigation menu, click **Identity and Security**, and then select **Bastion**.
   b. Select **Create bastion**.
   c. Enter a name for the bastion, or use the generated name.
   d. In the Configure networking pane, confirm the target virtual cloud network compartment and target subnet compartment are correct.
   e. Enter the **Target virtual cloud network** for the compartment.
   f. Enter the **Target subnet** for the subnet compartment.
   g. Enter `0.0.0.0/0` in the **CIDR block allowlist**. Click **Create bastion**.
2. Create an SSH port forwarding session to create an SSH tunnel to a specific port on the target resource.
   a. Select the bastion you created.
   b. On the details page, select **Sessions**.
   c. Select **Create session**.
   d. Select **SSH port forwarding session** as the session type.
   e. Enter the **IP Address** of the VB instance, and specify port 443.
   f. Under **Add SSH key**, provide the public key file of the SSH key pair that you want to use for the session.

      You must provide the private key of the same SSH key pair when you connect to the session.
3. Connect to the SSH server.
   a. On the **Bastions** list page, select the bastion that contains the port forwarding session that you want to work with.
   b. On the details page, select **Sessions**, and locate the session that you want to use to connect to the intended target resource.
   c. From the **Actions** menu for the session, select **View SSH command**, and then, next to **SSH command**, select **Copy**. Select **Close**.

The copied SSH command might look something like this:

```
ssh -i <privateKey> -N -L <localPort>:10.4.0.22:443 -p 22
ocid1.bastionsession.oc1.us-sanjose-1.amaaaaaarnqxz5aa2vl
lvisdqikxdhsq@host.bastion.us-sanjose-1.oci.oraclecloud.com
```

d. In a text editor, edit the command to replace `<privateKey>` with the path to the private key for the public key used when you created the session, and change the `<localPort>` to port 443.

   The edited SSH command might look something like this (changed text in bold):

```
sudo ssh -i ~/Downloads/ssh-key-2025-02-10.key -N -L 443:10.4.0.22:443 -
p 22
ocid1.bastionsession.oc1.us-sanjose-1.amaaaaaarnqxz5aa2vl
lvisdqikxdhsq@host.bastion.us-sanjose-1.oci.oraclecloud.com
```

   You might need to use `sudo` to listen on port 443.

e. Open your command terminal and run the SSH command to start listening on port 443.

4. On your local system, add an entry in the `/etc/hosts` file for the service URL.

   The entry in the `hosts` might look something like this:

```
127.0.0.1 private-test-vb.builder.us-sanjose-1.ocp.oraclecloud.com
```

For as long as the bastion session is active, you can access the VB instance by opening the URL you specified in the hosts file (`private-test-vb.builder.us-sanjose-1.ocp.oraclecloud.com`) in your browser.

Private Endpoints Notes

Describes restrictions and notes for private endpoints on Visual Builder.

- After you update the network access to use a private endpoint, or after the provisioning completes where you configure a private endpoint, you can view the network configuration on the Visual Builder Details page under the **Network** section.

  The **Network** section shows the following information for a private endpoint:

  – **Subnet**: This includes a link for the subnet associated with the private endpoint.

  – **Private endpoint IP**: Shows the private endpoint IP for the private endpoint configuration.

  – **Network security groups**: This field includes links to the NSG(s) configured with the private endpoint.

- You can map a custom endpoint to a private endpoint during the provision process, or after provisioning completes.
- You can specify up to five NSGs to control access to your instance.
- You can change the private endpoint Network Security Group (NSG) for the instance.

  To change the NSG for a private endpoint, do the following:

  1. On the Visual Builder page, select the instance you want to edit.

2. On the **Visual Builder Details** page, click **Edit**. In the Edit instance, click **Show network advanced options** to open the pane and edit the details in the Network security groups (NSGs) pane.

- You can connect your private endpoint to an ATP database in the same VCN and subnet. See Access an ATP Database Configured as a Private Endpoint. To connect to a database in a different VCN, you need to configure private views using DNS in the VCNs.
- Modifying a private IP address is not allowed after you provision an instance, regardless of whether the IP address is automatically assigned or if you enter a value in the **Private IP address** field.
- You cannot change the node count for private endpoint-enabled Visual Builder instances. You will need to raise a service request to increase the node count for private-endpoint enabled Visual Builder instances.
- When using a load balancer in front of a private endpoint, use the private endpoint IP for the Load Balancer Backend, and forward traffic on port 443. You also need to share the IP address of the public load balancer with your DevOps/Networking team so they can update the DNS registration to make the instance URL publicly accessible.

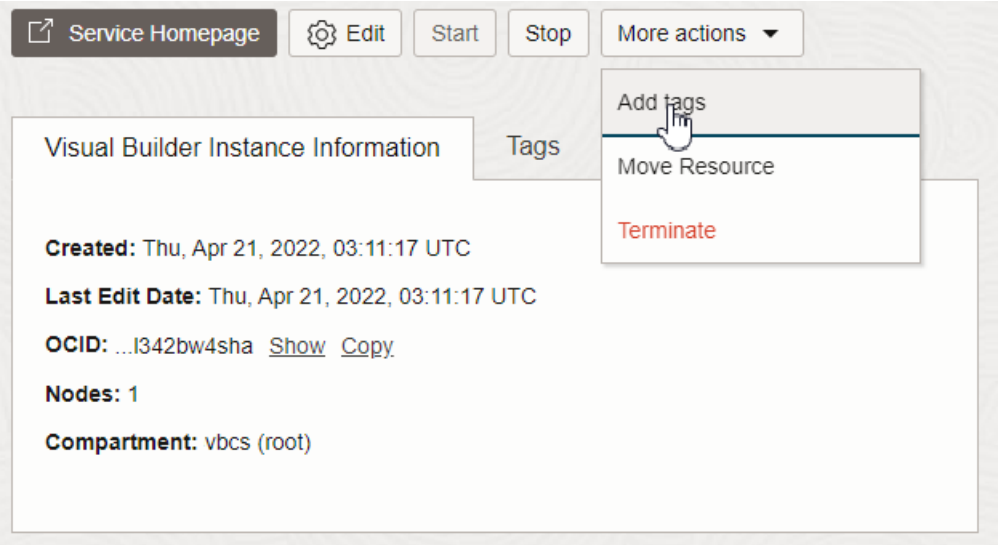
Manage Visual Builder Tags

OCI tags enable you to tag your OCI resources, such as your Visual Builder instance, and help you organize resources based on your business needs. You can manage tags from the Visual Builder Instance Details page.

If you're new to OCI tags, see Tagging Overview.

Action	How To
Add a tag	1. On the Visual Builder Instance Details page, click **Add Tags** in the **More Actions** menu. 2. In the Add One Or More Tags To This Resource dialog box, enter the tag's namespace, key, and value. 3. Click **Add Tags**.
Edit a tag	1. On the Visual Builder Instance Details page, click the **Tags** tab. 2. To edit a tag, click its **Edit** icon. 3. In the Edit Tag dialog box, edit the tag and click **Save**.
Remove a tag	1. On the Visual Builder Instance Details page, click the **Tags** tab. 2. To edit a tag, click its **Edit** icon. 3. In the Edit Tag dialog box, click **Remove Tag**.

You can see the Add Tags item in the More Actions menu in this image.



View Instance Activities

You can view a list of instance life cycle activities, such as when the instance was created or updated, in the Activities table. You can also download log files for each activity.

To view the instance's life cycle activities:

1. On the Visual Builder Instances page, find the instance you want to work with and open its details page.
2. On the details page, select **Activities** under **Resources** in the left navigation pane to display the Activities table.

   The Activities table lists the different types of activities, the status of each action, and when the action was performed.
3. (Optional) In the Activities table, from the **Actions** menu in the row of the activity you are interested in, select **Download Logs** or **Download Errors**.

View Instance Metrics

You can use the Metrics pane in the OCI Console to view data about your instance's resource consumption and the number of logged-in users.

To view instance metrics during a specific period of time.

1. Open the Visual Builder Instance Details page of your instance.
2. Click **Metrics** under **Resources** in the left navigation pane.
3. In the Metrics pane, select the Start Time and End Time of the period you want to examine. You can also select a period in the Quick Selects dropdown list.

The Metrics pane contains charts displaying details about your instance:

- The OCPU Consumption chart displays the percentage of each of the instance's CPUs that is being used during a given period.
- The Concurrent Users chart displays the number of users that are logged in to the instance during a given period.

- The Database Usage chart displays how much data is stored in the database during a given period. To see the usage, you need to define a period of time, either by using the Start Time and End Time fields to select dates, or the Quick Select to select a period (for example, "Last 12 hours").
- The Memory Usage chart displays the space currently in use, measured in pages, and expressed as a percentage of used pages versus unused pages.

Each chart contains Interval and Statistic dropdown menus for modifying how the metrics are displayed. Each chart also has an actions menu with additional options. When you hover over a chart, a tooltip is displayed with more detailed metrics.

View Services Associated With Your Instance

You can use the Associated Services table in the OCI Console to see a list of the services that are attached to your VB instance.

When a VB instance is created by a service other than Visual Builder, for example, if an instance is created by enabling Visual Builder in Oracle Integration Cloud (OIC) service, the service and the VB instance are then "attached" to each other. If a VB instance is attached to a service, the attached service is listed in the Associated Services table in the OCI Console.

To open the Associated Services table:

- Open the Visual Builder Instance Details page, and then click **Associated services** under the Resources menu.

You can open the home page of the attached service using the Service console URL in the table.

In some cases, an attachment can be via another service. For example, an Oracle Cloud Application service might trigger an OIC service to create a Visual Builder instance. In this case, the Associated Services table would show this relationship using "parent" and "child" labels: the Oracle Cloud Application service's role would be "Parent", and the OIC service's role would be "Child".

3

Upgrade from Oracle Integration Generation 2 to Oracle Integration 3

Oracle is in the process of upgrading Oracle Integration Generation 2 to Oracle Integration 3. The upgrade is available at no extra cost.

There are some Visual Builder tasks you need to perform before and after your Oracle Integration instance is upgraded to Generation 3. For the full details on upgrading Oracle Integration, see Upgrade from Oracle Integration Generation 2 to Oracle Integration 3 in *Provisioning and Administering Oracle Integration 3*.

Oracle performs the majority of the work for upgrading Oracle Integration Generation 2 instances for you, including upgrading your eligible Visual Builder instances.

For a list of known issues in Visual Builder after your Oracle Integration instance is upgraded to Generation 3, see Known Issues for Live/Staged Apps Post-Upgrade to Oracle Integration 3 in *Known Issues for Oracle Integration 3*.

Upgrade Readiness Checks

Oracle periodically performs some prechecks to determine your upgrade readiness so that your upgrade runs smoothly. If the prechecks don't pass, you may need to perform tasks to correct the issues. You can see the precheck status of your Oracle Integration Generation 2 instance, or run a precheck again, in the instance's Settings page. For more, see Check Upgrade Readiness and Correct Precheck Issues in *Provisioning and Administering Oracle Integration 3*.

The following prechecks are performed for Visual Builder instances in Oracle Integration Generation 2:

- Is a custom endpoint defined in the Visual Builder instance?
  The instance cannot be upgraded to Oracle Integration 3 if a custom endpoint is defined in the Visual Builder instance. To upgrade the instance, wait until Oracle starts upgrades for custom endpoints, or, if you want to move forward with the upgrade now, delete the custom endpoint from your Visual Builder instance.

  After the instance is upgraded to Oracle Integration 3, you can define a new custom endpoint in the Visual Builder instance.

  > **Note**
  >
  > A custom endpoint defined in a Visual Builder instance is not the same as a custom endpoint created in Oracle Integration. During upgrade, any custom endpoints that you created in Oracle Integration will be configured in Visual Builder, so, after the upgrade, only Visual Builder apps will be accessible through the custom endpoint. You cannot use the same custom endpoint for both Visual Builder and Oracle Integration after the upgrade. If you use the same custom endpoint, you might run into issues.

- Does Visual Builder use an Oracle database instance?

If you are using your own Oracle database instance (BYODB) with your Visual Builder instance, the instance can be upgraded to Oracle Integration 3. However, Autonomous Transaction Processing (ATP) must be up and running during the upgrade. Make sure you complete the additional tasks described in Prepare Visual Builder for the Upgrade below.

Prepare Visual Builder for the Upgrade

Examine your applications, especially your live and staged applications, to check whether they might break due to any of the Known Issues for Live/Staged Apps Post-Upgrade to Oracle Integration 3. For these cases, follow the relevant instructions in Tasks to Complete After the Upgrade below.

- Confirm that you have upgraded your applications to the supported VB runtime and JET versions, and verify that your applications work correctly. If you miss this step, your staged or live application might stop working after the upgrade. For more, see Visual Builder apps on obsolete VB runtime versions in *Known Issues for Oracle Integration 3*.
- If you use the "delegate authentication" authentication type in any service connections or backends, you must switch the authentication type to "Oracle Cloud Account". The "delegate authentication" authentication type is deprecated. After you change the authentication type, test your service connections and apps to confirm that they still work correctly.
- If you use Visual Builder with your own Oracle database instance (BYODB), there are several steps you need to take to prepare for upgrade:
  1. Keep Autonomous Transaction Processing (ATP) up and running during the upgrade.
  2. If you have an allowlist (also known as an ACL) configured in ATP, add the Visual Builder VCN OCID to the ATP allowlist.
  3. Ensure that the user name and password for your ATP instance are configured correctly in Visual Builder.
  4. Reset the expired password or wallet that is used for connecting to ATP.
     a. Download the latest wallet. See Download Database Connection Information in *Using Oracle Autonomous Database Serverless*.
     b. Upload the wallet on the instance's **Tenant Settings** page. See Update Your ATP Wallet and Reset an Expired Password.

Tasks to Complete After the Upgrade

- Configure an IAM policy for Visual Builder in Oracle Integration 3. See Set the IAM Policy for Managing the Visual Builder Instance.
- Update the allowlist and network access rules to add the Visual Builder service VCN OCID in Oracle Integration 3. See Allow Your Instance to Access Services.
- Update DNS records for custom endpoints.
  - If your instance has a primary custom endpoint configured to use WAF V2 or a load balancer, you need to update the custom endpoint DNS record's CNAME with the Visual Builder instance host name or the IP address of the VB Generation 2 load balancer. To update the load balancer:
    * Add a backend using the load balancer
    * Add a new rule to the route table for the VCN (NAT Gateway) for the public load balancer CIDR (IP/32)
  - For your instance's alternate custom endpoints, after migration the alternate endpoints will not be located on the same load balancer as the primary endpoint, so updating the

DNS CNAME will not work for your alternate endpoints. For your alternate endpoints, you will need to file a service request (SR) to get the endpoint details you need to update the DNS records.

- Update backends and service connections.
  - Oracle Integration 3 does not accept Basic Auth, and recommends OAuth. For more details, see When is Basic Authentication Supported in Oracle Integration 3?

    After migration, you need to update backends and service connections to the Oracle Integration REST API that use Basic Auth, to change them to use an appropriate OAuth mechanism (OAuth 2.0 Resource Owner Password is the most similar to Basic Auth). For backends and service connections connecting to Oracle Integration REST API using "Oracle Cloud Account" authentication, you don't need to change the authentication.
  - If your backend (or service connection) had the connection type set to "Dynamic, Service supports CORS", confirm this is set to "Always Use Proxy". This is to avoid any Chrome 119 issues that might arise due to redirects of the Oracle Integration REST API.
  - If your service connections to Integration REST APIs were created from the catalog, they will continue to work after the migration (provided Basic Auth is replaced with a suitable OAuth mechanism, and the connection type is updated to "Always Use Proxy").
  - Service connections to the Oracle Integration design-time (or factory) APIs, created using the endpoint flow in Visual Builder, that are not listed in the catalog, will not work after the migration, and you need to update them.

    An example of such an endpoint is a lookup endpoint (see Lookups REST Endpoints in *Developer API for Oracle Integration 3*). When you test them in the service connection's Test tab, these endpoints will typically give error code HTTP 403, with the error "Unable to verify URL against allowed list". See SendRequest in *Developer API for Oracle Integration 3* for more on the design-time URL and factory APIs.

    To rework these service connections using a custom backend:

    1. Create a custom backend called `icsFactoryApi` (for example). For the URL, use the design-time URL (for example, `https://design.integration.region.ocp.oraclecloud.com`), and use OAuth authentication (generally OAuth 2.0 Resource Owner Password). Set the connection type to "Always Use Proxy" (recommended).
    2. For existing service connections based on design-time APIs:
       a. Change the URL from `vb-catalog://backends/ics` to point to `vb-catalog://backends/icsFactoryAPI`. (If the URL is `https://<integration_base_url>`, then replace `<integration_base_url>` with `vb-catalog://backends/icsFactoryAPI`).
       b. Navigate to the Request tab, and then add the static query parameter of "integrationInstance", and add the appropriate value.
    3. The static query parameter introduced this way becomes part of the visual application code. This parameter will be different for different Integration instances, so you would need to update the parameter if you import the app to another instance (for example, a prod instance). If you use a build pipeline, you would need to modify the parameter in the packaging jobs.
  - (Optional, but recommended) Set and store the details about the authentication type, connection type, and credentials on the backend, and have all the service connections based on the backend. Defining details only on the backend can help you manage the

details, so you don't have to duplicate them as much. See the blog post Streamlining Service Connections to use Backends for more information.

- PWA apps installed on user devices will stop receiving automatic updates after your Visual Builder instance is upgraded from Oracle Integration Gen 2 to Oracle Integration 3.

  After the upgrade:

  1. Republish the PWAs.
  2. Provide users with the new URL. You should instruct the users that they need to uninstall the old apps and install the new ones from the new Visual Builder location in Oracle Integration 3.

  The apps will receive automatic updates after they are re-installed. For more, see PWA app no longer receives updates automatically after migration to OIC 3 in *Known Issues for Oracle Integration 3*.

- If your application code has any references to the Oracle Integration Gen 2 Visual Builder URL, you need to update them to the new Oracle Integration 3 VB URL after the upgrade for your apps to work correctly. For more, see Unable to access VB business object URLs from service connections in live app and Custom code accessing pre-upgrade Visual Builder URL failing in *Known Issues for Oracle Integration 3*.

4

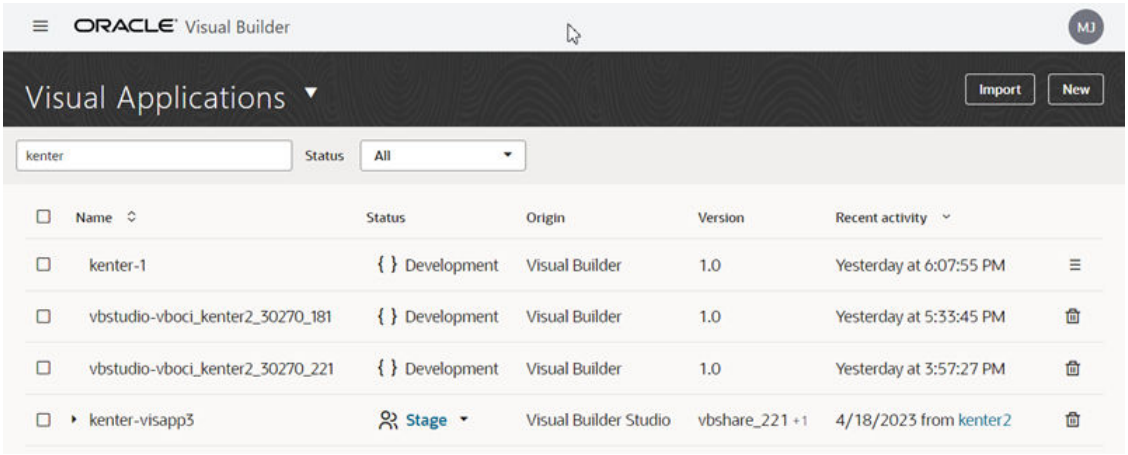
Configure Tenant Settings

After a Visual Builder service instance is created, an identity domain administrator assigns one or more users the Visual Builder Administrator role for the service instance. A Visual Builder Administrator can manage and set general options for applications in the service instance.

Manage Applications in the Service Instance

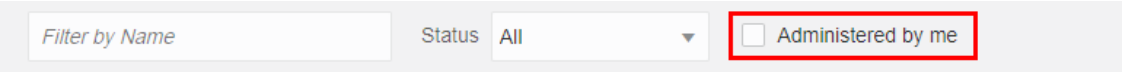
The Visual Builder Home page helps you manage visual applications created on the current Visual Builder service instance, as well as apps deployed to the instance from a different VB instance or Visual Builder Studio (VB Studio).

Each row on the Home page represents an application created either on a Visual Builder instance, or on a VB Studio instance that is associated with this Visual Builder instance. The Origin column on the Home page contains "Visual Builder Studio" if the app was shared or published in Visual Builder Studio:



The first row in the image above represents an application created on this VB instance, which can be managed using the ☰ in the right column. The rows below represent applications created on a VB Studio instance, and the only option on the Home page for these is to click 🗑 in the right column to remove them from the VB instance. To manage an app created in VB Studio, you should open the app's VB Studio project. The VB Studio project provides tools for managing an application's lifecycle that are equivalent to those in the Options menu for applications created on the VB instance. For example, in the project's Builds tab you can configure a build job to set the app's version and then stage it. For more about managing an application's lifecycle in VB Studio, see Preview, Share, and Deploy Visual Applications in *Building Responsive Applications with Visual Builder Studio*.

If you're an administrator, you can manage (which includes deleting) any of the apps shown on the Home page. To see the full list of applications, select the **Administered by me** checkbox next to the Status dropdown list.



If you're a developer, you can manage the apps that you've created in this instance of Visual Builder, or if you're a team member of the app. You can also manage VB Studio apps from the Home page, as long as you've created, shared or published it. You will not see the **Administered by me** checkbox unless you have the role of administrator.

IDCS Client Applications

Each time a visual application is created on a Visual Builder instance provisioned with OIC, a companion *client application* is automatically created in IDCS. When you stage or publish the app to a different server governed by the same IDCS instance, another client application is created. This means there may be several client applications for the same visual application running on IDCS for the lifespan of the app.

If you have IDCS administrator privileges, you can see these client apps on the IDCS console, but you should not need to *manage* them in any way. When a visual application is removed from the server, the associated client apps are removed from IDCS. You may need to *interact* with the client apps if you want to set login-related policies for an instance—for example, to enable the **Keep me signed in** policy, or to assign the users and groups who can access the application.

When you want to view client applications in the IDCS console, make sure that you are looking at the correct IDCS app for the instance.VB instances provisioned with OIC share the OIC instance's IDCS app, and the client apps are listed in that OIC instance's IDCS app. You can look at the URL of the service host name to help determine if a VB instance was provisioned with Oracle Integration 3 or Oracle Integration Generation 2:

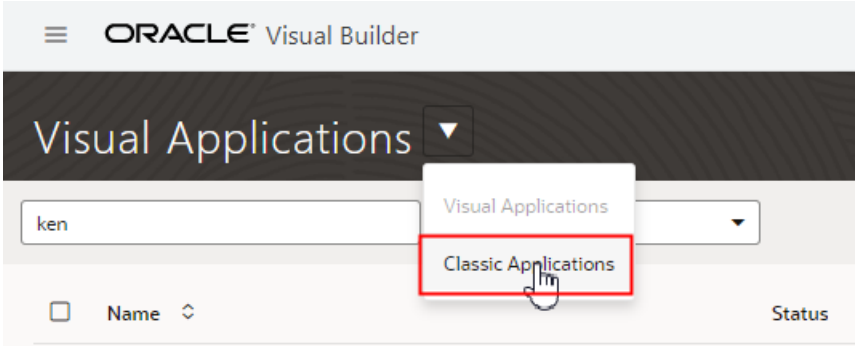
- When the URL of the service host name contains `<SUB-DOMAIN>.builder.ocp.oraclecloud.com`, the VB instance was provisioned with Oracle Integration 3 (or it's an Oracle Visual Builder instance).
- When the URL contains `<SUB-DOMAIN>.integration.ocp.oraclecloud.com`, the VB instance was provisioned with Oracle Integration Generation 2.

> **Note**
>
> For Oracle Visual Builder instances not provisioned with Oracle Integration, the URL also contains `<SUB-DOMAIN>.builder.ocp.oraclecloud.com`, however, the instances will have their own IDCS app, and the client apps are listed in the Oracle Visual Builder instances' IDCS apps.

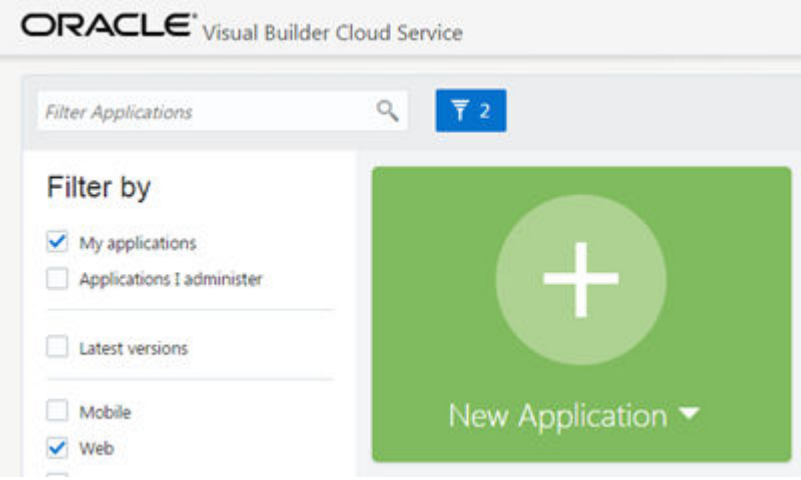
For more, see Typical Workflow for Managing Oracle Identity Cloud Service Applications in *Administering Oracle Identity Cloud Service*.

Visual Builder Classic Applications

If you have any classic applications (apps that have the older Visual Builder project structure), open the Visual Applications dropdown list in the header and select **Classic Applications**.



On the Home page for classic applications, administrators can select the **Applications I administer** checkbox in the Filter by pane to display the applications where they are not a team member.



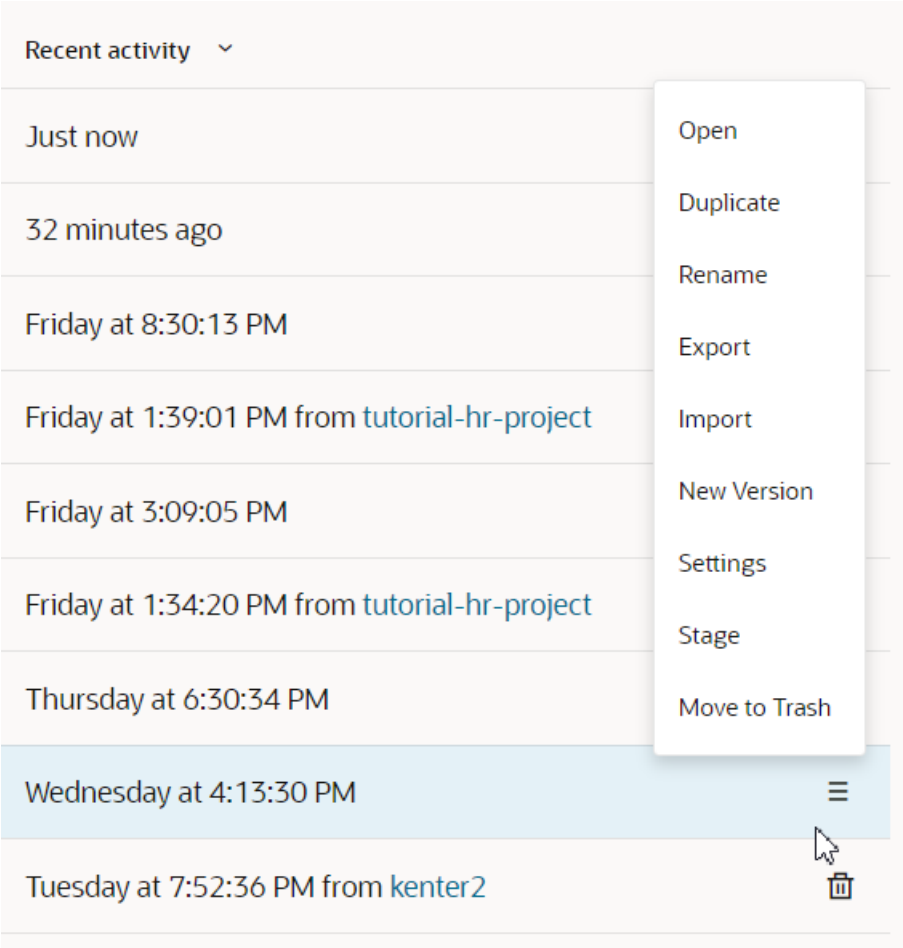
Manage Applications Created on the Instance

The table of applications on the Home page includes visual applications created on the Visual Builder instance. Each of these applications has an Options menu in the right column that developers and administrators can use to manage it, for example, to add and remove team members, and to open, stage and publish the application.

You can tell which applications were created on the instance by looking for ☰ in the right column. If you see a 🗑 button instead of ☰, the application was created on a different instance of Visual Builder. If an application was created on a different instance, you'll need to open the application on the instance where it was created and manage it there.

To manage an application created on the current Visual Builder instance:

- Click ☰ in the right column of the application, and then select a task in the Options menu:



The Options menu displays commands for managing the application (based on the application's status, some commands might be hidden):

Menu Item	Description
Open	Opens the development version of the application
Duplicate	Creates a clone of this version of the application, including the content of the database.
Rename	Opens a dialog box where you can change the name of the application.
Export	Creates a ZIP archive of the application that can be imported as a new application. When exporting the application, you can choose if you want the exported archive to include the data stored in your business objects.
Import	Opens a dialog that you can use to create an application by uploading an application archive (ZIP or OVB) from your local system.
New Version	Creates a new version of the same application. By default the new version is a development version. Version numbers are automatically increased incrementally.
Settings	Opens an editor for configuring the application’s settings and viewing the application API URLs. Each application version has a dedicated Settings editor.

Menu Item	Description
Stage	Opens a dialog box where you can specify the database option for the staged application. When an application is staged, a link to the staged version is displayed in the tile.
Publish	Opens a dialog box where you can specify the database option and publish the staged version of your application.
Lock / Unlock	Enables you to lock a live application to prevent any users from using the application. You would usually use this command when you are going to update the live application with a newer version. The Unlock option is displayed only when the live application is locked.
Rollback	Rolls back the live version to the previous live version. This is only available for the current live version, and there must be an older live version of the app.
Move to trash	Deletes the application from the Identity Domain. You have 30 days to recover the application after deleting it. For more, see Delete a Visual Application in *Developing Applications with Oracle Visual Builder in Oracle Integration Generation 2*.

Access Tenant Settings

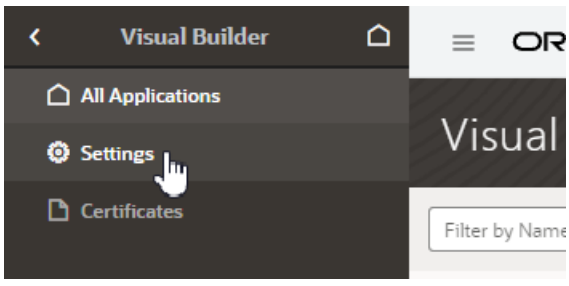
An instance administrator can access the Tenant Settings page for managing the instance's global settings from any Visual Builder page.

The Tenant Settings page contains three tabs: General, Tenant Database and Services. The General tab has panels for configuring security settings, specifying Access Denied messages, and configuring the Component Exchange details. You use the Tenant Database tab to switch to an Oracle database and to see how much database space your applications are using. You use the Services tab to add and edit the backend services that are accessible to apps in the tenant.

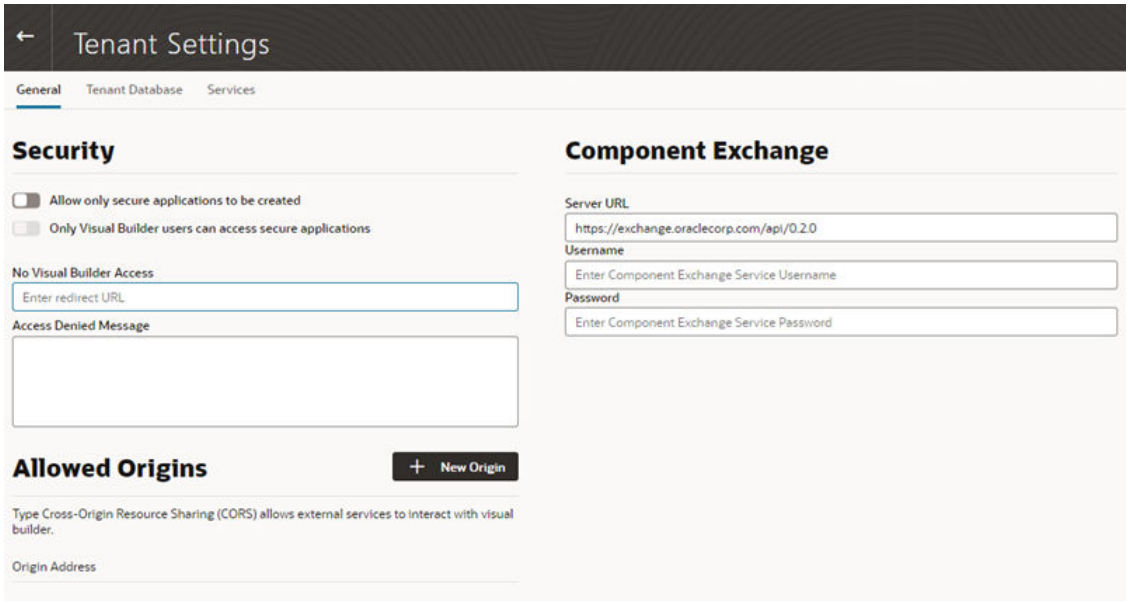
To open an instance's Tenant Settings page:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** ☰.
2. Click **Settings** in the main menu.

   If you are developing visual applications, open the main navigation pane on the Home page and select **Settings**.



The settings available for the instance are grouped in the General, Tenant Database and Services tabs on the page.



Choose Your Instance's Update Window

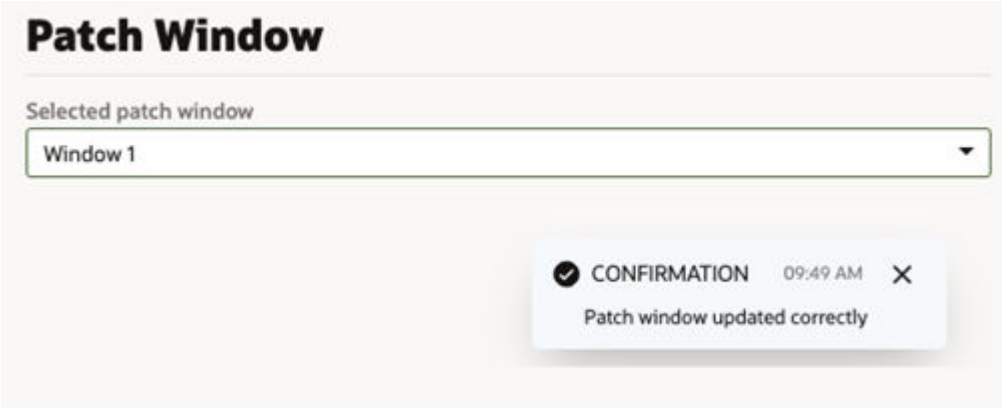
Functional updates for Visual Builder are provided in two windows, which are typically two weeks apart. You can select when you want an instance updated by selecting either `Window 1` or `Window 2`. We recommend that non-production instances be updated in the first window (`Window 1`) and production instances in the second window (`Window 2`). This allows you to test your applications in your test and development environments before the update is applied to your production environment.

> **Note**
>
> Oracle automatically sends notifications to the instance's account administrator each time it will be updated, confirming the instance's next update window. Once we send out the notification, it's too late to change your window for that update. If you do make a change, it won't be applied until the following update.

To set the update window option:

1. Open the Visual Builder instance's **Tenant Settings** editor.
2. In the General tab, select the window option in the Patch Window dropdown list.

   There are only two options: `Window 1` and `Window 2`. The default update window is `Window 1`.



Configure Security Options for Applications

Administrators can use the Security panel in the Tenant Settings page to require authentication for all applications in the instance.

When an administrator enables the **Allow only secure applications to be created** option, all published and staged applications in the instance will require user authentication. When the option is enabled, users must log in to access the applications in the instance, even if anonymous access is allowed in the application's settings. When the option is not enabled, applications can be created that allow access to anonymous users.

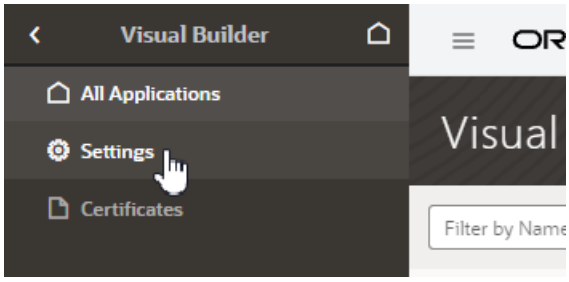
When an application has the default security settings, any user with a valid login can access the pages in an application. A developer can modify the default security settings to define the roles that can access applications, pages and components.

When the secure application option is enabled, an administrator can enable the **Only Visual Builder Users can access secure applications** option so that only Visual Builder users (those assigned the default Service User role) can access the staged and published applications in the instance. For example, this allows you to configure security so that users assigned the Visual Builder Developer role can access the designer, but can't access the published application and data because they are not assigned the Visual Builder Service User role.

An administrator can also use IDCS roles when configuring the instance's security so that a user's access is limited to just the secure applications. Users assigned the selected IDCS role would be able to access the applications, but would be prevented from accessing Visual Builder or Oracle Integration resources external to the application, such as other Oracle Integration integrations.

To configure the security options for all applications in the instance:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** ☰.
2. Click **Settings** in the navigation menu to open Tenant Settings.

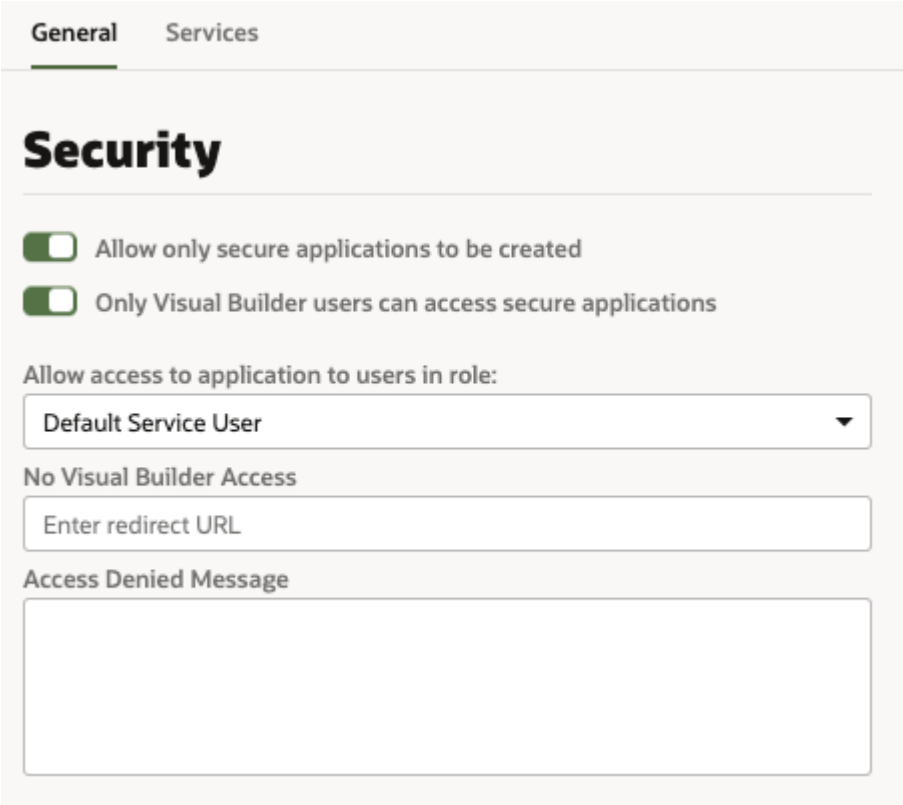


3. In the **Security** panel, enable **Allow only secure applications to be created**.

   Anonymous users can't access the applications when this secure applications option is enabled.

4. Select the **Only Visual Builder Users can access secure applications** option if you want to allow only Visual Builder users (users assigned the Service User role) access to the applications.

   To change the users allowed to access the application to those assigned a specific IDCS role *instead of* those assigned the default Service User role, select the IDCS role in the dropdown list under **Allow access to application to users in role**. This option is only available when both of the other security options are enabled.



5. Specify what users denied access to the secure application will see:
   a. Enter the URL you want the users redirected to when they can't access the app.
   b. Enter an Access Denied message that they will see when denied access to a page in the app.

Assign Roles for Users to Access an Application

Administrators must assign roles to users, so they have the permissions required to access Visual Builder applications.

Privileges associated with a user role determine what tasks users assigned those roles can perform. See What Users Can Do in Visual Builder by Role and Manage Access and Assign Roles in *Provisioning and Administering Oracle Integration 3*.

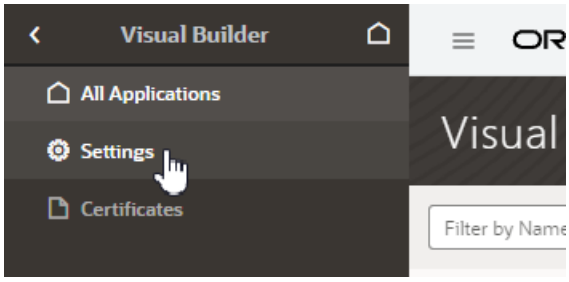
Set Page Messages for Access Denied Errors

Administrators can use the instance's settings page to specify a URL that users are navigated to when they are denied access to an application or page.

Authenticated users might see an Access Denied page or message when they attempt to access an application or page in an application that their user role is not permitted to access. Administrators can set the default page or message that users see when they are denied access to an application or page. Access Denied messages that are set at the application level in the General Settings of an application will override messages set in the instance's settings page. The default Access Denied page and message is used if the message options in this panel are not set.

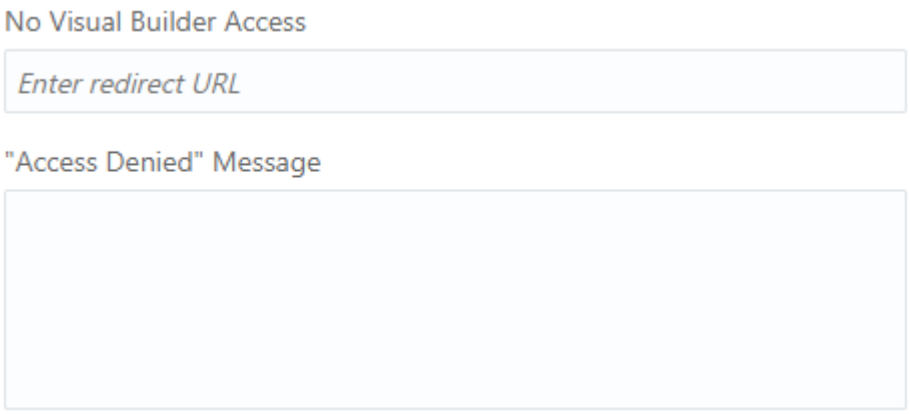
To specify an Access Denied page or message for applications in the instance:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** ☰.
2. Click **Settings** in the navigation menu to open Tenant Settings.



3. In the **Security** panel, type a URL that users are directed to when denied access to an application.

   The URL that you specify is used as the Access Denied page for all applications in the instance and should be accessible to users who are not logged in.



> **Note**
>
> If you are configuring settings for classic applications, the Access Denied settings are set in the **Messages** panel.

4. Type the message that you want users to see when they are denied access to a page.

   The message that you enter will be displayed in the Access Denied page for all applications in the instance except for those where a message was set at the application level in the application's General Settings page.

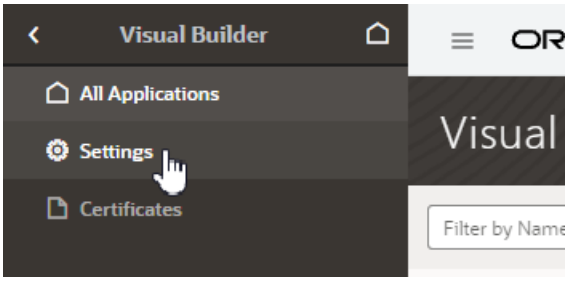
Allow Other Domains Access to Services

Use the Global Settings page to specify the domains that are permitted to interact with services in your instance.

Cross-Origin Resource Sharing (CORS) is a mechanism that enables you to specify the domains that are allowed to exchange data with applications in your instance. By default, incoming requests from domains not on your instance's list of allowed origins are blocked from accessing application resources.

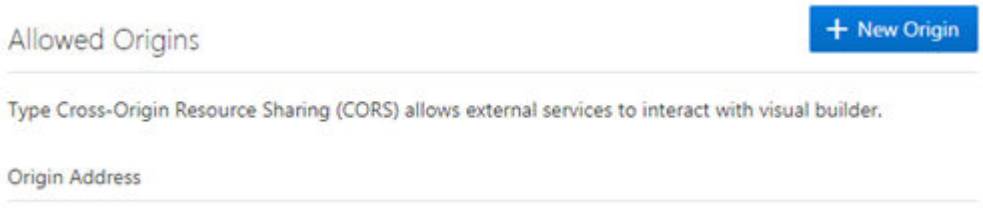
To add a domain to the list of allowed origins:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** ☰.
2. Click **Settings** in the navigation menu to open Tenant Settings.



3. In the **Allowed Origins** panel, click **New Origin** and type the URL of the domain that you want to allow. Click **Submit**.

The URL must be a fully-qualified domain, meaning it must contain `http://` or `https://`, for example, `https://myoracle.cloud.service`. You must explicitly enter each fully-qualified domain that you want to allow. To allow both `http://` and `https://` connections from a domain, you would need to add both domains (`https://myoracle.cloud.service` and `http://myoracle.cloud.service`).



The Allowed Origins panel lists all origins that are permitted to retrieve information from the instance.

Allow Your Instance to Access Services

If your Visual Builder instance needs to access an external service, your instance needs to be included in the service's allowlist (formerly a whitelist).

A service typically uses an Access Control List (ACL), called an allowlist, to restrict the networks and services that are allowed to access it. Only users from an IP address or Virtual Cloud Network (VCN) on the allowlist are allowed access to the service. The allowlist restrictions are in addition to the standard authorization mechanisms, such as user credentials, which are always in place.

Any Visual Builder instance that requires access to an external service, such as a REST web service, must be on the external service's allowlist. To get on a web service's allowlist, you'll need to work with the web service's administrator to add an ACL access rule for your VB instance. This may require filing a Service Request with the web service's administrator. You'll typically only need to do this when creating a new VB instance that will require access to a service, or when you plan to start using a new service in a VB instance. A VB instance can be added to an allowlist at any time, even before the instance has been created.

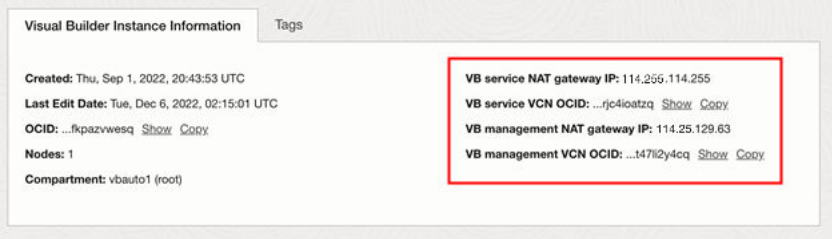
Depending on the location and type of the service your VB instance needs to access, you'll need to provide the service's administrator with:

- the Visual Builder *service* VCN,
- the Oracle Cloud ID (OCID) of the Visual Builder service VCN, or
- the NAT gateway IP address of the Visual Builder service VCN.

A VB instance's service VCN, OCID and NAT gateway IP address are determined by the instance's *region*. For example, `iad-vb-isovcn` is the VB service VCN for instances in the Ashburn region. For details on what these are, see Overview of VCNs and Subnets and NAT Gateway in the *OCI Documentation*.

> **Note**
>
> Visual Builder instances that use an Oracle DB service (ATP, DBaaS) will **also** have a VB *management* VCN. The VB management VCN OCID or NAT IP **must also** be added to the service's allowlist. Access from the VB management VCN is required so that schemas related to the VB service can be updated, for example, when patches or updates are applied to the instance.

You can view an instance's VB service NAT gateway IP and VCN OCID in the instance's Networking tab in the OCI Console. If the instance also has a VB management NAT gateway IP and VCN OCID, they will also be displayed in the tab:



The instance details you need to provide in the Service Request will depend upon the location and type of the service your instance needs to access:

- For a REST web service located in Oracle Service Network (OSN) (such as ORDS), provide:
  - the VB *service* VCN OCID

  The service administrator needs to configure one access rule, to allow access from the VB runtime service VCN.
- For an autonomous database located in OSN, like ATP, provide:
  - the VB *service* VCN OCID, and
  - the VB *management* VCN OCID

  The service administrator needs to configure two access rules, to allow access from the VB runtime service VCN and the VB management VCN.
- For an external REST web service, provide:
  - the NAT gateway IP address for the VB *service* VCN

  The service administrator needs to configure one access rule, to allow access from the IP address of the NAT gateway of the VB runtime service.

  An access rule configured for the NAT gateway is used when the service is not in the same region and OSN as your instance.
- For an external DBaaS database, provide:
  - the NAT gateway IP address for the VB *service* VCN
  - the NAT gateway IP address for the VB *management* VCN

  The service administrator needs to configure two access rules, to allow access from the VB runtime service VCN NAT gateway and the VB management VCN NAT gateway.

Access rules configured for the NAT gateways are used when the service is not in the same region and OSN as your instance.

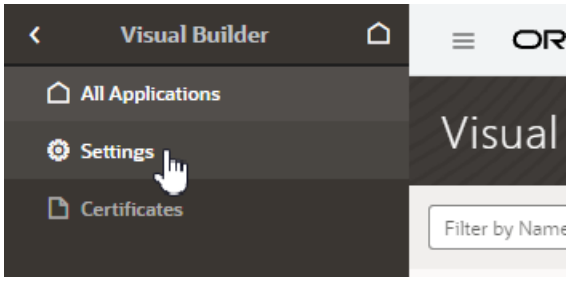
Inspect Database Usage

An administrator can view how much space in the tenant's database is being consumed by each of the tenant's applications.

The capacity of the tenant's database is 5GB, so by viewing the database usage you can see how much of the database's capacity remains. If you require more than 5GB of storage, you can Switch to Your Own Oracle DB Instance.

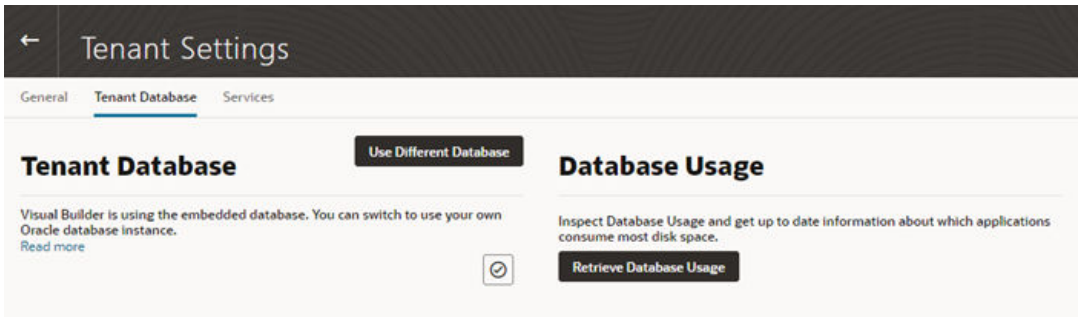
To inspect your instance's database usage:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** ☰.
2. Click **Settings** in the navigation menu to open Tenant Settings.

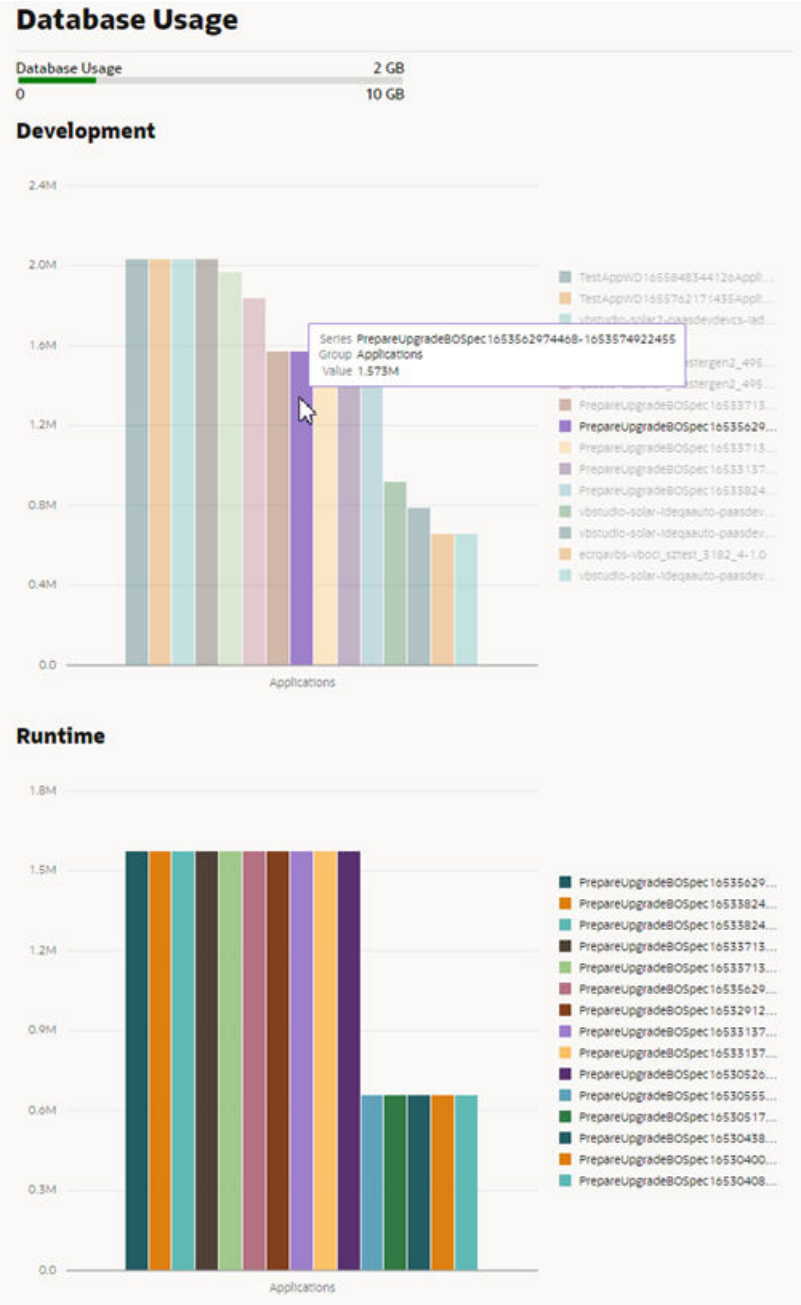


3. Open the Tenant Database tab.

   The Tenant Database has two panels: Tenant Database and Database Usage.



4. Click **Retrieve Database Usage** in the Database Usage pane.



The Database Usage meter in the panel shows how much of the database's capacity is currently used. The data usage is rendered in two charts:

- The Development chart shows the space used for storing the business objects in each application in the design-time.
- The Runtime chart shows the space used for each of the staged and published apps.

Switch to Your Own Oracle DB Instance

The database provisioned with your Visual Builder instance is used to store data for your business objects and your app's metadata, but this database has a 5GB limit and you can't access the data in the objects using regular SQL.

If the 5GB limit is insufficient for your tenant schema, you can configure your instance to use an Oracle DB instance that has more space instead of the default database. You can connect to an Oracle DBaaS or Autonomous Transaction Processing (ATP) database instance. Using an ATP database will give you more space and direct SQL access to the objects VB creates. You can also use a Free Forever Oracle ATP, which provides 20GB of storage for free.

To use a different Oracle DB instance, you use a wizard in the Tenant Settings to create a connection to the database instance and export the applications stored in the tenant's current database.

If you decide to use JDBC to connect to your DBaaS instance, you must include the privileges required to enable the ADMIN user to create a tenant schema. The following SQL shows the grants that are needed:

```
CREATE USER [adminuser] IDENTIFIED BY [password];
GRANT CONNECT, RESOURCE, DBA TO [adminuser];

GRANT SELECT ON SYS.DBA_PROFILES TO [adminuser] WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_USERS TO [adminuser] WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_DATA_FILES TO [adminuser] WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_SEGMENTS TO [adminuser] WITH GRANT OPTION;
```

If you decide to use ATP, you'll need to include the `wallet.zip` file in the wizard in addition to the connection info. You might want to create a new ATP ADMIN user with the correct admin privileges. The following SQL statement shows how to create a second ATP ADMIN user in SQL*Plus or SQL Developer.

```
DROP USER [adminuser] CASCADE;
CREATE USER [adminuser] IDENTIFIED BY [password];
GRANT CREATE USER, ALTER USER, DROP USER, CREATE PROFILE TO [adminuser] WITH
ADMIN OPTION;
GRANT CONNECT TO [adminuser] WITH ADMIN OPTION;
GRANT RESOURCE TO [adminuser] WITH ADMIN OPTION;
GRANT CREATE SEQUENCE, CREATE OPERATOR, CREATE SESSION,ALTER SESSION, CREATE
PROCEDURE, CREATE VIEW, CREATE JOB,CREATE DIMENSION,CREATE INDEXTYPE,CREATE
TYPE,CREATE TRIGGER,CREATE TABLE,CREATE PROFILE TO [adminuser] WITH ADMIN
OPTION;
GRANT UNLIMITED TABLESPACE TO [adminuser] WITH ADMIN OPTION;
GRANT SELECT ON SYS.DBA_PROFILES TO [adminuser] WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_USERS TO [adminuser] WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_DATA_FILES TO [adminuser] WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_SEGMENTS TO [adminuser] WITH GRANT OPTION;
```

> **Note**
>
> If you get an error `Failed to verify the target database` in the Change Tenant Database dialog when switching the database, it might be because you don't have the required privileges, or because the database is not reachable. (Visual Builder cannot reach databases in private subnets, except when Visual Builder is provisioned as a private endpoint in the same private subnet as the database.)
>
> If you see the error, confirm that the ADMIN user (`adminuser`) has the required privileges. You might also need to assign the SYSOPER and SYSDBA roles to the ADMIN user:
>
> ```
> GRANT SYSOPER, SYSDBA TO [adminuser];
> ```
>
> You can run the following query to confirm the ADMIN user has the necessary privileges:
>
> ```
> select * from v$pwfile_users;
> ```

In the wizard you need to select and export all the applications in your instance that you want to keep. After confirming that your instance is using the new database instance, you must import the exported applications into Visual Builder to save them in the new database instance.

> **Note**
>
> If you have live applications already on the instance:
>
> - Before switching to a new database, make sure to backup the data in their business objects using the export options in the Visual Builder data manager. You'll then be able to import that data back into the new apps you'll create from the application archives you export in the wizard.
> - Lock the live applications before changing the settings of your instance's database to prevent users from using them during the migration process. You can unlock the applications when the migration process is finished. You lock and unlock live applications in the Application Options menu on the Visual Builder Home page. See Manage an Application in *Developing Applications with Oracle Visual Builder in Oracle Integration 3*.

To switch to a different Oracle DB instance:

1. Open the Tenant Database tab.

   You can open your instance's Tenant Database tab from the instance Home Page, or by entering the URL directly in the browser window. It might be quicker to enter the URL directly if there is a problem loading the Home Page, for example, if the wallet is expired.

   - To open the Tenant Database tab using a URL, type the following in the browser's URL field:

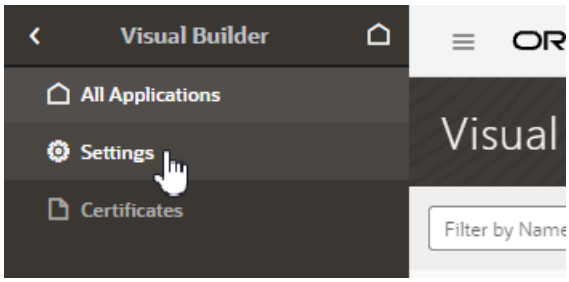
     ```
     https://<instance-url>/ic/builder?root=settings&settingsSection=tenant-database
     ```

In the URL above, replace <*instance-url*> with your instance's URL.

- To open the Tenant Database tab from the Home Page:

  a. On the Visual Builder Home Page, click **Navigation Menu** in the upper-left corner of the Visual Builder title bar.

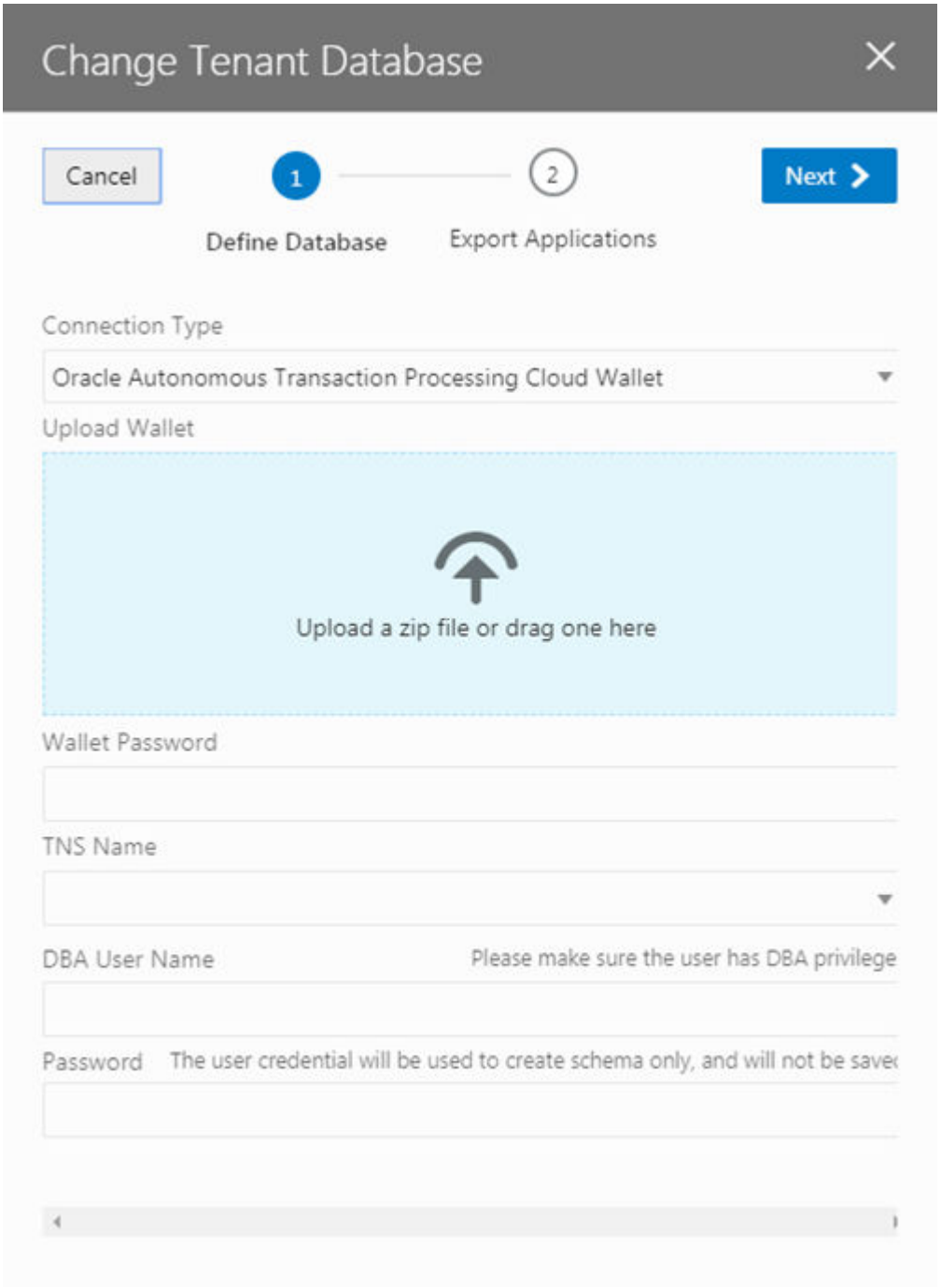
  b. Click **Settings** in the navigation menu to open Tenant Settings.



  c. Open the Tenant Database tab.

2. In the Tenant Database tab, click **Use Different Database** in the Tenant Database panel to open the Change Tenant Database wizard.

   In the Change Tenant Database wizard you supply the details for the connection to your Oracle DB instance.



3. Select a Connection Type in the drop-down list.

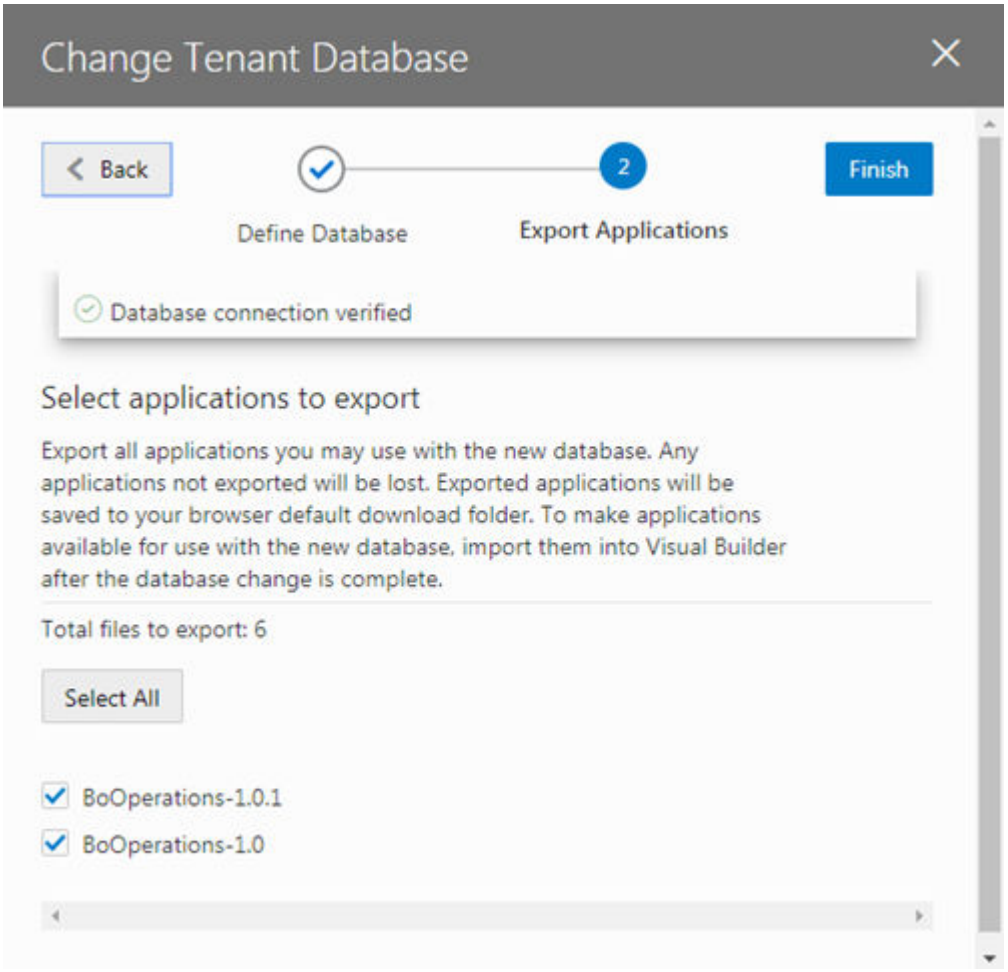
   You can connect to your Oracle DB instance using either JDBC or an ATP wallet.

4. Provide the details for connecting to your database. Click **Next**.

   The details you need to provide will depend upon the type of connection you selected.

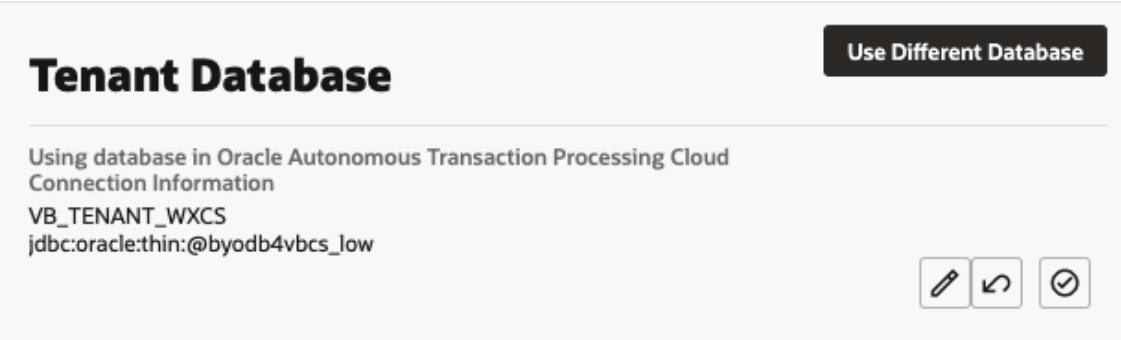
5. Select all the applications that you want to export. Click **Finish**.

   You must select and export all the applications that you want to keep. Any applications that are not exported will be lost.



When you click Finish, the applications that you selected are downloaded to your local file system. Exported application archives include the details about the application's user roles, and they will be available when you re-import your app into the new database.

After switching the database, the Tenant Database pane displays the connection information for your tenant's database. In the following image you can see that the instance is now using an Autonomous Transaction Processing (ATP) database instance.



> **Note**
>
> If you decide to revert back to using the embedded database, you can click  in the Tenant Database pane. You'll be prompted to confirm that you want to switch to using the instance's embedded database instead of the current one.
>
> When you revert to using the embedded database, the visual applications in your current database are not transferred automatically. You need to export the apps you want to keep before switching the database, and then import them into the embedded database.

Visual Builder automatically manages the schemas and tables it uses for apps and business objects in your new DB, so you don't need to do anything further.

If you would like to access the business objects using SQL, you'll find that VB creates users/schemas with names that start with `VB_` followed by randomly generated strings. By examining the data dictionary you'll be able to find the users that represent specific apps. Note that you'll see separate schemas for dev, stage, and published instances of an app. The schemas for the dev and test instances will be re-created with different names with every new version of the app that you create. If you want to prevent the schema name for a published app from changing, when you publish new versions of the app you should choose the option to not replace the data.

> **Note**
>
> Instead of having Visual Builder create and manage schemas, you can make a schema that already exists in your database available to applications, so developers can create business objects based on existing DB tables and views. If you choose to use your own schema, make sure you understand the requirements and limitations when using your own schema. For details, see Switch to Your Own Database Schema for Business Objects in *Developing Applications with Oracle Visual Builder*.
>
> If you use your own schema, only one schema is used for the app's dev, staged, and published instances. See Make Schemas in an Oracle DB Instance Available to Applications.

Switch From One ATP Database to Another

It's not possible to switch from one ATP database to another directly, so you'll first need to switch from your ATP database back to the embedded database. You can then use the Change Tenant Database wizard to switch from the embedded database to the new ATP database.

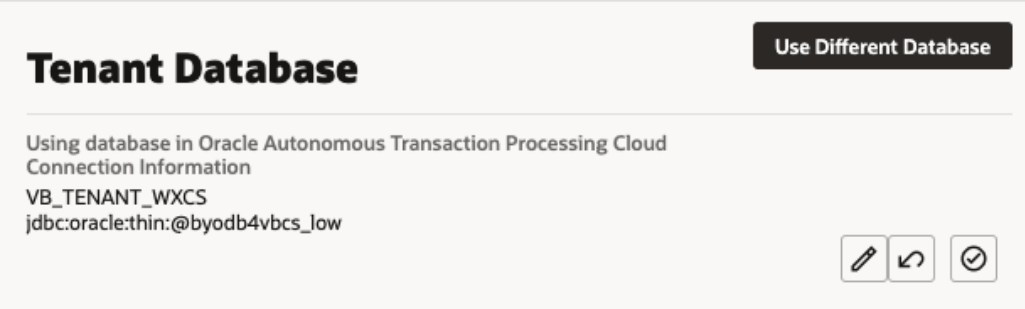
1. In Visual Builder, export each of your visual applications and save them to your local system.

   For details, see Export a Visual Application.

2. Revert to the embedded database.

   a. Open the instance's Tenant Settings page, and then open the Tenant Database tab.

   b. Click  in the Tenant Database pane to revert to the embedded database.



c. When prompted, confirm that you want to switch to the embedded database.

3. Switch to the new ATP database.

   Follow the steps in Switch to Your Own Oracle DB Instance above to switch from the embedded database to your new ATP database.

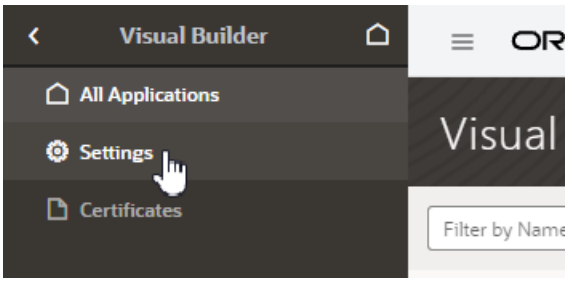
4. Import the applications you saved to your local system into the new ATP database.

Make Schemas in an Oracle DB Instance Available to Applications

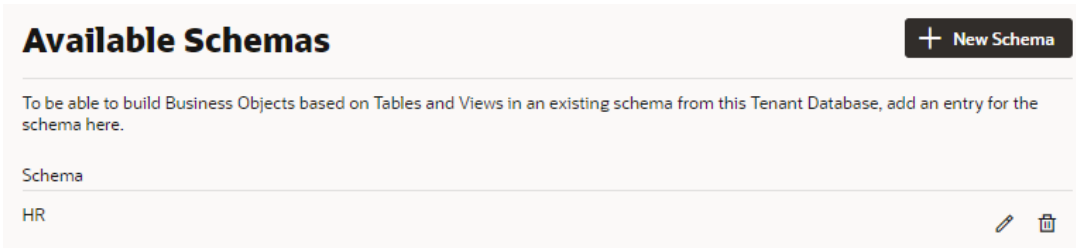
When you connect an Oracle database instance with your Visual Builder instance, application developers can use schemas predefined in the tenant database to create business objects based on existing tables and views for an application. But for developers to access these schemas, you'll first need to make them available to applications.

To make a tenant database's existing schema available to applications:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** ☰.
2. Click **Settings** in the navigation menu to open Tenant Settings.



3. Click **+ New Schema** in the Available Schemas panel.



4. Enter a name for the schema and click the check mark icon.

After the schema is added, you can edit its name or delete it entirely, but remember any changes you make might break applications that use the schema.

Schema that exists in the tenant database and has been added to the list of available schemas will become available for selection in an application's Settings editor (under Schema Selection in the Business Objects tab).

Update Your ATP Wallet and Reset an Expired Password

If you switch to your own Oracle DB instance and the credentials you use to access the instance expire, you can use the Update Tenant Database Connection dialog box to update your ATP wallet and renew expired credentials.

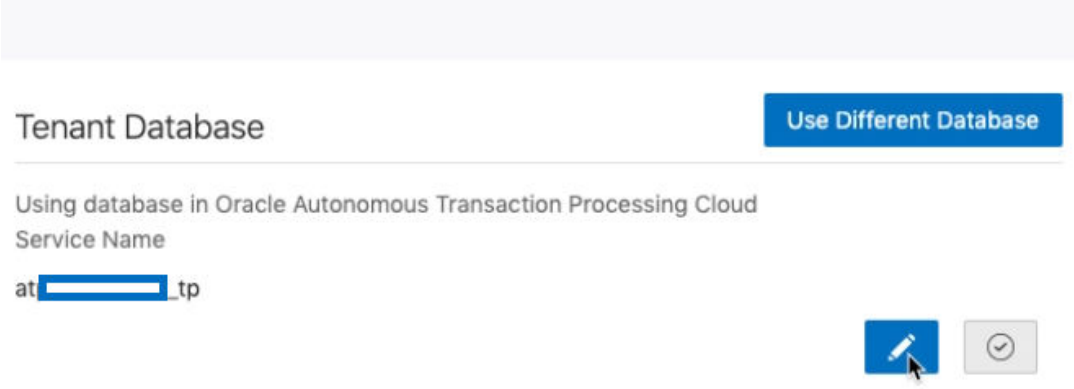
To regenerate the expired values, you need to provide the ADMIN user credentials that you provided when you first switched to your own Oracle DB instance. Visual Builder uses the ADMIN user credentials to generate new Visual Builder tenant credentials to replace the expired credentials. Visual Builder does not store the ADMIN user credentials that you supply.

1. Open the General tab of the instance's Tenant Settings page.

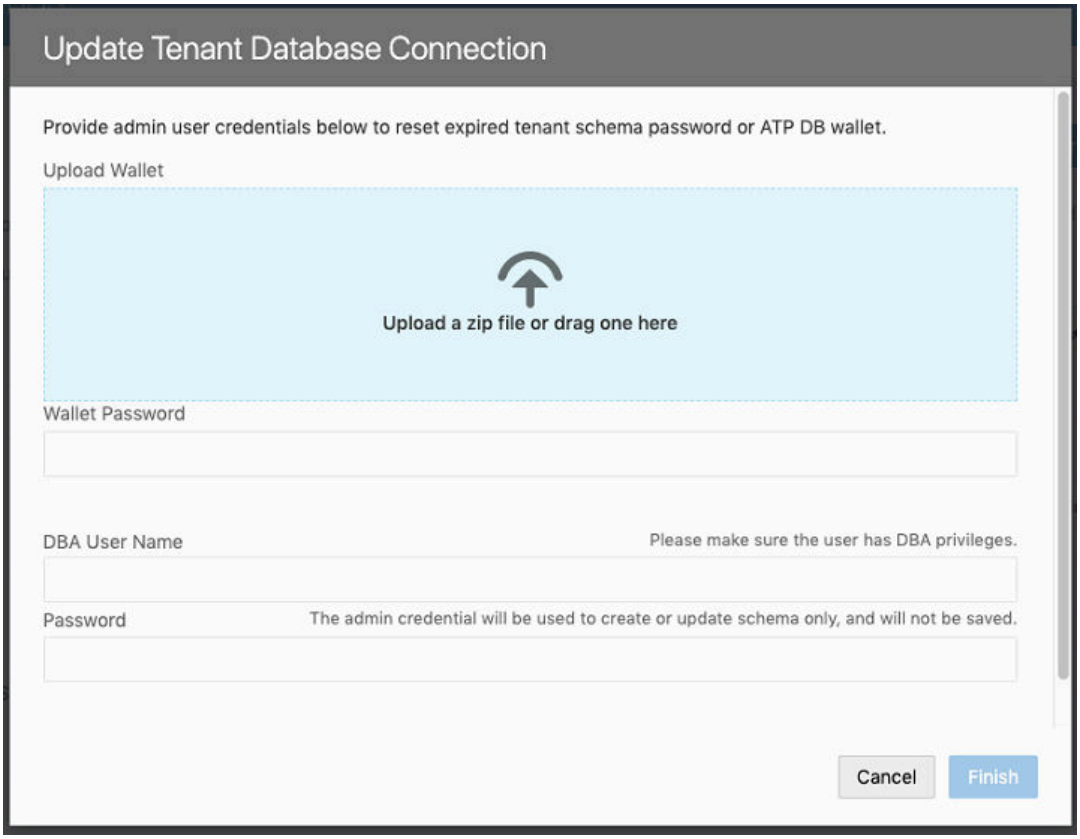
   If you cannot navigate to the Tenant Settings page from the navigation menu, you can open the page directly by entering the page's URL in the browser. The URL will be similar to

   `https://<Instance-URL>/ic/builder/?root=settings&settingsSection=tenant-database`.

2. In the Tenant Database field, click the Edit icon to open the Update Tenant Database Connection wizard.



3. In the Update Tenant Database Connection wizard:
   - Reset expired credentials by supplying the ADMIN user credentials and ATP wallet that Visual Builder will use, or
   - Update the wallet for your Oracle DB instance by uploading the new ATP wallet.



4. Click **Finish**.

Use ATP Database with Cross-Region Autonomous Data Guard for Disaster Recovery

The Cross-Region Autonomous Data Guard feature of Oracle Autonomous Database provides data protection and disaster recovery for your database.

If you enable Autonomous Data Guard with a cross-region standby database on your ATP, you can manually construct a new ATP wallet with single connection string containing both the primary and the standby database hostnames. This way, when you failover to a standby ATP database, Visual Builder will be automatically retry and connect to the ATP in the standby region, which then becomes the active database.

For details on how to manually construct a wallet that contains both the primary and the remote database connection strings, see Cross-Region Autonomous Data Guard Notes.

A database connection string in the wallet identifying the primary and standby hostnames might look something like this:

```
atpsample_tp = (description_list=
 (failover=on) (load_balance=off)
 (description= (retry_count=5)(retry_delay=3)(address=(protocol=tcps)
(port=1522)(host=adb.us-ashburn-1.oraclecloud.com))
(connect_data=(service_name=g0de0f6e24ce255_atpsample_tp_tp.adb.oraclecloud.co
m))(security=(ssl_server_dn_match=yes)))
 (description= (retry_count=5)(retry_delay=3)(address=(protocol=tcps)
(port=1522)(host=adb.ca-montreal-1.oraclecloud.com))
```

```
(connect_data=(service_name=g0de0f6e24ce255_atpsample_tp_tp.adb.oraclecloud.co
m))(security=(ssl_server_dn_match=yes))))
```

> **Note**
>
> When constructing the connection string, change the `retry_count` to "5", instead of the default "20".

Access an ATP Database Configured as a Private Endpoint

If you want to use an ATP database that is protected using a private endpoint (ATP-PE), you can configure the database instance to allow a public Visual Builder instance to connect to the database directly, without requiring a public load balancer.

Similarly, if you are already using an ATP database configured to use a public endpoint, and you want to switch to ATP-PE, you need to update the allowlists in the ATP-PE settings and the VB instance settings to allow connections to the database. For more on adding a VB instance to allowlists, see Allow Your Instance to Access Services.

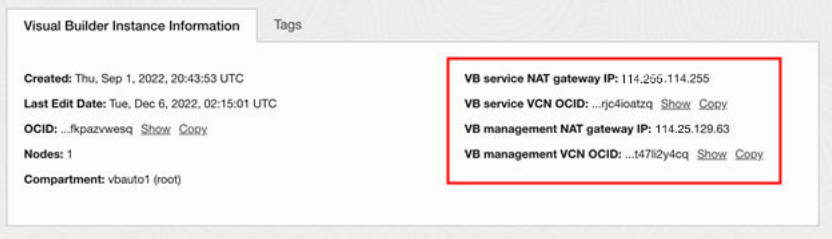
To connect your public VB instance to an ATP-PE instance:

1. On the Visual Builder Instances page, find the instance you want to work with and open its details page.
2. Collect the required details about your Visual Builder instance.

   To configure the access list in ATP-PE, you'll need to provide Visual Builder network gateway details:

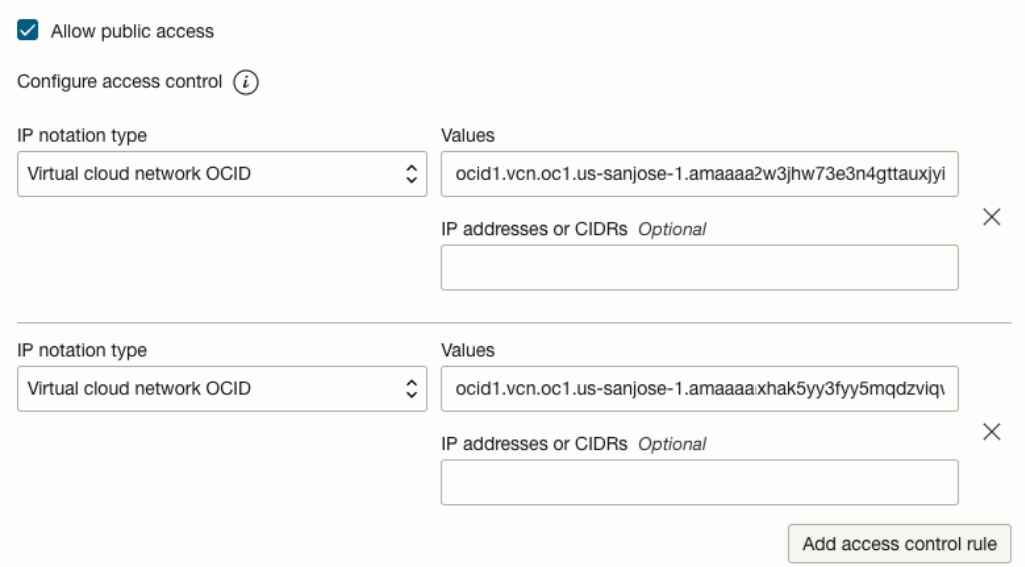
   - If VB and ATP-PE are in the same OCI region, you need theVB *service* VCN OCID and the VB *management* VCN OCID.
   - If VB and ATP-PE are in different OCI regions, you need the *service* outbound IP and the *management* outbound IP

   You can view an instance's VB service NAT gateway IP and VCN OCID in the instance's Visual Builder Instance Information tab in the OCI console. If the instance also has a VB management NAT gateway IP and VCN OCID, they will also be displayed in the tab:



3. Configure the ATP-PE instance's access control rule.
   a. Open the ATP-PE instance's details page.
   b. Select **More actions**, then select **Update network access**.
   c. In the Update network access panel, select **Allow public access** in the Private endpoint access pane.
   d. Enter the required VB VCN OCIDs or IP addresses. Click **Add access control rule**.

For example, if the VB and ATP-PE instances are in the same OCI region, you should select Virtual cloud network OCID in the two IP notation type drop-down lists, and enter the two required VCN OCIDs in the Values fields:



For details on the ATP access control settings, see Use a Private Endpoint with Public Access Allowed and Configure Private Endpoint Advanced Options in the Oracle Autonomous Database documentation.

4. Download the ATP wallet.

   If you are switching from an ATP database to ATP-PE, you will need to download the updated ATP wallet.

5. File a Service Request to update the ATP connection string and wallet in the Visual Builder backend.

Connect to a Database From a Private Endpoint-Enabled Instance

If your Visual Builder instance was provisioned as a private endpoint, you might need to do some additional configuration when switching to an Oracle database instance.

> **Note**
>
> If the private endpoint-enabled VB instance and the database service are in different virtual cloud networks (VCNs), you will need to create private views inside the VCN private resolver so that both VCNs can resolve hosts and endpoints in the other VCN. For more information, see About the DNS Domains and Hostnames.

To use an Oracle database with your private endpoint-enabled VB instance:

1. Get the connection details for the database instance.

   If you are not using an ATP wallet, you need to use JDBC to connect to your Oracle DB instance, and you will need to use the quick URL connection string, which contains only the database's host name, as the URL. You cannot use the service's long URL connection string, which contains both the private IP address and host name.

2. Add the appropriate database port to your private endpoint-enabled VB instance's security list or NSG rules.

   Typically the port is 1521, but you need to confirm the correct port for your database instance.

   For more on configuring rules, see Configure Private Endpoint Advanced Network Options .

3. Switch to the Oracle DB instance.

   See Switch to Your Own Oracle DB Instance for the steps.

If you have problems connecting to the database, you can try to debug the problem by creating a compute instance in the private subnet, and then connecting to the database from the compute instance using the database's host name. If you can successfully connect to the database from a compute instance residing in the same private subnet, then connecting to the database from the Visual Builder instance will also work.

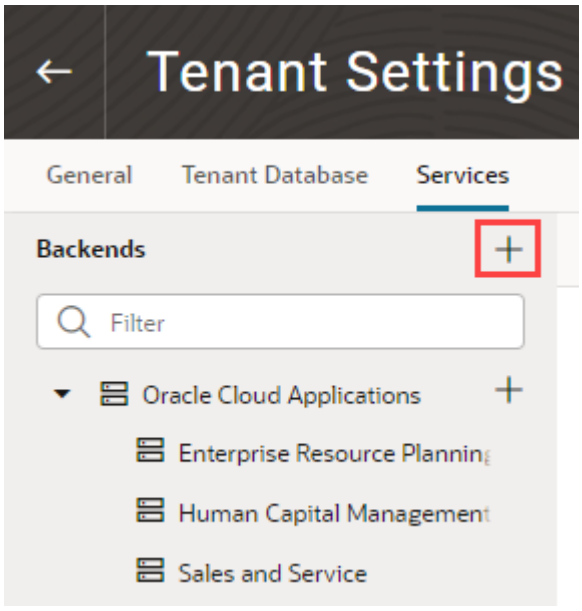
Add a Connection to Integration Applications

Administrators can use the Services tab in the Tenant Settings page to add a connection to an instance of Oracle Integration as a backend service.

If you are using multiple Visual Builder instances, for example, development and production instances, you might need to add connections to Oracle Integration in more than one instance.

To add a connection to an Oracle Integration instance:

1. Open the instance's Tenant Settings page.
2. In the Services tab, click **Create Backend** and choose **Integrations** in the Create Backend dialog.



3. In the dialog, type the Server URL of the backend service, configure other settings such as security as needed, and click **Create**.

See About Authentication and Connection Type in *Developing Applications with Oracle Visual Builder in Oracle Integration 3*.

Add a Connection to Oracle Cloud Applications

The list of REST services in the service catalog of a visual application is retrieved from an Oracle Cloud Applications backend service. Specify the instance URL of the Oracle Cloud Applications backend service in the Tenant Settings page.

All visual applications in the tenant will use the Oracle Cloud Applications instance URL specified in Tenant Settings, but a visual application can be configured to use a different Oracle Cloud Applications backend service by specifying a different instance URL in the Backends tab (which you access from the Navigator's Services tab). The tenant-level backend configuration is ignored if you or a visual application developer configures a different Oracle Cloud Applications backend service in a visual application's Backends tab.

The authentication choices available to configure a tenant-level Oracle Cloud Applications backend are:

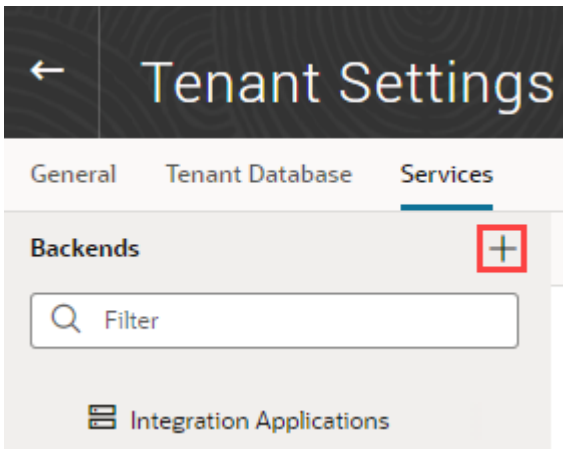
- Basic Auth: Uses a fixed username and password for authentication.
- Oracle Cloud Account: Needs federation between Oracle Cloud Applications and Visual Builder.
- Delegate Authentication (previously called Propagate Current User Identity): Same as Oracle Cloud Applications. That is, it needs federation between Oracle Cloud Applications and Visual Builder.
- None: This assumes your Oracle Cloud Applications REST API can be called without any authentication, which is not usually the case.

See About Authentication and Connection Type in *Developing Applications with Oracle Visual Builder in Oracle Integration 3*.

If the necessary prerequisites for setting a tenant-level Oracle Cloud Applications backend service are not available, then a visual application developer can set up a backend service at the visual application level where more options are available. Another option is for you (the service administrator) to configure the Oracle Cloud Applications backend with `None` and let the visual application developer override the authentication setting at the visual application level.

To specify an Oracle Cloud Applications service for the tenant:

1. Open the instance's Tenant Settings page.
2. In the Services tab, click **Create Backend**, then choose **Oracle Cloud Applications** in the Create Backend dialog.



When specifying the URL in the Tenant Settings, you (the service administrator) only need to provide the instance URL of the Oracle Cloud Applications backend service to retrieve the list of services.

3. In the dialog, type the Server URL of the backend service, and configure other settings, such as security, as needed.
4. (Optional) After you configure settings for the backend, add headers to the backend.

   Backend headers that you add will be applicable for any service connection to this backend, irrespective of the server or application profile that is used.
5. Click **Create**.

   Visual Builder automatically discovers the interfaceCatalogs endpoint of the Oracle Cloud Applications backend, which retrieves the list of services and their metadata. This endpoint is typically in the form:

   ```
   https://<My Oracle Cloud Applications Instance URL >/helpPortalApi/
   otherResources/latest/interfaceCatalogs
   ```

   This endpoint is publicly accessible without any authentication.

   If there is a problem creating the connection, verify the instance URL of the Oracle Cloud Applications instance.

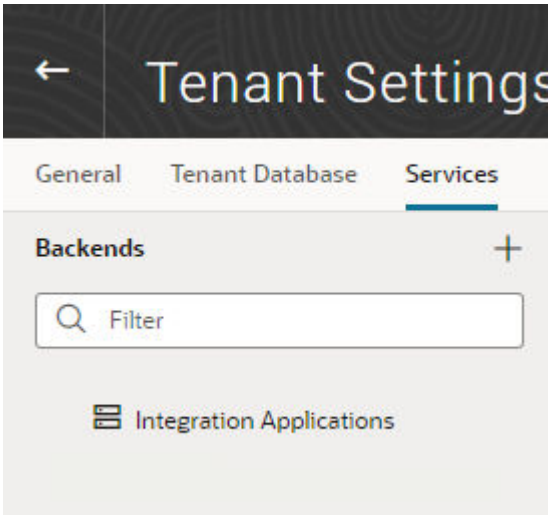
Add a Connection to Process Automation

Administrators can use the instance's Tenant Settings page to add a connection to an instance of Process Automation as a backend service.

If you are using multiple Visual Builder instances, for example, development and production instances, you might need to add connections to Process Automation in more than one instance.

To add a connection to a Process Automation instance:

1. Open the instance's Tenant Settings page.
2. In the Services tab, click **Create Backend** and choose **Process Automation** in the Create Backend dialog.



3. Enter the URL of the instance, configure other settings, such as security, as needed, and click **Create**.

Add a Connection to Process Cloud Service

Administrators can use the instance's Tenant Settings page to add a connection to an instance of Oracle Process Cloud Service as a backend service.

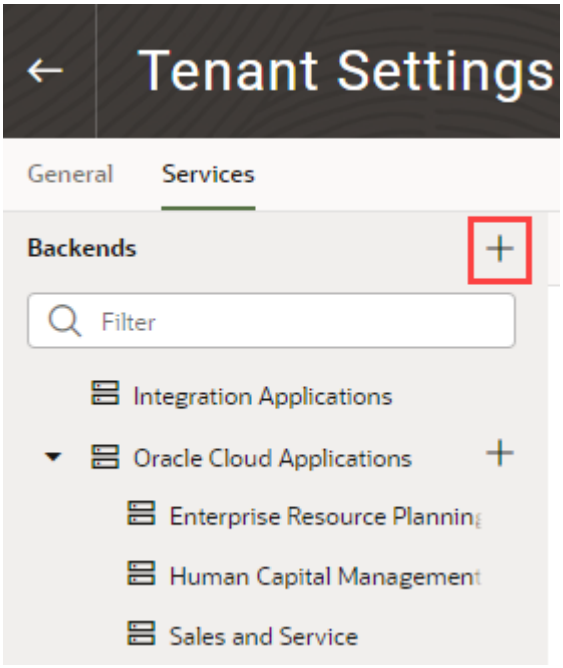
> **Note**
>
> Oracle Process Cloud Service, which is included in the Enterprise edition of Oracle Integration Gen 2, is now deprecated; for details, see Process Features.

To add a connection to an instance of Oracle Process Cloud Service as a backend service, the instance of Oracle Process Cloud Service should be co-hosted with Visual Builder because the authentication types that Visual Builder supports for this configuration is Oracle Cloud Account or Propagate Current User Identity. In most cases, this backend service (Oracle Process Cloud Service) will be preconfigured for your Visual Builder instance.

If you are using multiple Visual Builder instances, for example, development and production instances, you might need to add connections to Oracle Process Cloud Service in more than one instance.

To add a connection to an Oracle Process Cloud Service instance:

1. Open the instance's Tenant Settings page.
2. In the Services tab, click **Create Backend** and choose **Process** in the Create Backend dialog.



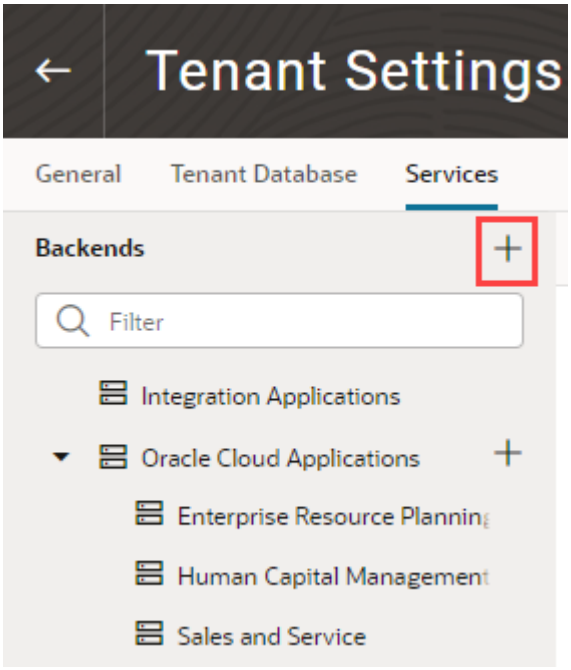
3. In the dialog, type the Server URL of the backend service, configure other settings, such as security, as needed, and click **Create**.

Add a Connection to a Custom Backend

You can create your own backend to map to a custom server other than the Oracle Integration, Process, and Oracle Cloud Applications backend services. You can create a custom backend with a free-form URL, or create a custom ADF backend when you know the Describe URL that points to an ADF Describe service.

To add a connection to a custom backend:

1. Open the instance's Tenant Settings page.
2. In the Services tab, click **Create Backend**.



3. In the Create Backend wizard, select the type of backend you want to create:
   - To create a backend with a free-form URL, click **Custom**.
   - To create a backend with the Describe URL of an ADF service, click **Custom ADF Describe**. Use this option only when your custom ADF Describe endpoint does not have any child backends.
4. In the Name field, enter a name and description for the custom backend.
5. Add headers to the backend. Backend headers that you add will be applicable for any service connection to this backend, irrespective of the server or application profile that is used.
6. Click **Next**.
7. Enter the instance URL for the custom backend, configure other settings, such as security, and click **Create**.

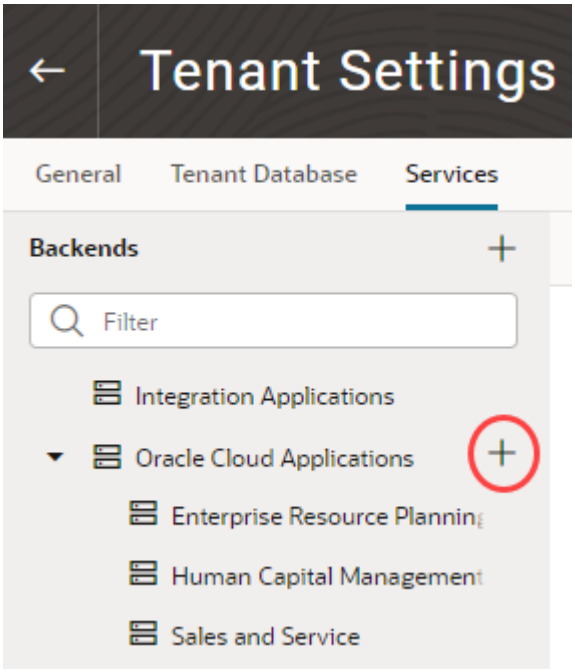
Create a Child Backend

You can create child backends to extend the functionality provided by the top-level Oracle Cloud Applications or custom backend registered to your Visual Builder instance.

Let's say your instance's Oracle Cloud Applications backend connects to an Oracle Cloud Applications instance that provides access to these service catalogs: Enterprise Resource Planning Supply Chain, Sales and Service, and Human Capital Management. Now if you want to access another catalog (say, Search), you can create a child backend to access the search service.
A child backend inherits the parent backend's definition, which you can override as required. Its server URL is derived from the top-level backend, with `vb-catalog://backends/` as the base URL. Continuing the Oracle Cloud Applications example, the Sales and Service child backend adds to the top-level Oracle Cloud Applications backend and has `vb-catalog://backends/fa/crmRestApi/resources` as its server URL.

Child backends can be created only for the Oracle Cloud Applications backend and custom backends that use a OpenAPI/Swagger service specification.

- To create a child backend for the Oracle Cloud Applications backend:
  1. Open the instance's Tenant Settings page.
  2. In the Services tab, click the **+** sign for the top-level Oracle Cloud Applications backend:

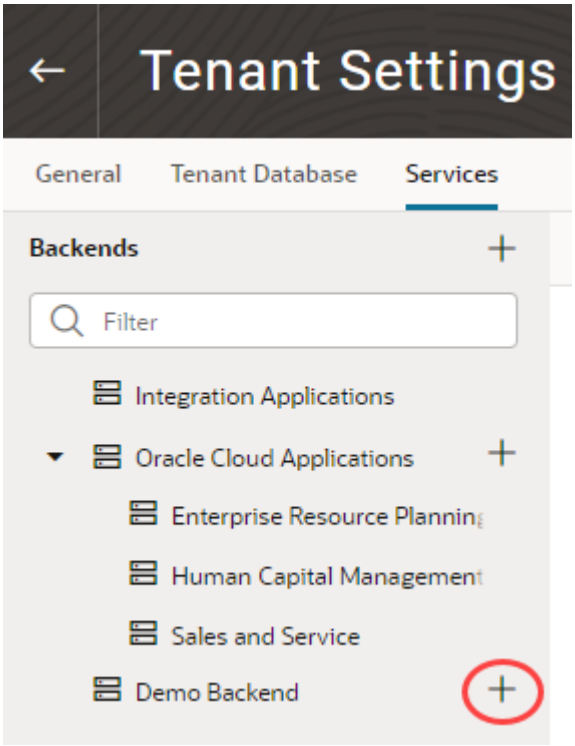


  3. Select **Custom ADF Describe** to create a backend with an ADF Describe URL. For backends not having a Describe URL, select **Custom**.
  4. Enter a name and description for the child backend. Optionally, add static headers.
  5. Click **Next**.
  6. Enter the instance URL for the child backend (for example, `vb-catalog://backends/fa/applcoreApi/search/`). The child backend's URL will usually start with `vb-catalog://backends/`*`oracle-cloud-app-BackendId`*.

     Tip

     To see the complete URL that the backend resolves to, click the **Detach** icon ( ).

  7. Enter other settings, such as security and headers.
  8. Click **Create**.
- To create a child backend for a top-level custom backend:
  1. Open the instance's Tenant Settings page.
  2. In the Services tab, click the **+** sign for a top-level custom backend:



3. Enter a name and description for the child backend. Optionally, add static headers.
4. Click **Next**.
5. Enter the instance URL for the child backend (for example, `vb-catalog://backends/demo/newdemo`). You can click the **Detach** icon to see the complete URL that the backend resolves to.
6. Enter other settings, such as security and headers.
7. Click **Create**.

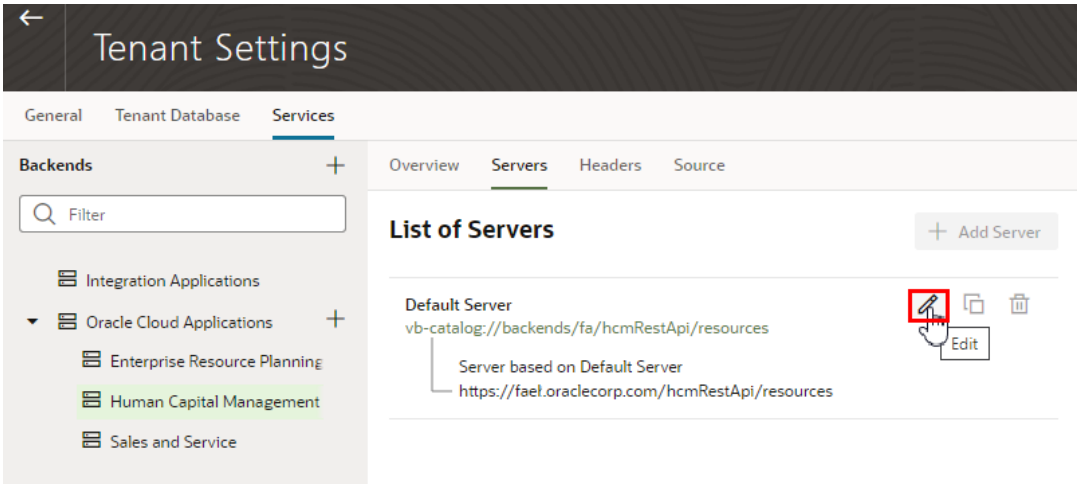
Edit Authentication for a Backend Service

Administrators can use the Services tab in the Tenant Settings page to edit the authentication and connection settings for the backend services available in the tenancy. Once a backend service is added, you can edit its details. In the case of child backends, you can also override the settings inherited from the backend service, for example, to allow connections to the child HCM backend to use basic authentication instead of using the Oracle Cloud Account authentication for logged-in users set at its parent backend.

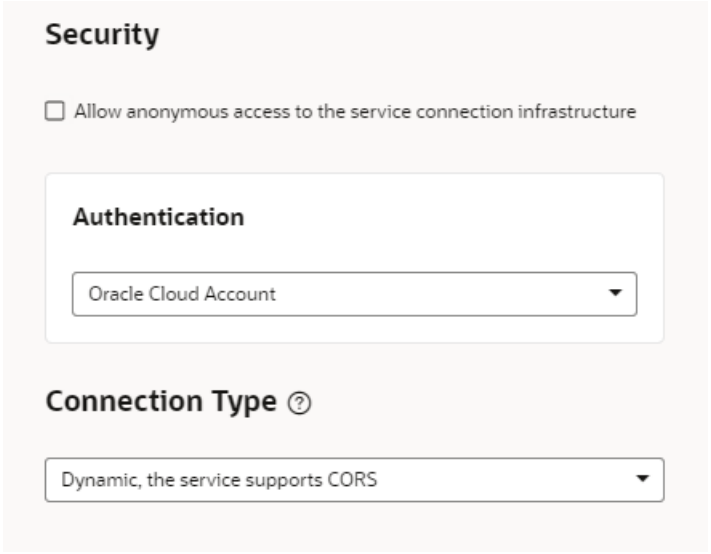
For a description of the authentication options, see About Authentication and Connection Type in .

To edit the connection details for a backend service:

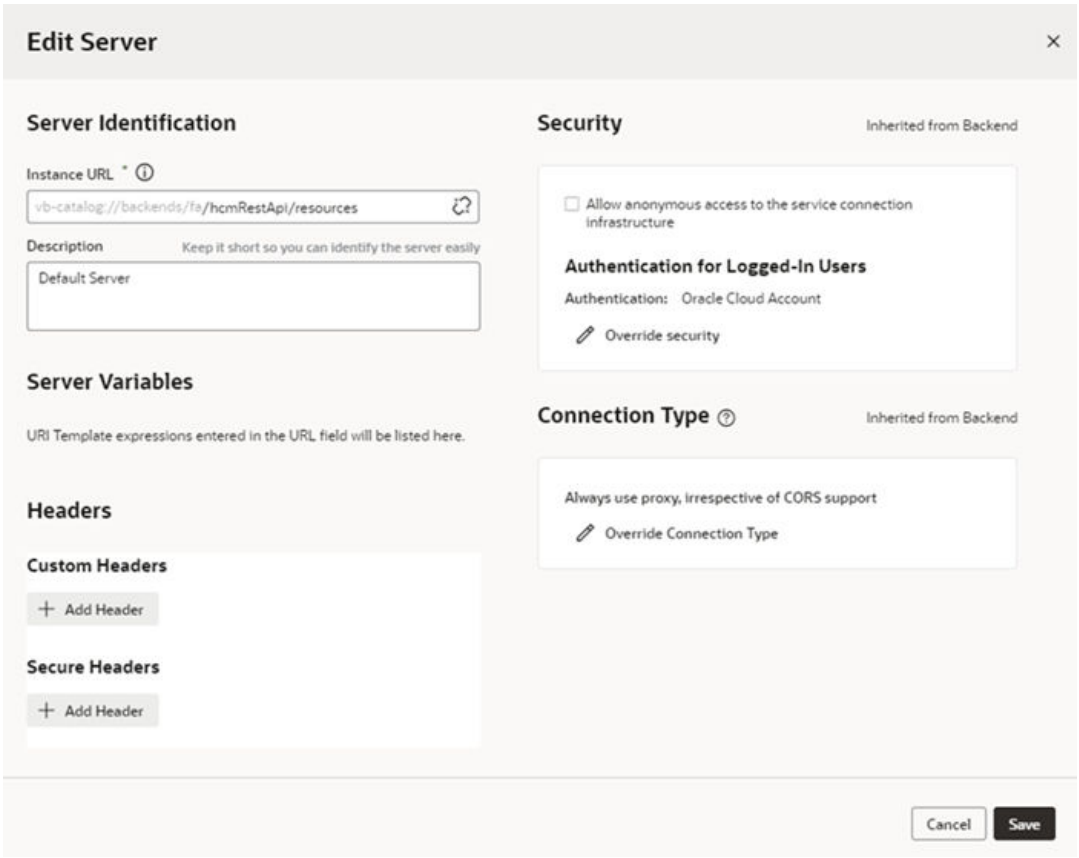
1. Open the instance's Tenant Settings page.
2. In the Services tab, select the backend you want to edit.
3. Open the Servers tab of your backend, and then click **Edit**.



4. In the Edit Server dialog box, edit the settings in the Security and Connection Type panes.

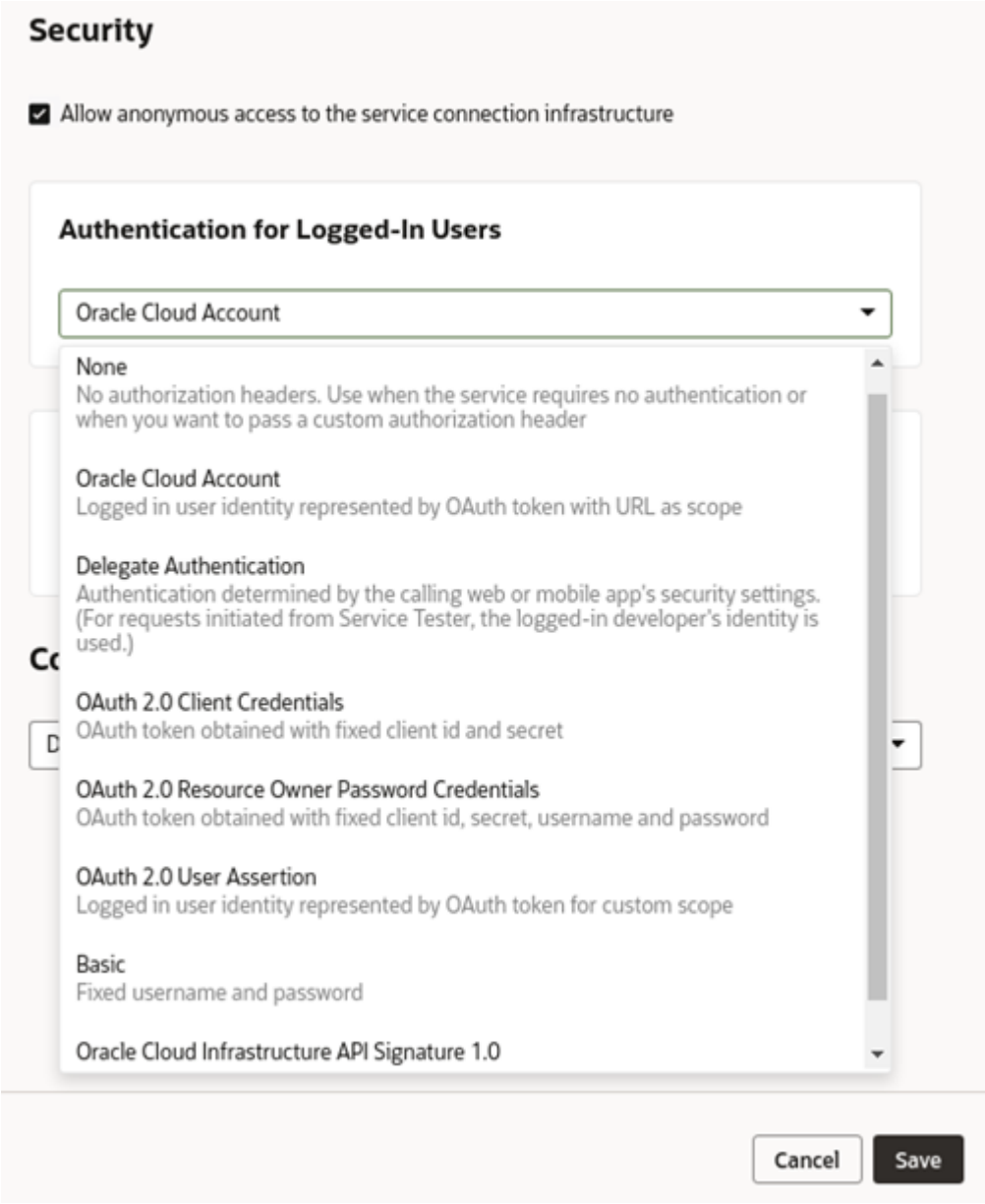


In the case of a child backend, the authentication and connection type details for your backend service are inherited from the parent service, so you'll need to override the settings. If you want to revert your changes for a child backend, you can click **Return to inherited** to restore the default inherited setting.



5. In the Security pane, select an authentication type in the dropdown list.

   If you are editing a child backend, you'll click **Override security** in the Security pane, and then select an authentication type in the dropdown list.



6. In the Connection Type pane, select a connection type in the dropdown list.

   In the case of a child backend, click **Override Connection Type** and then select the connection type.

7. Click **Save**.

Manage Self-signed Certificates

Administrators can use the Certificates page to upload and manage the self-signed certificates used by the instance to enable inbound and outbound SSL communications to a service's REST APIs

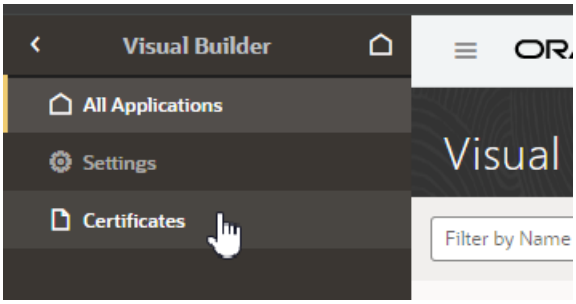
When creating connections to REST services that use self-signed certificates, you might need to add an API's certificate to your Visual Builder instance to validate SSL connections to that service. You can use the Certificates page to upload and remove certificate files (`.pem`) for

services. Uploading a service's certificate file to the keystore will allow all applications in the instance to communicate with that service. The Certificates page displays a list of certificates that have been added. You can click the Delete button in a row to remove the certificate.

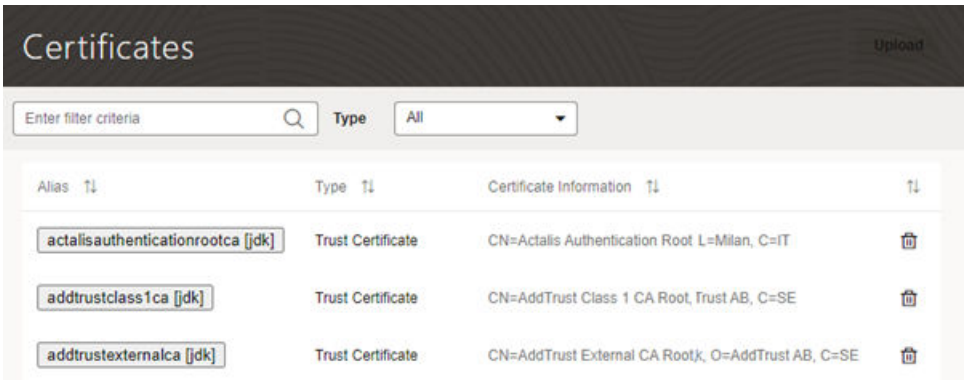
> **Note**
>
> Your staged or published apps might stop working if they use service connections with self-signed certificates and the certificates have expired. Any certificates issued after 2020-09-01T00:00:00.00Z will automatically expire 398 days after they have been issued. If your apps use certificates issued before 2020-09-01T00:00:00.00Z, the certificates will not expire, but you should update them with a newer certificate.
>
> To avoid disruptions, you should plan regular updates to refresh the self-signed certificates before they expire (for example, every 6 months). It's not recommended to use self-signed certificates in production apps.

To upload a self-signed certificate:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** ☰.
2. Click **Certificates** in the navigation menu to open the Certificates page.

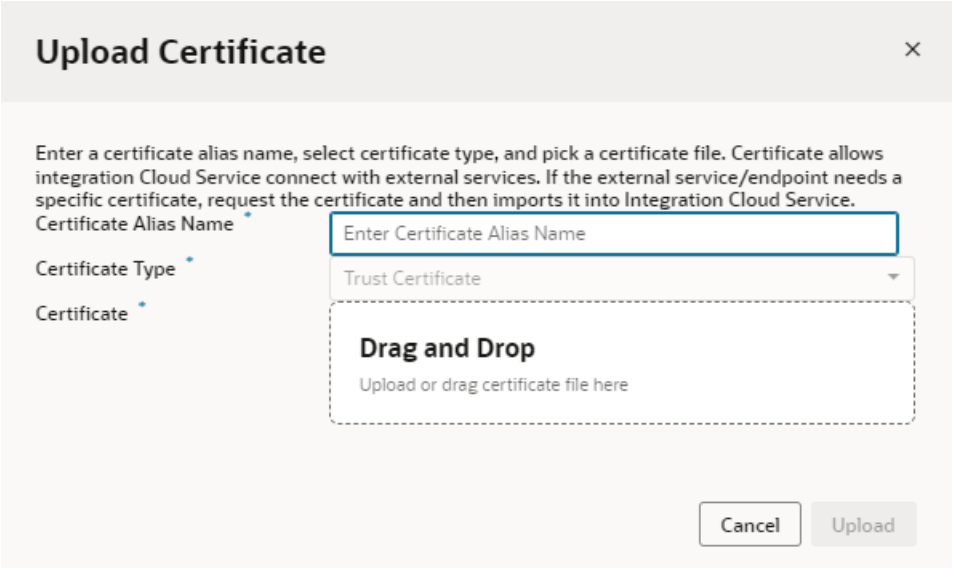


The Certificates page displays a list of the certificates already uploaded to the instance.



3. Click **Upload** to open the Upload Certificate dialog box.

   You use the Upload Certificate dialog box to create an alias for the certificate and upload the service's certificate file from your local system.



4. Type the alias in the Certificate Alias Name field.

   The alias is used to identify the certificate in the table in the Certificates page. The Certificate Type dropdown list is read-only because only Trust Certificates are supported.

5. Drag the certificate file from your local system into the upload target area, or click the upload target area to browse your local system.

6. Click **Upload** to add the certificate to the service keystore.

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 available in VB Studio. You can use these components in your visual applications, such as web components and application templates. Many of the components provided by Oracle can be installed from a component exchange.

To integrate a component exchange with a Visual Builder instance, you provide the exchange's URL and credentials in the Tenant Settings. The exchange can be a private exchange in a VB Studio project or one of the exchanges maintained by Oracle.

If your organization develops or uses proprietary components, these components can be published to a private exchange hosted by a VB Studio project. For example, if you have a web component designed to be used in applications in your tenant, you can set up your own exchange and use it to distribute the component to developers in the tenant. Additionally, components provided by Oracle are automatically available from all private component exchanges.

Oracle maintains two component exchanges containing components validated by Oracle that are publicly available to all developers. If you don't have a private exchange but you want to

give developers access to these Oracle components, you can add one of the following exchanges maintained by Oracle. If your instance is in the US, use the following details.

Field	Value
Service URL	`https://component-exchange-soctesting2-phx.developer.ocp.oraclecloud.com/component-exchange-soctesting2-phx/s/component-exchange-soctesting2-phx_compcatalog_30314/compcatalog/0.2.0`
Username	compcatalog.user
Password	k9fz-0Pw4x-q

If your instance is in Europe, use the following details.

Field	Value
Service URL	`https://component-exchange-soctesting4-fra.developer.ocp.oraclecloud.com/component-exchange-soctesting4-fra/s/component-exchange-soctesting4-fra_compcatalog_11494/compcatalog/0.2.0`
Username	compcatalog.user
Password	k9fz-0Pw4x-q

About Component Exchanges Hosted in Visual Builder Studio Projects

A Visual Builder Studio project can host a secure component exchange to store and distribute components only available to developers in the instance.

Each Visual Builder Studio project includes the component exchange 'compcatalog', which is the service used to access components stored in the project. The compcatalog service is provisioned by default with each project. Any project can be used to host an exchange if storage is enabled for the Visual Builder Studio instance. Component developers can use the service's APIs to publish components to the exchange.

To integrate a private exchange in a Visual Builder Studio project with a Visual Builder instance, an administrator specifies the URL for the project's compcatalog service and the credentials for a user that can access the project. The credentials used to connect to the exchange must be an owner or member of the Visual Builder Studio project hosting the exchange. All developers in the tenant use these credentials to connect to the exchange to get the components and application templates they want to use in their projects.

The URL for the project's compcatalog service has the following form: `https://<hostname>/<org_id>/s/<project_id>/compcatalog/0.2.0/`

In the URL, "compcatalog" is the exchange service and "0.2.0" is the API version of the service.

To determine the URL for the compcatalog service, you need to know the following details about the Visual Builder Studio project:

- *<hostname>*. This is the Visual Builder Studio server where the project is hosted.
- *<org_id>*. This is the organization (tenant) name.
- *<project_id>*. This is a project identifier unique to the tenant. This is not the same as the project display name entered by the project owner and is not displayed in the Visual Builder Studio UI.

If you do not know the <*project_id*> for the project hosting the exchange, you can get it from the Git or Maven configuration, or by using the Visual Builder Studio Projects API. The following table describes how to get the <*project_id*>.

Method	Steps
From a Git or Maven configuration	1. In Visual Builder Studio, open the project and locate the Repositories tab on the project's Home Page. 2. Expand the the Git or Maven section and copy the repository URL. The Git repository URL will be similar to the following: `https://{user_id}@{hostname}/{org_id}/s/my-org_testproject_5/scm/my-repo.git` The Maven repository URL will be similar to the following: `http://{hostname}/{org_id}/s/my-org_testproject_5/maven/` In these examples, "my-org_testproject_5" is the project identifier. In this case, the URL for the 'compcatalog' service will be similar to `https://{hostname}/my-org/s/my-org_testproject_5/compcatalog/0.2.0/`

Using Visual Builder Studio Projects API	If you know the name of the project sharing your exchange instance, you can get the project metadata using a REST call to the Visual Builder Studio API.
	For example, you can use cURL to send a REST call similar to the following:
	`curl -X GET -u '{username}:{password}'https://{hostname}/{org_id}/api/v2/projects/info/name:TestProject`
	The return should be similar to the following:
	`[ { "organization":"my-org", "identifier":"my-org_testproject_5", "name":"TestProject", "urlId":"testproject", "description":null, "accessibility":"PRIVATE", "template":false, "state":"READY", "locked":false, "relation": {"membership":"OWNER","favorite":false} } ]`
	In this example, the identifier property in the return is the project identifier that is needed for the "compcatalog" service URL.

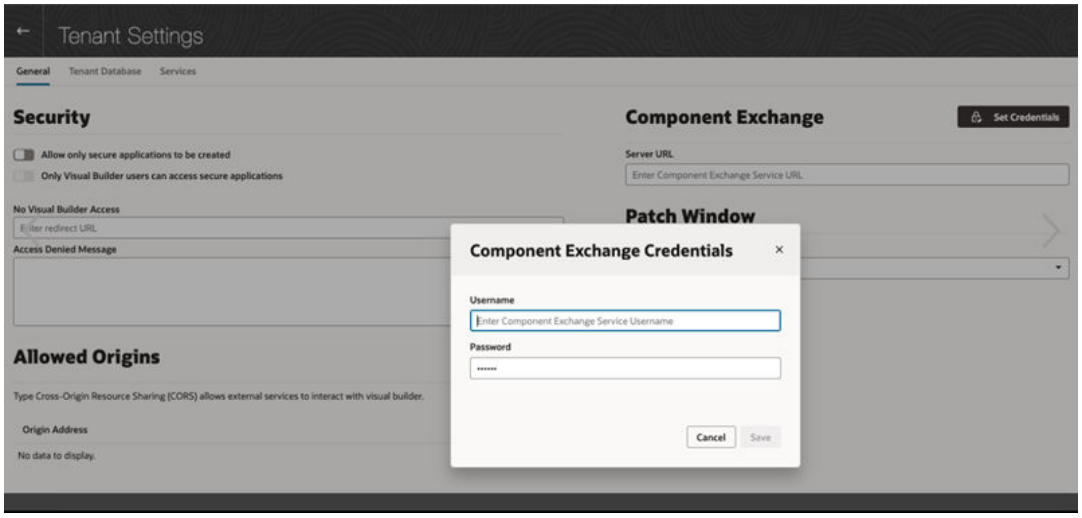
Add a Connection to a Component Exchange

When an instance is integrated with a component exchange, all developers using the instance can access and install components stored there.

After an exchange is added to the instance, all developers can use the Components tab in the Navigator to install and manage the components from the exchange that they want to use in their applications. When creating an application in the Create Application wizard, developers can also select any of the application templates that have been published to the exchange.

To add a connection to the Component Exchange:

1. Open the instance's Tenant Settings page.
2. In the Component Exchange panel, enter the URL for the component exchange, then click **Set Credentials**.
3. In the Component Exchange Credentials dialog box, enter the username and password for the user whose credentials will be used to authenticate access to the Component Exchange. Click **Save**.



If you are adding a connection to a private component exchange, it is recommended that the credentials you provide are for an administrator who is a member of the VB Studio project hosting the exchange or the project owner.

Configure Support for a Custom Domain

If you want your customers to see a different URL than the one generated by Visual Builder, you can map a custom domain, also called a vanity URL, to an app in your instance. A custom domain is a customer-provided hostname and domain (FQDN) created by adding a subdomain to your domain. After configuring an app to use a custom domain, app users accessing the app using the custom domain will not see the typical Oracle domain (for example, `myvbinstance-accountname.builder.ocp.oraclecloud.com`) in the URL, but instead would see something like `mycustom.example.org`.

To use a custom domain:

- Configure the custom URL for your Oracle Integration 3 instance, using the instructions in Create and Configure a Custom Endpoint.

> **Note**
>
> Custom URLs are supported on Oracle Integration instances, as well as on Visual Builder instances.
>
> Depending on whether you're on Oracle Integration, Visual Builder, or a Visual Builder instance that was provisioned as part of a SaaS order, the process for enabling custom URLs varies. If you are using a Visual Builder instance, use the instructions in Configure Support for a Custom Domain in *Administering Oracle Visual Builder*.

- Set the custom domain in the visual application's Settings editor and publish the app. See Specify a Custom App URL in *Developing Applications with Oracle Visual Builder in Oracle Integration 3*.

> **Note**
>
> You can configure only one custom domain for each Oracle Integration 3 instance.

After configuring a custom domain:

- Users can access a single web app by typing just the custom domain URL in the browser, for example, `mycustom.example.org`. The app is loaded from the custom domain root ("/"), and no additional path information or query parameters are required in the URL.
- `http` can be redirected to `https`, so if a user types "mycustom.example.com", this will resolve to `https://mycustom.example.com`, and load the default web app.
- For applications that contain business objects, the Business Object REST API can also use the custom domain configuration.
- Developers can access the Designer in Visual Builder using a custom domain.
- If you create and stage an application from a custom domain (`https://mycustom.example.com/ic/builder/designer`), you'll be automatically redirected to the custom domain (`https://mycustom.example.com/ic/builder/rt/appid/version/...`) when you open the app using a URL that isn't the application's custom domain (for example, your instance's URL `https://servicename.oraclecloud.com/ic/builder/rt/appid/version/...`).

Custom domains are also subject to other limitations:

- If the custom endpoint is selected as the Vanity URL in the application's Settings editor, after the app is published it can only be accessed from the custom domain root (for example, `https://mycustom.example.com`).
- If you publish a different web app in your visual application, it immediately becomes the default app for the custom domain, and the previous web app will no longer be available at the custom domain.
- A custom domain can only be used to access one live app (in the visual application configured for the root URL). You can access other live apps in the same instance only by using the full Oracle Cloud URL or by creating and configuring a different custom domain and visual application.
- If a visual application contains more than one web app, only one of them can be accessed using the custom domain. It's not possible to specify which app in a visual application will

be available at the custom domain because the domain is configured in the Settings for the visual application, not for individual web apps. If you are going to use a custom domain, it is recommended that the visual application only contain one web app to ensure that the correct app is loaded.

5

Reference

Reference topics for Oracle Visual Builder.

Topics:

Configure a Custom URL Using Oracle Web Application Firewall Service V2

You can use Oracle's Load Balancer and Web Application Firewall Service V2 (WAF V2) to help you configure support for a custom URL for a Visual Builder instance.

When you configure a custom URL for a Visual Builder instance, (for example, `https://<my-custom-url.com>/ic/builder/`), you can access your instance directly using the custom URL. When you publish an application from the custom URL, the application will use the custom URL (for example, `https://<my-custom-url.com>/ic/builder/rt/`).

You can also configure a Visual Builder app to use a custom URL, also called a vanity URL, so that customers can access the app using just the base custom URL (`https://<my-custom-url.com>`).

The WAF V2 service allows you to define a policy that will map a custom domain name to the WAF service as the front end for your VB service as the origin server. To do this you'll use a public load balancer for managing the certificates in your tenancy. You can set up the load balancer in Oracle's Load Balancer service.

By using WAF to map your chosen DNS name to a VB service, you can manage the mapping of your DNS name and uploading your associated certificate and private key yourself instead of configuring the VB instance to manage them.

These instructions assume you have direct access to a Visual Builder instance and to the Oracle Cloud Infrastructure (OCI) Console. For more details on using your instance behind a WAF or an API Gateway, see:

- Certificates for Web Application Firewall
- Setting Up Custom Domains and TLS Certificates for API Gateways

Before You Configure the Custom URL

Before you start configuring the custom URL, you'll need to know some details about your instance, and you should also be aware of the limitations of using a custom URL for a Visual Builder instance.

What you'll need to configure the custom URL:

- **Visual Builder instance**: You'll need to have already provisioned a Visual Builder (or Integration) instance on Oracle Cloud. The instance must be a PSM/OCI based Oracle-managed instance.
- **VB instance public loadbalancer IP**: You can obtain the load balancer IP address by performing a dig on the hostname using the URL. For example, for the URL: `https://vbmyinst-vb-axdkj3wttbhm.builder.us-ashburn-1.ocp.oraclecloud.com/ic/builder/`

  Run the following command from the terminal to obtain the IP address required to configure backends:

  ```
  dig https://vbmyinst-vb-axdkj3wttbhm.builder.us-ashburn-1.ocp.oraclecloud.com
  ```
- **DNS name**: You must decide what DNS name will be used to access the system, and that name must be in a DNS domain that you own.
- **SSL certificate**: You must have a CA signed SSL certificate with a private key for the DNS name.

Known Limitations

Custom URLs are subject to the following limitations:

- Only one web application at a time can be accessed using the root context (‘/’) of the custom URL.

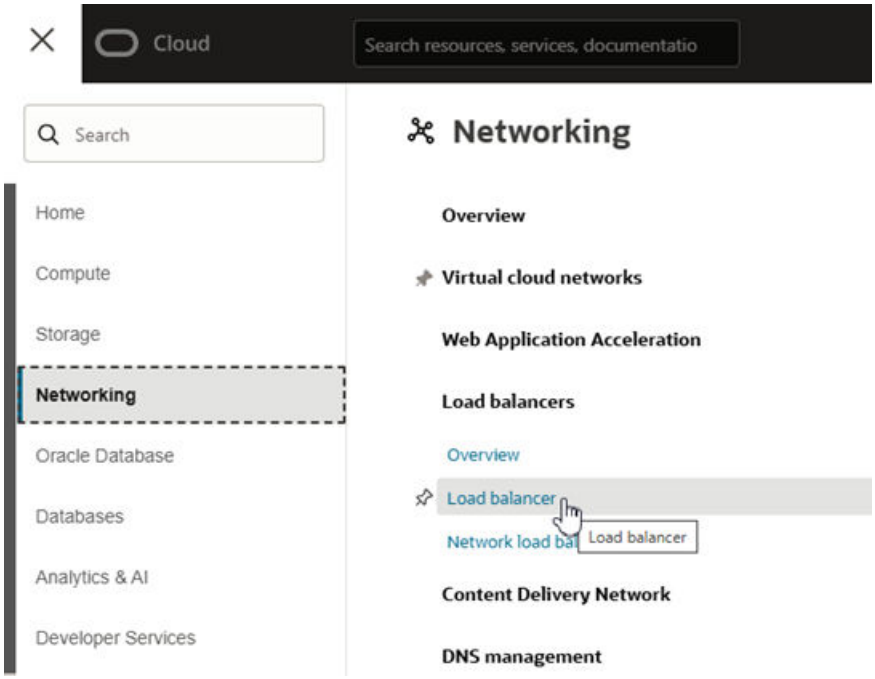
Create a Load Balancer and Configure a Hostname

You can use the Oracle Load Balancer service to create a public load balancer for managing the certificates in your tenancy.

A load balancer provides automated traffic distribution from one entry point to servers reachable from your virtual cloud network. For more about Oracle Load Balancer, see Overview of Oracle Load Balancer and Creating a Load Balancer.

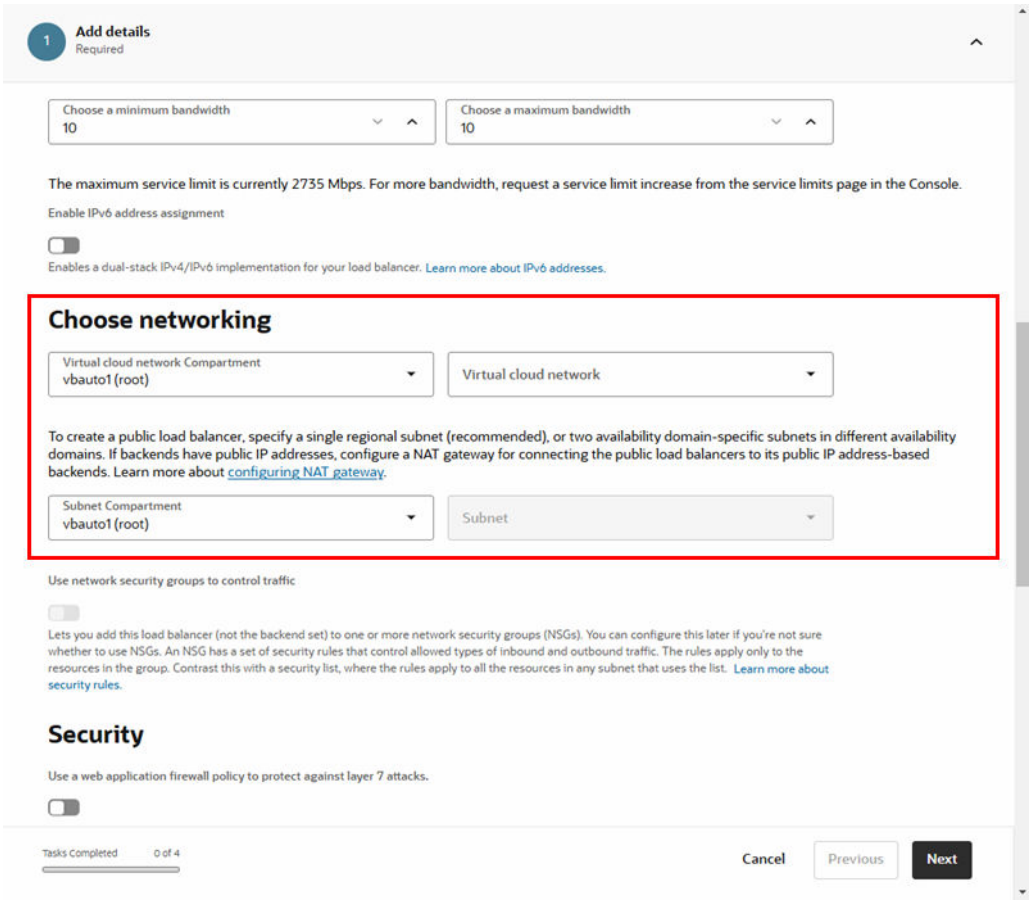
To create a load balancer:

1. In the OCI Console, click **Navigation Menu** ☰, select **Networking**, and then select **Load Balancer**.



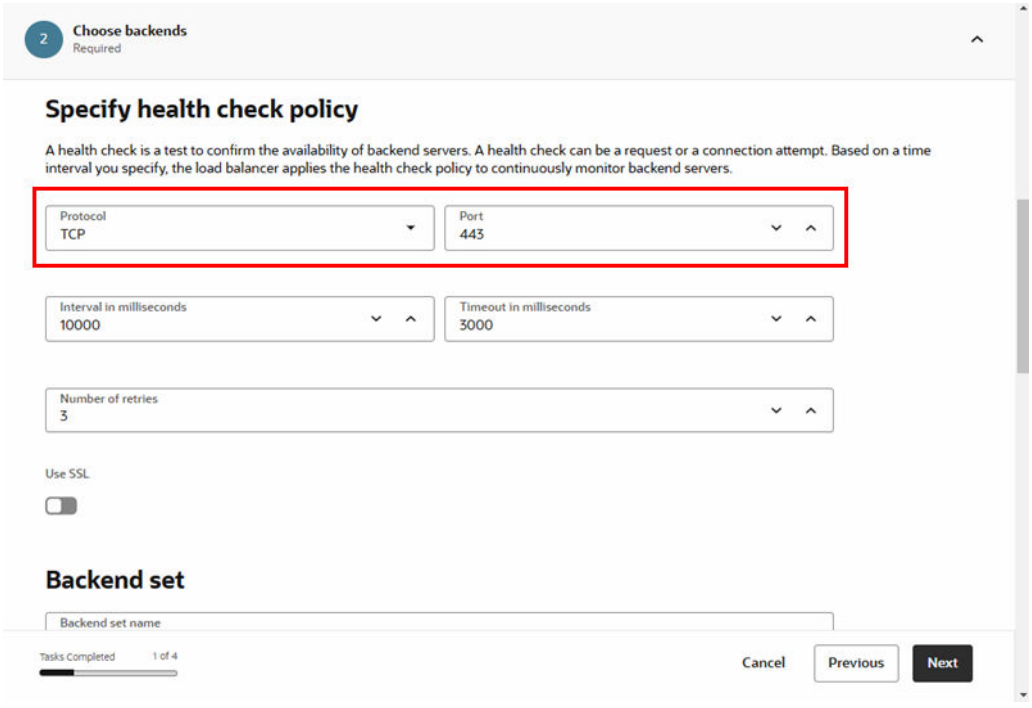
2. Create a new load balancer:
   a. On the Load Balancers page, click **Create load balancer**.
   b. Select `Load Balancer` as the type, then click **Create Load Balancer** to open the Create Load Balancer page to define the load balancer's details.
   c. On the Add Details page, select the defaults for the shapes and networking options.

      In the Choose networking section, you need to select a Virtual cloud network and Subnet, if they are not already selected.



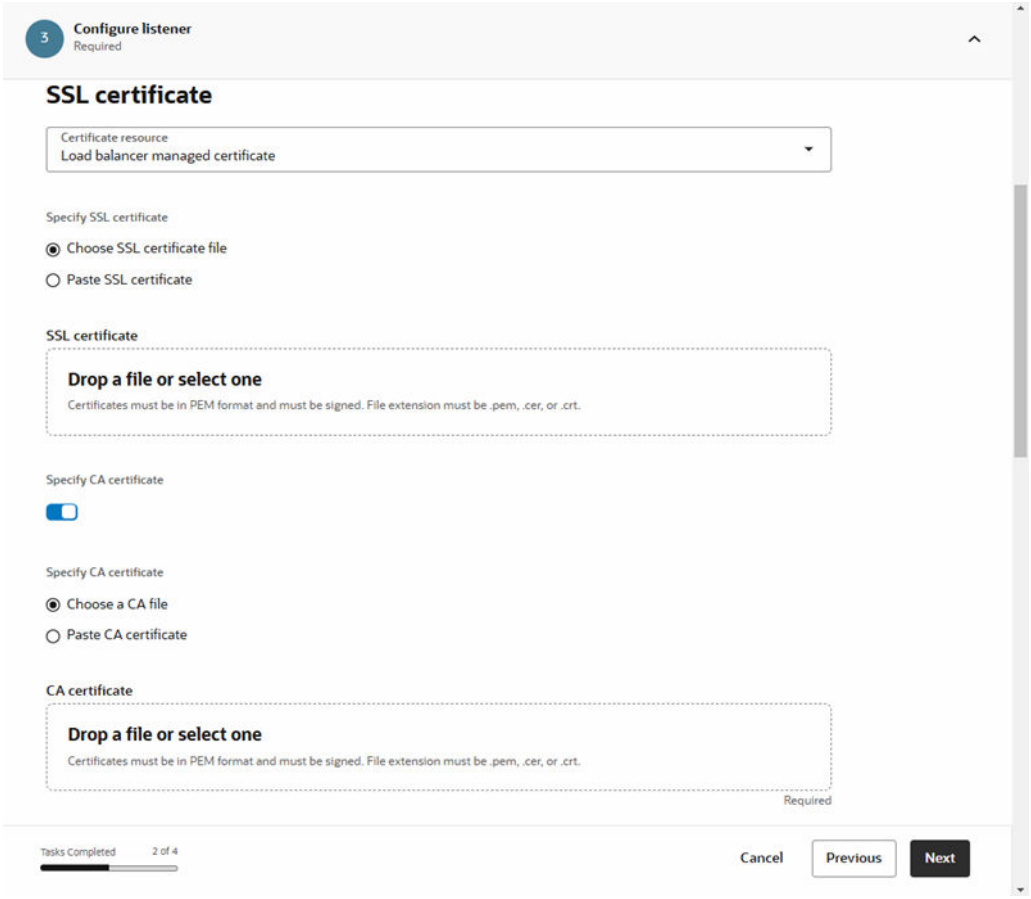
Click **Next**.

d. In the Specify Health Check Policy pane on the Choose Backends page, select **TCP** as the Protocol and set the port to **443**. Click **Next**.



e. In the SSL Certificate pane on the Configure Listener page, select **Load Balancer Managed Certificate** in the Certificate Resource dropdown list.

f. Provide your certificate chain and private key. Click **Next**.

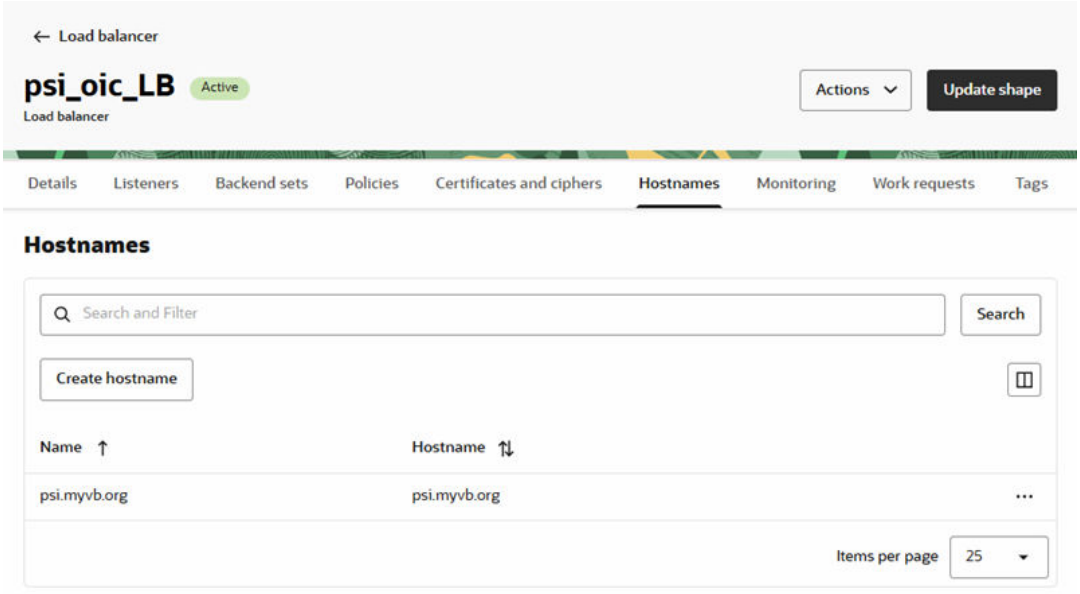


g. On the Manage Logging page, accept the default settings. Click **Submit** to create the load balancer.

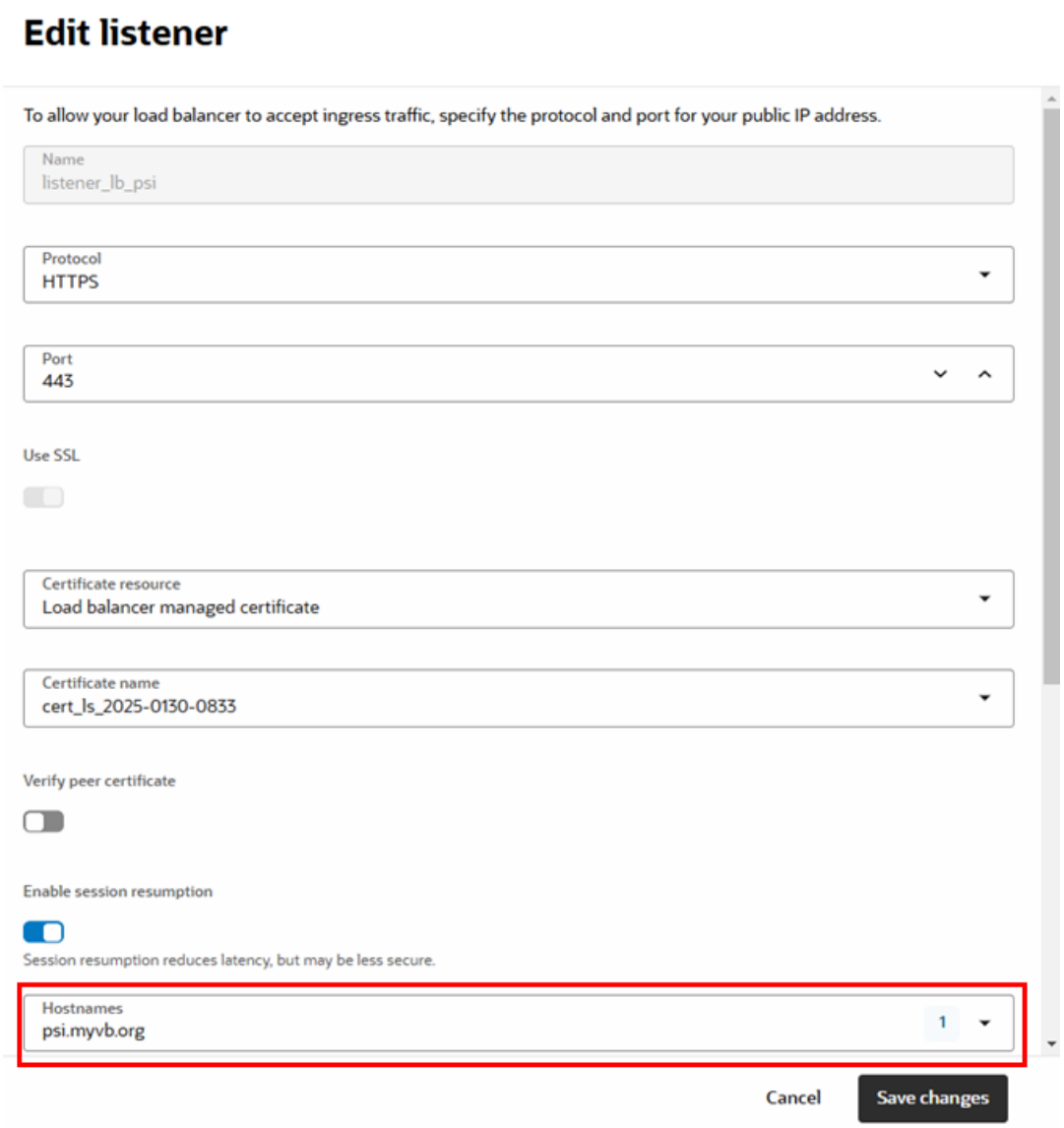
> **Note**
>
> It will take a few minutes to provision the load balancer

3. After the load balancer is provisioned, click the name of the new load balancer on the Load Balancers page to open its Details tab.
4. Open the **Hostnames** tab, and then click **Create hostname**.
5. Enter a Name and Hostname in the Create hostname page. Click **Create**.

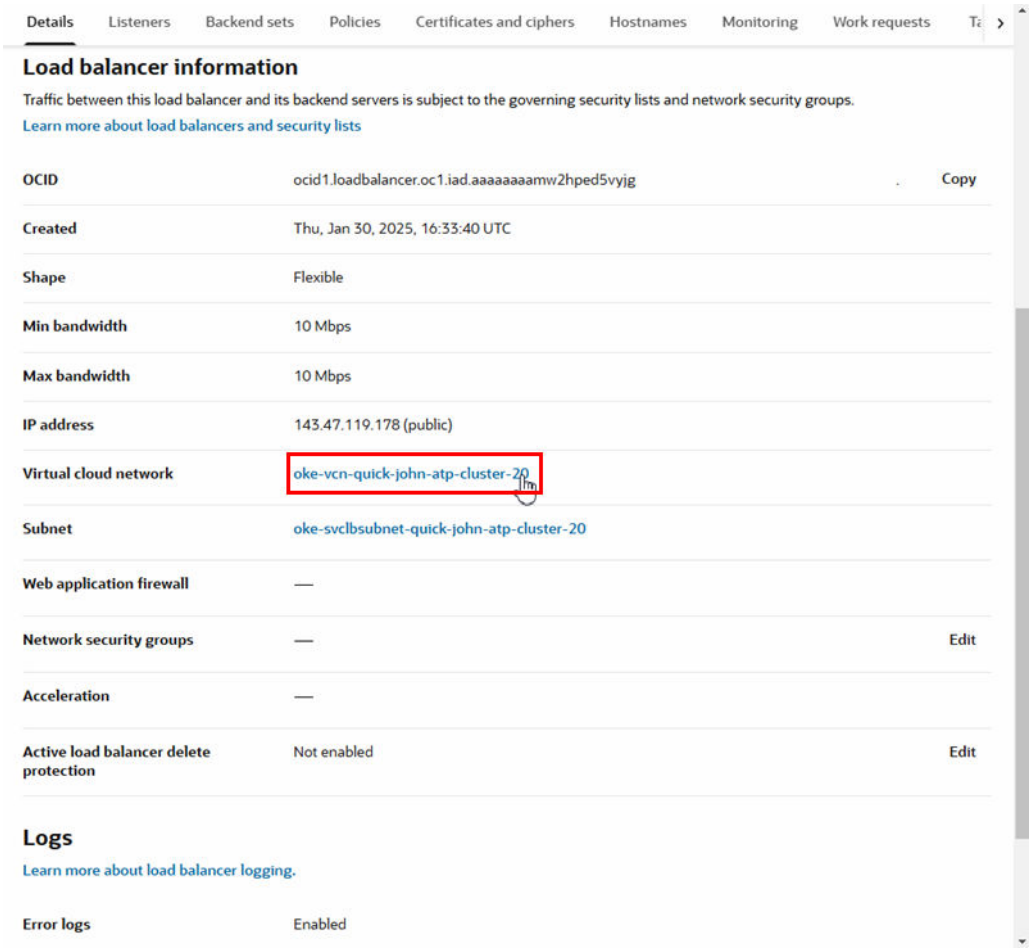
   The hostname will be your custom endpoint.



6. Open the **Listeners** tab, and edit your listener to add the hostname. Click **Save changes**.



7. Configure the load balancer Virtual Cloud Network (VCN):
   a. Open the load balancer's Details tab, and then click the Virtual Cloud Network (VCN) link to open the VCN's Details tab:

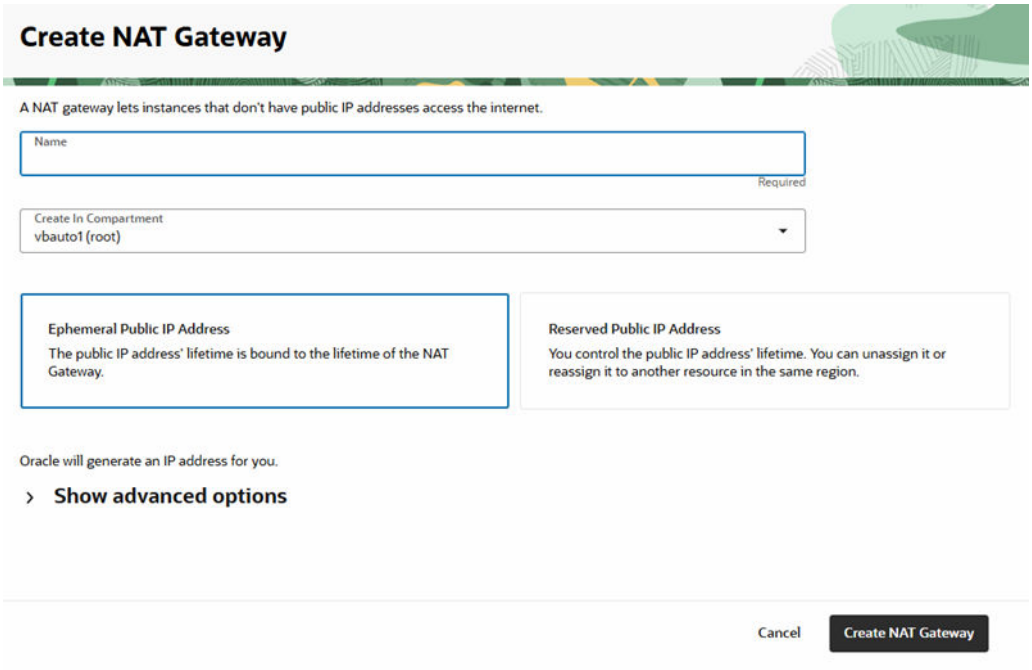


b. Open the VCN's **Gateways** tab, and then click **Create Internet Gateway**.

c. In the Create Internet Gateway page, enter a name, and then click **Create Internet Gateway** to return to the Gateways tab.

d. In the Gateways tab, click **Create NAT Gateway**.

e. In the Create NAT Gateway page, enter a name for the gateway and select **Ephemeral Public IP Address**. Click **Create NAT Gateway**.



f. Open the **Routing** tab, and then click **Create Route Table**.

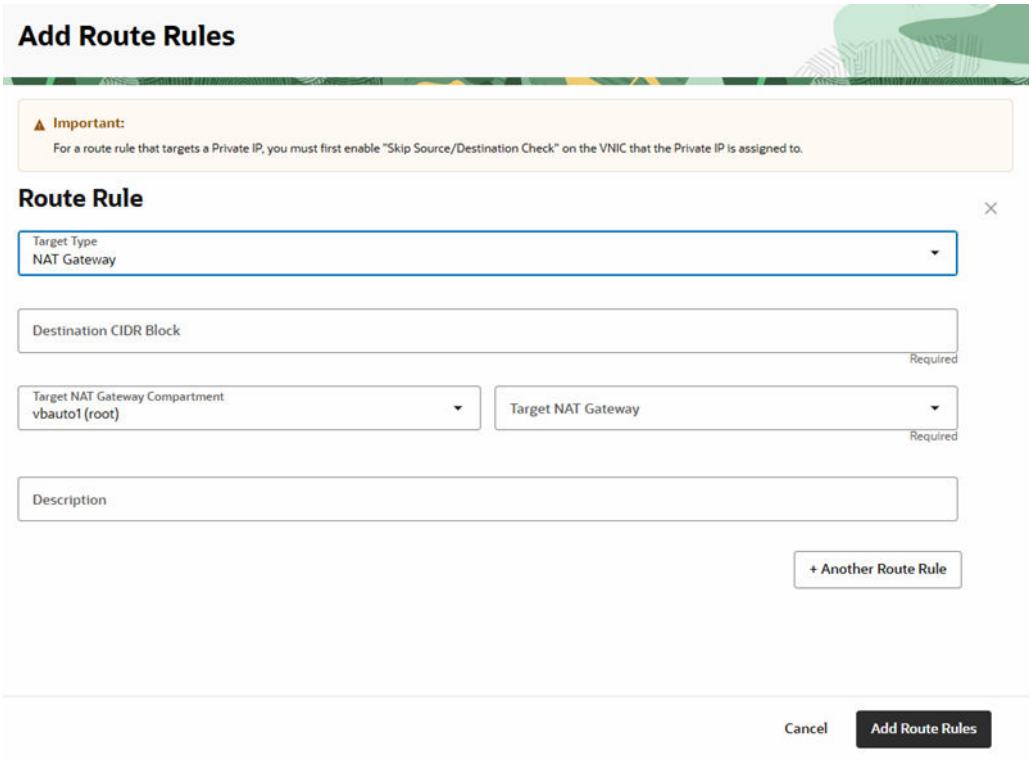
g. In the Create Route Table page, enter a name, and then click **Create Route Table** to return to the Routing tab.

h. Click the new route table to open its details page.

i. Open the **Route Rules** tab, and then click **Add Route Rules**.

j. In the Add Route Rules page, enter these details for the NAT gateway route rule:

- **Target Type:** NAT Gateway.
- **Destination CIDR Block:** Provide the Visual Builder instance public load balancer IP (see Setup above on how to obtain it). If it is a single IP, append /32 to it to form a single IP CIDR Block.
- **Compartment:** Leave as is.
- **Target:** Select the NAT gateway you created.
- **Description:** An optional description of the rule.



You need to create a NAT gateway route rule for each of your Visual Builder instance public load balancer IPs. To add a route rule, click **+ Another Route Rule**.

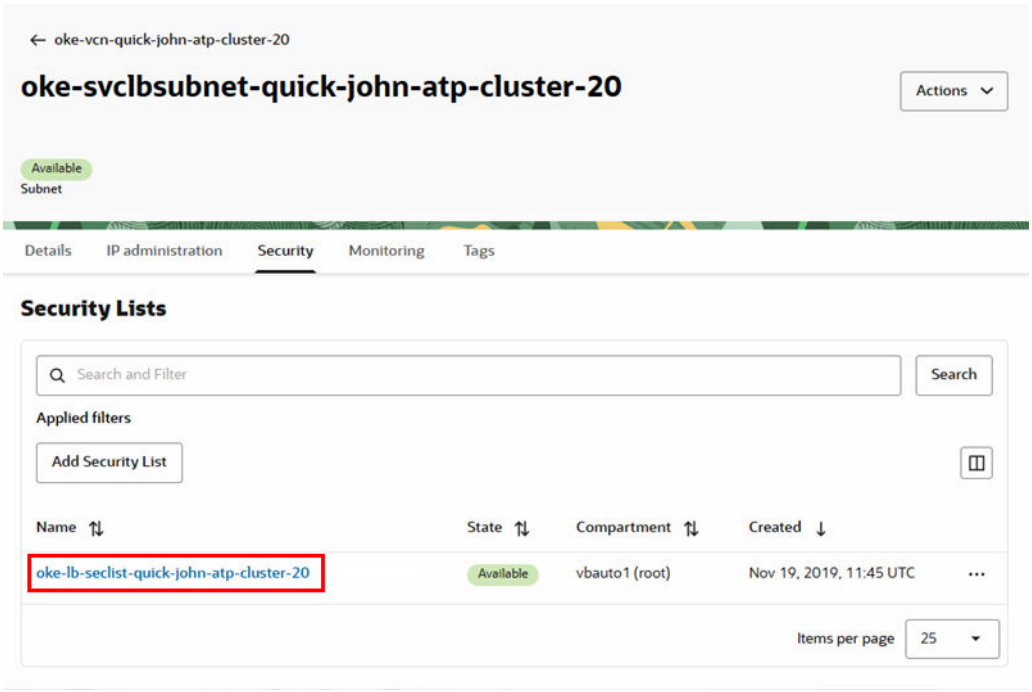
k. Click **+ Another Route Rule**, and enter these details for the internet gateway route rule:

- **Target Type:** Internet Gateway.
- **Destination CIDR Block:** 0.0.0.0/0
- **Compartment:** Leave as is.
- **Target Internet Gateway:** Select the internet gateway you created.
- **Description:** An optional description of the rule.

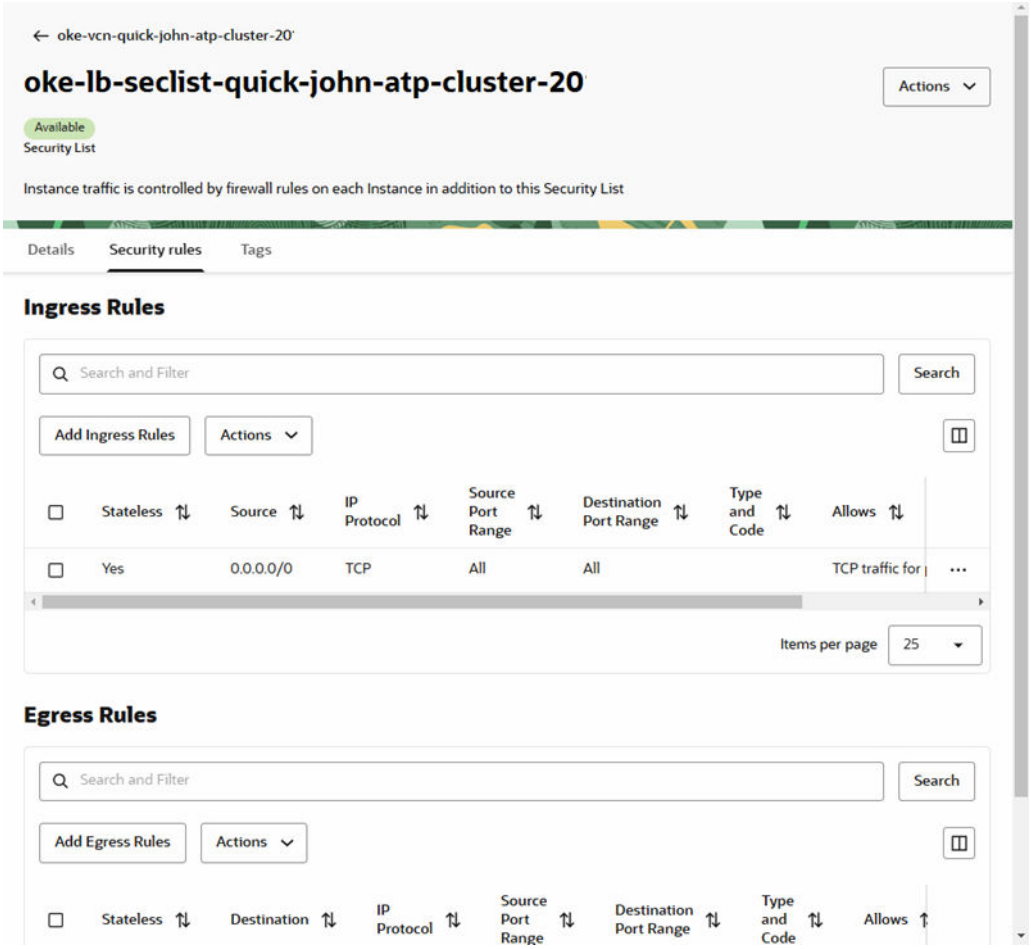
Click **Add Route Rules**.

l. Confirm that the health check status for your Backend Set is OK.

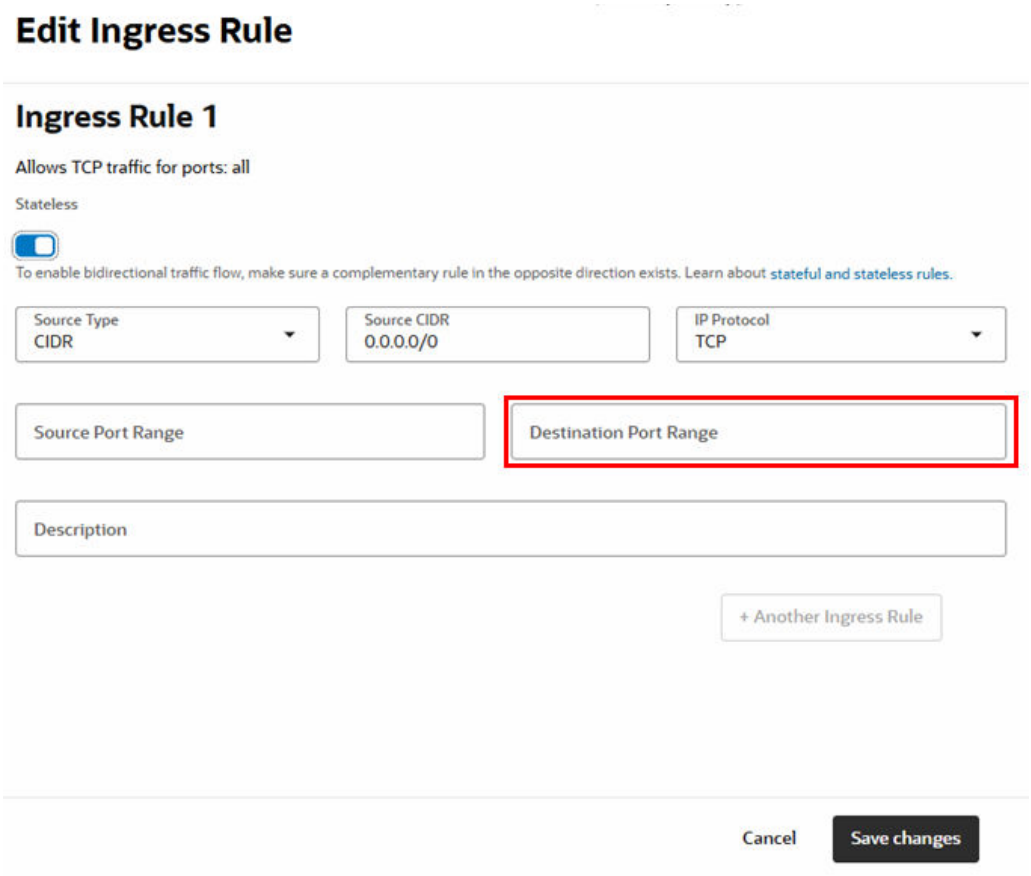
8. Return to the load balancer's Details tab.
9. Configure the load balancer subnet:
   a. On the load balancer's Details tab, click the Subnet link to open its details page.
   b. Open the subnet's **Security** tab, and then click the default security list in the table to open its Details pane.



c. Open the **Security rules** tab.



d. Edit the rule for entry `0.0.0.0/0` in the Ingress Rules table to change the Destination Port Range to **443**. Click **Save changes**.

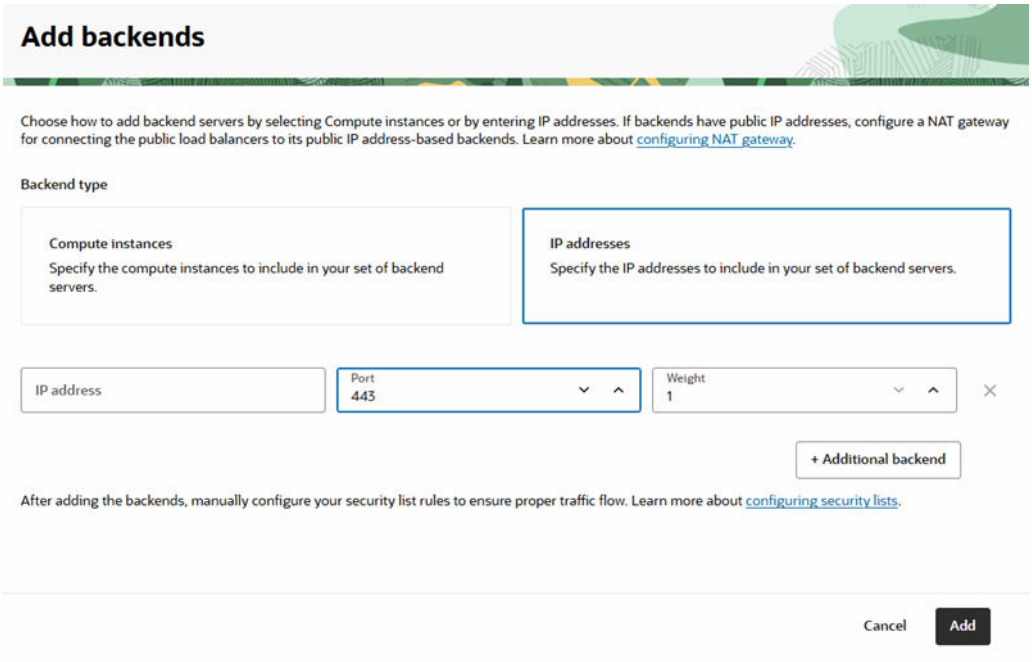


10. Set the SSL option for the backend:
    a. On the Backends page, select the **SSL** option.
    b. Select the **Load Balancer Managed Certificate** option.
    c. Select **Load Balancer managed certificate** and select the certificate from the dropdown list.

> **Note**
>
> If you get an error that a CA certificate is missing, create a new Load Balancer Managed Certificate and provide the server cert and intermediate cert separately instead of a combined chain.

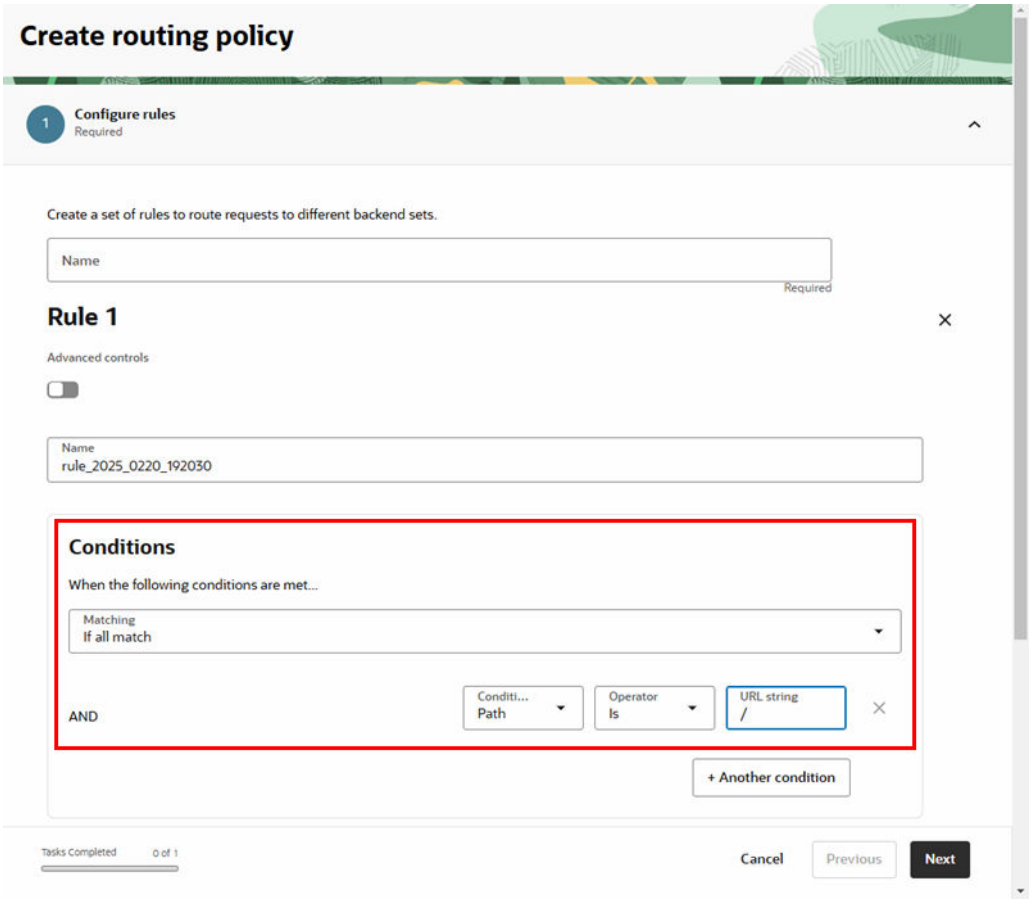
11. Add a new backend:
    a. Open the **Backend Sets** tab, and then click the backend set link in the table to open its Details tab.
    b. Open the **Backends** tab, and then click **Add Backend**.
    c. Select the **IP Addresses** option, and set the following backend details:
        - IP Address: Provide the IP address for the load balancer. This is the IP address you obtained when you used the `dig` command on the Visual Builder hostname.

- Port: Set the port to 443.



Click **Add**.

12. (Optional) If you want to restrict access:
    a. Open the **Policies** tab, and then click **Create routing policy**.
    b. In the Create routing policy page, enter a name for the routing policy.
    c. In the Conditions pane, configure the policy by setting the following:
        - When the following conditions are met: Set to `If All Match`
        - Condition Type: Set to `Path`
        - Operator: Set to `Is`
        - URL String: Set to `/` .



d. In the Action pane, define the "Route to backend" action by selecting the backend set from the dropdown list. Click **Next**.

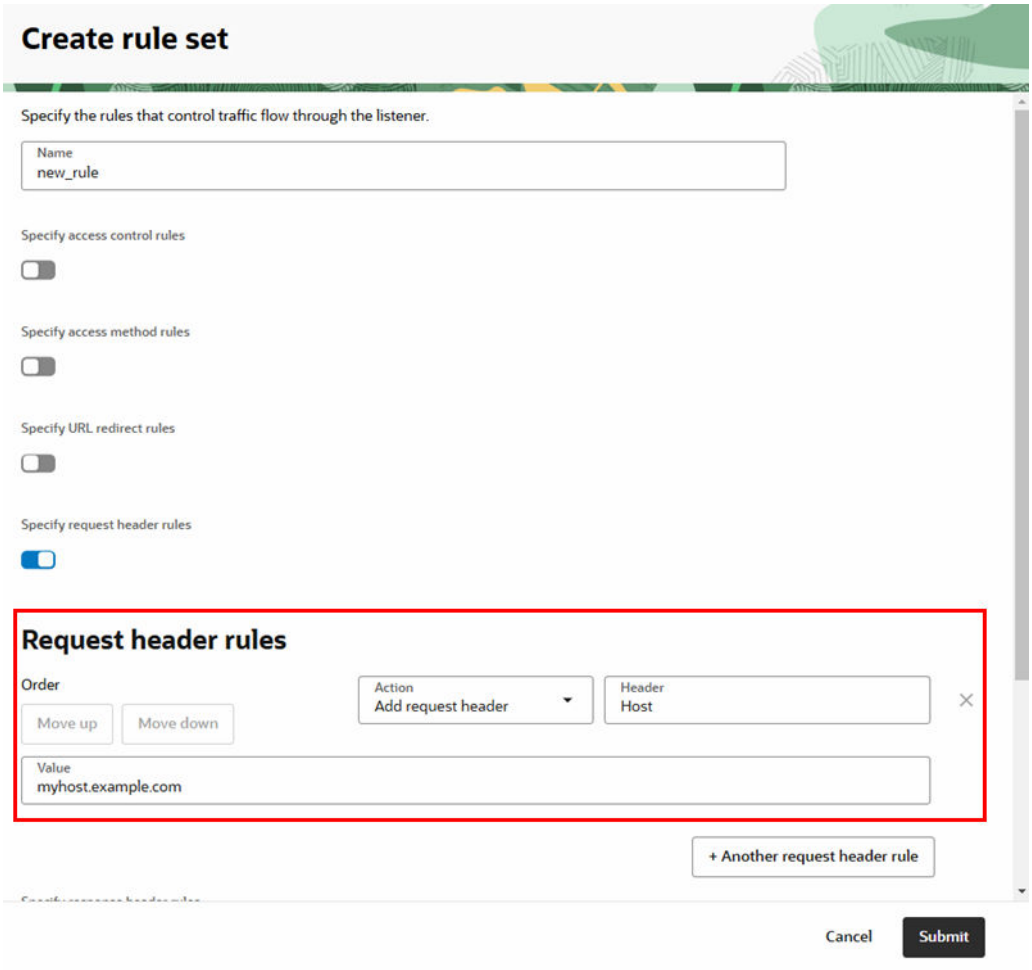
e. Set the order that policies should be performed, if needed. Click **Create routing policy** to return to the Policies tab.

f. In the Policies tab, click **Create rule set**.

g. In the Create rule set page, enter a name for the rule set.

h. Select **Specify request header rules**, and then enter the details:

- Action: `Add Request Header`
- Header: `Host`
- Value: Add your custom URL (for example: `myhost.example.com`)



Click **Submit** to return to the Policies tab.

Create a WAF Policy

You use a WAF policy to configure the access rules, rate limiting rules, and protection rules for your Web Application Firewall service.

When creating and configuring a WAF policy for your custom URL, you'll need to specify the load balancer used for your Visual Builder instance.

To create a WAF policy and specify the load balancer:

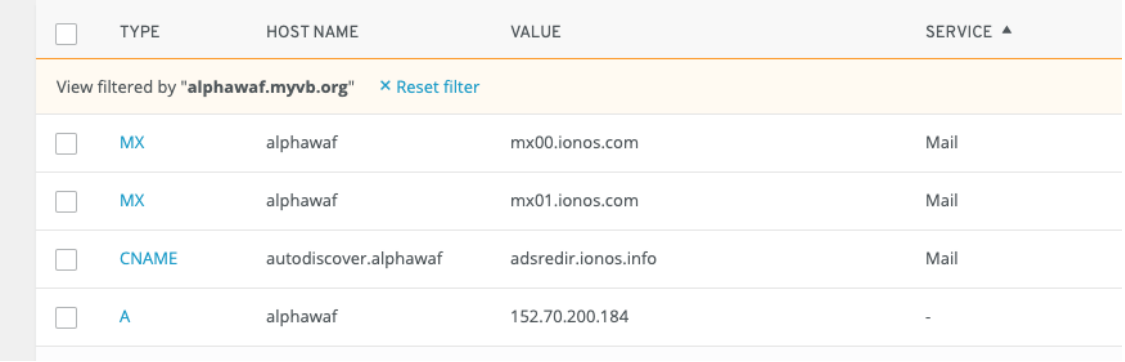
1. Sign in to the Oracle Cloud Infrastructure Console and open **WAF Policies** under **Security**.
2. Select the compartment you want the WAF policy to be created in and click **Create WAF Policy**.
3. Enter the policy name in the Create WAF Policy dialog box.
4. Accept all other defaults, and then click **Next** until you reach the **Select Enforcement Point** step.
5. In the **Select Enforcement Point** step, select the load balancer you created and complete the WAF configuration.
6. Click **Create WAF Policy**.

Now that the policy is created and you've configured it to use your load balancer, you can configure the policy rules. You can edit the policy configuration at any time. When configuring the policy, you can use the pre-defined actions, or create your own customized actions. For more about WAF policies, see Getting Started with Web Application Firewall Policies.

Configure the DNS

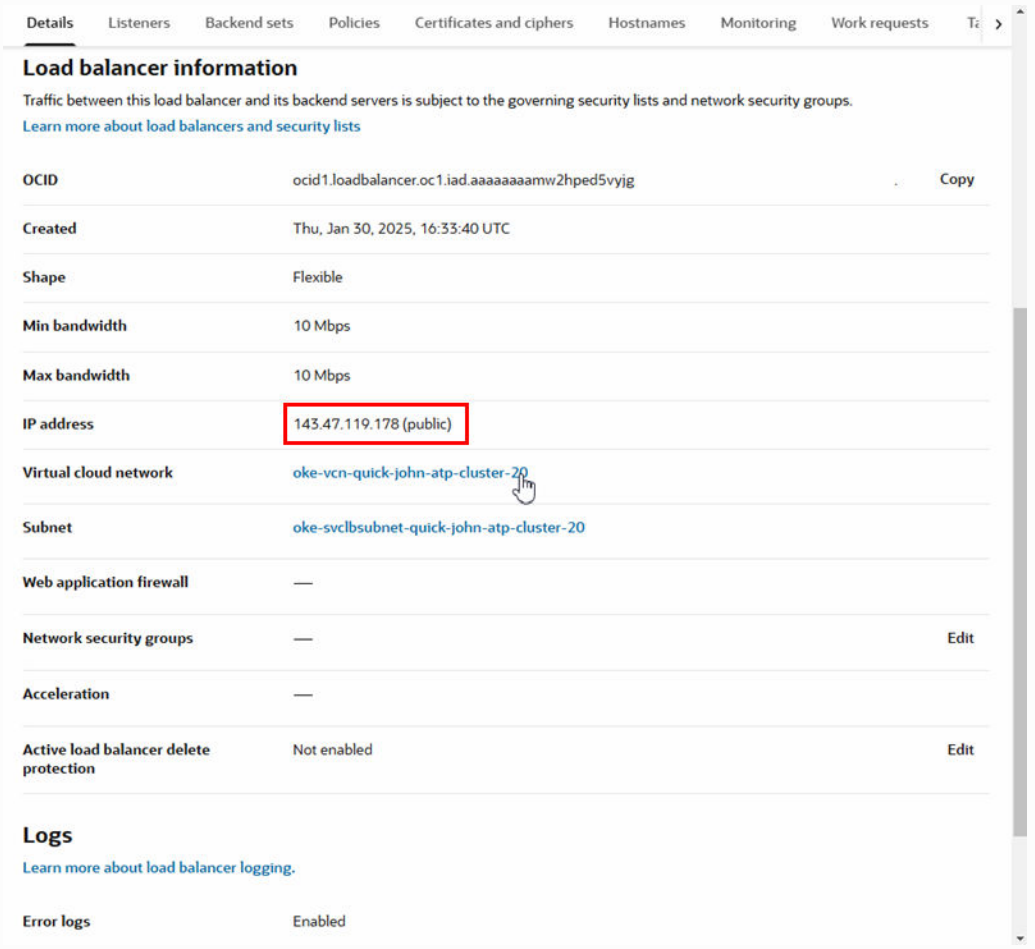
Register or update the custom DNS name with the load balancer public IP address.

In the DNS configuration for the name you've chosen to access the VB service instance, edit the A record to point to the public IP address of the load balancer. In the following image, the value of the A record is set to the public load balancer IP address `152.70.200.184`:



TYPE	HOST NAME	VALUE	SERVICE
View filtered by "**alphawaf.myvb.org**" × Reset filter			
MX	alphawaf	mx00.ionos.com	Mail
MX	alphawaf	mx01.ionos.com	Mail
CNAME	autodiscover.alphawaf	adsredir.ionos.info	Mail
A	alphawaf	152.70.200.184	-

> **Note**
>
> You can find your load balancer's public IP address in the Load Balancer Information tab on the Load Balancer page:



Configure a Vault for a Custom Endpoint

To create a custom endpoint for your Visual Builder instance, you can use the Key Management Service in OCI to create a vault to store the master encryption keys and secrets used to protect access to your custom endpoint.

In the OCI Console, you create an OCI vault in the compartment where you want to create your custom endpoint. For more details on working with vaults, see Working with Compartments, Overview of Vault, and Create a New Vault.

> **Note**
>
> If you are using a WAF and load balancer to protect your custom endpoint, you don't need to create a vault.

After you create and configure a vault in the OCI Console, you can configure your instance's first (primary) custom endpoint in the Visual Builder Instance details page. If your instance already has a primary endpoint and you want to add another, you need to create an alternate endpoint from the command line. Similarly, if your instance already has multiple custom endpoints and you want to edit any of them, you also need to do that from the command line. For details, see Create and Update Alternate Endpoints.

When creating the secret in your vault, you'll need to provide a secret certificate that contains:

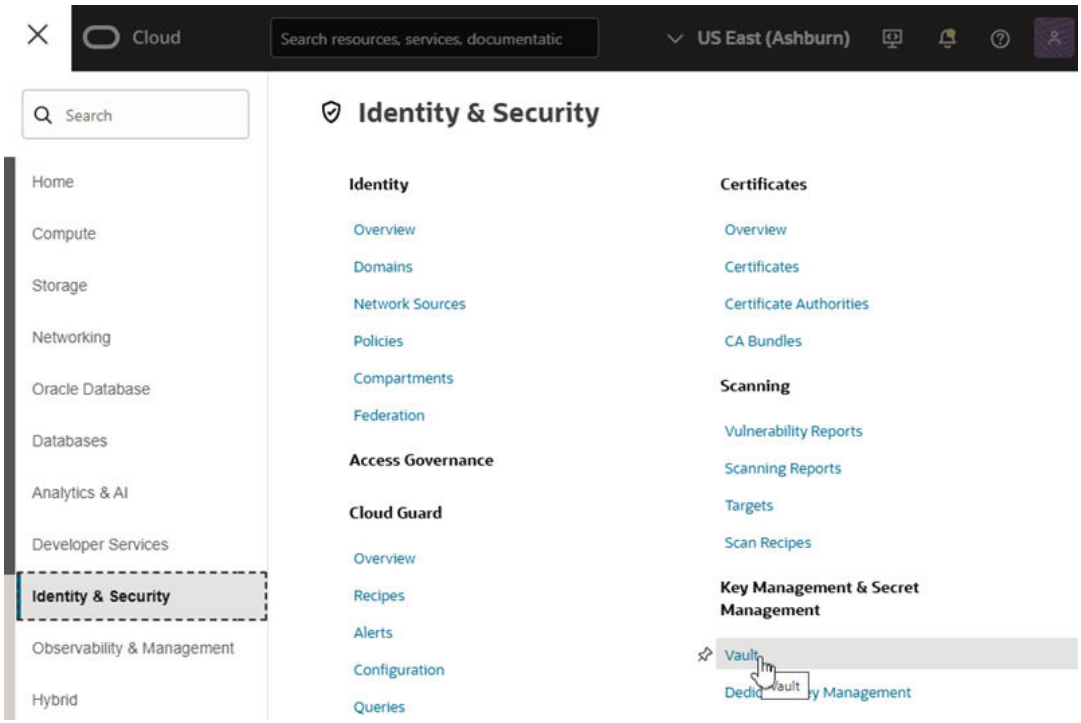
- the hostname's SSL certificate,
- the matching private key, and
- all intermediate certificates in the SSL chain.

You'll also need to provide a passphrase if the SSL certificate requires one. You can obtain these from your SSL certificate provider.

> **Note**
>
> You can use openssl to validate the SSL certificate and private key.

To create and configure an OCI vault in the OCI Console:

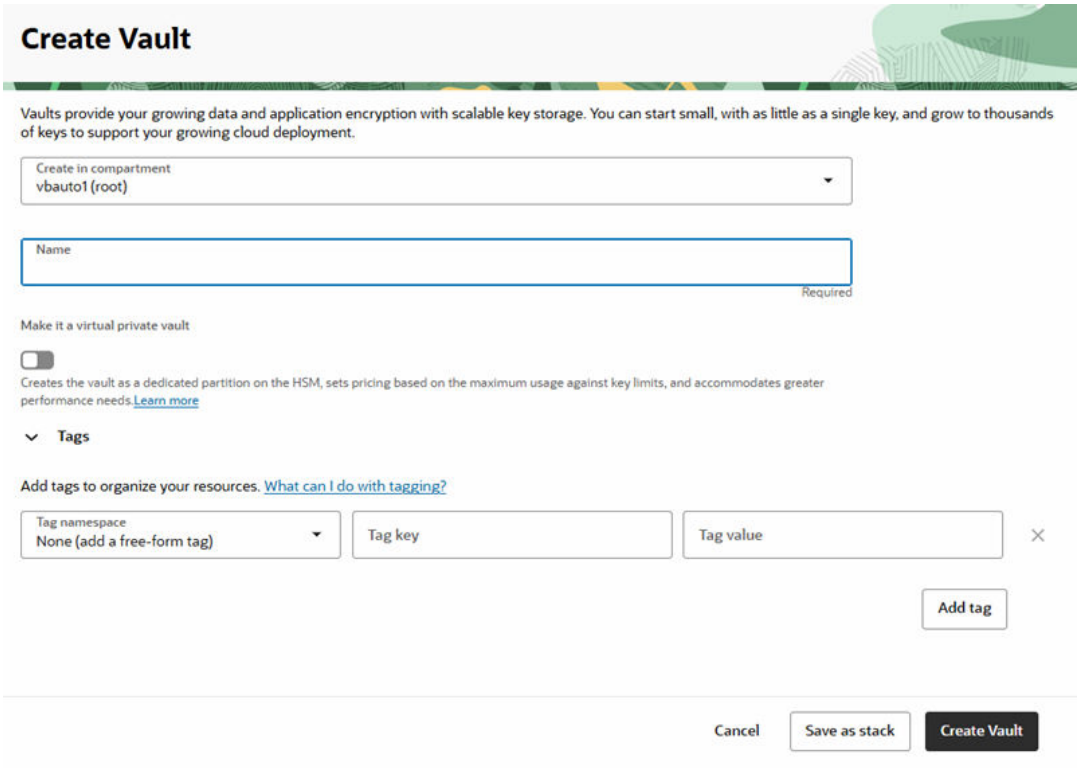
1. In the OCI Console, click **Navigation Menu** ☰, select **Identity & Security**, and then select **Vault** to open the Vaults page.



2. On the Vaults page, click **Create Vault** to open the Create Vault panel.

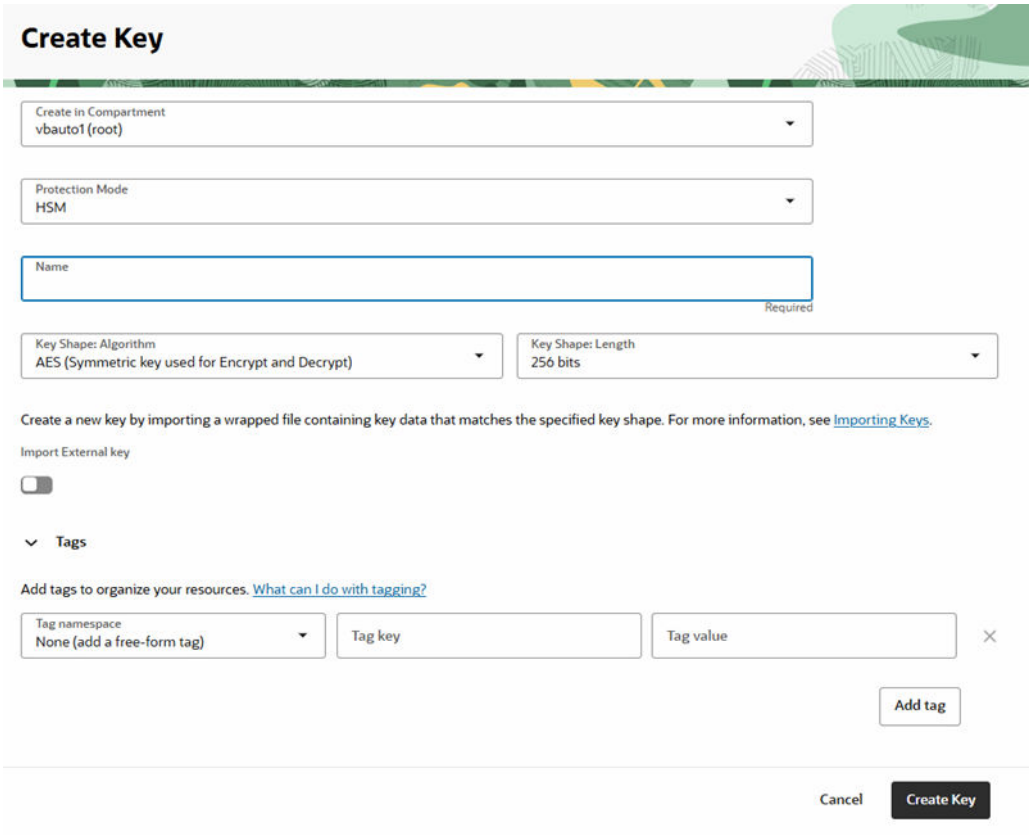
   In the Create Vault panel, confirm you are creating the vault in the correct compartment. If you are not in the correct compartment, select the compartment in the Create in Compartment dropdown list.

3. Enter a name for the vault. Click **Create Vault** to return to the Vaults page.



After you click Create Vault, it can take a few minutes for the new vault to appear in the table on the Vaults page.

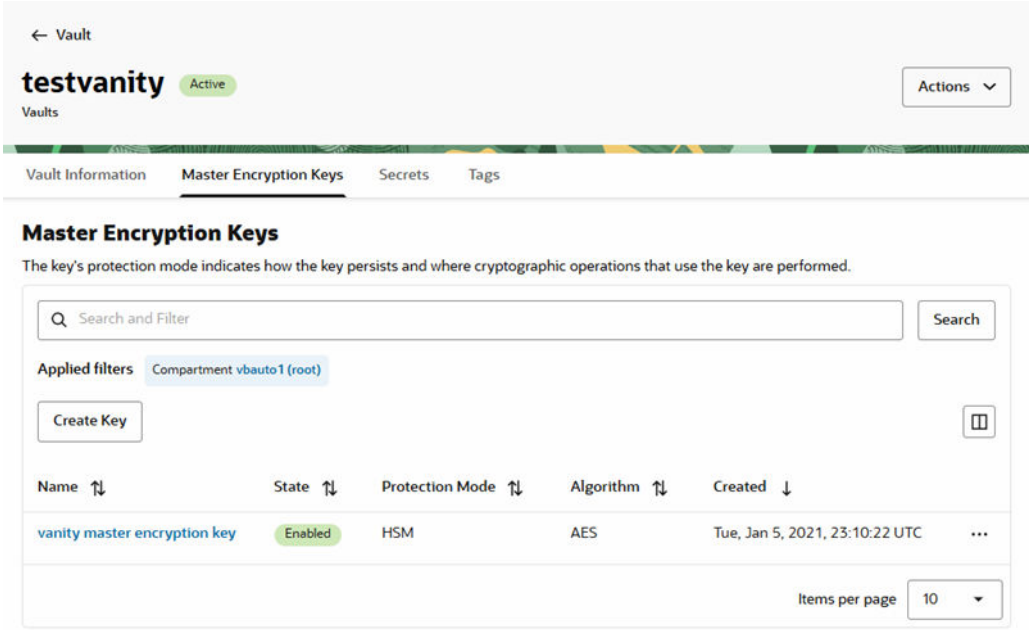
4. In the table on the Vaults page, click the name of the vault you created to open the vault's details page.
5. Create a master encryption key for the vault.
   a. Open the vault's **Master Encryption Keys** tab.
   b. Click **Create Key** to open the Create Key panel.



c. Enter a name for the key in the Name field.

   To create the key, you only need to enter a name. Use the default settings for the other options.

d. Click **Create Key** to return to the Master Encryption Keys tab.



6. Create the secret.

   Store the certificate as a secret in the OCI Vault. For more about secrets, see Create a New Secret.

   a. Open the vault's **Secrets** tab.

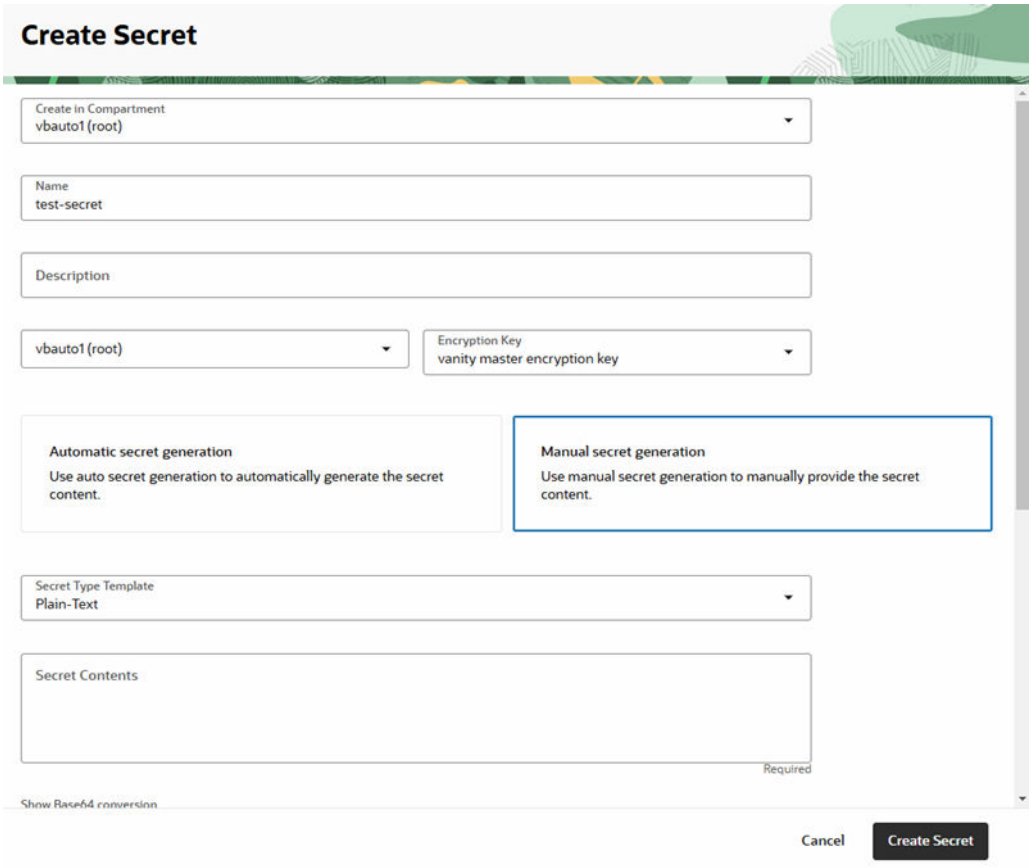
   b. Click **Create Secret** to open the Create Secret panel.

   c. Enter a name and description for the secret.

   d. In the **Encryption Key** dropdown list, select the key you created in the Master Encryption Keys tab.

   e. Select **Manual secret generation**.

      Make sure you explicitly select Manual secret generation. The default is Automatic secret generation.



   f. Generate the secret certificate and paste it into the Secret Contents field.

      Use the following format for the certificate:

```
{
  "key": "-----BEGIN PRIVATE KEY-----\n….-----END PRIVATE KEY-----\n",
  "cert": "-----BEGIN CERTIFICATE-----\n….-----END CERTIFICATE-----\n",
  "intermediates": [
    "-----BEGIN CERTIFICATE-----\n….-----END CERTIFICATE-----\n",
    "-----BEGIN CERTIFICATE-----\n….-----END CERTIFICATE-----\n"
  ],
  "passphrase": "<private key password if encrypted key is provided>"
}
```

When generating the certificate, note the following certificate requirements:

- The `key` and `cert` elements are required.
- Each intermediate certificate must be specified as a separate element in an `intermediates` array. In most cases there will only be one intermediate. The intermediate is provided by the SSL provider.
- Always ensure that the final root CA is specified as the last element in the array. For example, if there are three intermediate certificates for the leaf certificate, the certificate that issued the leaf certificate should go as the `intermediates[0]` element, the certificate that issued the `intermediates[0]` certificate should go in the `intermediates[1]` element, and the certificate that issued the `intermediates[1]` certificate should go in the `intermediates[2]` element.
- The `passphrase` attribute is only required if the private key is encrypted with a passphrase. Do not include the attribute if it's not required.
- If using an encrypted private key, the following format is required (`PKCS1` is supported):

  ```
  -----BEGIN RSA PRIVATE KEY-----
  Proc-Type: 4,ENCRYPTED
  -----END RSA PRIVATE KEY-----
  ```

  A JSON file with an encrypted private key looks as follows:

  ```
  {
    "key": "-----BEGIN RSA PRIVATE KEY-----\nProc-Type:
  4,ENCRYPTED\n....\n-----END RSA PRIVATE KEY-----",
  ..
  ..
    "passphrase": "<passphrase to decrypt the key>"
  }
  ```

  A JSON file with an unencrypted private key looks as follows:

  ```
  {
    "key": "-----BEGIN RSA PRIVATE KEY-----
  \nvRXUK08v31bw2rnDLw+vjuX2i8ujHWs\n....\n-----END RSA PRIVATE
  KEY-----",
  ..
  ..
  }
  ```

- If your private key is in PKCS8 format, you must convert it to `PKCS1` format:

  ```
  openssl rsa -in <input_pkcs8_encrypted_private_key> -out
  <converted_encrypted_private_key_file_name> -aes256
  ```

> **Note**
>
> It is strongly recommended that you generate the certificate JSON from the Linux/Unix command line, or Unix utilities, to ensure that the line endings are encoded correctly. Incorrect line endings will result in an error.
>
> - To avoid manual errors, you can also convert your PEM certificate into a single line containing `"\n"`, as expected, with the following `awk` commands.
>   For the leaf certificate:
>
>   ```
>   awk -v RS= '{gsub(/\n+/, "\\n")}1' <cert_pem_file>
>   ```
>
>   For each intermediate/root certificate:
>
>   ```
>   awk -v RS= '{gsub(/\n+/, "\\n")}1'
>   <each_intermediate_cert_pem_file>
>   ```
>
>   For the private key:
>
>   ```
>   awk -v RS= '{gsub(/\n+/, "\\n")}1' <private_key_pem_file>
>   ```
> - The **latest** version of the secret is used when you associate a custom endpoint with your instance either through the create instance or edit instance operation. For information on secret versions, see Secret Versions and Rotation States.
> - If you use a hostname certificate whose certificate authority (CA) is not in the Visual Builder trust store, you must also upload the certificate to your Visual Builder instance; otherwise, an exception is thrown in the scenarios the instance calls itself.

Use the default settings for the other options in the Create Secret page.

g. Click **Create Secret** to return to the Secrets tab.

7. Create an Identity and Access Management (IAM) policy to:

   a. Allow the Visual Builder service to read the version and contents of the secret.

   Here's the policy syntax for a Visual Builder service:

   ```
   allow group <group-name> to read secret-bundle in compartment <secrets-compartment>
   ```

   Here's an example:

   ```
   allow group VBInstanceAdmins to read secret-bundle in compartment MySecretCompartment
   ```

   If the VB instance is NOT in a default domain, then you need to include the domain prefix in front of the group name.

   Here's an example:

   ```
   Allow group mydomain/VBInstanceAdmins to read secret-bundle in compartment MySecretCompartment
   ```

b. Allow the admin group to access the secret, key, and vault (or create a new secret, key, and vault), while creating or updating a Visual Builder instance with a custom endpoint.

   Here's the policy syntax:

   ```
   allow group <group-name> to manage secrets in compartment <secrets-compartment>

   allow group <group-name> to manage keys in compartment <secrets-compartment>

   allow group <group-name> to manage vaults in compartment <secrets-compartment>
   ```

   and here are examples:

   ```
   Allow group VBInstanceAdmins to manage secrets in compartment MySecretCompartment

   Allow group VBInstanceAdmins to manage keys in compartment MySecretCompartment

   Allow group VBInstanceAdmins to manage vaults in compartment MySecretCompartment
   ```

   If the VB instance is NOT in a default domain, then you need to include the domain prefix in front of the group name.

   Here are examples:

   ```
   Allow group mydomain/VBInstanceAdmins to manage secrets in compartment MySecretCompartment

   Allow group mydomain/VBInstanceAdmins to manage keys in compartment MySecretCompartment

   Allow group mydomain/VBInstanceAdmins to manage vaults in compartment MySecretCompartment
   ```

Note that you should specify the resource to return in `<resource-type>`, as described in Details for the Vault Service.

For the policy statement syntax, see CreatePolicy API Request.

Update a Secret in a Vault

When updating SSL certificates in a vault, you will need to create a new version of your secret. After updating the secret, you update your instance to start using the new secret.

> **Note**
>
> - You must use the command line to update the instance after you update the endpoint's secret. See Create and Update Alternate Endpoints.
> - If you are using WAF to manage your certificates, you update the certificates in the load balancer. See Create a Load Balancer and Configure a Hostname.

To update a custom endpoint's SSL certificate:

1. Open the OCI Console.

2. Update the SSL certificate.

   The steps for updating a certificate will depend upon if you have already created a vault in your tenancy.

   - If you are already managing the SSL certificates in a vault yourself, meaning you have already created a vault, perform the following steps to create a new version of the secret in your vault and update the certificate:

     a. Open the vault containing the certificate you want to update, then select the **Secrets** tab.

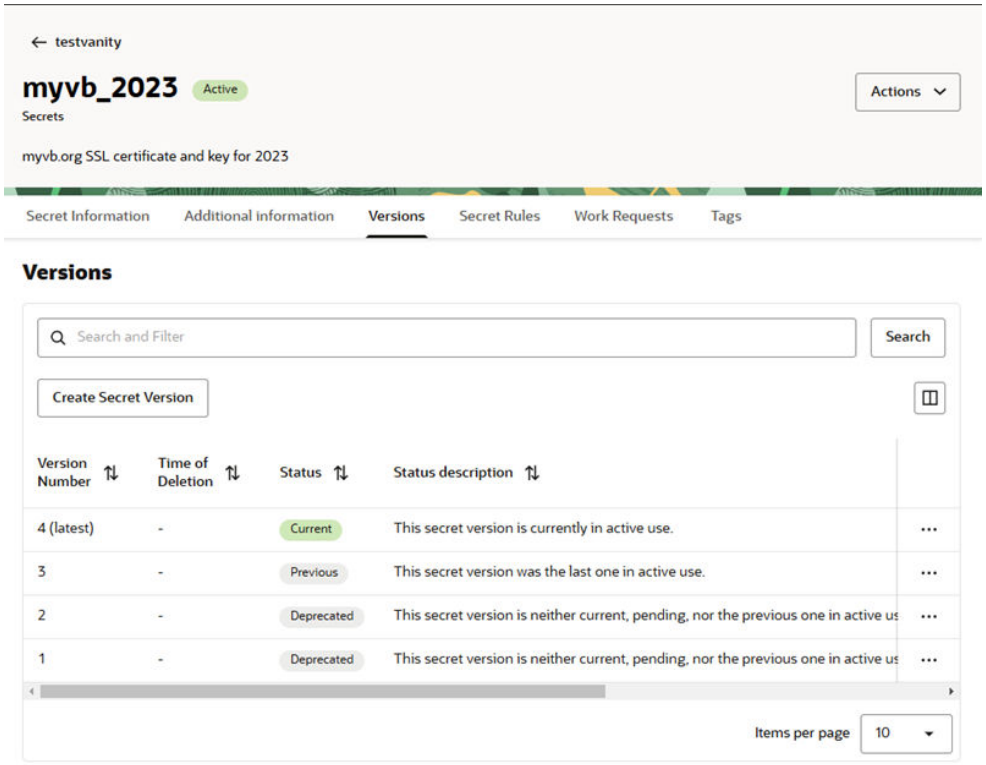
     b. In the Secrets tab, select the **Versions** tab, and then select **Create Secret Version**.

     c. In the Create Secret Version page, paste in the secret certificate JSON in the Secret Contents field.

     > **Note**
     >
     > When creating the secret certificate JSON, make sure the key and certificate are correct, and the JSON is correctly formated.
     >
     > It is strongly recommended that you generate the certificate JSON from the Linux/Unix command line, or Unix utilities, to ensure that the line endings are encoded correctly. Incorrect line endings will result in an error. For details on the correct certificate formatting, see Configure a Vault for a Custom Endpoint.

     Click **Create Secret Version**.

     After you create the new version, the Versions table is updated, and the new version is labeled "Current" in the Status column.



- If you have not been managing the SSL certificates in a vault yourself, meaning you have been using a vault created and managed by Oracle, you need to create a new vault in your tenancy before you can update your certificates.
  For example, Oracle Digital Cloud Service customers, after they have been migrated to their own tenancy, are responsible for managing their certificates, and might need to create a vault in their tenancy before they can update their certificates.

  Perform the following steps to create a vault and update the certificates:

  a. Create a vault and secret for the hostname used for your primary endpoint.
     You can see the details of your instance's primary endpoint in the Custom endpoint pane in the Visual Builder Instance details page.

     For the steps to create a vault and secret, see Configure a Vault for a Custom Endpoint.

3. Update your instance.

   You need to update your instance to start using the updated secret.

   - If you created a new version of the secret in your vault for a primary endpoint:

     a. Open the Visual Builder Instance details page.
        You will see a notification that a new secret has been created, and that you need to update your instance.

     b. From the **Actions** menu, select **Edit** to open the Edit visual builder instance panel.

     c. In the Edit visual builder instance panel, click **Save Changes** to update the instance with the new version of the secret.
        You do not need to change any of the custom endpoint settings.

   - If you created a new vault *and* secret for a primary endpoint:

     a. Open the Visual Builder Instance details page.

b. From the **Actions** menu, select **Edit** to open the Edit visual builder instance panel.

c. In the Custom endpoint pane, select the new vault and secret from the dropdown lists.
Do not change the hostname or compartment.

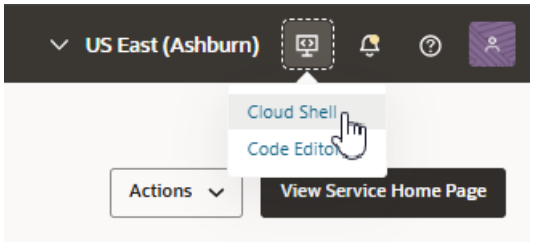
d. Click **Save Changes** to update the instance with the new vault and secret.

- If you created a new version of the secret in the vault for an alternate endpoint:
After creating a new version of the secret, you need to update the alternate endpoint for the new version. For alternate endpoints, you need to use the `update` command from the command line. You can run the command in the OCI Console's Cloud Shell editor.

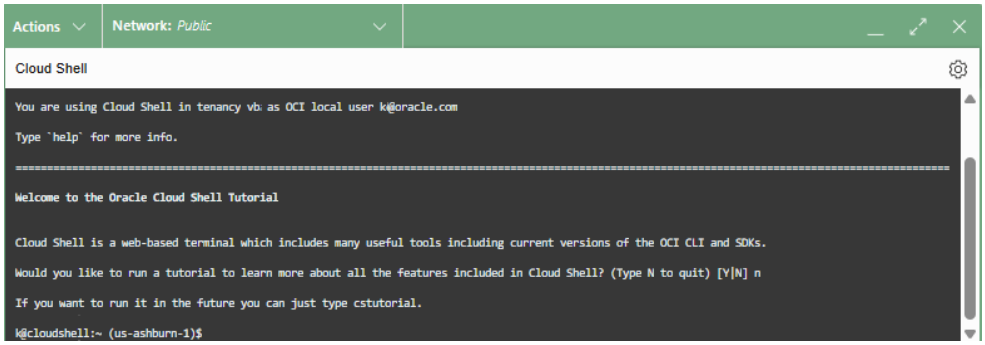
  When you run the `update` command, you don't need to explicitly specify the secret version because it is automatically updated to the most recent version.

  a. Open the Visual Builder Instance details page.

  b. Select the **Developer tools** menu in the header, and then select **Cloud Shell** to open the Cloud Shell editor.



The Cloud Shell editor opens in the bottom of your browser window:



It might take a minute for the editor to initialize.

  c. In the shell editor, check if the shell is working correctly by entering the following command at the prompt:

```
oci visual-builder vb-instance get --id <OCID>
```

For the `id` parameter, you need to provide the instance's OCID, which is listed in the Details tab. To copy the instance's <*OCID*>, click **Copy** next to the OCID.

The shell editor is context-sensitive, so the command should return details about the instance open in the details page.

  d. Run the `update` command with the `--alternate-custom-endpoints` parameter to update the alternate endpoints in the instance.

> **Note**
>
> When you run the command, confirm you have included the details of every alternate endpoint in the instance in the payload. **If you omit an alternate endpoint in the payload, that endpoint is deleted.**

In the `update` command, you need to provide the instance's OCID for the `id` parameter, and include a JSON array containing the details of **every** alternate endpoint in the instance as the payload of the `alternate-custom-endpoints` parameter:

- If you have any alternate endpoints using a vault to store certificates, you need to include in the payload the hostname and the certificate secret OCID of each endpoint:

  ```
  --alternate-custom-endpoints
  '[{"hostname":"hostname.com","certificateSecretId":"<SECRET_ID>"}
  ]'
  ```

- If you have any alternate endpoints using WAF for certificates, you only need to include the hostname of the alternate endpoints in the payload:

  ```
  --alternate-custom-endpoints '[{"hostname":"hostname.com"}]'
  ```

For example, if you have two alternate endpoints in your instance, and you want to update one of them, the `update` command might look something like this:

```
oci visual-builder vb-instance update --id <VB_INSTANCE_OCID>
--alternate-custom-endpoints
'[{"hostname":"hostname.com","certificateSecretId":"<SECRET_ID>"},
{"hostname":"hostname1.com","certificateSecretId":"<SECRET_ID>"}]'
```

Notice that although in this case you are only updating one endpoint, the `alternate-custom-endpoints` parameter payload contains the details for the instance's two alternate endpoints (`hostname.com` and `hostname1.com`).

Create and Update Alternate Endpoints

After adding the first custom endpoint (primary endpoint) to your instance, you need to use the command line in a shell when you want to update the instance to add more endpoints (alternate endpoints). The OCI Console provides a shell editor you can use to add and update alternate endpoints.

Additionally, when it comes time to update the SSL certificate in a secret, you need to use the command line to trigger an instance update after updating the secret in the associated vault. For details, see Update a Secret in a Vault.

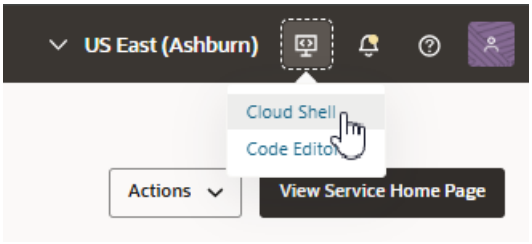
> **Note**
>
> If you have not been managing your instance yourself, meaning your instance was managed by Oracle, after you are migrated to your own tenancy you are responsible for managing your instance's alternate endpoints and associated vaults. This includes updating the SSL certificates for alternate endpoints.

To create and update alternate endpoints in a Visual Builder instance, you use the command line to send a JSON payload via the`vb-instance update` command. In the command, the payload is included as `alternate-custom-endpoints` parameters. For details on the `vb-instance update` command, see vb-instance update in the OCI CLI Command Reference, and UpdateCustomEndpointDetails Reference in the Visual Builder API.

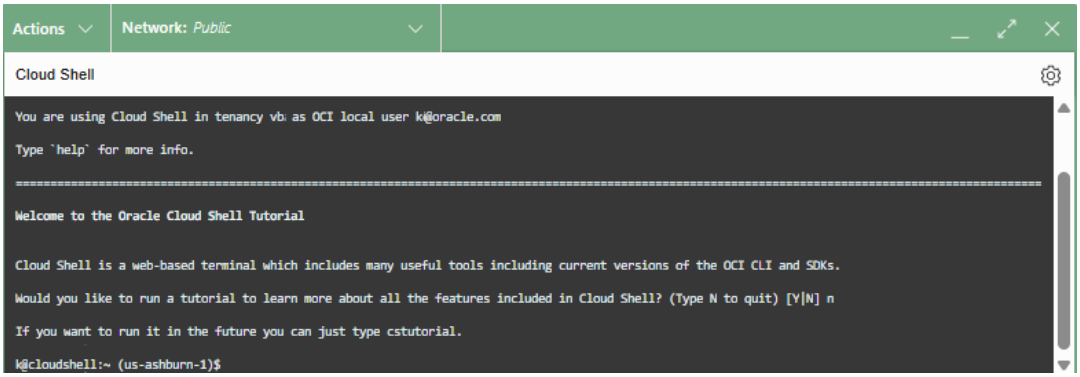
> **Warning**
>
> When updating alternate endpoint details using the command line, your payload must include the details of every alternate endpoint in your instance, including the details for endpoints not being updated. For example, if your instance has two alternate endpoints, and you want to update the secret in the vault for one of the alternate endpoints, the payload must still contain the details for both alternate endpoints.

To create or update an alternate endpoint:

1. On the Visual Builder Instances page, find the instance you want to work with and open its details page.
2. Select the **Developer tools** menu in the header, and then select **Cloud Shell** to open the Cloud Shell editor.



The Cloud shell editor opens in the bottom of your browser window:



It might take a minute for the editor to initialize.

3. In the shell editor, check that the shell is working correctly.

   The shell editor is context-sensitive, so the command returns details about the instance open in the details page.

   Enter the following `get` command at the prompt:

   ```
   oci visual-builder vb-instance get --id <OCID>
   ```

   For the `id` parameter, you need to provide the instance's OCID, which is listed in the Details tab. To copy the instance's <*OCID*>, click **Copy** next to the OCID.

   When you run the command, you should see details about the instance in the shell editor.

4. Run the `update` command in the shell editor.

   You use the `update` command to update existing alternate endpoints and to create new alternate endpoints.

   > **Note**
   >
   > In the `update` command, you need to provide the instance's OCID for the `id` parameter, and include a JSON array containing the details of every alternate endpoint in the instance as the payload of the `alternate-custom-endpoints` parameter:
   >
   > - If you are using a vault to store a certificate for an alternate endpoint, you need to include in the payload the hostname and the certificate secret OCID of each endpoint:
   >
   >   ```
   >   --alternate-custom-endpoints
   >   '[{"hostname":"hostname.com","certificateSecretId":"<SECRET_ID>"}
   >   ]'
   >   ```
   >
   > - If you are using WAF for an alternate endpoint's certificate, you only need to include the hostname in the payload:
   >
   >   ```
   >   --alternate-custom-endpoints '[{"hostname":"hostname.com"}]'
   >   ```
   >
   > - If you fail to include an endpoint in the payload when you run the `update` command, the endpoint is deleted.

   - To update the details of an alternate endpoint:
     Run the `update` command. When you run the command, confirm you have included the details of every alternate endpoint in the instance. For example, if you have two alternate endpoints in your instance, and you want to update one of them, the `update` command might look something like this:

     ```
     oci visual-builder vb-instance update --id <VB_INSTANCE_OCID>
     --alternate-custom-endpoints
     '[{"hostname":"hostname.com","certificateSecretId":"<SECRET_ID>"},
     {"hostname":"hostname1.com","certificateSecretId":"<SECRET_ID>"}]'
     ```

Notice that although in this case you are only updating one endpoint, the `alternate-custom-endpoints` parameter payload contains the details for the two alternate endpoints (`hostname.com` and `hostname1.com`).

- To create a new alternate endpoint:
  By default, you can create up to three alternate endpoints in your instance. If you need more than this, contact VB Dev Ops to increase the limit.

  a. Confirm you have configured the hostname for the new alternate endpoint using WAF or a vault and secret.
  For details, see Create a Load Balancer and Configure a Hostname and Configure a Vault for a Custom Endpoint.

  b. Run the `update` command. When you run the command, in addition to the details of the new endpoint, confirm you have included the details of every existing alternate endpoint in the instance, just as you would when updating alternate endpoint details. For example, if you have one alternate endpoints in your instance (`hostname.com`), and you want to create a new one (`hostname1.com`), the `update` command might look something like this:

  ```
  oci visual-builder vb-instance update --id <VB_INSTANCE_OCID>
  --alternate-custom-endpoints
  '[{"hostname":"hostname.com","certificateSecretId":"<SECRET_ID>"},
  {"hostname":"hostname1.com","certificateSecretId":"<SECRET_ID>"}]'
  ```

  Notice that the details you need to provide in the `update` command when updating alternate endpoint details is the same as when creating a new alternate endpoint.

  c. Configure the DNS record for the new endpoint.
  After creating an alternate endpoint, to configure the DNS record for the new endpoint you need to provide either the CNAME (the hostname) or the IP address of the load balancer.

  > **Note**
  >
  > The load balancer for an alternate endpoint can be different from the load balancer for the instance. You'll need to file a ticket with VB Dev Ops to verify the details. Note that this is a one-time action, so once configured, the load balancer details will not change.