

Oracle® Cloud

Terraform Scripts in Oracle WebLogic Server for OCI



F35192-13
April 2022



Oracle Cloud Terraform Scripts in Oracle WebLogic Server for OCI,

F35192-13

Copyright © 2020, 2022, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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

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

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

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

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

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

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

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

Contents

1	Terraform Scripts in Oracle WebLogic Server for OCI	
	About configuring Terraform Scripts	1-1
	Prerequisites	1-2
	Download a Terraform Configuration File	1-2
	Invoke Terraform Scripts	1-4
	Variables in Terraform Scripts	1-6
	Sample Scripts	1-20

1

Terraform Scripts in Oracle WebLogic Server for OCI

Learn to access the Terraform Scripts of an Oracle WebLogic Server for Oracle Cloud Infrastructure (Oracle WebLogic Server for OCI) and modify it as required.

The Oracle Cloud Infrastructure (OCI) Command Line Interface (CLI) is a small-footprint tool that you can use as a standalone or with the Console to complete Oracle Cloud Infrastructure tasks. The CLI provides the same core functionality as the Console, plus additional commands. Some of these, such as the ability to run scripts, extend Console functionality.

Note:

Support ended for Terraform v0.11.x. So, you would not be able to scale out or destroy the stacks created with Terraform v0.11.x. However, you can configure the stacks in your local environment by downloading the terraform configurations. See Terraform Scripts Migration.

Topics:

- [About configuring Terraform Scripts](#)
- [Prerequisites](#)
- [Download a Terraform Configuration File](#)
- [Invoke Terraform Scripts](#)
- [Variables in Terraform Scripts](#)
- [Sample Scripts](#)

About configuring Terraform Scripts

You can create a single or multi-node Oracle WebLogic Server cluster with Oracle Cloud Infrastructure database or Oracle Autonomous Transaction Processing (ATP) database as an infra database, optionally, fronted-by a load balancer. Here you would create a stack and modify it as required.

When you create an instance by using the scripts, you will create a public or private subnet in Oracle WebLogic Server for OCI.

By default subnets span an entire region in Oracle Cloud Infrastructure. Oracle WebLogic Server for OCI supports both regional and AD-scoped subnets. AD subnets are now supported for only existing subnets. In case of regional subnets, single regional subnet is created.

Prerequisites


Prerequisites you need to complete before you update the Terraform scripts in Oracle WebLogic Server for OCI.

- An Oracle Cloud Infrastructure tenancy.
- The OCID for the compartment where you wish to create your stack.
- A user account that includes the following:
 - An API signing key. See [Required Keys and OCIDs](#).
 - Required IAM permissions. See [How Policies Work](#).
 - If you want to use the Oracle Cloud Infrastructure CLI, install and configure the CLI first. See [Installing the CLI](#).
- Install and Configure Terraform. See [OCI Terraform Provider Configuration on Linux and Windows machine](#).

Download a Terraform Configuration File

Create a Oracle WebLogic Server for OCI stack and download the Terraform configuration file.

Complete the following steps:

1. Create a Stack. See [Get Started with Oracle WebLogic Server for Oracle Cloud Infrastructure](#).
2. Click the navigation menu , and select **Developer Services**. Under the **Resource Manager** group, click **Stacks**.
3. Select the **Compartment** that contains your stack.
4. Click the name of your stack.
5. In the **Stack Information** tab, click **Download** against **Terraform Configuration File (.zip)**.
6. Unzip the terraform configuration files to a folder.
The terraform configuration file includes the following files:
 - `provider.tf`
 - `provider.tf.cli`
 - `provider.tf.cloudshell`
7. If you are using CLI, replace the contents of the `provider.tf` file with the contents from the `provider.tf.cli` file.

If you are using Cloud Shell, replace the contents of the `provider.tf` file with the contents from the `provider.tf.cloudshell` file.

Following is a copy of the contents of the terraform configuration files:

- **provider.tf** file:

```
variable "user_id" {
}
variable "fingerprint" {
}
variable "private_key_path" {
}
variable "disable_auto_retries" {
  default = true
}
provider "oci" {
  version          = "=4.30.0"
  tenancy_ocid    = var.tenancy_ocid
  user_ocid       = var.user_id
  fingerprint     = var.fingerprint
  private_key_path = var.private_key_path
  region          = var.region
}
provider "oci" {
  version          = "=4.30.0"
  alias            = "home"
  region          = local.home_region
  tenancy_ocid    = var.tenancy_ocid
  user_ocid       = var.user_id
  fingerprint     = var.fingerprint
  private_key_path = var.private_key_path
  disable_auto_retries = var.disable_auto_retries
}
```

- **provider.tf.cli** file:

```
/*
 * Copyright (c) 2022, Oracle and/or its affiliates. All rights reserved.
 */

/* Variables and provider plugin defined here are required for CLI */

variable "user_id" {
}

variable "fingerprint" {
}

variable "private_key_path" {
}

provider "oci" {
  version          = "=4.65.0"
  tenancy_ocid    = var.tenancy_ocid
  user_ocid       = var.user_id
  fingerprint     = var.fingerprint
  private_key_path = var.private_key_path
  region          = var.region
}
```

```
provider "oci" {
  version      = "=4.65.0"
  alias        = "home"
  region       = local.home_region
  tenancy_ocid = var.tenancy_ocid
  user_ocid    = var.user_id
  fingerprint  = var.fingerprint
  private_key_path = var.private_key_path
}
```

- **provider.tf.cloudshell** file:

```
/*
 * Copyright (c) 2022, Oracle and/or its affiliates. All rights
 reserved.
 */

/* Variables and provider plugin defined here are required for
cloud shell */

variable "user_id" {
}

variable "fingerprint" {
}

variable "private_key_path" {
}

provider "oci" {
  version = "=4.65.0"
  tenancy_ocid = var.tenancy_ocid
  region = var.region
}

provider "oci" {
  version = "=4.65.0"
  alias = "home"
  region = local.home_region
  tenancy_ocid = var.tenancy_ocid
}
```

Invoke Terraform Scripts

Use specific commands to invoke the Terraform scripts in Oracle WebLogic Server for OCI.



Note:

You can download a previous version of the terraform script from: <https://releases.hashicorp.com/terraform/>

Topics:

- [Invoke terraform scripts in an infrastructure](#)
- [Update an infrastructure](#)
- [Destroy an infrastructure](#)

Invoke terraform scripts in an infrastructure

Complete the following steps:

1. Go to the directory where you unpacked the **Terraform Configuration File (.zip)** file using [Download a Terraform Configuration File](#).
2. In this directory, create a terraform file: `terraform.tfvars`
3. In the `terraform.tfvars` file, add the variables that are referenced in the `provider.tf` file.
These are the variables that you noted in the `provider.tf` file in [Download a Terraform Configuration File](#).

For a sample of `terraform.tfvars` file, see [Sample Scripts](#).

4. Run the following command to initialize the terraform provider plugin:

```
$ terraform init
```

5. Add in the variables required for the desired topology to the `terraform.tfvars` file.
For the list of variables, see [Variables in Terraform Scripts](#) and [Sample Scripts](#).
6. Run the following command to invoke apply:

```
$ terraform apply
```

7. *Optional:* Run the following command to create multiple instances from same solutions:

```
$ terraform apply -state=<file_name>
```

Where, `<file_name>` is the unique directory name or state file name for each stack.

Update an infrastructure:

Complete the following steps:

1. Go to the directory where you unpacked the **Terraform Configuration File (.zip)** file using [Download a Terraform Configuration File](#).
2. In this directory, create a terraform file: `terraform.tfvars`
3. In the `terraform.tfvars` file, add the variables that are referenced in the `provider.tf` file.
These are the variables that you noted in the `provider.tf` file in [Download a Terraform Configuration File](#).

For a sample of `terraform.tfvars` file, see [Sample Scripts](#).

4. Update the variables required for the desired topology to the `terraform.tfvars` file. For the list of variables you can update, see [Variables in terraform scripts](#).
5. Run the following command to invoke apply:

```
$ terraform apply
```


Destroy an infrastructure

Depending on your environment, run the following command to destroy an infrastructure:

```
$ terraform destroy
```

Tip:

You can invoke terraform scripts by using the Resource Manager. See [Get Started with Oracle WebLogic Server for Oracle Cloud Infrastructure and Managing Stacks and Jobs](#).

Variables in Terraform Scripts

The variables you need input to the terraform scripts in Oracle WebLogic Server for OCI.

Criteria for VCN peering

If `wls_vcn_name` or `wls_existing_vcn_id` and `ocidb_existing_vcn_id` are passed together and `vcn_ids` are different, complete the following steps:

1. For an existing DB subnet, set the following stateful security rules:
 - Allow WebLogic VCN CIDR (for example, 11.0.0.0/16) access to the database port 1521.
 - Allow database DNS subnet CIDR (for example, 10.0.7.0/24) access to TCP/53 and UDP/53 ports.
 - Setup the database to use the default DHCP options for the VCN.
2. For an existing WebLogic subnet, set the following stateful security rules:
 - Allow 0.0.0.0/0 access to ICMP/3,4
 - Allow 0.0.0.0/0 access to TCP/22
 - Allow 0.0.0.0/0 access to TCP/7001-7002 (optional, only required for accessing console).
7001 and 7002 are examples, use the ports used for WebLogic server admin console port and admin console SSL port.
 - Allow LB subnet CIDRs access to TCP/7003-7004 ports.
7003 and 7004 are example, use the ports used for WebLogic managed server port and managed server SSL port.
 - Allow WebLogic subnet CIDR (for example, 11.0.3.0/24) access to TCP/All ports. This is for VM to VM communication.
 - Allow WLS DNS subnet CIDR (for example, 11.0.7.0/24) access to TCP/53 and UDP/53 ports.
 - WebLogic subnet should be setup to use the default DHCP options for the VCN.

The following table lists all the variables in terraform scripts:

Table 1-1 Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
Authentication Information					
Note: Use <code>env_vars_template</code> to create <code>env_vars</code> and source it as: <code>source ./env_vars</code> before running <code>terraform init</code> .					
Fingerprint	String	-	-	Yes	Fingerprint of your public key.
Path to private key	String	-	-	-	Path to the private key that matches the fingerprint.
Tenancy OCID	String	-	-	-	OCID of the tenancy in which you want to perform changes.
User OCID	String	-	-	-	OCID of the signed in user. That is, your OCID.
WebLogic Server Variables					
compartment_ocid	String	-	-	-	OCID of the compartment for Weblogic instances.
instance_shape	String	-	-	Yes	Shape for all WebLogic Server compute instances.
region	String	-	-	-	Region for provisioning.
service_name	String	-	-	-	Prefix for stack resources. The names of all the related compute and network resources begins with the prefix you assign here.

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
wls_admin_password_ocid	String	-	-	-	OCID of the Secret that contains the password for the administrator in the WebLogic Server domain.
wls_admin_user	String	weblogic	-	-	Name of the administrator in the WebLogic Server domain.
wls_node_count	Number	1	-	Yes	Number of WebLogic server compute instances and the number of managed servers in the domain. The maximum is node count is 8. For 11g Standard Edition, the maximum is 4.
wls_ocpu_count	Number	1	-	Yes	Number of OCPU count for instances. Note: Required only for VM.Standard.E3.Flex shape. The maximum is 64.
wls_version	String	12.2.1.4	-	-	WebLogic server version. Supported versions: <ul style="list-style-type: none"> • 12c (12.2.1.3): 12.2.1.3 • 12c (12.2.1.4): 12.2.1.4 • 14c (14.1.1.0): 14.1.1.0
create_service_tag	Boolean	false	-	-	Indicates if you want create a service tag.

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
service_tag	Object	freeformTags , definedTags	-	-	The type of service type to be used.
wls_node_count_limit	Number	8	-	-	The maximum number of WebLogic managed servers.
bastion_instance_shape	String	VM.Standard2 .1	-	-	The default shape of bastion VM instances.
wls_nm_port	String	5556	-	-	The node manager port.
wls_external_admin_port	String	7001	-	-	The WebLogic console port.
wls_external_admin_ssl_port	String	7002	-	-	The WebLogic console SSL port.
wls_cluster_multi_port	String	5555	-	-	The WebLogic multi-cluster port.
wls_admin_port	String	9071	-	-	The WebLogic admin port.
wls_ssl_admin_port	String	9072	-	-	The WebLogic admin SSL port.
wls_managed_external_port	String	7003	-	-	The WebLogic managed server external HTTP port.
wls_managed_external_ssl_port	String	7004	-	-	The WebLogic managed server external SSL port.
wls_managed_server_port	String	9073	-	-	The WebLogic managed server port.
wls_managed_server_ssl_port	String	9074	-	-	The WebLogic managed server SSL port.
allow_manual_domain_extension	Boolean	false	-	-	Indicates if the domain is manually extended for managed servers.

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
wls_14c_jdk_version	-	jdk8	-	-	The JDK version.
General	-	-	-	-	-
create_policies	Boolean	true	-	-	Create policies to read Secrets from Vault and manage ATP database (if applicable).
deploy_sample_app	Boolean	true	-	-	Indicates if you can deploy a sample application.
Network Variables	-	-	-	-	-
assign_weblogic_public_ip	Boolean	true	-	-	Indicates if you are using any private subnets. If <i>false</i> , the WebLogic server compute instances are created in a new private subnet.
bastion_ssh_private_key	String	-	-	-	Private ssh key for existing bastion instance. See Additional information .
bastion_subnet_cidr	String	-	-	-	CIDR of the new public subnet to create for a Bastion compute instance. This field is required only if you do not assign public IP addresses to the WebLogic server.

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
bastion_subnet_id	String	-	-	-	An existing public subnet to use for a Bastion compute instance. This subnet must already be present in the chosen VCN and required only if you do not assign public IP addresses to the WebLogic server.
bastion_subnet_name	String	-	-	-	Name of new bastion subnet.
existing_bastion_instance_id	String	-	-	-	OCID for existing bastion instance.
is_bastion_instance_required	Boolean	true	-	-	Creates bastion for the stack. If true, it provisions a bastion compute instance on a public subnet to provide access to the WebLogic server compute instances on a private subnet.
network_compartment_id	String	-	Yes	-	Compartment where you want to create the network resources, such as: Virtual Cloud Network (VCN), security lists, route tables and gateways.
use_regional_subnet	Boolean	true	-	-	Indicates use of regional subnets.

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
wls_admin_port_source_cidr	String	-	Yes	-	Create a security list to allow access to the WebLogic Administration Console port to the source CIDR range. Note: Keeping the default 0.0.0.0/0 CIDR exposes the console to the internet. You must change the CIDR range to allow access to a trusted IP range.
wls_availability_domain_name	String	-	-	-	Name of the availability domain where you want to create the WebLogic server compute instances.
wls_existing_vcn_id	String	-	-	-	OCID of an existing VCN where you want to create the compute instances, network resources, and load balancers.

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
wls_expose_admin_port	Boolean	-	-	-	Indicates if you are enabling access to administration console port. Note: Enabling this option exposes the console to the internet if the default 0.0.0.0/0 CIDR is used. In the wls_admin_port_source_cidr variable, specify the CIDR range to allow access to a trusted IP range.
wls_subnet_cidr	String	-	-	-	CIDR of the new subnet, which would be used in the WebLogic server compute instances.
wls_subnet_id	String	-	-	-	OCID for existing subnet for weblogic instances.
wls_subnet_name	String	-	-	-	Name of new subnet.
wls_vcn_cidr	String	-	-	-	CIDR to assign to the new VCN) to create for the service. This field is not required if you want to use an existing VCN.
wls_vcn_name	String	-	-	-	Name of the new VCN to create for the service.
lb_subnet_1_availability_domain_name	String	-	-	-	Availability domain for load balancer

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
lb_subnet_1_availability_domain_name	String	-	-	-	Availability domain for load balancer
is_bastion_with_reserved_public_ip	Boolean	false	-	Yes	Indicates if a reserved public IP is created for the bastion instance.
Load Balancer Variables	-	-	-	-	-
add_load_balancer	Boolean	false	-	-	If you want to provision a load balancer in Oracle Cloud Infrastructure to distribute application traffic to the managed servers in the domain.
is_lb_private	Boolean	false	-	-	If you want to use a private load balancer.
lb_min_bandwidth	Number	10	-	Yes	Minimum size of the flexible load balancer shape.
lb_max_bandwidth	Number	400	-	Yes	Maximum size of the flexible load balancer shape.
lb_subnet_1_cidr	String	-	-	Yes	CIDR of new primary regional subnet.
lb_subnet_1_id	String	-	-	-	OCID for existing regional subnet for primary load balancer.
lb_subnet_1_name	String	-	-	-	Name of new primary regional subnet.
lb_subnet_2_cidr	String	-	-	-	CIDR of secondary regional subnet.

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
mount_target_subnet_cidr	String	-	-	-	CIDR of new primary regional subnet.
mount_target_subnet_id	String	-	-	-	OCID for existing subnet for mount target.
lb_subnet_2_id	String	-	-	-	OCID for existing AD subnet for secondary load balancer.
add_lb_reserved_public_ip_ocid	Boolean	false	-	-	Indicates use of reserved public IP for Public Load balancer.
lb_reserved_public_ip_ocid	String	-	-	-	Value for load balancer reserved public IP OCID
IDCS-related Variables	-	-	-	-	-
idcs_client_id	String	-	-	-	IDCS client ID value.
idcs_client_secret_ocid	String	-	-	-	IDCS client secret OCID value.
idcs_cloudgate_port	Number	9999	-	-	IDCS cloud gate port value.
idcs_host	String	identity.oraclecloud.com	-	-	IDCS host value.
idcs_port	Number	443	-	-	IDCS port value.
idcs_tenant	String	-	-	-	IDCS tenant value.
is_idcs_selected	Boolean	false	-	-	If you want an IDCS has to be provisioned.
FSS Variables	-	-	-	-	-
add_fss	Boolean	false	-	-	Add FSS to instances.
fss_availability_domain	String				AD for FSS subnet.
mountTarget_id	String				Existing mount target ID.

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
mountTarget_ compartment_ id	String				Compartment OCID for the mount target.
mount_path	String	/u01/shared			Default mount path for FSS.
VCN Peering Variables for OCI Database	-	-	-	-	-
dns_instance_ _shape	String	-	-	-	Shape of the DNS instance
ocidb_dns_su bnet_cidr	String	-	-	-	CIDR value of the subnet to be used for database DNS instance.
use_local_vc n_peering	Boolean	true	-	-	Indicates use of VCN peering, if database and WebLogic server are on different VCNs.
wls_dns_subn et_cidr	String	-	-	-	CIDR value of the subnet to be used for DNS instance.
OCI Database Variables	-	-	-	-	-
oci_db_passw ord_ocid	String	-	-	-	OCID of the OCI database password.
oci_db_user	String	sys	-	-	OCI database username.
ocidb_compar tment_id	String	-	-	-	OCID of the OCI database compartment.
ocidb_databa se_id	String	-	-	-	OCID of the OCI database.
ocidb_dbhome _id	String	-	-	-	ID of the OCI database system DB home.
ocidb_dbsyst em_id	String	-	-	-	OCID of the OCI database system.

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
ocidb_existing_vcn_add_security_list	Boolean	true	-	-	If you want a security list to the database subnet that allows connections from the WebLogic server subnet.
ocidb_existing_vcn_id	String	-	-	-	VCN ID of the existing OCI database.
ocidb_network_compartment_id	String	-	-	-	OCID of the OCI database network. It is usually same as the <code>ocidb_compartment_id</code> variable.
ocidb_pdb_service_name	String	-	-	-	PDB name of the OCI database.
db_port	Number	1521	-	-	Value for OCI database port.
use_database_connect_string	Boolean	false	-	-	Use OCI database connect string.
db_connect_string	String	-	-	-	Value of OCI DB connect string.
ATP Database Variables	-	-	-	-	-
atp_db_compartment_id	String	-	-	-	OCID of the ATP database compartment.
atp_db_id	String	-	-	-	OCID of the ATP database.
atp_db_level	String	-	-	-	ATP database level value.
atp_db_password_ocid	String	-	-	-	OCID of the ATP database password.
OCI Database-related (used for AppDB) Variables	-	-	-	-	-

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
app_db_passw ord_ocid	String	-	-	-	OCID of the Application database password.
app_db_user	String	-	-	-	User name of the Application database.
appdb_compar tment_id	String	-	-	-	OCID of the Application database compartment.
appdb_databa se_id	String	-	-	-	OCID of the Application database.
appdb_dbhome _id	String	-	-	-	ID of the Application database system DB home.
appdb_dbsyst em_id	String	-	-	-	OCID of the Application database system.
appdb_existi ng_vcn_add_s eclist	Boolean	true	-	-	If you want a security list to the Application database subnet that allows connections from the WebLogic server subnet.
appdb_existi ng_vcn_id	String	-	-	-	VCN ID of the existing Application database.
appdb_networ k_compartmen t_id	String	-	-	-	OCID of the Application database network.
appdb_pdb_se rvice_name	String	-	-	-	PDB name of the Application database.
appdb_port	Number	1521	-	-	Database port number.
configure_ap p_db	Boolean	false	-	-	Indicates that the Application Database is requested.

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
ATP Database-related (used for AppDB) Variables	-	-	-	-	-
app_atp_db_compartment_id	String	-	-	-	Compartment where the ATP database for the Application database resides.
app_atp_db_id	String	-	-	-	OCID of the selected ATP database.
app_atp_db_level	String	low	-	-	ATP database DB level. Supported levels: <ul style="list-style-type: none"> • low • tp • tp_urgent
app_atp_db_password_ocid	String	-	-	-	Secret OCID that contains the ATP database user password.
app_atp_db_username	String	-	-	-	User name for creating the datasource.
VCN Peering Variables for Application Database	-	-	-	-	-
appdb_wls_dns_subnet_cidr	String	-	-	-	CIDR value of the subnet to be used for DNS instance.
appdbdns_instance_shape	String	-	-	-	Shape of the DNS instance.
ociappdb_dns_subnet_cidr	String	-	-	-	CIDR value of the subnet to be used for database DNS instance.
OCI Logging	-	-	-	-	-

Table 1-1 (Cont.) Variables in terraform scripts

Variables	Type	Default Value	Optional	Can be updated?	Description
dynamic_group_oci	String	-	-	-	Dynamic group OCID for OCI logging agent configuration, when the create policies is not set.
use_oci_logging	Boolean	false	-	-	Indicated if OCI logging is enabled.

 **Note:**

Support for existing bastion host to be used in provisioning WebLogic server with private subnet is enabled in terraform CLI only. This can be achieved by using the variables: `is_bastion_instance_required`, `existing_bastion_instance_id`, and `bastion_ssh_private_key`. For existing WebLogic server subnet, you will need to open port 22 for bastion IP/subnet CIDR. For a new WebLogic server subnet we create security list with bastion private IP.

Sample Scripts

Following are a few sample scripts in Oracle WebLogic Server for OCI.

Following is a sample script for file: `terraform.tfvars`:

```
tenancy_oci = "<Tenancy OCID>"
user_id = "<User OCID>"
fingerprint = "<Fingerprint>"
private_key_path = "<path-to-oci_api_key.pem>"
compartment_oci = "<Stack Compartment OCID>"
region = "<Region>"

// Stack config
service_name = "<Stack Service Prefix>"
wls_edition = "EE"
wls_version = "12.2.1.4"

// WLS Instance config
instance_shape = "<WebLogic Instance Shape-example-VM.Standard.E3.Flex>"
instance_image_id = "<WebLogic Image OCID>"
wls_admin_password_oci = "<WebLogic Admin Password Secret OCID>"
wls_node_count = 2
ssh_public_key = "<SSH Public Key text>"
```

```
// Network config - Existing VCN, Create New Subnet
network_compartment_id = "<Network Compartment OCID>"
wls_existing_vcn_id = "<VCN OCID>"
wls_subnet_cidr = "10.0.1.0/24"
bastion_subnet_cidr = "10.0.2.0/24"
bastion_instance_shape = "<Bastion Instance Shape-example-
VM.Standard.E3.Flex>"
add_load_balancer = true
lb_subnet_1_cidr = "10.0.3.0/24"
lb_max_bandwidth = "100"
```

Following is a sample script for file: **env-var.sh**:

```
# Use this template to create a file env_vars and source it before running
terraform.

### Region
export TF_VAR_region="us-phoenix-1"

### Public/private keys used on the instance
export TF_VAR_ssh_public_key=$(cat ~/xperiment_rsa.pub)
export TF_VAR_ssh_private_key=$(cat ~/xperiment_rsa)

## NOTE: These are not your api keys. More info on the right keys see
## https://docs.oracle.com/en-us/iaas/Content/Compute/Tasks/
managingkeypairs.htm
## https://docs.cloud.oracle.com/iaas/Content/API/Concepts/apisigningkey.htm
```

Following is a sample script for file: **oci_db.tfvars**. Use this file to provision WLSC with OCI database, along with instance.tfvars.

```
#DB VCN ID
ocidb_existing_vcn_id = "<VCN OCID>"

#DB Compartment
ocidb_compartment_id = "<Compartment OCID>"

#DB Network Compartment
#ocidb_network_compartment_id = "<Network Compartment OCID>"

#DB System
ocidb_dbssystem_id = "<OCID>"

#Database
ocidb_database_id = "<OCID>"

#PDB Name
ocidb_pdb_service_name = "PDB"

#Provide DB user creds.
oci_db_user = "sys"
oci_db_password_ocid = "<OCID>"
```



```
#VCN peering param for OCI DB
ocidb_dns_subnet_cidr = "10.0.7.0/11"
```

Following is a sample script for file: **atp_db.tfvars**. Use this file to provision WLSC with ATP database, along with `instance.tfvars`.

```
atp_db_level = "low"
atp_db_id = "<OCID>"
atp_db_password="<password>"
atp_db_compartment_id="<OCID>"
```

Following is a sample script for file: **instance_idcs.tfvars**. This is a sample template for provisioning 2 node WLSC, an existing VCN, an existing public subnet, loadbalancer, and IDCS.

```
// Region for provisioning
region = "<region>"

// WLS config
service_name = "WLSCIDCS"

// WLS instance shape
instance_shape = "VM.Standard2.2"

// WLS instances count
wls_node_count = "2"

// Add LB
add_load_balancer = true
lb_use_https = false

// Add IDCS
is_idcs_selected = true
idcs_host = "<host>"
idcs_port = "443"
idcs_tenant = "<tenant>"
idcs_client_id = "<ID>"
idcs_client_secret_ocid = "<OCID>"
idcs_cloudgate_port = "9999"

// WLS admin creds
wls_admin_user = "weblogic"
wls_admin_password_ocid = "<OCID>"

// WLS custom image ID
instance_image_id = "<OCID>"

// WLS and VCN compartment ID
compartment_ocid = "<OCID>"
network_compartment_id = "<OCID>"

// WLS existing VCN
wls_existing_vcn_id = "<OCID>"
```

```
// WLS existing VCN
wls_subnet_id = "<OCID>"

// LB existing Subnet
lb_subnet_1_id = "<OCID>"
#lb_subnet_2_id = "" -- not used for regional subnet - uncomment for non-
regional.

wls_edition = "EE"

wls_version = "12.2.1.4"
// "12.2.1.4", 14.1.1.0

// Creating policies
create_policies = true

// Private Subnet Config
assign_weblogic_public_ip = true

// Regional subnet config
use_regional_subnet = true

wls_console_port="7001"
```

Following is a sample script with all variables: `instance.tfvars`:

```
// Region for provisioning
region = "<region>"

// WLS config
service_name = "abc"

// WLS Edition
wls_edition = "EE"
# SE, EE, SUITE

// WLS Version
wls_version = "12.2.1.4"
# 12.2.1.4, 14.1.1.0

// WLS instance shape
instance_shape = "VM.Standard2.2"

// WLS instances count
wls_node_count = "1"

// Database port
db_port = "1521"

// Add LB
add_load_balancer = false
lb_use_https = false
is_lb_private= false
```

```
// Add IDCS
is_idcs_selected = true
idcs_host = "<host>"
idcs_port = "443"
idcs_tenant = "<tenant>"
idcs_client_id = "<ID>"
idcs_client_secret_ocid = "<OCID>"
idcs_cloudgate_port = "9999"

// WLS admin Credentials
wls_admin_user = "<User>"
wls_admin_password_ocid = "<password OCID>"

// WLS custom image ID
instance_image_id = "<OCID>"

// WLS and VCN compartment ID
compartment_ocid = "<OCID>"
network_compartment_id = "<OCID>"

// Existing_vcn
wls_existing_vcn_id="<OCID>"

// Existing_subnet
wls_subnet_id="<OCID>"

// Existing LB Subnet
lb_subnet_id="<OCID>"

// Existing Bastion Subnet
bastion_subnet_id="<OCID>"

// This is for New VCN
wls_vcn_name="wls_vcn"

// WLS subnet CIDR, LB Backend and Frontend Subnet CIDRs for new
subnets that will be created.
wls_subnet_cidr = "20.0.0.0/10"

lb_subnet_1_cidr = "10.0.94.0/11"
lb_subnet_2_cidr = "10.0.95.0/11"
// not used for regional subnet - uncomment for non-regional.

// Bastion host subnet CIDR.
// Required for existing VCN and if not using existing bastion subnet
(see bastion_subnet_id)
// And if assign_weblogic_public_ip = false (using private subnet).
bastion_subnet_cidr="10.0.86.0/11"

// VCN Peering Parameters
ocidb_dns_subnet_cidr = "10.0.7.0/11"
wls_dns_subnet_cidr = "13.0.7.0/11"

// Private Subnet Config
```

```
assign_weblogic_public_ip = true

// Regional subnet config
use_regional_subnet = true

// WLS Console Port
wls_console_port="7001"
```

Following is a sample structure to create free-form tags and defined tags for the `service_tag` field. Use this sample structure in the variables if you want to assign tags to the resources.

```
# create freeformTags
service_tag={"freeformTags"={"free_tag1"="free_tag1_value","free_tag2"="free_
tag2_value"},"definedTags"={}}

# create definedTags
service_tag={"freeformTags"= {}, "definedTags"={"wls-tag-namespace.wls-oci-
tag1"="test1","wls-tag-namespace.wls-oci-tag2"="test2"}}
#Note: Ensure that the namespace, wls-tag-namespace used in the example is
already created.

# create both freeformTags and definedTags
service_tag={"freeformTags"={"free_tag1"="free_tag1_value","free_tag2"="free_
tag2_value"} ,"definedTags"={"wls-tag-namespace.wls-oci-tag1"="test1","wls-
tag-namespace.wls-oci-tag2"="test2"}}
```