Oracle® Cloud

Terraform Scripts in Oracle WebLogic Server for OKE (Release 21.3.2 or earlier)

F47894-02
September 2021

Oracle Cloud Terraform Scripts in Oracle WebLogic Server for OKE (Release 21.3.2 or earlier),

F47894-02

Primary Author: Oracle Corporation

Contents

1 Terraform Scripts in Oracle WebLogic Server for OKE

About Configuring Terraform Scripts 1-1

Prerequisites 1-2

Download a Terraform Configuration File 1-2

Invoke Terraform Scripts 1-3

Variables in Terraform Scripts 1-4

Sample Scripts 1-11

1

Terraform Scripts in Oracle WebLogic Server for OKE

Learn to access the Terraform Scripts of an Oracle WebLogic Server for Oracle Cloud Infrastructure Container Engine for Kubernetes (Oracle WebLogic Server for OKE), and modify it as required.

> **Note:**
>
> If you are using Oracle WebLogic Server for OKE (**Release 21.3.3 or later**), see Terraform Scripts in Oracle WebLogic Server for OKE.

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.

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 database (ATP) as an infra database, optionally, fronted-by a load balancer. Here you would create a stack and modify them as required.

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

By default subnets span an entire region in Oracle Cloud Infrastructure. Oracle WebLogic Server for OKE 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 OKE.

- 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 OKE stack, download the terraform configuration file, and update the `provider.tf` file.

Complete the following steps:

1. Create a Stack. See Get Started with Oracle WebLogic Server for OKE.
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.
7. The terraform configuration files includes the `provider.tf` file that you cannot use for the CLI option.
   Update the contents of the terraform file `provider.tf`, with the following content:

```
# Copyright  2019, 2021 Oracle Corporation and/or affiliates.  All
rights reserved.
# Licensed under the Universal Permissive License v 1.0 as shown at
http://oss.oracle.com/licenses/upl
# Identity and access parameters
variable "api_private_key_path" {
  description = "path to oci api private key"
}
variable "api_fingerprint" {
  description = "fingerprint of oci api private key"
}
variable "user_id" {
  type        = string
  description = "user id"
}
# general oci parameters
variable "disable_auto_retries" {
  default = true
}
provider "oci" {
  version              = ">=4.7.0"
  tenancy_ocid         = var.tenancy_ocid
  user_ocid            = var.user_id
  fingerprint          = var.api_fingerprint
  private_key_path     = var.api_private_key_path
  region               = var.region
  disable_auto_retries = var.disable_auto_retries
}
provider "oci" {
  version              = ">=4.7.0"
  alias                = "home"
  region               = local.home_region
  tenancy_ocid         = var.tenancy_ocid
  user_ocid            = var.user_id
  fingerprint          = var.api_fingerprint
  private_key_path     = var.api_private_key_path
  disable_auto_retries = var.disable_auto_retries
}
```

Invoke Terraform Scripts

Use specific commands to invoke the terraform scripts in Oracle WebLogic Server for OKE.

Topics:

- Invoke terraform scripts in an infrastructure
- Update an infrastructure
- Destroy an infrastructure

The following section reference files in the input directory that you need create. Depending on the type of stack (JRF or non-JRF), create the appropriate files in the inputs directory. For information about the input files, see Sample Scripts.

To invoke terraform scripts in an infrastructure:

Complete the following steps:

1. Go to the directory, where you unzipped the terraform configuration files.
2. Initialize the terraform provider plugin:

   ```
   $ terraform init
   ```

3. Initialize the environment with terraform environment var files:

   ```
   $ source inputs/env_vars
   ```

4. Invoke apply passing all `*.tfvars` files as input:

> **Tip:**
>
> If you do not specify the -var-file, then the defaults in `vars.tf` will apply.

- WebLogic Non-JRF:

  ```
  $ terraform apply -var-file=inputs/instance.tfvars
  ```

- WebLogic JRF with OCI database:

  ```
  $ terraform apply -var-file=inputs/instance.tfvars -var-
  file=inputs/oci_db.tfvars
  ```

- WebLogic JRF with ATP database:

  ```
  $ terraform apply -var-file=inputs/instance.tfvars -var-
  file=inputs/atp_db.tfvars
  ```

5. Create multiple instances from same solutions:

   ```
   $ terraform apply -var-file=inputs/instance.tfvars -
   state=<file_name>
   ```

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

To update an infrastructure:

Complete the following steps:

1. Update the variables. For the list of variables you can update, see Table 1-1.
2. Complete step 1 through step 4 in Invoke terraform scripts.

To destroy an infrastructure:

```
$ terraform destroy -var-file=inputs/instance.tfvars
```

Delete the Resources and Stack

> **Tip:**
>
> You can invoke terraform scripts by using the Resource Manager. See Get Started with Oracle WebLogic Server for OKE and Managing Stacks and Jobs.

Variables in Terraform Scripts

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

> **Note:**
>
> If you are using Oracle WebLogic Server for OKE (**Release 21.3.3 or later**), see Terraform Scripts in Oracle WebLogic Server for OKE.

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 `env_vars_template` to create `env_vars` and source it as: source `./env_vars` before running terraform init.	-	-	-	-	-
`FingerPrint`	String	-	-	Yes	Fingerprint of the OCI API private 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.
`region`	String	-	-	-	Region for provisioning.

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

Variables	Type	Default Value	Optional	Can be updated?	Description
`service_name`	String	-	-	-	Prefix for stack resources. The names of all the related compute and network resources begins with the prefix you assign here.
`ssh_public_key`	String	-	-	-	Content of public key for access.
`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`	Yes	-	Name of the administrator in the WebLogic Server domain.
General	-	-	-	-	-
`create_policies`	Boolean	`true`	Yes	-	Create policies to read Secrets from Vault and manage ATP database (if applicable).
Container Cluster (OKE) Configuration	-	-	-	-	-
`existing_cluster_id`	String	-	-	-	Existing cluster ID value.
`kubernetes_version`	String	Latest Kubernetes version is displayed by default.	Yes	-	Kubernetes version.

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

Variables	Type	Default Value	Optional	Can be updated?	Description
non_wls_node_pool_count	String	1	-	Yes	Count of the non-WLS node pool. **Note:** If you update the node pool count, then the node pool is recreated. If you scale the compute instance and the WebLogic server Operator does not connect to the WebLogic server Persistence store, see Scaling Compute Instances.
non_wls_node_pool_shape	String	VM.Standard2.1	-	Yes	Shape of the non-WLS node pool.
pods_cidr	String	-	Yes	-	CIDR value of the OKE pod.
services_cidr	String	-	Yes	-	CIDR value of the services.
wls_node_pool_count	String	1	-	Yes	Count of the WLS node pool.
wls_node_pool_shape	String	VM.Standard2.1	-	Yes	Shape of the WLS node pool.
Container Cluster (OKE) Administration Instances	-	-	-	-	-
admin_availability_domain	String	-	-	-	Name of the availability domain for the administrator instance.
admin_shape	String	VM.Standard.E2.1	-	-	Shape for administrator instance.
bastion_shape	String	VM.Standard.E2.1	-	-	Shape for bastion instance.

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

Variables	Type	Default Value	Optional	Can be updated?	Description
Network Variables	-	-	-	-	-
`existing_vcn_id`	String	-	-	-	OCID of an existing VCN where you want to create the compute instances, network resources, and load balancers.
`existing_lb_subnet_id`	String	-	-	-	OCID of an existing load balancer subnets.
`existing_bastion_subnet_id`	String	-	-	-	OCID for an existing bastion subnet.
`existing_oke_workers_subnet_id`	String	-	-	-	OCID for an OKE worker node subnet.
`existing_admin_subnet_id`	String	-	-	-	OCID for an existing administrator subnet.
`existing_fss_subnet_id`	String	-	-	-	OCID for an existing FSS subnet.
`existing_nat_gw_id`	String	-	Yes	-	OCID for an existing NAT gateway. **Note:** You need to specify either the NAT gateway (`existing_nat_gw_id`) or service gateway (`existing_service_gw_id`).

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

Variables	Type	Default Value	Optional	Can be updated?	Description
`existing_service_gw_id`	String	-	Yes	-	OCID for an existing service gateway. **Note:** You need to specify either the NAT gateway (`existing_nat_gw_id`) or service gateway (`existing_service_gw_id`).
`is_bastion_instance_required`	Boolean	`true`	Yes	-	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.
Load Balancer Variables	-	-	-	-	-
`lb_shape`	String	`flexible`	-	-	Shape of the public load balancer.
`ingress_lb_shape`	String	`flexible`	-	-	Shape of the ingress load balancer.
Shared File System Variables	-	-	-	-	-
`fss_availability_domain`	String	-	-	-	OCID of the availability domain for Shared File System.
`mountTarget_id`	String	-	Yes	-	OCID for the mount target.

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

Variables	Type	Default Value	Optional	Can be updated?	Description
`mountTarget_compartment_id`	String	-	Yes	-	OCID of the compartment for the mount target. This variable is required if `mountTarget_id` is updated.
OCIR Variables		-	-		
`ocir_user`	String	-	-	-	OCIR user name.
`ocir_auth_token_ocid`	String	-	-	-	OCID token for the OCIR user name.
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`	Yes	-	IDCS cloud gate port value.
`idcs_host`	String	`identity.oraclecloud.com`	Yes	-	IDCS host value.
`idcs_port`	Number	`443`	Yes	-	IDCS port value.
`idcs_tenant`	String	-	-	-	IDCS tenant value.
`is_idcs_selected`	Boolean	`false`	Yes	-	If you want an IDCS has to be provisioned.
OCI Database Variables	-	-	-	-	-
`ocidb_compartment_id`	String	-	-	-	OCID of the OCI database compartment.
`ocidb_dbsystem_id`	String	-	-	-	OCID of the OCI database system.
`ocidb_database_id`	String	-	-	-	OCID of the OCI database.
`ocidb_pdb_service_name`	String	-	-	-	PDB name of the OCI database.

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

Variables	Type	Default Value	Optional	Can be updated?	Description
`oci_db_user`	String	`sys`	Yes	-	OCI database username
`oci_db_password`	String	-	-	-	OCID of the OCI database password.
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.

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 varaibles: `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 OKE.

Following is a sample script for file: **`env_vars:`**

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

### Authentication details
export TF_VAR_tenancy_ocid="<tenancy_ocid>"
export TF_VAR_user_ocid="<user_ocid>"
export TF_VAR_api_fingerprint="<fingerprint>"
export TF_VAR_api_private_key_path="/home/<path>/.oci/oci_api_key.pem"
```

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 System
ocidb_dbsystem_id ="<OCID>"

#Database
ocidb_database_id = "<OCID>"

#PDB Name
ocidb_pdb_service_name = "PDB1"

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

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_compartment_id="<OCID>"
atp_db_password_ocid="<password>"
```

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

```
# Copyright 2019, 2020, Oracle Corporation and/or affiliates.  All
rights reserved.

# Identity and access parameters

#Compartment for resources- MyCompartmentWLSC
compartment_ocid = "<Compartment OCID>"

#Network compartment -OCID HERE is for Networks compartment
network_compartment_id = "<Network Compartment OCID>"

region = "<region>"

# ssh keys
ssh_public_key = "<ssh public key>"

# general oci parameters
resource_prefix = "<prefix>"

# networking
vcn_cidr = "<CIDR>"
```

```
# admin
admin_shape = "VM.Standard2.1"

# which AD where to place non-OKE resources
admin_availability_domain=""

#depends on the subnet type
assign_admin_public_ip = false

# oke
cluster_name = "oke-cluster"
worker_mode = "private"

allow_node_port_access = false
allow_worker_ssh_access = false

dashboard_enabled = true
kubernetes_version = "v1.17.9"

pods_cidr = "<CIDR>"
services_cidr = "<CIDR>"

# ocir
ocir_region="phx"

ocir_user="<firstname.lastname@email.com>"
ocir_auth_token_ocid="<OCID>"


#weblogic parameters
wls_domain_name="myDomain"
wls_admin_user="weblogic"
wls_admin_password_ocid="<OCID>"
deploy_sample_app=true

#fss parameters- for existing-mount-ad-2
fss_availability_domain="<FSS Domain>"

#ingress
ingress_lb_shape="400Mbps"
ingress_enable_http_port=false

#workaround to provision all nodes in single AD
node_pool_single_ad="<AD>"

#If use_encryption flag is true, then the vault key is used for OKE
encryption
vault_key_ocid="<OCID>"

#Optional

#use existing network -oke-vcn
existing_vcn_id ="<OCID>"
existing_lb_subnet_id ="<OCID>"
```

```
existing_bastion_subnet_id ="<OCID>"
existing_oke_workers_subnet_id ="<OCID>"
existing_admin_subnet_id="<OCID>"
existing_fss_subnet_id="<OCID>"
existing_service_gw_id="<OCID>"

#existing-mount-ad-2- for WLSOnOke compartment
mountTarget_id="<OCID>"
mountTarget_compartment_id="<OCID>"

existing_cluster_id=""
# No proxy is configured for Jenkins if this is undefined or is empty
string
#jenkins_proxy="<proxy>"
```