



Resource Manager

Level 100

Flavio Pereira
February 2019

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Objectives

After completing this lesson, you should be able to:

- Describe the basic components of Resource Manager
- Describe the benefits of Resource Manager
- Prepare Terraform files for Resource Manager
- Resource Manager Demo

Introducing the Oracle Cloud Infrastructure Resource Manager

Manage your infrastructure resources using Terraform

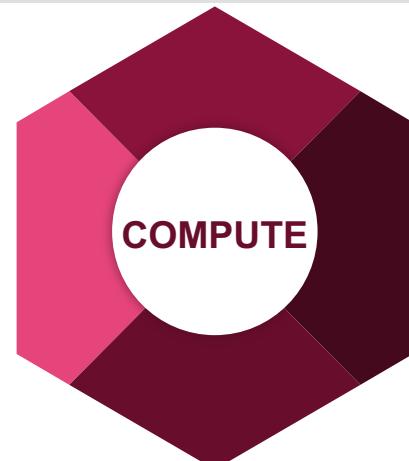


Developers and
DevOps



Architects and
IT Ops

Resource Manager



Resource Manager Benefits

- Automate and standardize your infrastructure and easily replicate environments
- Deep integration with OCI Platform and its services
- Seamlessly manages state files and improves team collaboration
- Fully managed service for the Terraform engine
- You only pay for the underlying compute, storage, network or any other resource you provision using Resource Manager.

Resource Manager Components

You can think of the Resource Manager as Terraform-as-a-Service for Oracle Cloud Infrastructure resources. Once you have your terraform files ready and your variables file adjusted for Resource Manager you can start building **Stacks** and executing **Jobs**:

Stack - Represents a set of OCI resources you want to create in a compartment. Each Stack has a *configuration*, which is a set of Terraform files that specify the resources you want to manage together using the Resource Manager.

Job - Represents a request to take a Terraform Action on a Stack. Resource Manager provides three job-types: **plan, apply and destroy**

Resource Manager - Access Control & Permissions

- To use the Resource Manager, you must have the required OCI Identity and Access Management (IAM) permissions.
- Following are example policy statements that grant a group called ADMIN-XYZ the ability to manage all the Resource Manager resources in the tenancy:

Allow group ADMIN-XYZ to manage orm-stacks in tenancy
Allow group ADMIN-XYZ to manage orm-jobs in tenancy

- DEV-XYZ is a developer group is able to read Stacks and to execute Jobs - except for destroy – only within in a specific compartment:

Allow group DEV-XYZ to use orm-stacks in compartment XYZ
Allow group DEV-XYZ to use orm-jobs in compartment XYZ where target.job.operation != 'DESTROY'

Prepare your Terraform files to work with Resource Manager

- With the Resource Manager all that is required is an OCI Identity and Access Management (IAM) permissions.
- You can omit the user OCID, private key, fingerprint, and tenancy OCID from provider configuration.

Open Source OCI Terraform Provider

```
variable "user_ocid" {}
variable "fingerprint" {}
variable "private_key_path" {}
variable "private_key_password" {}

provider "oci" {
    tenancy_ocid      = "${var.tenancy_ocid}"
    user_ocid         = "${var.user_ocid}"
    fingerprint       = "${var.fingerprint}"
    private_key_path  = "${var.private_key_path}"
    private_key_password = "${var.private_key_password}"
    region            = "${var.region}"
}
```

Resource Manager

```
variable "region" {}
variable "compartment_ocid" {}

provider "oci" {
    region = "${var.region}"
}
```

Resource Manager Variables

Resource Manager give you a possibility to enter extra variables to help with your deployment. Here is an example of adding a public ssh key:

Terraform Resource Manager file

```
variable "region" {}
variable "compartment_ocid" {}
variable "ssh_public_key" {}
```

Variables

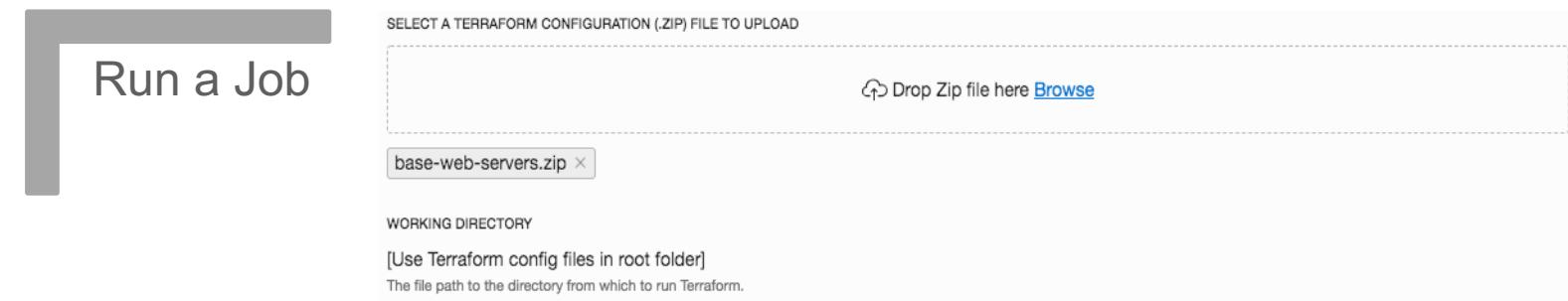
Terraform variables for this stack.

KEY	VALUE
region	us-phoenix-1
compartment_ocid	ocid1.compartment.oc1..aaaaaaaaaaaaaaaexqasvooxzeqt3prj4axpraya uedvpitfnkskrz7kakfxn2qouq
ssh_public_key	ssh-rsa

+ Additional Variable

Resource Manager Workflow: Step 1

Define configuration



- Create the Terraform configuration for resources
- Write optional Terraform modules
- Create a zip file containing the Terraform files
- Avoid supplying confidential information in your configuration like passwords and SSH keys.

Resource Manager Workflow: Step 2

Create a Stack



Define
configuration

Create a
Stack

Run a Job

- Stack represents a set of resources you manage within a compartment
- Each Stack maps to Terraform configuration files and a Terraform state file

Create Stack

[help](#) [cancel](#)

CREATE IN COMPARTMENT

Demo
bmo-flaviop (root)/Demo

NAME OPTIONAL

Web-Servers-Deployment

DESCRIPTION OPTIONAL

This stack creates 3 Web-Servers on different ADs

SELECT A TERRAFORM CONFIGURATION (.ZIP) FILE TO UPLOAD

Drop Zip file here [Browse](#)

base-web-servers.zip

WORKING DIRECTORY

[Use Terraform config files in root folder]
The file path to the directory from which to run Terraform.

Variables

Terraform variables for this stack.

KEY	VALUE
region	us-phoenix-1
compartment_ocid	ocid1.compartment.oc1..aaaaaaaaafc3xqsvobxzegt3brj4axpfyduedvijffnskrz7kakfxn2qouq
ssh_public_key	ssh-rsa

+ Additional Variable

TAGS OPTIONAL

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

[Learn more about tagging](#)

TAG NAMESPACE

No namespace (Free-Form tag)

TAG KEY

VALUE

+ Additional Tag

Create

Resource Manager Workflow: Step 3

Run a Terraform Job



Define
configuration

Create a
Stack

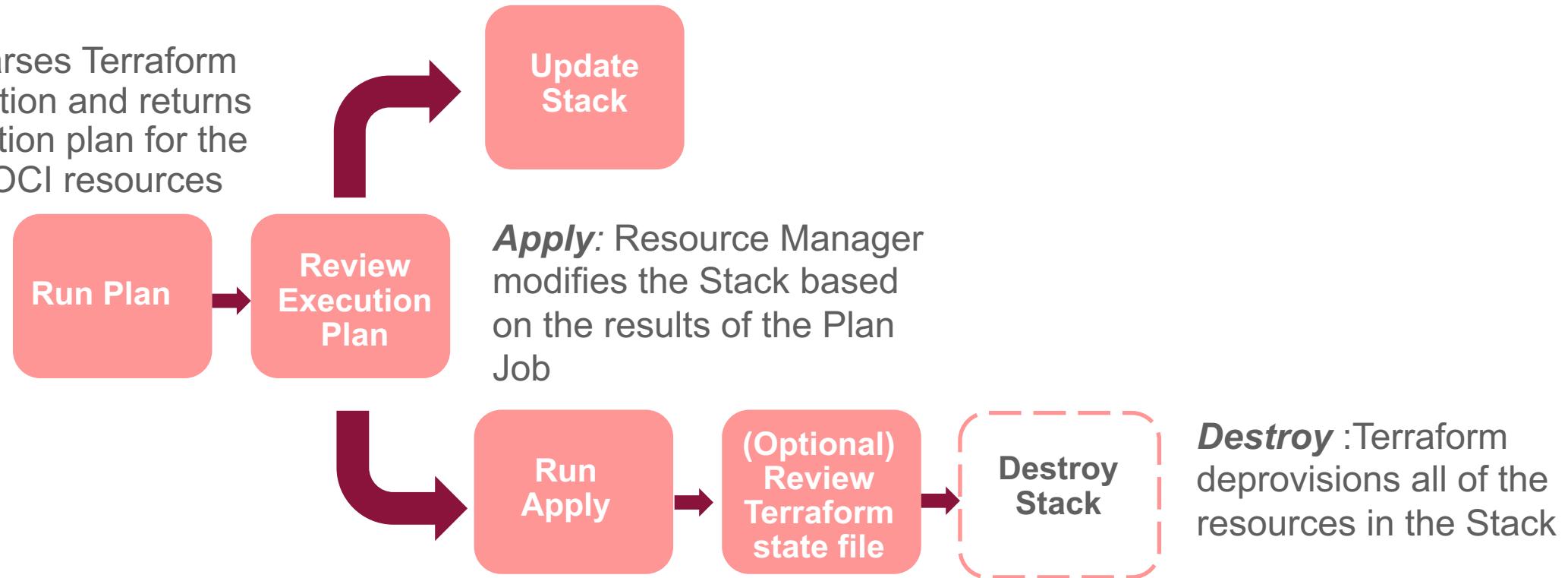
Run a
Job

- A Job is a Terraform Action executed against a Stack
- Job actions include Plan, Apply, and Destroy

The screenshot shows the Oracle Cloud Resource Manager interface. At the top, there's a navigation bar with the Oracle Cloud logo and a menu icon. Below it, the URL 'Resource Manager > Stacks > Stack Details' is visible. The main content area displays a stack named 'Web-Servers-Deployment'. On the right side of the stack details, there are buttons for 'Edit Stack', 'Terraform Actions' (with sub-options 'Plan', 'Apply', and 'Destroy'), 'Delete Stack', and 'Add Tag(s)'. Below these buttons, the stack's OCID and creation date ('Created: Fri, 08 Feb 2019 22:36:46 GMT') are shown. To the left of the stack details, there's a large green button with a white 'S' on it. Further down, under the heading 'Jobs', there's a table with two columns: 'Name' and 'Type'. The table currently has three empty rows.

Resource Manager Execution

Plan : Parses Terraform configuration and returns an execution plan for the effected OCI resources



Resource Manager Demo

Summary

- Build on Open Source Software, Resource Manager is fully-managed service that makes easier to use Terraform on Oracle Cloud Infrastructure
- You can leverage your existing Terraform templates to deploy with Resource Manager
- There are no charges for using the Oracle Cloud Infrastructure Resource Manager.
- You can try Resource Manager, by sign up for a free trial OCI account here: <https://cloud.oracle.com/tryit> and follow the steps on this guide: <http://bit.ly/hol-orm>
- Resource Manager Documentation:
<https://docs.cloud.oracle.com/iaas/Content/ResourceManager/Concepts/resourcemanager.htm>

ORACLE®
Cloud Infrastructure

cloud.oracle.com/iaas

cloud.oracle.com/tryit

Resource Manager Demo: Step 1

Go to **Menu → Resource Manager → Stack** and click **Create Stack**

The screenshot shows the Oracle Cloud Resource Manager interface. The top navigation bar includes the Oracle Cloud logo, a search bar, and account information for 'us-phoenix-1'. The left sidebar has a 'Resource Manager' section with 'Stacks' selected, and other options like 'Jobs' and 'List Scope'. Below that is a 'COMPARTMENT' dropdown set to 'Demo', with a note 'bmc-flaviop (root)/Demo'. The main content area is titled 'Stacks *in Demo Compartment*'. It features a 'Create Stack' button and a table with columns 'Name', 'Description', 'State', and 'Created'. A message 'No items' is displayed below the table. In the bottom right corner of the main area, it says 'Showing 0 Stacks < Page 1 >'.

Resource Manager Demo: Step 2

- Select the target compartment
- Enter a Name and Description
- Upload your Terraform zip file
- Enter the variables
- Click Create

You will see your Stack created showing an **Active State**

Resource Manager

Stacks

Jobs

List Scope

COMPARTMENT

Demo

Stacks in Demo Compartment

Create Stack

Name	Description
Web-Servers-Deployment	This will deploy 3 web servers.

Create

Create Stack

CREATE IN COMPARTMENT

Demo
bmc-flavio (root)/Demo

NAME OPTIONAL

Web-Servers-Deployment

DESCRIPTION OPTIONAL

This will deploy 3 web servers.

SELECT A TERRAFORM CONFIGURATION (.ZIP) FILE TO UPLOAD

Drop Zip file here [Browse](#)

base-web-servers.zip

WORKING DIRECTORY

[Use Terraform config files in root folder]
The file path to the directory from which to run Terraform.

Variables

Terraform variables for this stack.

KEY	VALUE
region	us-phoenix
compartment_ocid	ocid1.compartment.oc1..aaaaaaaaafc3xqsvobxzegt3brj4axpfidyduedvpjffnskrz7kakfxn2qouq
ssh_public_key	anyconnect-10-154-138-76.vpn.oracle.com
ssh_private_key	uaSTH6tMSOahfREHWUm98k4CgxK1SQIAN52WZHGoxXFQG3mEq8AZg==

+ Additional Variable

TAGS OPTIONAL

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

[Learn more about tagging](#)

TAG NAMESPACE

No namespace (Free-Form tag)

TAG KEY

VALUE

+ Additional Tag

Create

Resource Manager Demo: Step 3

- Select the Stack you created
- Select "Plan" on the Terraform Actions menu
- Give it a name and click the "Plan" button

Plan

NAME OPTIONAL

TAGS OPTIONAL

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

[Learn more about tagging](#)

TAG NAMESPACE	TAG KEY	VALUE
No namespace (Free-Form tag)		

+ Additional Tag

Plan

Resource Manager » Stacks » Stack Details

Web-Servers-Deployment

[Edit Stack](#) [Terraform Actions ▾](#) [Delete Stack](#) [Add Tag\(s\)](#)

Stack Information

Plan

Apply

Description: [Destroy](#)

OCID: ...so2hfa [Show](#) [Copy](#)

Created: Fri, 08 Feb 2019 22:51:25 GMT

Resource Manager Demo: Step 4

- Select "Apply" on the Terraform Actions menu
- Give it a name and click the “Apply” button

Resource Manager » Stacks » Stack Details



Web-Servers-Deployment

Edit Stack Terraform Actions ▾ Delete Stack Add Tag(s)

Stack Inform
Plan
Apply

Description: **Destroy**

OCID: ...cczqvq Show Copy

Created: Fri, 08 Feb 2019 23:03:05 GMT

Apply [help](#) [cancel](#)

NAME OPTIONAL

TAGS OPTIONAL

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

[Learn more about tagging](#)

TAG NAMESPACE	TAG KEY	VALUE
No namespace (Free-Form tag) ▾		

+ Additional Tag

Apply

Resources

Jobs

Jobs

Variables

Name	Type	State	Start Time ▾	End Time	⋮
Web-Servers-Plan	Plan	● Succeeded	Fri, 08 Feb 2019 23:03:17 GMT	Fri, 08 Feb 2019 23:07:02 GMT	Showing 1 Jobs < Page 1 >

Resource Manager Demo: Step 5

Under Jobs you can see the job history and the state of the actions.

Resource Manager » Stacks » Stack Details

Web-Servers-Deployment

[Edit Stack](#) [Terraform Actions](#) [Delete Stack](#) [Add Tag\(s\)](#)

[Stack Information](#) [Tags](#)

Description: This stack will deploy 3 Web servers.

Compartment: bmc-flaviop (root)/Demo

OCID: ...cczqvq [Show](#) [Copy](#)

Terraform Config (.zip): Uploaded [Upload](#) [New](#) [Download](#)

Created: Fri, 08 Feb 2019 23:03:05 GMT

Resources

Jobs

Name	Type	State	Start Time	End Time
Web-Servers-Apply	Apply	Accepted	Fri, 08 Feb 2019 23:09:46 GMT	-
Web-Servers-Plan	Plan	Succeeded	Fri, 08 Feb 2019 23:03:17 GMT	Fri, 08 Feb 2019 23:07:02 GMT

Showing 2 Jobs [Page 1](#)

Resource Manager Demo: Step 6

Navigate to **Compute > Instances** and see the created Web-Servers

Create Instance					
Sort by:		Created Date (Desc)		Displaying 3 Instances < Page 1 >	
	tf-server-2	Shape: VM.Standard2.1	Region: phx Availability Domain: Akfl:PHX-AD-2 Fault Domain: FAULT-DOMAIN-1	Created: Fri, 08 Feb 2019 23:13:14 GMT Maintenance Reboot: -	•••
	tf-server-3	Shape: VM.Standard2.1	Region: phx Availability Domain: Akfl:PHX-AD-3 Fault Domain: FAULT-DOMAIN-3	Created: Fri, 08 Feb 2019 23:13:14 GMT Maintenance Reboot: -	•••
	tf-server-1	Shape: VM.Standard2.1	Region: phx Availability Domain: Akfl:PHX-AD-1 Fault Domain: FAULT-DOMAIN-2	Created: Fri, 08 Feb 2019 23:13:14 GMT Maintenance Reboot: -	•••
Displaying 3 Instances < Page 1 >					

Resource Manager Demo: Step 7

For each Job, download the Logs, Terraform configuration and Terraform State

Resource Manager » Stacks » Web-Servers-Deployment » Job Details

Web-Servers-Plan

[Download Terraform Config](#) [Add Tag\(s\)](#)

[Job Information](#) [Tags](#)

OCID: ...6a6fsq [Show](#) [Copy](#)
Job Type: Plan
Working Directory: [Not Specified]
End Time: Fri, 08 Feb 2019 23:07:02 GMT

Resource Manager » Stacks » Web-Servers-Deployment » Job Details

Web-Servers-Apply

[Download Terraform Config](#) [Download Terraform State](#) [Add Tag\(s\)](#)

[Job Information](#) [Tags](#)

OCID: ...wveysa [Show](#) [Copy](#)
Job Type: Apply
Working Directory: [Not Specified]
End Time: Fri, 08 Feb 2019 23:15:36 GMT

Compartment: bmc-flavio (root)/Demo
State: Succeeded
Start Time: Fri, 08 Feb 2019 23:09:46 GMT

Resources

Logs

[Download Logs](#)

```
2019-02-08T23:07:00.116Z [INFO]
2019-02-08T23:07:00.121Z [INFO] Initializing provider plugins...
2019-02-08T23:07:00.121Z [INFO]
2019-02-08T23:07:00.121Z [INFO] The following providers do not have any version constraints in configuration,
2019-02-08T23:07:00.121Z [INFO] so the latest version was installed.
2019-02-08T23:07:00.121Z [INFO]
2019-02-08T23:07:00.121Z [INFO] To prevent automatic upgrades to new major versions that may contain breaking
2019-02-08T23:07:00.121Z [INFO] changes, it is recommended to add version = "... " constraints to the
2019-02-08T23:07:00.121Z [INFO] corresponding provider blocks in configuration, with the constraint strings
2019-02-08T23:07:00.121Z [INFO] suggested below.
2019-02-08T23:07:00.121Z [INFO]
2019-02-08T23:07:00.121Z [INFO] * provider.oci: version = "~> 3.13"
2019-02-08T23:07:00.121Z [INFO] Terraform has been successfully initialized!
2019-02-08T23:07:00.121Z [INFO]
2019-02-08T23:07:00.121Z [INFO] You may now begin working with Terraform. Try running "terraform plan" to see
2019-02-08T23:07:00.121Z [INFO] any changes that are required for your infrastructure. All Terraform commands
```

Resources

Logs

[Download Logs](#)

```
2019-02-08T23:13:09.908Z [INFO]
2019-02-08T23:13:09.913Z [INFO] Initializing provider plugins...
2019-02-08T23:13:09.913Z [INFO]
2019-02-08T23:13:09.913Z [INFO] The following providers do not have any version constraints in configuration,
2019-02-08T23:13:09.913Z [INFO] so the latest version was installed.
2019-02-08T23:13:09.913Z [INFO]
2019-02-08T23:13:09.913Z [INFO] To prevent automatic upgrades to new major versions that may contain breaking
2019-02-08T23:13:09.913Z [INFO] changes, it is recommended to add version = "... " constraints to the
2019-02-08T23:13:09.913Z [INFO] corresponding provider blocks in configuration, with the constraint strings
2019-02-08T23:13:09.913Z [INFO] suggested below.
2019-02-08T23:13:09.913Z [INFO]
2019-02-08T23:13:09.913Z [INFO] * provider.oci: version = "> 3.13"
2019-02-08T23:13:09.913Z [INFO] Terraform has been successfully initialized!
2019-02-08T23:13:09.913Z [INFO]
2019-02-08T23:13:09.913Z [INFO] You may now begin working with Terraform. Try running "terraform plan" to see
2019-02-08T23:13:09.913Z [INFO] any changes that are required for your infrastructure. All Terraform commands
```