

# Oracle Cloud Infrastructure Guide

## Introduction

Oracle provides a number of Ansible modules to interact with Oracle Cloud Infrastructure (OCI). In this guide, we will explain how you can use these modules to orchestrate, provision and configure your infrastructure on OCI.

## Requirements

To use the OCI Ansible modules, you must have the following prerequisites on your control node, the computer from which Ansible playbooks are executed.

1. [An Oracle Cloud Infrastructure account](#).
2. A user created in that account, in a security group with a policy that grants the necessary permissions for working with resources in those compartments. For guidance, see [How Policies Work](#).
3. The necessary credentials and OCID information.

## Installation

1. Install the Oracle Cloud Infrastructure Python SDK ([detailed installation instructions](#)):

```
pip install oci
```

2. Install the Ansible OCI Modules in one of two ways:

- a. From Galaxy:

```
ansible-galaxy install oracle.oci_ansible_modules
```

- b. From GitHub:

```
$ git clone https://github.com/oracle/oci-ansible-modules.git
```

```
$ cd oci-ansible-modules
```

Run one of the following commands:

- If Ansible is installed only for your user:

```
$ ./install.py
```

- If Ansible is installed as root:

```
$ sudo ./install.py
```

## Configuration

When creating and configuring Oracle Cloud Infrastructure resources, Ansible modules use the authentication information outlined [here](#).

## Examples

### Launch a compute instance

This [sample launch playbook](#) launches a public Compute instance and then accesses the instance from an Ansible module over an SSH connection. The sample illustrates how to:

- Generate a temporary, host-specific SSH key pair.
- Specify the public key from the key pair for connecting to the instance, and then launch the instance.
- Connect to the newly launched instance using SSH.

### Create and manage Autonomous Data Warehouses

This [sample warehouse playbook](#) creates an Autonomous Data Warehouse and manage its lifecycle. The sample shows how to:

- Set up an Autonomous Data Warehouse.
- List all of the Autonomous Data Warehouse instances available in a compartment, filtered by the display name.
- Get the "facts" for a specified Autonomous Data Warehouse.
- Stop and start an Autonomous Data Warehouse instance.

- Delete an Autonomous Data Warehouse instance.

## Create and manage Autonomous Transaction Processing

This [sample playbook](#) creates an Autonomous Transaction Processing database and manage its lifecycle. The sample shows how to:

- Set up an Autonomous Transaction Processing database instance.
- List all of the Autonomous Transaction Processing instances in a compartment, filtered by the display name.
- Get the "facts" for a specified Autonomous Transaction Processing instance.
- Delete an Autonomous Transaction Processing database instance.

You can find more examples here: [Sample Ansible Playbooks](#).