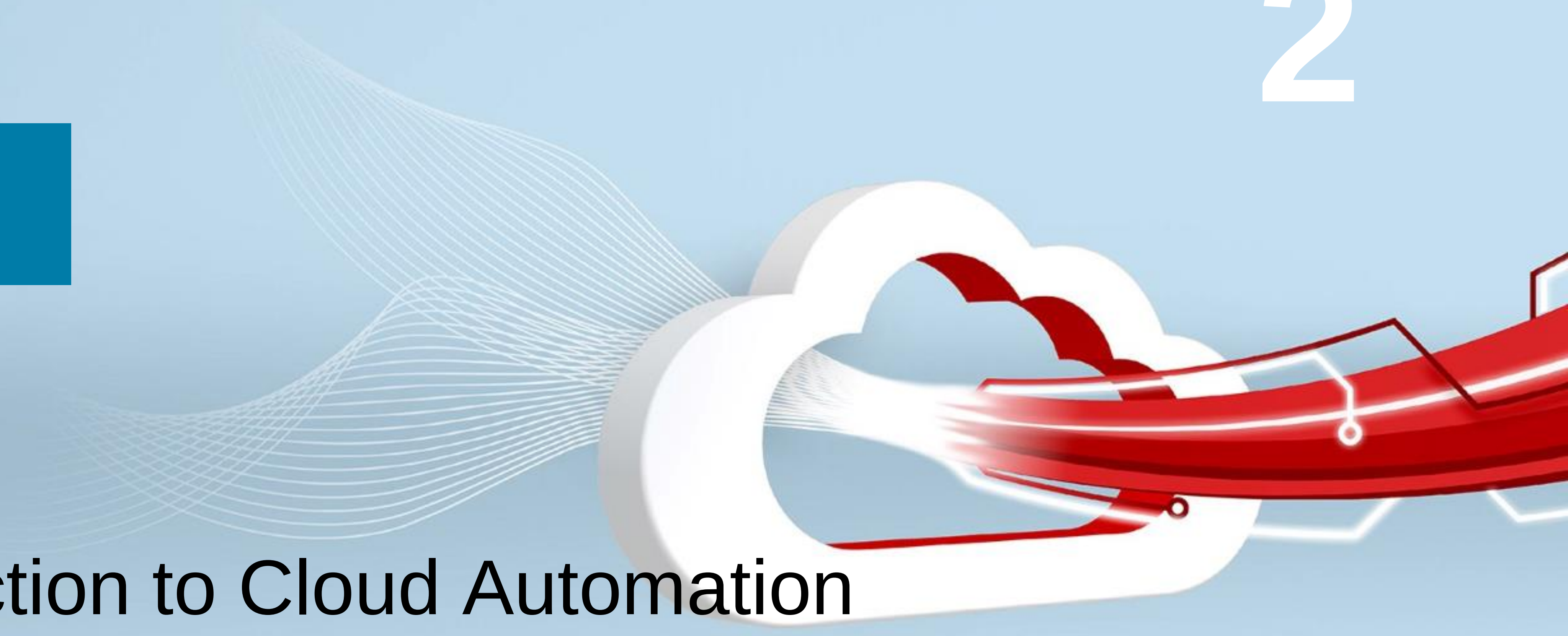


Introduction to Cloud Automation

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Module Objectives

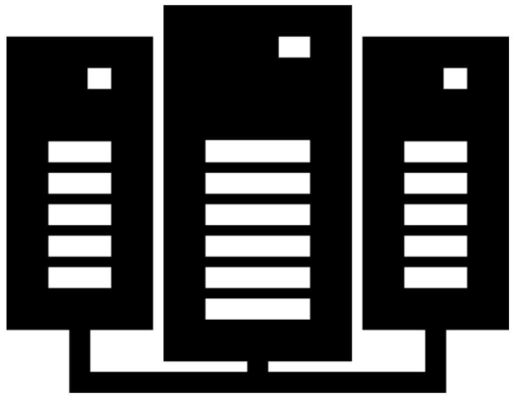
- Define **Cloud Scale** and understand operating challenges
- Understand the basics of automation
- Introduction to the core automation tools for OCI
- Compare and contrast the different automation tools

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

What is **Cloud Scale**?

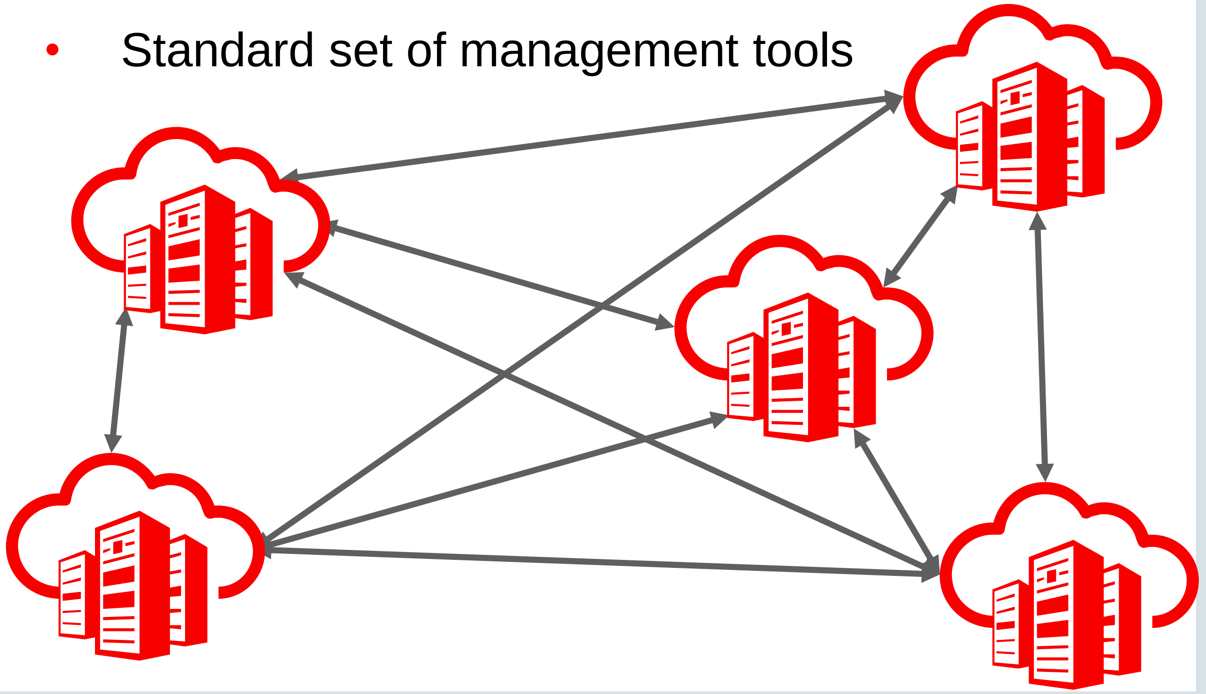
On-Premises Data Center

- Fixed amount of capacity
- Often limited to a single, physical location
- Non-standard set of management tools



Oracle Cloud Infrastructure

- Massive amounts of capacity
- 16 locations globally, and growing
- Single, unified API for management
- Standard set of management tools



ORACLE

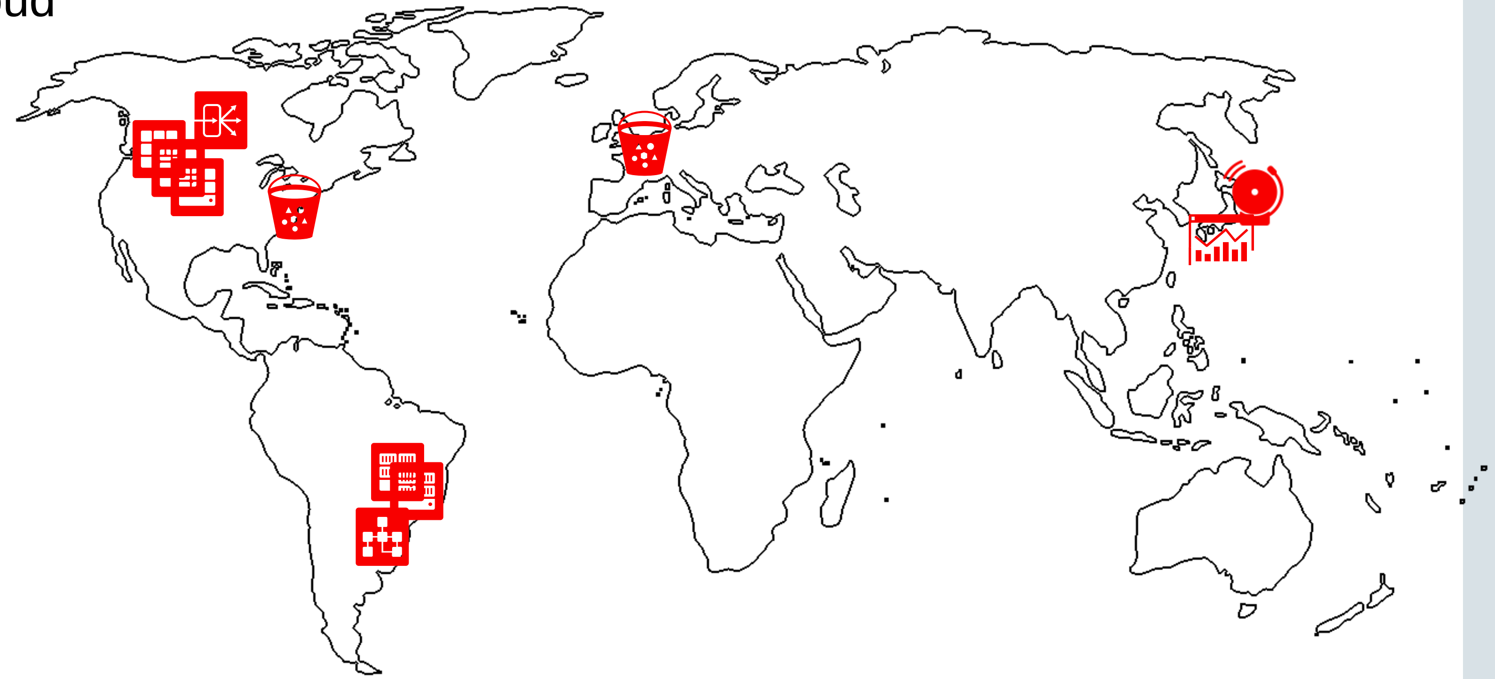
Common Terminology

- **Idempotent:** this means a change or other action is not applied more than once. Some tools validate whether a task has been completed before applying and will avoid duplicating efforts. This saves cycles and limits potential impact to running resources.
- **Immutable:** is a common term referring to a type of infrastructure or service. It means you don't ever make changes to it. When it comes time to troubleshoot or upgrade, just replace the resource.
- **Ephemeral:** a term used to refer to impermanent resources or temporary resource assignments.
- **Stateless (Application)**: The notion that an application is constructed in such a way as to avoid reliance on any single component to manage transactional or session-related information. Often times a stateless application may leverage immutable instances as part of the deployment strategy.
- **Infrastructure as Code (IaC)**: The process of managing and provisioning cloud resources and services through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.

ORACLE
Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Cloud Scale Challenges

- The scope and number of resources under your management may increase significantly
- You pay for what you use…so if you're not using it, stop paying for it
- You may begin to manage new technologies
- Things may move much faster in the cloud



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Knowledge Check

You have recently joined the Site Reliability team at a new company and find out their most heavily used application utilizes immutable instances within the Web tier. During a cursory review of the monitoring dashboard you notice 1 of the 11 instances appears to be unhealthy. What is the first action you would perform in an attempt to resolve this issue?

A. Connect to the server via SSH and view the contents of the file */var/log/messages*
B. Stop the instance, wait 5 minutes, then start the instance back up
C. Terminate the instance and replace it using the appropriate image
D. Resign immediately; this job isn’t going to be any fun

ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Knowledge Check

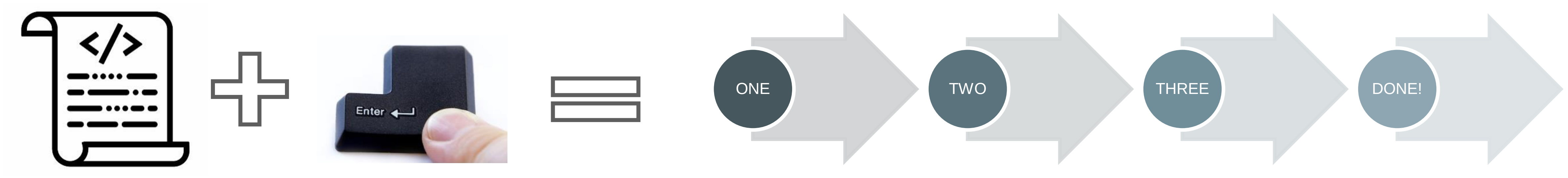
You have recently joined the Site Reliability team at a new company and find out their most heavily used application utilizes immutable instances within the Web tier. During a cursory review of the monitoring dashboard you notice 1 of the 11 instances appears to be unhealthy. What is the first action you would perform in an attempt to resolve this issue?

A. Connect to the server via SSH and view the contents of the file */var/log/messages*
B. Stop the instance, wait 5 minutes, then start the instance back up
C. Terminate the instance and replace it using the appropriate image
D. Resign immediately; this job isn't going to be any fun

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Automation Basics

- Look closely and you might find, there is a lot of manual effort you can leave behind.
  - Administration: Creating / deploying new resources
  - Troubleshooting: changing configuration, restarting services, gathering log files
  - Cleanup: destroy an entire sandbox environment with a single click



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Oracle Cloud Infrastructure – Automation Tools

APIs
REST endpoints

SDKs
Java, Python, Ruby, Go, CLI

Terraform
Infrastructure as code

Ansible
Deployment Playbooks

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Automation Tools – Basic Capabilities

	APIs / SDKs	Terraform / Ansible
Programming	Yes*	No
Provisioning	Yes	Yes
Monitoring	Yes	No
Actions	Yes	No
Multi-Cloud	No	Yes

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Software Development Kits

The OCI Software Development Kits enable you to programmatically interact with your Oracle Cloud Infrastructure. Consider the following scenarios:

- You have built an application that accepts customer uploads and stores their files durably in object storage. Using the SDK you could design your application to interact directly with Object Storage Service.
- You are leveraging an open source monitoring tool and would like to extend its functionality. You could write custom code to evaluate or even alter running resources.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Example - Python

The following example illustrates how to instantiate the OCI SDK in Python and authenticate with an IAM Instance Principal.

```
import oci
import os
compartment_id = ‘ocid…’
signer = oci.auth.signers.InstancePrincipalsSecurityTokenSigner()

object_storage_client = oci.object_storage.ObjectStorageClient(config={}, signer=signer)
print(object_storage_client.get_namespace().data)
print(object_storage_client.list_buckets(namespace_name=object_storage_client
     .get_namespace().data, compartment_id=compartment_id).data)
```

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Example - Java

```java
import com.oracle.bmc.Region;
import com.oracle.bmc.auth.AuthenticationDetailsProvider;
import com.oracle.bmc.auth.ConfigFileAuthenticationDetailsProvider;
import com.oracle.bmc.objectstorage.ObjectStorage;
import com.oracle.bmc.objectstorage.ObjectStoarageClient;
import com.oracle.bmc.objectstorage.model.GetBucketRequest;

public class ObjectStorageGetBucketExample {
  public static void main(String[] args) throws Exception {
    String configurationFilePath = “~/.oci/config”;
    String profile = “DEFAULT”;

    AuthenticationDetailsProvider provider =
          new ConfigFileAuthenticationDetailsProvider(configurationFilePath, profile);

    ObjectStorage client = new ObjectStorageClient(provider);
    client.setRegion(Region.US_PHOENIX_1);
```

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

OCI Command Line Interface (CLI)

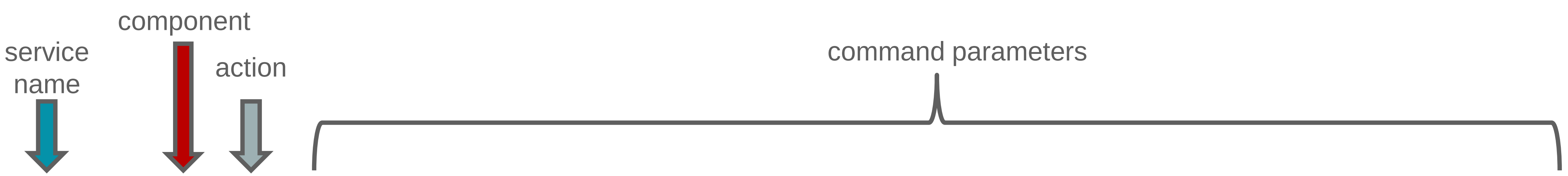
The CLI is an essential tool for managing your OCI resources. It provides much of the same functionality found in the console and sometimes offers extended capabilities. When combined with PowerShell or Bash scripts it can provide power automation capabilities.

- Built with the Python SDK
- Compatible with Python 2.7.5+ or 3.5+
- Works on Mac, Windows, and Linux
- Direct OCI API interactions

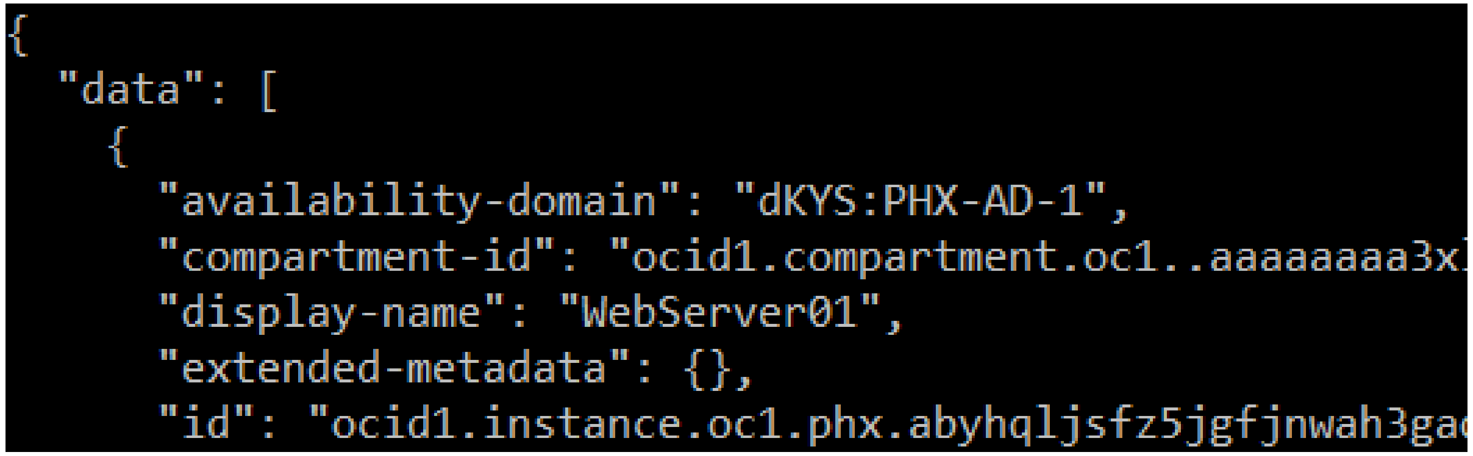
Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

CLI Command Construct

The OCI CLI is a unified tool that allows interaction with most service through a single command. After entering the program command specify the service, the action, and any additional switches.



```
$ oci compute instance list --region us-phoenix-1 --availability-domain gKOA:PHX-AD-1    --limit 2 --sort-by TIMECREATED
```



```
{
  "data": [
    {
      "availability-domain": "dKYS:PHX-AD-1",
      "compartment-id": "ocid1.compartment.oc1..aaaaaaaa3x
      "display-name": "WebServer01",
      "extended-metadata": {},
      "id": "ocid1.instance.oc1.phx.abyhqljsfz5jgfjnwah3ga
```

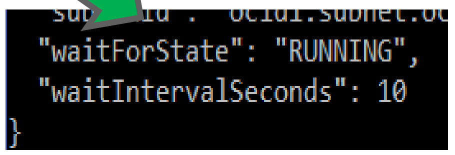
Sample: Bash script using OCI CLI

The following is an example of a simple BASH script. In the script we launch a new instance, wait for it to respond to an SSH request, then test the sample website.

```
#!/bin/bash
instance_id=$(oci compute instance launch --from-json file://compute_template.json \
  --query ‘data.id’ --raw-output)
pub_ip=$(oci compute instance list-vnics --instance-id $instance_id --query \
  ‘data[*]|[0].”public-ip”’ --raw-output)

# verify SSH connectivity
ssh -qi ~/.ssh/id_rsa opc@$pub_ip
while [ $? -ne 0 ]; do
  echo "Checking SSH connectivity“ && sleep 10
  ssh -qi ~/.ssh/id_rsa opc@$pub_ip
done
echo "SSH is up - lets move on!“ && sleep 3

#run a simple test
curl http://$pub_ip/testpage.html
```



```
  su[illegible]id": "ocid1.subnet.oc
  "waitForState": "RUNNING",
  "waitIntervalSeconds": 10
}
```

Infrastructure as Code – Introduction to Terraform

Terraform is an open-source tool for managing Infrastructure as Code. You can think of it as a platform interpreter that reads declarative text and converts it into API calls.

- Uses **Providers** to interpret declarations for more than 70 platforms.
- Manages resource lifecycle including dependencies, ensuring resources are created and deleted in the proper sequence.
- Can be used to **export** an existing environment that was created manually.
- Often coupled with a configuration management tool like Chef or Puppet

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Terraform provider for OCI

The OCI Provider is clearly documented with innumerable samples available. There is also a set of OCI Modules for Terraform which can be found in the Registry.

```
# Configure the Oracle Cloud Infrastructure provider with an API Key
provider "oci" {
  tenancy_ocid = "${var.tenancy_ocid}"
  user_ocid = "${var.user_ocid}"
  fingerprint = "${var.fingerprint}"
  private_key_path = "${var.private_key_path}"
  region = "${var.region}"
}

# Get a list of Availability Domains
data "oci_identity_availability_domains" "ads" {
  compartment_id = "${var.tenancy_ocid}"
}

# Output the result
output "show-ads" {
  value = "${data.oci_identity_availability_domains.ads.availability_domains}"
}
```

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Ansible is…

- Simple
  - Human readable automation
  - No special coding skills needed
  - Tasks are executed in Order
- Powerful
  - App and infrastructure deployment
  - Configuration Management
- Agentless
  - Uses OpenSSH and WinRM
  - No agents to exploit or update

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Ansible Overview

ANSIBLE

For Whom

- System Administrators
- DevOps Engineers

What It Is

- Ansible
  - Automation/DevOps tool
  - Infrastructure as Code
- Ansible Modules
  - Discrete units of code
  - A large library of out-of-box modules available from Ansible, Cloud providers and developer community

What You Can Do

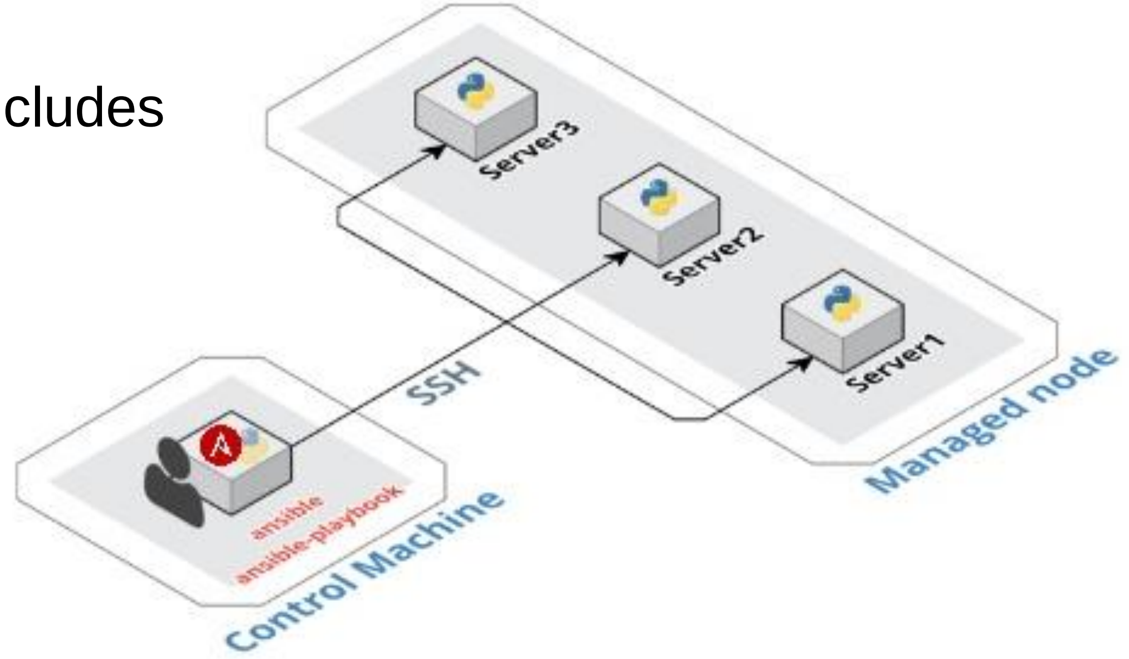
- Configuration Management
- Infrastructure and Resource Provisioning and Management
- One-off batch task executions across many hosts
- Automating complex orchestration flows
- Application Deployment, Release management, Audit
- Workflow automation, etc.

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

How Ansible Works

- Can be used to execute a variety of ad-hock commands initiated from a control point
- Utilizes small modules called “Playbooks” to perform command execution via remote SSH
- Utilizes SSH keypairs for authentication (Kerberos is supported)
- Inventory managed in a simple text file; also includes plugins to read from additional sources



ORACLE
Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Sample: Ansible Playbook

This playbook will install and start Apache on all hosts labeled **webservers** in the Ansible inventory file.

```
---
- name: Install and start Apache
  hosts: webservers

  tasks:
    - name: Install httpd
      yum: name=httpd state=present

    - name: Start httpd
      service: name=httpd state=running
```

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Comparing the Top Automation Tools

	CLI	Chef	Ansible	Terraform
Type	Task Automation	Config Mgmt	Config Mgmt+	Orchestration
Infrastructure	Mutable	Mutable	Mutable	Immutable
Language	Procedural	Procedural	Procedural	Declarative
Architecture	Client only	Client/Server	Client only	Client only

WHEN:		USE:
Automating simple, repeatable actions	$\longrightarrow$	OCI CLI
Managing app deployment and configuration	$\longrightarrow$	Ansible / Chef
Creating / destroying complex application architectures	$\longrightarrow$	Terraform

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Knowledge Check

Your team has recently deployed a custom Java application to a collection of 10 OCI Compute instances. The application is only used 10 hours each day. To save money, you devise a plan to stop the instances at the end of the day when they are no longer needed, and start them each morning shortly before they will be used.

Which tool would provide the easiest method to help implement this plan?

A. Write a custom application using the Java SDK.
B. Use the OCI Command Line Interface.
C. Hire an intern to start and stop resources in the OCI Management Console each day.
D. Use Terraform.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Knowledge Check

Your team has recently deployed a custom Java application to a collection of 10 OCI Compute instances. The application is only used 10 hours each day. To save money, you devise a plan to stop the instances at the end of the day when they are no longer needed, and start them each morning shortly before they will be used.

Which tool would provide the easiest method to help implement this plan?

A. Write a custom application using the Java SDK.

B. Use the OCI Command Line Interface.

C. Hire an intern to start and stop resources in the OCI Management Console each day.

D. Use Terraform.

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Knowledge Check

Your team has recently deployed a custom Java application to a collection of 10 OCI Compute instances. The application is only used 10 hours each day. To save money, you devise a plan to stop the instances at the end of the day when they are no longer needed, and start them each morning shortly before they will be used.

Which tool would provide the easiest method to help implement this plan?

A. Write a custom application using the Java SDK.
B. Use the OCI Command Line Interface.
C. Hire an intern to start and stop resources in the OCI Management Console each day.
D. Use Terraform.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Summary: Automation Tools

- Got an overview of what it means to operate at cloud scale
- Discussed the importance of automation
- Reviewed some of the common automation tools and their use cases

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Resources

[OCI CLI Documentation]

[Terraform Provider for OCI]

[OCI Registry for Terraform modules]

[Getting started with Ansible]

[Ansible modules for OCI]

**Click on the link in Slide Show mode to go directly to the assets page, portal or mail.*

ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.