Caltech | Center for Technology & Management Education

Post Graduate Program in DevOps

Powered by simplilearn

Caltech | Center for Technology & Management Education

Configuration Management with Ansible and Terraform

Powerd by simplilearn

Ansible Ad Hoc Commands

A Day in the Life of a DevOps Engineer

You are working in an organization as a DevOps consultant which uses Ansible for configuration management. Some new employees recently joined your team and would require a walkthrough on the ad hoc commands to be used in Ansible.
The goal is to use Ad hoc commands to perform a quick task as they perform well for rarely repeated tasks.

The team is mainly considering the below scenarios:

- Validating the uptime of 1 to 200 remote servers
- Collecting the disk space available on remote hosts
- Validating if the server is alive or not
- Shutting down multiple remote hosts with a single command

To achieve all the above, along with some additional features, you will be learning a few concepts in this lesson that will help find a solution for the given scenario.

Learning Objectives

By the end of this lesson, you will be able to:

- Configure Ansible Ad hoc commands
- Evaluate parallelism and shell commands
- Evaluate Ansible modules for Ad hoc commands
- List the Ansible modules

Introduction to Ad Hoc Commands

Introduction

An Ansible Ad Hoc command can be executed independently to perform a quick task.

It excels at tasks that are rarely repeated.

Why Ad Hoc Commands?

Ad hoc commands are fast and easy, but they are not sustainable. These are some points that describe the need for ad hoc commands:

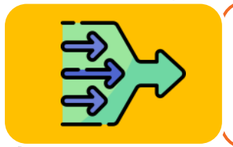
1. They are quick and run individually for a particular task.
2. It allows us to perform critical tasks efficiently without writing playbooks.
3. It makes use of the Ansible command line device to automate a single task on at least one managed node.

Features of Ansible Ad Hoc Commands

Ad hoc Command Features

It shows a faster way of representing data with the preferred servers without storing it for future use.

It shows the simplicity and power of Ansible.



It allows users to run simple tasks in the command line against one or all hosts directly.

Creation of Ansible Ad Hoc Command

Ansible ad hoc commands are the easiest way to understand the system and its capabilities.

Syntax:

```
ansible <hosts> [-m <module_name>] -a <"arguments"> -u
<username> [--become]
```

Ansible Ad Hoc Commands Syntax

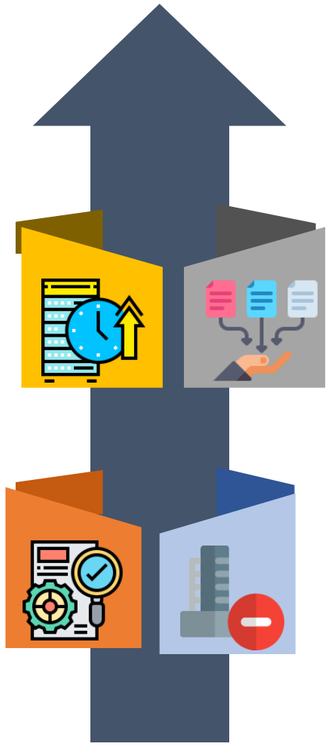
Ad hoc commands are temporary commands that perform a single task.

Host Group	Module	Arguments to the module
Web server	-m yum	-a”name=httpd state=latest
All servers	-m shell	-a”find/opt/oracle/-type f -mtime +10 -name*.log”’
App server	-m user	-a”name=saravak group =admins append=yes shell=bin/bash”

Common Ad Hoc Scenarios

Users can use Ad hoc commands in the following situations:

Validating the uptime of 1 to 200 remote servers



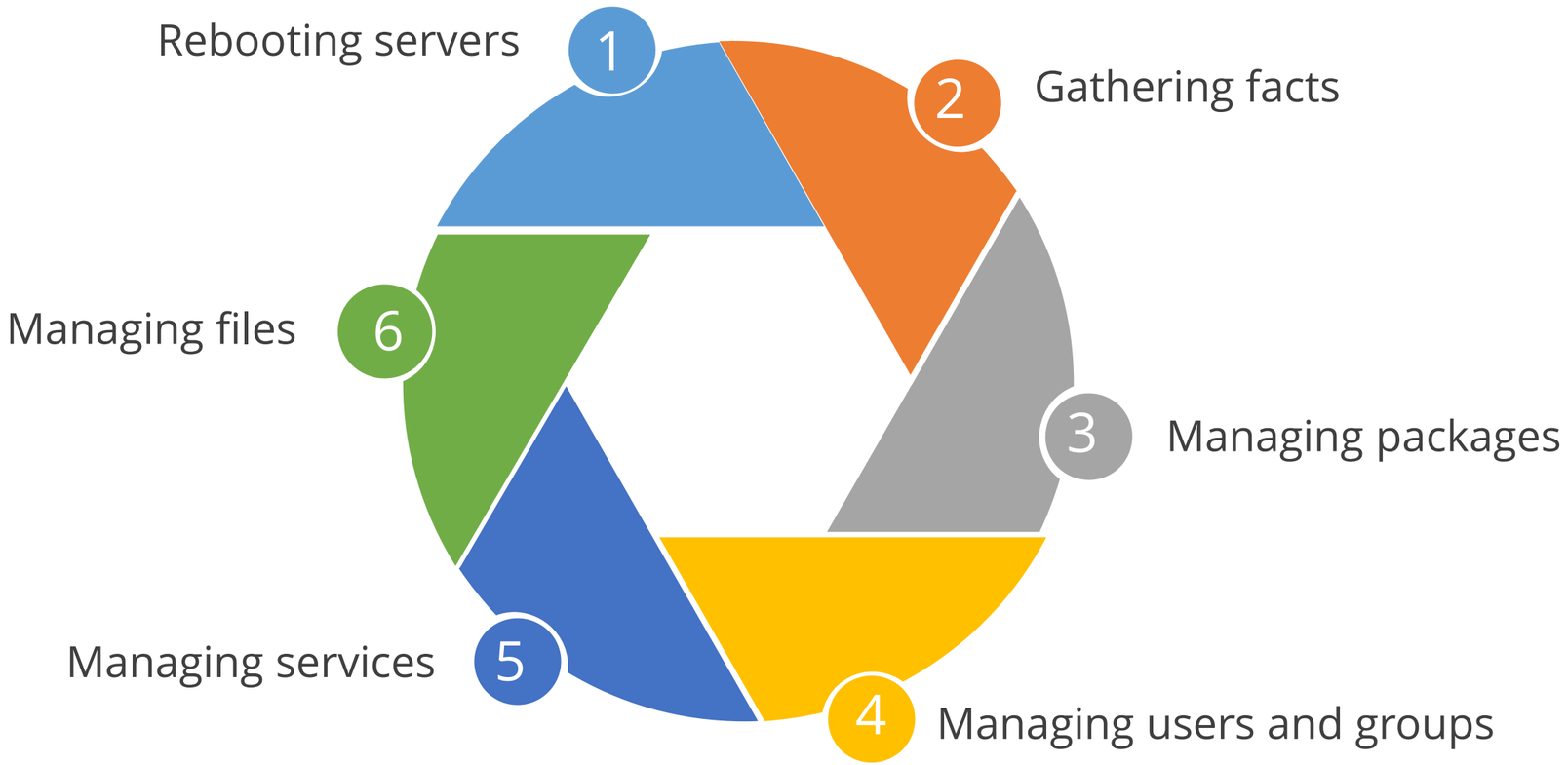
Collecting the disk space available on remote hosts

Validating if the server is alive or not

Shutting down multiple remote hosts with a single command

Use Cases for Ad Hoc Tasks

These are the use cases available for ad hoc tasks:



Ad Hoc Case

An ad hoc case is created for handling a business exception or tracking task which users may be processing.

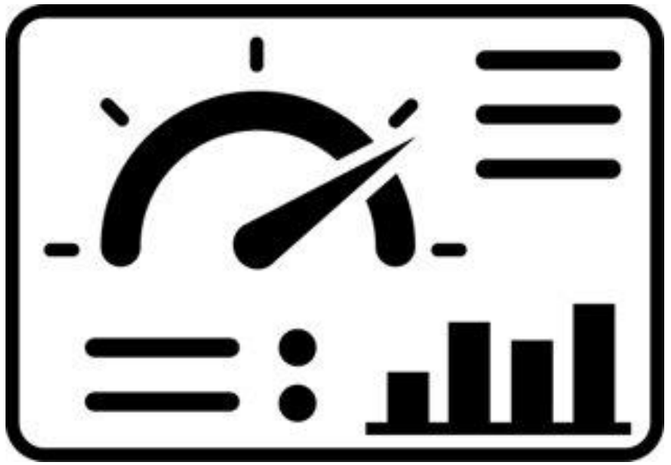
Example:

In any hiring firm, if the hiring specialist wants to carry out a background check for continuing with the hiring process, then an ad hoc case is created for this task.

Whenever an exception arises, operators employ ad hoc cases to avoid waiting for modification of the case-type structure.

Ad Hoc Case

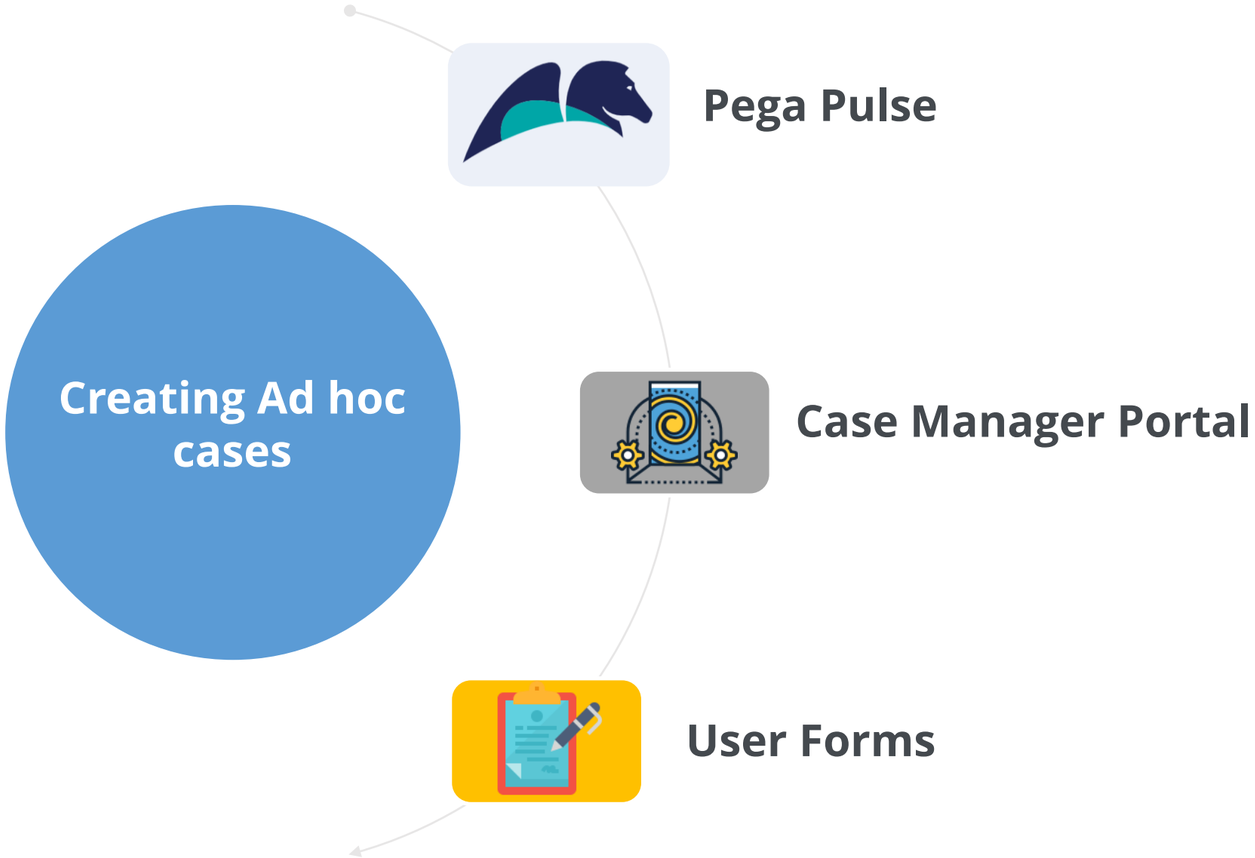
Operators create tasks in one assignment, namely Ad Hoc Case Dashboard, wherein the conventional complete task flow, tasks are processed as assignments.



The standard Simple Case WorkFlow (pySimpleCaseWorkFlow) generates the Work-Cover-SimpleCase class, which features instances known as ad hoc cases.

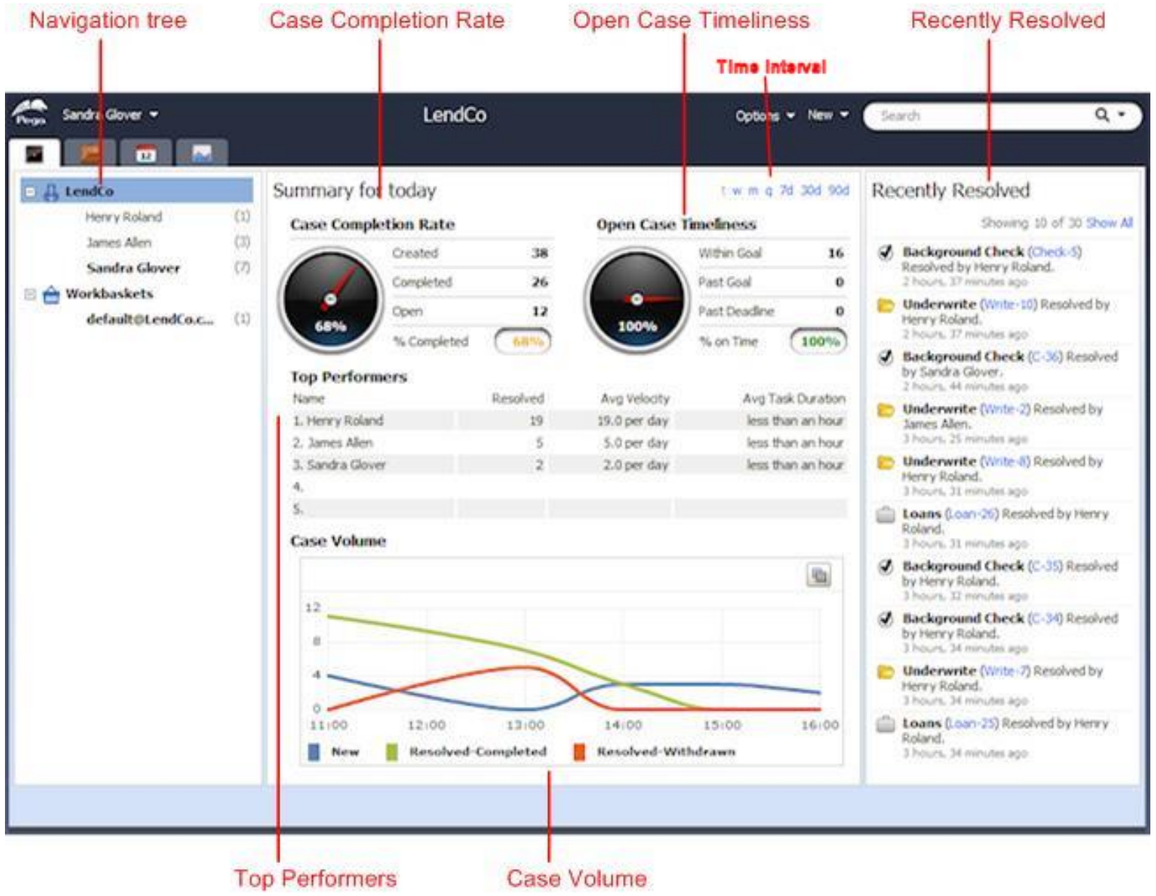
Creation of Ad Hoc Cases

There are three ways for creating ad hoc cases:



Case Manager Portal

The Case Manager Portal consists of four tabs as given below:



My Dashboard

My Cases

My Events

My Reports

Ansible Modules for Ad Hoc Commands

Introduction

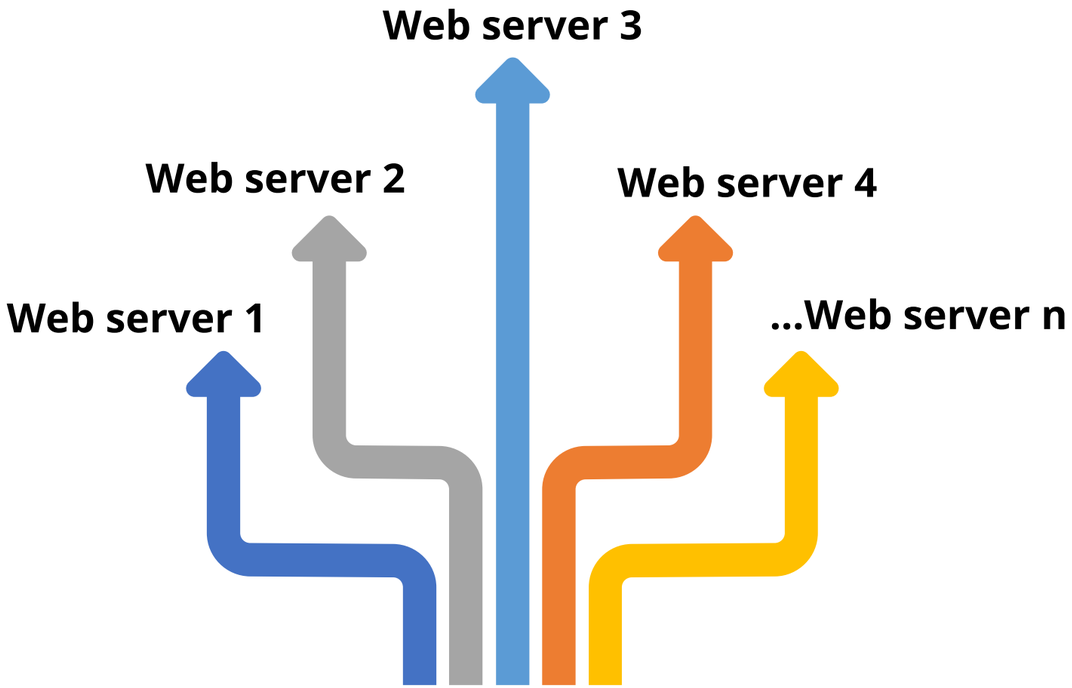
An Ansible ad hoc command makes use of the */usr/container/ansible* order line tool to automate a single task on at least one monitored node.

.

Through the concept of parallelism, the Ansible module allows the user to execute all tasks in the task sections of a playbook in parallel.

Parallelism

Parallelism helps users to create multiple resources at once while saving time.



Each command is executed on all servers in a group in 'n' parallel forks.

Shell Commands

The shell command is used in cases where users execute commands of their choice on remote servers. By default, all the commands are executed on the **/bin/sh** shell.

Shell Commands

The following points show how to execute a single command with Ansible Shell:

1. Initially run the playbook against a host group.
2. Execute a simple date command.
3. Save the output of the command into a register variable.
4. Print the registered variable.

File Transfer

Ansible can SCP lots of files to multiple machines in parallel.

Task	Command
To transfer a file directly to many servers	ansible atlanta -m copy -a "src=/etc/hosts dest=/tmp/hosts"
To create directories	ansible webservers -m file -a "dest=/srv/foo/a.txt mode=600"
To change ownership and permissions on files	ansible webservers -m file -a "dest=/srv/foo/b.txt mode=600 owner=mdehaan group=mdehaan"
To create directories via a particular user	ansible webservers -m file -a "dest=/path/to/c mode=755 owner=mdehaan group=mdehaan state=directory"
To delete directories and files	ansible webservers -m file -a "dest=/path/to/c state=absent"

Managing Packages

There are modules available for yum and apt. Here are some examples with yum:

Task	Command
To check a package is installed	ansible webservers -m yum -a "name=acme state=present"
To check a package is installed with a specific version	ansible webservers -m yum -a "name=acme-1.5 state=present"
To check a package has the latest version	ansible webservers -m yum -a "name=acme state=latest"
To check a package is not installed	ansible webservers -m yum -a "name=acme state=absent"

Users and Groups

The **User** module allows easy creation and manipulation of existing user accounts, as well as the removal of user accounts that may exist:

```
ansible all -m user -a "name=foo password=<crypted
password here>“

ansible all -m user -a "name=foo state=absent"
```

Managing Services

These are the commands to manage services in Ansible:

Task	Command
To check a service is started on all web servers	ansible webservers -m service -a "name=httpd state=started"
To restart a service on all web servers	ansible webservers -m service -a "name=httpd state=restarted"
To check a service is stopped	ansible webservers -m service -a "name=httpd state=stopped"

Gathering Facts

Facts are described in the playbooks section and represent discovered variables about a system. These can be used to implement conditional execution of tasks and to get ad hoc information about the system.

```
ansible all -m setup
```

It is also possible to filter this output to export particular facts by setting up the module.

Assisted Practice

Executing Ad hoc Commands

Duration: 10 Min.

Problem Statement:

You have been assigned the task of executing ad hoc commands to check memory usage, execute the command as root user on host, and create a UNIX user group.

Assisted Practice: Guidelines

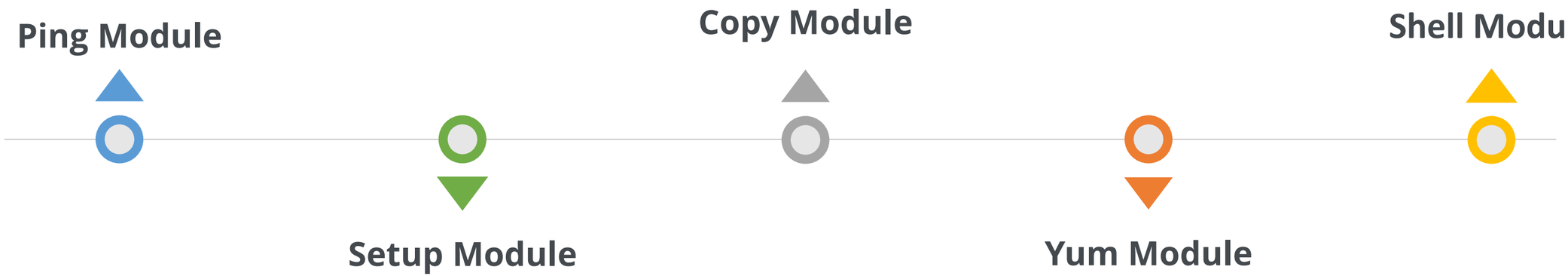
Steps to be performed:

1. Set up the web server
2. Execute Ansible modules on a local server
3. Execute free –m commands on remote hosts

Common Ansible Modules

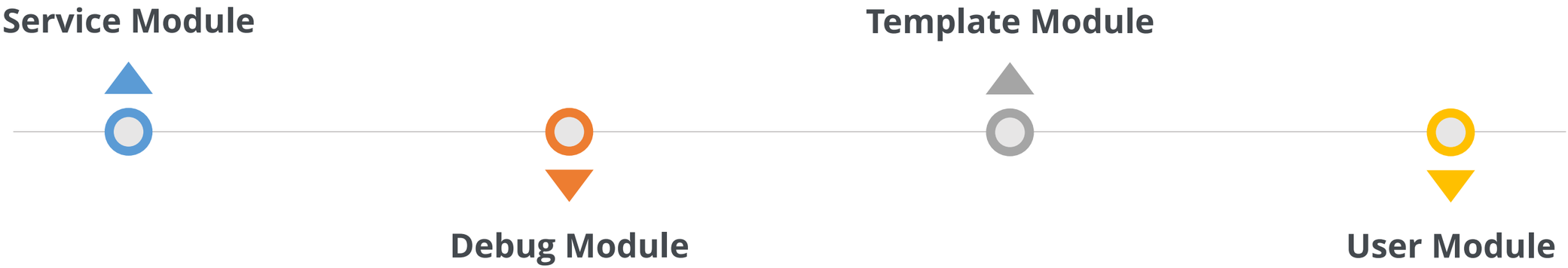
Ansible Modules

Ansible is all about using different modules in its Playbook. These modules include:



Ansible Modules

Ansible is all about using different modules in its Playbook. These modules include:



Ping Module

Ping helps the user to check whether the connection with their hosts mentioned in the inventory file is established or not.

The structure of a ping module:

```
Open command line and run:

ansible test-servers -m ping -u ec2-user
```

Ping changes to pong if an SSH connection is established.

Setup Module

The setup module allows the user to check all the hosts, their configurations, and comprehensive information.

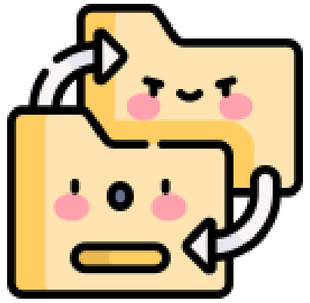
The structure of a setup module:

```
Open command line

Run ansible test-servers -m setup -u ec2-user
```

Copy Module

The copy module is frequently used in writing playbooks when copying a file from a distant server to target nodes.



Example:

If a user needs to transfer a file from a distant server to all destination computers

Yum Module

Yum is a package management tool. It allows users to obtain, install, delete, query, and manage packages.

The structure of a yum module:

```
Open command line and run:

ansible test-servers -m yum -a 'name=httpd state=present' -
become -u ec2-user
```

Shell Module

The Ansible shell module is used to run Shell commands on target Unix-based systems.

The structure of a shell module:

```
Open command line and run:

ansible test-servers -m shell -a 'ls -la' -u ec2-user
```

Service Module

The service module guarantees the status of the service that it is operating.

The structure of a service module:

```
Open command line and run:

ansible test-servers -m service -a 'name=httpd state=started'
-become -u ec2-user
```

Debug Module

Debug module prints a message on hosts.

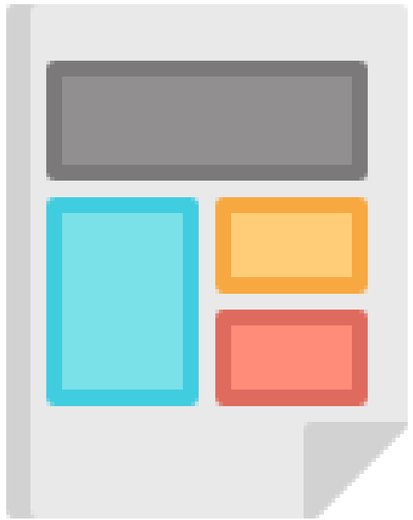
The structure of a debug module:

```
Open command line

Run ansible test-servers -m debug -a 'msg=Hello' -u ec2-user
```

Template Module

Ansible is a tool for managing numerous server and environment settings. Ansible template modules are useful in this situation.



User Module

The user module adds a particular user to the server.

Assisted Practice

Demonstrate Ansible Modules

Duration: 15 Min.

Problem Statement:

You have been assigned the task of checking the different functionalities of ad hoc modules, including setting up an SSH connection, setting up and configuring different hosts, installation of service, and printing a message on hosts.

Assisted Practice: Guidelines

Steps to be performed:

1. Executing Ansible modules on a local server

Key Takeaways

- Ad hoc commands execute themselves to complete a quick task
- An ad hoc case can handle any business exception very smoothly.
- Parallelism allows managing multiple web servers through a parallel fork at the same time while saving time.
- Ansible employs all its modules in its Playbook.

Integrate Ad-hoc Commands: PING, SHELL, APT

Duration: 25 Min.

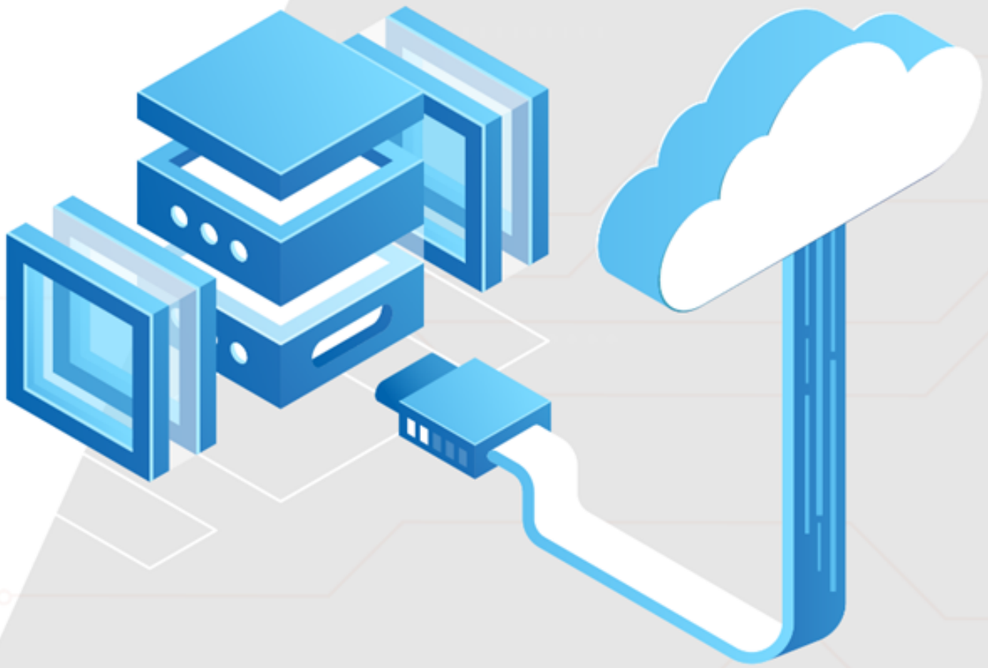
Project agenda: To integrate ad-hoc commands: PING, SHELL, APT

Description: There are multiple tasks in Ansible where you don't need to write a separate Ansible playbook for it; you can just run an ansible ad-hoc command for that task. These are one-liner command to perform a single task on the target host. These commands are present in /usr/bin/ansible.

Perform the following:

1. Install Ansible on main node (Ansible server)
2. Add host and groups to hosts file
3. Establish the connectivity between the hosts specified in the host file and the ansible server
4. Ad-hoc command with ping, shell, and apt module

LESSON-END PROJECT



Thank you