AWS-DEVOPS

DATE : 12.03.2024

DT/NT : DT

LESSON : ANSIBLE

SUBJECT: INTRODUCTION TO ANSIBLE

BATCH : B 224

techproeducation.com
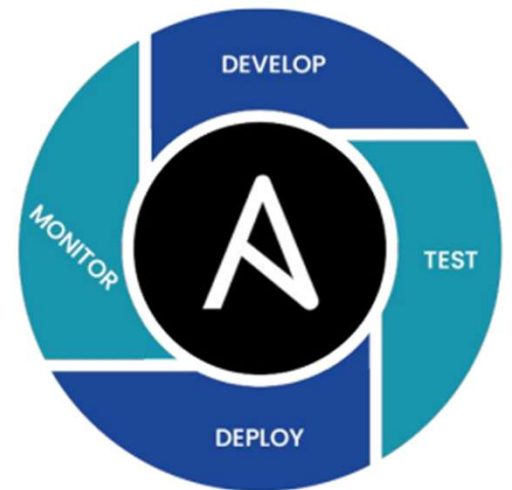
+1 (585) 304 29 59

TECHPRO
EDUCATION

Table of Contents

- ▶ About Ansible
- ▶ Installation
- ▶ Configuring Ansible
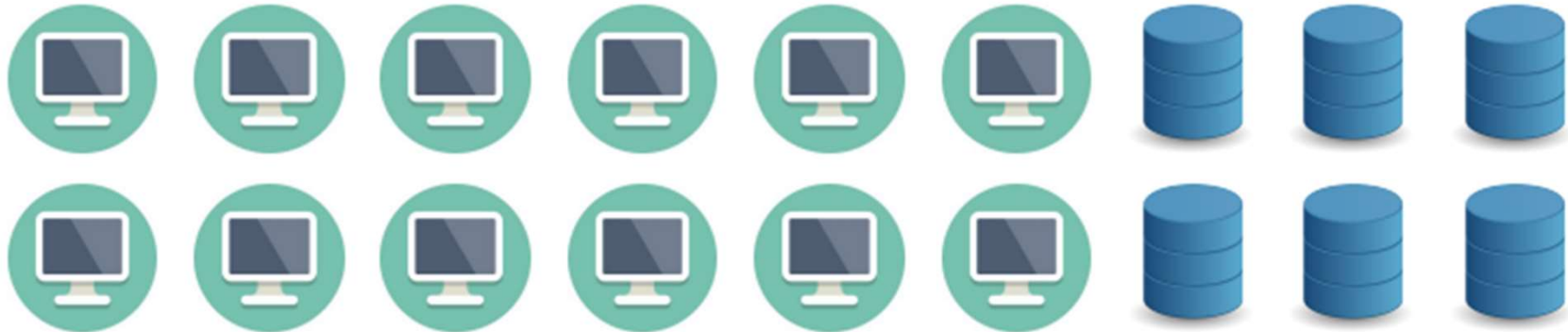- ▶ Ansible Concepts
- ▶ ad-hoc Commands

About Ansible

Ansible **is an open-source IT automation tool**. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments.

About Ansible

Scripts
- Time
- Coding Skills
- Maintenance

- Simple
- Powerfull
- Agentless

About Ansible

Scripts

```bash
#!/bin/bash
# Script to add a user to Linux system
if [ $(id -u) -eq 0 ]; then
    $username=johndoe
    read -s -p "Enter password : " password
    egrep "^$username" /etc/passwd >/dev/null
    if [ $? -eq 0 ]; then
        echo "$username exists!"
        exit 1
    else
        useradd -m -p $password $username
        [ $? -eq 0 ] && echo "User has been added
to system!" || echo "Failed to add a user!"
    fi
fi
```

Playbook

```yaml
- hosts: all_my_web_servers_in_DR
  tasks:
    - user:
        name: johndoe
```

About Ansible

SIMPLE

- Human readable automation
- No special codding skills needed
- Tasks executed in order

Get productive quickly

POWERFULL

- App deployment
- Configuration management
- Workflow orchestration

Orchestrate the app lifecycle

AGENTLESS

- Agentless architecture
- Uses Open SSH
- No agents to exploit or update

More efficient & more secure

Installation

We can install Ansible using yum and apt package managers.

To install on Amazon Linux :

sudo amazon-linux-extras install ansible2 –y

For install with apt:

sudo apt-get -y install ansible

**For install with pip:**

pip install ansible

Configuring Ansible

• Ansible supports several sources for configuring its behavior, including a file named ansible.cfg, environment variables, command-line options, playbook keywords, and variables.

• Certain settings in Ansible are adjustable via a configuration file (ansible.cfg).

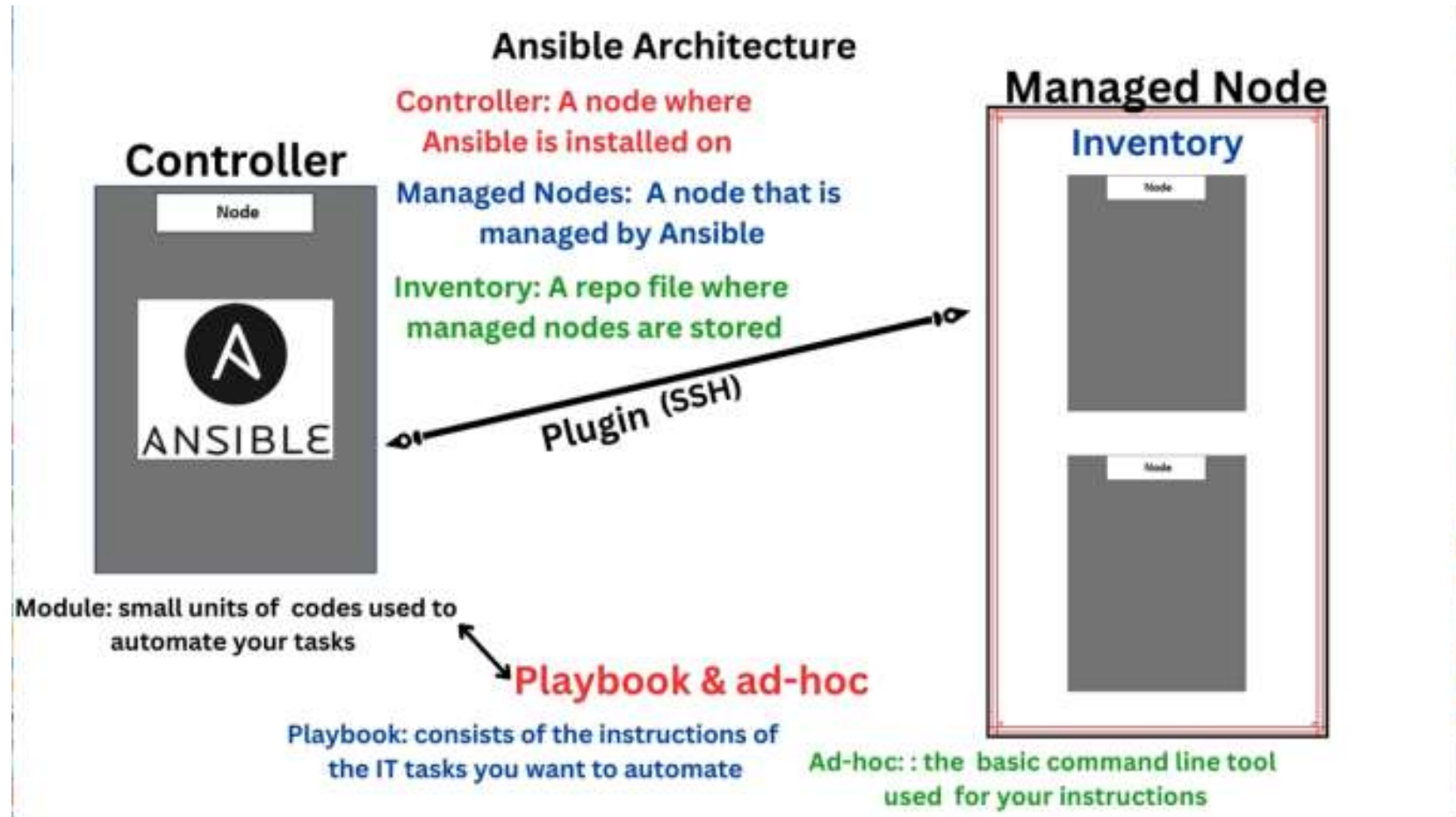• Changes can be made and used in a configuration file which will be searched for in the following order:
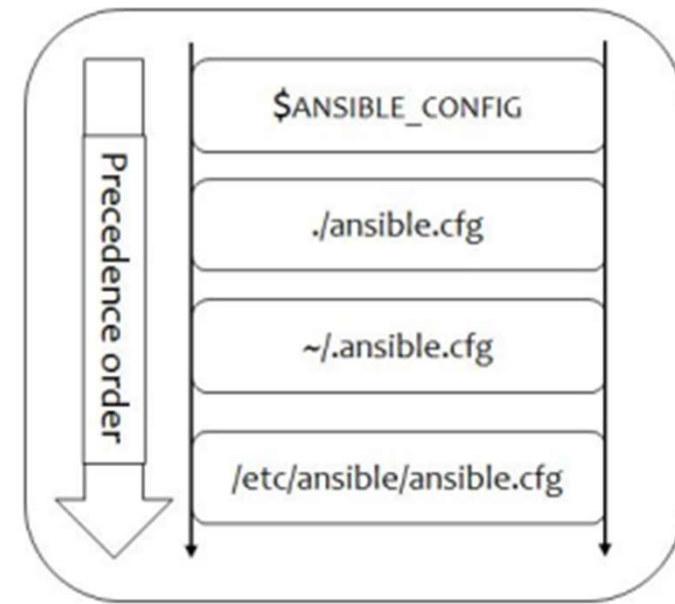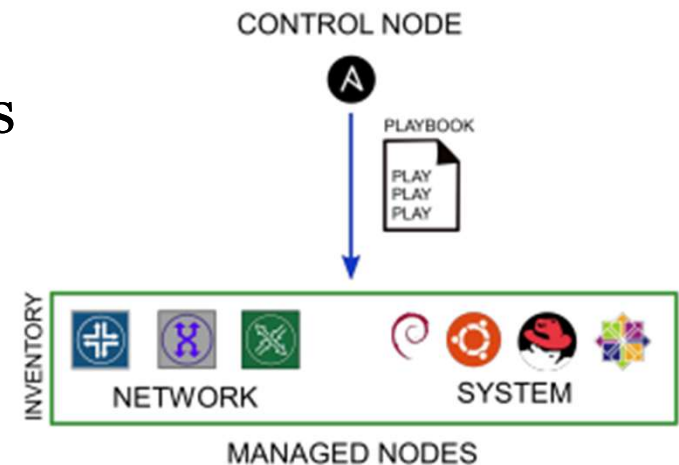
Ansible Concepts
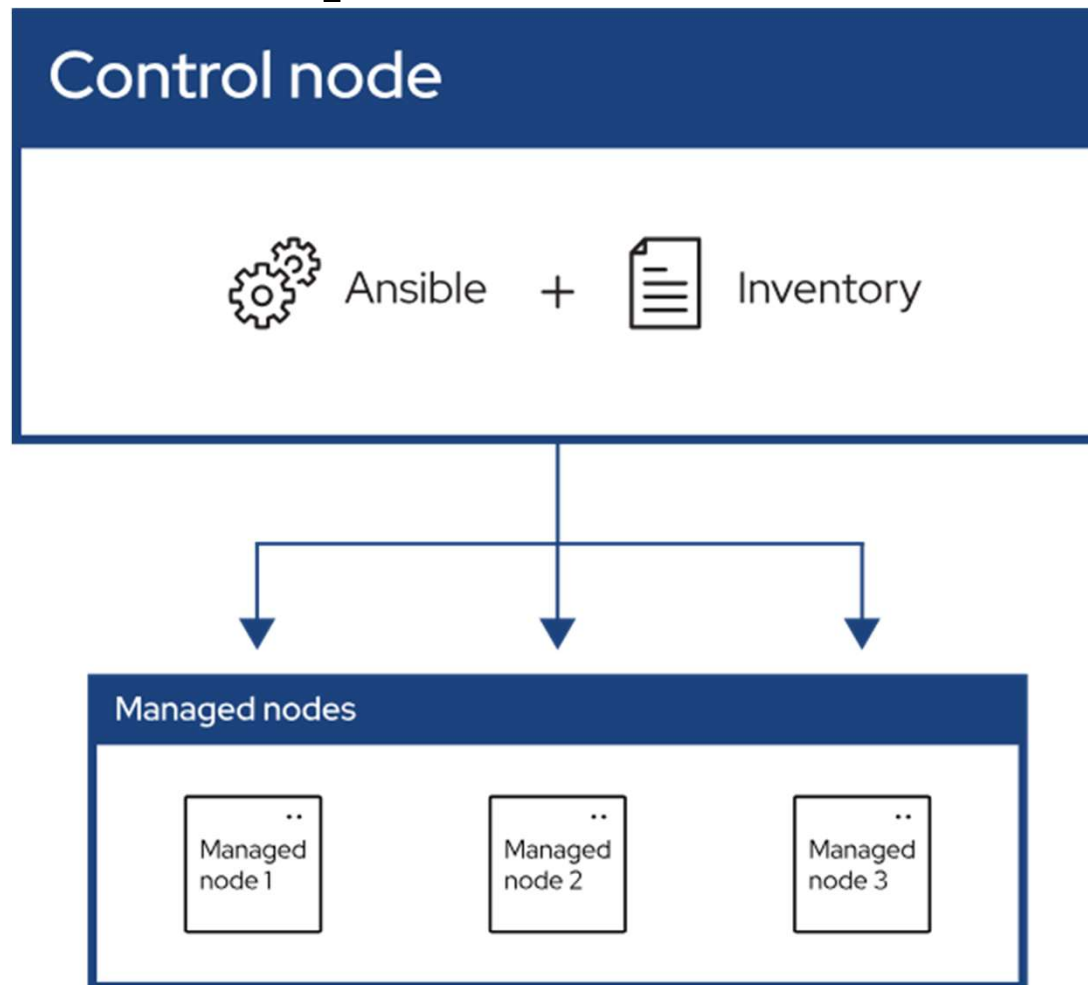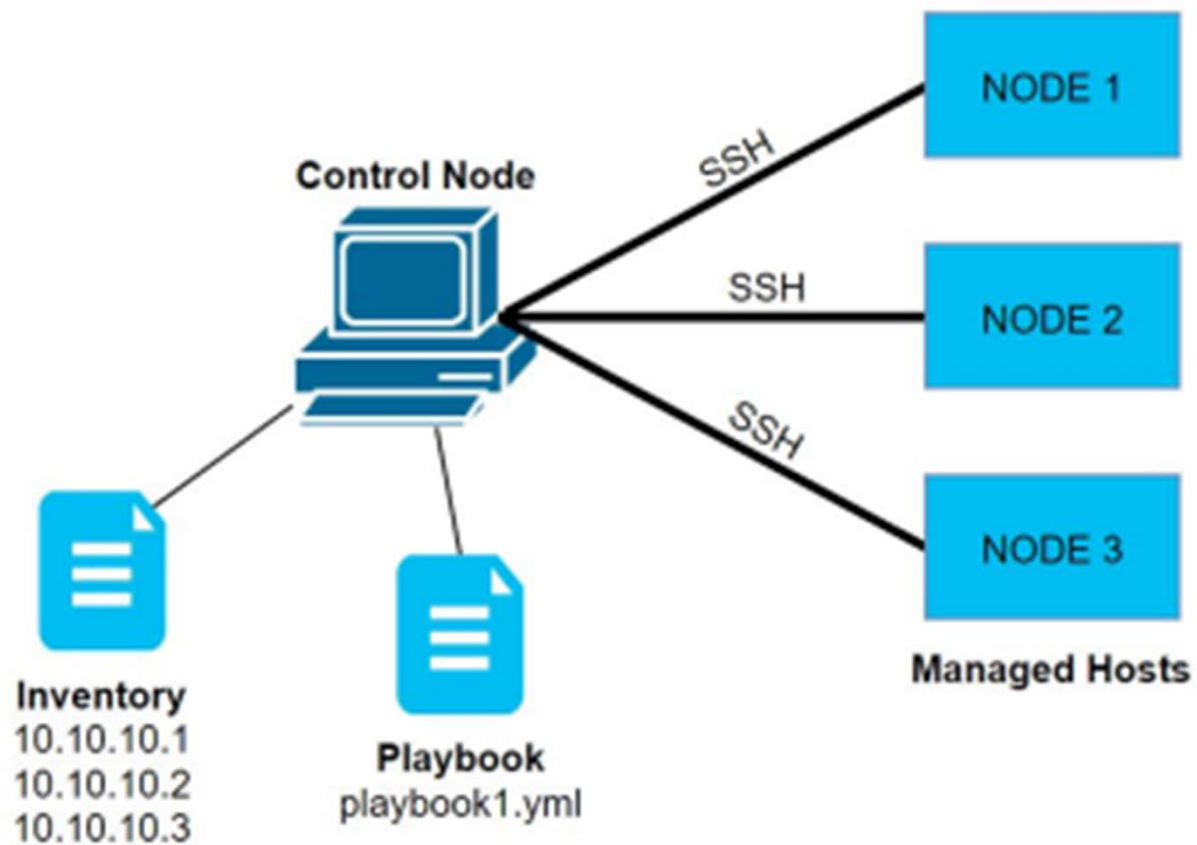
Ansible Concepts

Ansible Concepts

Ansible Concepts

Control node:

Any machine with Ansible installed. You can run commands and playbooks, invoking /usr/bin/ansible or /usr/bin/ansible-playbook, from any control node. You can use any computer that has Python installed on it as a control node - laptops, shared desktops, and servers can all run Ansible. However, you cannot use a Windows machine as a control node.
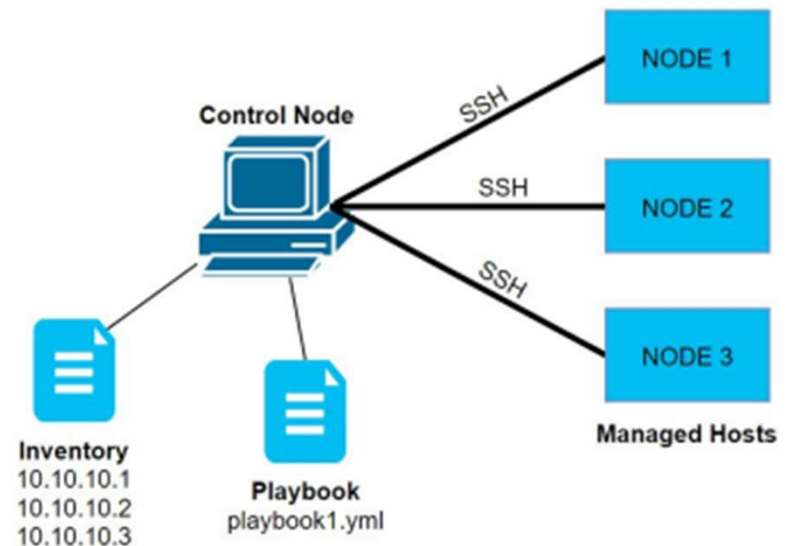
Ansible Concepts

Managed Nodes:

The network devices (and/or servers) you manage with Ansible.

Managed nodes are also sometimes called hosts. Ansible is not installed on managed nodes.

Ansible Concepts

Inventory:
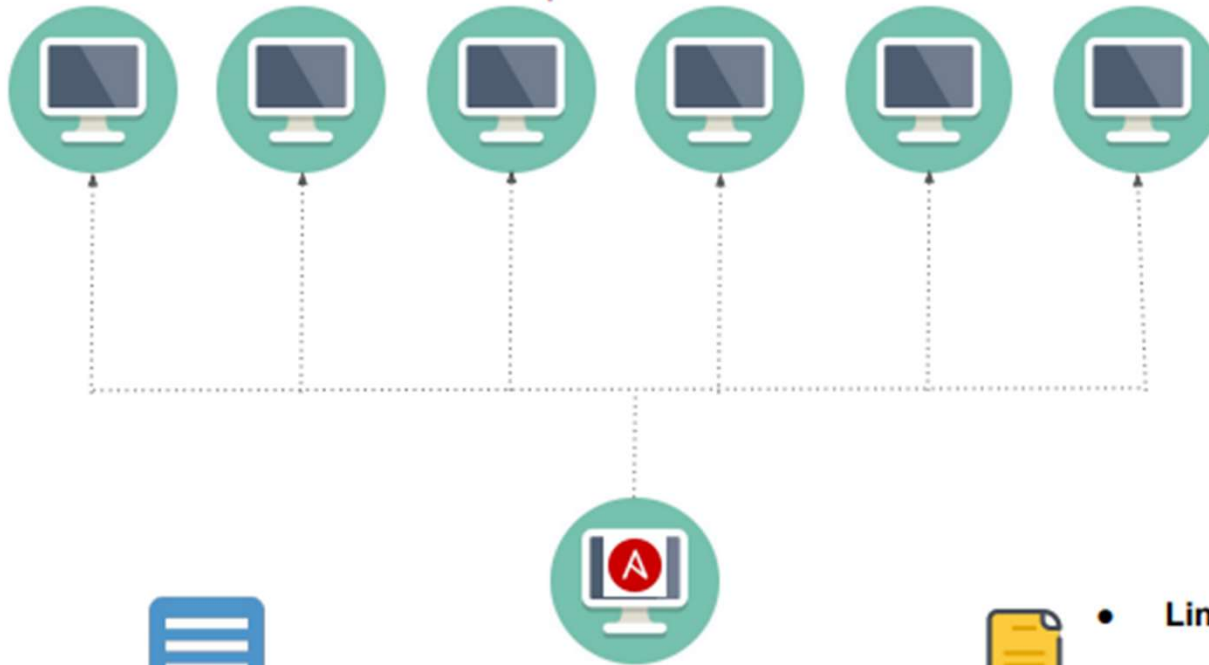
A list of managed nodes. An inventory file is also sometimes called a hostfile. Your inventory can specify information like IP address for each managed node. An inventory can also organize managed nodes, creating and nesting groups for easier scaling.

The inventory file

Where it is located
/etc/ansible/hosts

What is the format

```
[mailservers]
mail.example.com

[webservers]
foo.example.com ansible_ssh_user = user001
bar.example.com ansible_ssh_private_key_file =
/.ssh/ansible.key001

[dbservers]
one.example.com
two.example.com
db-[a:f].example.com
```

Ansible Concepts

Ansible Concepts

Group hosts for easier inventory selection and less conditional tasks -- the more groups the better.

WHAT	WHERE	WHEN
[db]	[east]	[dev]
db[1:4]	db1	db1
	web1	web1
[web]	db3	
web[1:4]	web3	[test]
		db3
	[west]	web3
	db2	
	web2	[prod]
	db4	db2
db1 = db, east, dev	web4	web2
		db4
		web4

Ansible Concepts

Playbooks:

Ordered lists of tasks, saved so you can run those tasks in that order repeatedly. Playbooks can include variables as well as tasks. Playbooks are written in YAML and are easy to read, write, share and understand.

Playbook

```
---
- hosts: webserver
  remote_user: ubuntu
  tasks:                     } a play
    - apt: name=git state=present

- hosts: dbserver
  remote_user: ubuntu
  tasks:                     } a play
    - apt: name=mysql state=present
```

Ansible Concepts

Tasks:

The units of action in Ansible. You can execute a single task once with an ad-hoc command.
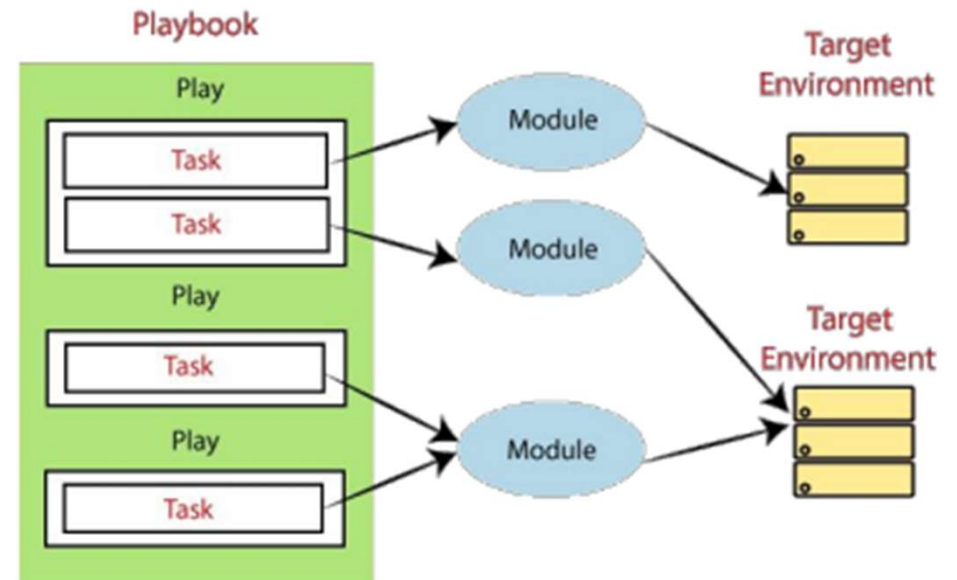
Ansible Concepts

Modules:

The units of code Ansible executes. Each module has a particular use, from administering users on a specific type of database to managing VLAN interfaces on a specific type of network device.

Modules	Module Categories							
System	User	Group	Iptables	Mount	Ping	Systemd	Service	Hostname
Commands	Command	Expect	Raw	Script	Shell			
Files	Acl	Archive	Find	Copy	Replace	Stat	File	Unarchive
Database	MySQL	MongoDB	MSSQL	PostgreSQL	ProxySQL	Vertica		
Cloud	Amazon	Azure	Google	Linode	Openstack	VMware	Docker	Atomic
Windows	Win_copy	Win_command	Win_msi	Win_ping	Win_msq	Win_shell	Win_path	Win_service

ad-hoc Commands

- An Ansible ad-hoc command uses the /usr/bin/ansible command-line tool to automate a single task on one or more managed nodes.
- Ad-hoc commands are quick and easy, but they are not reusable.
- Ad-hoc commands demonstrate the simplicity and power of Ansible.
- Ad-hoc commands are great for tasks you repeat rarely.

AD HOC command

| ansible | webserver | -m yum | -a "name=httpd state=latest" |

Ansible Playbook

```
---
- name: playbook name
  hosts: webserver
  tasks:
    - name: name of the task
      yum:
        name: httpd
        state: latest
```
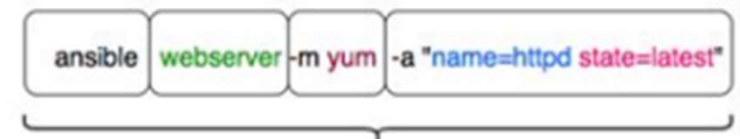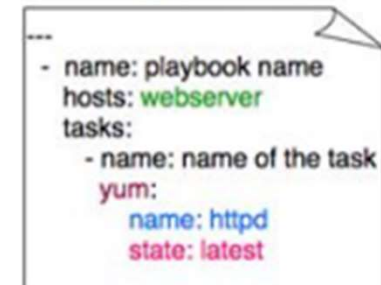
ad-hoc Commands

- **ansible** <inventory> -m

AD HOC command

| ansible | webserver | -m yum | -a "name=httpd state=latest" |

Runs a command or **calls a module** directly from the **command line**, no Playbook required

```
ansible <inventory> <options>
ansible web -a /bin/date
ansible web -m ping
ansible web -m yum -a "name=openssl state=latest"
```

Conclusion

Conclusion

Conclusion

Do you have any questions?

Send it to us! We hope you learned something new.

TECHPRO EDUCATION