

The background is a solid dark blue color. In the top right corner, there is a decorative pattern of overlapping triangles in various shades of blue, including a lighter blue and a darker blue, creating a geometric, abstract design.

# **ANSIBLE AUTOMATION AND ORCHESTRATION**

# AGENDER

- What is Ansible ?
- How ansible Master and Client Architecture
- Ansible Configuration Files
- Ansible Inventory/Host File
- Ansible PasswordLess Authentication/Key base Authentication
- Adhoc Commands

# AGENDA

---

WHAT IS ANSIBLE?

ANSIBLE PLAYBOOKS

INTRODUCTION TO ANSIBLE ROLES

CREATE ANSIBLE ROLE

ANSIBLE ROLE DIRECTORY STRUCTURE

HANDS-ON

# WHAT IS ANSIBLE?

ANSIBLE

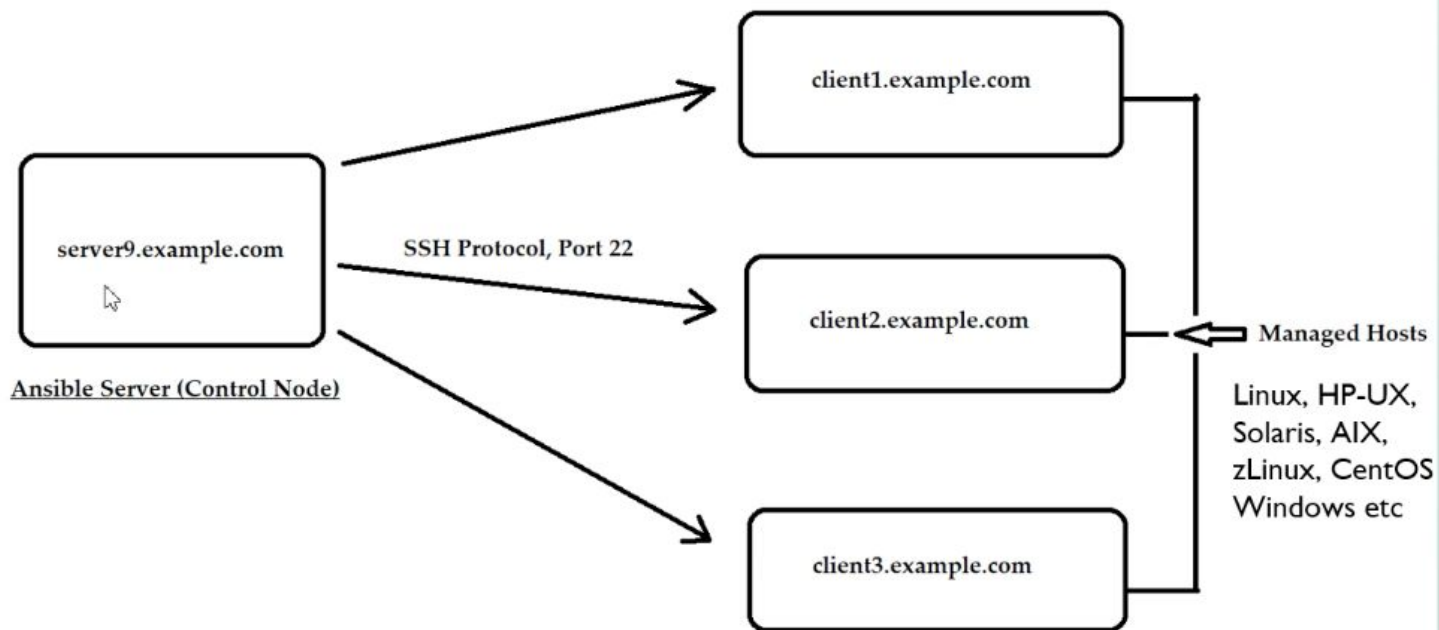
- ✓ Ansible is an **open source IT Configuration Management, Deployment & Orchestration tool**.
- ✓ It aims to **provide large productivity gains** to a wide variety of automation challenges.
- ✓ This tool is very simple to use yet powerful enough to **automate complex multi-tier IT application environments**.



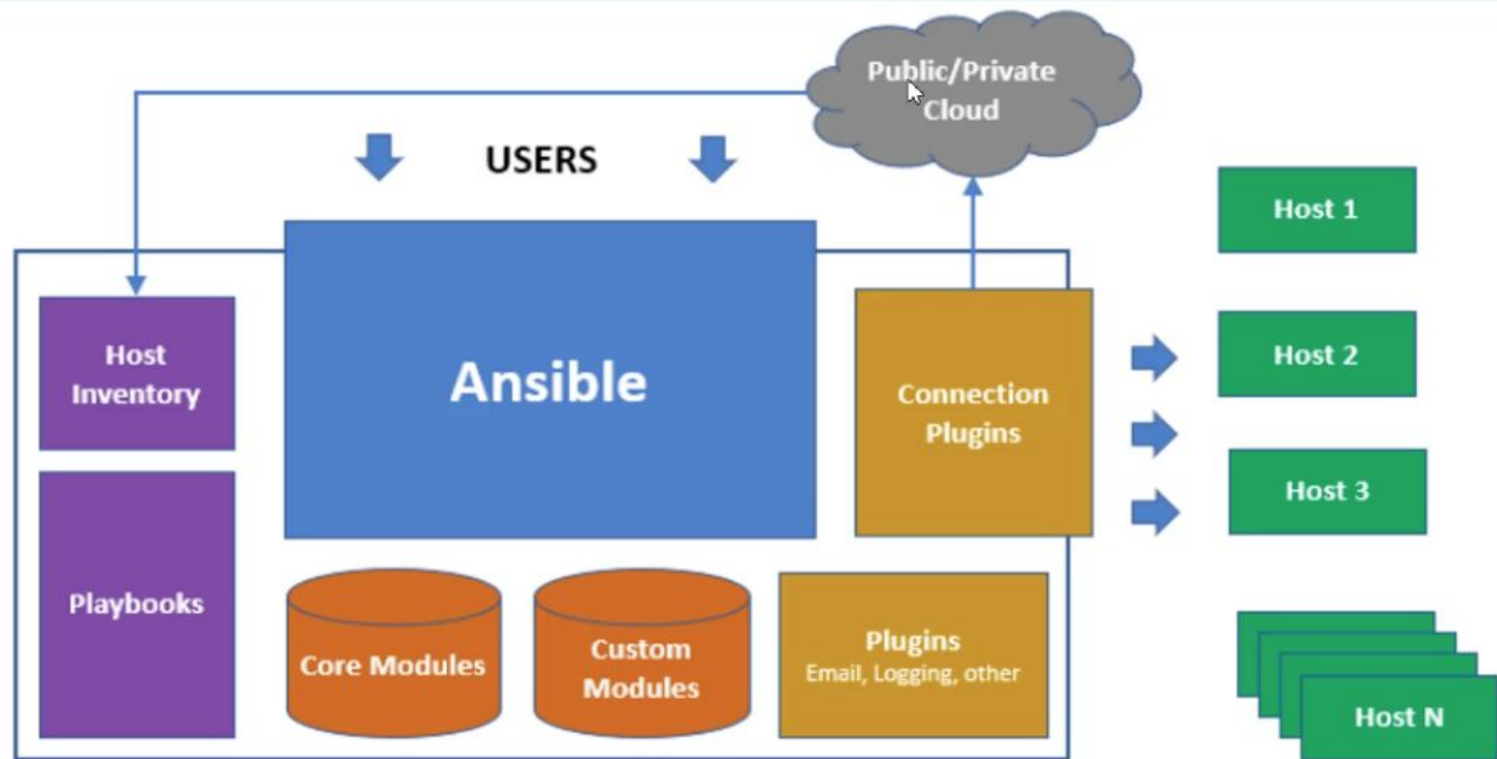
ANSIBLE

ANSIBLE USES THE PUSH APPROACH

# Ansible Automation



# Ansible Automation



02

---

# ANSIBLE PLAYBOOKS

# ANSIBLE PLAYBOOKS

## ANSIBLE PLAYBOOKS

Ansible Playbooks are the access points to Ansible provisioning. Written in YAML, playbooks provide a way to deploy and configure remote servers on various environments. On a higher level, these playbooks are used to handle multi-tier rollouts and load balancing tasks for the servers.





# SAMPLE PLAYBOOK TO START APACHE HTTPD PROGRAM

```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
    - name: ensure apache is at the latest version
      yum: name=httpd state=latest
    - name: write the apache config file
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf
      notify:
        - restart apache
    - name: ensure apache is running (and enable it at
      boot)
      service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

# SAMPLE PLAYBOOK TO START APACHE HTTPD PROGRAM

---

```
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
    - name: ensure apache is at the latest version
      yum: name=httpd state=latest
    - name: write the apache config file
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf
      notify:
        - restart apache
    - name: ensure apache is running (and enable it at
      boot)
      service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

Every YAML file starts with 3  
dashes [---]

# SAMPLE PLAYBOOK TO START APACHE HTTPD PROGRAM

## HOSTS

```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
    - name: ensure apache is at the latest version
      yum: name=httpd state=latest
    - name: write the apache config file
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf
      notify:
        - restart apache
    - name: ensure apache is running (and enable it at
      boot)
      service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

Hosts are a list one or more groups or host patterns, separated by colons

# SAMPLE PLAYBOOK TO START APACHE HTTPD PROGRAM

## VARIABLES

```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
    - name: ensure apache is at the latest version
      yum: name=httpd state=latest
    - name: write the apache config file
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf
      notify:
        - restart apache
    - name: ensure apache is running (and enable it at
      boot)
      service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

Used to enable more flexibility in playbooks and roles. They are also used to loop through a set of given values, access various information and replace certain strings in templates.

# SAMPLE PLAYBOOK TO START APACHE HTTPD PROGRAM

## USERS

```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
    remote_user: root
  tasks:
    - name: ensure apache is at the latest version
      yum: name=httpd state=latest
    - name: write the apache config file
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf
      notify:
        - restart apache
    - name: ensure apache is running (and enable it at
      boot)
      service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

User as the name suggests is  
the name of the user account.  
Here it is the root user.

# SAMPLE PLAYBOOK TO START APACHE HTTPD PROGRAM

```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
    - name: ensure apache is at the latest version
      yum: name=httpd state=latest
    - name: write the apache config file
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf
      notify:
        - restart apache
    - name: ensure apache is running (and enable it at
      boot)
      service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

## TASKS

Tasks allow you to break up bits of configuration policy into smaller files. Task includes pull from other files.

# SAMPLE PLAYBOOK TO START APACHE HTTPD PROGRAM

```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
    - name: ensure apache is at the latest version
      yum: name=httpd state=latest
    - name: write the apache config file
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf
      notify:
        - restart apache
    - name: ensure apache is running (and enable it at
      boot)
      service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

## HANDLERS

Handlers are just like regular tasks in an Ansible playbook, but only run if the Task contains a notify directive and also indicates that it changed something.



03

---

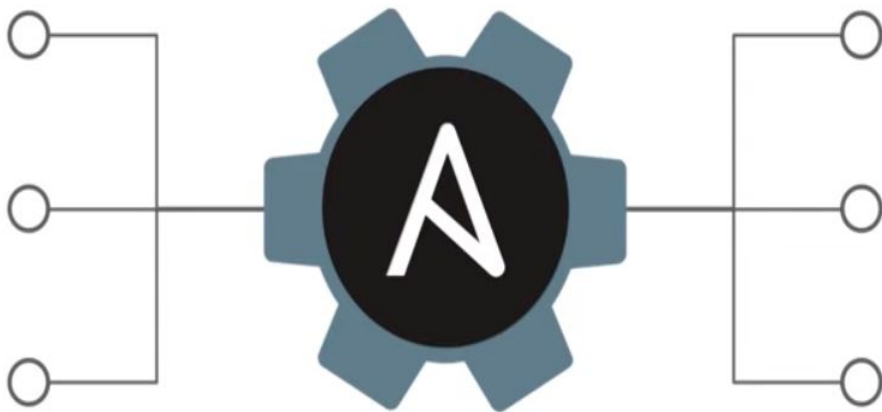
# ANSIBLE ROLES



# ANSIBLE ROLES

## ANSIBLE ROLES

- ✓ Ansible Role is a concept that deals with ideas rather than events. Its basically another level of abstraction used to organize playbooks.



# ANSIBLE ROLES

## ANSIBLE ROLES

- ✓ Ansible Role is a concept that deals with ideas rather than events. Its basically another level of abstraction used to organize playbooks.
- ✓ They provide a skeleton for an independent and reusable collection of variables, tasks, templates, files, and modules which can be automatically loaded into the playbook.



# CREATE ANSIBLE ROLE

To create a role, a tool called **Ansible-galaxy** comes bundled with Ansible.

**Syntax:**

```
ansible-galaxy init roleName
```

**Example:**

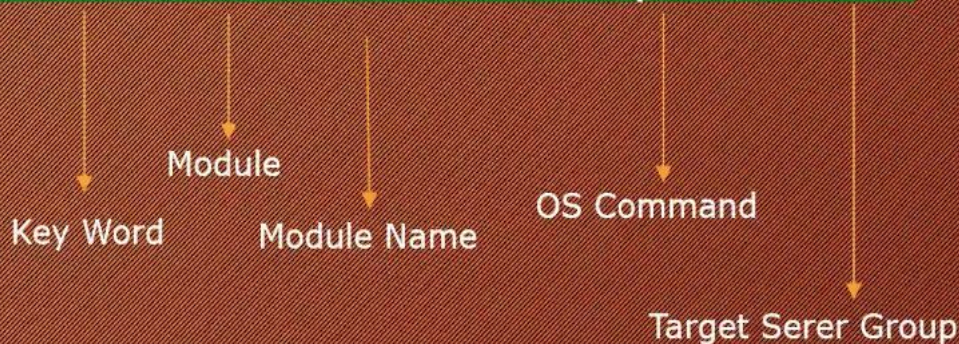
```
ansible-galaxy init samplerolename
```



# Automation with Ansible

How to run remote command using Ansible?

`Ansible -m command -a "uptime" dev`





# PRACTICAL LAP SESSION