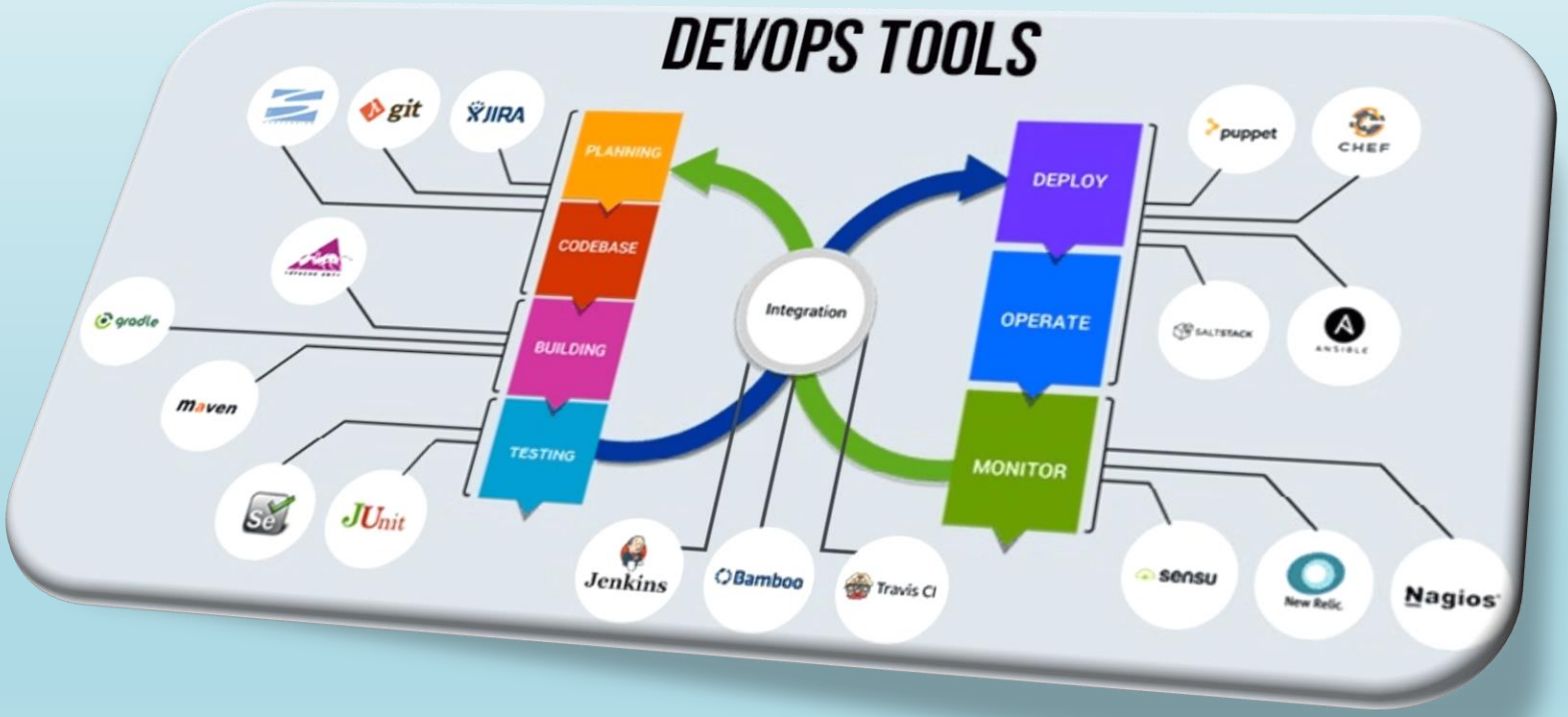
CLOUD TRAIN
ACCELERATE YOUR GROWTH

Configuration Management [Ansible]



CLOUD TRAIN
ACCELERATE YOUR GROWTH

Agenda

- WHAT IS ANSIBLE?
- WHY ANSIBLE?
- HOW DOES ANSIBLE WORK?
- CASE STUDY: NASA
- SETTING UP MASTER SLAVE
- ANSIBLE PLAYBOOKS
- ANSIBLE ROLES
- USING ROLES IN PLAYBOOK

What is Ansible?

What is Ansible?

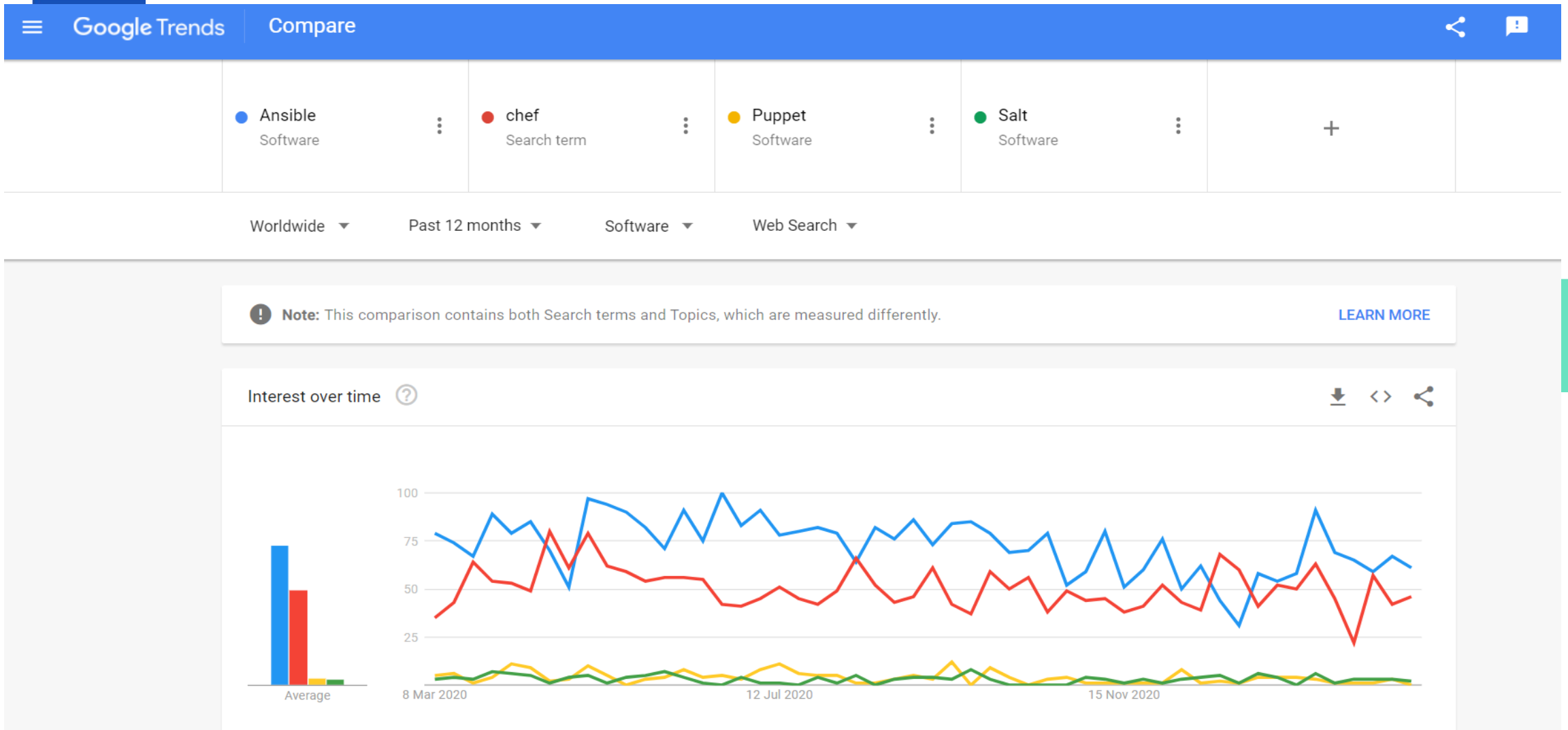
CLOUD TRAIN
ACCELERATE YOUR GROWTH

- Ansible is an open-source configuration management tool
- Used for configuration management
- Can solve wide range of automation challenges
- Written by Michael DeHaan
- Named after a fictional communication device, first used by Ursula K. LeGuin in her novel Rocannon's World in 1966
- In 2015 Red Hat acquired Ansible

ANSIBLE

Why Ansible?

Why Ansible?



Google Trends Results for Ansible

Career Opportunities of Ansible

CLOUD TRAIN
ACCELERATE YOUR GROWTH

DevOps Engineer

BlackBuck Logistics ★★★☆☆ 3 reviews - Bengaluru, Karnataka

₹15,00,000 - ₹17,00,000 a year

Responsibilities and Duties

- 3 - 8 years of experience
- Hands-on experience with any flavour of Linux and can perform basic administrative tasks
- Hands-on experience working with AWS (EC2, VPC, S3, EBS, RDS, IAM, etc)
- Familiarity with a CI/CD system (e.g. Jenkins, Ansible, Puppet)
- Familiarity with a monitoring & alerting system (e.g. Nagios, NewRelic, etc)
- Has an understanding of web architecture, distributed systems, single points of failures, etc.
- Hands-on with a scripting language (preferably Python)
- Good Networking Fundamentals - understands SSH, DNS, DHCP, Load Balancing, Firewalls, etc.
- Basic knowledge of Security good practices e.g. firewalls, etc.
- Worked in an Indian Startup before

BLACKBUCK

Career Opportunities of Ansible

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Software Engineer, Sr. Principal

Epsilon India ★★★½☆ 4 reviews - Bengaluru, Karnataka

Must Have:

- Strong knowledge of configuration management process using software such as Ansible, Puppet or Chef.
- Experience with monitoring tools like Nagios, Munin, Zenoss, etc.
- Experience with Release Engineering and Continuous Integration using tools like Maven, Jenkins, etc.
- Configuring, setting up and tuning of JBOSS, Tomcat, WebSphere, WebLogic, Apache, HAProxy servers or equivalent.
- Experience with using tools like Git, SVN etc and knowledge of SCM concepts.

EPSILON®

Advantage of Ansible

CLOUD TRAIN
ACCELERATE YOUR GROWTH

- Easy to learn
- Written in Python
- Easy installation and configuration steps
- No need to install ansible on slave
- Highly scalable

ANSIBLE

Popularity of Ansible

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Apple

NASA

Intel

Percussion

Cisco

Twitter

CLOUD TRAIN
ACCELERATE YOUR GROWTH

How does Ansible work?

How does Ansible work?

CLOUD TRAIN
ACCELERATE YOUR GROWTH

With the help of **Ansible Playbooks**,
which are written in a very simple language, **YAML**

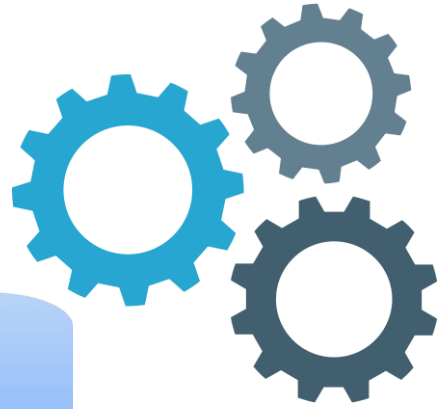
Configuration Management

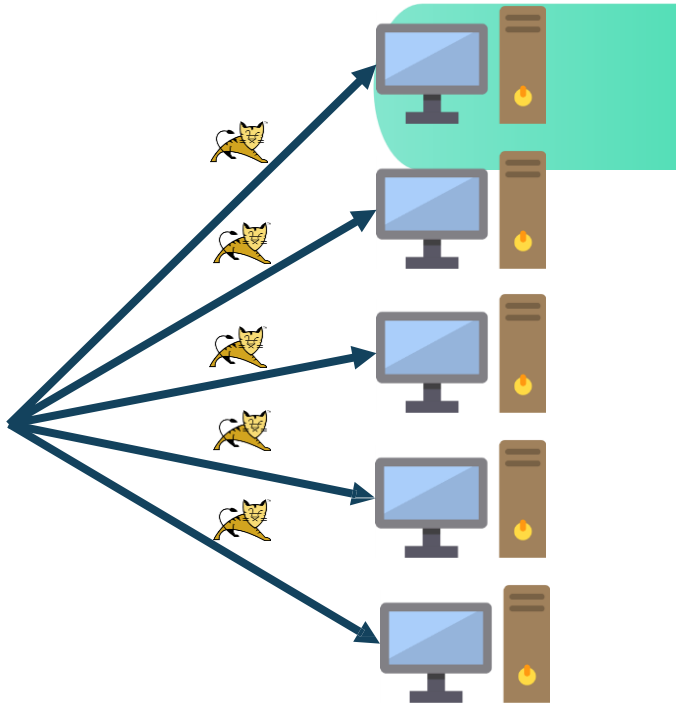
Problem Statement

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Say, Josh runs an enterprise, wants to install a new version of Apache Tomcat in all the systems

Configuration Management



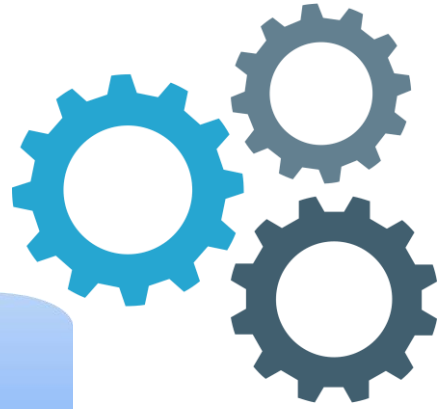


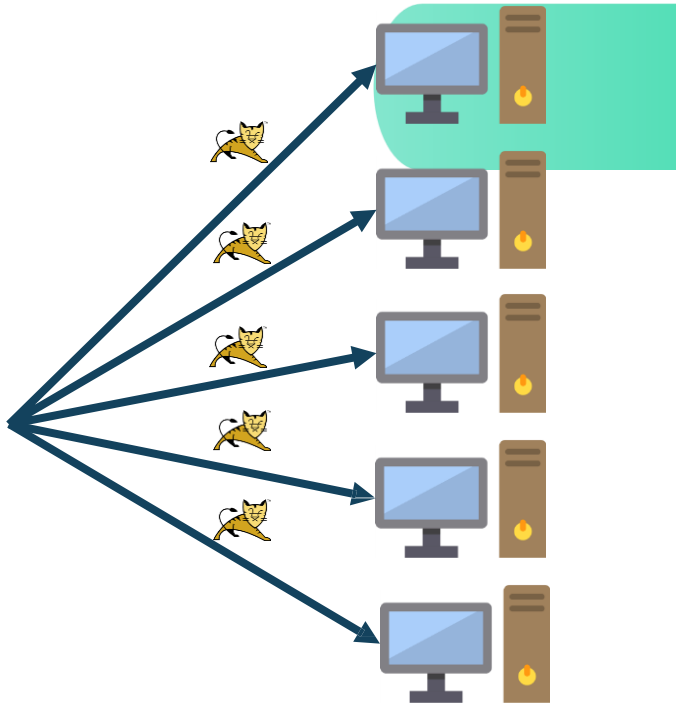
Problem Statement-Solution with Ansible

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Instead of going to each system, manually updating, Josh can use Ansible to automate the installation using Ansible Playbooks

Configuration Management

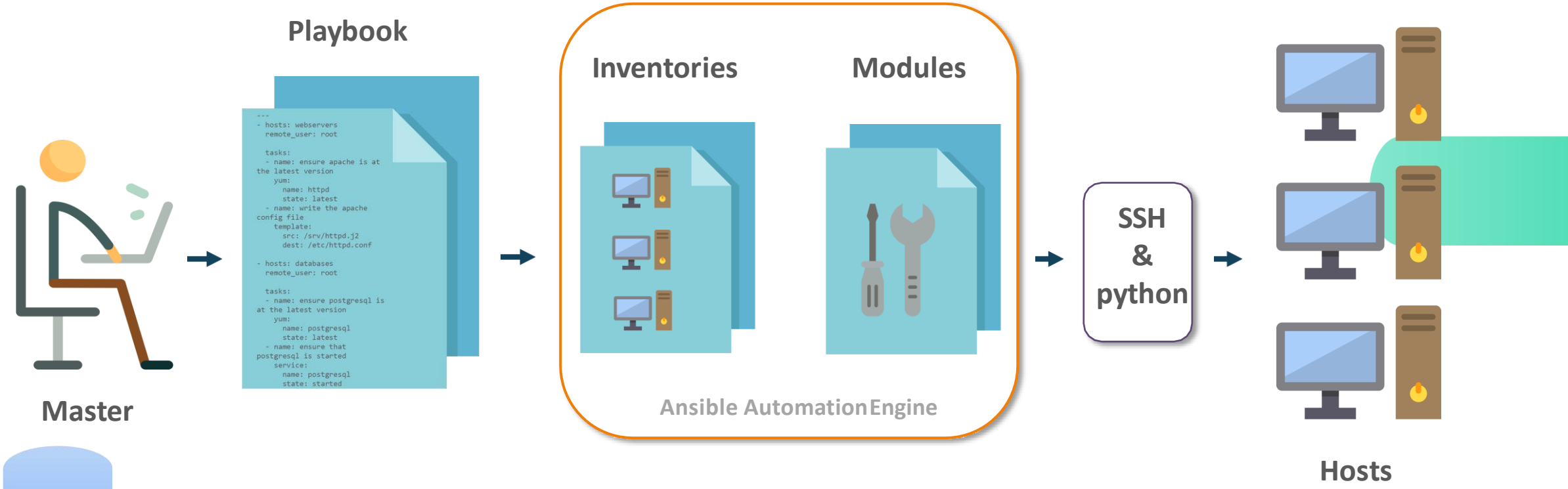




CLOUD TRAIN
ACCELERATE YOUR GROWTH

Ansible Architecture

Ansible Architecture



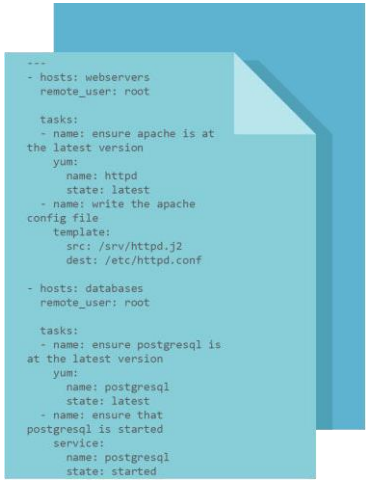
Basic Ansible Architecture

Ansible Architecture- Master

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Playbook

Master



```
---
- hosts: webservers
  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

- hosts: databases
  remote_user: root

  tasks:
  - name: ensure postgresql is
at the latest version
    yum:
      name: postgresql
      state: latest
  - name: ensure that
postgresql is started
    service:
      name: postgresql
      state: started
```

- Describes the tasks to be executed
- Written in simple language
- Playbooks are like instruction manuals

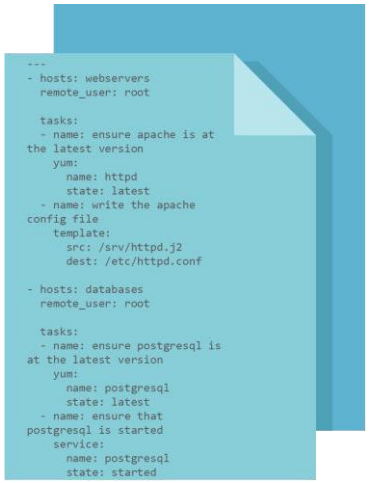
Ansible Architecture- Inventories

CLOUD TRAIN
ACCELERATE YOUR GROWTH

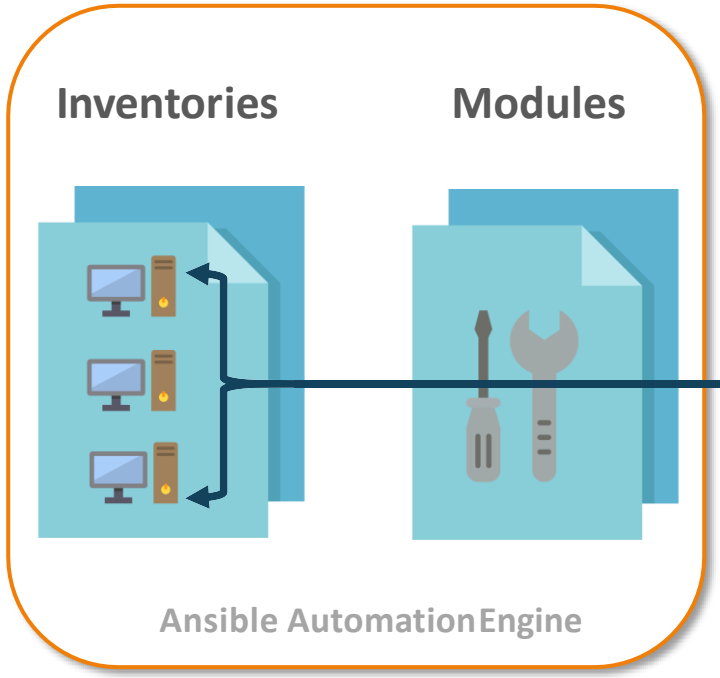
Play

Master

Playbook



```
---
- hosts: webservers
  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

- hosts: databases
  remote_user: root

  tasks:
  - name: ensure postgresql is
at the latest version
    yum:
      name: postgresql
      state: latest
  - name: ensure that
postgresql is started
    service:
      name: postgresql
      state: started
```



- List of hosts
- Where playbook tasks will be operated

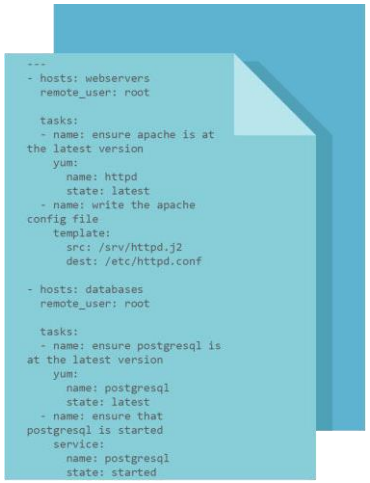
Ansible Architecture- Modules

CLOUD TRAIN
ACCELERATE YOUR GROWTH

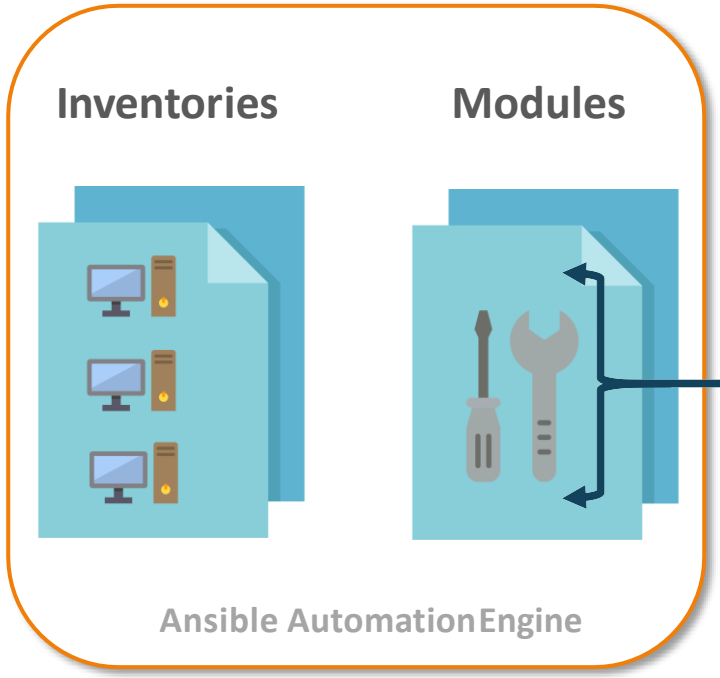
Play

Master

Playbook

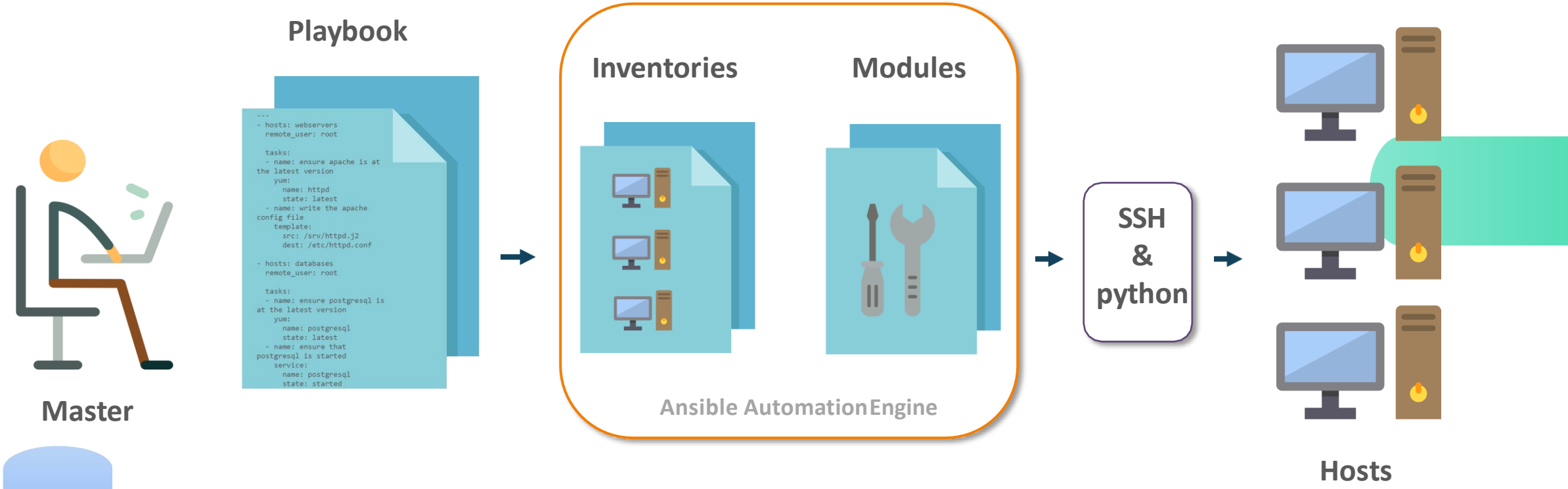


```
---
- hosts: webservers
  remote_user: root

  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

- hosts: databases
  remote_user: root

  tasks:
  - name: ensure postgresql is
at the latest version
    yum:
      name: postgresql
      state: latest
  - name: ensure that
postgresql is started
    service:
      name: postgresql
      state: started
```



- Modules are like tools
- Can control system resources, like services, packages etc.
- 500+ core modules
- Also allows custom

Ansible Architecture- Hosts



CLOUD TRAIN
ACCELERATE YOUR GROWTH

Case Study:

Ansible being used in NASA

Case Study- Business Challenge

CLOUD TRAIN
ACCELERATE YOUR GROWTH

NASA needed to move roughly 65+ applications from a Traditional Hardware Based Data Center to Cloud Based Environment for better agility and cost saving

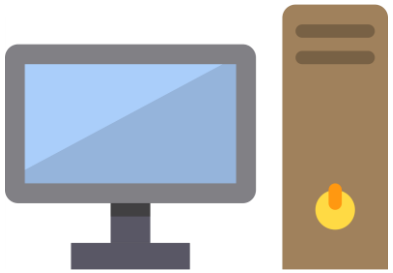
Traditional Hardware Based Data Center

Cloud Based Environment

Case Study- Solution

CLOUD TRAIN
ACCELERATE YOUR GROWTH

NASA used Ansible to manage and schedule the cloud environment



ANSIBLE

Traditional Hardware Based Data Center

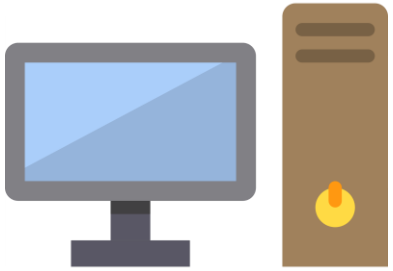
Cloud Based Environment

NASA

Case Study- Results

CLOUD TRAIN
ACCELERATE YOUR GROWTH

- Could provide better operations and security to its clients
- Increased team efficiency
- Patching updates went from a multi-day process to 45 minutes



ANSIBLE

Traditional Hardware Based Data Center

Cloud Based Environment

NASA

Installing Ansible

Installing Ansible

CLOUD TRAIN
ACCELERATE YOUR GROWTH

1 Install Ansible on Master

2 Configure SSH access to Ansible Host

3 Setting up Ansible Host and testing connection

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Ansible

CommandsLine

Ansible Command Line

CLOUD TRAIN
ACCELERATE YOUR GROWTH

ansible -m <module> -a <args> <host>

Examples:

```
ansible -m setup -a "filter=ansible_processor_vcpus" slave
ansible -m shell -a "ls -ltr /usr" slave
ansible -m shell -a "touch test.txt" slave
ansible -m shell -a "touch /root/test.txt" slave
ansible -m shell -a "touch /root/test.txt" slave -b
ansible -m shell -a "touch /root/test.txt" slave -u test
ansible -m shell -a "touch /root/test.txt" slave -u test --ask-pass
ansible -m file -a "path=myfile state=touch" slave
ansible -m apt -a "name=vim state=latest" slave
ansible -m file -a "path=myfile owner=root" slave
ansible -m file -a "path=myfile owner=root" slave  -b
ansible -m file -a "path=myfile owner=ubuntu" slave  -b
```

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Creating Ansible Playbooks

What is Ansible Playbook?

CLOUD TRAIN
ACCELERATE YOUR GROWTH

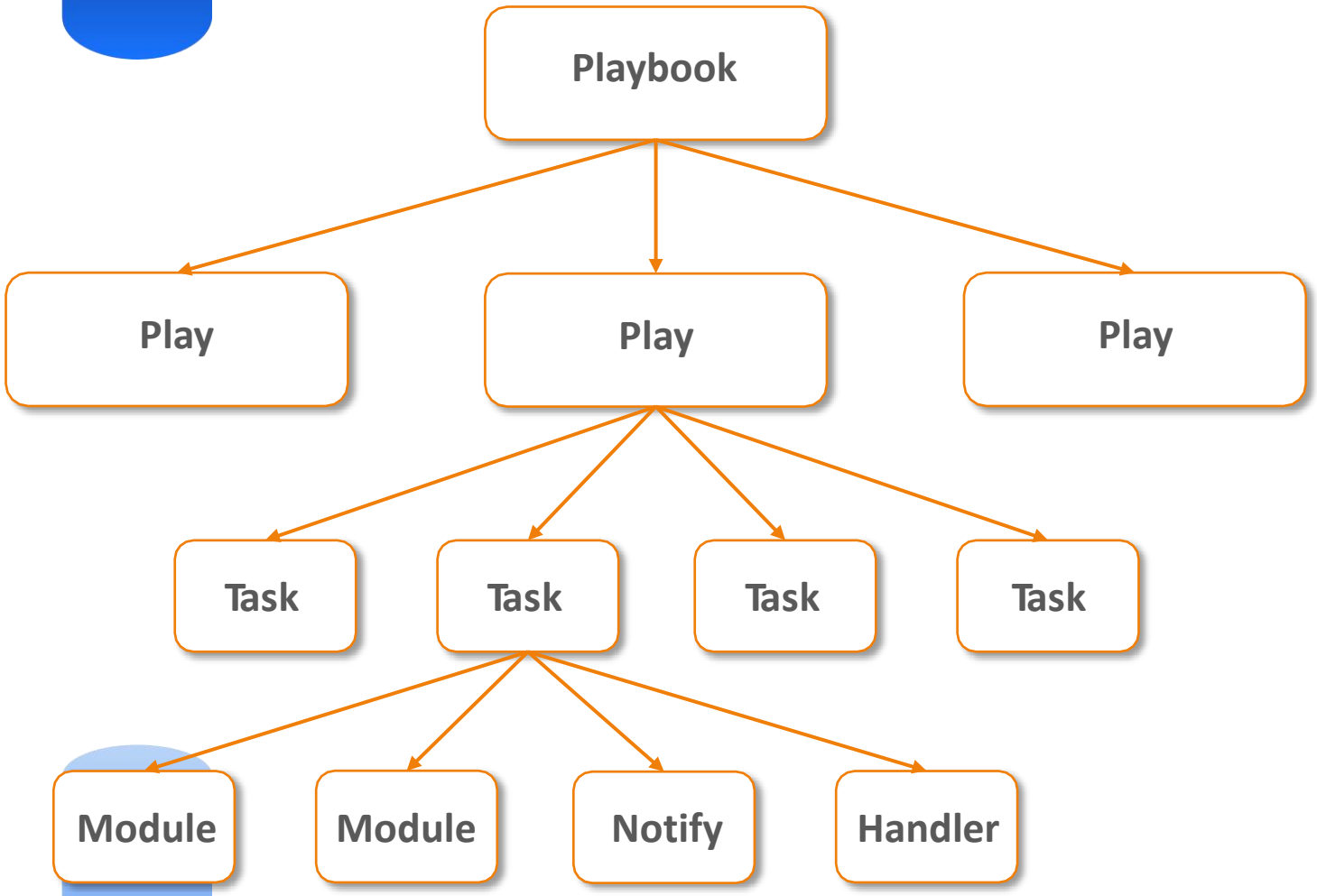
An organized unit of scripts
Defines work for a server configuration
Written in **YAML**

Ansible Playbook

YAML Ain't Markup Language

Ansible Playbook Structure

CLOUD TRAIN
ACCELERATE YOUR GROWTH



Playbook have number of **plays**

Play contains **tasks**

Tasks calls core or custom **modules**

Handler gets triggered from **notify** and executed at the end only once.

Ansible Playbook

Creating Ansible Playbook-Example

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Say, we want to create a playbook with two plays with following tasks

1 Execute a command in host1

2 Execute a script in host1

Play1

3 Execute a script in host2

4 Install nginx in host2

Play2

Creating Ansible Playbook-Example

CLOUD TRAIN
ACCELERATE YOUR GROWTH

```
---

- hosts: host1
  sudo: yes
  name: Play 1
  tasks:
     - name: Execute command 'Date'
       command: date
     - name: Execute script on server
       script: test_script.sh


- hosts: host2
  name: Play 2
  sudo: yes
  tasks:
     - name: Execute script on server
       script: test_script.sh
     - name: Install nginx
       apt: name=nginx state=latest
```

Say we want to create a playbook with two plays with following tasks

1. Execute a command in host1
2. Execute a script in host1
3. Execute a script in host2
4. Install nginx in host2

Creating Ansible Playbook-Example

CLOUD TRAIN
ACCELERATE YOUR GROWTH

```
---
- hosts: host1
  sudo: yes
  name: Play 1
  tasks:
     - name: Execute command 'Date'
       command: date
     - name: Execute script on server
       script: test_script.sh
- hosts: host2
  name: Play 2
  sudo: yes
  tasks:
     - name: Execute script on server
       script: test_script.sh
     - name: Install nginx
       apt: name=nginx state=latest
```

Play 1

Play 2

Start YAML file with [---]

Creating Ansible Playbook-Example

CLOUD TRAIN
ACCELERATE YOUR GROWTH

```
---
- hosts: host1
  sudo: yes
  name: Play 1
  tasks:
     - name: Execute command 'Date'
       command: date
     - name: Execute script on server
       script: test_script.sh

- hosts: host2
  name: Play 2
  sudo: yes
  tasks:
     - name: Execute script on server
       script: test_script.sh
     - name: Install nginx
       apt: name=nginx state=latest
```

Play 1

Play 2

- Start YAML file with [---]
- [-] Indicates an item in the list

Creating Ansible Playbook-Example

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Play 1

```
---
- hosts: host1
  sudo: yes
  name: Play 1
  tasks:
     - name: Execute command 'Date'
       command: date
     - name: Execute script on server
       script: test_script.sh
```

Play 2

```
- hosts: host2
  name: Play 2
  sudo: yes
  tasks:
     - name: Execute script on server
       script: test_script.sh
     - name: Install nginx
       apt: name=nginx state=latest
```

★ "hosts" can have one host or group of hosts from the inventory file /etc/ansible/hosts

Creating Ansible Playbook-Example

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Play 1

Play 2

```
---
- hosts: host1
  sudo: yes
  name: Play 1
  tasks:
     - name: Execute command 'Date'
       command: date
     - name: Execute script on server
       script: test_script.sh

- hosts: host2
  name: Play 2
  sudo: yes
  tasks:
     - name: Execute script on server
       script: test_script.sh
     - name: Install nginx
       apt: name=nginx state=latest
```

- "hosts" can have one host or group of hosts from the inventory file

- Each play is like a dictionary and has name, hosts, tasks. Order doesn't matter

Creating Ansible Playbook-Example

CLOUD TRAIN
ACCELERATE YOUR GROWTH

```
---
- hosts: host1
  sudo: yes
  name: Play 1
  tasks:
     - name: Execute command 'Date'
       command: date
     - name: Execute script on server
       script: test_script.sh

- hosts: host2
  name: Play 2
  sudo: yes
  tasks:
     - name: Execute script on server
       script: test_script.sh
     - name: Install nginx
       apt: name=nginx state=latest
```

Play 1

Play 2

- "hosts" can have one host or group of hosts from the inventory file
- Each play is like a dictionary and has name, hosts, tasks. Order doesn't matter
- So the playbook is a list of dictionaries

Creating Ansible Playbook-Example

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Play 1

```
---
- hosts: host1
  sudo: yes
  name: Play 1
  tasks:
    - name: Execute command 'Date'
      command: date
    - name: Execute script on server
      script: test_script.sh
```

Play 2

```
- hosts: host2
  name: Play 2
  sudo: yes
  tasks:
     - name: Execute script on server
       script: test_script.sh
     - name: Install nginx
       apt: name=nginx state=latest
```

- Similarly tasks are nothing but lists
  Denoted by [-]
- For tasks ordered collection.
  Position of entry matters
- First entry gets performed first

Creating Ansible Playbook-Example

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Create first_playbook.yml using
sudo vim <playbookname>

```
---
- hosts: host1
  become: yes
  name: Play 1
  tasks:
     - name: Execute command 'Date'
       command: date
     - name: Execute script on server
       script: test_script.sh
- hosts: host2
  name: Play 2
  become: yes
  tasks:
     - name: Execute script on server
       script: test_script.sh
     - name: Install nginx
       apt: name=nginx state=latest
```

Creating Ansible Playbook-Example

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Create test_script.sh using
sudo vim <file_name>

```
#!/bin/bash
echo "HELLO WORLD"
~
```

Creating Ansible Playbook-Example

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Syntax-check and execute ansible playbook using
ansible-playbook <playbook> --syntax-check and
ansible-playbook <playbook>

```
ubuntu@instance-1:~$ ansible-playbook playbook1.yml --syntax-check
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'
[WARNING]: Could not match supplied host pattern, ignoring: host2

playbook: playbook1.yml
```

```
ubuntu@instance-1:~$ ansible-playbook playbook1.yml

PLAY [Play 1] *****************************************************************************************

TASK [Gathering Facts] ********************************************************************************
ok: [localhost]

TASK [Execute command 'Date'] *************************************************************************
changed: [localhost]

TASK [Execute script on server] ***********************************************************************
changed: [localhost]

PLAY [Play 2] *****************************************************************************************

TASK [Gathering Facts] ********************************************************************************
ok: [slave1]

TASK [Execute script on server] ***********************************************************************
changed: [slave1]

TASK [Install nginx] **********************************************************************************
changed: [slave1]

PLAY RECAP ********************************************************************************************
localhost                  : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
slave1                     : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Ansible Roles

What is Ansible Roles?

CLOUD TRAIN
ACCELERATE YOUR GROWTH

An ansible role is group of tasks, files, and handlers stored in a standardized file structure.
Roles are small functionalities which can be used independently used but only within playbook

Ansible Playbook

Ansible playbook organizes tasks

Ansible Roles

Ansible roles organizes playbooks

Why do we need Ansible Roles?

CLOUD TRAIN
ACCELERATE YOUR GROWTH

- Roles simplifies writing complex playbooks
- Roles allows you to reuse common configuration steps between different types of servers
- Roles are flexible and can be easily modified

Structure of Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

```
new_role
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

Structure of an Ansible Role

Structure of an ansible role consists of below given components

Defaults: Store data about the role, also store default variables.

Files: Store files that needs to be pushed to the remote machine.

Handlers: Tasks that get triggered from some actions.

Meta: Information about author, supported platforms and dependencies.

Structure of Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

```
new_role
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

Structure of an Ansible Role

Structure of an ansible role consists of below given components:

Tasks: Contains the main list of tasks to be executed by the role

Templates: Contains templates which can be deployed via this role.

Tests: Contains tests built for automated testing process around our role.

Vars: Stores variables with higher priority than default variables. Difficult to override.

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

1 Use the *ansible-galaxy init <role name> --offline* command to create one Ansible role

★ Remember that Ansible roles should be written inside */etc/ansible/roles/*

```
ubuntu@instance-1:~$ ansible-galaxy init apache_demo
- Role apache_demo was created successfully
ubuntu@instance-1:~$
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

2 Install tree package using *sudo apt install tree*. Use tree command to view structure of the role

★ Use *tree <role name>* to see the role structure

```
ubuntu@instance-1:~$ sudo apt install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.
Need to get 40.6 kB of archives.
After this operation, 138 kB of additional disk space will be used.
Get:1 http://us-central1.gce.archive.ubuntu.com/ubuntu xenial/universe amd64 tree amd64 1.7.0-3 [40.6 kB]
Fetched 40.6 kB in 0s (1,744 kB/s)
Selecting previously unselected package tree.
(Reading database ... 156816 files and directories currently installed.)
Preparing to unpack .../tree_1.7.0-3_amd64.deb ...
Unpacking tree (1.7.0-3) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up tree (1.7.0-3) ...
```

```
ubuntu@instance-1:~$ tree apache_demo/
apache_demo/
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

8 directories, 8 files
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

3 Go inside task folder inside apache directory. Edit **main.yml** using *sudo vim main.yml*. Make changes as shown. Save and then exit.

★ Keeping install, configure and service files separately helps us reduce complexity.

```
---
# tasks file for apache_demo
- include: install.yml
- include: configure.yml
- include: service.yml
~
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

4 Create **install.yml**, **configure.yml** and **service.yml** to include in the **main.yml**

★ To install apache2 in the remote machine

```
---
- name: install apache2
  apt: name=apache2 update_cache=yes state=latest
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

4 Create **install.yml**, **configure.yml** and **service.yml** to include in the **main.yml**

★ To configure the apache2.conf file and to send copy.html file to the remote machine. Add notify too, based on which handlers will get triggered

```
---
#configure apache2.conf and send copy.html file
  - name: apache2.conf file
    copy: src=apache2.conf dest=/etc/apache2/
    notify:
     - restart apache2 service

  - name: send copy.html file
    copy: src=copy.html dest=/home/ubuntu/
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

4 Create **install.yml**, **configure.yml** and **service.yml** to include in the **main.yml**

★ To start apache2 service in the remote machine

```
---
- name: starting apache2 service
  service: name=apache2 state=started
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

5 Now go inside files. Store the files that needs to be pushed to the remote machine

★ Copy the apache2.conf file and create one html file

```
ubuntu@instance-1:~/apache_demo$ ls -ltr files
total 12
-rw-r--r-- 1 ubuntu ubuntu 7224 Feb 12 10:57 apache2.conf
-rw-rw-r-- 1 ubuntu ubuntu   98 Feb 12 10:59 copy.html
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

6 Go inside handlers and add the action that needs to be performed after notify from configure.yml is executed.

★ Once the notify gets executed restart the apache2 service

```
---
# handlers file for apache_demo
- name: restart apache2 service
  service: name=apache2 state=restarted
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Remember that notify name and handler name should match.

```
---
#configure apache2.conf and send copy.html file
  - name: apache2.conf file
    copy: src=apache2.conf dest=/etc/apache2/
    notify:
      - restart apache2 service

  - name: send copy.html file
    copy: src=copy.html dest=/home/ubuntu/
```

```
---
# handlers file for apache demo
- name: restart apache2 service
  service: name=apache2 state=restarted
```

IMPORTANT

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

7 Go inside meta and add information related to the role

★ Add author information, role descriptions, company information etc.

```
galaxy_info:
  author: DevOps
  description: DevOps role description
  company: Tutorial
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Structure of the role after adding all the required files

```
apache
├── README.md
├── defaults
│   └── main.yml
├── files
│   ├── apache2.conf
│   └── copy.html
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   ├── configure.yml
│   ├── install.yml
│   ├── main.yml
│   └── service.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

8 Go to the */etc/ansible/* and create one top level file where we can add hosts and roles to be executed

★ Execute *apache role* on the hosts that is under the group name *servers,* added in the inventory file */etc/ansible/hosts*

```
---
- hosts: servers
  become: yes
  roles:
    - apache_demo
~
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

9 Before we execute our top level yml file we will check for syntax errors.

★ Use ansible-playbook <filename.yml> --syntax-check

```
ubuntu@instance-1:~$ ansible-playbook masterplay.yml --syntax-check
[WARNING]: Could not match supplied host pattern, ignoring: servers

playbook: masterplay.yml
ubuntu@instance-1:~$
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

10 Execute the top level yml file

★ Use ansible-playbook <filename.yml>

```
ubuntu@instance-1:~$ ansible-playbook masterplay.yml

PLAY [servers] ***********************************************************************************************************

TASK [Gathering Facts] ***************************************************************************************************
[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host instance-1 should use /usr/bin/python3, but is using /usr/bin/python fo
release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_a
 be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [instance-1]
ok: [slave1]

TASK [apache_demo : install apache2] *************************************************************************************
changed: [instance-1]
changed: [slave1]

TASK [apache_demo : apache2.conf file] ***********************************************************************************
ok: [instance-1]
ok: [slave1]

TASK [apache_demo : send copy.html file] *********************************************************************************
changed: [instance-1]
changed: [slave1]

TASK [apache_demo : starting apache2 service] ****************************************************************************
ok: [instance-1]
ok: [slave1]

PLAY RECAP ***************************************************************************************************************
instance-1                 : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
slave1                     : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Creating an Ansible Role

CLOUD TRAIN
ACCELERATE YOUR GROWTH

11 Test the Webpage in Browser

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Using Roles in Playbook

Using Roles in Playbook

CLOUD TRAIN
ACCELERATE YOUR GROWTH

- To use ansible roles along with other tasks in playbook Use *import_role* and *include_role.*

- Here we have created one playbook called *playbookrole.yml* to execute on *servers* along with two *debug* tasks before and after *apache* role.

ubuntu@ip-172-31-40-83: /etc/ansible

```
  GNU nano 2.9.3                                        playbookrole.yml

---
  - hosts: servers
    sudo: yes
    tasks:
    - debug:
        msg: "before we run our role"
    - import_role:
        name: apache
    - include_role:
        name: apache
    - debug:
        msg: "after we ran our role"
```

Using Roles in Playbook

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Check for syntax error and execute the playbook with roles.

ubuntu@ip-172-31-40-83: /etc/ansible

```
ubuntu@ip-172-31-40-83:/etc/ansible$ ansible-playbook playbookrole.yml --syntax-check

playbook: playbookrole.yml
```

ubuntu@ip-172-31-40-83: /etc/ansible

```
ubuntu@ip-172-31-40-83:/etc/ansible$ ansible-playbook playbookrole.yml
PLAY [servers] ***********************************************************

TASK [Gathering Facts] ***************************************************
ok: [host1]
ok: [host2]

TASK [debug] *************************************************************
ok: [host1] => {
    "msg": "before we run our role"
}
ok: [host2] => {
    "msg": "before we run our role"
}

TASK [apache : install apache2] ******************************************
ok: [host1]
ok: [host2]

TASK [apache : apache2.conf file] ****************************************
ok: [host1]
ok: [host2]

TASK [apache : send copy.html file] **************************************
ok: [host1]
ok: [host2]

TASK [apache : starting apache2 service] *********************************
ok: [host1]
ok: [host2]
```

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Hands-on: Configuring Multiple Nodes using Ansible

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Got queries or need more info?

Contact us

TO ACCELERATE YOUR CAREER GROWTH

For questions and more details:

please call @ +91 98712 72900 or

visit https://www.thecloudtrain.com/ or

email at join@thecloudtrain.com or

WhatsApp us >>