

# ECE530: Cloud Computing

## Ansible

Suyog Joshi – 101846426 – [sjoshi@unm.edu](mailto:sjoshi@unm.edu)

Ansible is a straightforward IT automation tool that simplifies tasks like cloud provisioning, configuration management, application deployment, and intra-service orchestration. It was designed from the ground up for multi-tier deployments, allowing you to model your IT infrastructure based on the relationships between systems rather than managing them individually. Ansible doesn't require agents or custom security infrastructure, making it easy to set up. It uses a simple language (YAML in Ansible Playbooks) that allows you to write automation tasks in near-plain English.

The key features of ansible are as follows:

1) Configuration Management:

Ansible is built to provide a straightforward, dependable, and consistent approach to configuration management. For instance, it can modify application, operating system, or device settings; manage service start and stop actions; install or update software; enforce security policies; and handle a broad range of other configuration tasks.

2) Application Deployment:

Ansible streamlines the DevOps process by automating the deployment of in-house applications across your environment. It enables rapid, effortless multi-tier application deployment. Instead of coding custom automation solutions, you simply define the necessary tasks in a playbook, and Ansible determines how to bring your systems to the desired state. This eliminates the need to manually configure applications on each machine.

3) Orchestration:

In application deployment, you must oversee front-end and back-end services, databases, networks, storage, and more, ensuring everything is executed in the correct sequence. Ansible simplifies this orchestration by utilizing automated workflows and provisioning. Once you've defined your infrastructure with Ansible playbooks, you can replicate that orchestration wherever it's required.

4) Security and Compliance:

Similar to application deployment, site-wide security policies (such as firewall rules) can be enforced through automated processes. By setting up security configurations on the control machine and executing the corresponding playbook, all remote hosts will be automatically updated with those settings. This eliminates the need to manually check each machine for security compliance.

5) Cloud Provisioning:

Ansible allows you to provision cloud platforms, virtual hosts, network devices, and bare-metal servers.

For this homework, we will set up Ansible on a primary node and use two secondary nodes to deploy MongoDB. We've configured three Linux/UNIX virtual machines platforms in AWS (Amazon Web Services):.

Instances (4) <a href="#">Info</a>										
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>				All states ▾		< 1 > ⓘ				
<input type="checkbox"/>	Name ↗	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IPv4 DNS	Public IPv4 ... ▾	Elastic IP
<input type="checkbox"/>	Ansible_Target1	i-075450185590145e4	Running ⓘ ⓘ	t2.micro	2/2 checks passed View alarms +		us-east-2c	ec2-18-188-178-86.us-...	18.188.178.86	-
<input type="checkbox"/>	Ansible_Target3	i-0b8064c441f159973	Running ⓘ ⓘ	t2.micro	2/2 checks passed View alarms +		us-east-2c	ec2-52-14-6-89.us-east...	52.14.6.89	-
<input type="checkbox"/>	Ansible_Target2	i-00aa04a289b71d172	Running ⓘ ⓘ	t2.micro	2/2 checks passed View alarms +		us-east-2c	ec2-18-219-110-86.us-...	18.219.110.86	-
<input type="checkbox"/>	Ansible_Master	i-03482d45dace55668	Running ⓘ ⓘ	t2.micro	2/2 checks passed View alarms +		us-east-2c	ec2-18-221-114-177.us...	18.221.114.177	-

Setting up ansible:

```
[ec2-user@ip-172-31-37-133 ~]$ ansible --version
ansible [core 2.15.3]
  config file = /home/ec2-user/ansible.cfg
  configured module search path = ['/home/ec2-user/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible
  ansible collection location = /home/ec2-user/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.9.16 (main, Mar 28 2024, 00:00:00) [GCC 11.4.1 20230605 (Red Hat 11.4.1-2)] (/usr/bin/python3.9)
  jinja version = 3.1.2
  libyaml = True
[ec2-user@ip-172-31-37-133 ~]$
```

Public key share:

Generated a public key (i.e. id\_rsa.pub) and shared it with the other nodes to setup ssh as ansible connects to nodes using ssh.

```
[ec2-user@ip-172-31-37-133 ~]$ cd ~/.ssh/
[ec2-user@ip-172-31-37-133 .ssh]$ ls
ansible-keypair.pem  authorized_keys  id_rsa  id_rsa.pub  known_hosts  known_hosts.old
[ec2-user@ip-172-31-37-133 .ssh]$
```

The public key inside id\_rsa.pub

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDdI1hVd1kPQSVK9HmK39jkyekdX5sfijdTnpXr1wg+3Iz9SsHjVMSQdBy
3udeM9CBY9kLVtd0HaHJo4bAwTxFHbmiU+6ZpNX9/3iVSDEiz1NyC9Sf9UjfeTIs40VoyTKK0wDJ/3z4a2urYbzn2LGT4V
HT0jyKVtSSujikNbVWHxiqb7dX5hnP7GoyLGmWjzvEQ8Qb4uSrrQ2W+Zyx/ACJ1R8NLeFXwnWwwqaZ23ICbXW0CvagtzIRY
KGhxy/LdlQgX1pvfymzu/8jdIMSyykTyMV0vZrd5jRLV4yXwVA9aVzY7kizbTmIBqr/n6ESTzfzLizJtVILWD3CeI4WEr0Q
zHw9csW00DzYBCXuNsAWmcabKTBAbxD+CwI1BeKzBY3xaxpRPo99o+BQYpKM1SUxXxa3q+OXnHq0p8RziQ832Y+2uNhaQyV
Xg5phUqcyQoQpTfcaDegjdmLS+R6SXd9AyA4+8nCT9R7E0FLs0KJ44dsh0fSvAz5dY3kiko/gT6s= ec2-user@ip-172-3
1-37-133.us-east-2.compute.internal
```

Created the ansible config file (i.e., ansible.cfg)

```
[defaults]
inventory = inventory.txt
~
~
```

Setup the servers, define the primary nodes, and secondary nodes

```
[servers]
ansible-target-1 ansible_host=18.188.178.86 ansible_connection=ssh ansible_user=ec2-user
ansible-target-2 ansible_host=18.219.110.86 ansible_connection=ssh ansible_user=ec2-user
ansible-target-3 ansible_host=52.14.6.89 ansible_connection=ssh ansible_user=ec2-user

[primarynode]
primarynode ansible_host=18.188.178.86 ansible_connection=ssh ansible_user=ec2-user

[secondarynode]
ansible-target-2 ansible_host=18.219.110.86 ansible_connection=ssh ansible_user=ec2-user
ansible-target-3 ansible_host=52.14.6.89 ansible_connection=ssh ansible_user=ec2-user
```

### 1) First Playbook

```
---
- name: FirstPlaybook
  hosts: servers
  tasks:
    - name: Create file
      copy:
        dest: teammate_names.txt
        content: "Atal Pandey, Suyog Joshi"
```

## Running the firstplaybook.yml

```
[ec2-user@ip-172-31-37-133 ~]$ ansible-playbook firstplaybook.yml
[WARNING]: Found both group and host with same name: primarynode

PLAY [FirswtPlaybook] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host ansible-target-2 is using the discovered Python interpreter
at /usr/bin/python3.9, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [ansible-target-2]
[WARNING]: Platform linux on host ansible-target-3 is using the discovered Python interpreter
at /usr/bin/python3.9, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [ansible-target-3]
[WARNING]: Platform linux on host ansible-target-1 is using the discovered Python interpreter
at /usr/bin/python3.9, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [ansible-target-1]

TASK [Create file] *****
changed: [ansible-target-2]
changed: [ansible-target-3]
changed: [ansible-target-1]

PLAY RECAP *****
ansible-target-1      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    res
cued=0    ignored=0
ansible-target-2      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    res
cued=0    ignored=0
ansible-target-3      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    res
cued=0    ignored=0

[ec2-user@ip-172-31-37-133 ~]$
```

Ssh to ansible-target-1 ( 18.188.178.86 ) ansibe-target-2 ( 52.14.6.89 ) and ansible-target-3 ( 18.219.110.86 ) and verified the content with the team mates names.

ansible-target-1 ( 18.188.178.86 )

```
[ec2-user@ip-172-31-37-133 ~]$ ssh 18.188.178.86
,      #_
~\ _   #####_      Amazon Linux 2023
~~  \_#####\
~~   \###|
~~   \#/  ___  https://aws.amazon.com/linux/amazon-linux-2023
~~      V~'  '->
~~~
~~. _ .  _/
  _/  _/
    _/m/'

Last login: Tue May  7 04:06:45 2024 from 18.221.114.177
[ec2-user@ip-172-31-39-49 ~]$ ls
teammate_names.txt
[ec2-user@ip-172-31-39-49 ~]$ cat teammate_names.txt
Atal Pandey, Suyog Joshi[ec2-user@ip-172-31-39-49 ~]$
```

ansibe-target-2 ( 52.14.6.89 )

```
[ec2-user@ip-172-31-37-133 ~]$ vim firstplaybook.yml
[ec2-user@ip-172-31-37-133 ~]$ ssh 52.14.6.89
,      #_
~\ _   #####_      Amazon Linux 2023
~~  \_#####\
~~   \###|
~~   \#/  ___  https://aws.amazon.com/linux/amazon-linux-2023
~~      V~'  '->
~~~
~~. _ .  _/
  _/  _/
    _/m/'

Last login: Tue May  7 04:06:45 2024 from 18.221.114.177
[ec2-user@ip-172-31-43-107 ~]$ ls
teammate_names.txt
[ec2-user@ip-172-31-43-107 ~]$ vim teammate_names.txt
[ec2-user@ip-172-31-43-107 ~]$ cat teammate_names.txt
Atal Pandey, Suyog Joshi[ec2-user@ip-172-31-43-107 ~]$
```

ansible-target-3 ( 18.219.110.86 )

```
[ec2-user@ip-172-31-37-133 ~]$ ssh 18.219.110.86
,      #_
~\_    #####_      Amazon Linux 2023
~~  \_#####\
~~      \###|
~~      \#/  ___  https://aws.amazon.com/linux/amazon-linux-2023
~~      V~'  '->
~~~
~~._.  _/_/
  _/_/_/
   _/m/'

Last login: Tue May  7 04:06:45 2024 from 18.221.114.177
[ec2-user@ip-172-31-42-195 ~]$ ls
teammate_names.txt
[ec2-user@ip-172-31-42-195 ~]$ cat teammate_names.txt
Atal Pandey, Suyog Joshi[ec2-user@ip-172-31-42-195 ~]$
```

2) Create a role and provision primary mongoDB node

Create a new role (role1) inside role

```
[ec2-user@ip-172-31-37-133 roles]$ ls
role1  role2
[ec2-user@ip-172-31-37-133 roles]$
```

Verify role1

```
[ec2-user@ip-172-31-37-133 roles]$ ls
role1  role2
[ec2-user@ip-172-31-37-133 roles]$ tree role1
role1
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

8 directories, 8 files
[ec2-user@ip-172-31-37-133 roles]$
```



Configure execute.yml to run role1

```
[ec2-user@ip-172-31-37-133 roles]$ cd
[ec2-user@ip-172-31-37-133 ~]$ ls
ansible          executable.yml    inventory.txt     play.yml
ansible-keypair.pem  executable2.yml  mongodb-org-7.0.repo  roles
ansible.cfg       firstplaybook.yml  mongoinstall.yml
[ec2-user@ip-172-31-37-133 ~]$ cat executable.yml
---
- hosts: primarynode
  become: yes
  roles:
    - role1
[ec2-user@ip-172-31-37-133 ~]$
```

Configure the tasks in ~/role/roles1/tasks/main.yml

```
- name: Upgrade all packages
  yum:
    name: '*'
    state: latest

- name: MongoDB Repository
  yum_repository:
    name: mongodb-org-6.0
    description: MongoDB Repository Repo
    baseurl: https://repo.mongodb.org/yum/redhat/9/mongodb-org/6.0/x86_64/
    gpgkey: https://pgp.mongodb.com/server-6.0.asc

- name: Install MongoDB
  yum:
    name: mongodb-org
    state: present
    update_cache: yes

- name: Enable MongoDB to run on boot
  service:
    name: mongod
    enabled: yes
    state: started
```

Running the main.yml to provision mongoDB on primary node

```
[ec2-user@ip-172-31-37-133 ~]$ vim inventory.txt
[ec2-user@ip-172-31-37-133 ~]$ ansible-playbook executable.yml
[WARNING]: Found both group and host with same name: primarynode

PLAY [primarynode] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host primarynode is using the discovered Python interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [primarynode]

TASK [role1 : Upgrade all packages] *****
ok: [primarynode]

TASK [role1 : MongoDB Repository] *****
changed: [primarynode]

TASK [role1 : Install MongoDB] *****
changed: [primarynode]

TASK [role1 : Enable MongoDB to run on boot] *****
changed: [primarynode]

PLAY RECAP *****
primarynode      : ok=5    changed=3    unreachable=0    failed=0    skipped=0    res
cued=0    ignored=0
```

### 3) Making your Primary Node role generic to use variables

We then generalized the playbook by setting reusable variables in  
~/roles/role1/vars/main.yml

```
---
# vars file for role1
names:
  - '*'
  - mongodb-org-6.0
  - mongodb-org
  - mongod
repos:
  - 'https://repo.mongodb.org/yum/redhat/9/mongodb-org/6.0/x86_64/'
  - 'https://pgp.mongodb.com/server-6.0.asc'
```

Update the ~/roles/role1/tasks/main.yml with the variables created as shown above.

```
- name: Upgrade all packages
  yum:
    name: "{{ names[0] }}"
    state: latest

- name: MongoDB Repository
  yum_repository:
    name: "{{ names[1] }}"
    description: MongoDB Repository Repo
    baseurl: "{{ repos[0] }}"
    gpgkey: "{{ repos[1] }}"

- name: Install MongoDB
  yum:
    name: "{{ names[2] }}"
    state: present
    update_cache: yes

- name: Enable MongoDB to run on boot
  service:
    name: "{{ names[3] }}"
    enabled: yes
    state: started
```

Running the playbook with the variables after updating it.

```
[ec2-user@ip-172-31-37-133 ~]$ ansible-playbook executable.yml
[WARNING]: Found both group and host with same name: primarynode

PLAY [primarynode] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host primarynode is using the discovered Python interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [primarynode]

TASK [role1 : Upgrade all packages] *****
ok: [primarynode]

TASK [role1 : MongoDB Repository] *****
ok: [primarynode]

TASK [role1 : Install MongoDB] *****
ok: [primarynode]

TASK [role1 : Enable MongoDB to run on boot] *****
ok: [primarynode]

PLAY RECAP *****
primarynode : ok=5   changed=0    unreachable=0    failed=0    skipped=0    res
cued=0     ignored=0
```

4) Create a role that provisions at least one secondary node

Create a new role (role2)

```
[ec2-user@ip-172-31-37-133 roles]$ ls
role1  role2
[ec2-user@ip-172-31-37-133 roles]$
```

```
[ec2-user@ip-172-31-37-133 roles]$ tree role2
role2
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
│   └── mongod.conf.j2
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

8 directories, 9 files
[ec2-user@ip-172-31-37-133 roles]$
```

Create executable1.yml that runs on the nodes

```
[ec2-user@ip-172-31-37-133 ~]$ cat executable1.yml
---
- hosts: servers
  become: yes
  roles:
    - role1
[ec2-user@ip-172-31-37-133 ~]$
```

Configure the ~/roles/role2/tasks/main.yml with script to provision mongoDB and run the playbook execute1.yml

```
[ec2-user@ip-172-31-37-133 ~]$ ansible-playbook executable1.yml
[WARNING]: Found both group and host with same name: primarynode

PLAY [servers] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host ansible-target-2 is using the discovered Python interpreter
at /usr/bin/python3.9, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [ansible-target-2]
[WARNING]: Platform linux on host ansible-target-3 is using the discovered Python interpreter
at /usr/bin/python3.9, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [ansible-target-3]
[WARNING]: Platform linux on host ansible-target-1 is using the discovered Python interpreter
at /usr/bin/python3.9, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [ansible-target-1]

TASK [role1 : Upgrade all packages] *****
ok: [ansible-target-3]
ok: [ansible-target-2]
ok: [ansible-target-1]

TASK [role1 : MongoDB Repository] *****
changed: [ansible-target-2]
changed: [ansible-target-1]
changed: [ansible-target-3]

TASK [role1 : Install MongoDB] *****
ok: [ansible-target-3]
ok: [ansible-target-2]
ok: [ansible-target-1]

TASK [role1 : Enable MongoDB to run on boot] *****
ok: [ansible-target-3]
ok: [ansible-target-2]
ok: [ansible-target-1]

PLAY RECAP *****
ansible-target-1      : ok=5    changed=1    unreachable=0    failed=0    skipped=0    res
cued=0    ignored=0
ansible-target-2      : ok=5    changed=1    unreachable=0    failed=0    skipped=0    res
cued=0    ignored=0
ansible-target-3      : ok=5    changed=1    unreachable=0    failed=0    skipped=0    res
cued=0    ignored=0
```

## Verifying the mongo installation on all the nodes

```
[ec2-user@ip-172-31-39-49 ~]$ systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; preset: disabled)
   Active: active (running) since Mon 2024-05-06 23:58:21 UTC; 5h 55min ago
     Docs: https://docs.mongodb.org/manual
   Main PID: 35423 (mongod)
    Memory: 131.9M
      CPU: 1min 39.046s
    CGroup: /system.slice/mongod.service
           └─35423 /usr/bin/mongod -f /etc/mongod.conf

May 06 23:58:21 ip-172-31-39-49.us-east-2.compute.internal systemd[1]: Started mongod.service >
May 06 23:58:21 ip-172-31-39-49.us-east-2.compute.internal mongod[35423]: {"t":{"$date":"2024->
```

```
[ec2-user@ip-172-31-43-107 ~]$ systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; preset: disabled)
   Active: active (running) since Mon 2024-05-06 04:34:42 UTC; 1 day 1h ago
     Docs: https://docs.mongodb.org/manual
   Main PID: 14412 (mongod)
    Memory: 135.2M
      CPU: 5min 27.564s
    CGroup: /system.slice/mongod.service
           └─14412 /usr/bin/mongod -f /etc/mongod.conf

May 06 04:34:42 ip-172-31-43-107.us-east-2.compute.internal systemd[1]: Started mongod.service>
May 06 04:34:42 ip-172-31-43-107.us-east-2.compute.internal mongod[14412]: {"t":{"$date":"2024->
```

```
[ec2-user@ip-172-31-42-195 ~]$ systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; preset: disabled)
   Active: active (running) since Mon 2024-05-06 04:34:42 UTC; 1 day 1h ago
     Docs: https://docs.mongodb.org/manual
   Main PID: 15990 (mongod)
    Memory: 135.4M
      CPU: 5min 34.306s
    CGroup: /system.slice/mongod.service
           └─15990 /usr/bin/mongod -f /etc/mongod.conf

May 06 04:34:42 ip-172-31-42-195.us-east-2.compute.internal systemd[1]: Started mongod.service>
May 06 04:34:42 ip-172-31-42-195.us-east-2.compute.internal mongod[15990]: {"t":{"$date":"2024->
```



- 5) Create a generic mongoDB secondary node that uses variables and templates

Defining variables in group vars

```
[ec2-user@ip-172-31-37-133 role2]$ ls
README.md  defaults  files     handlers  meta      tasks    templates  tests  vars
[ec2-user@ip-172-31-37-133 role2]$ cd vars
[ec2-user@ip-172-31-37-133 vars]$ ls
main.yml
[ec2-user@ip-172-31-37-133 vars]$ cat main.yml
---
# vars file for role2
secondarynode:
  - hostname: "ansible-target-2"
    ip: "18.219.110.86"
  - hostname: "ansible-target-3"
    ip: "52.14.6.89"

[ec2-user@ip-172-31-37-133 vars]$
```

Creating a mongoDB configuration template. Create a jinja2 template that will serve as the mongoDB configuration file (mongodv.conf.j2 )  
Navigate to ~/roles/role2/templates and create the mongod.conf.j2 file

```
[ec2-user@ip-172-31-37-133 ~]$ cd roles/role2/templates/
[ec2-user@ip-172-31-37-133 templates]$ cat mongod.conf.j2
# mongod.conf
storage:
  dbPath: /var/lib/mongo
  journal:
    enabled: true

net:
  port: 27017
  bindIp: {{ item.ip }}

replication:
  replSetName: rs0

[ec2-user@ip-172-31-37-133 templates]$
```



Creating the roles task file by editing the main.yml file inside the tasks directory of the role2 to configure mongoDB using the template and variables.

Navigate to ~/roles/role2/tasks/main.yml

```
---
- name: Create MongoDB data directory
  file:
    path: /var/lib/mongo
    state: directory
    owner: mongod
    group: mongod
    mode: '0755'

- name: Create mongod.conf file from template
  template:
    src: mongod.conf.j2
    dest: /etc/mongod.conf
    owner: root
    group: root
    mode: '0644'
  loop: "{{ secondarynode }}"
  loop_control:
    loop_var: item

- name: Start MongoDB service
  service:
    name: mongod
    state: started
    enabled: yes
```

Writing the playbook executable2.yml and running it.

```
[ec2-user@ip-172-31-37-133 ~]$ cat executable2.yml
---
- hosts: secondarynode
  become: yes
  roles:
    - role2
[ec2-user@ip-172-31-37-133 ~]$
```

```
[ec2-user@ip-172-31-37-133 ~]$ ansible-playbook executable2.yml
[WARNING]: Found both group and host with same name: primarynode

PLAY [secondarynode] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host ansible-target-2 is using the discovered Python interpreter
at /usr/bin/python3.9, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [ansible-target-2]
[WARNING]: Platform linux on host ansible-target-3 is using the discovered Python interpreter
at /usr/bin/python3.9, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [ansible-target-3]

TASK [role2 : Upgrade all packages] *****
ok: [ansible-target-3]
ok: [ansible-target-2]

TASK [role2 : MongoDB Repository] *****
ok: [ansible-target-2]
ok: [ansible-target-3]

TASK [role2 : Install MongoDB] *****
ok: [ansible-target-2]
ok: [ansible-target-3]

TASK [role2 : Enable MongoDB to run on boot] *****
ok: [ansible-target-2]
ok: [ansible-target-3]

PLAY RECAP *****
ansible-target-2      : ok=5    changed=0    unreachable=0    failed=0    skipped=0    res
cued=0    ignored=0
ansible-target-3      : ok=5    changed=0    unreachable=0    failed=0    skipped=0    res
cued=0    ignored=0
```

### Conclusion:

Through this exercise, we've gained valuable insights into provisioning services with Ansible. Ansible proves invaluable for automating build and deployment tasks, streamlining infrastructure and configuration management by eliminating redundant processes. While Ansible stands as a leading choice, there are several alternatives worth exploring, including Puppet, Chef, SaltStack, and Terraform. Each offers its unique features and strengths, catering to diverse needs in the realm of automation and orchestration.

### References:

<https://docs.ansible.com/>

<https://docs.ansible.com/ansible/latest/collections/community/mongodb/index.html>

[https://docs.ansible.com/ansible/latest/playbook\\_guide/playbooks\\_reuse\\_roles.html](https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_reuse_roles.html)

