

## What is Provisioning Tool?

- helps in creating a new Virtual Machine in the onPrem infrastructure, public/private/hybrid cloud
- automating OS installation
- Examples
  - Docker, Vagrant, Cloudformation, Terraform
- Terraform also supports some minimal Configuration Management Features

## What is Container Orchestration Platform?

- helps in managing your containerized application workloads
- helps in setting up High Available application within Container Orchestration Platform
- provides inbuilt monitoring tools
- supports scale up/down depending on the user traffic to your application workloads
- helps in exposing your application to external world using external Services
- helps in restriction your application access to only within the cluster using internal service
- self-healing platform
- Examples
  - Docker SWARM
  - Google Kubernetes
  - Red Hat OpenShift

## What is Configuration Management Tool?

- helps in automating software installation and configuration on a existing OS/Virtual machine in OnPrem servers, public/private/hybrid cloud environments
- also helps create users with specific access
- Ansible also supports some minimal provisioning features

## Provisioner vs Configuration Management Tool

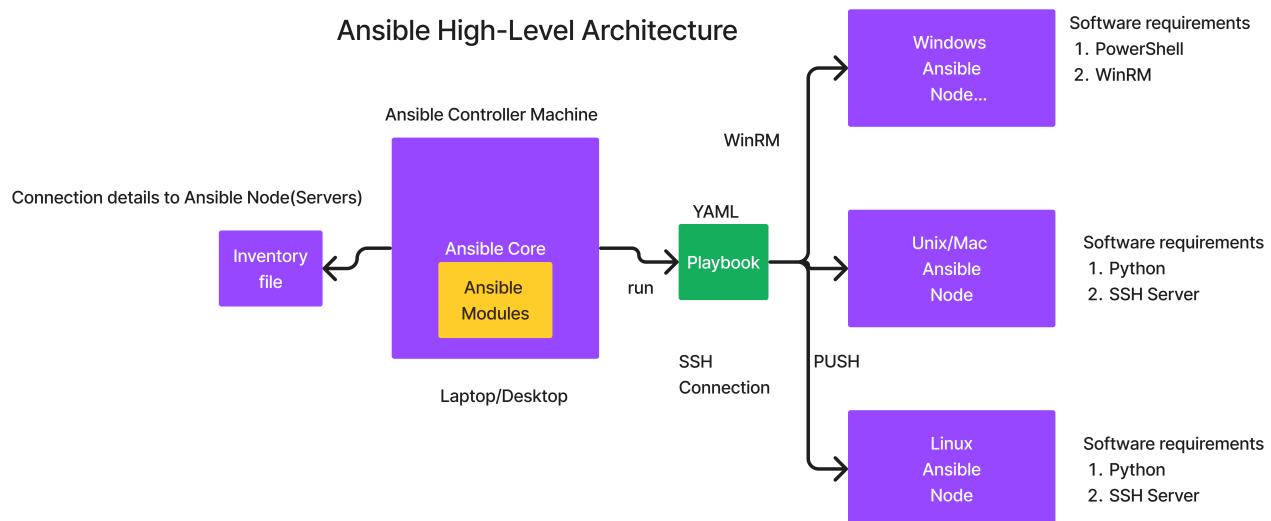
- The strength of Provisioner tools is in creating a new Virtual Machine in OnPrem/Cloud environments
- Provisioner also supports some basic configuration management features, but it doesn't or can't replace the Configuration Management tools
- Generally provisioning tools like Terraform creates the Virtual Machine and then invoke Ansible to further configure the machine

## Ansible Overview

- is a Configuration Management tool
- developed by Michael Deehan in Python language
- helps in automating software installations/configuration typically automating any administrative activities on a existing OS/Virtual machine
- comes in 3 flavours
  1. Ansible Core - opensource, supports command-line interface only, can be installed only in Linux/Mac

2. AWX - opensource, supports Web Interface, playbook can be executed but not create, developed on top of Ansible Core
  3. Red Hat Ansible Tower - developed on top of AWX, hence supports Web Interface
- Domain Specific Language(DSL)
    - the language in which the automation code is written
    - DSL used by Ansible is YAML (Yet Another Markup Language)

## Ansible High Level Architecture



## Ansible Alternate Tools

- Puppet
- Chef
- Salt/SaltStack

## Installing Ansible

[https://docs.ansible.com/ansible/latest/installation\\_guide/installation\\_distro.html](https://docs.ansible.com/ansible/latest/installation_guide/installation_distro.html)

## Ansible Commands

---

### Lab - Cloning TekTutor Training Repository

```
cd ~
git clone https://github.com/tektutor/ansible-sep-2023.git
cd ansible-sep-2023
```

## Lab - Finding Ansible version

```
ansible --version
```

Expected output

```
└──(jegan@tektutor.org)-[~/ansible-sep-2023/Day2]
└─$ ansible --version
ansible [core 2.14.9]
  config file = None
  configured module search path = ['/home/jegan/.ansible/plugins/modules',
  '/usr/share/ansible/plugins/modules']
    ansible python module location = /usr/lib/python3/dist-packages/ansible
    ansible collection location =
/home/jegan/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0]
(/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
```

## Lab - Building Custom Docker Image

We need to create key pair as shown below with default options

```
ssh-keygen
```

## Expected output

The screenshot shows a terminal window with three tabs, all titled 'jegan@tektutor: ~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu'. The current tab is active and displays the following command and its output:

```
(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jegan/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jegan/.ssh/id_rsa
Your public key has been saved in /home/jegan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:0X7UnfH4/ewUz8JeM4eTlqv4lJl9AIDe9h0MD2zixfA jegan@tektutor
The key's randomart image is:
+---[RSA 3072]---+
|       .o+   |
|     . .oB    |
|   . o +E=.  |
| ..+. o+=   |
| S.....=o..  |
| . o   .* .B+ |
| .. = XoX    |
| . o o O=    |
| ..o.o.+    |
+---[SHA256]---+
(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$
```

## Lab - Building custom ubuntu ansible docker image

```
cd ~/ansible-sep-2023
git pull
cd Day2/CustomDockerImages/ubuntu
cp ~/.ssh/id_rsa.pub authorized_keys

docker build -t tektutor/ansible-ubuntu-node:latest .
```

## Expected output

```
jegan@tektutor: ~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu
$ cp ~/.ssh/id_rsa.pub authorized_keys

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$ ls
authorized_keys Dockerfile

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$ cat Dockerfile
FROM ubuntu:16.04
MAINTAINER Jeganathan Swaminathan <jegan@tektutor.org>

RUN apt-get update && apt-get install -y openssh-server python3
RUN mkdir /var/run/sshd
RUN echo 'root:root' | chpasswd
RUN sed -i 's/PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config

# SSH login fix. Otherwise user is kicked off after login
RUN sed 's@session\s*required\s*pam_loginuid.so@session optional pam_loginuid.so@g' -i /etc/pam.d/sshd

RUN mkdir -p /root/.ssh
COPY authorized_keys /root/.ssh/authorized_keys

EXPOSE 22
EXPOSE 80
CMD ["/usr/sbin/sshd", "-D"]

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$ chpasswd
^C

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$
```

```
jegan@tektutor:~/ansible-sep-2023          jegan@tektutor:~/ansible-sep-2023          jegan@tektutor:~/ansible-sep-2023
----> Running in 4e779a52c005
Removing intermediate container 4e779a52c005
----> 6b899ee638a1f
Step 6/12 : RUN sed -i 's/PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
----> Running in 74120164adc2
Removing intermediate container 74120164adc2
----> e7b2ff6d80a7
Step 7/12 : RUN sed 's@session\s*required\s*pam_loginuid.so@session optional pam_loginuid.so@g' -i /etc/pam.d/sshd
----> Running in 06103c28e3a3
Removing intermediate container 06103c28e3a3
----> 2a3984f25e56
Step 8/12 : RUN mkdir -p /root/.ssh
----> Running in 9ff06abb04a6
Removing intermediate container 9ff06abb04a6
----> 8b0458108842
Step 9/12 : COPY authorized_keys /root/.ssh/authorized_keys
----> 22f92b3b7455
Step 10/12 : EXPOSE 22
----> Running in 0932c2089bda
Removing intermediate container 0932c2089bda
----> 6088e63b1667
Step 11/12 : EXPOSE 80
----> Running in 1cba3dde717
Removing intermediate container 1cba3dde717
----> 1aa3cef8bd24
Step 12/12 : CMD ["/usr/sbin/sshd", "-D"]
----> Running in 85ba26565aad
Removing intermediate container 85ba26565aad
----> 7e779b4b95bb
Successfully built 7e779b4b95bb
Successfully tagged tektutor/ansible-ubuntu-node:latest

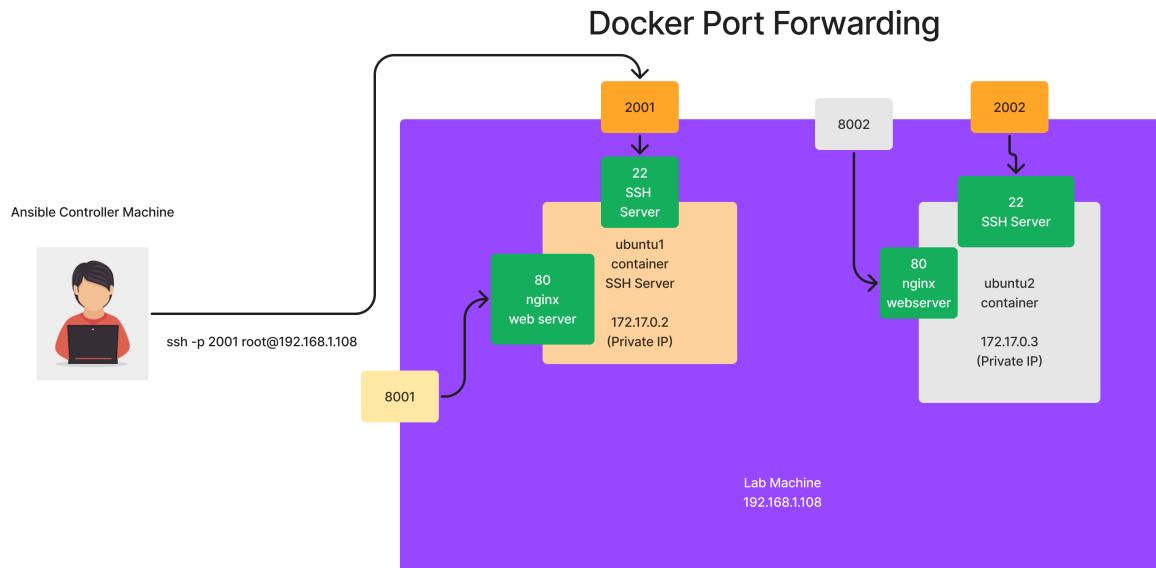
[jegan@ tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$ 

-----
```

```
jegan@tektutor:~/ansible-sep-2023          jegan@tektutor:~/ansible-sep-2023          jegan@tektutor:~/ansible-sep-2023
----> 2a3984f25e56
Step 8/12 : RUN mkdir -p /root/.ssh
----> Running in 9ff06abb04a6
Removing intermediate container 9ff06abb04a6
----> 8b0458108842
Step 9/12 : COPY authorized_keys /root/.ssh/authorized_keys
----> 22f92b3b7455
Step 10/12 : EXPOSE 22
----> Running in 0932c2089bda
Removing intermediate container 0932c2089bda
----> 6088e63b1667
Step 11/12 : EXPOSE 80
----> Running in 1cba3dde717
Removing intermediate container 1cba3dde717
----> 1aa3cef8bd24
Step 12/12 : CMD ["/usr/sbin/sshd", "-D"]
----> Running in 85ba26565aad
Removing intermediate container 85ba26565aad
----> 7e779b4b95bb
Successfully built 7e779b4b95bb
Successfully tagged tektutor/ansible-ubuntu-node:latest

[jegan@ tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
tektutor/ansible-ubuntu-node    latest   7e779b4b95bb  3 minutes ago  220MB
nginx               latest   eea7b3dcba7e   2 weeks ago   187MB
ubuntu              22.04   c6b84b685f35  2 weeks ago   77.8MB
hello-world         latest   9c7a54a9a43c  4 months ago  13.3kB
centos              7.9.2009  eeb6ee3f44bd  23 months ago  204MB
ubuntu              16.04   b6f507652425  2 years ago   135MB

[jegan@ tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$ 
```



## Lab - Creating ubuntu1 and ubuntu2 container using our Custom Ubuntu Ansible node image

```
docker run -d --name ubuntu1 --hostname ubuntu1 -p 2001:22 -p 8001:80
tektutor/ansible-ubuntu-node:latest
docker run -d --name ubuntu2 --hostname ubuntu2 -p 2002:22 -p 8002:80
tektutor/ansible-ubuntu-node:latest
docker images
```

### Expected output

```
jegan@tektutor:~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
5beb94a4715af7696388c1cfa2217dcf6f47ad5a84e8b3082074a894cac14a3f   tektutor/ansible-ubuntu-node:latest
5a5add3c3c290509764b19ab7d380cf760512cc8c6c72d84aa6380376a2fc5f   tektutor/ansible-ubuntu-node:latest
jegan@tektutor:~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu$ docker run -d --name ubuntu1 --hostname ubuntu1 -p 2001:22 -p 8001:80 tektutor/ansible-ubuntu-node:latest
jegan@tektutor:~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu$ docker run -d --name ubuntu2 --hostname ubuntu2 -p 2002:22 -p 8002:80 tektutor/ansible-ubuntu-node:latest
jegan@tektutor:~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
5a5add3c3c2         tektutor/ansible-ubuntu-node:latest   "/usr/sbin/sshd -D"   3 seconds ago     Up 2 seconds      0.0.0.0:2002->22/tcp, :::200
2->22/tcp, 0.0.0.0:8002->80/tcp, :::8002->80/tcp      ubuntu2
5beb94a4715a        tektutor/ansible-ubuntu-node:latest   "/usr/sbin/sshd -D"   19 seconds ago    Up 19 seconds     0.0.0.0:2001->22/tcp, :::200
1->22/tcp, 0.0.0.0:8001->80/tcp, :::8001->80/tcp      ubuntu1
jegan@tektutor:~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED             SIZE
tektutor/ansible-ubuntu-node  latest   7e779b4b95bb  About an hour ago  220MB
nginx              latest   eea7b3dcba7e  2 weeks ago       187MB
ubuntu              22.04   c6b84b685f35  2 weeks ago       77.8MB
hello-world        latest   9c7a54a9a43c  4 months ago      13.3KB
centos             7.9.2009  eeb6ee3f44bd  23 months ago     204MB
ubuntu              16.04   b6f507652425  2 years ago       135MB
jegan@tektutor:~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu$
```

# Lab - Testing if we are able to ssh into ubuntu1 and ubuntu2 container

```
ssh -p 2001 root@localhost
exit
ssh -p 2002 root@localhost
exit
```

## Expected output

```
(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
tektutor/ansible-ubuntu-node    latest   7e779b4b95bb  About an hour ago  220MB
nginx               latest   eea7b3dcb7e   2 weeks ago   187MB
ubuntu              22.04   c6b84b685f35  2 weeks ago   77.8MB
hello-world         latest   9c7a54a9a43c  4 months ago  13.3KB
centos              7.9.2009 eeb6ee3f44bd  23 months ago  204MB
ubuntu              16.04   b6f507652425  2 years ago   135MB

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$ ssh -p 2001 root@localhost
The authenticity of host '[localhost]:2001 ([::1]:2001)' can't be established.
ED25519 key fingerprint is SHA256:7rs8lMkA7TzqfbDwwO5vJ54N4sTj6vlNr5sW0YKBiI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2001' (ED25519) to the list of known hosts.
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 6.1.0-kali9-amd64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@ubuntu1:~# exit
logout
Connection to localhost closed.

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$ ssh -p 2002 root@localhost
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@ubuntu1:~# exit
logout
Connection to localhost closed.

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/CustomDockerImages/ubuntu]
$
```

## Lab - Running ansible ad-hoc command to ping the ansible nodes

```
cd ~/ansible-sep-2023
git pull
cd Day2/static-inventory
ansible -i inventory all -m ping
```

### Expected output

## Lab - Ansible ping with verbosity enabled upto 4 levels for troubleshooting/debugging purpose

```
cd ~/ansible-sep-2023
git pull
cd Day2/static-inventory
ansible -i inventory all -m ping -vvvv
```

## Expected output

```
jegan@tektutor:~/ansible-sep-2023/Day2/static-inventory
jegan@tektutor:~/ansible-sep-2023/Day2/static-inventory x jegan@tektutor:~/ansible-sep-2023/Day2/static-inventory x jegan@tektutor:~/ansible-sep-2023/Day2/static-inventory x jegan@tektutor:~/ansible-sep-2023/Day2/CustomDockerImage x
└$ ansible -i inventory all -m ping -vvvv
ansible [core 2.14.9]
  config file = None
  configured module search path = ['/home/jegan/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/jegan/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
No config file found; using defaults
setting up inventory plugins
host_list declined parsing /home/jegan/ansible-sep-2023/Day2/static-inventory/inventory as it did not pass its verify_file() method
script declined parsing /home/jegan/ansible-sep-2023/Day2/static-inventory/inventory as it did not pass its verify_file() method
auto declined parsing /home/jegan/ansible-sep-2023/Day2/static-inventory/inventory as it did not pass its verify_file() method
Parsed /home/jegan/ansible-sep-2023/Day2/static-inventory/inventory inventory source with ini plugin
Loading callback plugin minimal of type stdout, v2.0 from /usr/lib/python3/dist-packages/ansible/plugins/callback/minimal.py
Skipping callback 'default', as we already have a stdout callback.
Skipping callback 'minimal', as we already have a stdout callback.
Skipping callback 'oneline', as we already have a stdout callback.
<localhost> ESTABLISH SSH CONNECTION FOR USER: root
<localhost> SSH: EXEC ssh -vvv -C -o ControlMaster=auto -o ControlPersist=60s -o Port=2001 -o 'IdentityFile="/home/jegan/.ssh/id_rsa"' -o KbdInteractiveAuthentication=no -o PreferredAuthentications=gssapi-with-mic,gssapi-keyex,hostbased,publickey -o PasswordAuthentication=no -o 'User="root"' -o ConnectTimeout=10 -o 'ControlPath="/home/jegan/.ansible/cp/69ebd5c64d"' localhost '/bin/sh -c "echo ~root && sleep 0"'
jegan@tektutor:~/ansible-sep-2023/Day2/static-inventory
jegan@tektutor:~/ansible-sep-2023/Day2/static-inventory x jegan@tektutor:~/ansible-sep-2023/Day2/static-inventory x jegan@tektutor:~/ansible-sep-2023/Day2/static-inventory x jegan@tektutor:~/ansible-sep-2023/Day2/CustomDockerImage x
}
},
"ping": "pong"
}
<localhost> (0, b'', b"OpenSSH_9.2p1 Debian-2, OpenSSL 3.0.8 7 Feb 2023\r\ndebug1: Reading configuration data /etc/ssh/ssh_config\r\ndebug1: /etc/ssh/ssh_config line 19: include /etc/ssh/ssh_config.d/*.conf matched no files\r\ndebug1: /etc/ssh/ssh_config line 21: Applying options for *\r\ndebug3: expanded UserKnownHostsFile '/.ssh/known_hosts' -> '/home/jegan/.ssh/known_hosts'\r\ndebug3: expanded UserKnownHostsFile '/~/.ssh/known_hosts2' -> '/home/jegan/.ssh/known_hosts2'\r\ndebug1: auto-mux: Trying existing master\r\ndebug2: fd 3 setting O_NONBLOCK\r\ndebug2: mux_client_hello_exchange: master version 4\r\ndebug3: mux_client_forwards: request forwardings: 0 local, 0 remote\r\ndebug3: mux_client_request_session: entering\r\ndebug3: mux_client_request_alive: entering\r\ndebug3: mux_client_request_alive: done pid = 19378\r\ndebug3: mux_client_request_session: session request sent\r\ndebug1: mux_client_request_session: master session id: 2\r\ndebug3: mux_client_read_packet: read header failed: Broken pipe\r\ndebug2: Received exit status from master 0\r\n")
ubuntu1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "invocation": {
    "module_args": {
      "data": "pong"
    }
  },
  "ping": "pong"
}
(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/static-inventory]
$
```

## Lab - Using Ansible configuration file to point out the inventory ansible must be using

```
cd ~/ansible-sep-2023
git pull
cd Day2/static-inventory
cat ansible.cfg
ansible all -m ping
```

## Expected output

```
jegan@tektutor: ~/ansible-sep-2023/Day2/static-inventory
[jegan@tektutor.org] - [~/ansible-sep-2023/Day2/static-inventory]
$ cat ansible.cfg
[defaults]
inventory=./inventory

[jegan@tektutor.org] - [~/ansible-sep-2023/Day2/static-inventory]
$ ansible all -m ping
ubuntu2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}

[jegan@tektutor.org] - [~/ansible-sep-2023/Day2/static-inventory]
$
```

## Lab - Using ansible setup module to collect facts about ubuntu1 ansible node

```
cd ~/ansible-sep-2023
git pull
cd Day2/static-inventory
ansible all -m setup
```

## Expected output

```
jegan@tektutor: ~/ansible-sep-2023/Day2/static-inventory
[jegan@tektutor.org] - [~/ansible-sep-2023]
$ cd Day2/static-inventory

[jegan@tektutor.org] - [~/ansible-sep-2023/Day2/static-inventory]
$ ansible ubuntu1 -m setup
ubuntu1 | SUCCESS => {
    "ansible_facts": {
        "ansible_apparmor": {
            "status": "disabled"
        },
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "05/29/2023",
        "ansible_bios_vendor": "Dell Inc.",
        "ansible_bios_version": "2.34.0",
        "ansible_board_asset_tag": "NA",
        "ansible_board_name": "060K5C",
        "ansible_board_serial": "/22RCFD3/CNFCW0011L00QK/",
        "ansible_board_vendor": "Dell Inc.",
        "ansible_board_version": "A05",
        "ansible_chassis_asset_tag": "NA",
        "ansible_chassis_serial": "22RCFD3",
        "ansible_chassis_vendor": "Dell Inc.",
        "ansible_chassis_version": "NA",
        "ansible_cmdline": {
            "BOOT_IMAGE": "/boot/vmlinuz-6.1.0-kali9-amd64",
            "quiet": true,
            "ro": true,
            "root": "UUID=2af7aa39-8e72-4a24-953c-88e781058d9c",
            ...
        }
    }
}
```

```
jegan@tektutor:~/ansible-sep-2023          jegan@tektutor:~/ansible-sep-2023/Day2/static-inventory          jegan@tektutor:~/ansible-sep-2023          jegan@tektutor:~/ansible-sep-2023/CustomDockerImage
jegan@tektutor:~/ansible-sep-2023          jegan@tektutor:~/ansible-sep-2023/Day2/static-inventory          jegan@tektutor:~/ansible-sep-2023          jegan@tektutor:~/ansible-sep-2023/CustomDockerImage
{
    "sas_device_handle": null,
    "scheduler_mode": "mq-deadline",
    "sectors": "2097151",
    "sectorsize": "512",
    "size": "1024.00 MB",
    "support_discard": "0",
    "vendor": "HL-DT-ST",
    "virtual": 1
}
},
"ansible_distribution": "Ubuntu",
"ansible_distribution_file_parsed": true,
"ansible_distribution_file_path": "/etc/os-release",
"ansible_distribution_file_variety": "Debian",
"ansible_distribution_major_version": "16",
"ansible_distribution_release": "xenial",
"ansible_distribution_version": "16.04",
"ansible_dns": {
    "nameservers": [
        "8.8.8.8",
        "8.8.4.4"
    ],
    "options": {
        "edns0": true,
        "trust-ad": true
    }
},
"ansible_domain": "",
"ansible_effective_group_id": 0,
"ansible_environ": {
    "PWD": "/root",
    "SHELL": "/bin/bash",
    "SHLVL": "1",
    "SSH_CLIENT": "172.17.0.1 35722 22",
    "SSH_CONNECTION": "172.17.0.1 35722 172.17.0.2 22",
    "SSH_TTY": "/dev/pts/0",
    "TERM": "xterm-256color",
    "USER": "root",
    "": "/bin/sh"
},
"ansible_fibre_channel_wwn": [],
"ansible_fips": false,
"ansible_form_factor": "Desktop",
"ansible_fqdn": "ubuntu1",
"ansible_hostname": "ubuntu1",
"ansible_hostqn": "",
"ansible_is_chroot": false,
"ansible_iscsi_iqn": "",
"ansible_kernel": "6.1.0-kali9-amd64",
"ansible_kernel_version": "#1 SMP PREEMPT_DYNAMIC Debian 6.1.27-1kali1 (2023-05-12)",
"ansible_loadavg": {
    "15m": 1.05,
    "1m": 0.96,
    "5m": 1.05
},
"ansible_local": {},
"ansible_lsb": {
    "codename": "xenial",
    "description": "Ubuntu 16.04.7 LTS",
    ...
}
```



```
jegan@tekutor: ~/ansible-sep-2023
jegan@tekutor: ~/ansible-sep-2023/Day2/static-inventory
jegan@tekutor: ~/ansible-sep-2023
jegan@tekutor: ~/ansible-sep-2023/CustomDockerImage

{
    "minor": 5,
    "releaselevel": "final",
    "serial": 0
},
"version_info": [
    3,
    5,
    2,
    "final",
    0
],
"ansible_python_version": "3.5.2",
"ansible_real_group_id": 0,
"ansible_real_user_id": 0,
"ansible_selinux": {
    "status": "disabled"
},
"ansible_selinux_python_present": true,
"ansible_service_mgr": "sshd",
"ansible_ssh_host_key_dsa_public": "AAAAB3NzaC1kc3MAAACBAIJZv0vCmfK1Cd0tJUnM7nD+pXL3EvHChse+VXiA3/1ozC4ZnH1PogNb1eW41kuj5UoBoBNTVL+8G8n51P8ScJgLiYoBKYM9GA2N0GMqsozVSmbymNMilMMYZNLfw6hnIMpvfhQ/ASkfZIX1tXVnPpqJdmh//+bghZrcgWuSzFAAAAFQC0Cx2DNMZ5XJv76HAMX7G50oVGgQAAAIAXWd3OD0Jfyz0mR2c4cH+crt9m16PLaSeH6TOV3Ahw70L+qHkvpwYAEs3jukE2E0gIB7H84etVsFbnjjK6nyGKMinwD2AmD7kG8H8yYjVURK+T3zhe5QMGYxvgHymZsa9erKxLtzAqiHLHwvJitk90TskUD85VE/DjW/XqPtfWyAAIAIB5UZhHaLc4EtY4fTAK5ne3vD8i61WGbENFH0BIHuUKfTqXbtUthFnfMQlaebYo22vg6iYnWJ1A/oru8/bJN6//GrK6d47d67ab2w+4UCiapp/gh50kj4/VnksFGyvJsI3GhWa2Db0Lyn303SH3LzbW31GZIJpzLTnVFTTeXQDw==",
"ansible_ssh_host_key_dsa_pubkeytype": "ssh-dss",
"ansible_ssh_host_key_ecdsa_public": "AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAIBmlzdHAyNTYAAABBBGoFQ13YDySwyr3N6MZGFtNNoHzJWxRz80kIEzUr9LZplUQyrAIpffBw7SLSRxrd8Gg3D1LYv5FA9Q1JFlAFo8=",
```

```
jegan@tekutor: ~/ansible-sep-2023
jegan@tekutor: ~/ansible-sep-2023/Day2/static-inventory
jegan@tekutor: ~/ansible-sep-2023
jegan@tekutor: ~/ansible-sep-2023/CustomDockerImage

{
    "ansible_uptime_seconds": 19029,
    "ansible_user_dir": "/root",
    "ansible_user_gecos": "root",
    "ansible_user_gid": 0,
    "ansible_user_id": "root",
    "ansible_user_shell": "/bin/bash",
    "ansible_user_uid": 0,
    "ansible_userspace_architecture": "x86_64",
    "ansible_userspace_bits": "64",
    "ansible_virtualization_role": "guest",
    "ansible_virtualization_tech_guest": [
        "container",
        "docker"
    ],
    "ansible_virtualization_tech_host": [
        "kvm"
    ],
    "ansible_virtualization_type": "docker",
    "discovered_interpreter_python": "/usr/bin/python3",
    "gather_subset": [
        "all"
    ],
    "module_setup": true
},
"changed": false
}

(jegan@tekutor.org)-[~/ansible-sep-2023/Day2/static-inventory]
```

## Lab - Getting help about ansible modules

```
ansible-doc -l
```

## Expected output

```
(jegan@tektutor.org)-[~/ansible-sep-2023/Day2]
$ ansible-doc -l
amazon.aws.autoscaling_group
amazon.aws.autoscaling_group_info
amazon.aws.aws_az_info
amazon.aws.aws_caller_info
amazon.aws.cloudformation
amazon.aws.cloudformation_info
amazon.aws.cloudtrail
amazon.aws.cloudtrail_info
amazon.aws.cloudwatch_metric_alarm
amazon.aws.cloudwatch_metric_alarm_info
amazon.aws.cloudwatchevent_rule
amazon.aws.cloudwatchlogs_log_group
amazon.aws.cloudwatchlogs_log_group_info
amazon.aws.cloudwatchlogs_log_group_metric_filter
amazon.aws.ec2_ami
amazon.aws.ec2_ami_info
amazon.aws.ec2_eip
amazon.aws.ec2_eip_info
amazon.aws.ec2_eni
amazon.aws.ec2_eni_info
amazon.aws.ec2_instance
amazon.aws.ec2_instance_info
amazon.aws.ec2_key
amazon.aws.ec2_metadata_facts
amazon.aws.ec2_security_group
amazon.aws.ec2_security_group_info
Create or ...
Gather inf...
Gather inf...
Get inform...
Create or ...
Obtain inf...
manage Clo...
Gather inf...
Create/upd...
Gather inf...
Manage Clo...
create or ...
Get inform...
Manage Clo...
Create or ...
Gather inf...
manages EC...
List EC2 E...
Create and...
Gather inf...
Create & m...
Gather inf...
Create or ...
Gathers fa...
Maintain a...
Gather inf...
```

## Lab - Finding total number of ansible modules supported by your version of Ansible

```
ansible-doc -l | wc -l
```

## Expected output

```
(jegan@tektutor.org)-[~/ansible-sep-2023/Day2]
$ ansible-doc -l | wc -l
7736

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2]
```

## Lab - Getting help info about any ansible module

```
ansible-doc ping
ansible-doc shell
ansible-doc copy
ansible-doc file
ansible-doc template
ansible-doc service
ansible_doc docker_image
```

```
ansible_doc docker_container
ansible_doc setup
```

## Expected output

The screenshot shows a terminal window with three tabs, each displaying the output of the `ansible-doc` command. The first tab shows the total number of pages (7736). The second tab shows the ping module, which is a trivial test module that always returns 'pong' on successful contact. The third tab shows the historical module, which is added in historical versions.

```
jegan@tektutor:~/ansible-sep-2023/Day2
$ ansible-doc -l | wc -l
7736

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2]
$ ansible-doc ping
> ANSIBLE.BUILTIN.PING      (/usr/lib/python3/dist-packages/ansible/modules/ping.py)

A trivial test module, this module always returns 'pong' on successful contact.
It does not make sense in playbooks, but it is useful from '/usr/bin/ansible' to
verify the ability to login and that a usable Python is configured. This is NOT
ICMP ping, this is just a trivial test module that requires Python on the remote-
node. For Windows targets, use the [ansible.windows.win_ping] module instead. For
Network targets, use the [ansible.netcommon.net_ping] module instead.

ADDED IN: historical

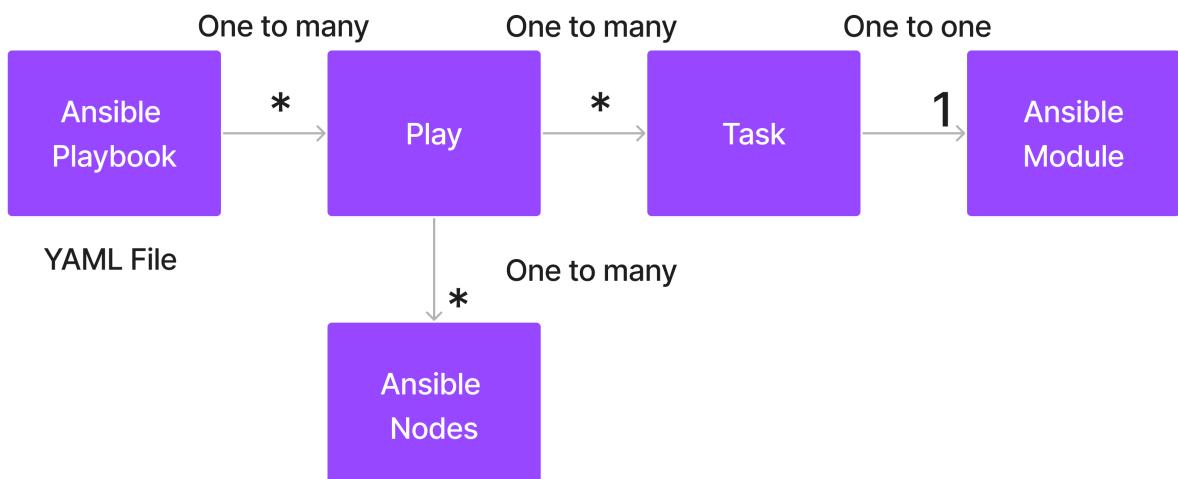
OPTIONS (= is mandatory):

- data
  Data to return for the 'ping' return value.
  If this parameter is set to 'crash', the module will cause an exception.
  default: pong
  type: str

ATTRIBUTES:
```

## Ansible Playbook Structure

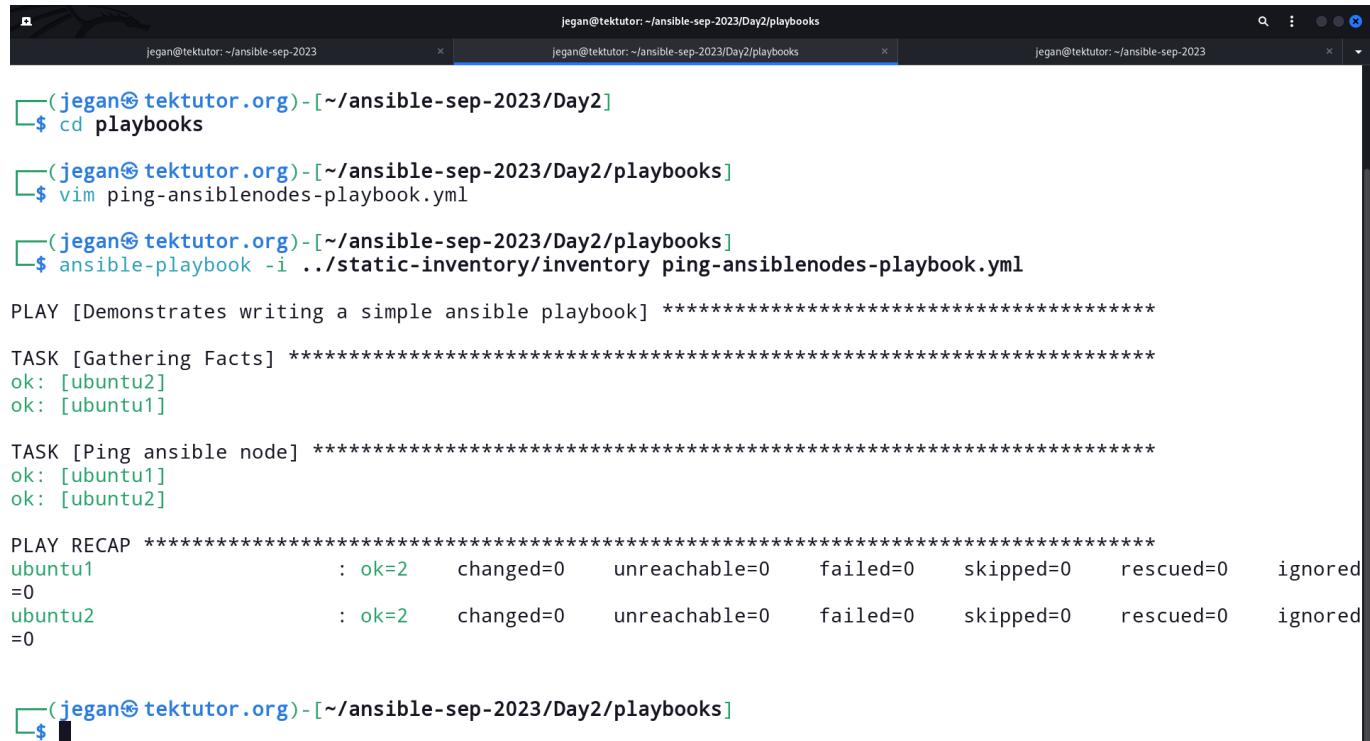
### Structure of Ansible Playbook



## Lab - Running your first ansible playbook

```
cd ~/ansible-sep-2023
git pull
cd Day2/playbooks
ansible-playbook -i ../static-inventory/inventory ping-ansiblenode-
playbook.yml
```

## Expected output



The screenshot shows three terminal windows side-by-side, all running under the user jegan@tektutor.org. The first window shows the command \$ cd playbooks. The second window shows the command \$ vim ping-ansiblenodes-playbook.yml. The third window shows the execution of the playbook with output indicating two hosts (ubuntu1 and ubuntu2) were pinged successfully (ok=2). The final line shows the command \$.

```
(jegan@tektutor.org)-[~/ansible-sep-2023/Day2]
$ cd playbooks

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ vim ping-ansiblenodes-playbook.yml

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ ansible-playbook -i ../static-inventory/inventory ping-ansiblenodes-playbook.yml

PLAY [Demonstrates writing a simple ansible playbook] ****
TASK [Gathering Facts] ****
ok: [ubuntu2]
ok: [ubuntu1]

TASK [Ping ansible node] ****
ok: [ubuntu1]
ok: [ubuntu2]

PLAY RECAP ****
ubuntu1 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$
```

## Lab - Performing syntax checking without running the playbook

```
cd ~/ansible-sep-2023
git pull
cd Day2/playbooks
ansible-playbook -i ../static-inventory/inventory ping-ansiblenode-
playbook.yml --syntax-check
```

## Expected output

The screenshot shows a terminal window with three tabs, all titled 'jegan@tektutor: ~/ansible-sep-2023/Day2/playbooks'. The first tab shows the command '\$ vim ping-ansiblenodes-playbook.yml'. The second tab shows the contents of the playbook file:

```
(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ cat ping-ansiblenodes-playbook.yml
- name: Demonstrates writing a simple ansible playbook
  hosts: all
  tasks:
    - name: Ping1 ansible node
      ping:
    - name: Ping2 ansible node
      ping:
```

The third tab shows the command '\$ ansible-playbook -i ../static-inventory/inventory ping-ansiblenodes-playbook.yml --syntax-check' followed by an error message:

```
$ ansible-playbook -i ../static-inventory/inventory ping-ansiblenodes-playbook.yml --syntax-check
ERROR! unexpected parameter type in action: <class 'ansible.parsing.yaml.objects.AnsibleSequence'>
```

The error message indicates that there is a syntax problem in the playbook, specifically on line 4, column 5.

The offending line appears to be:

```
tasks:
- name: Ping1 ansible node
  ^ here
```

The terminal prompt '\$' is visible at the bottom of the third tab.

## Lab - Refactoring the inventory file, moving all common variable as group variables

```
cd ~/ansible-sep-2023
git pull
cd Day2/playbooks
cat ansible.cfg
cat hosts
ansible all -m ping
```

## Expected output

The screenshot shows a terminal window with three tabs, all titled 'jegan@tektutor: ~/ansible-sep-2023'. The first tab shows the contents of 'ansible.cfg' and 'hosts' files. The second tab shows the execution of 'ansible all -m ping' on hosts 'ubuntu1' and 'ubuntu2', with both returning success status. The third tab shows the command being run.

```
jegan@tektutor: ~/ansible-sep-2023
$ cat ansible.cfg
[defaults]
inventory=./hosts

[jegan@tektutor.org]-[~/ansible-sep-2023/Day2/playbooks]
$ cat hosts
[all]
ubuntu1 ansible_port=2001
ubuntu2 ansible_port=2002

[all:vars]
ansible_user=root
ansible_host=localhost
ansible_private_key_file=~/ssh/id_rsa

[jegan@tektutor.org]-[~/ansible-sep-2023/Day2/playbooks]
$ ansible all -m ping
ubuntu1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}

[jegan@tektutor.org]-[~/ansible-sep-2023/Day2/playbooks]
$
```

Capturing the output of ansible ad-hoc command

```
ansible ubuntu1 -m ping -vvvv > out.yml 2>&1
```

## Lab - Running the install nginx playbook

```
cd ~/ansible-sep-2023
git pull
cd Day2/playbooks
cat ansible.cfg
cat hosts
ansible-playbook install-nginx-playbook.yml
```

## Expected output

```
jegan@tektutor: ~/ansible-sep-2023/Day2/playbooks
jegan@tektutor: ~/ansible-sep-2023/Day2/playbooks
jegan@tektutor: ~/ansible-sep-2023

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ ansible-playbook install-nginx-playbook.yml

PLAY [This playbook will install nginx, configures custom path of web root folder and deploys custom web page] ***
TASK [Gathering Facts] *****
ok: [ubuntu1]
ok: [ubuntu2]

TASK [Install nginx in Ubuntu node] *****
changed: [ubuntu2]
changed: [ubuntu1]

PLAY RECAP *****
ubuntu1      : ok=2    changed=1    unreachable=0    failed=0     skipped=0    rescued=0    ignored=0
ubuntu2      : ok=2    changed=1    unreachable=0    failed=0     skipped=0    rescued=0    ignored=0

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$
```

```
jegan@tektutor: ~/ansible-sep-2023/Day2/playbooks
jegan@tektutor: ~/ansible-sep-2023/Day2/playbooks
jegan@tektutor: ~/ansible-sep-2023

$ cd Day2/playbooks
(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ ls
ansible.cfg  hosts  install-nginx-playbook.yml  ping-ansiblenodes-playbook.yml

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ cat install-nginx-playbook.yml
- name: This playbook will install nginx, configures custom path of web root folder and deploys custom web page
  hosts: all
  tasks:
  - name: Install nginx in Ubuntu node
    apt: name=nginx state=latest update_cache=yes

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ ansible-playbook install-nginx-playbook.yml

PLAY [This playbook will install nginx, configures custom path of web root folder and deploys custom web page] *****
TASK [Gathering Facts] *****
ok: [ubuntu2]
ok: [ubuntu1]

TASK [Install nginx in Ubuntu node] *****
ok: [ubuntu1]
ok: [ubuntu2]

PLAY RECAP *****
ubuntu1      : ok=2    changed=0    unreachable=0    failed=0     skipped=0    rescued=0    ignored=0
ubuntu2      : ok=2    changed=0    unreachable=0    failed=0     skipped=0    rescued=0    ignored=0

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$
```

Let's us check if the html page is accessible from ubuntu1 and ubuntu2 ansible nodes

```
curl http://localhost:8001
curl http://localhost:8002
ansible ubuntu1 -m shell -a "service nginx status"
```

## Expected output

```
jegan@tektutor:~/ansible-sep-2023/Day2/playbooks
(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ docker ps
CONTAINER ID   IMAGE           COMMAND          CREATED        STATUS          NAMES
5a5add3c3c2   tektutor/ansible-ubuntu-node:latest   "/usr/sbin/sshd -D"   3 hours ago   Up 3 hours   ubuntu2
5beb94a4715a   tektutor/ansible-ubuntu-node:latest   "/usr/sbin/sshd -D"   3 hours ago   Up 3 hours   ubuntu1

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ curl http://localhost:8001
curl: (56) Recv failure: Connection reset by peer

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ curl http://localhost:8002
curl: (56) Recv failure: Connection reset by peer

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ ansible ubuntu1 -m shell -a "service nginx status"
ubuntu1 | FAILED | rc=3 >>
* nginx is not running non-zero return code

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$
```

Let's run the refactored playbook

```
cd ~/ansible-sep-2023
git pull
cd Day2/playbooks
cat ansible.cfg
cat hosts
ansible-playbook install-nginx-playbook.yml

curl http://localhost:8001
curl http://localhost:8002
```

## Expected output

```
jegan@tektutor:~/ansible-sep-2023/Day2/playbooks
jegan@tektutor:~/ansible-sep-2023/Day2/playbooks
jegan@tektutor:~/ansible-sep-2023/Day2/playbooks
(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ vim install-nginx-playbook.yml

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ ansible-playbook install-nginx-playbook.yml

PLAY [This playbook will install nginx, configures custom path of web root folder and deploys custom web page] ***

TASK [Gathering Facts] ****
ok: [ubuntu1]
ok: [ubuntu2]

TASK [Install nginx in Ubuntu node] ****
ok: [ubuntu2]
ok: [ubuntu1]

TASK [Start nginx web server in Ubuntu nodes] ****
changed: [ubuntu2]
changed: [ubuntu1]

PLAY RECAP ****
ubuntu1           : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2           : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

(jegan@tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ curl http://localhost:8001
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
  width: 35em;
  margin-left: auto;
  margin-right: auto;
}
```

The screenshot shows a terminal window with three tabs. The first tab displays Ansible's PLAY RECAP for two hosts: ubuntu1 and ubuntu2. Both hosts show 3 OK and 1 CHANGED status across various tasks. The second tab shows the output of the curl command to localhost:8001, displaying the HTML response from Nginx. The third tab shows the output of the curl command to localhost:8002, which only displays the 'Thank you for using nginx.' message.

```
jegan@tektutor: ~/ansible-sep-2023/Day2/playbooks
PLAY RECAP ****
ubuntu1 : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2 : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

(jegan@ tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ curl http://localhost:8001
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

(jegan@ tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$ curl http://localhost:8002
<p><em>Thank you for using nginx.</em></p>
</body>
</html>

(jegan@ tektutor.org)-[~/ansible-sep-2023/Day2/playbooks]
$
```