

Day3

What is Configuration Management Tool?

- a tool that supports automating administrative activities
- we can automate software installations/uninstallations, updates/upgrades, creating users with certain permissions, setup firewalls/proxy, configuring db servers, importing databases, rebooting, user management, etc.,
- examples
 - Puppet
 - Chef
 - Salt/Saltstack
 - Ansible

What is DSL?

- Domain Specific Language
- the language used to write the automation script
- the DSL used by Puppet/Chef tool is Ruby scripting language
- the choice of DSL makes a configuration management tool easy or tough

Puppet/Chef

- DSL used in Puppet/Chef is Ruby
- the installation process is very tough
- requires many servers/machines
- learning curve is steep
- architecture is complex
- uses proprietary tools for everything
- client/server architecture
- the machines where Puppet/Chef performs software installation automation is called Puppet/Chef nodes
- In the Puppet/Chef nodes we need to install some Puppet/Chef agents
- The agents will periodically connect to the Puppet/Chef servers looking for new automation scripts or any updates in existing scripting
- When the agents find a new/updated script, it pulls the scripts and runs on the Puppet/Chef agents
- follows Pull based architecture

Why Ansible?

- agentless
- doesn't use any proprietary tools on the Ansible nodes
- DSL used is YAML, which is easy to learn
- Ansible itself is developed in Python language, but we don't need to know Python to write Ansible playbooks
- Ansible architecture very simple

- Ansible comes in 3 flavours
 - Ansible Core (open source, supports only command-line)
 - Ansible AWX (open source, supports Web Interface, developed on top Ansible Core)
 - Red Hat Ansible Tower (Requires license, developed on top of Ansible AWX)
- follows PUSH based architecture
- very easy to learn and install
- follows a very simple architecture
- developed by Michael Deehan
- Michael Deehan is former employee of Red Hat, after quitting Red Hat he started a company called Ansible Inc
- Ansible core was developed as an open source product
- Once Ansible Core gained popularity, Red Hat acquired Ansible Inc, hence Ansible is a Red Hat product
- Later, IBM acquired Red Hat, hence it is an IBM product

What are the Ansible Alternatives

- Puppet
- Chef and
- Salt

What is an Ansible ad-hoc command?

- through ansible ad-hoc commands, we call one ansible module at a time
- generally used for poc or as a support tool
- for complex automations, we need use Ansible Playbook

What is an Ansible Playbook?

- is the automation script written in YAML file
- this helps in software installation and configuration automation
- it would be invoked one or more Ansible Modules in a specific order one after the other

Lab - Finding the ansible version

```
ansible --version
```

Expected output

```
└──(jegan@tektutor.org) - [~/devops-dec-2023]
└─$ 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 = /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 - Creating a Custom Ubuntu Ansible node docker image

```
cd ~/devops-dec-2023
git pull
cd Day3/ansible/CustomDockerImages/ubuntu
ssh-keygen
cp ~/.ssh/id_rsa.pub authorized_keys
ls
docker build -t tektutor/ansible-ubuntu-node:latest .
docker images
```

Expected output

The screenshot shows a terminal window with the following session:

```
jegan@tektutor:~/devops-dec-2023/Day3/ansible/CustomDockerImages/ubuntu
-rw-r--r-- 1 jegan jegan 575 Dec  8 11:56 Dockerfile
[jegan@tektutor.org ~] $ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jegan/.ssh/id_rsa):
/home/jegan/.ssh/id_rsa already exists.
Overwrite (y/n)? y
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:
SHA256:8LTf0cbC82p06NvEHPiJGtPFl98v4cVlk4ZDQ9K+ZRto jegan@tektutor
The key's randomart image is:
+---[RSA 3072]----+
| .+.. |
| +.. |
| . . o.. |
| + . + o o+ |
| S . O B+. |
| o O &.=+ |
| o = X.Eoo |
| = +.oo . |
| . o++ ... |
+---[SHA256]----+
[jegan@tektutor.org ~] $ cp /home/jegan/.ssh/id_rsa.pub authorized_keys
```

```
jegan@tektutor:~/devops-dec-2023/Day3/ansible/CustomDockerImages/ubuntu
$ ls
authorized_keys Dockerfile

[jegan@tektutor.org]-(~/.../Day3/ansible/CustomDockerImages/ubuntu)
$ docker build -t tektutor/ansible-ubuntu-node:latest .
Sending build context to Docker daemon 4.096kB
Step 1/12 : FROM ubuntu:latest
latest: Pulling from library/ubuntu
5e8117c0bd28: Pull complete
Digest: sha256:8eab65df33a6de2844c9aef19efe8ddb87b7df5e9185a4ab73af936225685bb
Status: Downloaded newer image for ubuntu:latest
--> b6548eacb063
Step 2/12 : MAINTAINER Jeganathan Swaminathan <jegan@tektutor.org>
--> Running in 183d9bef2800
Removing intermediate container 183d9bef2800
--> e046742030a9
Step 3/12 : RUN apt-get update && apt-get install -y openssh-server python3
--> Running in df1eb199e84a
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1282 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [44.0 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1036 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1512 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [164 kB]

[jegan@tektutor.org]-(~/.../Day3/ansible/CustomDockerImages/ubuntu)
$ docker images
REPOSITORY          TAG      IMAGE ID   CREATED        SIZE
tektutor/ansible-ubuntu-node    latest   91cae68762a9  4 seconds ago  234MB
ubuntu              latest   b6548eacb063  6 days ago   77.8MB
tektutor/hello-microservice     v2       fd0985e6a33e  6 days ago   408MB
tektutor/hello-microservice     v1       3a8b73745255  6 days ago   408MB
tektutor/openshift-maven       latest   11f8b18e433e  8 days ago   408MB
bitnami/postgresql            latest   54c74bafb64f  10 days ago  274MB
releases-docker.jfrog.io/jfrog/artifactory-oss  latest   d7bff3713f0d  2 weeks ago  2.19GB
mysql                latest   a3b66088989d6  6 weeks ago  596MB
tektutor/maven              latest   41537e04c0f5  2 months ago 408MB
registry.access.redhat.com/ubi8/openjdk-11      latest   c71cc6972823  3 months ago 391MB
nginx                latest   eea7b3dcba7e  3 months ago 187MB
hello-world           latest   9c7a54a9a43c  7 months ago 13.3kB
centos               7.9.2009  eeb6ee3f44bd  2 years ago  204MB
centos               centos7.9.2009  eeb6ee3f44bd  2 years ago  204MB
```

Lab - Creating a Custom CentOS ansible node docker image

```
cd ~/devops-dec-2023
git pull
cd Day3/ansible/CustomDockerImages/centos
cp ~/.ssh/id_rsa.pub authorized_keys
ls
docker build -t tektutor/ansible-centos-node:latest .
docker images
```

Expected output

The screenshot shows a terminal window with two tabs. The left tab is titled '(jegan@tektutor.org)-[~/.../Day3/ansible/CustomDockerImages/centos]' and the right tab is titled 'jegan@tektutor: ~/devops-dec-2023'. The left tab contains the following command-line session:

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible/CustomDockerImages/centos
[jegan@tektutor.org] ~/.../Day3/ansible/CustomDockerImages/centos
$ cp /home/jegan/.ssh/id_rsa.pub authorized_keys

[jegan@tektutor.org] ~/.../Day3/ansible/CustomDockerImages/centos
[jegan@tektutor.org] ~/.../Day3/ansible/CustomDockerImages/centos
$ ls
authorized_keys  Dockerfile

[jegan@tektutor.org] ~/.../Day3/ansible/CustomDockerImages/centos
[jegan@tektutor.org] ~/.../Day3/ansible/CustomDockerImages/centos
$ docker build -t tektutor/ansible-centos-node:latest .
Sending build context to Docker daemon 4.096kB
Step 1/13 : FROM centos:centos7
centos7: Pulling from library/centos
Digest: sha256:be65f488b7764ad3638f236b7b515b3678369a5124c47b8d32916d6487418ea4
Status: Downloaded newer image for centos:centos7
--> eeb6ee3f44bd
Step 2/13 : MAINTAINER Jeganathan Swaminathan <jegan@tektutor.org>
--> Running in 125bd0c61366
Removing intermediate container 125bd0c61366
--> 07de6b46a2e4
Step 3/13 : RUN yum install -y which openssh-clients openssh-server python3
--> Running in 5670e62b8561
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: mirrors.nxtgen.com
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Resolving Dependencies
--> Running transaction check
--> Package openssh-clients.x86_64 0:7.4p1-23.el7_9 will be installed
```

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible/CustomDockerImages/centos
Removing intermediate container c3ed938dd823
--> f27f1a2b6df1
Step 12/13 : EXPOSE 80
--> Running in 8cdcb941093f
Removing intermediate container 8cdcb941093f
--> 40d981a644a1
Step 13/13 : ENTRYPOINT ["/usr/sbin/sshd", "-D"]
--> Running in 83253dd35eb0
Removing intermediate container 83253dd35eb0
--> 833be4f773b8
Successfully built 833be4f773b8
Successfully tagged tektutor/ansible-centos-node:latest
```

```
(jegan@tektutor.org)-[~/.../Day3/ansible/CustomDockerImages/centos]
$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
tektutor/ansible-centos-node    latest   833be4f773b8  30 seconds ago  488MB
tektutor/ansible-ubuntu-node    latest   91cae68762a9  17 minutes ago  234MB
ubuntu              latest   b6548eacb063  6 days ago    77.8MB
tektutor/hello-microservice     v2      fd0985e6a33e  7 days ago    408MB
tektutor/hello-microservice     v1      3a8b73745255  7 days ago    408MB
tektutor/openshift-maven       latest   11f8b18e433e  8 days ago    408MB
bitnami/postgresql            latest   54c74bafb64f  10 days ago   274MB
releases-docker.jfrog.io/jfrog/artifactory-oss  latest   d7bff3713f0d  2 weeks ago   2.19GB
mysql                latest   a3b6608898d6  6 weeks ago   596MB
tektutor/maven              latest   41537e04c0f5  2 months ago  408MB
registry.access.redhat.com/ubi8/openjdk-11    latest   c71cc6972823  3 months ago  391MB
nginx                latest   eea7b3dcba7e  3 months ago  187MB
hello-world           latest   9c7a54a9a43c  7 months ago  13.3kB
```

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible/CustomDockerImages/centos
Removing intermediate container c3ed938dd823
--> f27f1a2b6df1
Step 12/13 : EXPOSE 80
--> Running in 8cdcb941093f
Removing intermediate container 8cdcb941093f
--> 40d981a644a1
Step 13/13 : ENTRYPOINT ["/usr/sbin/sshd", "-D"]
--> Running in 83253dd35eb0
Removing intermediate container 83253dd35eb0
--> 833be4f773b8
Successfully built 833be4f773b8
Successfully tagged tektutor/ansible-centos-node:latest
```

```
(jegan@tektutor.org)-[~/.../Day3/ansible/CustomDockerImages/centos]
$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
tektutor/ansible-centos-node    latest   833be4f773b8  30 seconds ago  488MB
tektutor/ansible-ubuntu-node    latest   91cae68762a9  17 minutes ago  234MB
ubuntu              latest   b6548eacb063  6 days ago    77.8MB
tektutor/hello-microservice     v2      fd0985e6a33e  7 days ago    408MB
tektutor/hello-microservice     v1      3a8b73745255  7 days ago    408MB
tektutor/openshift-maven       latest   11f8b18e433e  8 days ago    408MB
bitnami/postgresql            latest   54c74bafb64f  10 days ago   274MB
releases-docker.jfrog.io/jfrog/artifactory-oss  latest   d7bff3713f0d  2 weeks ago   2.19GB
mysql                latest   a3b6608898d6  6 weeks ago   596MB
tektutor/maven              latest   41537e04c0f5  2 months ago  408MB
registry.access.redhat.com/ubi8/openjdk-11    latest   c71cc6972823  3 months ago  391MB
nginx                latest   eea7b3dcba7e  3 months ago  187MB
hello-world           latest   9c7a54a9a43c  7 months ago  13.3kB
```

Lab - Creating two containers using our custom ubuntu ansible node

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

Expected output

```
jegan@tektutor:~/devops-dec-2023
$ docker run -dit --name ubuntu1 --hostname ubuntu1 -p 2001:22 -p 8001:80 tektutor/ansible-ubuntu-node:latest
97d19fbf48dcc96e722ea7fed1a7af8ad92eb291066318a6b0f1fb897b39fa7b

(jegan@tektutor.org)-[~/devops-dec-2023]
$ docker run -dit --name ubuntu2 --hostname ubuntu2 -p 2002:22 -p 8002:80 tektutor/ansible-ubuntu-node:latest
4a6a5e172cae55471c7ce634bde248c482c66fd98b4c2095ca8c0cae7cf4d89

(jegan@tektutor.org)-[~/devops-dec-2023]
$ docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED             STATUS              PORTS
S
4a6a5e172cae   tektutor/ansible-ubuntu-node:latest   "/usr/sbin/sshd -D"
0.0:2002->22/tcp, :::2002->22/tcp, 0.0.0.0:8002->80/tcp, :::8002->80/tcp
97d19fbf48dc   tektutor/ansible-ubuntu-node:latest   "/usr/sbin/sshd -D"
0.0:2001->22/tcp, :::2001->22/tcp, 0.0.0.0:8001->80/tcp, :::8001->80/tcp

(jegan@tektutor.org)-[~/devops-dec-2023]
$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED             SIZE
tektutor/ansible-centos-node    latest   833be4f773b8  9 minutes ago   488MB
tektutor/ansible-ubuntu-node    latest   91cae68762a9  26 minutes ago  234MB
ubuntu              latest   b6548eacb063  6 days ago       77.8MB
tektutor/hello-microservice     v2      fd0985e6a33e  7 days ago       408MB
tektutor/hello-microservice     v1      3a8b73745255  7 days ago       408MB
tektutor/openshift-maven       latest   11f8b18e433e  8 days ago       408MB
bitnami/postgresql            latest   54c74bafb64f  10 days ago      274MB
releases-docker.jfrog.io/jfrog/artifactory-oss  latest   d7bff3713f0d  2 weeks ago      2.19GB
```

Lab - Testing the ubuntu1 and ubuntu2 ansible node containers for ssh connectivity with a key-based login authentication

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

Expected output

```
jegan@tektutor:~/devops-dec-2023
$ ssh -p 2001 root@localhost
The authenticity of host '[localhost]:2001 ([::1]:2001)' can't be established.
ED25519 key fingerprint is SHA256:ngPSJ1PldtfmJ03e1dDj25PoyTehIVJv/vPCZ7IawoM.
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 22.04.3 LTS (GNU/Linux 6.4.0-kali3-amd64 x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

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: ~/devops-dec-2023
[jegan@tektutor.org]-(~/devops-dec-2023]
$ ssh -p 2002 root@localhost
The authenticity of host '[localhost]:2002 ([::1]:2002)' can't be established.
ED25519 key fingerprint is SHA256:ngPSJ1PlDtfmJ03e1dDj25PoyTehIVJv/vPCZ7IawoM.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:2: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2002' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.4.0-kali3-amd64 x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

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@ubuntu2:~# exit
logout
Connection to localhost closed.
```

Lab - Running an ansible ad-hoc command using ping module

```
cd ~/devops-dec-2023
git pull
cd Day3/ansible
ansible -i inventory all -m ping
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
$ cat inventory
[all]
ubuntu1 ansible_port=2001 ansible_user=root ansible_host=localhost ansible_private_key_file=~/ssh/id_rsa
ubuntu2 ansible_port=2002 ansible_user=root ansible_host=localhost ansible_private_key_file=~/ssh/id_rsa

[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
$ ansible -i inventory 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]-(~/devops-dec-2023/Day3/ansible]
```

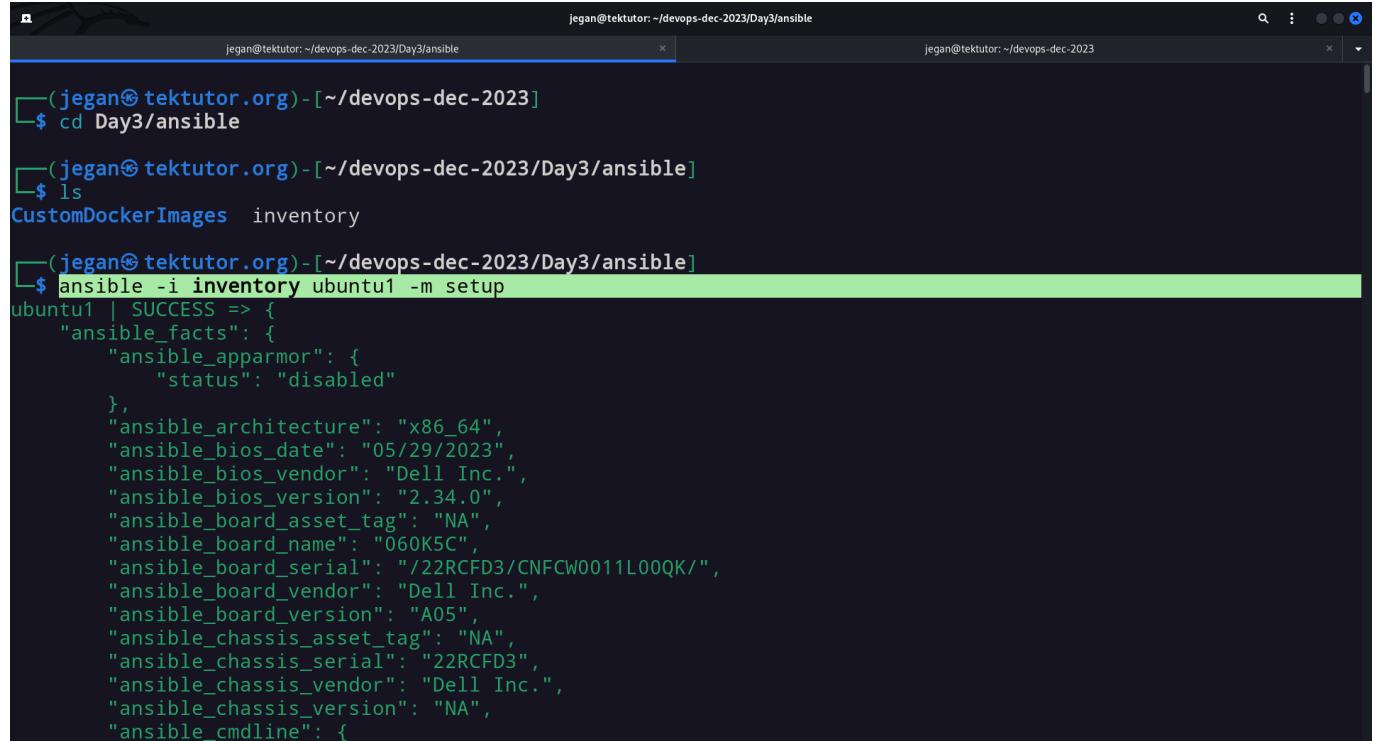
Lab - Ansible ad-hoc command to retrieve facts about the ansible nodes using setup ansible module

Things to know about setup ansible module

- the setup modules gets invoked as the very first task in a playbook by default
- it collects many useful details, like what OS the ansible node has, what is the python version installed, hardware details, kernel details, etc.,,

```
cd ~/devops-dec-2023
git pull
cd Day3/ansible
ansible -i inventory all -m setup
ansible -i inventory all -m setup | grep ansible_distribution
ansible -i inventory all -m setup | grep ansible_os_family
ansible -i inventory all -m setup | grep ansible_pkg
ansible -i inventory all -m setup | grep python
```

Expected output



The screenshot shows a terminal window with two tabs. The active tab displays the following command-line session:

```
(jegan@tektutor.org)-[~/devops-dec-2023]
$ cd Day3/ansible
(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ls
CustomDockerImages  inventory

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ansible -i inventory 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": {
```

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ansible -i inventory ubuntu1 -m setup | grep ansible_distribution
  "ansible_distribution": "Ubuntu",
  "ansible_distribution_file_parsed": true,
  "ansible_distribution_file_path": "/etc/os-release",
  "ansible_distribution_file_variety": "Debian",
  "ansible_distribution_major_version": "22",
  "ansible_distribution_release": "jammy",
  "ansible_distribution_version": "22.04",

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ansible -i inventory ubuntu1 -m setup | grep ansible_os_family
  "ansible_os_family": "Debian",

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ansible -i inventory ubuntu1 -m setup | grep ansible_pkg
  "ansible_pkg_mgr": "apt",

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ansible -i inventory ubuntu1 -m setup | grep python
  "ansible_python": {
    "executable": "/usr/bin/python3",
    "type": "cpython",
    "ansible_python_version": "3.10.12",
    "ansible_selinux_python_present": true,
    "discovered_interpreter_python": "/usr/bin/python3",

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$
```

Lab - Running your first ansible playbook

```
cd ~/devops-dec-2023
git pull
cd Day3/ansible
ansible-playbook -i inventory ping-playbook.yml
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ vim ping-playbook.yml

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ansible-playbook -i inventory ping-playbook.yml

PLAY [This playbook demonstrates using ping ansible module] ****
TASK [Gathering Facts] ****
ok: [ubuntu1]
ok: [ubuntu2]

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

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)-[~/devops-dec-2023/ansible]
$
```

Lab - Getting ansible help about any ansible module

To exit the exit, type letter 'q' without quotes

ansible-doc file

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
$ vim install-nginx-playbook.yml

[jegan@tektutor.org]-[~/devops-dec-2023/Day3/ansible]
$ ansible-doc file
> ANSIBLE.BUILTIN.FILE      (/usr/lib/python3/dist-packages/ansible/modules/file.py)

    Set attributes of files, directories, or symlinks and their targets.
    Alternatively, remove files, symlinks or directories. Many other modules support
    the same options as the `file` module - including [ansible.builtin.copy],
    [ansible.builtin.template], and [ansible.builtin.assemble]. For Windows targets,
    use the [ansible.windows.win_file] module instead.

ADDED IN: historical

OPTIONS (= is mandatory):

- access_time
    This parameter indicates the time the file's access time should be set to.
    Should be 'preserve' when no modification is required, 'YYYYMMDDHHMM.SS' when
    using default time format, or 'now'.
    Default is 'None' meaning that 'preserve' is the default for
    `state=[file,directory,link,hard]' and 'now' is default for `state=touch'.
    default: null
    type: str
    added in: version 2.7 of ansible-core

- access_time_format
    When used with `access_time`, indicates the time format that must be used.

- owner
    Name of the group that should own the filesystem object, as would be fed to
    'chown'.
    When left unspecified, it uses the current group of the current user unless you
    are root, in which case it can preserve the previous ownership.
    default: null
    type: str

- mode
    The permissions the resulting filesystem object should have.
    For those used to '/usr/bin/chmod' remember that modes are actually octal
    numbers. You must either add a leading zero so that Ansible's YAML parser knows
    it is an octal number (like '0644' or '01777') or quote it (like "'644'" or
    "'1777'") so Ansible receives a string and can do its own conversion from string
    into number.
    Giving Ansible a number without following one of these rules will end up with a
    decimal number which will have unexpected results.
    As of Ansible 1.8, the mode may be specified as a symbolic mode (for example,
    '+rwx' or 'u=rw,g=r,o=r').
    If 'mode' is not specified and the destination filesystem object *does not*
    exist, the default 'umask' on the system will be used when setting the mode for
    the newly created filesystem object.
    If 'mode' is not specified and the destination filesystem object *does* exist,
    the mode of the existing filesystem object will be used.
    Specifying 'mode' is the best way to ensure filesystem objects are created with
    the correct permissions. See CVE-2020-1736 for further details.
    default: null
    type: raw

- modification_time
```

```
jegan@tektutor: ~/devops-dec-2023
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
```

```
- state
  If 'absent', directories will be recursively deleted, and files or symlinks will
  be unlinked. In the case of a directory, if 'diff' is declared, you will see the
  files and folders deleted listed under 'path_contents'. Note that 'absent' will
  not cause 'file' to fail if the 'path' does not exist as the state did not
  change.
  If 'directory', all intermediate subdirectories will be created if they do not
  exist. Since Ansible 1.7 they will be created with the supplied permissions.
  If 'file', with no other options, returns the current state of 'path'.
  If 'file', even with other options (such as 'mode'), the file will be modified if
  it exists but will NOT be created if it does not exist. Set to 'touch' or use the
  [ansible.builtin.copy] or [ansible.builtin.template] module if you want to create
  the file if it does not exist.
  If 'hard', the hard link will be created or changed.
  If 'link', the symbolic link will be created or changed.
  If 'touch' (new in 1.4), an empty file will be created if the file does not
  exist, while an existing file or directory will receive updated file access and
  modification times (similar to the way 'touch' works from the command line).
  Default is the current state of the file if it exists, 'directory' if
  'recreate=yes', or 'file' otherwise.
  choices: [absent, directory, file, hard, link, touch]
  default: null
  type: str

- unsafe_writes
  Influence when to use atomic operation to prevent data corruption or inconsistent
  reads from the target filesystem object.
  By default this module uses atomic operations to prevent data corruption or
```

```
jegan@tektutor: ~/devops-dec-2023
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
```

```
be unlinked. In the case of a directory, if 'diff' is declared, you will see the
files and folders deleted listed under 'path_contents'. Note that 'absent' will
not cause 'file' to fail if the 'path' does not exist as the state did not
change.
If 'directory', all intermediate subdirectories will be created if they do not
exist. Since Ansible 1.7 they will be created with the supplied permissions.
If 'file', with no other options, returns the current state of 'path'.
If 'file', even with other options (such as 'mode'), the file will be modified if
it exists but will NOT be created if it does not exist. Set to 'touch' or use the
[ansible.builtin.copy] or [ansible.builtin.template] module if you want to create
the file if it does not exist.
If 'hard', the hard link will be created or changed.
If 'link', the symbolic link will be created or changed.
If 'touch' (new in 1.4), an empty file will be created if the file does not
exist, while an existing file or directory will receive updated file access and
modification times (similar to the way 'touch' works from the command line).
Default is the current state of the file if it exists, 'directory' if
'recreate=yes', or 'file' otherwise.
choices: [absent, directory, file, hard, link, touch]
default: null
type: str

- unsafe_writes
  Influence when to use atomic operation to prevent data corruption or inconsistent
  reads from the target filesystem object.
  By default this module uses atomic operations to prevent data corruption or
```

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]

Lab - Installing nginx using ansible playbook on the ubuntu ansible node containers

In this lab exercise, we will do the following

- install nginx in ubuntu ansible nodes
- configure nginx to pick html pages from our custom folder
- deploy custom web page that has machine specific details

```
cd ~/devops-dec-2023
git pull
```

```
cd Day3/ansible
ansible-playbook -i inventory install-nginx-playbook.yml
```

Expected output

```
(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ vim install-nginx-playbook.yml

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ansible-playbook -i inventory install-nginx-playbook.yml

PLAY [This playbook will install nginx, configures nginx to pick html from custom directory and deploy custom html web page] ****
TASK [Gathering Facts] ****
ok: [ubuntu2]
ok: [ubuntu1]

TASK [Install nginx in Ubuntu ansible 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)-[~/devops-dec-2023/Day3/ansible]
$ curl http://localhost:8001
curl: (56) Recv failure: Connection reset by peer

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ curl http://localhost:8002
curl: (56) Recv failure: Connection reset by peer

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$
```

Things to note

- green color indicates the task was successfully executed but ansible didn't modify anything on the ansible node to make it successfully
- yellow color indicates, the task was successfully executed but ansible had to make some changes on the ansible node to make it successsful

Understanding idempotency property of Ansible Configuration Management tool

```
cd ~/devops-dec-2023
git pull
cd Day3/ansible
ansible-playbook -i inventory install-nginx-playbook.yml
```

Expected output

The screenshot shows a terminal window with two tabs. The left tab is titled '(jegan@tektutor.org)-[~/devops-dec-2023]' and the right tab is titled 'jegan@tektutor: ~/devops-dec-2023'. The terminal content is as follows:

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
$ cd Day3/ansible
(jegan@tektutor.org)-[~/devops-dec-2023]
$ ansible-playbook -i inventory install-nginx-playbook.yml

PLAY [This playbook will install nginx, configures nginx to pick html from custom directory and deploy custom html web page] ****
TASK [Gathering Facts] ****
ok: [ubuntu1]
ok: [ubuntu2]

TASK [Install nginx in Ubuntu 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)-[~/devops-dec-2023/Day3/ansible]
$
```

Things to note

- if we rerun the ansible playbook, it would report everything in green color, this is because ansible will first compare the current state of the machine with the desired state of the machine as expected in the playbook. If there is deviation then ansible executes the task to match the actual state of the machine to desired machine state
- if the machine's current state already matches with the desired then ansible will simply the task as success in green color without executing the task, this property is called Idempotency
- the task with title "Gathering facts" is nothing but setup module which collects many facts about the ansible node. This modules gets invoked as the first task in every play that appears in the playbook.

We are not able to access the web page from nginx web server, as the nginx service wasn't started after installing. You could verify this by trying out the below commands

```
cd ~/devops-dec-2023
git pull
cd Day3/ansible
ansible -i inventory ubuntu1 -m shell -a "service nginx status"
ansible -i inventory ubuntu2 -m shell -a "service nginx status"
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
jegan@tektutor: ~/devops-dec-2023

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

TASK [Install nginx in Ubuntu 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]-[~/devops-dec-2023/Day3/ansible]
$ curl http://localhost:8002
curl: (56) Recv failure: Connection reset by peer

[jegan@tektutor.org]-[~/devops-dec-2023/Day3/ansible]
$ curl http://localhost:8001
curl: (56) Recv failure: Connection reset by peer

[jegan@tektutor.org]-[~/devops-dec-2023/Day3/ansible]
$ ansible -i inventory ubuntu1 -m shell -a "service nginx status"
ubuntu1 | FAILED | rc=3 >
* nginx is not runningnon-zero return code

[jegan@tektutor.org]-[~/devops-dec-2023/Day3/ansible]
$ ansible -i inventory ubuntu2 -m shell -a "service nginx status"
ubuntu2 | FAILED | rc=3 >
* nginx is not runningnon-zero return code

[jegan@tektutor.org]-[~/devops-dec-2023/Day3/ansible]
$
```

Now let's run the updated ansible playbook to start the nginx web server using shell module

```
cd ~/devops-dec-2023
git pull
cd Day3/ansible
ansible-playbook -i inventory install-nginx-playbook.yml
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
jegan@tektutor: ~/devops-dec-2023

$ vim install-nginx-playbook.yml

[jegan@tektutor.org]-[~/devops-dec-2023/Day3/ansible]
$ ansible-playbook -i inventory install-nginx-playbook.yml

PLAY [This playbook will install nginx, configures nginx to pick html from custom directory and deploy custom html web page] *****
TASK [Gathering Facts] *****
ok: [ubuntu2]
ok: [ubuntu1]

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

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

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]-[~/devops-dec-2023/Day3/ansible]
$ curl http://localhost:8001
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: sans-serif;
  }
</style>
```

The screenshot shows a terminal window with three tabs. The left tab shows the output of an Ansible playbook run, indicating successful completion with 3 tasks completed and 0 errors. The middle tab shows the HTML response from the first host (ubuntu1), which is the default Nginx welcome page. The right tab shows the HTML response from the second host (ubuntu2), which is identical to the first. Both responses include links to the Nginx documentation and support pages.

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
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]-[~/devops-dec-2023/Day3/ansible]
$ 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]-[~/devops-dec-2023/Day3/ansible]
$ curl http://localhost:8002
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
a href="http://nginx.com/">nginx.com</a>.</p>

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

[jegan@tektutor.org]-[~/devops-dec-2023/Day3/ansible]
$ curl http://localhost:8002
<!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>
```

Copying the default nginx config file from ubuntu1 container to local directory

```
cd ~/devops-dec-2023
cd Day3/ansible
docker cp ubuntu1:/etc/nginx/sites-available/default .
ls -l
```

Expected output

```

jegan@tektutor: ~/devops-dec-2023/Day3/ansible
jegan@tektutor: ~/devops-dec-2023/Day3/ansible

<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>
root@ubuntu1:/var/www/html# exit
exit

[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
$ ls
CustomDockerImages  install-nginx-playbook.yml  inventory  ping-playbook.yml

[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
$ docker cp ubuntu1:/etc/nginx/sites-available/default .

[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
$ ls -l
total 20
drwxr-xr-x 4 jegan jegan 4096 Dec  8 12:15 CustomDockerImages
-rw-r--r-- 1 jegan jegan 2412 May 30 2023 default
-rw-r--r-- 1 jegan jegan  334 Dec  8 14:25 install-nginx-playbook.yml
-rw-r--r-- 1 jegan jegan  218 Dec  8 12:38 inventory
-rw-r--r-- 1 jegan jegan  125 Dec  8 13:07 ping-playbook.yml

[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
$ 

```

Running the updated playbook

```

cd ~/devops-dec-2023
git pull
cd Day3/ansible
ansible-playbook -i inventory install-nginx-playbook.yml]
curl http://localhost:8001
curl http://localhost:8002

```

Expected output

```

jegan@tektutor: ~/devops-dec-2023/Day3/ansible
jegan@tektutor: ~/devops-dec-2023/Day3/ansible

[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
$ ansible-playbook -i inventory install-nginx-playbook.yml

PLAY [This playbook will install nginx, configures nginx to pick html from custom directory and deploy custom ht
ml web page] ***

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

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

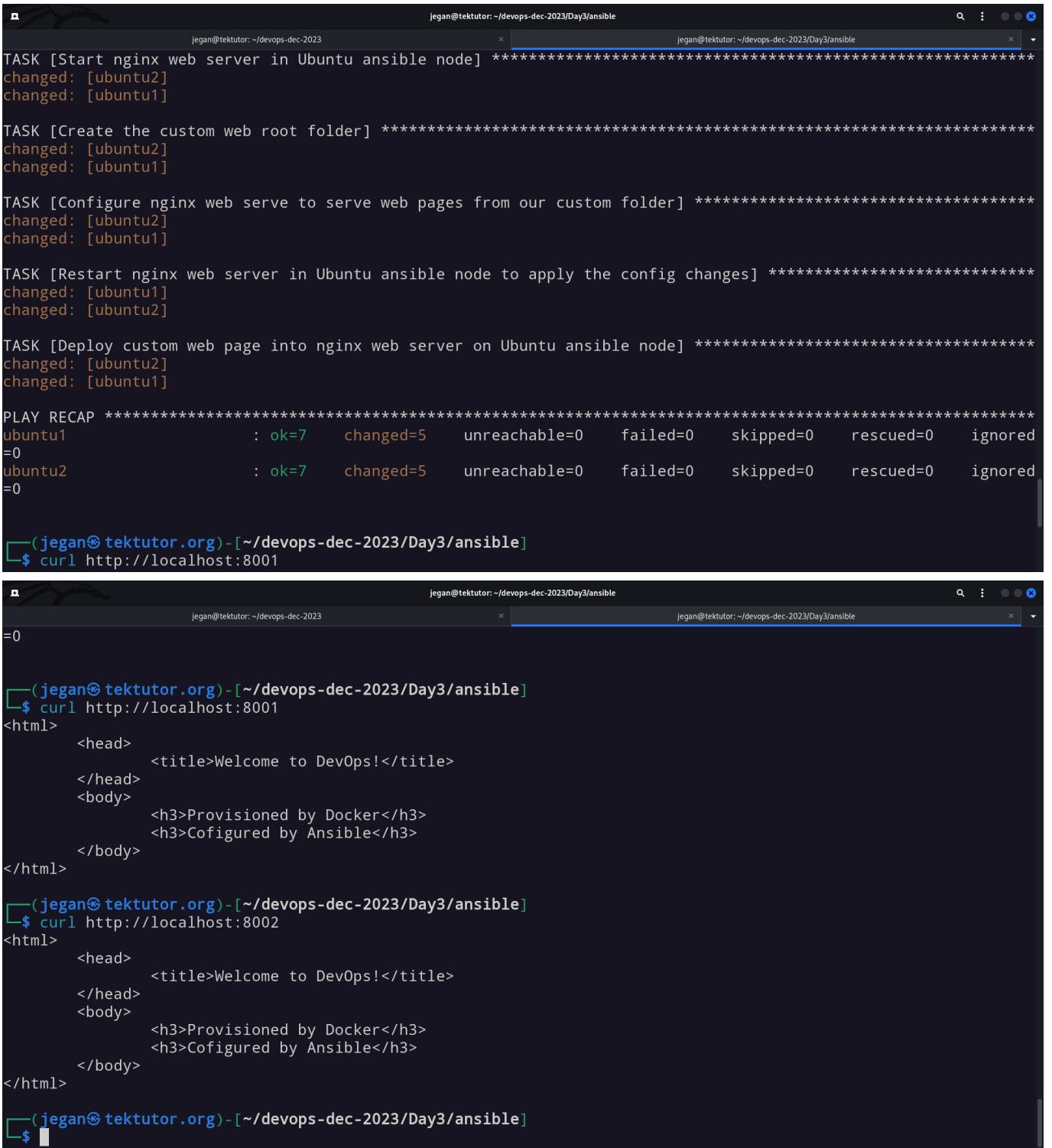
TASK [Start nginx web server in Ubuntu ansible node] ****
changed: [ubuntu2]
changed: [ubuntu1]

TASK [Create the custom web root folder] ****
changed: [ubuntu2]
changed: [ubuntu1]

TASK [Configure nginx web serve to serve web pages from our custom folder] ****
changed: [ubuntu2]
changed: [ubuntu1]

TASK [Restart nginx web server in Ubuntu ansible node to apply the config changes] ****
changed: [ubuntu1]

```



```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
TASK [Start nginx web server in Ubuntu ansible node] ****
changed: [ubuntu2]
changed: [ubuntu1]

TASK [Create the custom web root folder] ****
changed: [ubuntu2]
changed: [ubuntu1]

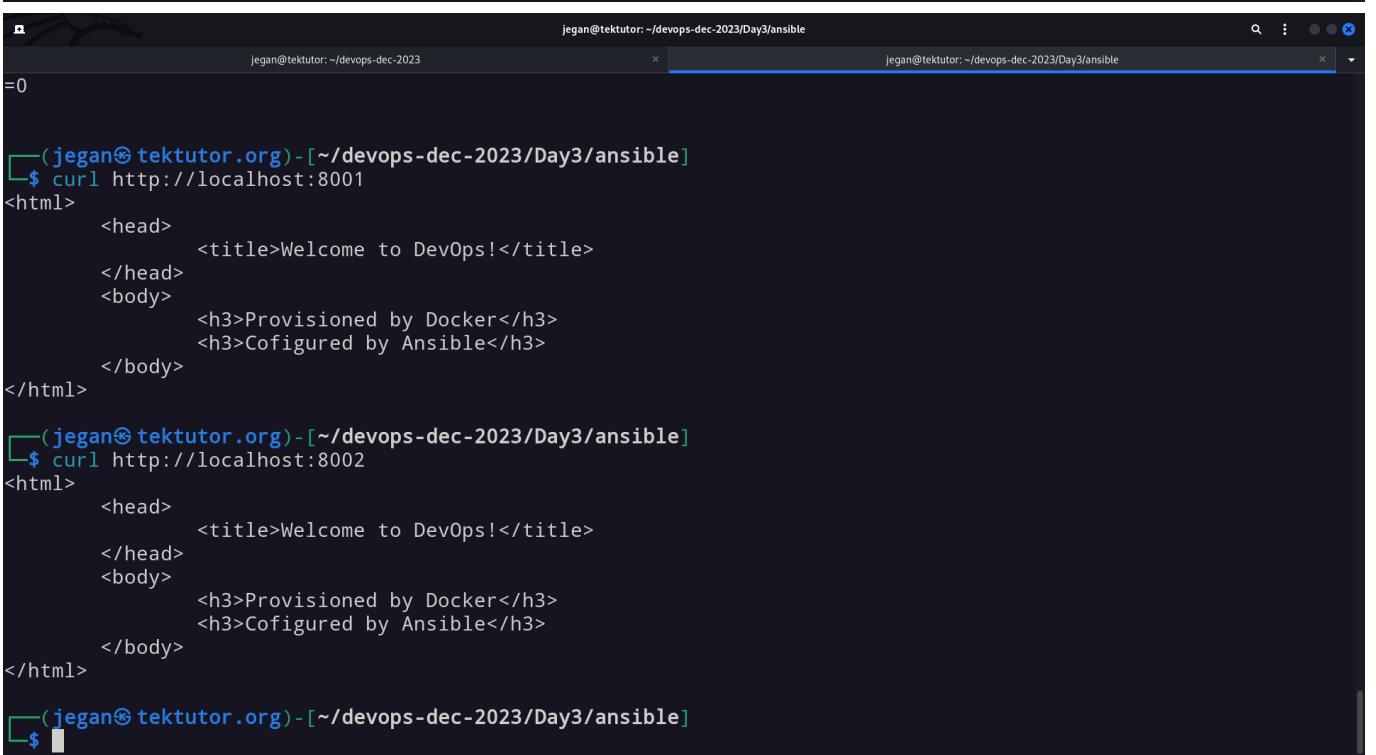
TASK [Configure nginx web serve to serve web pages from our custom folder] ****
changed: [ubuntu2]
changed: [ubuntu1]

TASK [Restart nginx web server in Ubuntu ansible node to apply the config changes] ****
changed: [ubuntu1]
changed: [ubuntu2]

TASK [Deploy custom web page into nginx web server on Ubuntu ansible node] ****
changed: [ubuntu2]
changed: [ubuntu1]

PLAY RECAP ****
ubuntu1          : ok=7    changed=5      unreachable=0      failed=0      skipped=0      rescued=0      ignored=0
ubuntu2          : ok=7    changed=5      unreachable=0      failed=0      skipped=0      rescued=0      ignored=0
```

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]\$ curl http://localhost:8001



```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
<html>
  <head>
    <title>Welcome to DevOps!</title>
  </head>
  <body>
    <h3>Provisioned by Docker</h3>
    <h3>Configured by Ansible</h3>
  </body>
</html>
```

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]\$ curl http://localhost:8002

```
<html>
  <head>
    <title>Welcome to DevOps!</title>
  </head>
  <body>
    <h3>Provisioned by Docker</h3>
    <h3>Configured by Ansible</h3>
  </body>
</html>
```

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]\$

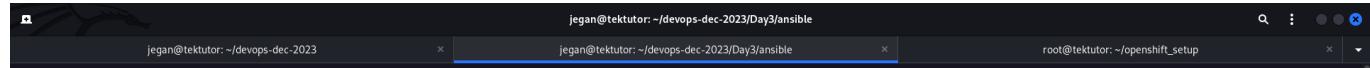
Things to note

- So far we have used the below ansible modules
 - ping - helps test if ansible is able to communicate with the ansible nodes
 - apt - helps in installing/uninstalling/updating/upgrading softwares in Ubuntu Linux distribution
 - file - helps in create folders, files with specific permissions
 - shell - used to execute any shell commands within the ansible nodes
 - copy - helps in copying files from local machine to ansible nodes and vice versa
 - setup - helps in collecting facts about ansible nodes

Lab - Using ansible template module to customize the html page

```
cd ~/devops-dec-2023
git pull
cd Day3/ansible
ansible-playbook -i inventory install-nginx-playbook-with-template-
module.yml
curl http://localhost:8001
curl http://localhost:8002
```

Expected output



```
(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ansible-playbook -i inventory install-nginx-playbook-with-template-module.yml

PLAY [This playbook will install nginx, configures nginx to pick html from custom directory and deploy custom ht
ml web page] ***

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

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

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

TASK [Create the custom web root folder] ****
ok: [ubuntu2]
ok: [ubuntu1]

TASK [Configure nginx web serve to serve web pages from our custom folder] ****
ok: [ubuntu1]
ok: [ubuntu2]

TASK [Restart nginx web server in Ubuntu ansible node to apply the config changes] ****
changed: [ubuntu2]
```

```
jegan@tektutor:~/devops-dec-2023/Day3/ansible
TASK [Configure nginx web serve to serve web pages from our custom folder] ****
ok: [ubuntu1]
ok: [ubuntu2]

TASK [Restart nginx web server in Ubuntu ansible node to apply the config changes] ****
changed: [ubuntu2]
changed: [ubuntu1]

TASK [Retrieve IP address of the ansible node] ****
changed: [ubuntu2]
changed: [ubuntu1]

TASK [Deploy custom web page into nginx web server on Ubuntu ansible node] ****
ok: [ubuntu2]
ok: [ubuntu1]

PLAY RECAP ****
ubuntu1 : ok=8    changed=3    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu2 : ok=8    changed=3    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]

```
jegan@tektutor:~/devops-dec-2023/Day3/ansible
[jegan@tektutor.org]-[~/devops-dec-2023/Day3/ansible]
$ curl http://localhost:8001
<html>
  <head>
    <title>Welcome to DevOps!</title>
  </head>
  <body>
    <h3>Provisioned by Docker</h3>
    <h3>Configured by Ansible</h3>
    <h3>Hostname - ubuntu1</h3>
    <h3>IP address - 172.17.0.2</h3>
    <h3>Linux - Ubuntu v22.04</h3>
  </body>
</html>

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ curl http://localhost:8002
<html>
  <head>
    <title>Welcome to DevOps!</title>
  </head>
  <body>
    <h3>Provisioned by Docker</h3>
    <h3>Configured by Ansible</h3>
    <h3>Hostname - ubuntu2</h3>
    <h3>IP address - 172.17.0.3</h3>
    <h3>Linux - Ubuntu v22.04</h3>
  </body>
</html>
```

Things to Note

- In this lab exercise, we learned how we could use template module to customize the index.html page with variables
- Using template modules makes the html page dynamic as it is configured with machine specific details
- Ansible template modules internally uses Jinja2 python library

Recommended reference <https://jinja.palletsprojects.com/en/3.1.x/>

Lab - Download jar from JFrog Artifactory server using Ansible playbook

Just in case, your JFrog Artifactory container is not running, you need to create it as shown below

```
docker run -d --name jfrog --hostname jfrog -p 8081-8082:8081-8082
releases-docker.jfrog.io/jfrog/artifactory-oss:latest
docker ps
```

You can access the JFrog Artifactory from your RPS Lab web browser

```
http://localhost:8081
```

You need to change the password to 'Rps@12345' without quotes.

Then you need to deploy the application jars as shown below

```
cd ~/devops-dec-2023
git pull
cd Day1/multi-module-project
mvn deploy
mvn clean
```

Now you can proceed as shown below

```
cd ~/devops-dec-2023
git pull
cd Day3/ansible
ls -l
ansible-playbook download-artifacts-from-jfrog-artifactory-using-
playbook.yml
ls -l
```

Expected output

The screenshot shows a terminal window with three tabs, all titled 'jegan@tektutor: ~/devops-dec-2023/Day3/ansible'. The first tab contains the command '\$ ansible-playbook download-artifacts-from-jfrog-artifactory-using-playbook.yml' and its output, which includes several warning messages about inventory and hosts lists, followed by the play execution details. The second tab shows the command '\$ ls -l' and its output, listing files including 'CustomDockerImages', 'dal-1.0.0.jar', 'default', 'download-artifacts-from-jfrog-artifactory-using-playbook.yml', 'index.html', and 'index.html.j2'. The third tab is empty.

```
(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ansible-playbook download-artifacts-from-jfrog-artifactory-using-playbook.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not
match 'all'

PLAY [This playbook will download application jar file from JFrog Artifactory] ****
TASK [Gathering Facts] ****
ok: [localhost]

TASK [Download jar from JFrog Artifactory server] ****
changed: [localhost]

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

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ls -l
total 40
drwxr-xr-x 4 jegan jegan 4096 Dec  8 12:15 CustomDockerImages
-rw-r--r-- 1 jegan jegan 1778 Dec  8 15:49 dal-1.0.0.jar
-rw-r--r-- 1 jegan jegan 2511 Dec  8 14:35 default
-rw-r--r-- 1 jegan jegan  335 Dec  8 15:44 download-artifacts-from-jfrog-artifactory-using-playbook.yml
-rw-r--r-- 1 jegan jegan  151 Dec  8 15:04 index.html
-rw-r--r-- 1 jegan jegan  345 Dec  8 15:14 index.html.j2
```

Lab - Passing extra variables to ansible playbook

```
cd ~/devops-dec-2023
git pull
cd Day3/ansible
curl http://localhost:8002
ansible-playbook install-nginx-playbook-with-template-module.yml -e
greeting_msg=Welcome
curl http://localhost:8002
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ansible-playbook install-nginx-playbook-with-template-module.yml -e greeting_msg=Welcome
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not
match 'all'

PLAY [This playbook will install nginx, configures nginx to pick html from custom directory and deploy custom ht
ml web page] ***
skipping: no hosts matched

PLAY RECAP ****

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ansible-playbook -i inventory install-nginx-playbook-with-template-module.yml -e greeting_msg=Welcome
PLAY [This playbook will install nginx, configures nginx to pick html from custom directory and deploy custom ht
ml web page] ***

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

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

TASK [Start nginx web server in Ubuntu ansible node] ****
changed: [ubuntu1]

jegan@tektutor: ~/devops-dec-2023/Day3/ansible
(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ curl http://localhost:8002
<html>
  <head>
    <title>Welcome</title>
  </head>
  <body>
    <h3>Provisioned by Docker</h3>
    <h3>Configured by Ansible</h3>
    <h3>Hostname - ubuntu2</h3>
    <h3>IP address - 172.17.0.3</h3>
    <h3>Linux - Ubuntu v22.04</h3>
  </body>
</html>

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ curl http://localhost:8001
<html>
```

Demo - Ansible Tower

Installing Ansible Tower Opensource (AWX) https://medium.com/@jegan_50867/installing-ansible-tower-awx-e46d5231357d

When you login to Ansible Tower Web Interface looks as shown below

The screenshot shows the Ansible Tower dashboard. On the left, a sidebar menu includes Views (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics, Subscription Usage), Resources (Templates, Credentials, Projects, Inventories, Hosts), Access (Organizations, Users, Teams), and Administration (Credential Types). The main dashboard area displays six summary boxes: 1 Hosts, 0 Failed hosts, 1 Inventories, 0 Inventory sync failures, 1 Projects, and 0 Project sync failures. Below these are tabs for Job status, Recent Jobs, and Recent Templates, followed by a chart titled 'Job Runs' showing activity from November 8 to December 8. The chart has a y-axis for 'Job Runs' (0 to 5) and an x-axis for 'Date'. A legend indicates 'Past month', 'All job types', and 'All jobs'.

Create a new project (GitHub Integration)

The screenshot shows the Ansible Tower Projects page. The sidebar menu is identical to the dashboard, with the 'Projects' option selected under Resources. The main area displays a table for the 'Demo Project'. The columns are Name (sorted), Status (Successful), Type (Git), and Revision (347e44f). There are also 'Actions' buttons for each row. Navigation links at the bottom indicate 1-1 of 1 items and 1 of 1 page.

Projects > TekTutor Training Repository

Edit Details

Name *

Description

Organization *

Execution Environment

Source Control Type *

Content Signature Validation Credential

Type Details

Source Control URL *

Source Control Branch/Tag/Commit

Source Control Refspec

Source Control Credential

Options

Clean Delete Track submodules Update Revision on Launch Allow Branch Override

Option Details

Cache Timeout

Save **Cancel**

Projects > TekTutor Training Repository

Details

Last Job Status Successful

Name	TekTutor Training Repository	Organization	Default
Source Control Type	Git	Source Control Revision	4b2a07b
Cache Timeout	0 Seconds	Project Base Path	/var/lib/awx/projects
Created	12/8/2023, 4:44:25 PM by admin	Last Modified	12/8/2023, 4:44:25 PM by admin
Enabled Options	Update revision on job launch		

Edit **Sync** **Delete**

```

10 PLAY [Install content with ansible-galaxy command if necessary]      16:44:30
17 TASK [Warn about disabled content sync] ****
18 skipping: [localhost]                                              16:44:30
19
20
21 TASK [End play due to disabled content sync] ****
22 skipping: [localhost]                                              16:44:30
23
24 TASK [Fetch galaxy roles from requirements.(yml/yaml)] ****
25 [WARNING]: Unable to find
26 '/var/lib/awx/projects/_8_tektutor_training_repository/roles' in expected
27 paths (use -vvvv to see paths)
28 skipping: [localhost]
29
30 TASK [Fetch galaxy collections from collections/requirements.(yml/yaml)] ****
31 [WARNING]: Unable to find
32 '/var/lib/awx/projects/_8_tektutor_training_repository/collections' in
33 expected paths (use -vvvv to see paths)
34 skipping: [localhost]
35
36 PLAY RECAP ****
37 localhost : ok=3    changed=1   unreachable=0   failed=0   skipped=3   rescued=0   ignored=0

```

We need to add a new credential and paste the private key we created in the RPS machine and save it.

Credentials > ACMPrivateKey

Edit Details

Name *	Description	Organization
ACMPrivateKey		
Credential Type *		
Machine		
Type Details		
Username	Password	<input type="checkbox"/> Prompt on launch
jegan		
SSH Private Key		
Drag a file here or browse to upload \$encrypted\$		
Signed SSH Certificate		
Drag a file here or browse to upload		

The screenshot shows the Ansible Automation Platform web interface. The left sidebar is dark-themed and includes sections for Views (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics, Subscription Usage), Resources (Templates, Credentials, Projects, Inventories, Hosts), Access (Organizations, Users, Teams), and Administration (Credential Types). The 'Credentials' section is currently selected. The main content area is titled 'Credentials' and displays a list of three entries:

Name	Type	Actions
ACMPrivateKey	Machine	
Ansible Galaxy	Ansible Galaxy/Automation Hub API Token	
Demo Credential	Machine	

Pagination at the bottom indicates 1-3 of 3 items, with 1 of 1 page.

We need to create an inventory

The screenshot shows the 'Edit details' page for an inventory named 'Ubuntu'. The left sidebar is identical to the previous screenshot. The main content area shows the 'Edit details' form for the 'Ubuntu' inventory:

Edit details

Name *	Ubuntu	Description
Variables	VARIABLES JSON	
<pre>1 --- 2 ansible_port: 2001 3 ansible_user: root 4 ansible_host: 192.168.1.127</pre>		

At the bottom are 'Save' and 'Cancel' buttons.

The screenshot shows the Red Hat Ansible Automation Platform web interface. The left sidebar is titled "Red Hat Ansible Automation Platform" and contains the following sections:

- Views**: Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics, Subscription Usage.
- Resources**: Templates, Credentials, Projects, **Inventories** (selected), Hosts.
- Access**: Organizations, Users, Teams.
- Administration**: Credential Types.

The main content area shows the "Host details" for "Ubuntu2". The URL in the browser bar is <https://tektrutor-ansible-tower-aap.apps.ocp.tektrutor-ocp-labs/#/inventories/inventory/2/hosts/3/details>. The page includes the following information:

- Details Tab**: Shows the host is active ("On") and has been last modified on 12/8/2023 at 4:56:57 PM by "admin".
- Name**: Ubuntu2
- Activity**: Shows a green checkmark and five red error icons.
- Created**: 12/8/2023, 4:49:49 PM by admin
- Last Modified**: 12/8/2023, 4:56:57 PM by admin
- Variables**: YAML tab selected, showing variables:

```
1 ansible_port: 2002
2 ansible_user: root
3 ansible_host: 192.168.1.127
```
- Actions**: Edit, Delete buttons.

The screenshot displays two consecutive pages from the Red Hat Ansible Automation Platform web interface.

Inventories Page:

- Header:** Shows three tabs: "tekutor-ansible-tower-", "Ansible Automation Plat", and "10. Credentials — Autom". The URL is "https://tekutor-ansible-tower-aap.apps.ocp.tektutor-ocp-labs/#/inventories".
- Sidebar:** "Inventories" is selected under the "Resources" category.
- Table:** Shows one inventory entry: "Demo Inventory" (Name), "Disabled" (Sync Status), "Inventory" (Type), and "Default" (Organization). Actions include edit and delete.
- Pagination:** Shows "1-1 of 1 items" and a single-page navigation bar.

Hosts Page:

- Header:** Shows three tabs: "tekutor-ansible-tower-", "Ansible Automation Plat", and "10. Credentials — Autom". The URL is "https://tekutor-ansible-tower-aap.apps.ocp.tektutor-ocp-labs/#/inventories/inventory/2/hosts".
- Sidebar:** "Inventories" is selected under the "Resources" category.
- Table:** Shows two hosts: "Ubuntu1" and "Ubuntu2". Both have "On" status in the "Actions" column.
- Pagination:** Shows "1-2 of 2 items" and a single-page navigation bar.

We need to create a Job Template to configure and run an Ansible Playbook within Ansible Tower

The screenshot shows the Ansible Automation Platform interface. The left sidebar is collapsed. The main area displays a table titled "Templates". The table has columns for Name, Type, Organization, and Last Ran. A single row is visible: "Demo Job Template" (Type: Job Template, Organization: Default). At the top of the table are search and filter fields, and buttons for "Add" and "Delete". Below the table is a pagination control showing "1-1 of 1 items".

The screenshot shows the Ansible Automation Platform interface. The left sidebar is collapsed. The main area displays the "Details" page for a template named "Install0 Nginx Playbook". The page includes tabs for Back to Templates, Details, Access, Notifications, Schedules, Jobs, and Survey. The "Details" tab is selected. The template details are as follows:

Name	Value	Type	Value	Organization	Value
Inventory	DockerInventory	Project	TekTutor Training Repository	Execution Environment	Default execution environment
Playbook	Day3/ansible/install-nginx-playbook-with-template-module.yml	Forks	0	Verbosity	0 (Normal)
Timeout	0	Show Changes	Off	Job Slicing	1
Created	12/8/2023, 4:52:17 PM by admin	Last Modified	12/8/2023, 4:52:17 PM by admin		
Credentials	SSH:ACMPPrivateKey				

Below the table, there is a "Variables" section with tabs for YAML and JSON. A code editor window shows a single line of YAML: "1 ---". At the bottom of the page are buttons for "Edit", "Launch", and "Delete".

We need to launch the Job to run the Ansible Playbook

The screenshot shows the Red Hat Ansible Automation Platform web interface. The left sidebar is dark-themed and includes sections for Views, Dashboard, Jobs (which is selected), Schedules, Activity Stream, Workflow Approvals, Host Metrics, Subscription Usage, Resources (Templates, Credentials, Projects, Inventories, Hosts), Access (Organizations, Users, Teams), and Administration (Credential Types). The main content area has a light background and displays a job titled "Install0 Nginx Playbook" with a status of "Pending". The "Output" tab is active, showing a "Stdout" dropdown set to "Stdout" and a search bar. Below the search bar is a button labeled "Unfollow". A large, empty text area is present, with a small icon of three stacked cubes in the center and the text "Waiting for job output..." below it. At the top of the main content area, there are navigation links: "Back to Jobs", "Details", and "Output". On the far right of the main content area, there are several small icons for filtering and sorting. The top of the browser window shows tabs for "tektnur-ansible-tower...", "Ansible Automation Plat...", and "10: Credentials — Autom...". The address bar indicates the URL is "https://tektnur-ansible-tower-aap.apps.ocp.tektnur-ocp-labs/#/jobs/playbook/2/output". The top right corner of the browser window shows a user profile for "admin".

The screenshot shows two separate sessions of the Red Hat Ansible Automation Platform interface. Both sessions are viewing the output of a playbook named 'Install0 Nginx Playbook'. The top session shows the first run, which completed successfully with 1 task, 9 hosts, and an elapsed time of 00:00:21. The bottom session shows the second run, also successful, with 9 tasks, 2 hosts, and an elapsed time of 00:00:21. Both sessions include a sidebar with navigation links for Views, Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics, Subscription Usage, Resources, Templates, Credentials, Projects, Inventories, Hosts, Access, Organizations, Users, Teams, Administration, and Credential Types.

```

 0 Identity added: /runner/artifacts/10/ssh_key_data (jegan@tektutor)
 1 [DEPRECATION WARNING]: Specifying a list of dictionaries for vars is deprecated
 2 in favor of specifying a dictionary. This feature will be removed in version
 3 2.18. Deprecation warnings can be disabled by setting
 4 deprecation_warnings=False in ansible.cfg.
 5
 6 PLAY [This playbook will install nginx, configures nginx to pick html from custom directory and deploy custom html web page] ***
 7
 8   TASK [Gathering Facts] *****
 9     ok: [Ubuntu2]
10     ok: [Ubuntu1]
11
12   TASK [Install nginx in Ubuntu ansible node] *****
13     ok: [Ubuntu1]
14     ok: [Ubuntu2]
15
16   TASK [Start nginx web server in Ubuntu ansible node] *****
17     changed: [Ubuntu1]
18     changed: [Ubuntu2]
19
20   TASK [Create the custom web root folder] *****

```



```

25
26   ok: [Ubuntu1]
27
28   TASK [Restart nginx web server in Ubuntu ansible node to apply the config changes] ***
29     changed: [Ubuntu1]
30     changed: [Ubuntu2]
31
32   TASK [Retrieve IP address of the ansible node] *****
33     changed: [Ubuntu1]
34     changed: [Ubuntu2]
35
36   TASK [debug] *****
37     skipping: [Ubuntu1]
38     skipping: [Ubuntu2]
39
40   TASK [Deploy custom web page into nginx web server on Ubuntu ansible node] *****
41     changed: [Ubuntu2]
42     changed: [Ubuntu1]
43
44   PLAY RECAP *****
45   Ubuntu1      : ok=8    changed=4    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
46   Ubuntu2      : ok=8    changed=4    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

```

Lab - Using Ansible vault to protect sensitive data like login credentials

When prompts for password, type 'Rps@12345' without quotes.

```

cd ~/devops-dec-2023
git pull
cd Day3/ansible
ansible-vault create jfrog-credentials.yml
cat jfrog-credentials.yml

```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
jegan@tektutor: ~/devops-dec-2023/Day3/ansible

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ls
CustomDockerImages           install-nginx-playbook-with-template-module.yml
default                       install-nginx-playbook.yml
download-artifacts-from-jfrog-artifactory-using-playbook.yml inventory
index.html                     ping-playbook.yml
index.html.j2

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ vim download-artifacts-from-jfrog-artifactory-using-playbook.yml

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ansible-vault create jfrog-credentials.yml
New Vault password:
Confirm New Vault password:

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ cat jfrog-credentials.yml
$ANSIBLE_VAULT;1.1;AES256
65613530346537343435636230383164633232303732316330663339356338386638366532663562
3030396536303061653231656266383531643237303038370a343234613331656461663930303664
3730373265316631373934313236386236363631326634316662353261323939334393231653337
6530666138313031300a38386636663336433346637353238636364656266643035333733303632
3938363066373062323639653761663164356664376332633262663564323062666665323165
6364346534376539643639646165333631326364373335663666

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
```

Now, let's use the jfrog-crentials which is ansible vault protected in our ansible playbook, when prompts for password type 'RpsW12345' without quotes.

```
cd ~/devops-dec-2023
git pull
cd Day3/ansible
cat download-artifacts-from-jfrog-artifactory-using-playbook-with-vault.yml
ansible-playbook download-artifacts-from-jfrog-artifactory-using-playbook-
with-vault.yml
cat jfrog-credentials.yml
ansible-playbook download-artifacts-from-jfrog-artifactory-using-playbook-
with-vault.yml --ask-vault-pass
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day3/ansible
[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
$ ansible-playbook download-artifacts-from-jfrog-artifactory-using-playbook-with-vault.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not
match 'all'
ERROR! Attempting to decrypt but no vault secrets found

[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
$ cat jfrog-credentials.yml
$ANSIBLE_VAULT;1.1;AES256
65613530346537343435636230383164633232303732316330663339356338386638366532663562
3030396536303061653231656266383531643237303038370a343234613331656461663930303664
37303732653166313739343132363862363636313266343166623532613239393334393231653337
6530666138313031300a383866366633364334663735323863636465626664303533373303632
39383630663730623236396537616631643566643763326332633262663564323062666665323165
6364346534376539643639646165333631326364373335663666

[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
$ ansible-playbook download-artifacts-from-jfrog-artifactory-using-playbook-with-vault.yml --ask-vault-pass
Vault password:
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not
match 'all'

PLAY [This playbook will download application jar file from JFrog Artifactory] ****
TASK [Gathering Facts] ****
ok: [localhost]

jegan@tektutor: ~/devops-dec-2023/Day3/ansible
[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
$ cat jfrog-credentials.yml
$ANSIBLE_VAULT;1.1;AES256
65613530346537343435636230383164633232303732316330663339356338386638366532663562
3030396536303061653231656266383531643237303038370a343234613331656461663930303664
37303732653166313739343132363862363636313266343166623532613239393334393231653337
6530666138313031300a383866366633364334663735323863636465626664303533373303632
39383630663730623236396537616631643566643763326332633262663564323062666665323165
6364346534376539643639646165333631326364373335663666

[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
$ ansible-playbook download-artifacts-from-jfrog-artifactory-using-playbook-with-vault.yml --ask-vault-pass
Vault password:
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not
match 'all'

PLAY [This playbook will download application jar file from JFrog Artifactory] ****
TASK [Gathering Facts] ****
ok: [localhost]

TASK [Download jar from JFrog Artifactory server] ****
changed: [localhost]

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

[jegan@tektutor.org]-(~/devops-dec-2023/Day3/ansible]
```

The screenshot shows a terminal window with three tabs. The central tab displays the output of an Ansible playbook run on a local host. The output includes task details like 'ok' and 'changed' counts, and a play recap summary. Below the play recap, the terminal shows the directory listing of the playbook directory, including files like 'CustomDockerImages', 'dal-1.0.0.jar', and various configuration and template files. The bottom tab shows a command prompt ready for input.

```
TASK [Download jar from JFrog Artifactory server] *****
changed: [localhost]

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

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$ ls -l
total 48
drwxr-xr-x 4 jegan jegan 4096 Dec  8 12:15 CustomDockerImages
-rw-r--r-- 1 jegan jegan 1778 Dec  8 17:08 dal-1.0.0.jar
-rw-r--r-- 1 jegan jegan 2511 Dec  8 14:35 default
-rw-r--r-- 1 jegan jegan  391 Dec  8 17:07 download-artifacts-from-jfrog-artifactory-using-playbook-with-vault.yml
-rw-r--r-- 1 jegan jegan  335 Dec  8 17:02 download-artifacts-from-jfrog-artifactory-using-playbook.yml
-rw-r--r-- 1 jegan jegan  151 Dec  8 15:04 index.html
-rw-r--r-- 1 jegan jegan  345 Dec  8 15:14 index.html.j2
-rw-r--r-- 1 jegan jegan 1072 Dec  8 15:21 install-nginx-playbook-with-template-module.yml
-rw-r--r-- 1 jegan jegan  821 Dec  8 15:01 install-nginx-playbook.yml
-rw-r--r-- 1 jegan jegan  218 Dec  8 12:38 inventory
-rw----- 1 jegan jegan  484 Dec  8 17:03 jfrog-credentials.yml
-rw-r--r-- 1 jegan jegan  125 Dec  8 13:07 ping-playbook.yml

(jegan@tektutor.org)-[~/devops-dec-2023/Day3/ansible]
$
```

Other interesting features supported by Ansible vault

```
ansible-vault view jfrog-credentials.yml
ansible-vault edit jfrog-credentials.yml
ansible-vault decrypt jfrog-credentials.yml
ansible-vault encrypt jfrog-credentials.yml
ansible-vault rekey jfrog-credentials.yml
```