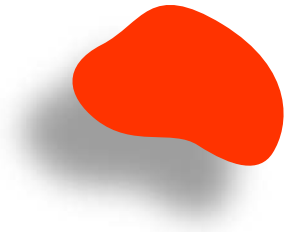




The background is a dark red overlay on a collage of images. The collage includes a laptop keyboard in the top left, a pen in the middle left, a stack of pink paper in the bottom center, and a donut with sprinkles on a plate in the bottom right. There are also some abstract orange shapes like a circle, a triangle, and a cross scattered around.

Ansible Certification

Red Hat Certified Specialist in Ansible
Automation exam



Mumshad Mannambeth

Founder & Trainer, KodeKloud



I am an IT Solutions Architect and specializes in Cloud, Automation and DevOps. I am passionate about learning new technology and teaching. I believe the best way to learn is to learn by doing and in a fun way. I have authored multiple courses on DevOps, Cloud and Automation technologies and I teach over 120,000 Students world wide. My courses focus on providing students with an interactive and hands-on experience in learning new technology that makes learning really interesting.



Total students	Courses	Reviews
121,387	13	31,553

Courses you're teaching



Certified Kubernetes Administrator (CKA) with Docker

Mumshad Mannambeth



Kubernetes Certified Application Developer

Mumshad Mannambeth, Kode K...



Chef for the Absolute Beginners - DevOps

Mumshad Mannambeth, Yoges...



DevOps - The Pre-Requisite Course

Mumshad Mannambeth



{K}ODE{K}LOUD

www.kodekloud.com

{K}ODE{K}LOUD

www.kodekloud.com

Overview

Objectives

What you need to know

Study points for the exam

We recommend that candidates become a [Red Hat Certified Engineer \(RHCE®\)](#) or, at a minimum, a [Red Hat Certified System Administrator \(RHCSA®\)](#) before attempting this exam, but neither is required.

To help you prepare, the exam objectives highlight the task areas you can expect to see covered in the exam. Red Hat reserves the right to add, modify, and remove exam objectives. Such changes will be made public in advance.

You should be able to:

- Understand core components of Ansible
 - Inventories
 - Modules
 - Variables
 - Facts
 - Plays
 - Playbooks
 - Configuration files
- Install and configure an Ansible control node
 - Install required packages
 - Create a static host inventory file
 - Create a configuration file



The background is a dark red overlay on a light-colored collage. The collage includes a laptop keyboard in the top left, a pen in the middle left, a stack of pink paper in the bottom center, and a donut with sprinkles on a plate in the bottom right. There are also some abstract orange shapes and a small orange 'x' mark.

+ Certification ▾ Details

Red Hat Certified Specialist in Ansible
Automation exam



Audience

- Red Hat Certified Specialist in Ansible Automation (EX407)
- Red Hat Certified Engineer exam for Red Hat Enterprise Linux 8(EX294) (Partly)



Pre-Requisites

- No Prior Certification Required
- Red Hat® Enterprise Linux® 7.5 and Ansible 2.7

RHEL vs CentOS





Exam Format

- Hands-on Performance Based
- 4 Hours



Signup Format

- Classroom
- On-Site Exam
- Individual



The background is a dark red overlay on a collage of images. The collage includes a laptop keyboard in the top left, a pen and glasses in the middle left, a stack of pink paper in the bottom center, and a donut with sprinkles on a plate in the bottom right. There are also some orange geometric shapes like triangles, a circle, and a plus sign scattered around.

Pre-Requisites

Ansible Pre-Requisites

Ansible for the Absolute Beginners

- Setup Basic Lab
- YAML
- Inventory
- Playbooks
- Variables
- Modules
- Loops





Linux Pre-Requisites

- SSH Keys, Authorized Keys
- Users, Groups
- Package Managers
- Services
- Cron
- SELinux
- Devices, Filesystems, LVM
- Firewalls
- Archiving



The Curriculum

Red Hat Certified Ansible Specialist

Core Components

- ✓ Inventories
- ✓ Modules
- ✓ Variables
- ✓ Plays
- ✓ Playbooks
- Configuration Files
- Facts

- Install and Configure Ansible Control Node
- Configure Ansible Managed Nodes
- Create simple shell scripts that run ad hoc Ansible commands
- Dynamic inventories
- Ansible Plays and Playbooks
- Ansible Modules
- Customized Configuration Files

Notes

- Do not use the code in the slides as is (Things are hidden at times). Refer to the references and git repo for the actual code and working samples.
- Code might get copied in a different format.

```
- name: Deploy web application
hosts: server1
tasks:
  - name: Install dependencies
    << code hidden >>

  - name: Install MySQL Database
    << code hidden >>

  - name: Start MySQL Service
    << code hidden >>

  - name: Install Python Flask Dependencies
    << code hidden >>

  - name: Run web-server
    << code hidden >>
```



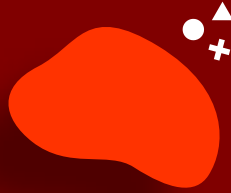
The Curriculum

Red Hat Certified Ansible Specialist

Core Components

- ✓ Inventories
- ✓ Modules
- ✓ Variables
- ✓ Facts
- ✓ Plays
- ✓ Playbooks
- Configuration Files

- Install and Configure Ansible Control Node
- Configure Ansible Managed Nodes
- Create simple shell scripts that run ad hoc Ansible commands
- Dynamic inventories
- Ansible Plays and Playbooks
- Ansible Modules
- Customized Configuration Files



Ansible Configuration Files

Ansible Configuration Files

```
/etc/ansible/ansible.cfg
```

```
[defaults]
```

```
[inventory]
```

```
[privilege_escalation]
```

```
[paramiko_connection]
```

```
[ssh_connection]
```

```
[persistent_connection]
```

```
[colors]
```

Ansible Configuration Files

```
/etc/ansible/ansible.cfg
```

```
[defaults]
```

```
inventory          = /etc/ansible/hosts
```

```
log_path           = /var/log/ansible.log
```

```
library            = /usr/share/my_modules/
```

```
roles_path         = /etc/ansible/roles
```

```
action_plugins     = /usr/share/ansible/plugins/action
```

```
gathering          = implicit
```

```
# SSH timeout
```

```
timeout            = 10
```

```
forks              = 5
```

```
[inventory]
```

```
enable_plugins     = host_list, virtualbox, yaml, constructed
```

Ansible Configuration Files

`/etc/ansible/ansible.cfg`



`/opt/web-playbooks`



`/opt/web-playbooks/ansible.cfg`

`/opt/db-playbooks`



`/opt/db-playbooks/ansible.cfg`

`/opt/network-playbooks`



`/opt/network-playbooks/ansible.cfg`

Ansible Configuration Files

/etc/ansible/ansible.cfg

/opt/ansible-web.cfg



/opt/web-playbooks

/opt/db-playbooks

/opt/network-playbooks



/opt/db-playbooks/ansible.cfg

/opt/network-playbooks/ansible.cfg

```
$ANSIBLE_CONFIG=/opt/ansible-web.cfg ansible-playbook playbook.yml
```

Ansible Configuration Files

1 /opt/ansible-web.cfg



/opt/web-playbooks



2 /opt/web-playbooks/ansible.cfg

4 /etc/ansible/ansible.cfg



/opt/db-playbooks



/opt/db-playbooks/ansible.cfg

3 .ansible.cfg



/opt/network-playbooks



/opt/network-playbooks/ansible.cfg

1 \$ANSIBLE_CONFIG=/opt/ansible-web.cfg

Ansible Configuration Files

`/etc/ansible/ansible.cfg`



`/opt/web-playbooks`



`/opt/db-playbooks`



`/opt/network-playbooks`



`/opt/storage-playbooks`



`/etc/ansible/ansible.cfg`

`gathering = implicit`

`ANSIBLE_GATHERING=explicit`

Ansible Configuration Variables

```
$ ANSIBLE_GATHERING=explicit ansible-playbook playbook.yml
```

```
$ export ANSIBLE_GATHERING=explicit  
$ ansible-playbook playbook.yml
```

```
/opt/web-playbooks/ansible.cfg  
gathering                = explicit
```

View Configuration

```
$ ansible-config list # Lists all configurations
```

```
$ ansible-config view # Shows the current config file
```

```
$ ansible-config dump # Shows the current settings
```

```
$ export ANSIBLE_GATHERING=explicit  
$ ansible-config dump | grep GATHERING  
DEFAULT_GATHERING(env: ANSIBLE_GATHERING) = explicit
```



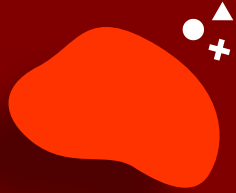
The Curriculum

Red Hat Certified Ansible Specialist

Core Components

- ✓ Inventories
- ✓ Modules
- ✓ Variables
- ✓ Plays
- ✓ Playbooks
- Configuration Files
- Facts

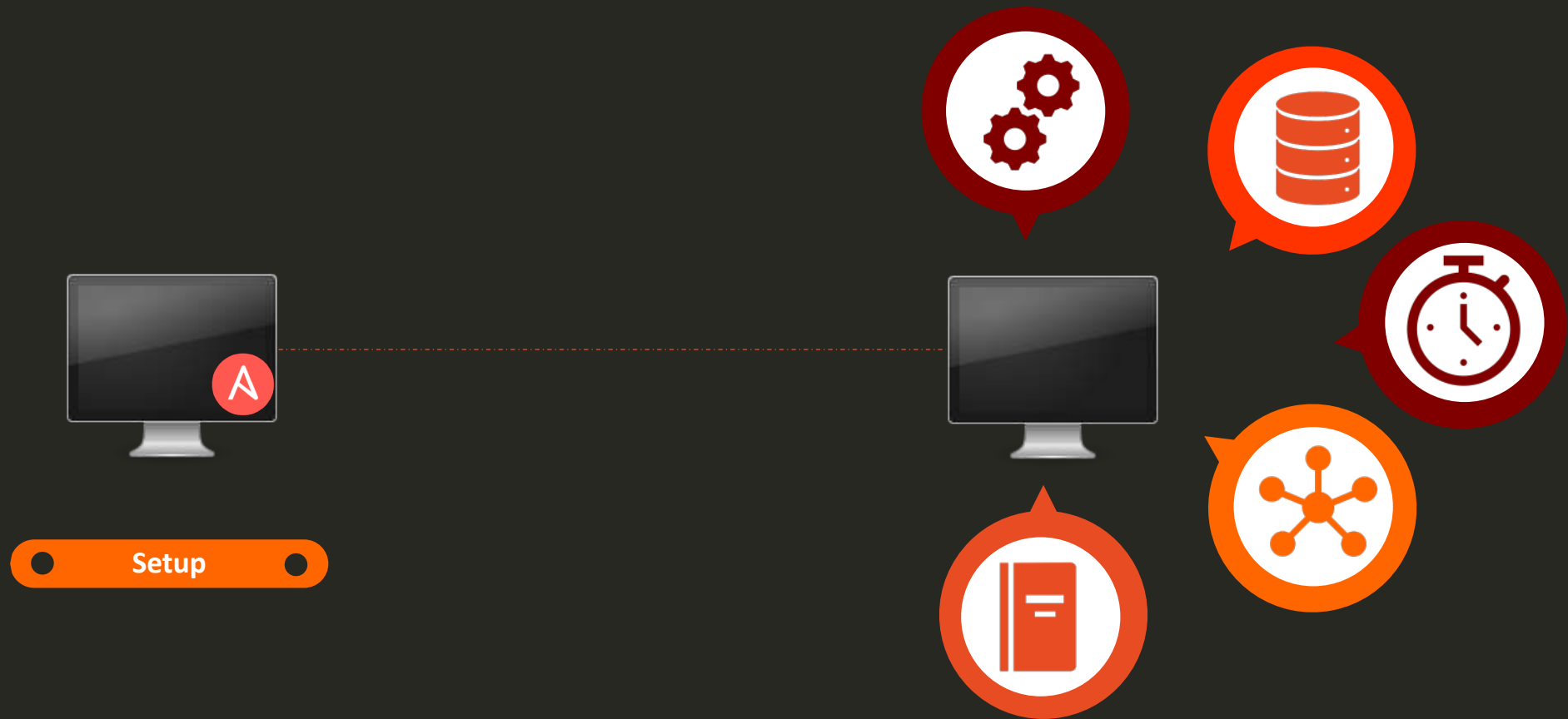
- Install and Configure Ansible Control Node
- Configure Ansible Managed Nodes
- Create simple shell scripts that run ad hoc Ansible commands
- Dynamic inventories
- Ansible Plays and Playbooks
- Ansible Modules
- Customized Configuration Files



Ansible

FACTS

FACTS



```
- name: Print hello message
hosts: all
tasks:
- debug:
    msg: Hello from Ansible!
```

PLAY [Print hello message]

TASK [Gathering Facts]

ok: [web2]

ok: [web1]

TASK [debug]

ok: [web1] => {

"msg": "Hello from Ansible!"

}

ok: [web2] => {

"msg": "Hello from Ansible!"

}

```
- name: Print hello message
hosts: all
tasks:
- debug:
    var: ansible_facts
```

PLAY [Reset nodes to previous state]

TASK [Gathering Facts]

ok: [web2]

ok: [web1]

TASK [debug] *****

```
ok: [web1] => {
  "ansible_facts": {
    "all_ipv4_addresses": [
      "172.20.1.100"
    ],
    "architecture": "x86_64",
    "date_time": {
      "date": "2019-09-07",
    },
    "distribution": "Ubuntu",
    "distribution_file_variety": "Debian",
    "distribution_major_version": "16",
    "distribution_release": "xenial",
    "distribution_version": "16.04",
    "dns": {
      "nameservers": [
        "127.0.0.11"
      ],
    },
    "fqdn": "web1",
    "hostname": "web1",
    "interfaces": [
      "lo",
      "eth0"
    ],
    "machine": "x86_64",
    "memfree_mb": 72,
    "memory_mb": {
      "real": {
        "free": 72,
        "total": 985,
        "used": 913
      }
    }
  }
}
```

```
- name: Print hello message
hosts: all
tasks:
- debug:
    var: ansible_facts
```

```
"interfaces": [
  "lo",
  "eth0"
],
"machine": "x86_64",
"memfree_mb": 72,
"memory_mb": {
  "real": {
    "free": 72,
    "total": 985,
    "used": 913
  },
  },
"memtotal_mb": 985,
"module_setup": true,
"mounts": [
  {
    "block_available": 45040,
    "block_size": 4096,
    "block_total": 2524608,
    "block_used": 2479568,
  },
],
"nodename": "web1",
"os_family": "Debian",
"processor": [
  "",
  "GenuineIntel",
  "Intel(R) Core(TM) i9-9980HK CPU @ 2.40GHz",
],
"processor_cores": 2,
"processor_count": 1,
"processor_threads_per_core": 1,
"processor_vcpus": 2,
"product_name": "VirtualBox",
"product_serial": "",
"product_uuid": "18A31B5D-FAC9-445F-9B6F-95B4B587F485",
"product_version": "1.2",
}
}
```

```
---
- name: Print hello message
  hosts: all
  gather_facts: no
  tasks:
    - debug:
      var: ansible_facts
```

```
PLAY [Print hello message]
*****

TASK [debug]
*****
ok: [web1] => {
  "ansible_facts": {}
}
ok: [web2] => {
  "ansible_facts": {}
}
```


```
---
- name: Print hello message
  hosts: all
  gather_facts: no
  tasks:
    - debug:
      var: ansible_facts
```

```
PLAY [Print hello message]
*****

TASK [debug]
*****
ok: [web1] => {
  "ansible_facts": {}
}
ok: [web2] => {
  "ansible_facts": {}
}
```

/etc/ansible/ansible.cfg

```
# plays will gather facts by default, which contain information about
# smart - gather by default, but don't regather if already gathered
# implicit - gather by default, turn off with gather_facts: False
# explicit - do not gather by default, must say gather_facts: True
gathering = implicit
```



```
---  
- name: Print hello message  
  hosts: web1  
  tasks:  
    - debug: ansible_facts
```

```
/etc/ansible/hosts
```

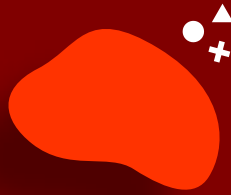
```
web1  
web2
```



The Curriculum

Red Hat Certified Ansible Specialist

- Core Components
- Install and Configure Ansible Control Node
 - Install Required Packages
 - Create a Static Host Inventory File
 - Create a Configuration File
- Configure Ansible Managed Nodes
- Create simple shell scripts that run ad hoc Ansible commands
- Dynamic inventories
- Ansible Plays and Playbooks
- Ansible Modules
- Customized Configuration Files
- Variables and Facts
- Roles



Ansible

Install

Control Node



Redhat or CentOS – `$ sudo yum install ansible`



Fedora – `$ sudo dnf install ansible`



Ubuntu – `$ sudo apt-get install ansible`



PIP – `$ sudo pip install ansible`

Additional Options:

- Install from source on GIT
- Build RPM yourself



Ansible Control Machine

- Playbooks
- Inventory
- Modules



Control Machine - Linux Only

https://docs.ansible.com/ansible/latest/installation_guide/

Install Control Node on Redhat or CentOS



Redhat or CentOS –

```
$ sudo yum install ansible
```



Install via PIP

Install pip if not present

```
$ sudo yum install epel-release
```

```
$ sudo yum install python-pip
```

Install Ansible using pip

```
$ sudo pip install ansible
```

Upgrade Ansible using pip

```
$ sudo pip install --upgrade ansible
```

Install Specific Version of Ansible using pip

```
$ sudo pip install ansible==2.4
```

Ansible Inventory



Redhat or CentOS –

```
sudo yum install ansible
```

/etc/ansible/hosts

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110
```

/opt/my-playbook/hosts

```
web1 ansible_host=192.168.1.100
web2 ansible_host=192.168.1.101
```

Ansible Configuration File



Redhat or CentOS –

```
sudo yum install ansible
```

/etc/ansible/ansible.cfg

```
[defaults]
inventory          = /etc/ansible/hosts
log_path           = /var/log/ansible.log

library            = /usr/share/my_modules/
roles_path          = /etc/ansible/roles
action_plugins     = /usr/share/ansible/plugins/action

gathering           = implicit

# SSH timeout
timeout            = 10

display_skipped_hosts = True
nocolor            = 1

forks               = 5
```

/opt/my-playbook/ansible.cfg

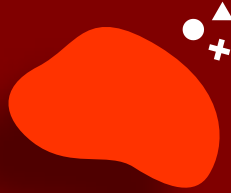
```
[defaults]
gathering           = explicit
```



The Curriculum

Red Hat Certified Ansible Specialist

- Core Components
- Install and Configure Ansible Control Node
- Configure Ansible Managed Nodes
 - Create and Distribute SSH Keys
 - Configure Privilege Escalation on Managed Nodes
 - Validate using Adhoc Commands
- Create simple shell scripts that run ad hoc Ansible commands
- Dynamic inventories
- Ansible Plays and Playbooks
- Ansible Modules
- Customized Configuration Files
- Variables and Facts
- Roles



Ansible

Creating and Distributing SSH Keys

Inventory File

```
/etc/ansible/hosts
```

```
web1 ansible_host=172.20.1.100 ansible_ssh_pass=Passw0rd  
web2 ansible_host=172.20.1.101 ansible_ssh_pass=Passw0rd
```

```
▶ ssh-keygen  
id_rsa id_rsa.pub
```



Private Key



Public Key

```
▶ ssh -i id_rsa user1@server1  
Successfully Logged In!
```

```
▶ cat ~/.ssh/authorized_keys  
ssh-rsa AAAAB3NzaC1yc...KhtUBfoTz1BqR  
V1NThv0o4opzEwRQo1mWx user1
```



```
▶ ssh-keygen  
id_rsa id_rsa.pub
```



Private Key Public Lock

```
▶ ssh -i id_rsa user1@server1  
Successfully Logged In!
```

```
▶ ssh-copy-id -i id_rsa user1@server1  
Number of key(s) added: 1
```

```
▶ cat ~/.ssh/authorized_keys  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAKhtURfoTz1BqR  
V1NThvOo4opzEwRQo1mWx user1
```



Inventory File

```
/etc/ansible/hosts
```

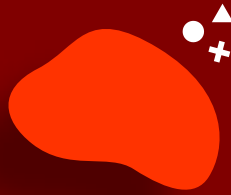
```
web1 ansible_host=172.20.1.100 ansible_user=user1 ansible_ssh_private_key_file=/some-path/private-key
web2 ansible_host=172.20.1.101 ansible_user=user1 ansible_ssh_private_key_file=/some-path/private-key
```



The Curriculum

Red Hat Certified Ansible Specialist

- Core Components
- Install and Configure Ansible Control Node
- Configure Ansible Managed Nodes
 - Create and Distribute SSH Keys
 - Configure Privilege Escalation on Managed Nodes
 - Validate using Adhoc Commands
- Create simple shell scripts that run ad hoc Ansible commands
- Dynamic inventories
- Ansible Plays and Playbooks
- Ansible Modules
- Customized Configuration Files
- Variables and Facts
- Roles



Ansible

AdHoc

Commands



```
playbook.yml
```

```
---  
- name: Ping Servers  
  hosts: all  
  tasks:  
    - :
```



```
ansible-playbook playbook.yml
```

```
ansible -m ping
```



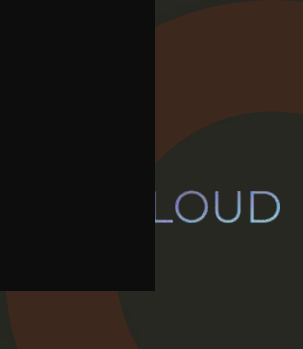


```
playbook.yml
```

```
---  
- name: Ping Servers  
  hosts:  
  tasks:  
  - :
```

```
▶ ansible-playbook playbook.yml
```

```
▶ ansible -m ping all
```

```
web2 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}  
web1 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}
```



LOUD

```
▶ ansible -m ping all
```

```
web2 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}  
web1 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}
```

```
▶ ansible -a 'cat /etc/hosts' all
```

```
web1 | CHANGED | rc=0 >>  
127.0.0.1 localhost  
::1 localhost ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
172.20.1.100 web1
```

```
web2 | CHANGED | rc=0 >>  
127.0.0.1 localhost  
::1 localhost ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters
```

```
▶ ansible -a 'yum install nginx' all  
--become  
--become-user nginx
```

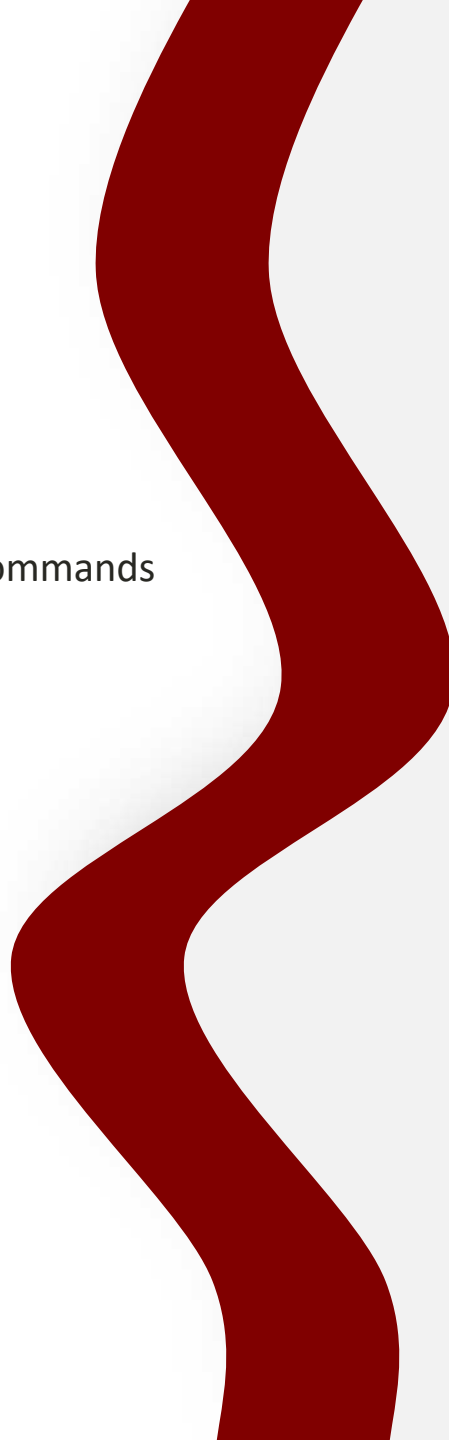


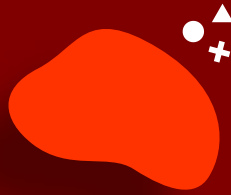


The Curriculum

Red Hat Certified Ansible Specialist

- Core Components
- Install and Configure Ansible Control Node
- Configure Ansible Managed Nodes
- Create simple shell scripts that run ad hoc Ansible commands
- Dynamic inventories
- Ansible Plays and Playbooks
- Ansible Modules
- Customized Configuration Files
- Variables and Facts
- Roles
- Ansible Vault
- Documentation






Ansible

Shell

Scripts



```
$ ansible -m ping all
```

```
$ ansible -a 'cat /etc/hosts' all
```

```
$ export ANSIBLE_GATHERING=explicit  
$ ansible-playbook playbook.yml
```




```
shell-script.sh
```

```
export ANSIBLE_GATHERING=explicit
```

```
ansible -m ping all
```

```
ansible -a 'cat /etc/hosts' all
```

```
ansible-playbook playbook.yml
```

```
▶ sh shell-script.sh
```

```
▶ chmod 755 shell-script.sh
```

```
▶ ./shell-script.sh
```



The Curriculum

Red Hat Certified Ansible Specialist

- Core Components
- Install and Configure Ansible Control Node
- Configure Ansible Managed Nodes
 - Create and Distribute SSH Keys
 - Configure Privilege Escalation on Managed Nodes
 - Validate using Adhoc Commands
- Create simple shell scripts that run ad hoc Ansible commands
- Dynamic inventories
- Ansible Plays and Playbooks
- Ansible Modules
- Customized Configuration Files
- Variables and Facts
- Roles



Ansible Privilege Escalation

Users



root



admin



admin



developer

nginx



monitor



mysql



Users Workflow



admin

```
▶ ssh -i id_ras admin@server1  
Successfully Logged In!
```



nginx

```
▶ sudo yum install nginx  
Package Installed!
```

```
▶ su nginx
```

```
# Configure nginx
```



mysql

```
▶ su mysql
```

```
# Configure MySQL
```



root

Install Packages

Become Super user (sudo)

Become Method – sudo (pfexec, doas, ksu, runas)

Become another user

Users Workflow

inventory

```
lamp-dev1 ansible_host=172.20.1.100 ansible_user=admin
```

playbook

```
---  
- name: Install nginx  
  hosts: all  
  tasks:  
    - yum:  
      name: nginx  
      state: latest
```

Permission Denied



Install Packages

root

Become Super user (sudo)

Become Method – sudo (pfexec, doas, ksu, runas)

Become another user

Become Super User

inventory

```
lamp-dev1 ansible_host=172.20.1.100 ansible_user=admin
```

playbook

```
---  
- name: Install nginx  
  become: yes  
  hosts: all  
  tasks:  
    - yum:  
      name: nginx  
      state: latest
```

Package Installed!



Install Packages

root

Become Super user (sudo)

Become Method – sudo (pfexec, doas, ksu, runas)

Become another user

Become Method

inventory

```
lamp-dev1 ansible_host=172.20.1.100 ansible_user=admin
```

playbook

```
---  
- name: Install nginx  
  become: yes  
  become_method: doas  
  hosts: all  
  tasks:  
    - yum:  
      name: nginx  
      state: latest
```

Package Installed!



Install Packages

root

Become Super user (sudo)

Become Method – sudo (pfexec, doas, ksu, runas)

Become another user

Become Another User

inventory

```
lamp-dev1 ansible_host=172.20.1.100 ansible_user=admin
```

playbook

```
---  
- name: Install nginx  
  become: yes  
  become_user: nginx  
  hosts: all  
  tasks:  
    - yum:  
      name: nginx  
      state: latest
```

Package Installed!



Install Packages

root

Become Super user (sudo)

Become Method – sudo (pfexec, doas, ksu, runas)

Become another user

Inventory File



Install Packages

root

inventory

```
lamp-dev1 ansible_host=172.20.1.100 ansible_user=admin ansible_become=yes  ansible_become_user=nginx
```

playbook

```
---  
- name: Install nginx
```

```
hosts: all  
tasks:  
- yum:  
    name: nginx  
    state: latest
```

Package Installed!

Become Super user (sudo)

Become Method – sudo (pfexec, doas, ksu, runas)

Become another user

Configuration File

```
/etc/ansible/ansible.cfg
```

```
become                = True
become_method         = doas
become_user           = nginx
```

```
inventory
```

```
lamp-dev1 ansible_host=172.20.1.100 ansible_user=admin ansible_become=yes  ansible_become_user=nginx
```

```
playbook
```

```
---
- name: Install nginx

hosts: all
tasks:
- yum:
    name: nginx
    state: latest
```



root

Install Packages

Become Super user (sudo)

Become Method – sudo (pfexec, doas, ksu, runas)

Become another user

Command Line



Install Packages

root

```
/etc/ansible/ansible.cfg
```

```
become                = True
become_method         = doas
become_user           = nginx
```

```
inventory
```

```
lamp-dev1 ansible_host=172.20.1.100 ansible_user=admin ansible_become=yes  ansible_become_user=nginx
```

```
playbook
```

```
command line
```

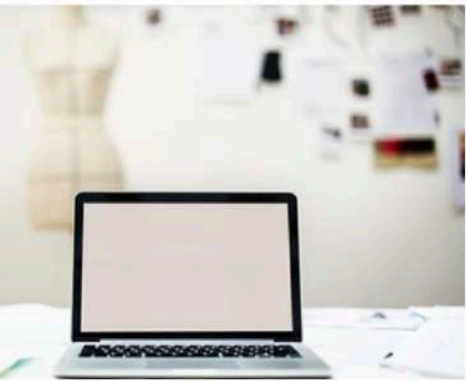
```
$ ansible-playbook --become --become-method=doas --become-user=nginx --ask-become-pass
```





KodeKloud Project

Product List



Macbook Pro

Purchase MB at the lowest price **100\$**



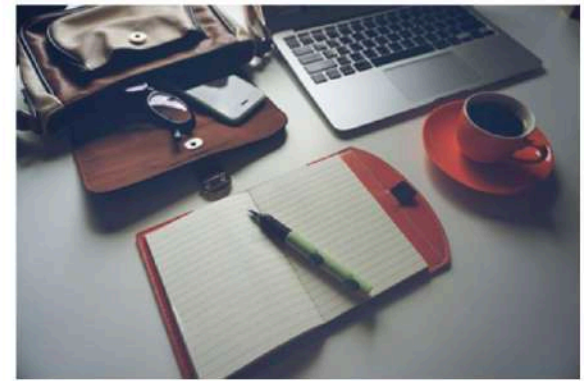
Drone

Purchase Multifunctional drones **200\$**



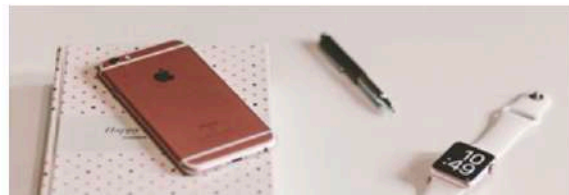
VR

Explore our VR Devices



Macbook air

Purchase MB at the lowest price **500\$**



Linux Apache MariaDB Php

1



Install Firewall

2



Install httpd
Configure httpd
Configure Firewall
Start httpd

3



Install MariaDB
Configure MariaDB
Start MariaDB
Configure Firewall
Configure Database
Load Data

4



Install Php
Configure Code



Install Firewall



Install httpd
Configure httpd
Configure Firewall
Start httpd



Install MariaDB
Configure MariaDB
Start MariaDB
Configure Firewall
Configure Database
Load Data



Install php
Download Code
Test



Install Firewall

Install MariaDB

Configure MariaDB

Start MariaDB

Configure Firewall

Configure Database

Load Data



Install httpd

Install php

Configure Firewall

Configure httpd

Start httpd

Download Code

Test





Install Firewall

```
$ sudo yum install firewalld
$ sudo service firewalld start
$ sudo systemctl enable firewalld
```



Install MariaDB

Configure MariaDB

Start MariaDB

```
$ sudo yum install mariadb-server
```

```
$ sudo vi /etc/my.cnf # configure the file with the right port
```

```
$ sudo service mariadb start
```

```
$ sudo systemctl enable mariadb
```

Configure Firewall

```
$ sudo firewall-cmd --permanent --zone=public --add-port=3306/tcp
```

```
$ sudo firewall-cmd --reload
```

Configure Database

```
$ mysql
MariaDB > CREATE DATABASE ecomdb;
MariaDB > CREATE USER 'ecomuser'@'localhost' IDENTIFIED BY 'ecompassword';
MariaDB > GRANT ALL PRIVILEGES ON *.* TO 'ecomuser'@'localhost';
MariaDB > FLUSH PRIVILEGES;
```

Load Data

```
$ mysql < db-load-script.sql
```



Install httpd

Install php

Configure Firewall

Configure httpd

Start httpd

Download Code

Test

```
$ sudo yum install -y httpd php php-mysql
```

```
$ sudo firewall-cmd --permanent --zone=public --add-port=80/tcp
```

```
$ sudo firewall-cmd --reload
```

```
$ sudo vi /etc/httpd/conf/httpd.conf #  
# configure DirectoryIndex to use index.php instead of index.html
```

```
$ sudo service httpd start
```

```
$ sudo systemctl enable httpd
```

```
$ sudo yum install -y git
```

```
$ git clone https://github.com/<application>.git /var/www/html/
```

```
# Update index.php to use the right database address, name and credentials
```

```
$ curl http://localhost
```

Deployment Model- Single Node



Deployment Model- Multi Node

172.20.1.101



172.20.1.102



```
$ mysql
MariaDB > CREATE DATABASE ecomdb;
MariaDB > CREATE USER 'ecomuser'@'172.20.1.102' IDENTIFIED BY 'ecompassword';
MariaDB > GRANT ALL PRIVILEGES ON *.* TO 'ecomuser'@'172.20.1.102';
MariaDB > FLUSH PRIVILEGES;
```

<?php

```
$link = mysqli_connect('172.20.1.101', 'ecomuser', 'ecompassword');

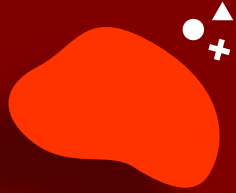
if ($link) {
    $res = mysqli_query($link, "select * from products;");
    while ($row = mysqli_fetch_assoc($res)) { ?>
```

HTML

```
98      <!--=====End Slider area=====-->
99
100     <section class="best_business_area row">
101         <div class="check_tittle wow fadeInUp" data-wow-delay="0.7s" id="product-list">
102             <h2>Product List</h2>
103         </div>
104         <div class="row it_works">
105             <?php
106
107                 $link = mysqli_connect('172.20.1.101', 'ecomuser', 'ecompassword', 'ecomdb');
108
109                 if ($link) {
110                     $res = mysqli_query($link, "select * from products;");
111                     while ($row = mysqli_fetch_assoc($res)) { ?>
112
113                         <div class="col-md-3 col-sm-6 business_content">
114                             <?php echo '' ?>
115                             <div class="media">
116                                 <div class="media-left">
117
118                                 </div>
119                                 <div class="media-body">
120                                     <a href="#"><?php echo $row['Name'] ?></a>
121                                     <p>Purchase <?php echo $row['Name'] ?> at the lowest price <span><?php echo $row['Price'] ?>$</span></p>
122                                 </div>
123                             </div>
124                         </div>
125
126                     <?php
127                         }
128                 }
```




KodeKloud Project



Ansible

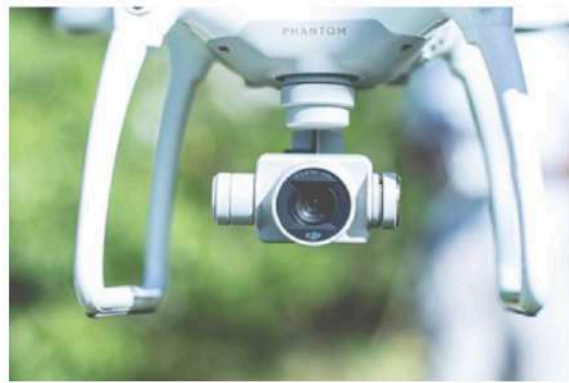
Complete Playbook

Product List



Macbook Pro

Purchase MB at the lowest price **100\$**



Drone

Purchase Multifunctional drones **200\$**



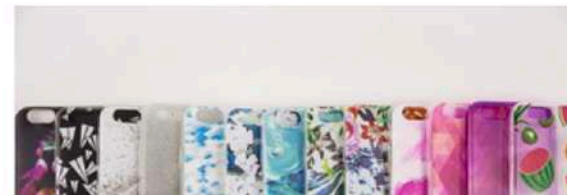
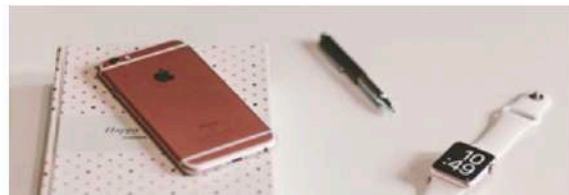
VR

Explore our VR Devices



Macbook air

Purchase MB at the lowest price **500\$**



Web application

- Web Server



- MySQL Database



<https://github.com/mmumshad/simple-webapp>

Web application

1



Identify Server

2



Python

3



Install
Configure
Start

4



Install Flask

5



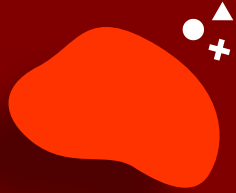
Source Code

6



Run





Ansible Playbook Visualization


```
[web_servers]
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101
web3 ansible_host=172.20.1.102
```

playbook.yml

```
-
  hosts: web_servers
  tasks:
    - name: Copy index.html to remote servers
      copy:
        src: index.html
        dest: /var/www/nginx-default/index.html
```

```
[web_servers]
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101
web3 ansible_host=172.20.1.102
```



Variable Interpolation

```
inventory_hostname=web1
ansible_host=172.20.1.100

ansible_facts=<Host Facts>
```

Gather Facts

```
inventory_hostname=web2
ansible_host=172.20.1.101

ansible_facts=<Host Facts>
```

```
inventory_hostname=web3
ansible_host=172.20.1.102

ansible_facts=<Host Facts>
```

Execute Playbook

```
playbook.yml
-
  hosts: web_servers
  tasks:
    - name: Copy index.html to remote servers
      copy:
        src: index.html
        dest: /var/www/nginx-default/index.html
```

```
playbook.yml
-
  hosts: web_servers
  tasks:
    - name: Copy index.html to remote servers
      copy:
        src: index.html
        dest: /var/www/nginx-default/index.html
```

```
playbook.yml
-
  hosts: web_servers
  tasks:
    - name: Copy index.html to remote servers
      copy:
        src: index.html
        dest: /var/www/nginx-default/index.html
```

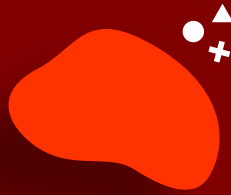
Create Subprocess

web1

web2

web3





Ansible

FAQ

YAML

```
- name: Gather facts

gather_facts: yes          no
gather_facts: true         false
gather_facts: TRUE         FALSE
gather_facts: True         False
```

YAML

```
---  
- name: Print dns server  
  hosts: all  
  tasks:  
    - debug:  
      msg: Hello
```

{{ }}

```
- name: Print dns server
  hosts: all
  tasks:
    - debug:
        msg: "{{ dns_server_ip }}"
        var: dns_server_ip

    when: ansible_host != 'web'
    with_items: "{{ db_servers }}"
```

```
msg: "{{ dns_server_ip }}"
```

```
msg: The DNS server is {{ dns_server_ip }}
```

ansible_ssh_pass or ansible_password

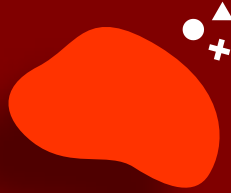
```
/etc/ansible/hosts
```

```
web1 ansible_host=172.20.1.100 ansible_ssh_pass=Passw0rd  
web2 ansible_host=172.20.1.101 ansible_ssh_pass=Passw0rd
```

```
/etc/ansible/hosts
```

```
web1 ansible_host=172.20.1.100 ansible_password=Passw0rd  
web2 ansible_host=172.20.1.101 ansible_password=Passw0rd
```



Ansible

Playbook Run

Options

Check Mode or Dry Run

```
---  
- name: Install httpd  
  hosts: all  
  tasks:  
    - yum:  
      name: httpd  
      state: installed
```

```
$ ansible-playbook playbook.yml --check
```

Start at

```
---  
- name: Install httpd  
  
  hosts: all  
  tasks:  
    - name: Install httpd  
      yum:  
        name: httpd  
        state: installed  
  
    - name: Start httpd service  
      service:  
        name: httpd  
        state: started
```

```
$ ansible-playbook playbook.yml --start-at-task "Start httpd service"
```

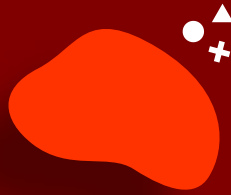
Tags

```
---  
- name: Install httpd  
  tags: install and start  
  hosts: all  
  tasks:  
    - yum:  
      name: httpd  
      state: installed  
      tags: install  
    - service:  
      name: httpd  
      state: started  
      tags: start httpd service
```

```
$ ansible-playbook playbook.yml --tags "install"
```

```
$ ansible-playbook playbook.yml --skip-tags "install"
```





Ansible

Modules

Packages

playbook

```
---
- name: Install web on CentOS
  hosts: all
  tasks:
    - yum:
        name: httpd
        state: installed
```

playbook

```
---
- name: Install web on Ubuntu
  hosts: all
  tasks:
    - apt:
        name: apache2
        state: installed
```

playbook

```
---
- name: Install web on Any Host
  hosts: all
  tasks:
    - package:
        name: httpd
        state: installed
```


Service

playbook

```
---  
- name: Start httpd service  
  hosts: all  
  tasks:  
    - service:  
      name: httpd  
      state: started  
      enabled: yes
```

Firewall Rules

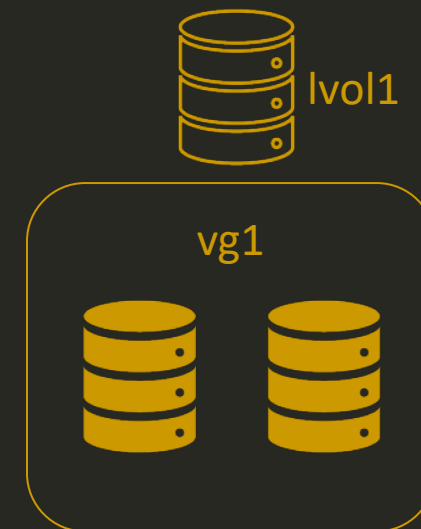
```
playbook
```

```
---  
- name: Add Firewall rule  
  hosts: all  
  tasks:  
    - firewall:  
  
      port: 8080/tcp  
      service: http  
      source: 192.0.0.0/24  
  
      zone: public  
  
      state: enabled  
  
      permanent: yes  
      immediate: yes
```

Storage

playbook

```
---  
- hosts: all  
  tasks:  
    - name: Create LVM Volume Group  
      lvg:  
        vg: vg1  
        pvs: /dev/sdb1,/dev/sdb2  
  
    - name: Create LVM Volume  
      lvol:  
        vg: vg1  
        lv: lvol1  
        size: 2g
```



Filesystem

playbook

```
---  
- hosts: all  
  tasks:  
    - name: Create Filesystem  
      filesystem:  
  
        fstype: ext4  
        dev: /dev/vg1/lvol1  
  
        opts: -cc  
  
    - name: Mount Filesystem  
      mount:  
  
        fstype: ext4  
        src: /dev/vg1/lvol1  
        path: /opt/app  
        state: mounted
```



File

playbook

```
---  
- hosts: all  
  tasks:  
    - name: Create Directory  
      file:  
        path: /opt/app/web  
        state: directory  
  
    - name: Create File  
      file:  
        path: /opt/app/web/index.html  
        state: touch  
        owner: app-owner  
        group: app-owner  
        mode: '0644'
```



/opt/app/web/index.html



/opt/app/web



/opt/app



lv01

vg1



Archive

playbook

```
---
- hosts: all
  tasks:
    - name: Compress a folder
      archive:
        path: /opt/app/web
        dest: /tmp/web.gz
        format: zip|tar|bz2|xz|gz

    - name: Uncompress a folder
      unarchive:
        src: /tmp/web.gz
        dest: /opt/app/web
        remote_src: yes
```

Cron

playbook

```
---  
- hosts: all  
  tasks:  
    - name: Create a scheduled task  
      cron:  
        name: Run daily health report  
        job: sh /opt/scripts/health.sh  
  
        month: 2  
        day: 19  
        hour: 8  
        minute: 10
```

Cron

playbook

```
---
- hosts: all
  tasks:
    - name: Create a scheduled task
      cron:
        name: Run daily health report
        job: sh /opt/scripts/health.sh

        month: *
        day: *
        hour: *
        minute: */2
        weekday: *
```

| | | | | |
|--------|------|-----|-------|---------|
| */2 | * | * | * | * |
| minute | hour | day | month | weekday |

Users and Groups

playbook

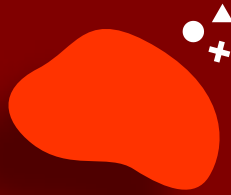
```
---
- hosts: all
  tasks:
    - name: Create a user Maria
      user:
        name: maria
        uid: 1001
        group: developers
        shell: /bin/bash

    - name: Create a group
      group:
        name: developers
```

playbook

```
---
- hosts: all
  tasks:
    - name: Configure ssh keys
      authorized_keys:
        user: maria
        state: present
        key: |
          ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC4WKn4K2G3iWg9HdCGo34gh+.....root@97a1b9c3a
```





Ansible

Variable

Precedence

Variable Precedence

/etc/ansible/hosts

```
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101  dns_server=10.5.5.4
web3 ansible_host=172.20.1.102
```

```
[web_servers]
```

```
web1
Web2
web3
```

```
[web_servers:vars]
```

```
dns_server=10.5.5.3
```

Group Vars

Host Vars

web1

web2

web3

Variable Precedence

```
---  
- name: Configure DNS Server  
  hosts: all  
  vars:  
    dn! dns_server: 10.5.5.5  
  tasks:  
    - nsupdate:  
      server: '{{ dns_server }}
```

Group Vars

Host Vars

Play Vars

dns_server=10.5.5.3

web1

dns_server=10.5.5.4

web2

dns_server=10.5.5.3

web3

Variable Precedence

```
$ ansible-playbook playbook.yml --extra-vars dns_server=10.5.5.6
```

Group Vars

Host Vars

Play Vars

Extra Vars

```
dns_server: 10.5.5.5
```

web1

```
dns_server: 10.5.5.5
```

web2

```
dns_server: 10.5.5.5
```

web3

Variable Precedence

Role Defaults

Group Vars

Host Vars

Host Facts

Play Vars

Role Vars

Include Vars

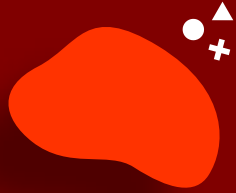
Set Facts

Extra Vars

- role defaults [1]
- inventory file or script group vars [2]
- inventory group_vars/all [3]
- playbook group_vars/all [3]
- inventory group_vars/* [3]
- playbook group_vars/* [3]
- inventory file or script host vars [2]
- inventory host_vars/*
- playbook host_vars/*
- host facts / cached set_facts [4]
- inventory host_vars/* [3]
- playbook host_vars/* [3]
- host facts
- play vars
- play vars_prompt
- play vars_files
- role vars (defined in role/vars/main.yml)
- block vars (only for tasks in block)
- task vars (only for the task)
- include_vars
- set_facts / registered vars
- role (and include_role) params
- include params
- extra vars (always win precedence)

https://docs.ansible.com/ansible/2.5/user_guide/playbooks_variables.htm





Ansible

Variable

Scopes

Variable Scopes

```
/etc/ansible/hosts
```

```
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101 dns_server=10.5.5.4
web3 ansible_host=172.20.1.102
```

```
---
- name: Print dns server
  hosts: all
  tasks:
    - debug:
        msg: '{{ dns_server }}
```

```
PLAY [Check /etc/hosts file]
```

```
*****

TASK [debug] *****
ok: [web1] => {
    "dns_server": "VARIABLE IS NOT DEFINED!"
}
ok: [web2] => {
    "dns_server": "10.5.5.4"
}
ok: [web3] => {
    "dns_server": "VARIABLE IS NOT DEFINED!"
}
```

Variable Scopes - Host



Variable Scopes - Host



Variable Scopes - Play

```
---
- name: Play1
  hosts: web1
  vars:
    ntp_server: 10.1.1.1
  tasks:
    - debug:
        var: ntp_server

- name: Play2
  hosts: web1
  tasks:
    - debug:
        var: ntp_server
```

```
PLAY [Play1]
*****

TASK [debug]
*****
ok: [web1] => {
    "ntp_server": "10.1.1.1"
}

PLAY [Play2] *****

TASK [debug]
*****
ok: [web1] => {
    "ntp_server": "VARIABLE IS NOT DEFINED!"
}
```

Variable Scopes - Global

```
$ ansible-playbook playbook.yml --extra-vars "ntp_server=10.1.1.1"
```

```
---
- name: Play1
  hosts: web1
  vars:
    ntp_server: 10.1.1.1
  tasks:
    - debug:
        var: ntp_server

- name: Play2
  hosts: web1
  tasks:
    - debug:
        var: ntp_server
```

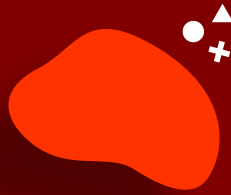
```
PLAY [Play1]
*****

TASK [debug]
*****
ok: [web1] => {
    "ntp_server": "10.1.1.1"
}

PLAY [Play2] *****

TASK [debug]
*****
ok: [web1] => {
    "ntp_server": "10.1.1.1"
}
```





Ansible

Register Variables

playbook

```
---  
- name: Check /etc/hosts file  
  hosts: all  
  tasks:  
    - shell: cat /etc/hosts  
      register: result  
  
- debug:  
  var:
```

```
PLAY [Check /etc/hosts file] *****
```

```
TASK [shell] *****
```

```
changed: [web1]
```

```
changed: [web2]
```

```
PLAY RECAP *****
```

| | | | | | | |
|------|-----------|---------------|----------|-----------|-----------|-----------|
| web1 | : | | | | | |
| ok=1 | changed=1 | unreachable=0 | failed=0 | skipped=0 | rescued=0 | ignored=0 |
| web2 | : | | | | | |
| ok=1 | changed=1 | unreachable=0 | failed=0 | skipped=0 | rescued=0 | ignored=0 |

Register Output

playbook

```
---
- name: Check /etc/hosts file
  hosts: all
  tasks:
    - shell: cat /etc/hosts
      register: result

    - debug:
        var: result
```

```
ok: [web2] => {
  "output": {
    "ansible_facts": {
      "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "cmd": "cat /etc/hosts",
    "failed": false,
    "rc": 0,
    "start": "2019-09-12 05:25:34.158877",
    "end": "2019-09-12 05:25:34.161974",
    "delta": "0:00:00.003097",
    "stderr": "",
    "stderr_lines": [],
    "stdout": "127.0.0.1\tlocalhost\n::1\tlocalhost ip6-loopback\nfe00::0\tip6-localnet\nff00::0\tip6-mcastprefix\nff02::1\tip6-allnodes\nff02::2\tip6-allrouters\n172.20.1.101\tweb2"
  },
  "rc": 0,
  "start": "2019-09-12 05:25:34.158877",
  "end": "2019-09-12 05:25:34.161974",
  "delta": "0:00:00.003097",
  "stderr": "",
  "stderr_lines": [],
  "stdout_lines": [
    "127.0.0.1\tlocalhost",
    "::1\tlocalhost ip6-loopback",
    "fe00::0\tip6-localnet",
    "ff00::0\tip6-mcastprefix",
    "ff02::1\tip6-allnodes",
    "ff02::2\tip6-allrouters",
    "172.20.1.101\tweb2"
  ]
}
```

Register Output Scope

playbook

```
---  
- name: Check /etc/hosts file  
  hosts: all  
  tasks:  
    - shell: cat /etc/hosts  
      register: result  
  
    - debug:  
      var: result.rc  
  
- name: Play2  
  hosts: all  
  tasks:  
    - debug:  
      var: result.rc
```

result

Web1

result

Web2

playbook

```
---  
- name: Check /etc/hosts file  
  hosts: all  
  tasks:  
    - shell: cat /etc/hosts
```

```
$ ansible-playbook -i inventory playbook.yml -v
```

```
PLAY [localhost] *****
```

```
TASK [Gathering Facts] *****
```

```
ok: [localhost]
```

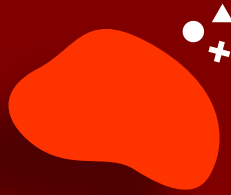
```
TASK [shell] *****
```

```
changed: [localhost] => {"changed": true, "cmd": "cat /etc/hosts", "delta": "0:00:00.282432", "end": "2019-09-24 07:37:26.440478", "rc": 0,  
"start": "2019-09-24 07:37:26.158046", "stderr": "", "stderr_lines": [], "stdout": "127.0.0.1\tlocalhost\n::1\tlocalhost ip6-localhost ip6-  
loopback\nfe00::0\tip6-localnet\nff00::0\tip6-mcastprefix\nff02::1\tip6-allnodes\nff02::2\tip6-allrouters\n172.20.1.2\tf6d0e5fbbc0d",  
"stdout_lines": ["127.0.0.1\tlocalhost", "::1\tlocalhost ip6-localhost ip6-loopback", "fe00::0\tip6-localnet", "ff00::0\tip6-mcastprefix",  
"ff02::1\tip6-allnodes", "ff02::2\tip6-allrouters", "172.20.1.2\tf6d0e5fbbc0d"]}
```

```
PLAY RECAP *****
```

```
localhost : ok=2 changed=1 unreachable=0 failed=0
```





Ansible

Magic

Variables

Variable Scopes

```
/etc/ansible/hosts
```

```
web1 ansible_host=172.20.1.100  
web2 ansible_host=172.20.1.101 dns_server=10.5.5.4  
web3 ansible_host=172.20.1.102
```

```
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101 dns_server=10.5.5.4
web3 ansible_host=172.20.1.102
```



```
---
- name: Print dns server
  hosts: all
  tasks:
    - debug:
        msg: '{{ dns_server }}
```

```
PLAY [Check /etc/hosts file]
*****

TASK [debug] *****
ok: [web1] => {
    "dns_server": "VARIABLE IS NOT DEFINED!"
}
ok: [web2] => {
    "dns_server": "10.5.5.4"
}
ok: [web3] => {
    "dns_server": "VARIABLE IS NOT DEFINED!"
}
```

Variable Interpolation

```
inventory_hostname=web1
ansible_host=172.20.1.100
```

```
inventory_hostname=web2
ansible_host=172.20.1.101
dns_server=10.5.5.4
```

```
inventory_hostname=web3
ansible_host=172.20.1.102
```

web1

web2

web3

Create Subprocess

DEK CLOUD


```
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101 dns_server=10.5.5.4
web3 ansible_host=172.20.1.102
```



```
---
- name: Print dns server
  hosts: all
  tasks:
    - debug:
        msg: '{{ hostvars['web2'].dns_server }}'
```

```
PLAY [Check /etc/hosts file]
*****

TASK [debug] *****
ok: [web1] => {
    "dns_server": "10.5.5.4"
}
ok: [web2] => {
    "dns_server": "10.5.5.4"
}
ok: [web3] => {
    "dns_server": "10.5.5.4"
}
```

Variable Interpolation

```
inventory_hostname=web1
ansible_host=172.20.1.100
```

```
inventory_hostname=web2
ansible_host=172.20.1.101
dns_server=10.5.5.4
```

```
inventory_hostname=web3
ansible_host=172.20.1.102
```

Create Subprocess





```
---  
- name: Print dns server  
  hosts: all  
  tasks:  
    - debug:  
      msg: '{{ hostvars['web2'].dns_server }}'
```

```
msg: '{{ hostvars['web2'].ansible_host }}'
```

```
msg: '{{ hostvars['web2'].ansible_facts.architecture }}'
```

```
msg: '{{ hostvars['web2'].ansible_facts.devices }}'
```

```
msg: '{{ hostvars['web2'].ansible_facts.mounts }}'
```

```
msg: '{{ hostvars['web2'].ansible_facts.processor }}'
```

```
---
- name: Print dns server
  hosts: all
  tasks:
  - debug:
      msg: '{{ hostvars['web2'].dns_server }}'
```

```
msg: '{{ hostvars['web2'].ansible_host }}'
```

```
msg: '{{ hostvars['web2'].ansible_facts.architecture }}'
```

```
msg: '{{ hostvars['web2'].ansible_facts.devices }}'
```

```
msg: '{{ hostvars['web2'].ansible_facts.mounts }}'
```

```
msg: '{{ hostvars['web2'].ansible_facts.processor }}'
```

=

```
msg: '{{ hostvars['web2']['ansible_facts']['processor'] }}'
```

Magic Variable - hostvars

```
msg: '{{ hostvars['web2'].ansible_host }}'
```

```
msg: '{{ hostvars['web2'].ansible_facts.architecture }}'
```

```
msg: '{{ hostvars['web2'].ansible_facts.devices }}'
```

```
msg: '{{ hostvars['web2'].ansible_facts.mounts }}'
```

```
msg: '{{ hostvars['web2'].ansible_facts.processor }}'
```

=

```
msg: '{{ hostvars['web2']['ansible_facts']['processor'] }}'
```

Magic Variable - groups

```
/etc/ansible/hosts
```

```
web1 ansible_host=172.20.1.100  
web2 ansible_host=172.20.1.101  
web3 ansible_host=172.20.1.102
```

```
[web_servers]
```

```
web1  
Web2  
web3
```

```
[americas]
```

```
web1  
web2
```

```
[asia]
```

```
web3
```

```
msg: '{{ groups['americas'] }}'
```

```
web1  
web2
```

Magic Variable – group_names

```
/etc/ansible/hosts
```

```
web1 ansible_host=172.20.1.100  
web2 ansible_host=172.20.1.101  
web3 ansible_host=172.20.1.102
```

```
[web_servers]
```

```
web1  
Web2  
web3
```

```
[americas]
```

```
web1  
web2
```

```
[asia]
```

```
web3
```

```
msg: '{{ group_names }}' # web1
```

```
web_servers  
americas
```

Magic Variable – inventory_hostname

```
/etc/ansible/hosts
```

```
web1 ansible_host=172.20.1.100  
web2 ansible_host=172.20.1.101  
web3 ansible_host=172.20.1.102
```

```
[web_servers]
```

```
web1  
Web2  
web3
```

```
[americas]
```

```
web1  
web2
```

```
[asia]
```

```
web3
```

```
msg: '{{ inventory_hostname }}' # web1  
web1
```

User Guide

Ansible Quickstart

Getting Started

Working with Command Line Tools

Introduction To Ad-Hoc Commands

Working with Inventory

Working With Dynamic Inventory

Working With Playbooks

Intro to Playbooks

Creating Reusable Playbooks

Using Variables

Creating valid variable names

Defining variables in inventory

Defining variables in a playbook

Defining variables in included
files and roles

Using variables with Jinja2

Transforming variables with
Jinja2 filters

Hey wait, a YAML gotcha

Variables discovered from
systems: Facts

Registering variables

Accessing complex variable data

Accessing information about
other hosts with magic variables

Accessing information about other hosts with magic variables

Whether or not you define any variables, you can access information about your hosts with the [Special Variables](#) Ansible provides, including “magic

The most commonly used magic variables are `hostvars`, `groups`, `group_names`, and `inventory_hostname`.

`hostvars` lets you access variables for another host, including facts that have been gathered about that host. You can access host variables at any point in a play to see the facts.

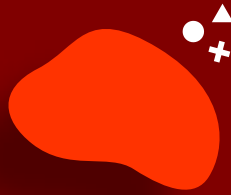
If your database server wants to use the value of a ‘fact’ from another node, or an inventory variable assigned to another node, it’s easy to do so with

```
{{ hostvars['test.example.com']['ansible_facts']['distribution'] }}
```

`groups` is a list of all the groups (and hosts) in the inventory. This can be used to enumerate all hosts within a group. For example:

```
{% for host in groups['app_servers'] %}  
    # something that applies to all app servers.  
{% endfor %}
```



Ansible

Conditionals

```
---  
- name: Install NGINX  
  hosts: debian_hosts  
  tasks:  
    - name: Install NGINX on Debian  
      apt:  
        name: nginx  
        state: present
```

```
---  
- name: Install NGINX  
  hosts: redhat_hosts  
  tasks:  
    - name: Install NGINX on Redhat  
      yum:  
        name: nginx  
        state: present
```

Conditional - when

```
---
- name: Install NGINX
  hosts: all
  tasks:
    - name: Install NGINX on Debian
      apt:
        name: nginx
        state: present
        when: ansible_os_family == "Debian"

    - name: Install NGINX on Redhat
      yum:
        name: nginx
        state: present
        when: ansible_os_family == "RedHat"
```

Operator - or

```
---
- name: Install NGINX
  hosts: all
  tasks:
    - name: Install NGINX on Debian
      apt:
        name: nginx
        state: present
      when: ansible_os_family == "Debian"

    - name: Install NGINX on Redhat
      yum:
        name: nginx
        state: present
      when: ansible_os_family == "RedHat" or
            ansible_os_family == "SUSE"
```

Operator - and

```
---
- name: Install NGINX
  hosts: all
  tasks:
    - name: Install NGINX on Debian
      apt:
        name: nginx
        state: present
        when: ansible_os_family == "Debian" and
              ansible_distribution_version == "16.04"

    - name: Install NGINX on Redhat
      yum:
        name: nginx
        state: present
        when: ansible_os_family == "RedHat" or
              ansible_os_family == "SUSE"
```

Conditionals in Loops

```
---
- name: Install Softwares
  hosts: all
  vars:
    packages:
      - name: nginx
        required: True
      - name: mysql
        required : True
      - name: apache
        required : False
  tasks:
    - name: Install "{{ item.name }}" on Debian
      apt:
        name: "{{ item.name }}"
        state: present

    loop: "{{ packages }}"
```

Conditionals in Loops

```
---
- name: Install Softwares
  hosts: all
  vars:
    packages:
      - name: nginx
        required: True
      - name: mysql
        required : True
      - name: apache
        required : False
  tasks:
    - name: Install "{{ item.name }}" on Debian
      apt:
        name: "{{ item.name }}"
        state: present

    loop: "{{ packages }}"
```

```
- name: Install "{{ item.name }}" on Debian
  vars:
    item:
      name: nginx
      required: True
  apt:
    name: "{{ item.name }}"
    state: present
  when: item.required == True
```

```
- name: Install "{{ item.name }}" on Debian
  vars:
    item:
      name: mysql
      required: True
  apt:
    name: "{{ item.name }}"
    state: present
  when: item.required == True
```

```
- name: Install "{{ item.name }}" on Debian
  vars:
    item:
      name: apache
      required: False
  apt:
    name: "{{ item.name }}"
    state: present
  when: item.required == True
```


Conditionals in Loops

```
---
- name: Install Softwares
  hosts: all
  vars:
    packages:
      - name: nginx
        required: True
      - name: mysql
        required : True
      - name: apache
        required : False
  tasks:
    - name: Install "{{ item.name }}" on Debian
      apt:
        name: "{{ item.name }}"
        state: present

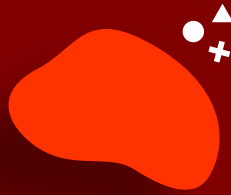
      when: item.required == True
      loop: "{{ packages }}"
```

Conditionals & Register

```
- name: Check status of a service and email if its down
hosts: localhost
tasks:
  - command: service httpd status
    register: result

  - mail:
      to: admin@company.com
      subject: Service Alert
      body: Httpd Service is down
      when: result.stdout.find('down') != -1
```





Ansible Blocks

```
-
hosts: server1
tasks:
  - name: Install MySQL
    yum: name=mysql-server state=present
    become_user: db-user
    when: ansible_facts['distribution'] == 'CentOS'

  - name: Start MySQL Service
    service: name=mysql-server state=started
    become_user: db-user
    when: ansible_facts['distribution'] == 'CentOS'

  - name: Install Nginx
    yum: name=nginx state=present
    become_user: web-user
    when: ansible_facts['distribution'] == 'CentOS'

  - name: Start Nginx Service
    service: name=nginx state=started
    become_user: web-user
    when: ansible_facts['distribution'] == 'CentOS'
```

```
-
hosts: server1
tasks:

- block:
  - name: Install MySQL
    yum: name=mysql-server state=present
    become_user: db-user
    when: ansible_facts['distribution'] == 'CentOS'

  - name: Start MySQL Service
    service: name=mysql-server state=started
    become_user: db-user
    when: ansible_facts['distribution'] == 'CentOS'

- block:
  - name: Install Nginx
    yum: name=nginx state=present
    become_user: web-user
    when: ansible_facts['distribution'] == 'CentOS'

  - name: Start Nginx Service
    service: name=nginx state=started
    become_user: web-user
    when: ansible_facts['distribution'] == 'CentOS'
```

```
-
hosts: server1
tasks:
  - block:
      - name: Install MySQL
        yum: name=mysql-server state=present
      - name: Start MySQL Service
        service: name=mysql-server state=started

    become_user: db-user
    when: ansible_facts['distribution'] == 'CentOS'

  - block:
      - name: Install Nginx
        yum: name=nginx state=present
        become_user: web-user
        when: ansible_facts['distribution'] == 'CentOS'

      - name: Start Nginx Service
        service: name=nginx state=started
        become_user: web-user
        when: ansible_facts['distribution'] == 'CentOS'
```

```
-
hosts: server1
tasks:
  - block:
      - name: Install MySQL
        yum: name=mysql-server state=present
      - name: Start MySQL Service
        service: name=mysql-server state=started

    become_user: db-user
    when: ansible_facts['distribution'] == 'CentOS'

  - block:
      - name: Install Nginx
        yum: name=nginx state=present
      - name: Start Nginx Service
        service: name=nginx state=started

    become_user: web-user
    when: ansible_facts['distribution'] == 'CentOS'
```


Error Handling

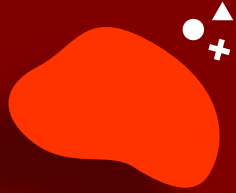
```
-
hosts: server1
tasks:
  - block:
      - name: Install MySQL
        yum: name=mysql-server state=present
      - name: Start MySQL Service
        service: name=mysql-server state=started

  become_user: db-user
  when: ansible_facts['distribution'] == 'CentOS'

  rescue:
    - mail:
        to: admin@company.com
        subject: Installation Failed
        body: DB Install Failed at {{ ansible_failed_task.name }}

  always:
    - mail:
        to: admin@company.com
        subject: Installation Status
        body: DB Install Status - {{ ansible_failed_result }}
```





Ansible Error Handling

Task failure

```
- name: Deploy web application
hosts: server1
tasks:
  - name: Install dependencies
    << code hidden >>

  - name: Install MySQL Database
    << code hidden >>

  - name: Start MySQL Service
    << code hidden >>

  - name: Install Python Flask Dependencies
    << code hidden >>

  - name: Run web-server
    << code hidden >>
```

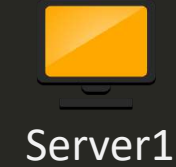
Dependencies

Install MySQL

Start DB

Install Flask

Run Server



Server1

Exit

Task failure

```
- name: Deploy web application
hosts: server1,server2,server3
tasks:
  - name: Install dependencies
    << code hidden >>

  - name: Install MySQL Database
    << code hidden >>

  - name: Start MySQL Service
    << code hidden >>

  - name: Install Python Flask Dependencies
    << code hidden >>

  - name: Run web-server
    << code hidden >>
```

Dependencies

Install MySQL

Start DB

Install Flask

Run Server

Server1

Server2

Server3

Skip

Task failure

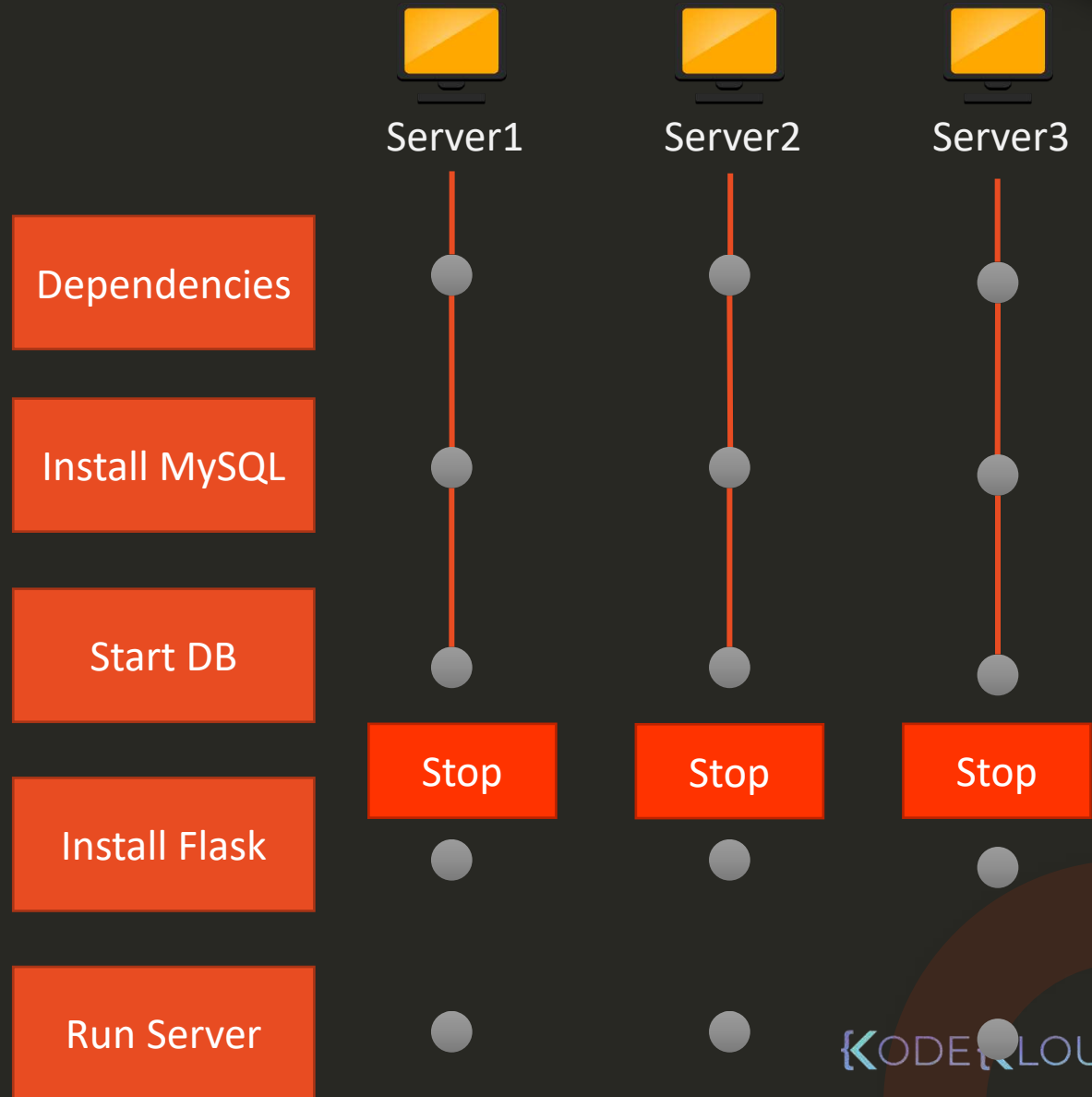
```
- name: Deploy web application
hosts: server1,server2,server3
any_errors_fatal: true
tasks:
  - name: Install dependencies
    << code hidden >>

  - name: Install MySQL Database
    << code hidden >>

  - name: Start MySQL Service
    << code hidden >>

  - name: Install Python Flask Dependencies
    << code hidden >>

  - name: Run web-server
    << code hidden >>
```



Task failure

```
- name: Deploy web application
hosts: server1,server2,server3
max_fail_percentage: 30
tasks:
  - name: Install dependencies
    << code hidden >>

  - name: Install MySQL Database
    << code hidden >>

  - name: Start MySQL Service
    << code hidden >>

  - name: Install Python Flask Dependencies
    << code hidden >>

  - name: Run web-server
    << code hidden >>
```

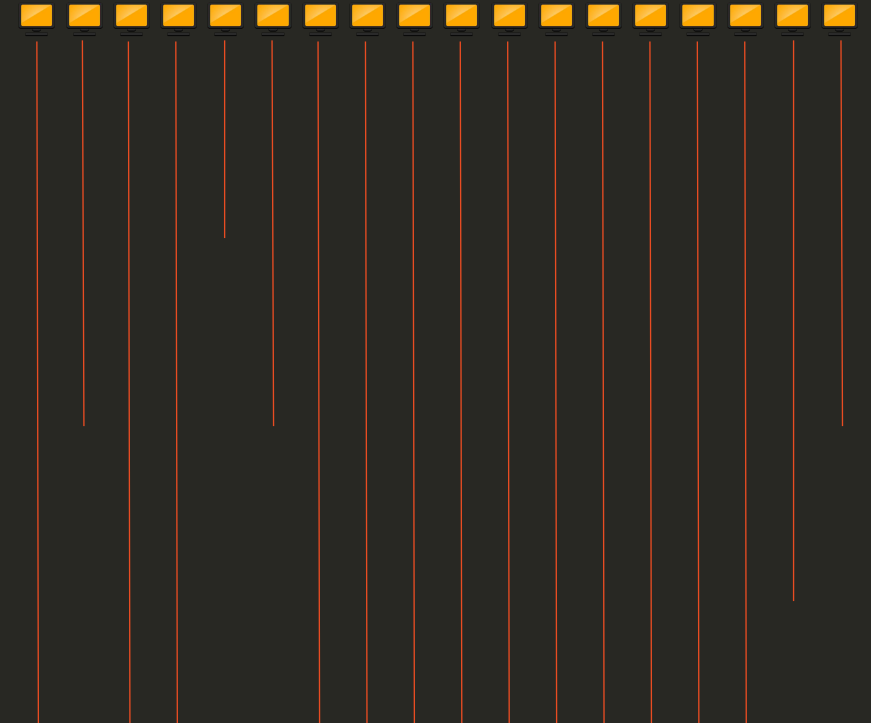
Dependencies

Install MySQL

Start DB

Install Flask

Run Server



Ignore errors

```
- name: Deploy web application
hosts: server1,server2,server3
any_errors_fatal: true
tasks:
  - name: Install dependencies
    << code hidden >>
  - name: Install MySQL Database
    << code hidden >>
  - name: Start MySQL Service
    << code hidden >>
  - name: Install Python Flask Dependencies
    << code hidden >>
  - name: Run web-server
    << code hidden >>
  - mail:
    to: admin@company.com
    subject: Server Configured
    body: Web server has been configured
ignore_errors: yes
```


failed_when

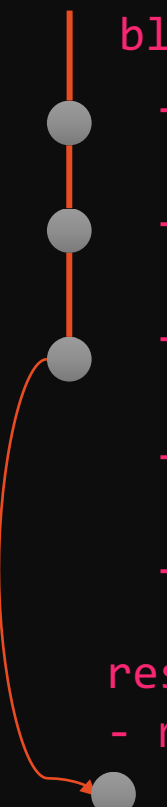
```
- name: Deploy web application
hosts: server1,server2,server3
any_errors_fatal: true
tasks:
  - name: Install dependencies
    << code hidden >>
  - name: Install MySQL Database
    << code hidden >>
  - name: Start MySQL Service
    << code hidden >>
  - name: Install Python Flask Dependencies
    << code hidden >>
  - name: Run web-server
    << code hidden >>

- command: cat /var/log/server.log
  register: command_output
  failed_when: 'ERROR' in command_output.stdout

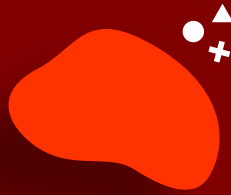
- mail:
  to: admin@company.com
  subject: Server Configured
  body: Web server has been configured
```

Blocks

```
- name: Deploy web application
  hosts: server1,server2,server3
  any_errors_fatal: true
  tasks:
    - name: Install web Application
      block:
        - name: Install dependencies
          << code hidden >>
        - name: Install MySQL Database
          << code hidden >>
        - name: Start MySQL Service
          << code hidden >>
        - name: Install Python Flask Dependencies
          << code hidden >>
        - name: Run web-server
          << code hidden >>
      rescue:
        - mail:
            to: admin@company.com
            subject: Playbook Failed
            body: Web server configuration failed
```







Jinja2

Templating?

Template

Hi ,

I am glad to invite you along with your family members - , to attend the party arranged by us on the completion of 10 successful years of our company. We would be happy to mark your presence along with family at the party and would love to celebrate the success together.

Sincerely,

Andrews,
CEO

Variables

Sam

Mary and Adam

Anil

Achu and George

Michelle

Sarah

Shabab

Aliah and Medina



Templating Engine

Hi Sam,

I am glad to invite you along with your family members - Mary and Adam, to attend the party arranged by us on the completion of 10 successful years of our company. We would be happy to mark your presence along with family at the party and would love to celebrate the success together.

Sincerely,

Andrews,
CEO

Hi Anil ,

I am glad to invite you along with your family members - Achu and George, to attend the party arranged by us on the completion of 10 successful years of our company. We would be happy to mark your presence along with family at the party and would love to celebrate the success together.

Sincerely,

Andrews,
CEO

Hi Michelle ,

I am glad to invite you along with your family members - Sarah, to attend the party arranged by us on the completion of 10 successful years of our company. We would be happy to mark your presence along with family at the party and would love to celebrate the success together.

Hi Shabab ,

I am glad to invite you along with your family members - Aliah and Medina, to attend the party arranged by us on the completion of 10 successful years of our company. We would be happy to mark your presence along with family at the party and would love to celebrate the success together.

HTML

Template

```
<!DOCTYPE html>
<html>
  <head>
    <title>{{ title }}</title>
  </head>
  <body>
    {{ msg }}
  </body>
</html>
```

Variables

title: Our Site
msg: Welcome!

Outcome

```
<!DOCTYPE html>
<html>
  <head>
    <title>Our Site</title>
  </head>
  <body>
    Welcome!
  </body>
</html>
```

ANSIBLE

Template

```
- hosts: web1
  tasks:
    -
      file:
        path: {{ file }}
        state: touch
```

Variables

file: /tmp/1.txt

Outcome

```
- hosts: web1
  tasks:
    -
      file:
        path: /tmp/1.txt
        state: touch
```

Template

```
[mysqld]
innodb-buffer-pool-size={{ pool_size }}
datadir={{ datadir }}
user={{ mysql_user }}
symbolic-links={{ link_id }}
port={{ mysql_port }}
```

Variables

pool_size: 5242880
datadir: /var/lib/mysql
mysql_user: mysql
link_id: 0
mysql_port: 3306

Outcome

```
[mysqld]
innodb-buffer-pool-size=5242880
datadir=/var/lib/mysql
user=mysql
symbolic-links=0
port=3306
```


jinja2

Project Links

[Donate to Pallets](#)

[Jinja Website](#)

[PyPI releases](#)

[Source Code](#)

[Issue Tracker](#)

Quick search

Go



YOUR AD HERE

Reach over 7 million devs
each month when you
advertise with Read the Docs.

Sponsored · Ads served ethically



Jinja is a modern and designer-friendly templating language for Python, modelled after Django's templates. It is fast, widely used and secure with the optional sandboxed template execution environment:

```
<title>{% block title %}{% endblock %}</title>
<ul>
{% for user in users %}
  <li><a href="{{ user.url }}">{{ user.username }}</a></li>
{% endfor %}
</ul>
```

Features:

- sandboxed execution
- powerful automatic HTML escaping system for XSS prevention
- template inheritance
- compiles down to the optimal python code just in time
- optional ahead-of-time template compilation
- easy to debug. Line numbers of exceptions directly point to the correct line in the template.
- configurable syntax

Contents:

- [Introduction](#)
 - [Prerequisites](#)
 - [Installation](#)
 - [Basic API Usage](#)
 - [Experimental Python 3 Support](#)
- [API](#)
 - [Basics](#)
 - [Unicode](#)
 - [High Level API](#)
 - [Autoescaping](#)
 - [Notes on Identifiers](#)

String manipulation - FILTERS

The name is `{{ my_name }}` => The name is Bond

The name is `{{ my_name | upper }}` => The name is BOND

The name is `{{ my_name | lower }}` => The name is bond

The name is `{{ my_name | title }}` => The name is Bond

The name is `{{ my_name | replace("Bond", "Bourne") }}` => The name is Bourne

The name is `{{ first_name | default("James") }} {{ my_name }}` => The name is James Bond

- Substitute
- Upper
- Lower
- Title
- replace
- default

Filters - List and set

```
{{ [ 1, 2, 3 ] | min }}           => 1
{{ [ 1, 2, 3 ] | max }}           => 3
{{ [ 1, 2, 3, 2 ] | unique }}     => 1, 2, 3
{{ [ 1, 2, 3, 4 ] | union( [ 4, 5 ] ) }} => 1, 2, 3, 4, 5
{{ [ 1, 2, 3, 4 ] | intersect( [ 4, 5 ] ) }} => 4
{{ 100 | random }}                => Random number
{{ [ "The", "name", "is", "Bond" ] | join(" ") }} => The name is Bond
```



- min
- max
- unique
- union
- intersect
- random
- join

Loops

```
{% for number in [0,1,2,3,4] %}  
{{ number }}  
{% endfor %}
```

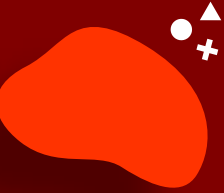
```
0  
1  
2  
3  
4
```

Conditions

```
{% for number in [0,1,2,3,4] %}  
    {% if number == 2 %}  
        {{ number }}  
    {% endif %}  
  
{% endfor %}
```

2





Ansible

Jinja2

in Ansible

Ansible Filters

abs()	float()	lower()	round()	tojson()
attr()	forceescape()	map()	safe()	trim()
batch()	format()	max()	select()	truncate()
capitalize()	groupby()	min()	selectattr()	unique()
center()	indent()	pprint()	slice()	upper()
default()	int()	random()	sort()	urlencode()
dictsort()	join()	reject()	string()	urlize()
escape()	last()	rejectattr()	striptags()	wordcount()
filesizeformat()	length()	replace()	sum()	wordwrap()
first()	list()	reverse()	title()	xmlattr()

[b64decode\(\)](#)

[b64encode\(\)](#)

[to_uuid\(\)](#)

[to_json\(\)](#)

[to_nice_json\(\)](#)

[from_json\(\)](#)

[to_yaml\(\)](#)

[to_nice_yaml\(\)](#)

[from_yaml\(\)](#)

[from_yaml_all\(\)](#)

[basename\(\)](#)

[dirname\(\)](#)

[expanduser\(\)](#)

[expandvars\(\)](#)

[realpath\(\)](#)

[relpath\(\)](#)

[splitext\(\)](#)

[win_basename\(\)](#)

[win_dirname\(\)](#)

[win_splitdrive\(\)](#)

[combine\(\)](#)

[extract\(\)](#)

[flatten\(\)](#)

[dict2items\(\)](#)

[items2dict\(\)](#)

[subelements\(\)](#)

[random_mac\(\)](#)

[rejectattr\(\)](#)

[comment\(\)](#)

[mandatory\(\)](#)

Filters - file

<code>{{ "/etc/hosts" basename }}</code>	<code>=> hosts</code>
<code>{{ "c:\windows\hosts" win_basename }}</code>	<code>=> hosts</code>
<code>{{ "c:\windows\hosts" win_splitdrive }}</code>	<code>=> ["c:", "\windows\hosts"]</code>
<code>{{ "c:\windows\hosts" win_splitdrive first }}</code>	<code>=> "c:"</code>
<code>{{ "c:\windows\hosts" win_splitdrive last }}</code>	<code>=> "\windows\hosts"</code>

Jinja2 in Playbooks

/etc/ansible/hosts

```
web1 ansible_host=172.20.1.100 dns_server=10.5.5.4
web2 ansible_host=172.20.1.101 dns_server=10.5.5.4
web3 ansible_host=172.20.1.102 dns_server=10.5.5.4
```



```
---
- name: Update dns server
  hosts: all
  tasks:
    - nsupdate:
        server: '{{ dns_server }}
```



```
---
- name: Update dns server
  hosts: all
  tasks:
    - nsupdate:
        server: 10.5.5.4
```



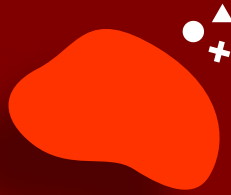


The Curriculum

Red Hat Certified Ansible Specialist

- Core Components
- Install and Configure Ansible Control Node
- Configure Ansible Managed Nodes
- Create simple shell scripts that run ad hoc Ansible commands
- Dynamic inventories
- Ansible Plays and Playbooks
- Ansible Modules
- Customized Configuration Files with Jinja2
- Variables and Facts
- Roles
- Ansible Vault
- Documentation





Ansible Templates

Templates

/etc/ansible/hosts

```
[web_servers]
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101
web3 ansible_host=172.20.1.102
```

playbook.yml

```
-
  hosts: web_servers
  tasks:
    - name: Copy index.html to remote servers
      copy:
        src: index.html
        dest: /var/www/nginx-default/index.html
```

index.html

```
<!DOCTYPE html>
<html>
<body>

This is a Web Server

</body>
</html>
```

Templates

```
/etc/ansible/hosts
```

```
[web_servers]  
web1 ansible_host=1  
web2 ansible_host=1
```

This is a Web Server

This is web1 server

This is web2 server

This is web3 server

```
src: index.  
dest: /var/
```

web1

```
index.html
```

```
<!DOCTYPE html>  
<html>  
<body>
```

This is a Web Server

```
</body>
```

web2

```
index.html
```

```
<!DOCTYPE html>  
<html>  
<body>
```

This is a Web Server

```
</body>
```

web3

```
index.html
```

```
<!DOCTYPE html>  
<html>  
<body>
```

This is a Web Server

```
</body>
```

Templates

/etc/ansible/hosts

```
[web_servers]
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101
web3 ansible_host=172.20.1.102
```

playbook.yml

```
-
  hosts: web_servers
  tasks:
    - name: Copy index.html to remote servers
      copy:
        src: index.html
        dest: /var/www/nginx-default/index.html
```

index.html

```
<!DOCTYPE html>
<html>
<body>
```

This is a Web Server

```
</body>
</html>
```

web1

index.html

```
<!DOCTYPE html>
<html>
<body>
```

This is web1 Server

```
</body>
</html>
```

web2

index.html

```
<!DOCTYPE html>
<html>
<body>
```

This is web2 Server

```
</body>
</html>
```

web3

index.html

```
<!DOCTYPE html>
<html>
<body>
```

This is web3 Server

```
</body>
</html>
```


Templates

/etc/ansible/hosts

```
[web_servers]
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101
web3 ansible_host=172.20.1.102
```

playbook.yml

```
-
  hosts: web_servers
  tasks:
    - name: Copy index.html to remote servers
      copy:
        src: index.html
        dest: /var/www/nginx-default/index.html
```

index.html

```
<!DOCTYPE html>
<html>
<body>

This is a Web Server

</body>
</html>
```

web3

index.html

```
<!DOCTYPE html>
<html>
<body>

This is {{ name }} Server

</body>
</html>
```

Templates

/etc/ansible/hosts

```
[web_servers]
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101
web3 ansible_host=172.20.1.102
```

playbook.yml

```
-
  hosts: web_servers
  tasks:
    - name: Copy index.html to remote servers
      template:
        src: index.html.j2
        dest: /var/www/nginx-default/index.html
```

index.html.j2

```
<!DOCTYPE html>
<html>
<body>

This is {{ inventory_hostname }} Server

</body>
</html>
```

```
[web_servers]
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101
web3 ansible_host=172.20.1.102
```



Variable Interpolation

Gather Facts

Execute Playbook

Create file from
Template

Copy to target host

Create Subprocess

```
inventory_hostname=web1
ansible_host=172.20.1.100

ansible_facts=<Host Facts>

playbook.yml

-
  hosts: web_servers
  tasks:
    - name: Copy index.html to remote servers
      copy:
        src: index.html
        dest: /var/www/nginx-default/index.html
```

```
inventory_hostname=web2
ansible_host=172.20.1.101

ansible_facts=<Host Facts>

playbook.yml

-
  hosts: web_servers
  tasks:
    - name: Copy index.html to remote servers
      copy:
        src: index.html
        dest: /var/www/nginx-default/index.html
```

```
inventory_hostname=web3
ansible_host=172.20.1.102

ansible_facts=<Host Facts>

playbook.yml

-
  hosts: web_servers
  tasks:
    - name: Copy index.html to remote servers
      copy:
        src: index.html
        dest: /var/www/nginx-default/index.html
```

web1

index.html

<!DOCTYPE html>
<html>
<body>

This is web1 Server

</body>
</html>

web2

index.html

<!DOCTYPE html>
<html>
<body>

This is web2 Server

</body>
</html>

web3

index.html

<!DOCTYPE html>
<html>
<body>

This is web3 Server

</body>
</html>

Template Examples

nginx.conf.j2

```
server {
    location / {
        fastcgi_pass  {{host}}:{{port}};
        fastcgi_param QUERY_STRING  $query_string;
    }

    location ~ \ gif|jpg|png $ {
        root {{ image_path }};
    }
}
```

nginx.conf

```
server {
    location / {
        fastcgi_pass  localhost:9000
        fastcgi_param QUERY_STRING  $query_string;
    }

    location ~ \ gif|jpg|png $ {
        root /data/images;
    }
}
```

Template Examples

redis.conf.j2

```
bind {{ ip_address }}

protected-mode yes

port {{ redis_port | default('6379') }}

tcp-backlog 511

# Unix socket.
timeout 0

# TCP keepalive.
tcp-keepalive {{tcp_keepalive | default('300') }}

daemonize no

supervised no
```

redis.conf

```
bind 192.168.1.100

protected-mode yes

port 6379

tcp-backlog 511

# Unix socket.
timeout 0

# TCP keepalive.
tcp-keepalive 300

daemonize no

supervised no
```

Template Examples

/etc/resolv.conf.j2

```
{% for name_server in name_servers %}  
nameserver    name_server  
{% endfor %}
```

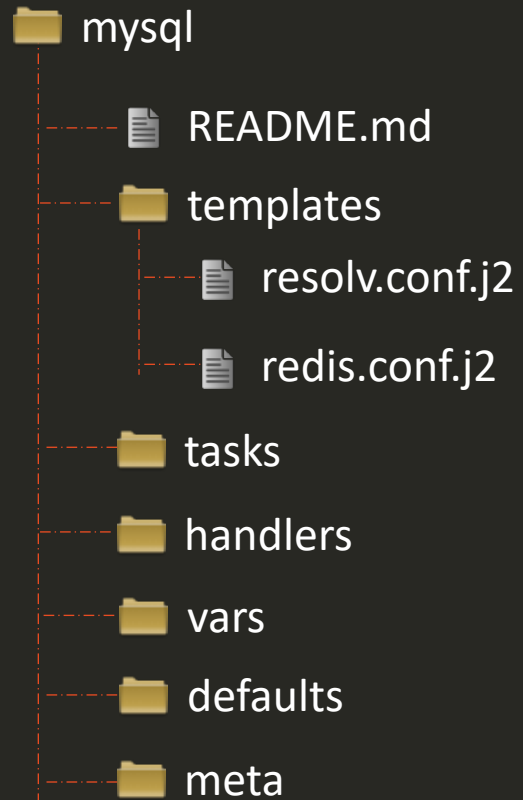
/etc/resolv.conf

```
nameserver 10.1.1.2  
nameserver 10.1.1.3  
nameserver 8.8.8.8
```

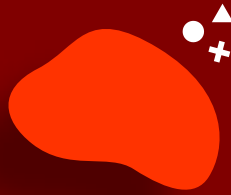
variable

```
name_servers:  
- 10.1.1.2  
- 10.1.1.3  
- 8.8.8.8
```

Templates in Roles







Ansible

Includes

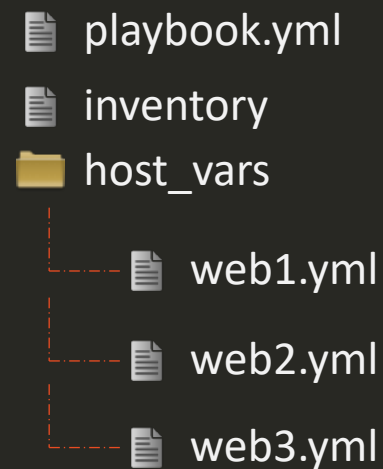
inventory

```
[web_servers]
web1 ansible_host=172.20.1.100 dns_server=10.1.1.5
web2 ansible_host=172.20.1.101 dns_server=10.1.1.5
web3 ansible_host=172.20.1.102 dns_server=10.1.1.5
```

web1.yml

web2.yml

web3.yml



inventory

[web_servers]

web1

web2

web3

web1.yml

ansible_host: 172.20.1.100

dns_server: 10.1.1.5

web2.yml

ansible_host: 172.20.1.101

dns_server: 10.1.1.5

web3.yml

ansible_host: 172.20.1.103

dns_server: 10.1.1.5

web_servers.yml

playbook.yml

inventory

host_vars

web1.yml

web2.yml

web3.yml

group_vars

web_servers.yml

inventory

[web_servers]

web1

web2

web3

web1.yml

ansible_host: 172.20.1.100

web2.yml

ansible_host: 172.20.1.101

web3.yml

ansible_host: 172.20.1.103

web_servers.yml

dns_server: 10.1.1.5

playbook.yml

inventory

host_vars

web1.yml

web2.yml

web3.yml

group_vars

web_servers.yml

inventory

[web_servers]

web1

web2

web3

web1.yml

ansible_host: 172.20.1.100

web2.yml

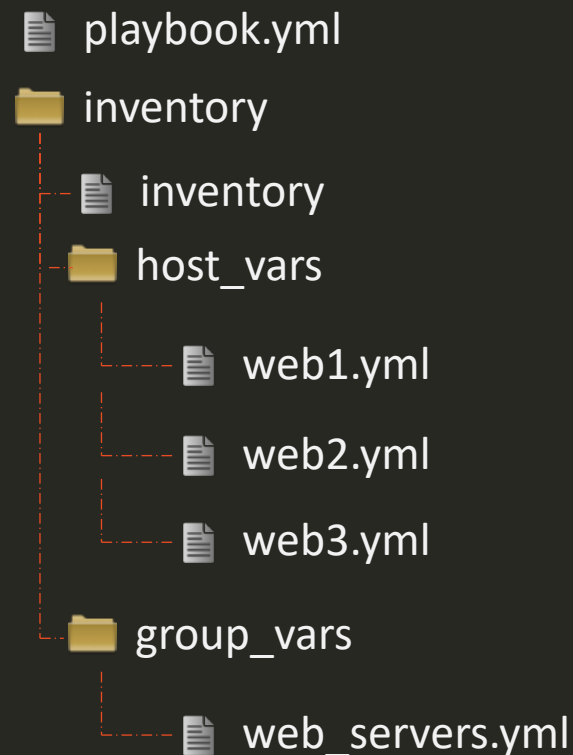
ansible_host: 172.20.1.101

web3.yml

ansible_host: 172.20.1.103

web_servers.yml

dns_server: 10.1.1.5



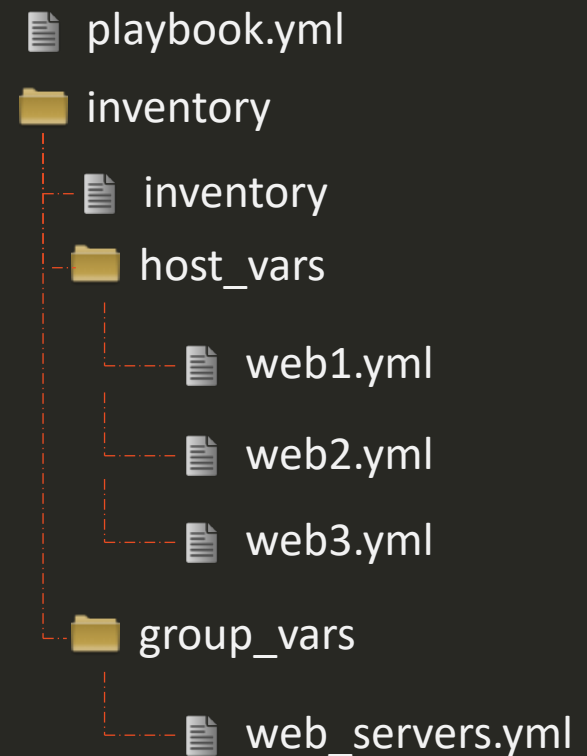
Include Vars

playbook.yml

```
- name: Deploy Web & DB Server
  hosts: web-db-server
  tasks:
    - mail:
        to: admin@company.com
        subject: Service Alert
        body: Httpd Service is down
```

/opt/apps/common-data/email/info.yml

```
admin_email: admin@company.com
```



/opt/apps/common-data/email

info.yml

Include Vars

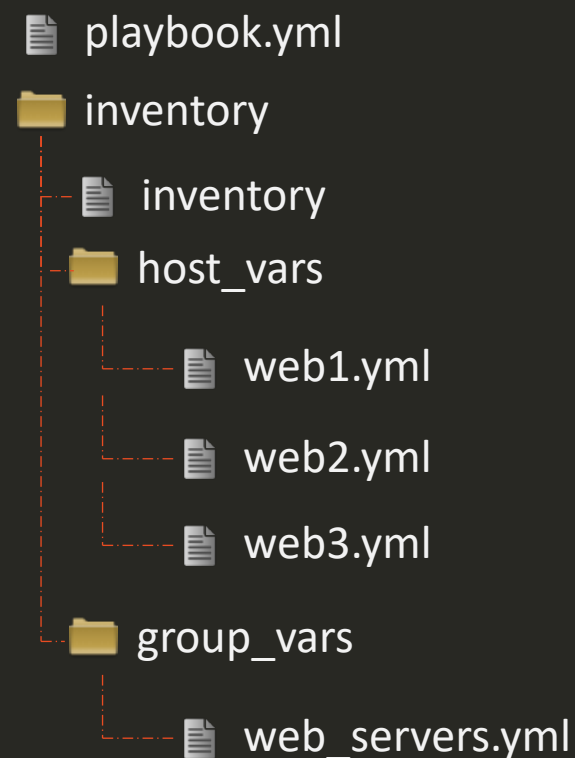
playbook.yml

```
- name: Deploy Web & DB Server
  hosts: web-db-server
  tasks:
    - include_vars:
        file: /opt/apps/common-data/email/info.yml
        name: email_data

    - mail:
        to: {{ email_data.admin_email }}
        subject: Service Alert
        body: Httpd Service is down
```

/opt/apps/common-data/email/info.yml

```
admin_email: admin@company.com
```

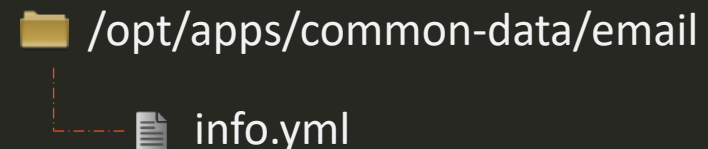
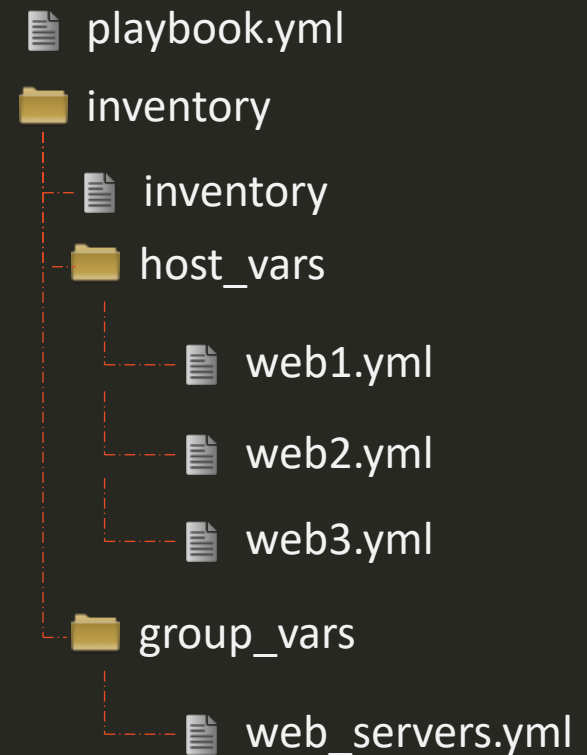


/opt/apps/common-data/email

info.yml

Ansible-Inventory

```
$ ansible-inventory -i inventory/ -y
all:
  children:
    ungrouped: {}
    web_servers:
      hosts:
        web1:
          ansible_host: 172.20.1.100
          ansible_ssh_pass: Passw0rd
          dns_server: 8.8.8.8
          size: big
        web2:
          ansible_host: 172.20.1.101
          ansible_ssh_pass: Passw0rd
          dns_server: 8.8.8.8
          size: small
```



Include Tasks

playbook.yml

```
- name: Deploy Web & DB Server
  hosts: web-db-server
  tasks:
    - name: Install MySQL Packages
      << code hidden >>

    - name: Start MySQL Service
      << code hidden >>

    - name: Configure Database
      << code hidden >>

    - name: Install Python Flask Dependencies
      << code hidden >>

    - name: Run web-server
      << code hidden >>
```

tasks/db.yml

tasks/web.yml

Include Tasks

playbook.yml

- name: Deploy Web & DB Server
hosts: web-db-server
tasks:
 - include_tasks: tasks/db.yml
 - include_tasks: tasks/web.yml

tasks/db.yml

- name: Install MySQL Packages
<< code hidden >>
- name: Start MySQL Service
<< code hidden >>
- name: Configure Database
<< code hidden >>

tasks/web.yml

- name: Install Python Flask Dependencies
<< code hidden >>
- name: Run web-server
<< code hidden >>

Include Tasks

playbook.yml

- name: Deploy Web & DB Server
hosts: web-db-server
tasks:
 - include_tasks: tasks/db.yml
 - include_tasks: tasks/web.yml

playbook-db.yml

- name: Deploy a DB Server
hosts: db-server
tasks:
 - include_tasks: tasks/db.yml

playbook-web.yml

- name: Deploy a Web Server
hosts: web-server
tasks:
 - include_tasks: tasks/web.yml

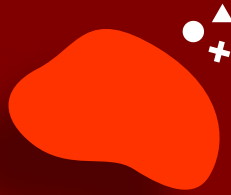
tasks/db.yml

- name: Install MySQL Packages
<< code hidden >>
- name: Start MySQL Service
<< code hidden >>
- name: Configure Database
<< code hidden >>

tasks/web.yml

- name: Install Python Flask Dependencies
<< code hidden >>
- name: Run web-server
<< code hidden >>





Ansible

Roles



Doctor



mysql



Engineer



nginx



Astronaut



redis



Police



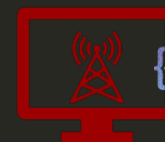
backup



Chef



monitor





Doctor



- Go to medical school
- Earn medical degree
- Complete Residency Program
- Obtain License



Engineer



- Go to engineering school
- Earn bachelor's degree
- Gain field experience
- Gain postgraduate degree



mysql



- Installing Pre-requisites
- Installing mysql packages
- Configuring mysql service
- Configuring database and users



nginx



- Installing Pre-requisites
- Installing nginx packages
- Configuring nginx service
- Configuring custom web pages

```
- name: Install and Configure MySQL
hosts: db-server
tasks:
  - name: Install Pre-Requisites
    yum: name=pre-req-packages state=present

  - name: Install MySQL Packages
    yum: name=mysql state=present

  - name: Start MySQL Service
    service: name=mysql state=started

  - name: Configure Database
    mysql_db: name=db1 state=present
```



mysql



- Installing Pre-requisites
- Installing mysql packages
- Configuring mysql service
- Configuring database and users



nginx



- Installing Pre-requisites
- Installing nginx packages
- Configuring nginx service
- Configuring custom web pages



Re-Use



mysql



- Installing Pre-requisites
- Installing mysql packages
- Configuring mysql service
- Configuring database and users

```
- name: Install and Configure MySQL
hosts: db-server1.....db-server100
roles:
  - mysql
```

MySQL-Role

tasks:

- name: Install Pre-Requisites
yum: name=pre-req-packages state=present
- name: Install MySQL Packages
yum: name=mysql state=present
- name: Start MySQL Service
service: name=mysql state=started
- name: Configure Database
mysql_db: name=db1 state=present



Organize



Re-Use



mysql



- Installing Pre-requisites
- Installing mysql packages
- Configuring mysql service
- Configuring database and users

MySQL-Role

tasks

```
tasks:
- name: Install Pre-Requisites
  yum: name=pre-req-packages state=present

- name: Install MySQL Packages
  yum: name=mysql state=present

- name: Start MySQL Service
  service: name=mysql state=started

- name: Configure Database
  mysql_db: name=db1 state=present
```

vars

```
mysql_packages:
- mysql
- mysql-server
db_config:
  db_name: db1
```

defaults

```
mysql_user_name: root
mysql_user_password: root
```

handlers

templates



ansistrano

rollback

Ansible role to rollback scripting applications like PHP, Python, Ruby, etc. in a Capistrano style



cloud

web

build passing

2.3 / 5 Score 61691 Downloads

Last Imported: 12 days ago



andrewrothst...

terraform

terraform role



cloud

infrastructure

terraform

4.2 / 5 Score 59591 Downloads

Last Imported: 8 days ago



sbaerlocher

do-agent

Cross-distro installation of the DigitalOcean monitoring agent



cloud

monitoring

build passing

42166 Downloads

Last Imported: a year ago



CyVerse-Ansible

ez

This role sets up the ez cli and other convenience functions commands by placing bash scripts into the /etc/profile.d of a system.



ansible

bash

cloud

cyverse

shell

35349 Downloads

Last Imported: 2 years ago

Login

LOUD



Organize



Re-Use



Share

```
$ ansible-galaxy init mysql
```

- mysql
 - README.md
 - templates
 - tasks
 - handlers
 - vars
 - defaults
 - meta

- my-playbook
 - playbook.yml
 - roles

playbook.yml

```
- name: Install and Configure MySQL
  hosts: db-server
  roles:
    - mysql
```



Organize

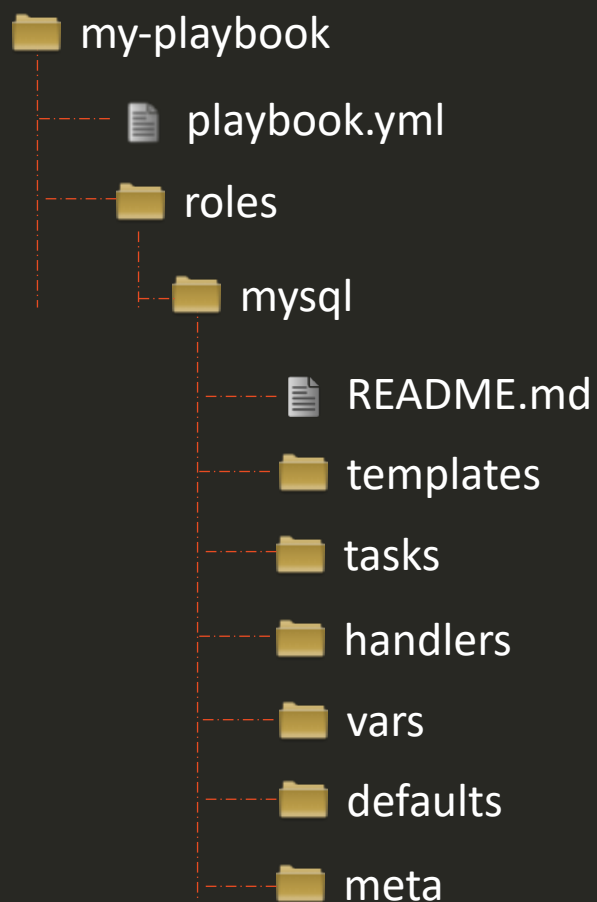


Re-Use



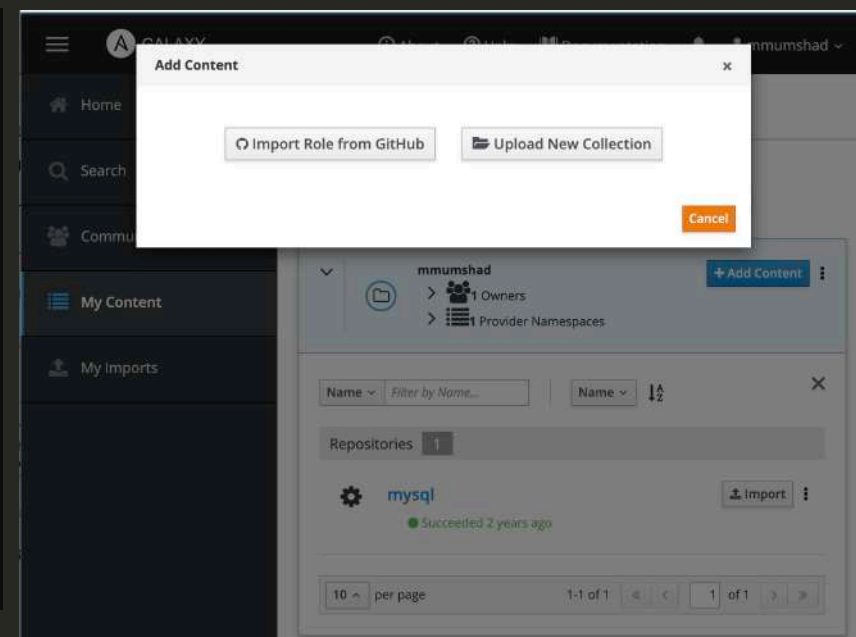
Share

```
$ ansible-galaxy init mysql
```



playbook.yml

```
- name: Install and Configure MySQL
  hosts: db-server
  roles:
    - mysql
```



Find Roles


Q Search

mysql (288 results)

Type Filter by Collection or Role... Best Match

288 Results Active filters: Tag: database Clear All Filters


Roles 288



mysql
MySQL server for RHEL/CentOS and Debian/Ubuntu.

database db mariadb mysql sql


3.2 / 5 Score 512737 Downloads
Last Imported: 5 days ago



php-mysql
PHP MySQL support for Linux.

database mysql php web


5 / 5 Score 133181 Downloads
Last Imported: 3 days ago



mysql
Install and configure mysql on your system.

alpine centos database debian fedora installer mariadb mysql package rhel ubuntu

4.8 / 5 Score 14762 Downloads
Last Imported: 5 days ago



mysql
MySQL server for RHEL/CentOS and Debian/Ubuntu.

database db mariadb mysql sql

5 / 5 Score 23304 Downloads
Last Imported: 4 months ago

```
$ ansible-galaxy search mysql
```

Found 1126 roles matching your search. Showing first 1000.

Name	Description
0utsider.ansible_zabbix_agent	Installing and maintaining zabbix-agent for
1mr.unattended	install and configure unattended upgrade
1nfinity.mysql	Simply installs MySQL 5.7 on Xenial.
4linuxdevops.mysql-server	Instalacao e Configuracao do servidor MySQL
5KYDEV0P5.skydevops-mysql	Install and configure MySQL Database
AAbouZaid.yourls	Manage Yourls, a URL shortener web app.
AAROC.AAROC_fg-db	your description
aaronpederson.ansible-autodeploy	Simple deployment tool with hooks
abednarik.mysql-exporter	Install and configure mysqld_exporter
abelboldu.openstack-glance	
abelboldu.openstack-keystone	
abelboldu.openstack-neutron-controller	OpenStack Neutron controller node
abelboldu.openstack-nova-controller	OpenStack Nova controller node
achaussier.mysql-backup	configure mysql-backup with xtrabackup and
achaussier.mysql-server	Install mysql-server package
achilleskal.ansible_mysql8	your description
adarnimrod.mysql	Provision a MySQL server

Use Role

```
$ ansible-galaxy install geerlingguy.mysql
```

- downloading role 'mysql', owned by geerlingguy
- downloading role from <https://github.com/geerlingguy/ansible-role-mysql/archive/2.9.5.tar.gz>
- extracting geerlingguy.mysql to `/etc/ansible/roles/geerlingguy.mysql`
- geerlingguy.mysql (2.9.5) was installed successfully

playbook.yml

```
-  
  name: Install and Configure MySQL  
  hosts: db-server  
  roles:  
    - geerlingguy.mysql
```

```
-  
  name: Install and Configure MySQL  
  hosts: db-server  
  roles:  
    - role: geerlingguy.mysql  
      become: yes  
      vars:  
        mysql_user_name: db-user
```

Use Role

Playbook-all-in-one.yml

```
-  
  name: Install and Configure MySQL  
  hosts: db-and-webserver  
  roles:  
    - geerlingguy.mysql  
    - nginx
```



mysql



Playbook-distributed.yml

```
-  
  name: Install and Configure MySQL  
  hosts: db-server  
  roles:  
    - geerlingguy.mysql  
  
-  
  name: Install and Configure Web Server  
  hosts: web-server  
  roles:  
    - nginx
```



mysql



nginx



List Roles

```
$ ansible-galaxy list
```

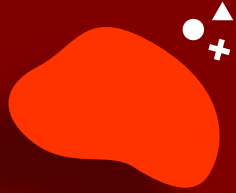
```
- geerlingguy.mysql  
- kodekloud1.mysql
```

```
$ ansible-config dump | grep ROLE
```

```
EFAULT_PRIVATE_ROLE_VARS(default) = False  
DEFAULT_ROLES_PATH(default) = [u'/root/.ansible/roles', u'/usr/share/ansible/roles', u'/etc/ansible/roles']  
GALAXY_ROLE_SKELETON(default) = None  
GALAXY_ROLE_SKELETON_IGNORE(default) = ['^\.git$', '^\.*/\.git_keep$']
```

```
$ ansible-galaxy install geerlingguy.mysql -p ./roles
```





Ansible Strategy

Strategy

```
- name: Deploy web application
hosts: server1
tasks:
  - name: Install dependencies
    << code hidden >>

  - name: Install MySQL Database
    << code hidden >>

  - name: Start MySQL Service
    << code hidden >>

  - name: Install Python Flask Dependencies
    << code hidden >>

  - name: Run web-server
    << code hidden >>
```

Dependencies

Install MySQL

Start DB

Install Flask

Run Server



Server1

Strategy - LINEAR

```
- name: Deploy web application
hosts: server1,server2,server3
tasks:
  - name: Install dependencies
    << code hidden >>

  - name: Install MySQL Database
    << code hidden >>

  - name: Start MySQL Service
    << code hidden >>

  - name: Install Python Flask Dependencies
    << code hidden >>

  - name: Run web-server
    << code hidden >>
```

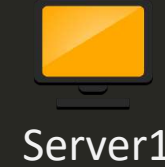
Dependencies

Install MySQL

Start DB

Install Flask

Run Server



Server1



Server2



Server3

Strategy - FREE

```
- name: Deploy web application
hosts: server1,server2,server3
strategy: free
tasks:
  - name: Install dependencies
    << code hidden >>

  - name: Install MySQL Database
    << code hidden >>

  - name: Start MySQL Service
    << code hidden >>

  - name: Install Python Flask Dependencies
    << code hidden >>

  - name: Run web-server
    << code hidden >>
```

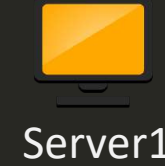
Dependencies

Install MySQL

Start DB

Install Flask

Run Server



Server1



Server2



Server3

Strategy - BATCH

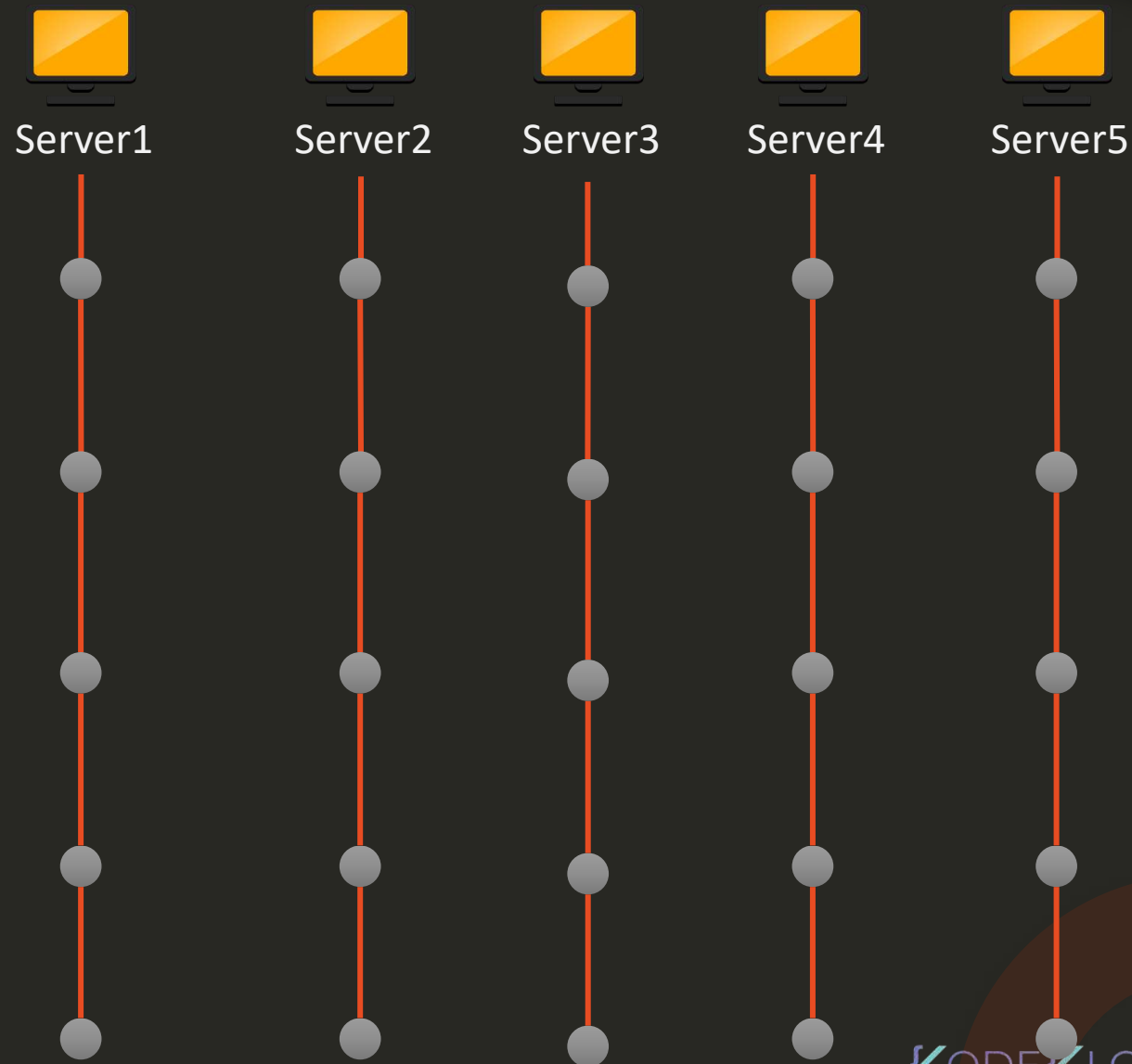
```
- name: Deploy web application
hosts: server1,server2,server3,server4,server5
serial: 3
tasks:
  - name: Install dependencies
    << code hidden >>

  - name: Install MySQL Database
    << code hidden >>

  - name: Start MySQL Service
    << code hidden >>

  - name: Install Python Flask Dependencies
    << code hidden >>

  - name: Run web-server
    << code hidden >>
```



forks

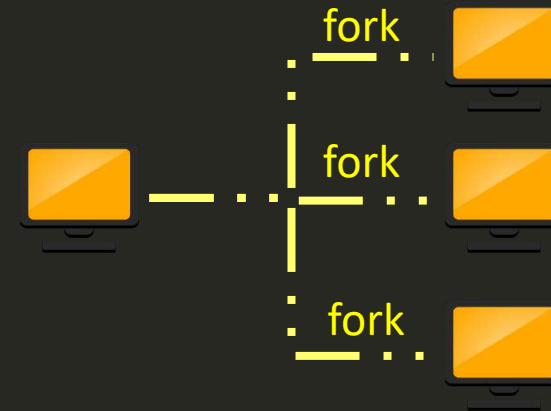
```
- name: Deploy web application
hosts: server1,server2,server3... server100
serial: 3
tasks:
  - name: Install dependencies
    << code hidden >>

  - name: Install MySQL Database
    << code hidden >>

  - name: Start MySQL Service
    << code hidden >>

  - name: Install Python Flask Dependencies
    << code hidden >>

  - name: Run web-server
    << code hidden >>
```



```
/etc/ansible/ansible.cfg
```

```
forks = 5
```






Ansible Vault

inventory

```
web1 ansible_host=172.20.1.100 ansible_ssh_pass=Passw0rd
web2 ansible_host=172.20.1.101 ansible_ssh_pass=Passw0rd
```

```
$ ansible-vault encrypt inventory
```

inventory

```
$ANSIBLE_VAULT;1.1;AES256
61383464383939633238383239356239666432313565333463636435326462363863323263636261
6432623864313032636434613931316262646534633165340a323664333661323961666361326430
62636562333738636638376631326233646130386133646438633739623362646238626438356265
6534663335386138370a623133653339356138623831306638383838363839303866303031643038
33373061653863303664383935316662623065316137343361313435313761303332633637333932
64623362623565396665393237356430653966616339643666393832346333636632663136306633
61343865376362643166356466653836613937666236626235646130633238393361396633613162
65633033386663383638323265646365363465366533313161313166323133633830306263663039
66633239633832366339336137336564646434343831323134323037356265386431643233346631
62636133653530393866666638643133636564366530366663633565386363366236323763363837
36383565663835623966643739666237626264353333363464346665333731323265623530353736
62343266386138336563356164333030616238306132666537623963393361363336313138633238
6137
```

```
$ ansible-playbook playbook.yml -i inventory
```

```
ERROR! Attempted to read "inventory.txt" as ini file: Decryption failed on inventory.txt
```

```
$ ansible-playbook playbook.yml -i inventory --ask-vault-pass
```

```
root@controller:/opt/first_project # ansible-playbook /tmp/temp_playbook.yml -i inventory.txt --ask-vault-pass
Vault password:

PLAY [Test Template playbook] *****

TASK [Gathering Facts] *****
ok: [target1]
```

```
$ ansible-playbook playbook.yml -i inventory -vault-password-file ~/.vault_pass.txt
```

```
$ ansible-playbook playbook.yml -i inventory -vault-password-file ~/.vault_pass.py
```

```
$ ansible-vault view inventory
```

```
$ ansible-vault create inventory
```

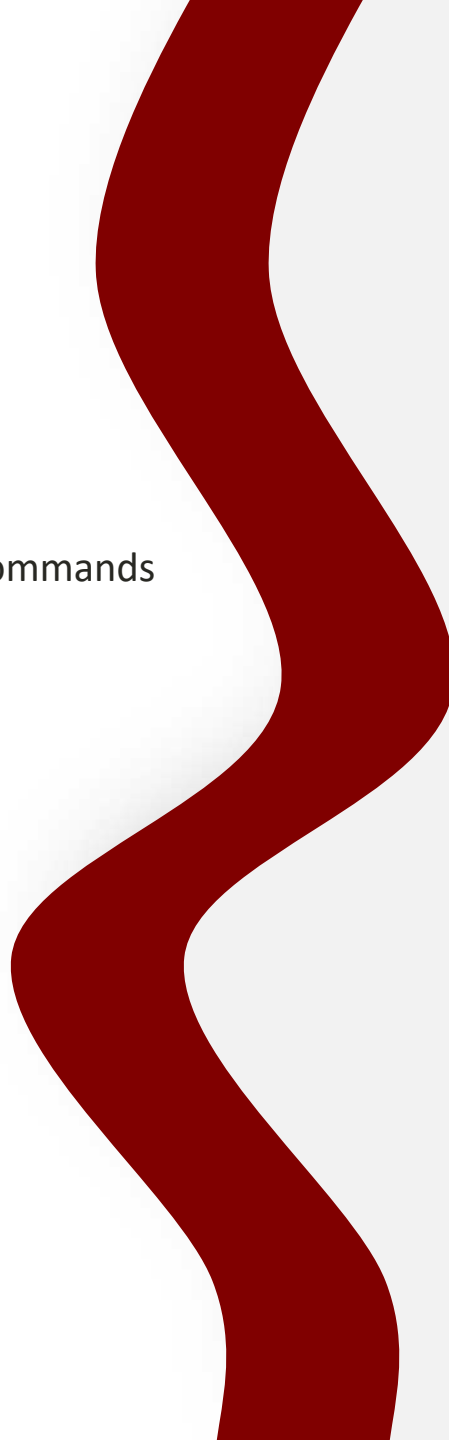


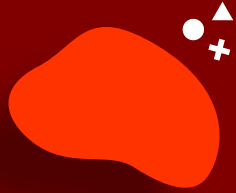


The Curriculum

Red Hat Certified Ansible Specialist

- Core Components
- Install and Configure Ansible Control Node
- Configure Ansible Managed Nodes
- Create simple shell scripts that run ad hoc Ansible commands
- Dynamic inventories
- Ansible Plays and Playbooks
- Ansible Modules
- Customized Configuration Files
- Variables and Facts
- Roles
- Ansible Vault
- Documentation





Ansible

Dynamic

Inventory

Static Inventory

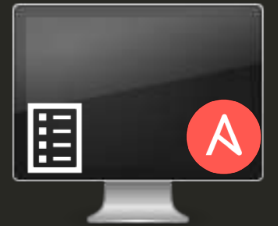
```
/etc/ansible/hosts
```

```
web1 ansible_host=172.20.1.100 ansible_ssh_pass=Passw0rd  
web2 ansible_host=172.20.1.101 ansible_ssh_pass=Passw0rd
```

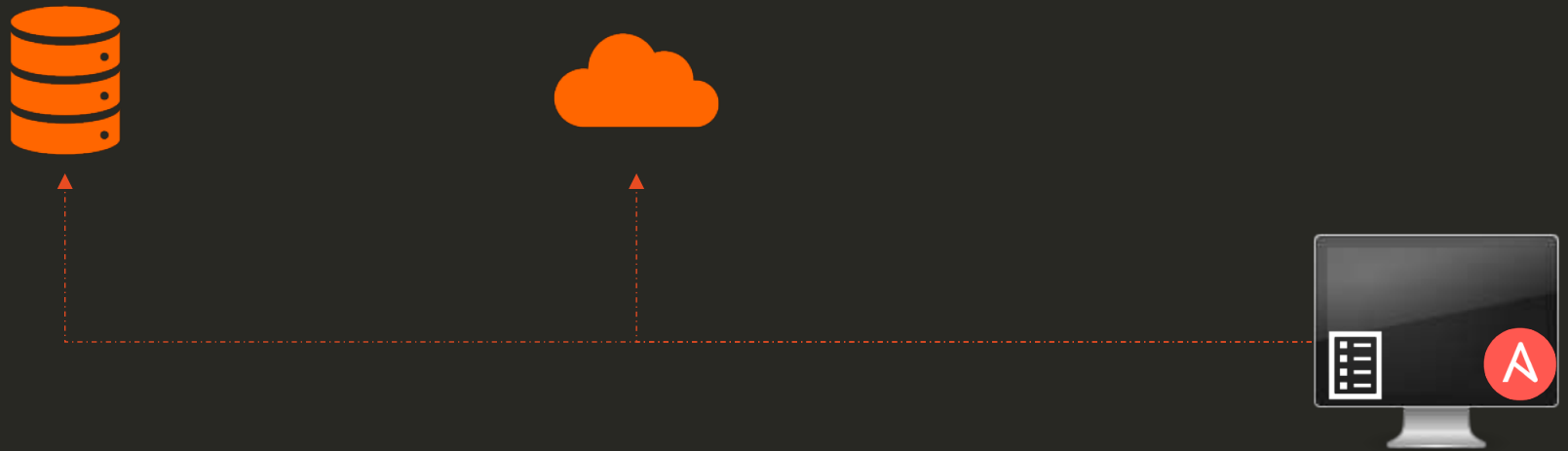
```
[web_servers]
```

```
web1
```

```
web2
```



Dynamic Inventory



inventory.txt

```
web1 ansible_host=172.20.1.100 ansible_ssh_pass=Passw0rd
web2 ansible_host=172.20.1.101 ansible_ssh_pass=Passw0rd
```

```
[web_servers]
web1
web2
```

```
$ ansible-playbook playbook.yml -i inventory.txt
```

```
$ ansible-playbook playbook.yml -i inventory.py
```

inventory.py

```
#!/usr/bin/env python

import json
import argparse

# Get inventory data from source - CMDB or any other API

def get_inventory_data():
    return {
        "web_servers": {
            "hosts": ["web1", "web2"]
        },
        "_meta": {
            "hostvars": {
                "web1": {
                    "ansible_host": "172.20.1.100",
                    "ansible_ssh_pass": "Passw0rd"
                },
                "web2": {
                    "ansible_host": "172.20.1.101",
                    "ansible_ssh_pass": "Passw0rd"
                }
            }
        }
    }

# Default main function

if __name__ == "__main__":
    read_cli_args();
    inventory_data = get_inventory_data()
    if args.list:
        print(json.dumps(inventory_data))
<Code Hidden>
```

+ Test Inventory Script

```
$ ./inventory.py --list
```

```
{
  "web_servers": {
    "hosts": [
      "web1",
      "web2"
    ]
  },
  "_meta": {
    "hostvars": {
      "web2": {
        "ansible_host": "172.20.1.101",
        "ansible_ssh_pass": "Passw0rd"
      },
      "web1": {
        "ansible_host": "172.20.1.100",
        "ansible_ssh_pass": "Passw0rd"
      }
    }
  }
}
```

```
$ ./inventory.py --host web1
```

```
{
  "ansible_host": "172.20.1.100",
  "ansible_ssh_pass": "Passw0rd"
}
```

inventory.py

```
#!/usr/bin/env python
```

```
import json
import argparse
```

```
# Get inventory data from source - CMDB or any other API
```

```
def get_inventory_data():
    return {
        "web_servers": {
            "hosts": ["web1", "web2"]
        },
        "_meta": {
            "hostvars": {
                "web1": {
                    "ansible_host": "172.20.1.100",
                    "ansible_ssh_pass": "Passw0rd"
                },
                "web2": {
                    "ansible_host": "172.20.1.101",
                    "ansible_ssh_pass": "Passw0rd"
                }
            }
        }
    }
```

```
# Default main function
```

```
if __name__ == "__main__":
    read_cli_args();
    inventory_data = get_inventory_data()
    if args.list:
        print(json.dumps(inventory_data))
<Code Hidden>
```

Inventory Scripts

[ansible](#) / [ansible](#)

Used by 8,418

Watch 2,031

Star 39,250

Fork 16,678

<> Code

Issues 4,062

Pull requests 1,990

Actions


Projects 26

Security

Insights

Tree: 3cd98a9fcc [ansible / contrib / inventory](#)

Create new file Upload files Find file History

 **samdoran** and **s-hertel** Use ansible.module_utils.six in inventory scripts (#55000) ...

Latest commit 3cd98a9 on 10 Apr

..

abiquo.ini	Fix some typos (#16498)	3 years ago
abiquo.py	Use ansible.module_utils.six in inventory scripts (#55000)	5 months ago
apache-libcloud.py	Use ansible.module_utils.six in inventory scripts (#55000)	5 months ago
apstra_aos.ini	[inventory/aos] Few fixes and small tweaks (#22259)	3 years ago
apstra_aos.py	Update bare exceptions to specify Exception.	9 months ago
azure_rm.ini	add group_by_os_family in azure dynamic inventory (#40702)	last year
azure_rm.py	Use ansible.module_utils.six in inventory scripts (#55000)	5 months ago
brook.ini	Clean up shebangs for various files.	3 years ago
brook.py	Use ansible.module_utils.six in inventory scripts (#55000)	5 months ago
cloudforms.ini	Add ability to select to prefer IPv4 addresses for ansible_ssh_host (#...	2 years ago
cloudforms.py	Use six.moves to import configparser in cloudforms script (#54465)	5 months ago
cloudstack.ini	poreted log_plays, syslog_json and osx_say callbacks to v2	4 years ago
cloudstack.py	cloudstack: inventory: consider more keys optional (#49364)	9 months ago
cobbler.ini	add cobbler api authentication options	3 years ago

EC2 Inventory Script

```
$ export AWS_ACCESS_KEY_ID=AK123  
$ export AWS_SECRET_ACCESS_KEY_ID=ABC123
```

```
$ ansible-playbook playbook.yml -i ec2.py
```

Ansible INI

```
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101
```

```
[web_servers]
web1
web2
```

Script

```
#!/usr/bin/env python

import json
import argparse

# Get inventory data from source - CMDB or
any other API

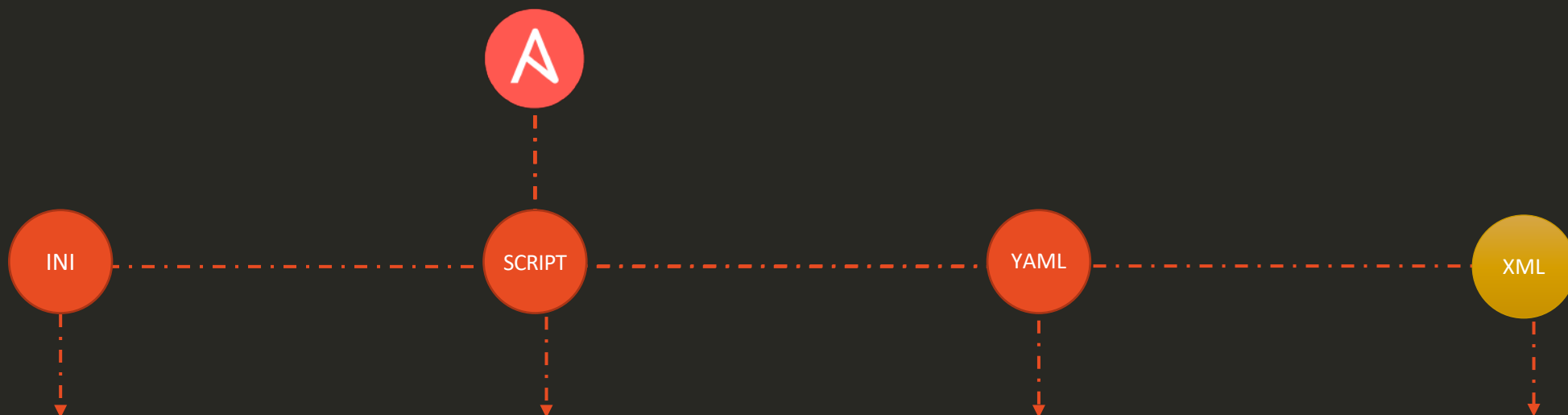
def get_inventory_data():
    return {
        "web_servers": {
            "hosts": ["web1", "web2"]
        },
        "_meta": {
            "hostvars": {
                "web1": {
                    "ansible_host": "172.20.1.100",
                    "ansible_ssh_pass": "Passw0rd"
                },
                "web2": {
                    "ansible_host": "172.20.1.101",
                    "ansible_ssh_pass": "Passw0rd"
                }
            }
        }
    }

# Default main function

if __name__ == "__main__":
    read_cli_args()
    inventory_data = get_inventory_data()
    if args.list:
```

YAML

```
web_servers:
  hosts:
    web1:
      ansible_host: 172.20.1.100
      ansible_ssh_pass: Passw0rd
    web2:
      ansible_host: 172.20.1.101
      ansible_ssh_pass: Passw0rd
```



Ansible INI

```
web1 ansible_host=172.20.1.100
web2 ansible_host=172.20.1.101
```

```
[web_servers]
web1
web2
```

Script

```
#!/usr/bin/env python

import json
import argparse

# Get inventory data from source - CMDB or
# any other API

def get_inventory_data():
    return {
        "web_servers": {
            "hosts": ["web1", "web2"]
        },
        "_meta": {
            "hostvars": {
                "web1": {
                    "ansible_host": "172.20.1.100",
                    "ansible_ssh_pass": "Passw0rd"
                },
                "web2": {
                    "ansible_host": "172.20.1.101",
                    "ansible_ssh_pass": "Passw0rd"
                }
            }
        }
    }
```

YAML

```
web_servers:
  hosts:
    web1:
      ansible_host: 172.20.1.100
      ansible_ssh_pass: Passw0rd
    web2:
      ansible_host: 172.20.1.101
      ansible_ssh_pass: Passw0rd
```

XML

```
<web_servers>>
  <hosts>:
    web1:
      ansible_host: 172.20.1.100
      ansible_ssh_pass: Passw0rd
    web2:
      ansible_host: 172.20.1.101
      ansible_ssh_pass: Passw0rd
  </
</web_servers>>
```

Inventory Plugin Configuration

```
/etc/ansible/ansible.cfg
```

```
[inventory]  
enable_plugins      = host_list, script, auto, yaml, ini
```


Inventory Scripts vs Plugins

Script

```
#!/usr/bin/env python

import json
import argparse

# Get inventory data from source - CMDB or
any other API

def get_inventory_data():
    return {
        "web_servers": {
            "hosts": ["web1", "web2"]
        },
        "_meta": {
            "hostvars": {
                "web1": {
                    "ansible_host": "172.20.1.100",
                    "ansible_ssh_pass": "Passw0rd"
                },
                "web2": {
                    "ansible_host": "172.20.1.101",
                    "ansible_ssh_pass": "Passw0rd"
                }
            }
        }
    }

# Default main function

if __name__ == "__main__":
    read_cli_args()
    inventory_data = get_inventory_data()
    if args.list:
        print(json.dumps(inventory_data))
<Code Hidden>
```

Plugins

```
# Make coding more python3-ish
from __future__ import absolute_import, division, print_function
__metaclass__ = type

import hashlib
import os
import string

from ansible.errors import AnsibleError, AnsibleParserError
from ansible.inventory.group import to_safe_group_name as original_safe
from ansible.parsing.utils.addresses import parse_address
from ansible.plugins import AnsiblePlugin
from ansible.plugins.cache import CachePluginAdjudicator as CacheObject
from ansible.module_utils._text import to_bytes, to_native

display = Display()

def expand_hostname_range line=None :
    all_hosts = []
    if line:
        (head, nrange, tail) = line.replace('[', '|').replace(']', '|').split('|')
        bounds = nrange.split(":")
        if len(bounds) != 2 and len(bounds) != 3:
            raise AnsibleError("host range must be begin:end or begin:end:step")
        beg = bounds[0]
        end = bounds[1]
        if len(bounds) == 2:
            step = 1
        else:
            step = bounds[2]
<Code Hidden>
```

Ansible-Inventory

```
$ ansible-inventory -i ec2.py
```

```
{
  "_meta": {
    "hostvars": {
      "172.20.1.109": {
        "ansible_ssh_pass": "Passw0rd"
        "ansible_ssh_user": "root"
        "ec2_region": "ca-central-1"
        "ec2_state": "Running"
      }
      "172.20.1.110": {
        "ansible_ssh_pass": "Passw0rd"
        "ansible_ssh_user": "root"
        "ec2_region": "us-east-1"
        "ec2_state": "Running"
      }
    }
  }
  "all": {
    "children":
    "group"
    "ungrouped"
  }
  "group": {
    "hosts":
    "172.20.1.109"
    "172.20.1.110"
  }
  "ungrouped": {}
}
```