

## **Getting Started**





www.kodekloud.com









## MUMSHAD MANNAMBE











I am an IT Solutions Architect and specializes in Cloud, Automation and DevOps and the author of Top 3% of Udemy's courses. 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 88,000 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 **②** 112,079

Courses

Reviews

27,993

#### Courses you're teaching



**Certified Kubernetes** Administrator (CKA) wit...

Mumshad Mannambeth

**\* \* \* \* \* 4.7** (2,181)



**Kubernetes Certified** Application Developer...

Mumshad Mannambeth, Kode K...

**★★★★ 4.6** (1,631)



Chef for the Absolute Beginners - DevOps

Mumshad Mannambeth, Yoges...

**★ ★ ★ ★ 4.5** (351)



DevOps - The Pre-**Requisite Course** 

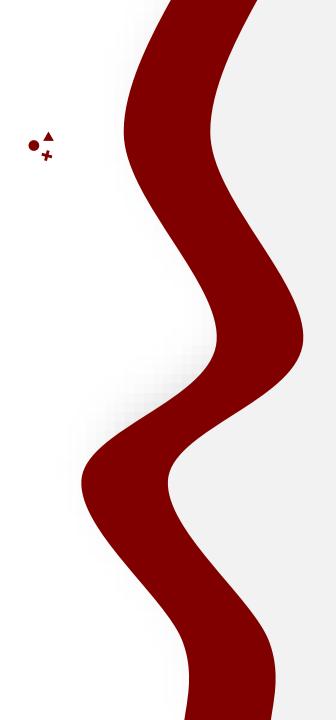
Mumshad Mannambeth

**★★★★★ 4.3** (630)





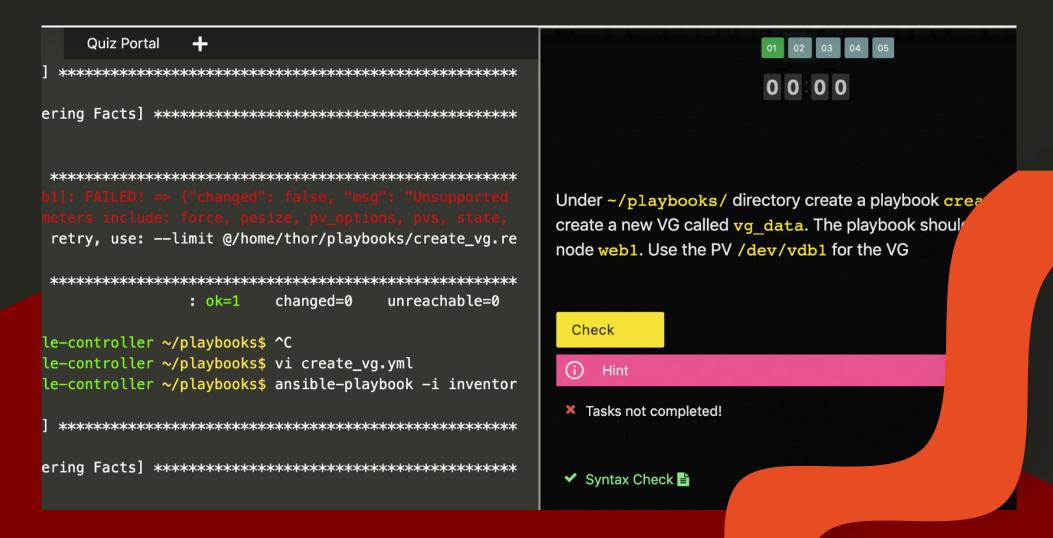
- Setting up Ansible on VirtualBox
- Introduction to YAML
- Inventory Files
- Playbooks
- Variables
- Conditionals
- Loops
- Roles







## Hands-On Exercises



### Note

- Do not copy code from this file directly as it may affect the formatting.
- Always refer to git repositories to access code.







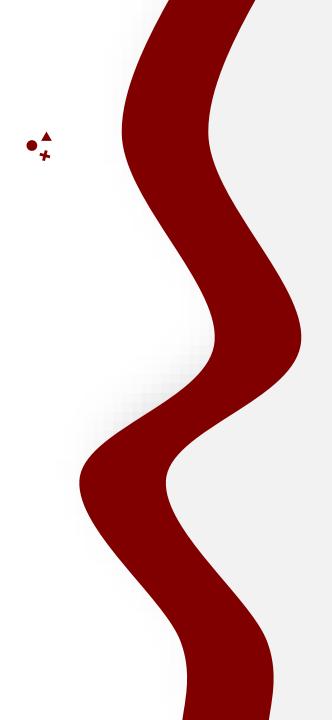


www.kodekloud.com





- Setting up Ansible on VirtualBox
- Introduction to YAML
- Inventory Files
- Playbooks
- Variables
- Conditionals
- Loops
- Roles







# Ansible

# Introduction



## Why Ansible?



Provisioning



Configuration Management



Continuous Delivery



Application Deployment



Security Compliance



Scripts

- **Coding Skills**
- Maintenance



- Simple
- Powerful
- Agentless





## Scripts

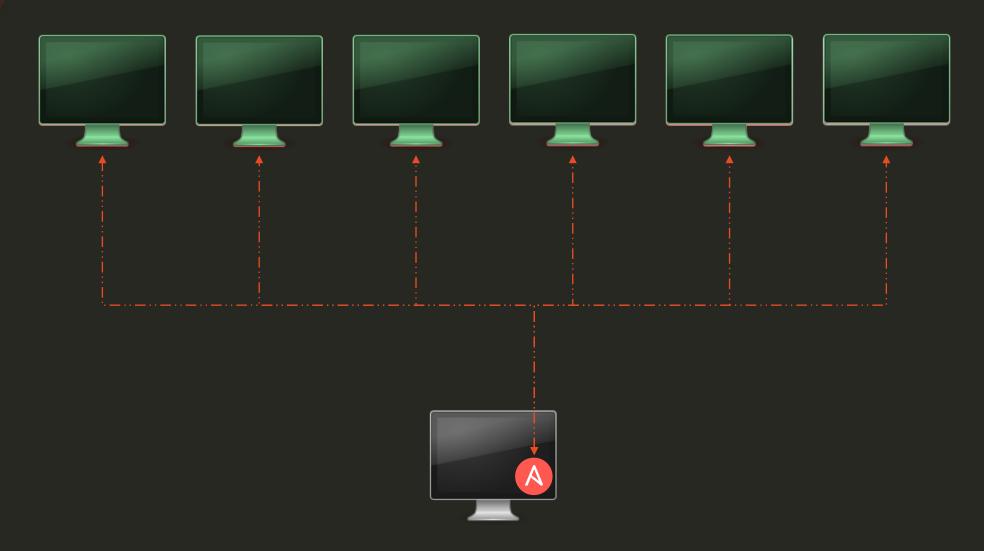
## vs Ansible Playbook

```
#!/bin/bash
# Script to add a user to Linux system
if [ $(id -u) -eq 0 ]; then
    $username=johndoe
    read -s -p "Enter password : " password
    egrep "^$username" /etc/passwd >/dev/null
    if [ $? -eq 0 ]; then
        echo "$username exists!"
        exit 1
    else
        useradd -m -p $password $username
        [ $? -eq 0 ] && echo "User has been added
to system!" || echo "Failed to add a user!"
    fi
fi
```

```
- hosts: all_my_web_servers_in_DR
  tasks:
    - user:
        name: johndoe
```

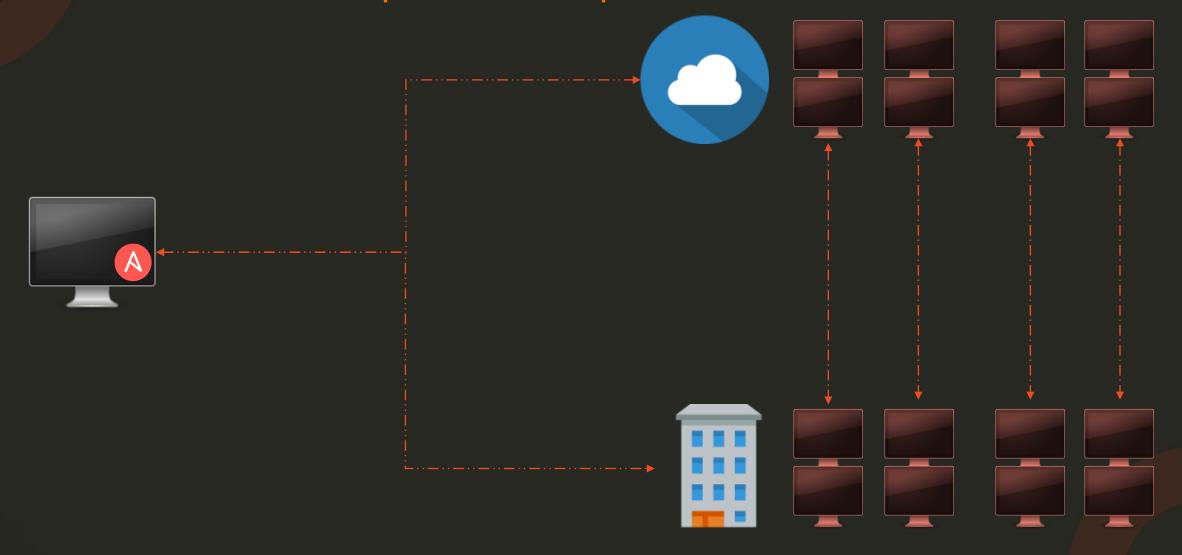


## Use case example - Simple



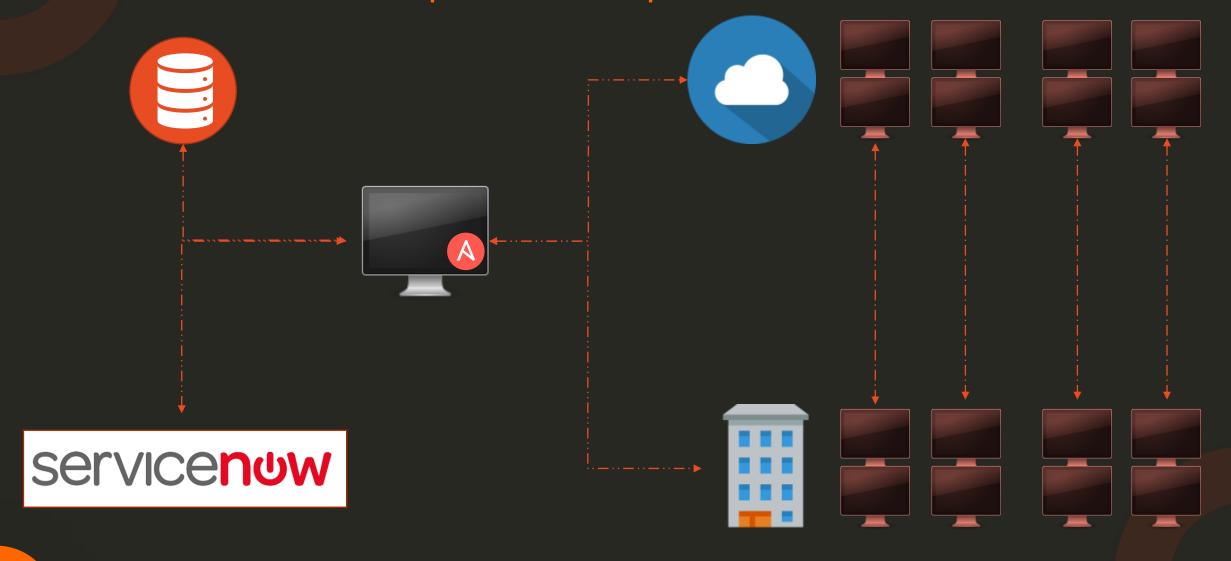


## Use case example - complex





## Use case example - complex







## Ansible Documentation







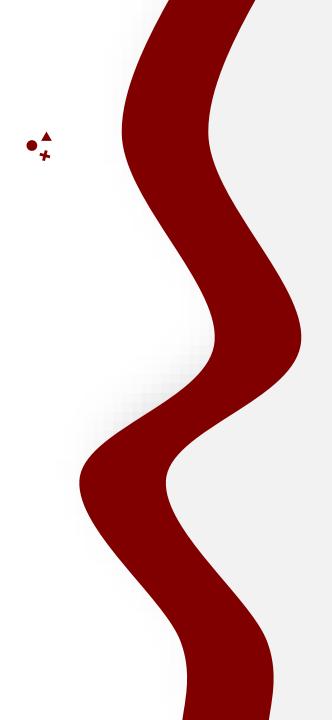


www.kodekloud.com





- Setting up Ansible on VirtualBox
- Introduction to YAML
- Inventory Files
- Playbooks
- Variables
- Conditionals
- Loops
- Roles







# Ansible

# Install







Redhat or CentOS - \$ sudo yum install ansible



Fedora –



Ubuntu –

\$ sudo dnf install ansible

\$ 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







## +

## Install Control Node on Redhat or CentOS





### 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

### Install Specific Version of Ansible using pip

\$ sudo pip install ansible==2.4

### Upgrade Ansible using pip

\$ sudo pip install --upgrade ansible









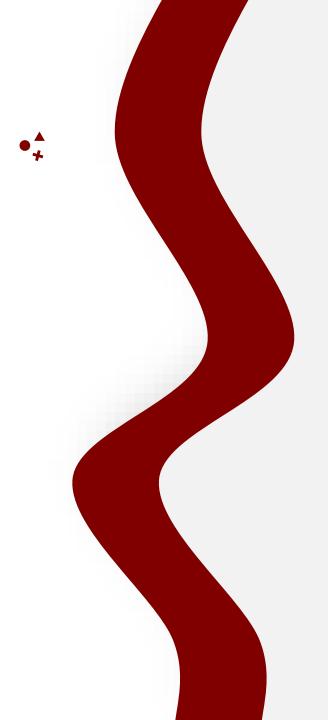


www.kodekloud.com





- Setting up Ansible on VirtualBox
- Introduction to YAML
- Inventory Files
- Playbooks
- Variables
- Conditionals
- Loops
- Roles







# Ansible

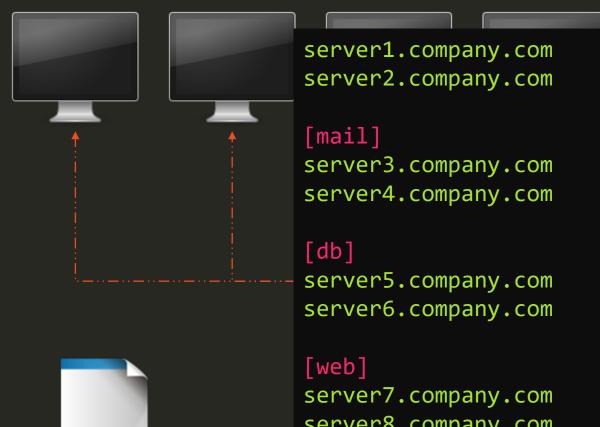
# Inventory



## inventory

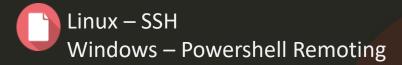
inventory

/etc/ansible/hosts





server8.company.com







## More on inventory files

```
server1.company.com
server2.company.com
server3.company.com
server4.company.com
```

```
ansible_connection=ssh
ansible_connection=winrm ansible_user=admin
ansible_connection=ssh
ansible_connection=winrm
```

ansible\_user=root ansible\_ssh\_pass=P@#

```
localhost ansible_connection=localhost
```



### **Inventory Parameters:**

- ansible connection ssh/winrm/localhost
- ansible\_port 22/5986
- ansible\_user root/administrator
- ansible ssh pass Password



Security: Ansible Vault









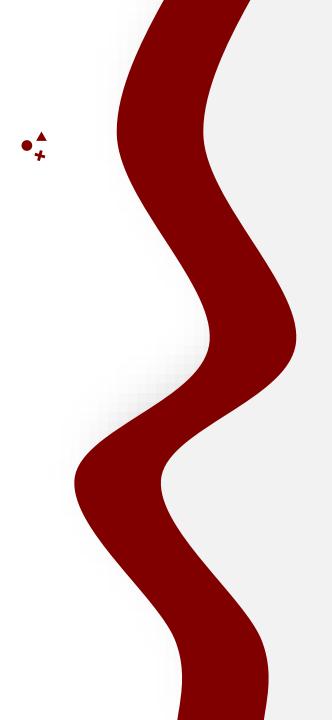


www.kodekloud.com





- Setting up Ansible on VirtualBox
- Introduction to YAML
- Inventory Files
- Playbooks
- Variables
- Conditionals
- Loops
- Roles







# Ansible

# Playbooks



## Ansible playbooks

#### # Simple Ansible Playbook

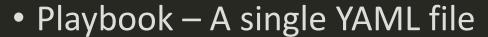
- Run command1 on server1
- Run command2 on server2
- Run command3 on server3
- Run command4 on server4
- Run command5 on server5
- Run command6 on server6
- Run command7 on server7
- Run command8 on server8
- Run command9 on server9
- Restarting Server1
- Restarting Server2
- Restarting Server3
- Restarting Server4
- Restarting Server5
- Restarting Server6
- Restarting Server7

#### # Complex Ansible Playbook

- Deploy 50 VMs on Public Cloud
- Deploy 50 VMs on Private Cloud
- Provision Storage to all VMs
- Setup Network Configuration on Private VMs
- Setup Cluster Configuration
- Configure Web server on 20 Public VMs
- Configure DB server on 20 Private VMs
- Setup Loadbalancing between web server VMs
- Setup Monitoring components
- Install and Configure backup clients on VMs
- Update CMDB database with new VM Information



## Playbook



- Play Defines a set of activities (tasks) to be run on hosts
  - Task An action to be performed on the host
    - Execute a command
    - > Run a script
    - > Install a package
    - ➤ Shutdown/Restart



### playbook.yml

name: Play 1

hosts: localhost

tasks:

- name: Execute command 'date'

command: date

- name: Execute script on server

script: test\_script.sh

- name: Install httpd service

yum:

name: httpd
state: present

- name: Start web server

service:

name: httpd
state: started



## Playbook format

playbook.yml

```
name: Play 1
 hosts: localhost
 tasks:
         - name: Execute command 'date'
          command: date
         - name: Execute script on server
          script: test_script.sh
name: Play 2
hosts: localhost
tasks:
       - name: Install web service
         yum:
                name: httpd
                state: present
         hame: Start web server
         service:
                name: httpd
                state: started
```



## Hosts

### playbook.yml

```
name: Play 1
hosts: localhost
tasks:
 - name: Execute command 'date'
   command: date
 - name: Execute script on server
   script: test_script.sh
 - name: Install httpd service
   yum:
    name: httpd
    state: present
 - name: Start web server
   service:
    name: httpd
    state: started
```

### inventory

### localhost

server1.company.com
server2.company.com

### [mail]

server3.company.com
server4.company.com

### [db]

server5.company.com
server6.company.com

### [web]

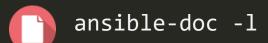
server7.company.com
server8.company.com





### playbook.yml

```
name: Play 1
hosts: localhost
tasks:
- name: Execute command 'date'
   command: date
 - name: Execute script on server
  script: test_script.sh
 - name: Install httpd service
  yum:
   name: httpd
   state: present
 - name: Start web server
   service
   name: httpd
   state: started
```

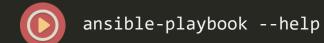






- Execute Ansible Playbook
- Syntax: ansible-playbook <playbook file name>













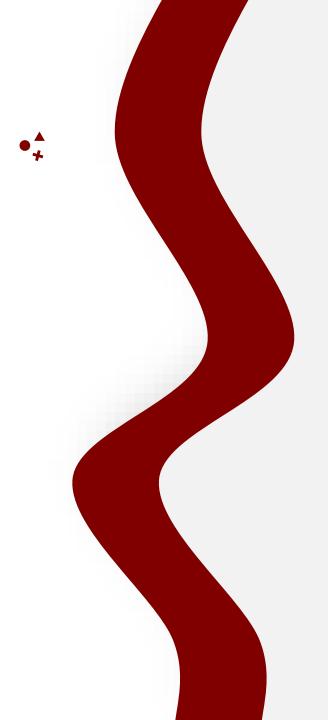


www.kodekloud.com





- Setting up Ansible on VirtualBox
- Introduction to YAML
- Inventory Files
- Playbooks
- Modules
- Variables
- Conditionals
- Loops
- Roles







## Ansible

# Modules



## modules

- System
- Commands
- Files
- Database
- Cloud
- Windows
- More..

- Win\_copy
- Win\_command
- Win\_domain
- Win\_file
- Win\_iis\_website
- Win\_msg
- Win\_msi
- Win\_package
- Win\_ping
- Win path
- Win\_robocopy
- Win\_regedit
- Win\_shell
- Win\_service
- Win\_user
- And more



## command

### Executes a command on a remote node

parameter chdir	comments cd into this directory before running the command
creates	a filename or (since 2.0) glob pattern, when it already exists, this step will <b>not</b> be run.
executable	change the shell used to execute the command. Should be an absolute path to the executable.
free_form	the command module takes a free form command to run. There is no parameter actually named 'free form'. See the examples!
removes	a filename or (since 2.0) glob pattern, when it does not exist, this step will <b>not</b> be run.
warn (added in 1.8)	if command warnings are on in ansible.cfg, do not warn about this particular line if set to no/false.

### playbook.yml

name: Play 1

hosts: localhost

tasks:

- name: Execute command 'date'

command: date

- name: Display resolv.conf contents

command: cat /etc/resolv.conf

- name: Display resolv.conf contents
command: cat resolv.conf chdir=/etc

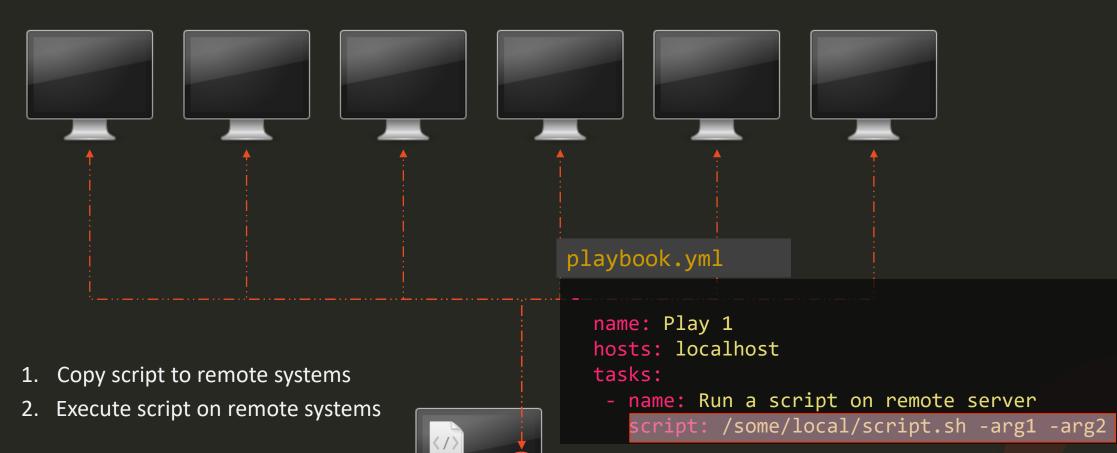
- name: Display resolv.conf contents
command: mkdir /folder creates=/folder

- name: Copy file from source to destination
copy: src=/source\_file dest=/destination



## script

• Runs a local script on a remote node after transferring it





### Service

• Manage Services – Start, Stop, Restart

### playbook.yml

name: Start Services in order
hosts: localhost

tasks:

- name: Start the database service

service: name=postgresql state=started

- name: Start the httpd service
 service: name=httpd state=started

- name: Start the nginx service

service:

name: nginx
state: started

### playbook.yml

name: Start Services in order

hosts: localhost

tasks:

- name: Start the database service

service:

name: postgresql
state: started



## idempotency

Why "started" and not "start"?

"Start" the service httpd "Started" the service httpd

Ensure service httpd is started

If httpd is not already started => start it
If httpd is already started, =>do nothing

#### Idempotency

An operation is idempotent if the result of performing it once is exactly the same as the result of performing it repeatedly without any intervening actions.



### lineinfile

Search for a line in a file and replace it or add it if it doesn't exist.

```
nameserver 10.1.250.1
nameserver 10.1.250.2

playbook.yml
-
   name: Add DNS server to resolv.conf
   hosts: localhost
   tasks:
    - lineinfile:
        path: /etc/resolv.conf
        line: 'nameserver 10.1.250.10'
```

### /etc/resolv.conf

/etc/resolv.conf

```
nameserver 10.1.250.1
nameserver 10.1.250.2
nameserver 10.1.250.10
```

nameserver 10.1.250.10

#### script.sh

```
#Sample script
echo "nameserver 10.1.250.10" >> /etc/resolv.conf
```

#### /etc/resolv.conf

```
nameserver 10.1.250.1
nameserver 10.1.250.2
nameserver 10.1.250.10
nameserver 10.1.250.10
```









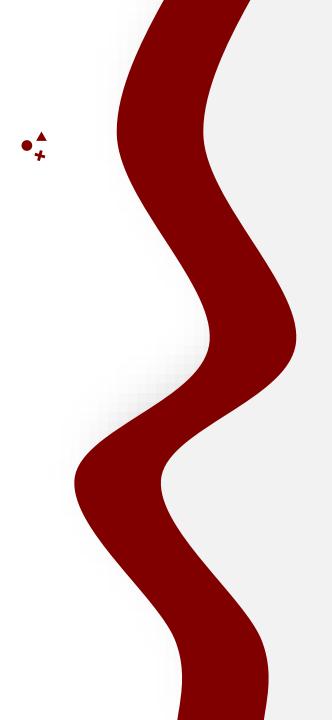


www.kodekloud.com





- Setting up Ansible on VirtualBox
- Introduction to YAML
- Inventory Files
- Playbooks
- Modules
- Variables
- Conditionals
- Loops
- Roles







## Ansible

# Variables



### Variable

Stores information that varies with each host

### inventory

```
Web1 ansible_host=server1.company.com ansible_connection=ssh ansible_shh_pass=P@ssW db ansible_host=server2.company.com ansible_connection=winrm ansible_shh_pass=P@ssW Web2 ansible_host=server3.company.com ansible_connection=ssh ansible_shh_pass=P@ssW
```

### Playbook.yml

```
name: Add DNS server to resolv.conf
hosts: localhost

twasks:
    dns_lserevierfill0.1.250.10
        path: /etc/resolv.conf
        line: 'nameserver 10.1.250.10'
```

#### variables

variable1: value1
variable2: value2



## Using variables

```
Playbook.yml
-
  name: Add DNS server to resolv.conf
hosts: localhost
  vars:
    dns_server: 10.1.250.10

  tasks:
    - lineinfile:
        path: /etc/resolv.conf
        line: 'nameserver {{ dns_server }}'
```



```
name: Set Firewall Configurations
hosts: web
tasks:
- firewalld:
   service: https
   permanent: true
   state: enabled
```

- firewalld:

port: 0{{/http\_port }}'/tcp permanent: true

state: disabled

- firewalld:

port: '{{-snmp\_port }}'/udp

permanent: true state: disabled

- firewalld:

source: '{{ inter\_ip\_range }}'/24

Zone: internal state: enabled

```
#Sample Inventory File
Web http_port=
                                   inter ip range=
                 snmp port=
#Sample variable File - web.yml
```

http\_port: 8081 snmp\_port: 161-162

inter\_ip\_range: 192.0.2.0

```
{{
       }}
```

Jinja2 Templating

```
source: {{ inter_ip_range }}
```

source: '{{ inter\_ip\_range }}'

source: SomeThing{{ inter\_ip\_range }}SomeThing







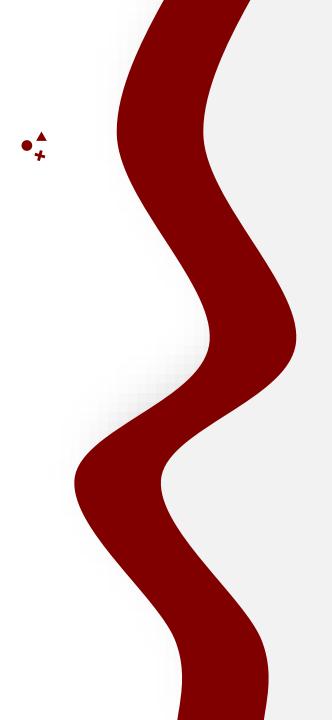


www.kodekloud.com





- Setting up Ansible on VirtualBox
- Introduction to YAML
- Inventory Files
- Playbooks
- Modules
- Variables
- Loops
- Conditionals
- Roles







## Ansible

# Loops



## LOOPS

- rani

name: Create users hosts: localhost tasks: - user: name=fo(€ item }}' state=present - loep: name=george state=present - userjoeame=ravi state=present - usergeomage=mani state=present - usernamame=kiran state=present - usermamiame=jazlan state=present - userkimame=emaan state=present userjaznbane=mazin state=present useremaame=izaan state=present - usermaname=mike state=present userizamame=menaal state=present usermikæme=shoeb state=present - usermemamb=rani state=present shoeb



```
name: Create users
hosts: localhost
tasks:
- user: name='{{ item }}' state=present
  loop:
    - joe
    george
    - ravi
    - mani
    - kiran
    - jazlan
    - emaan
    mazin
    - izaan
    - mike
    menaal
    shoeb
    - rani
```

```
name: Create users
hosts: localhost
tasks:
- var: item=
   user: name= "{{ item }}"
                                state=present
  var: item=
   user: name= "{{ item }}"
                                state=present
- var: item=
  user: name= "{{ item }}"
                               state=present
- var: item=
  user: name= "{{ item }}"
                               state=present
- var: item=
  user: name= "{{ item }}"
                               state=present
- var: item=
  user: name= "{{ item }}"
                               state=present
- var: item=
  user: name= "{{ item }}"
                               state=present
  var: item=
  user: name= "{{ item }}"
                               state=present
  var: item=
  user: name= "{{ item }}"
                               state=present
```

```
name: Create users
hosts: localhost
tasks:
- user: name='{{ item }}'
                            state=present
  loop:
    - joe
    george
    - ravi
    - mani
    - kiran
    jazlan
    emaan
    - mazin
    izaan
    - mike
    - menaal
    - shoeb
    - rani
```

```
name: Create users
hosts: localhost
tasks:
- var: item=joe
   user: name= "{{ item }}"
                                state=present
        item=george
  var:
   user: name= "{{ item }}"
                                state=present
       item=ravi
- var:
  user: name= "{{ item }}"
                                state=present
- var: item=mani
  user: name= "{{ item }}"
                                state=present
        item=kiran
  var:
  user: name= "{{ item }}"
                                state=present
        item=jazlan
- var:
  user: name= "{{ item }}"
                                state=present
        item=emaan
  var:
  user: name= "{{ item }}"
                                state=present
        item=mazin
  var:
  user: name= "{{ item }}"
                                state=present
       item=izaan
  var:
  user: name= "{{ item }}"
                                state=present
```

```
name: Create users
hosts: localhost
tasks:
- user: name '{{ ???? }}' state=present uid= '{{ ? }}'
  loop:
    - name: joe
      uid: 1010
    - name: george
      uid: 1011
    - name: ravi
      uid: 1012
    - name: mani
      uid: 1013
    - name: kiran
      uid: 1014
    - name: jazlan
      uid: 1015
    - name: emaan
      uid: 1016
    - name: mazin
      uid: 1017
    - name: izaan
      uid: 1018
    - name: mike
```

```
name: Create users
hosts: localhost
tasks:
- var: item=joe
   user: name= "{{ item }}"
                                state=present
        item=george
  var:
   user: name= "{{ item }}"
                                state=present
       item=ravi
- var:
  user: name= "{{ item }}"
                                state=present
- var: item=mani
  user: name= "{{ item }}"
                                state=present
        item=kiran
  var:
  user: name= "{{ item }}"
                                state=present
        item=jazlan
- var:
  user: name= "{{ item }}"
                                state=present
        item=emaan
  var:
  user: name= "{{ item }}"
                                state=present
        item=mazin
  var:
  user: name= "{{ item }}"
                                state=present
       item=izaan
  var:
  user: name= "{{ item }}"
                                state=present
```

```
name: Create users
hosts: localhost
tasks:
- user: name '{{ ???? }}' state=present uid= '{{ ? }}'
  loop:
    - name: joe
      uid: 1010
    - name: george
      uid: 1011
    - name: ravi
      uid: 1012
    - name: mani
      uid: 1013
    - name: kiran
      uid: 1014
    - name: jazlan
      uid: 1015
    - name: emaan
      uid: 1016
    - name: mazin
      uid: 1017
    - name: izaan
      uid: 1018
    - name: mike
```

```
name: Create users
hosts: localhost
tasks:
- var:
    item:
   user: name= "{{ ???? }}" state=present uid="{?}"
  var:
    item:
   user: name= "{{ ???? }}" state=present uid="{?}"
- var:
    item:
   user: name= "{{ ???? }}" state=present uid="{?}"
- var:
    item:
   user: name= "{{ ???? }}" state=present uid="{?}"
```

```
name: Create users
hosts: localhost
tasks:
 - user: name '{{ ???? }}' state=present uid= '{{ ? }}'
  loop:
    - name: joe
      uid: 1010
    - name: george
      uid: 1011
    - name: ravi
      uid: 1012
    - name: mani
      uid: 1013
    - name: kiran
      uid: 1014
    - name: jazlan
      uid: 1015
    - name: emaan
      uid: 1016
    - name: mazin
      uid: 1017
    - name: izaan
      uid: 1018
    - name: mike
```

```
name: Create users
hosts: localhost
tasks:
   var:
    item:
      name: joe
      uid: 1010
   user: name='{{ item.name }}' state=present uid='{{ item.uid
   var:
     item:
      name: george
      uid: 1011
    user: name='{{ item.name }}' state=present uid='{{ item.uid
   var:
    item:
      name: ravi
      uid: 1012
   user: name='{{ item.name }}' state=present uid='{{ item.uid
   var:
     item:
      name: mani
      uid: 1013
   user: name='{{ item.name }}' state=present uid='{{ item.uid
```

```
name: Create users
hosts: localhost
tasks:
- user: name= '{{ item.name }}' state=present uid='{{ item.uid }}'
  loop:
                        { name: joe, uid: 1010 }
    - name: joe
      uid: 1010
                    - { name: george, uid: 1011 }
    - name: george
      uid: 1011
    - name: ravi
                        { name: ravi, uid: 1012 }
      uid: 1012
    - name: mani
                        { name: mani, uid: 1013 }
      uid: 1013
                    - { name: kiran, uid: 1014 }
    - name: kiran
      uid: 1014
    - name: jazlan -
                       { name: jazlan, uid: 1015 }
      uid: 1015
                        { name: emaan, uid: 1016 }
    - name: emaan
      uid: 1016
    - name: mazin
                    - { name: mazin, uid: 1017 }
      uid: 1017
    - name: izaan
                        { name: izaan, uid: 1018 }
      uid: 1018
    - name: mike
                    - { name: mike, uid: 1019 }
```

```
name: Create users
hosts: localhost
tasks:
- var:
    item:
      name: joe
      uid: 1010
   user: name='{{ item.name }}' state=present uid='{{ item.uid
   var:
    item:
      name: george
      uid: 1011
    user: name='{{ item.name }}' state=present uid='{{ item.uid
   var:
    item:
      name: ravi
      uid: 1012
   user: name='{{ item.name }}' state=present uid='{{ item.uid
   var:
    item:
      name: mani
      uid: 1013
    user: name='{{ item.name }}' state=present uid='{{ item.uid
```

## With \*

```
name: Create users
hosts: localhost
tasks:
  - user: name='{{ item }}' state=present
  loop:
        - joe
        - george
        - ravi
        - mani
```

```
name: Create users
hosts: localhost
tasks:
   - user: name='{{ item }}' state=present
   with_items:
        - joe
        - george
        - ravi
        - mani
```



## With \*

```
name: Create users
hosts: localhost
tasks:
  - user: name='{{ item }}' state=present
    with_items:
        - joe
        - george
        - ravi
        - mani
```

```
name: Get from multiple URLs
hosts: localhost
tasks:
  - debug: var=item
    with_url:
    - "https://site1.com/get-servers"
    - "https://site2.com/get-servers"
    - "https://site3.com/get-servers"
```

```
name: View Config Files
hosts: localhost
tasks:
  - debug: var=item
    with_file:
    - "/etc/hosts"
    - "/etc/resolv.conf"
    - "/etc/ntp.conf"
```

```
name: Check multiple mongodbs
hosts: localhost
tasks:
  - debug: msg="DB={{ item.database }} PID={{ item.pid}}"
    with_mongodb:
        - database: dev
        connection_string: "mongodb://dev.mongo/"
        - database: prod
        connection_string: "mongodb://prod.mongo/"
```



## With \*

```
with_items
with_file
with_url
with_mongodb
with_dict
with_etcd
with env
with filetree
With_ini
With_inventory_hostnames
With k8s
With_manifold
With_nested
With_nios
With_openshift
With_password
With_pipe
With rabbitmq
```

With\_redis
With\_sequence
With\_skydive
With\_subelements
With\_template
With\_together
With\_varnames









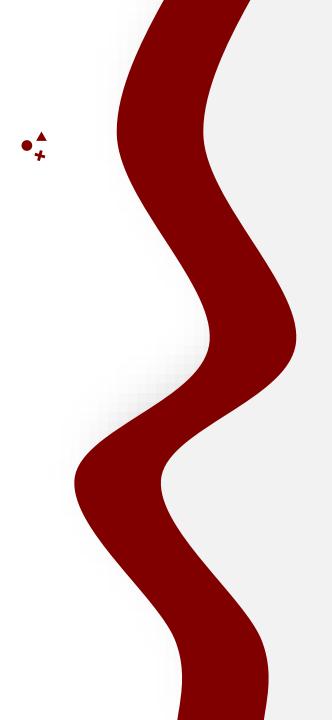


www.kodekloud.com





- Setting up Ansible on VirtualBox
- Introduction to YAML
- Inventory Files
- Playbooks
- Modules
- Variables
- Loops
- Conditionals
- Roles







## 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
     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"
          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"
          ansible_distribution_version == "16.04"
 - name: Install NGINX on Redhat
   yum:
     name: nginx
     state: present
   when: ansible_os_family == "RedHat"
          ansible_os_family == "SUSE"
```





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



# 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
         item.required == True
  when:
```

# 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
```











www.kodekloud.com







# Ansible

# Roles



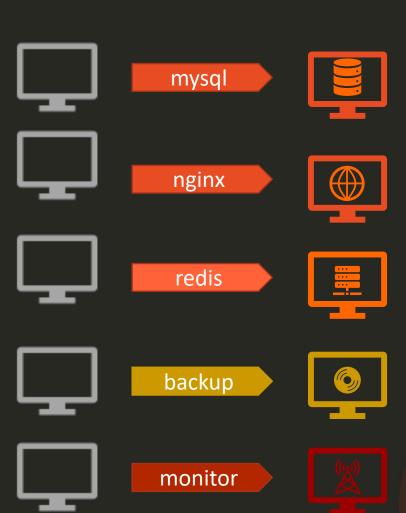


















#### Doctor



- 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













- 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







#### rollback

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



build passing

**9.2.3 / 5** Score **\$** 61691 Downloads Last Imported: 12 days ago



A GALAXY



terraform terraform role



infrastructure

terraform

**♦ 4.2 / 5** Score **♣** 59591 Downloads Login Last Imported: 8 days ago

**Q** Search

Home





sbaerlocher

do-agent

Cross-distro installation of the DigitalOcean monitoring agent



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.



cloud cyverse

♣ 35349 Downloads Last Imported: 2 years ago





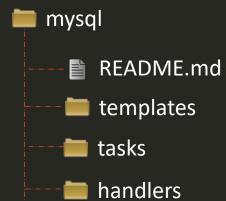


Organize

Re-Use

Share

\$ ansible-galaxy init mysql



nars vars

defaults

**m**eta



playbook.yml

- name: Install and Configure MySQL

hosts: db-server

roles:

mysql







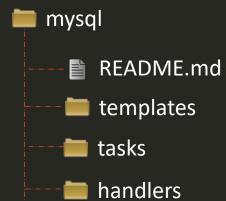


Organize

Re-Use

Share

\$ ansible-galaxy init mysql



nars vars

defaults

**m**eta



playbook.yml

- name: Install and Configure MySQL

hosts: db-server

roles:

mysql









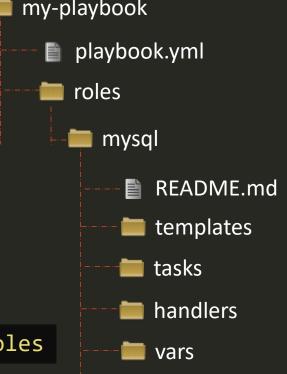
Share

#### \$ ansible-galaxy init mysql



/etc/ansible/ansible.cfg

roles\_path = /etc/ansible/roles



defaults

**m**eta

## playbook.yml

- name: Install and Configure MySQL

hosts: db-server

roles:

mysql



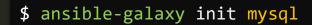








Share





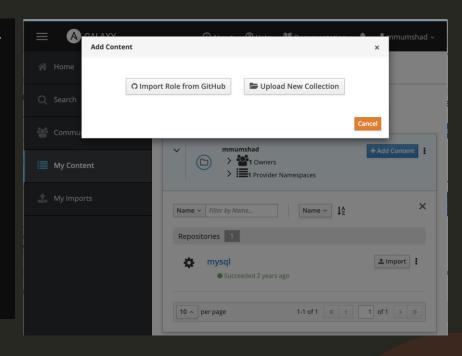
- playbook.yml
- = roles
  - mysql
    - README.md
    - **t**emplates
    - **t**asks
    - handlers
    - ---- **v**ars
    - defaults
    - meta

## playbook.yml

name: Install and Configure MySQL hosts: db-server

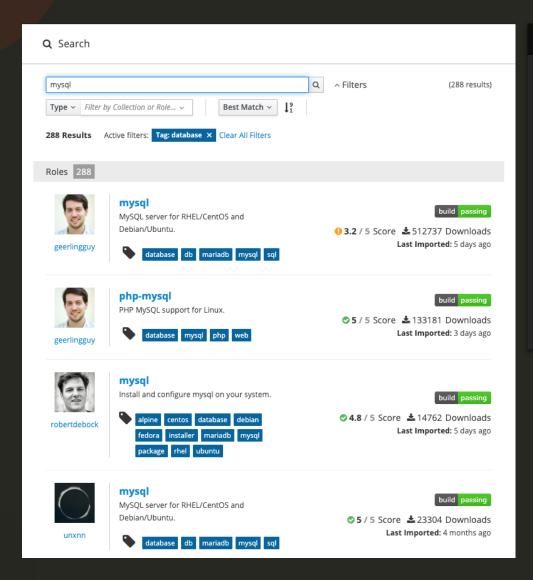
roles:

- mysql





# Find Roles



#### \$ ansible-galaxy search mysql

Found 1126 roles matching your search. Showing first 1000.

#### Name Description

Outsider.ansible zabbix agent 1mr.unattended 1nfinitum.mysql 4linuxdevops.mysql-server 5KYDEV0P5.skydevops-mysql AAbouZaid.yourls AAROC.AAROC fg-db aaronpederson.ansible-autodeploy abednarik.mysgld-exporter abelboldu.openstack-glance abelboldu.openstack-keystone abelboldu.openstack-neutron-controller abelboldu.openstack-nova-controller achaussier.mysql-backup achaussier.mysgl-server achilleskal.ansible mysql8 adarnimrod.mysql

Installing and maintaining zabbix-agent for install and configure unattended upgrade Simply installs MySQL 5.7 on Xenial.
Instalacao e Configuracao do servidor MySQL Install and configure MySQL Database Manage Yourls, a URL shortener web app. your description

Simple deployment tool with hooks Install and configure mysqld\_exporter

er OpenStack Neutron controller node
OpenStack Nova controller node
configure mysql-backup with xtrabackup and
Install mysql-server package
your description
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/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



## Playbook-distributed.yml

name: Install and Configure MySQL

hosts: db-server

roles:

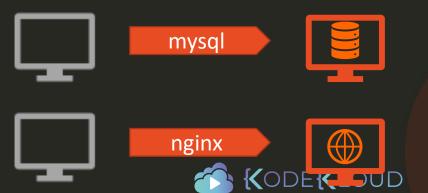
- geerlingguy.mysql

name: Install and Configure Web Server

hosts: web-server

roles:

- nginx



# 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







# **Getting Started**





www.kodekloud.com

