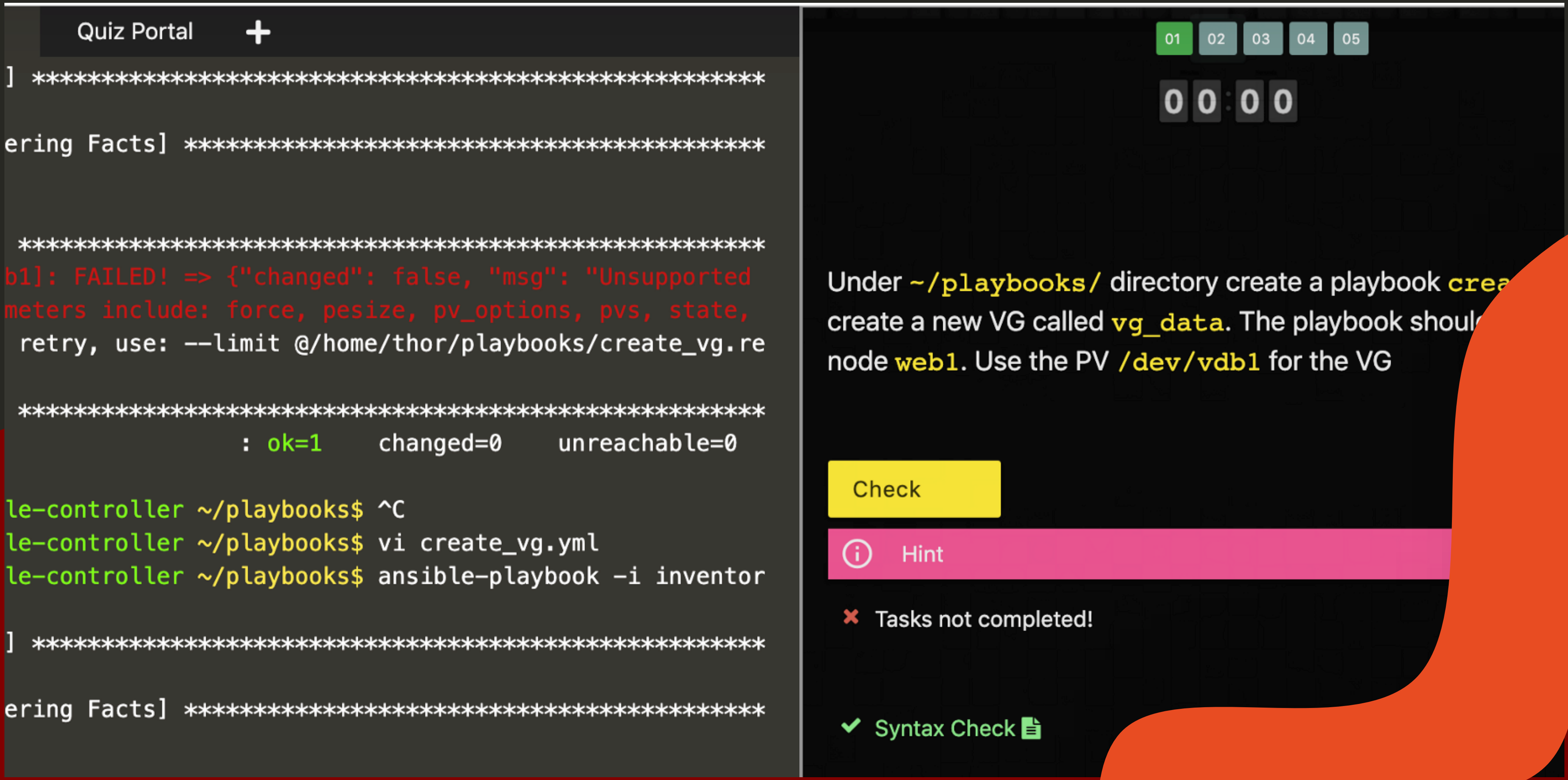
Ansible For the Absolute Beginner

Hands-On Exercises



```
] *****************************************************

ering Facts] *******************************************


 ******************************************************
b1]: FAILED! => {"changed": false, "msg": "Unsupported
meters include: force, pesize, pv_options, pvs, state,
 retry, use: --limit @/home/thor/playbooks/create_vg.re

 ******************************************************
                 : ok=1    changed=0    unreachable=0

le-controller ~/playbooks$ ^C
le-controller ~/playbooks$ vi create_vg.yml
le-controller ~/playbooks$ ansible-playbook -i inventor

] *****************************************************

ering Facts] *******************************************
```

Under ~/playbooks/ directory create a playbook crea
create a new VG called vg_data. The playbook shoul
node web1. Use the PV /dev/vdb1 for the VG

✗ Tasks not completed!

✓ Syntax Check

Note

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

The Curriculum

RedHat Ansible for Beginners

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

Ansible

Introduction

KODEKLOUD

Why Ansible?

Provisioning

Configuration Management

Continuous Delivery

Application Deployment

Security Compliance

Scripts

- Time
- Coding Skills
- Maintenance

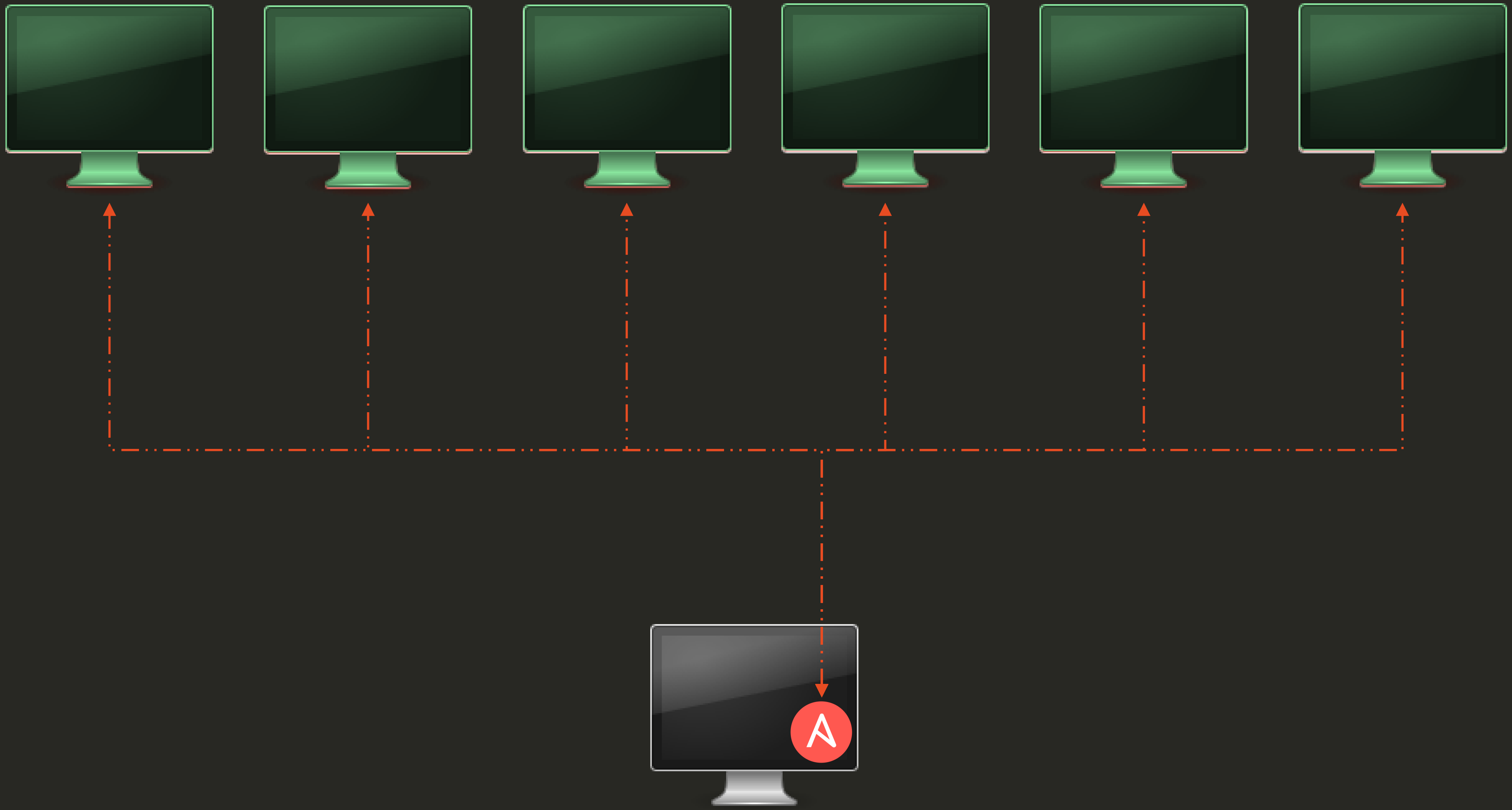
- Simple
- Powerful
- Agentless

Scripts vs Ansible Playbook

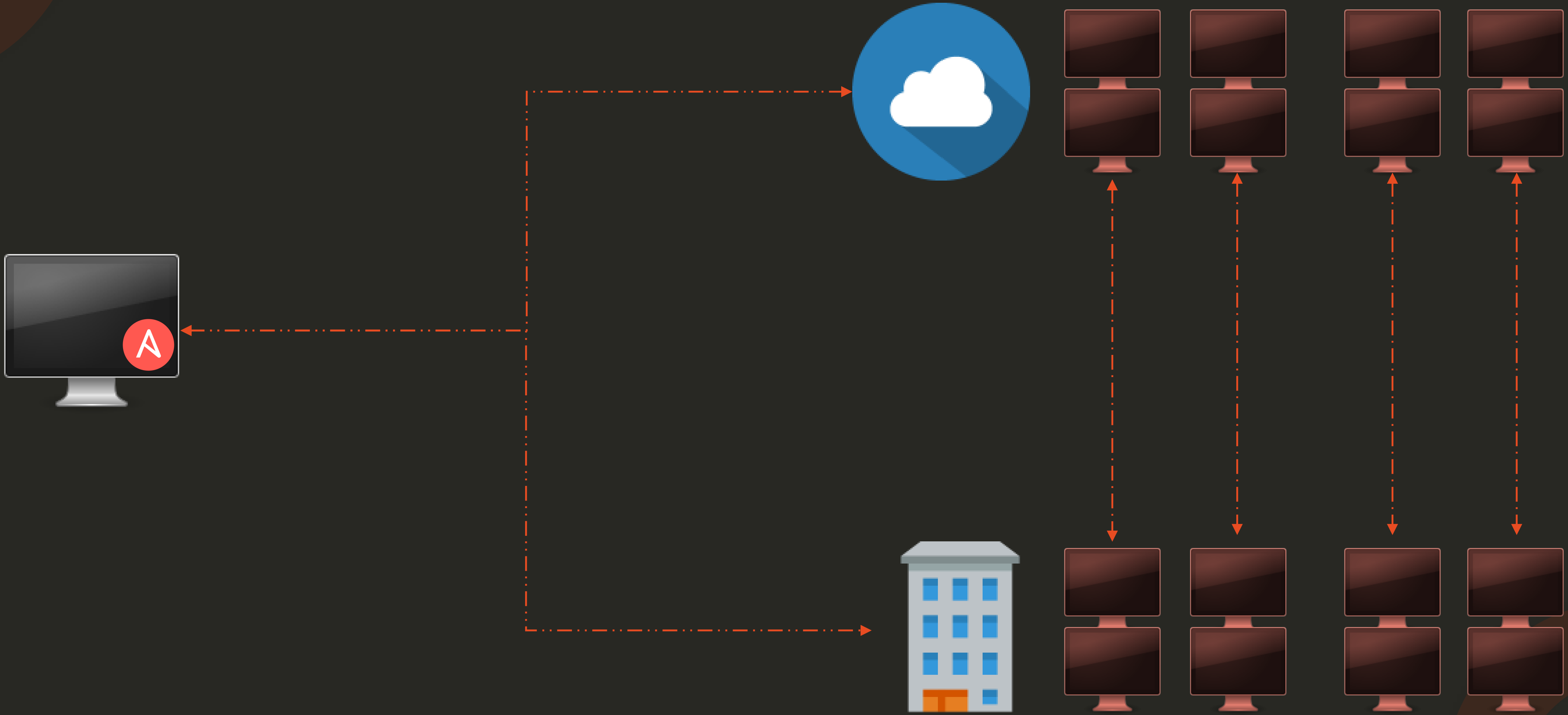
```bash
#!/bin/bash
# Script to add a user to Linux system
if [ $(id -u) -eq 0 ]; then
   $username=johndoe
   read -s -p "Enter password : " password
   egrep "^$username" /etc/passwd >/dev/null
   if [ $? -eq 0 ]; then
      echo "$username exists!"
      exit 1
   else
      useradd -m -p $password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

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

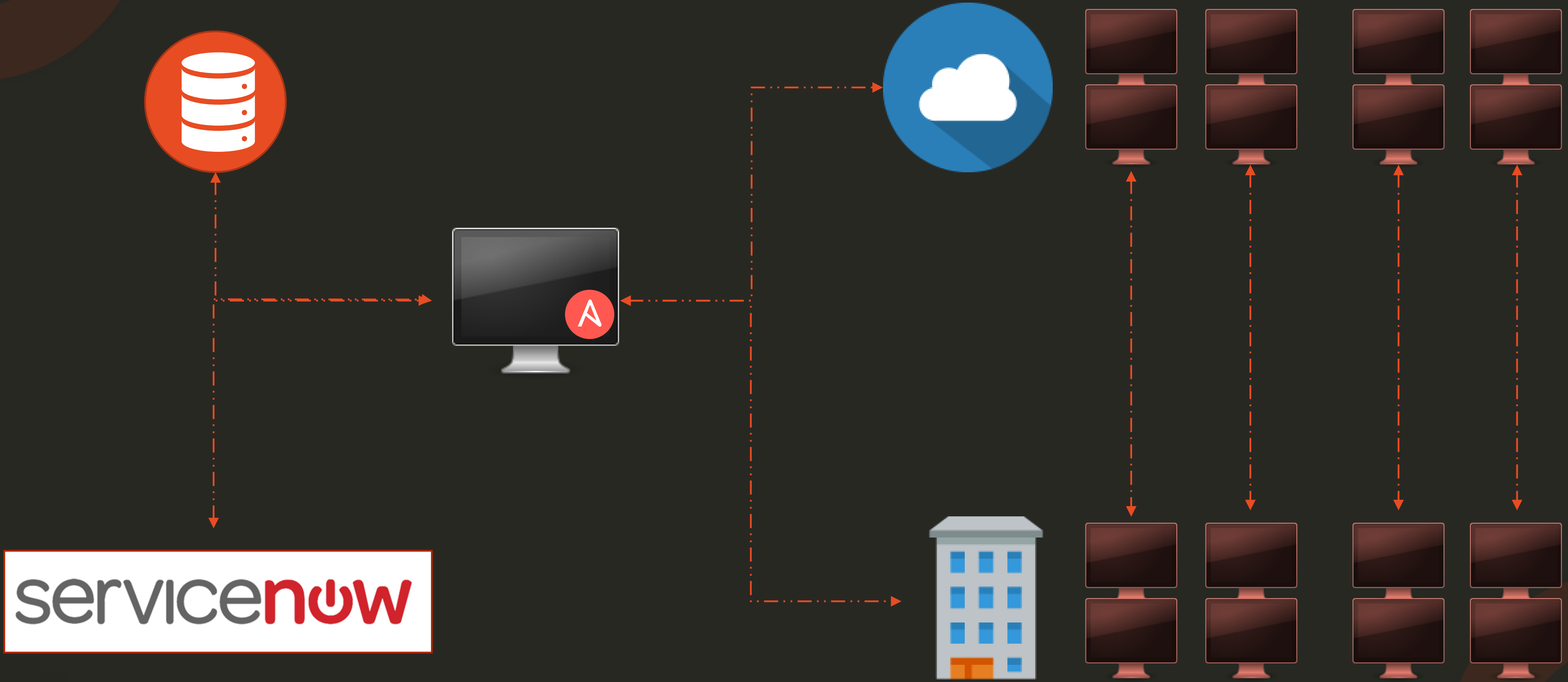
Use case example - Simple



Use case example - complex



Use case example - complex



Ansible Documentation

The Curriculum

RedHat Ansible for Beginners

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

Ansible Control Machine

- Playbooks
- Inventory
- Modules

Additional Options:

- Install from source on GIT
- Build RPM yourself

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
```

The Curriculum

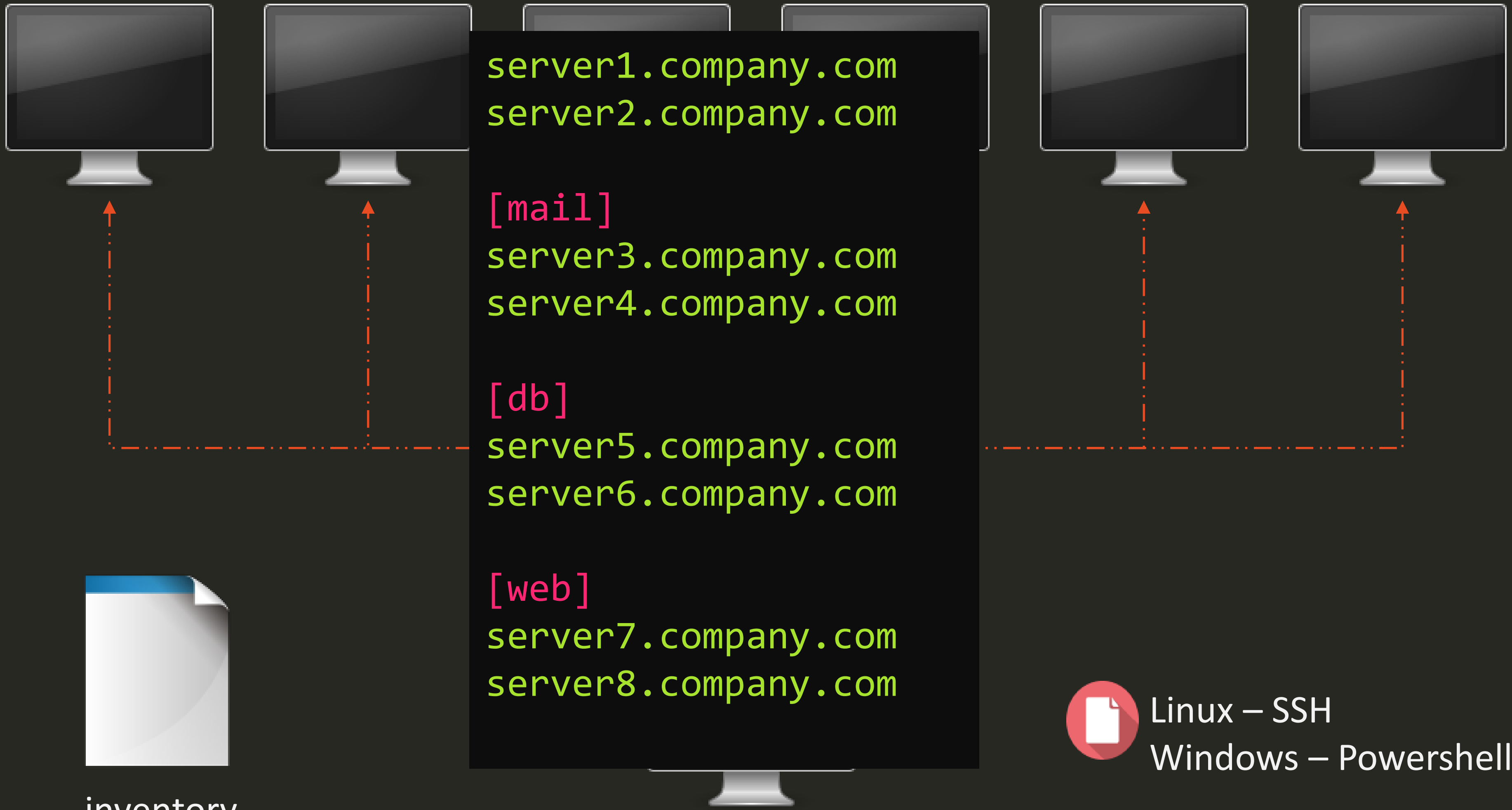
- Introduction to Ansible
- Setting up Ansible on VirtualBox
- Introduction to YAML
- Inventory Files
- Playbooks
- Variables
- Conditionals
- Loops
- Roles

Ansible

Inventory

KODEKLOUD

inventory



```
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
```

inventory
/etc/ansible/hosts

Linux – SSH
Windows – Powershell Remoting

Agentless

More on inventory files

```
#Sample Inventory File

server1.company.com                   ansible_connection=ssh    ansible_user=root
server2.company.com                   ansible_connection=winrm  ansible_user=admin
server3.company.com                   ansible_connection=ssh    ansible_ssh_pass=P@#
server4.company.com                   ansible_connection=winrm

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

The Curriculum

RedHat Ansible for Beginners

- Introduction to Ansible
- 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

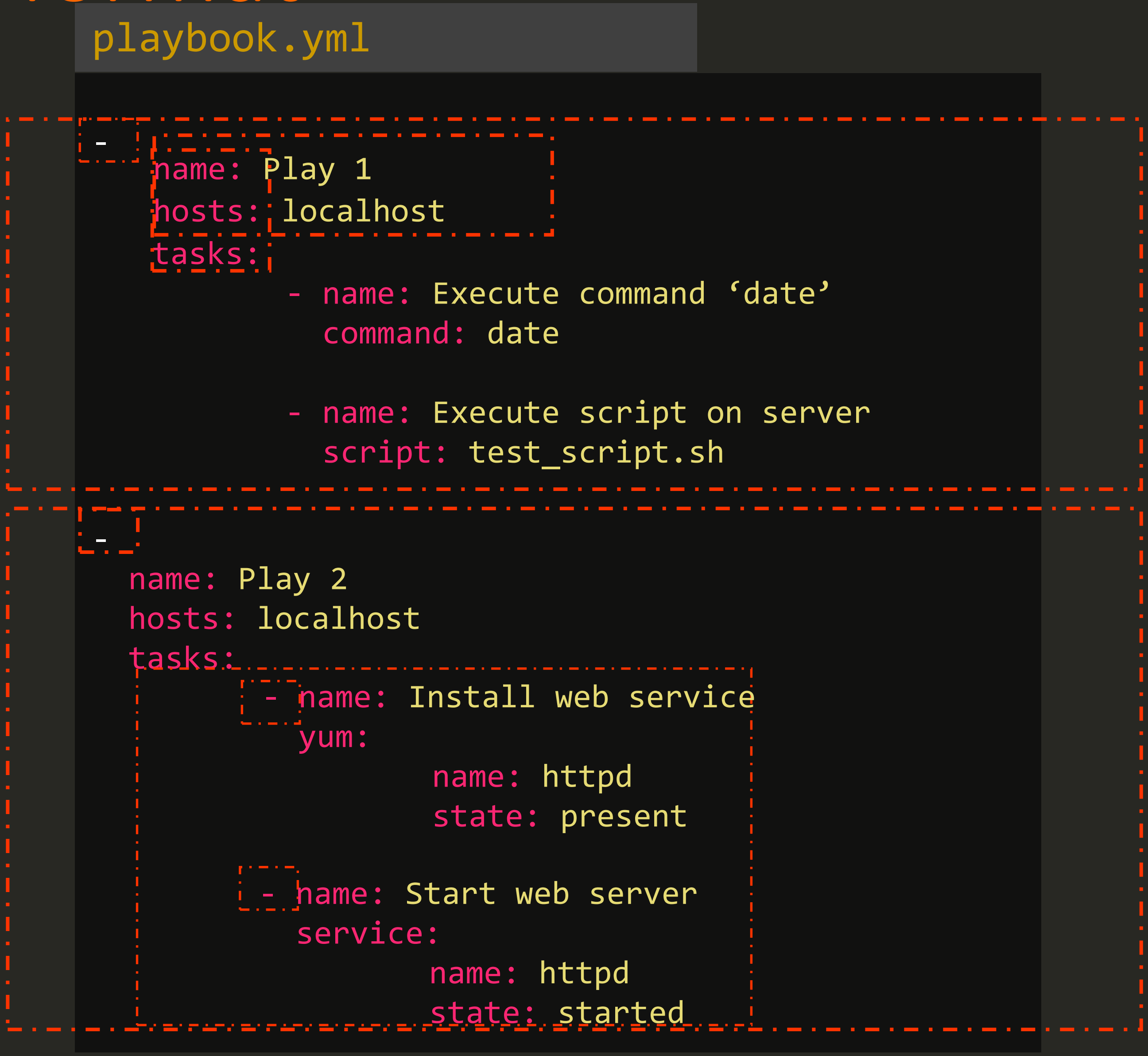
- Playbook – A single YAML file
  - Play – Defines a set of activities (tasks) to be run on hosts
    - Task – An action to be performed on the host
      - ➢ Execute a command
      - ➢ Run a script
      - ➢ Install a package
      - ➢ Shutdown/Restart

YAML 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: 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

      - name: 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
```

module

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
```

```
ansible-doc -l
```

Run

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

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

```
ansible-playbook --help
```

The Curriculum

RedHat Ansible for Beginners

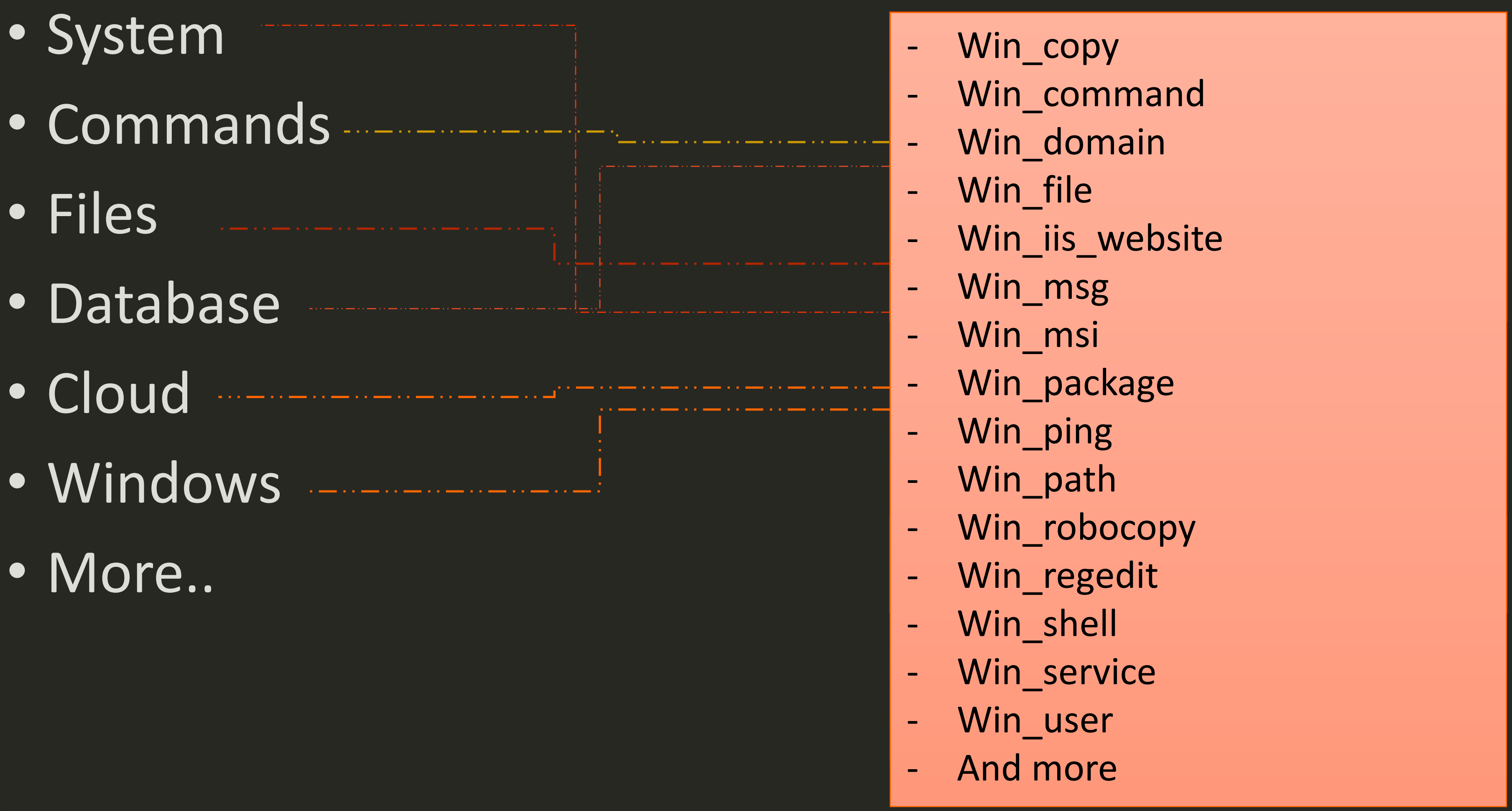
- Introduction to Ansible
- Setting up Ansible on VirtualBox
- Introduction to YAML
- Inventory Files
- Playbooks
- Modules
- Variables
- Conditionals
- Loops
- Roles

https://www.kodekloud.com

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	comments
chdir	cd into this directory before running the command
creates	a filename or (since 2.0) glob pattern, when it already exists, this step will **not** 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 **not** 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.

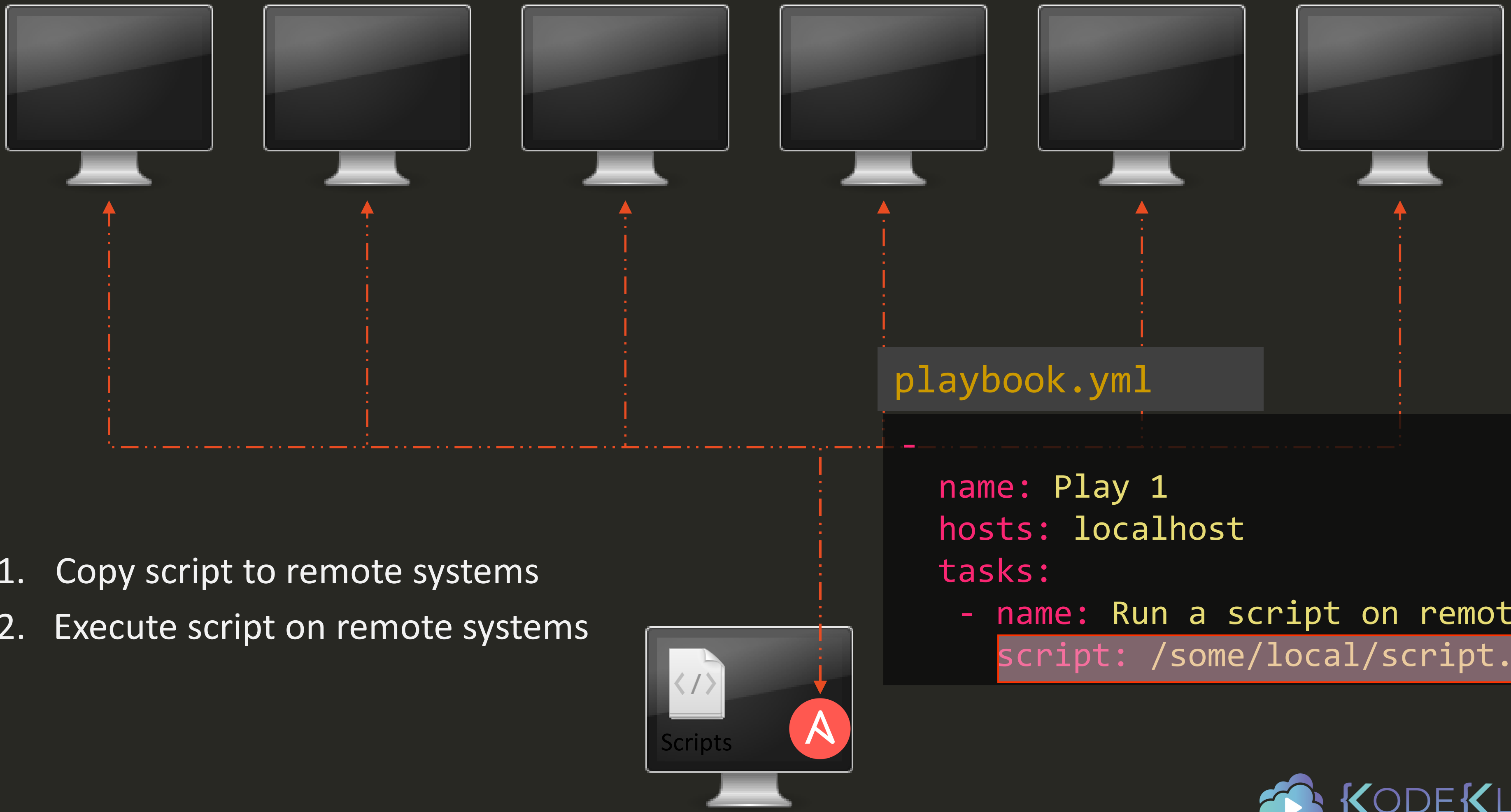
playbook.yml

```
-
  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



1. Copy script to remote systems
2. Execute script on remote systems

playbook.yml

```
-
  name: Play 1
  hosts: localhost
  tasks:
   - name: Run a script on remote server
     script: /some/local/script.sh -arg1 -arg2
```

Service

- Manage Services – Start, Stop, Restart

playbook.yml

```yaml
-
  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

```yaml
-
  name: Start Services in order
  hosts: localhost
  tasks:
   - name: Start the database service
     service:
      name: postgresql
      state: started
```

idempotency

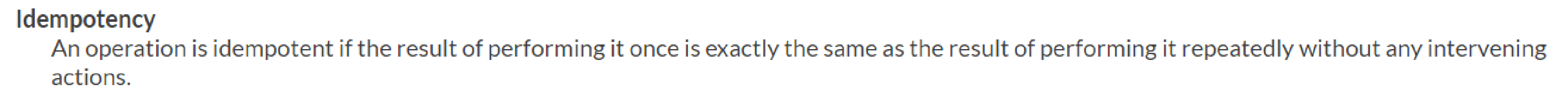
Why “started” and not “start”?

“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.

/etc/resolv.conf

```
nameserver 10.1.250.1
nameserver 10.1.250.2
```

playbook.yml

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

/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
nameserver 10.1.250.10
```

The Curriculum

RedHat Ansible for Beginners

- Introduction to Ansible
- 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@s
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
  tasks:
      - lineinfile:
          path: /etc/resolv.conf
          line: 'nameserver 10.1.250.10'
```

variables

```
variable1: value1
variable2: value2
```

Using variables

Playbook.yml

```yaml
-
  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: '{{ 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=     snmp_port=        inter_ip_range=
```

```
#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

The Curriculum

RedHat Ansible for Beginners

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

Ansible

Loops

LOOPS

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

LOOPS - Visualize

```
-
  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
```

LOOPS - Visualize

```
-
  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
  -  var:   item=george
     user: name= “{{ item }}”     state=present
  -  var:   item=ravi
     user: name= “{{ item }}”     state=present
  -  var:   item=mani
     user: name= “{{ item }}”     state=present
  -  var:   item=kiran
     user: name= “{{ item }}”     state=present
  -  var:   item=jazlan
     user: name= “{{ item }}”     state=present
  -  var:   item=emaan
     user: name= “{{ item }}”     state=present
  -  var:   item=mazin
     user: name= “{{ item }}”     state=present
  -  var:   item=izaan
     user: name= “{{ item }}”     state=present
```

LOOPS - Visualize

```
-
  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

  -  var:  item=george
     user: name= “{{ item }}”     state=present

  -  var:  item=ravi
     user: name= “{{ item }}”     state=present

  -  var:  item=mani
     user: name= “{{ item }}”     state=present

  -  var:  item=kiran
     user: name= “{{ item }}”     state=present

  -  var:  item=jazlan
     user: name= “{{ item }}”     state=present

  -  var:  item=emaan
     user: name= “{{ item }}”     state=present
  -  var:  item=mazin
     user: name= “{{ item }}”     state=present
  -  var:  item=izaan
     user: name= “{{ item }}”     state=present
```

LOOPS - Visualize

```
-
  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=“{?}”
```

LOOPS - Visualize

```
-
  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
```

LOOPS - Visualize

```
-
  name: Create users
  hosts: localhost
  tasks:
   - user: name= ‘{{ item.name }}’ state=present  uid=‘{{ item.uid }}’
     loop:
       - name: joe      -   { name: joe, uid: 1010 }
         uid: 1010
       - name: george   -   { name: george, uid: 1011 }
         uid: 1011
       - name: ravi     -   { name: ravi, uid: 1012 }
         uid: 1012
       - name: mani     -   { name: mani, uid: 1013 }
         uid: 1013
       - name: kiran    -   { name: kiran, uid: 1014 }
         uid: 1014
       - name: jazlan   -   { name: jazlan, uid: 1015 }
         uid: 1015
       - name: emaan    -   { name: emaan, uid: 1016 }
         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: View Config Files
  hosts: localhost
  tasks:
   - debug: var=item
     with_file:
      - "/etc/hosts"
      - "/etc/resolv.conf"
      - "/etc/ntp.conf"
```

```
-
  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: 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
```

The Curriculum

RedHat Ansible for Beginners

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

Ansible

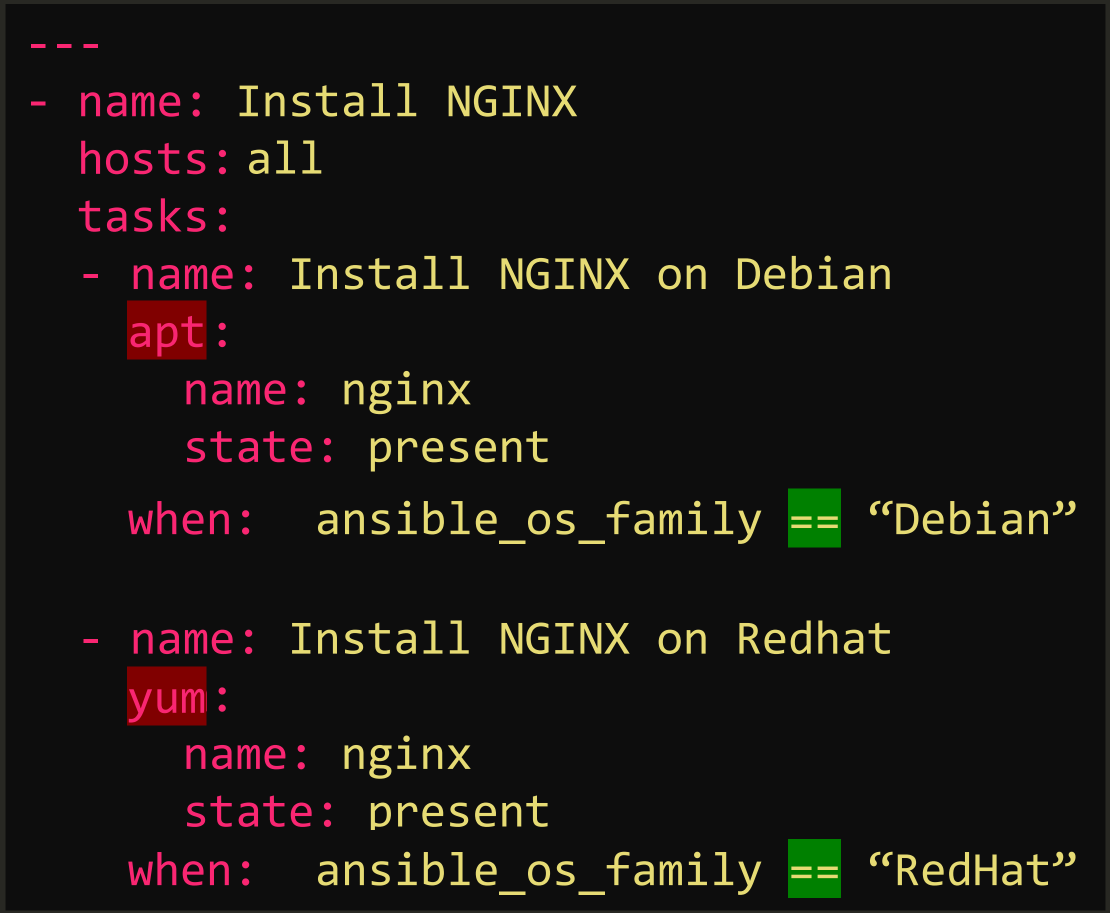
Conditionals

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

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

NOTE: PARTS OF CODE HIDDEN FOR BREVITY

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

```yaml
---
- 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
  when:   item.required == True
```

Conditionals in Loops

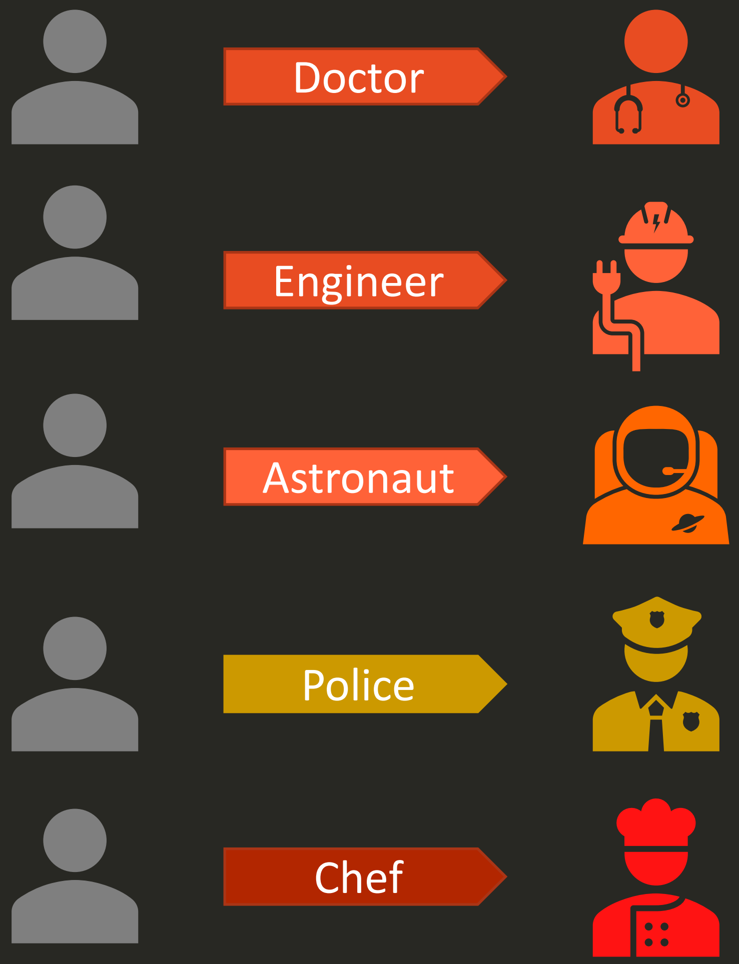
```
---
- name: Install Softwares
  hosts: all
  vars:
     packages:
        - name: nginx
          required: True
        - name: mysql
          required : True
        - name: apache
          required : False
  tasks:
  - name: Install  “{{ item.name }}” on Debian
    apt:
      name: “{{ item.name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

Roles



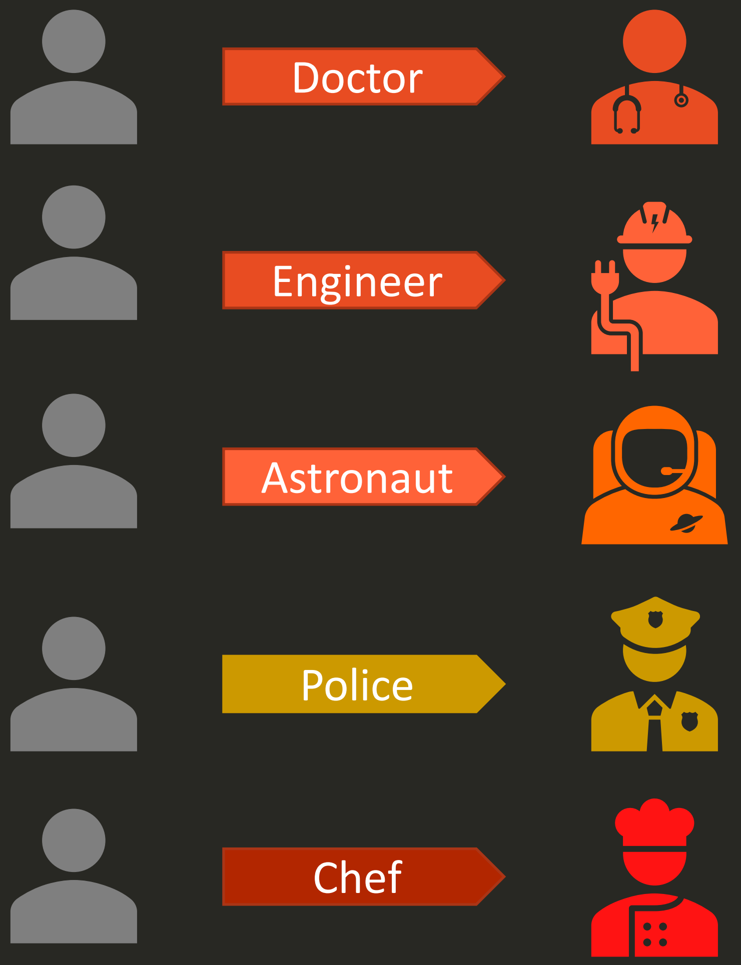
Doctor

Engineer

Astronaut

Police

Chef



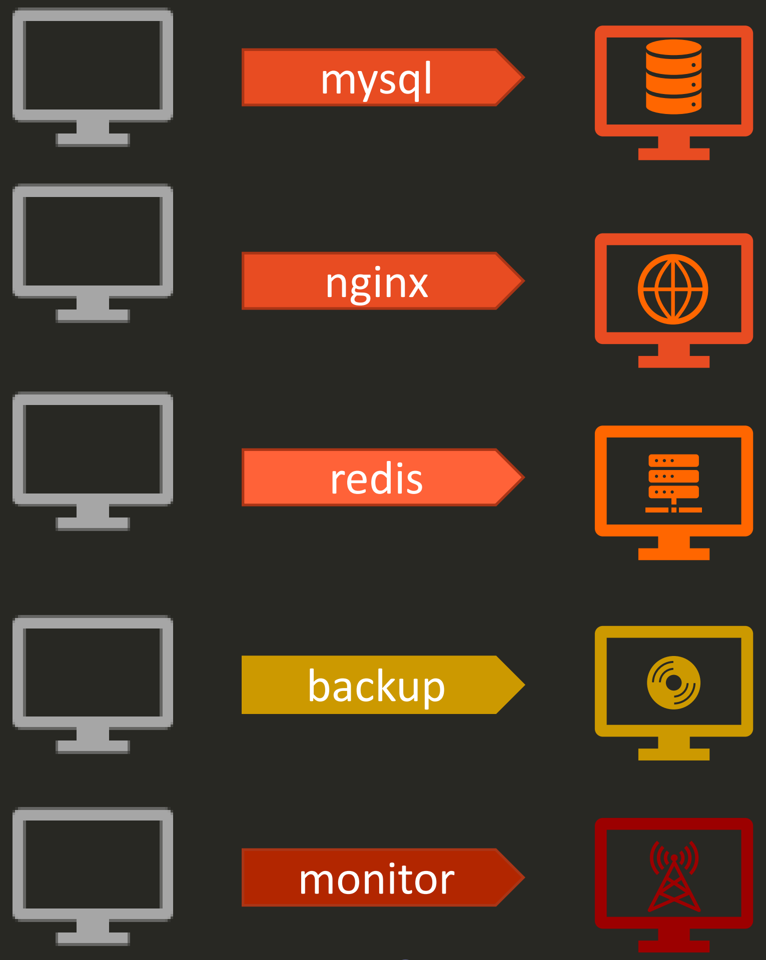
Doctor

Engineer

Astronaut

Police

Chef



mysql

nginx

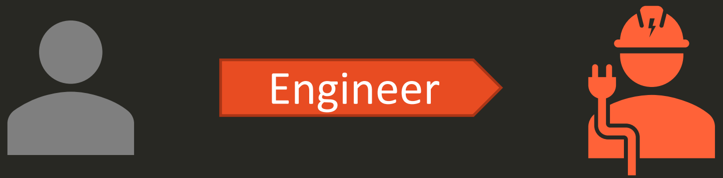
redis

backup

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

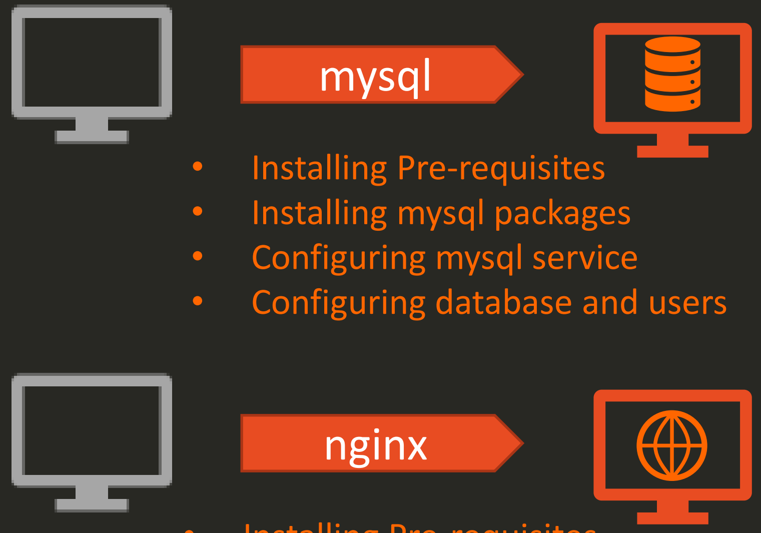
mysql

- 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

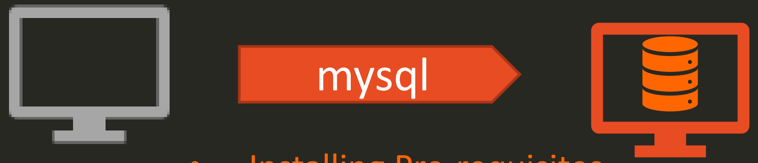
```
- name: Install and Configure MySQL
  hosts: db-server
  tasks:
     - name: Install Pre-Requisites
       yum: name=pre-req-packages state=present

     - 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
```



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

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

Re-Use



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

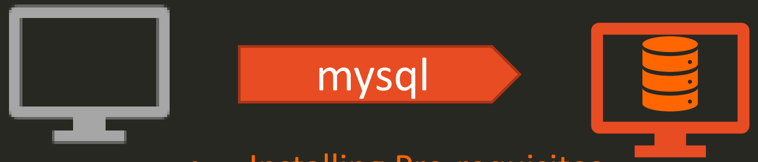
```
- name: Install and Configure MySQL
  hosts: db-server1……db-server100
  roles:
     - mysql
```

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



- 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

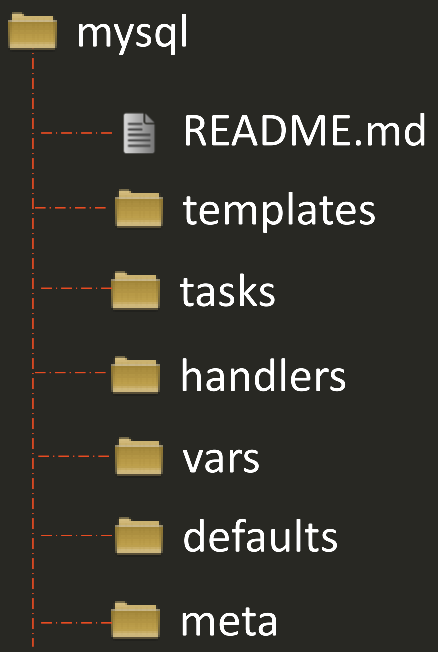
templates

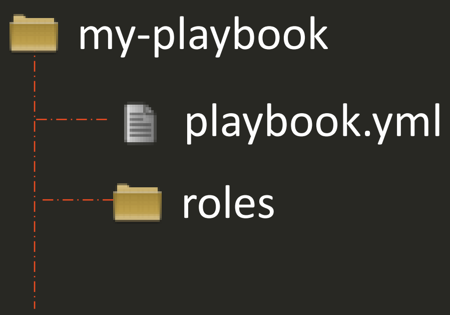
Organize

Re-Use

Share

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





playbook.yml

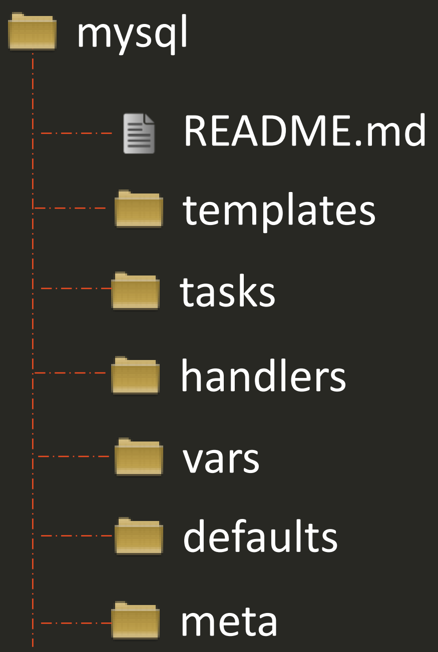
```
- name: Install and Configure MySQL
  hosts: db-server
 roles:
      - mysql
```

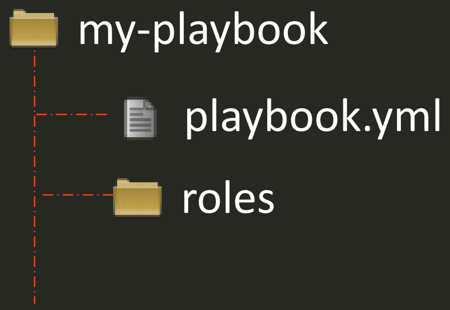
Organize

Re-Use

Share

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





playbook.yml

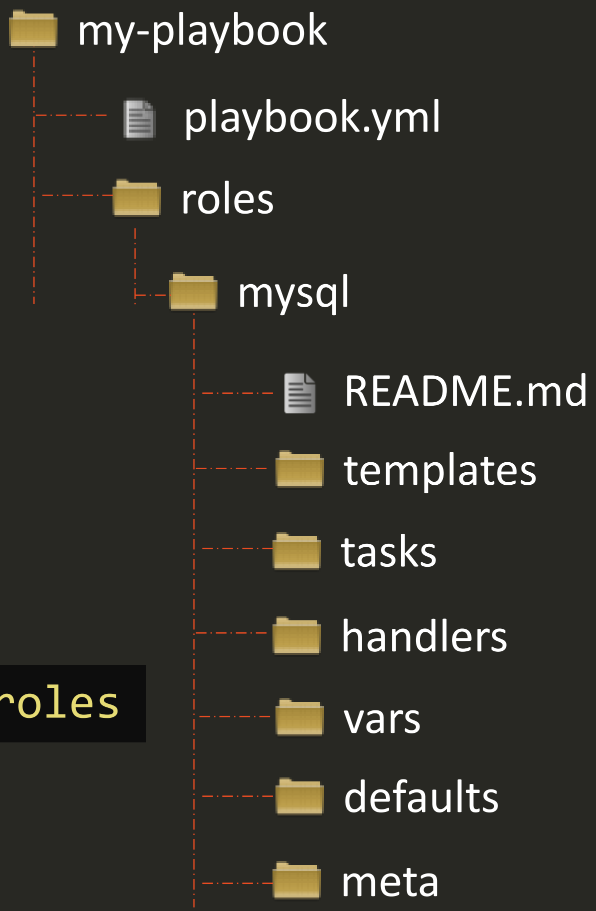
```
- name: Install and Configure MySQL
  hosts: db-server
 roles:
      - mysql
```

Organize

Re-Use

Share

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



```
my-playbook
├── playbook.yml
└── roles
    └── mysql
        ├── README.md
        ├── templates
        ├── tasks
        ├── handlers
        ├── vars
        ├── defaults
        └── meta
```

playbook.yml

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

/etc/ansible/ansible.cfg

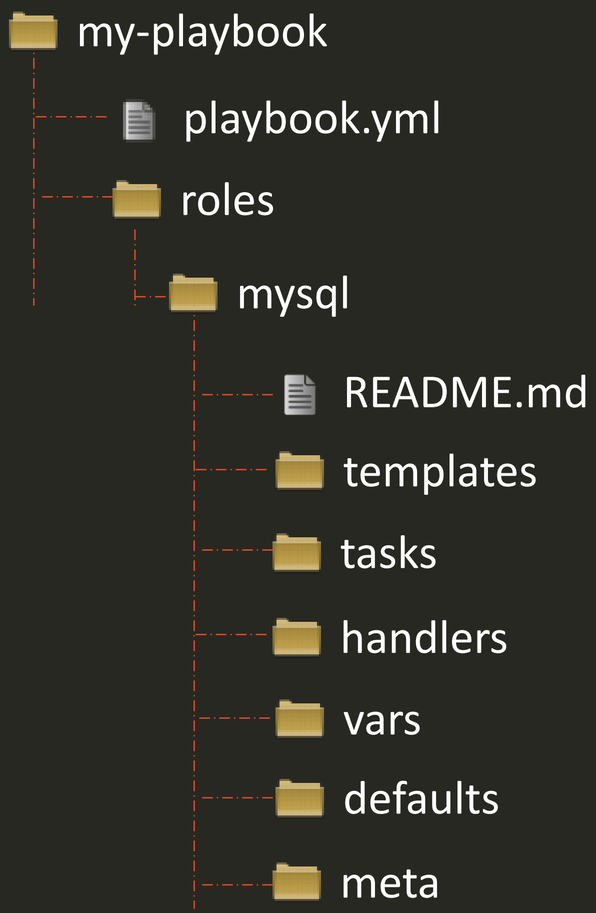
```
roles_path  = /etc/ansible/roles
```

Organize

Re-Use

Share

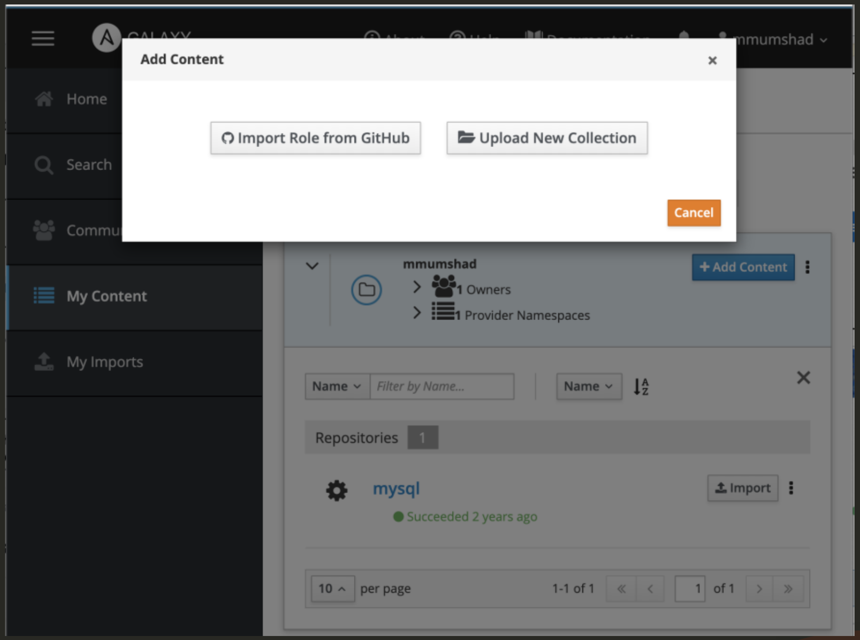
```
$ ansible-galaxy init mysql
```



- my-playbook
  - playbook.yml
  - roles
    - mysql
      - README.md
      - templates
      - tasks
      - handlers
      - vars
      - defaults
      - meta

playbook.yml

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



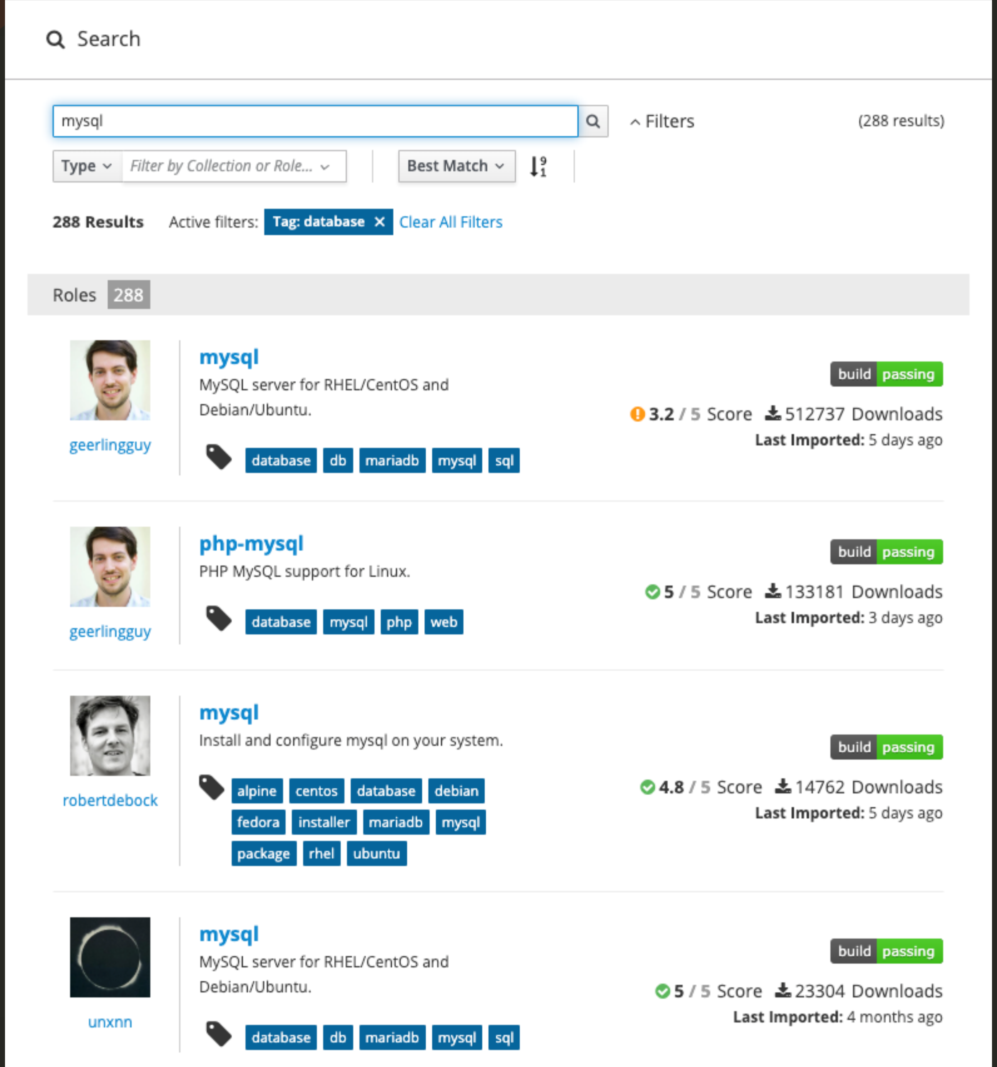
mmumshad
1 Owners
1 Provider Namespaces

Repositories 1

mysql
Succeeded 2 years ago

1-1 of 1 1 of 1

Find Roles



Search

mysql

Filters (288 results)

Type | Filter by Collection or Role... | Best Match

288 Results Active filters: **Tag: database** × Clear All Filters

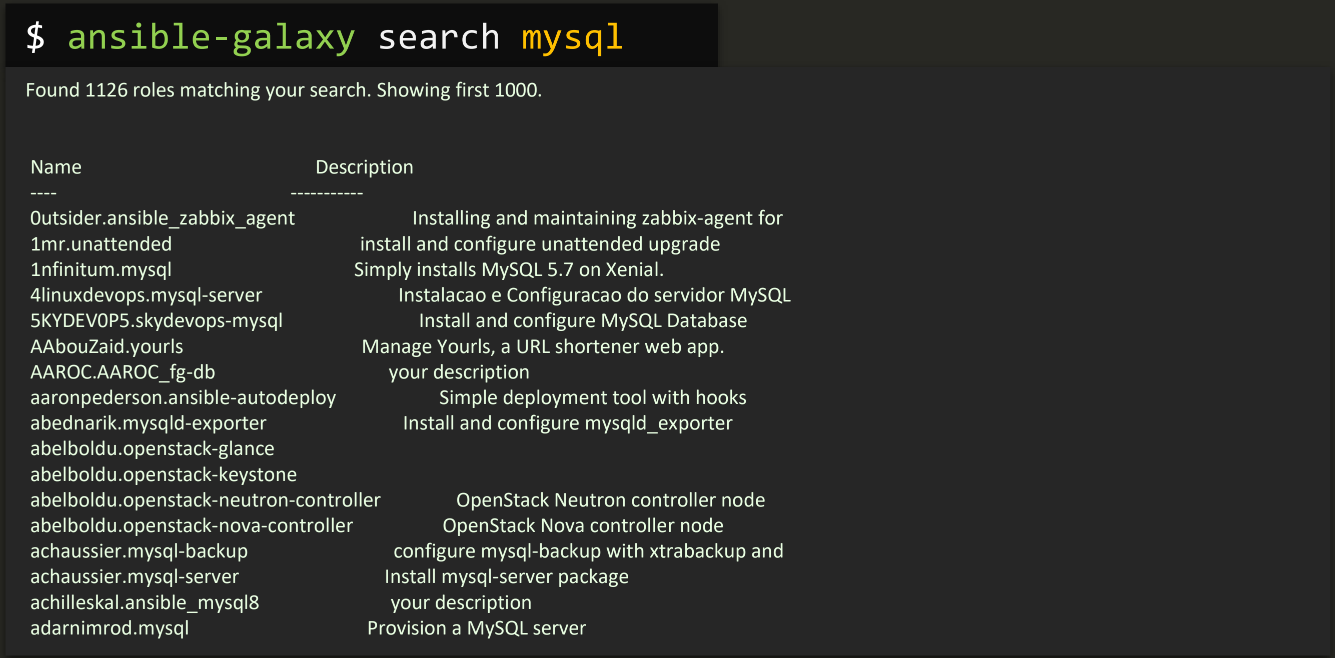
Roles 288

mysql — geerlingguy
MySQL server for RHEL/CentOS and Debian/Ubuntu.
build passing
3.2 / 5 Score 512737 Downloads
Last Imported: 5 days ago
database db mariadb mysql sql

php-mysql — geerlingguy
PHP MySQL support for Linux.
build passing
5 / 5 Score 133181 Downloads
Last Imported: 3 days ago
database mysql php web

mysql — robertdebock
Install and configure mysql on your system.
build passing
4.8 / 5 Score 14762 Downloads
Last Imported: 5 days ago
alpine centos database debian fedora installer mariadb mysql package rhel ubuntu

mysql — unxnn
MySQL server for RHEL/CentOS and Debian/Ubuntu.
build passing
5 / 5 Score 23304 Downloads
Last Imported: 4 months ago
database db mariadb mysql sql



```
$ 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
 1nfinitum.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.mysqld-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/etc/ansible/roles/geerlingguy.mysql
- geerlingguy.mysql (2.9.5) was installed successfully
```

playbook.yml

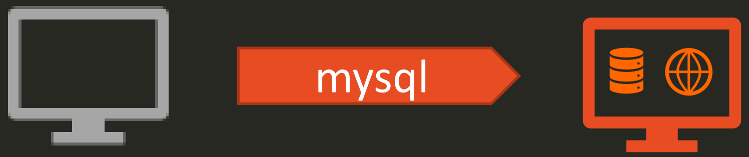
```yaml
-
  name: Install and Configure MySQL
  hosts: db-server
  roles:
    - geerlingguy.mysql
```

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

Use Role

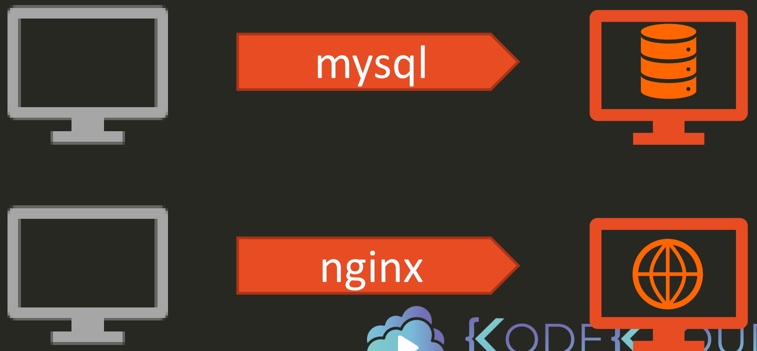
Playbook-all-in-one.yml

```
-
  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
```