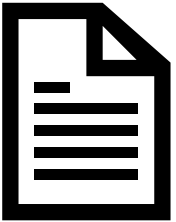


Linux RedHat Certified Engineer (RHCE - EX294)

Ansible Playbook Implementation

Creating First Playbook

- There are online yaml tools you can use to create Playbooks
 - <https://onlineyamltools.com/edit-yaml>
 - <https://codebeautify.org/yaml-editor-online>
- Tools to download
 - Notepad++ Windows
- Take a snapshot after installing Ansible software



```
# su - root
# mkdir /etc/ansible/playbooks
# cd /etc/ansible/playbooks
# vim first.yml
```

```
---
- name: "My first playbook"
  hosts: localhost

  tasks:
  - name: "test connectivity"
    ping:
```

- Check syntax of playbook

```
# ansible-playbook --syntax-check first.yml
```

Or to do a dry run

```
# ansible-playbook --check first.yml
```

- Run the playbook

```
# ansible-playbook /root/ansible/first.yml
```

Creating First Playbook

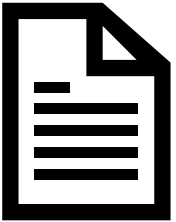
Please note:

- Running ansible without a playbook
ansible
- Running ansible with a playbook
ansible-playbook



Output Playbook

This playbook will print “Hello World” on localhost



```
# cd /etc/ansible/playbook
# vim helloworld.yml
```

```
---
```

```
- name: My Second playbook
  hosts: localhost
```

```
  tasks:
```

```
  - name: Print Hello World
    debug: msg="Hello World"
```

→ Name of the play or playbook

→ Run on localhost

→ Run the following task

→ Name of the task

→ Run **debug** module which prints statements during execution

Run the playbook

```
# ansible-playbook helloworld.yml
```

Multiple Tasks Playbook

The playbook will ping localhost and print “Hello World”

```
# vim mtask.yml
```

```
---
```

```
- name: Running 2 tasks
```

```
  hosts: localhost
```

```
  tasks:
```

```
    - name: Test connectivity
```

```
      ping:
```

```
    - name: Print Hello World
```

```
      debug: msg="Hello World"
```

→ Name of the play

→ Run it on local host

→ Run the following task

→ Name of the task

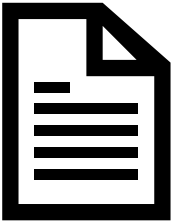
→ Run the **ping** module

→ Name of the 2nd task

→ Run the **debug** module

Run the playbook

```
# ansible-playbook mtask.yml
```



Installing and Starting a Package



```
# vim packinstall.yml

---
- name: Installing and Running apache
  hosts: localhost

  tasks:
    - name: Install apache
      yum:
        name: httpd
        state: present

    - name: start httpd
      service:
        name: httpd
        state: started
```

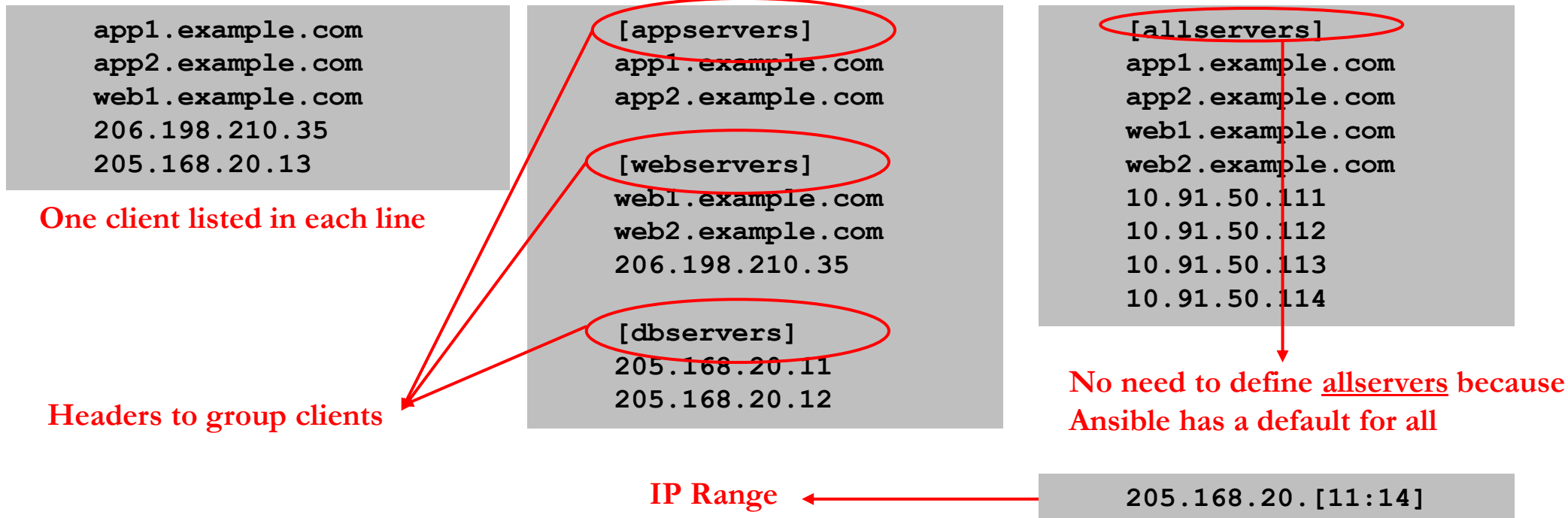
Run the playbook

```
# ansible-playbook packinstall.yml
```

Remote Clients hosts File Syntax

`/etc/ansible/hosts`

- All remote clients are considered inventory in Ansible
- Ansible keeps its inventory information in host file located: `/etc/ansible/hosts`
- The **hosts** file is created during Ansible installation



- You can specify different location of the file
`ansible-playbook -i /home/iafzal/ansible/hosts`
<https://t.me/learningnets>

Remote Clients hosts File Syntax

/etc/ansible/hosts

```
[servers]
server1 ansible_ssh_host=10.91.50.110
server2 ansible_ssh_host=10.91.50.111
server3 ansible_ssh_host=10.91.50.112
server4 ansible_ssh_host=10.91.50.113
server5 ansible_ssh_host=10.91.50.114
server6 ansible_ssh_host=10.91.50.115
```

```
[appserver]
server1
server2
```

```
[webserver]
server3
server4
```

```
[dbservers]
server5
server6
```

Aliases



Remote Clients hosts File Syntax

`/etc/ansible/hosts`

- Inventory host file can either be static or dynamic (*using additional plug-ins*)

- Listing host file

```
# ansible-inventory --list
```

```
OR
```

```
# ansible all --list-hosts
```



Establish Connection to Remote Clients

- Take a snapshot of our Linux client1 and then power it up
- Note down its IP address
- Populate the hosts file with IP or FQDN for our clients:

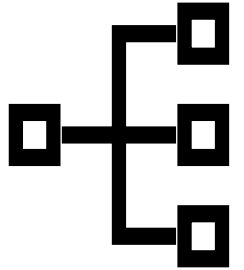
```
[labclients] = For grouping
10.253.1.18
10.253.1.20
```
- Generate SSH Keys on the control node and copy over to clients for password less SSH connections

```
# ssh-keygen
# Leave everything default and enter
# ssh-copy-id 10.253.1.18
# ssh-copy-id 10.253.1.20
```
- Now SSH into the clients to test

```
# ssh 10.253.1.18
```
- Run **Ansible** add-hoc to ping remote nodes (*make sure hosts file has remote clients IPs*)

```
# ansible all -m ping
# ansible -a "uptime" all (To run a command on the remote clients)
```

<https://t.me/learningnets>



Check Remote Clients Connectivity

```
# su - root
# cd /etc/ansible/playbooks
# vim clientstatus.yml

---
- name: "Check remote clients connectivity status"
  hosts: all

  tasks:
    - name: Test connectivity
      ping:
```

Run the playbook

```
# ansible-playbook clientstatus.yml
```

