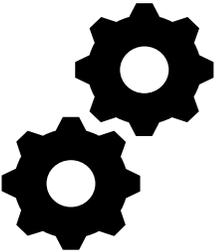


# Linux RedHat Certified Engineer (RHCE - EX294)

**Manage Complex Plays  
and Playbooks**

# Setup Apache and Open Firewall Port

- The playbook will
  1. Install httpd package
  2. Start httpd service
  3. Open http service port in firewall
  4. Restart firewalld service



```
# Login to LinuxCleint1
# rpm -qa | grep http
# systemctl status firewalld

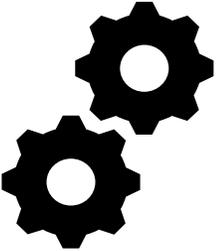
# Login to ControlNode

• Install additional Ansible collection for firewalld
# ansible-galaxy collection install ansible.posix

# cd /home/ansible/playbooks
# vim httpsetup.yml
```

- Ansible modules and options  
<https://docs.ansible.com/ansible/2.5/modules/>

# Setup Apache and Open Firewall Port



```
---
- name: Setup httpd and open firewall port
  hosts: all
  tasks:
  - name: Install apache packages
    yum:
      name: httpd
      state: present
  - name: Start httpd
    service:
      name: httpd
      state: started
  - name: Open port 80 for http access
    firewalld:
      service: http
      permanent: true
      state: enabled
  - name: Restart firewalld service to load firewall changes
    service:
      name: firewalld
      state: reloaded
```

**State** = What to do with the package?

- present or installed = Install
- absent or removed = Un-install
- latest = Upgrade

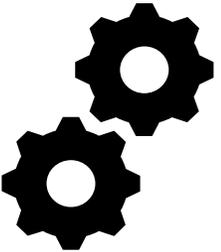
**State** = What to do with the service?

- started | stopped | reloaded | restarted

save httpsetup.yml

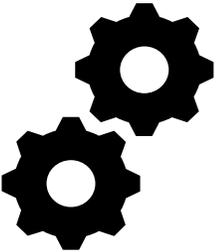
# Setup Apache and Open Firewall Port

- Login back to **LinuxCleint1**
- Check httpd package status  
# **rpm -qa | grep http**
- Check httpd package service status  
# **systemctl status httpd**
- Check firewalld service status  
# **systemctl status firewalld**
- Check if http service is enabled in firewalld  
# **firewall-cmd --list-all**
- Open FireFox and go to 10.253.1.115



# Run Shell Scripts on Remote Clients

- The playbook will run shell script on the remote client (LinuxClient1)
- Create `/home/iafzal/cfile.sh` script on `LinuxClient1`
- The `cfile.sh` script should create a new file `example1`



```
# vim shellscript.yml
```

```
---
```

```
- name: Playbook for shell script
```

```
  hosts: all or 10.253.1.115
```

```
  tasks:
```

```
  - name: Run shell script
```

```
    shell: "/home/iafzal/cfile.sh"
```

→ Description of the playbook

→ Run on client1

→ Run the following task

→ Name/description of the task

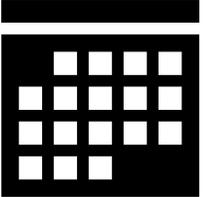
→ Run shell module which will execute shell script on LinuxClient1

Run the playbook

```
# ansible-playbook shellscript.yml
```

# Schedule a job (crontab)

- The playbook `cronjob.yml` will
  - Schedule a job as a `root`
  - Every thursday at 10am
  - Define job (`/home/iafzal/cfile.sh`) to be executed by `root`



```
# vim cronjob.yml

---
- name: Create a cron job
  hosts: all

  tasks:
    - name: Schedule cron:
      cron:
        name: This job is scheduled by Ansible
        minute: "0"
        hour: "10"
        day: "*"
        month: "*"
        weekday: "4"
        user: root
        job: "/home/iafzal/cfile.sh"
```

→ Day of the month

→ Day of the week

Run the playbook

```
# ansible-playbook cronjob.yml
```

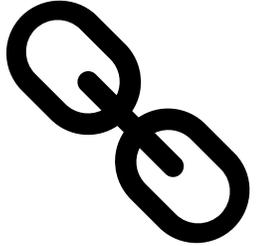
# Download Package from a URL

- The playbook tomcat.yml will
  - Create a directory for tomcat with required permissions
  - Download tomcat from a url and place it in that directory with modified permissions

```
vim tomcat.yml
---
- name: Download Tomcat from tomcat.apache.org
  hosts: localhost
  tasks:
    - name: Create a Directory /opt/tomcat
      file:
        path: /opt/tomcat
        state: directory
        mode: 0755
        owner: root
        group: root
    - name: Download Tomcat using get_url
      get_url:
        url: https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.78/bin/apache-tomcat-8.5.78.tar.gz
        dest: /opt/tomcat
        mode: 0755
        group: iafzal
        owner: iafzal
```

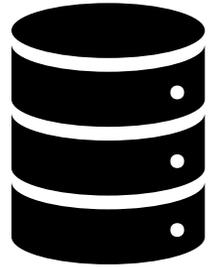
Run the playbook

```
# ansible-playbook tomcat.yml
```



# Create and Mount New Storage

- To create a new storage, we will power-off the VM add new disk (2GiB) from our virtualization software
- Also “**parted**” and “**mount**” module will be used in Ansible playbook
- Some Ansible distribution does not come with parted and mount module
  - Install **parted** and **mount** module
  - `ansible-galaxy collection install community.general`
  - `ansible-galaxy collection install ansible.posix`



**ERROR! couldn't resolve module/action 'mount'. This often indicates a misspelling, missing collection, or incorrect module path.**

# Create and Mount New Storage

```
# vim newstorage.yml
---
- name: Create and mount new storage
  hosts: all

  tasks:
    1 - name: create new partition
      parted:
        name: files
        label: gpt
        device: /dev/sdb
        number: 1
        state: present
        part_start: 1MiB
        part_end: 1GiB
    2 - name: Create xfs filesystem
      filesystem:
        dev: /dev/sdb1
        fstype: xfs
    3 - name: Create mount directory
      file:
        path: /data
        state: directory
    4 - name: mount the filesystem
      mount:
        src: /dev/sdb1
        fstype: xfs
        state: mounted
```

```
Run the playbook
# ansible-playbook newstorage.yml
```

