Using an Ansible playbook install firewalld on web1 node, start and enable its service as well. Name the playbook as firewall.yml and keep it under ~/playbooks.

```
[thor@ansible-controller playbooks]$ cat firewall.yml
---
- name: Install firewall
  hosts: all
  tasks:
  - name: Install firewalld
    yum:
      name: firewalld
      state: installed
  - name: STart service
    service:
      name: firewalld
      state: started
[thor@ansible-controller playbooks]$ ansible-playbook -i inventory firewall.yml
```

We have a requirement on web1 node to white list web2 node's IP address 172.20.1.101 in firewall. Create and run a playbook ~/playbooks/whitelist.yml to do so.

Add IP in internal zone.

```
- hosts: web1
  tasks:
    - firewalld:
      source: 172.20.1.101
      state: enabled
      zone: internal
      permanent: yes
      immediate: yes
```

Source : The source/network you would like to add/remove to/from firewalld.

State: - Enable or disable a setting. For ports: Should this port accept (enabled) or reject (disabled) connections. The states present and absent can only be used in zone level operations (i.e. when no other parameters but zone and state are set). Choices: "absent" "disabled" "enabled" "present"

Zone: - The firewalld zone to add/remove to/from. Note that the default zone can be configured per system but public is default from upstream. Available choices can be extended based on per-system configs, listed here are “out of the box” defaults.Possible values include block, dmz, drop, external, home, internal, public, trusted, work.

Permanent: - Should this configuration be in the running firewalld configuration or persist across reboots.
As of Ansible 2.3, permanent operations can operate on firewalld configs when it is not running (requires firewalld >= 0.3.9). Note that if this is false, immediate is assumed true. Choices: false true

Immediate : - Should this configuration be applied immediately, if set as permanent.Choices: false ← (default) true

We want to block 161/udp port on web1 node permanently. Make a playbook block.yml under ~/playbooks/ directory to do so.
Use zone: block

```
[thor@ansible-controller playbooks]$ cat block.yml
---
- name: Block port
  hosts: web1
  tasks:
  - name: Block port
    firewalld:
      port: 161/udp
      zone: block
      state: enabled
      permanent: yes
      immediate: yes
[thor@ansible-controller playbooks]$ ansible-playbook -i inventory block.yml
```

To verify, SSH to web1 server and run the following command:-
firewall-cmd --list-ports --zone=block

On web1 node add firewall rule in internal zone to enable https connection from Ansible controller machine and make sure that rule must persist even after system reboot. You can create a playbook https.yml under ~/playbooks/ directory.

IP address of ansible controller is 172.20.1.2.

Service: Name of a service to add/remove to/from firewalld. The service must be listed in output of firewall-cmd –get-services.

```
[thor@ansible-controller playbooks]$ cat https.yml
---
- name: https connection
  hosts: web1
  tasks:
  - name: https connection
    firewalld:
      source: 172.20.1.2
      service: https
      state: enabled
      permanent: yes
      zone: internal
  - name: system reload
    service:
      name: firewalld
      state: reloaded
[thor@ansible-controller playbooks]$ ansible-playbook -i inventory https.yml
```

We have a playbook ~/playbooks/web2-config.yml, it has some existing code to change apache's default port 80 to port 8082 as we want to run Apache on port 8082 on web2 node. Make some changes as given below before running the playbook.
A. Add an entry in ~/playbooks/inventory for web2 node, IP address of web2 node is 172.20.1.101 and ssh password and username are same as of web1 (username = root and password = Passw0rd).

```
[thor@ansible-controller playbooks]$ cat inventory
web1 ansible_host=172.20.1.100 ansible_ssh_pass=Passw0rd ansible_user=root
web2 ansible_host=172.20.1.101 ansible_ssh_pass=Passw0rd ansible_user=root
```

B. Update web2-config.yml to install httpd before updating its port in config, also start/enable its service.
C. Install firewalld package and start/enable its service.
D. As now Apache will listen on port 8082 so edit the playbook to add firewall rule in public zone so that Apache can allow all incoming traffic.

```
[thor@ansible-controller playbooks]$ cat web2-config.yml
---
- hosts: web2
  tasks:
    - name: install httpd
      yum:
        name: httpd, firewalld
        state: installed
    - name: start httpd
      service:
        name: "{{ item }}"
        state: started
        enabled: yes
      with_items:
        - httpd
        - firewalld
    - name: Change Apache port
      replace:
        path: /etc/httpd/conf/httpd.conf
        regexp: "Listen 80"
        replace: "Listen 8082"

    - name: Restart Apache service
      service:
        name: httpd
        state: restarted
    - name: Add firewall rule for Apache
      firewalld:
        port: 8082/tcp
        zone: public
        permanent: yes
        state: enabled
        immediate: true

[thor@ansible-controller playbooks]$ ansible-playbook -i inventory web2-config.yml
To verify firewall rules, SSH to web2 server and run the following commands:-
ssh root@web2
firewall-cmd --list-ports --zone=public
```