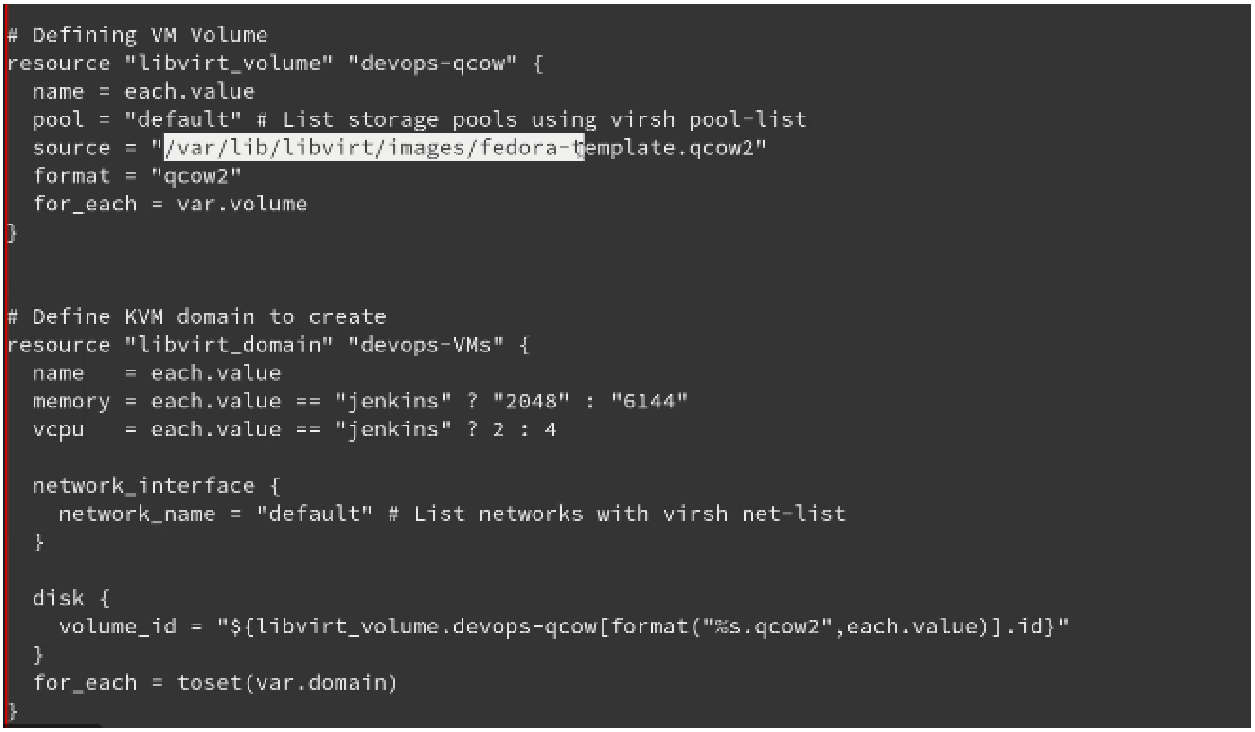
Final



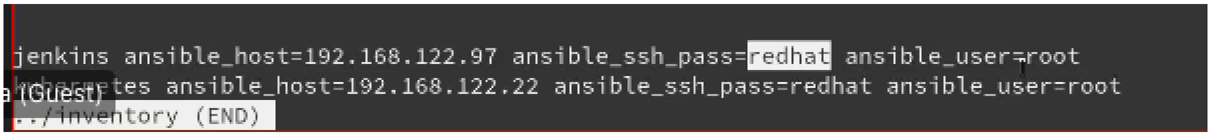
```
# Defining VM Volume
resource "libvirt_volume" "devops-qcow" {
  name = each.value
  pool = "default" # List storage pools using virsh pool-list
  source = "/var/lib/libvirt/images/fedora-template.qcow2"
  format = "qcow2"
  for_each = var.volume
}


# Define KVM domain to create
resource "libvirt_domain" "devops-VMs" {
  name   = each.value
  memory = each.value == "jenkins" ? "2048" : "6144"
  vcpu   = each.value == "jenkins" ? 2 : 4

  network_interface {
    network_name = "default" # List networks with virsh net-list
  }

  disk {
    volume_id = "${libvirt_volume.devops-qcow[format("%s.qcow2",each.value)].id}"
  }
  for_each = toset(var.domain)
}
```

We are creating 2 VMs jenkins and Kubernetes using terraform file.

Now, we will configure these VMs using ansible,

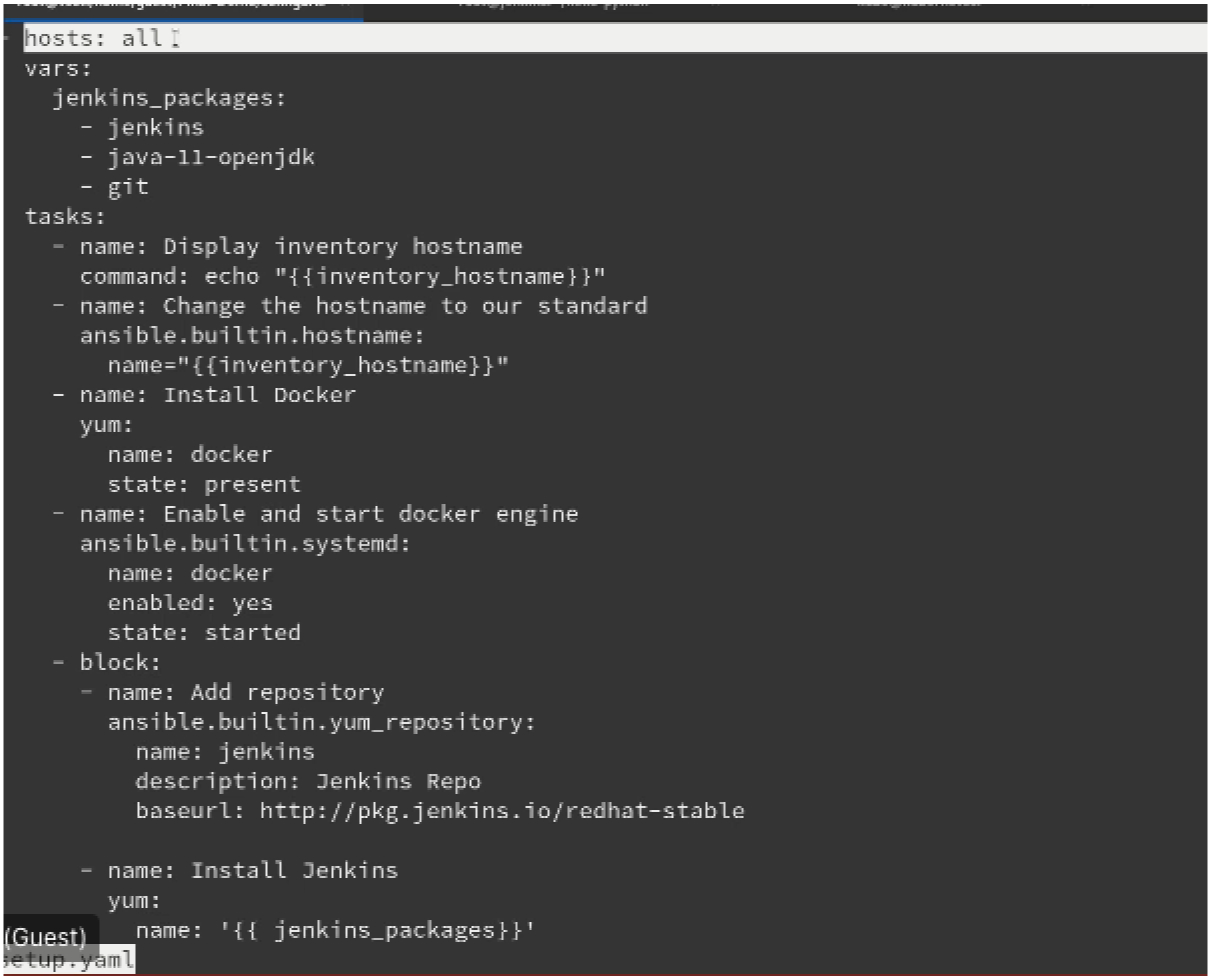
Terraform doesn’t have idempotency, whereas ansible is idempotent.

Ansible inventory



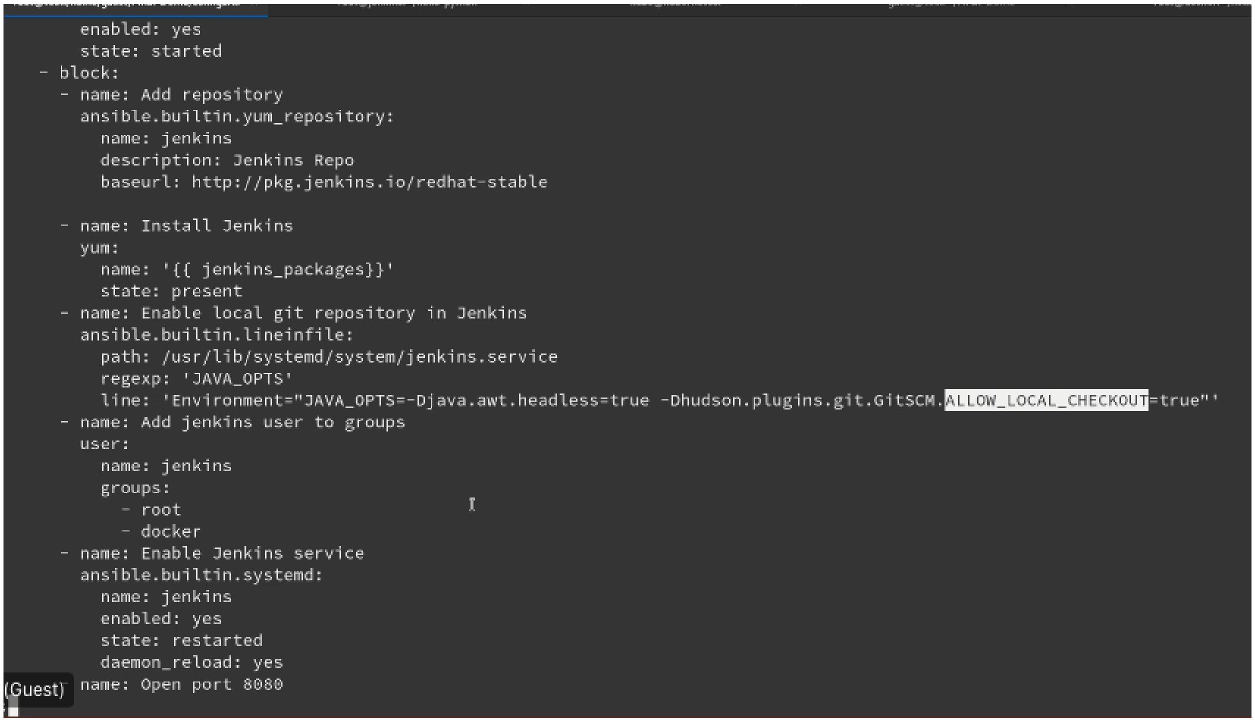
```
jenkins ansible_host=192.168.122.97 ansible_ssh_pass=redhat ansible_user=root
kubernetes ansible_host=192.168.122.22 ansible_ssh_pass=redhat ansible_user=root
../inventory (END)
```

ansible script:



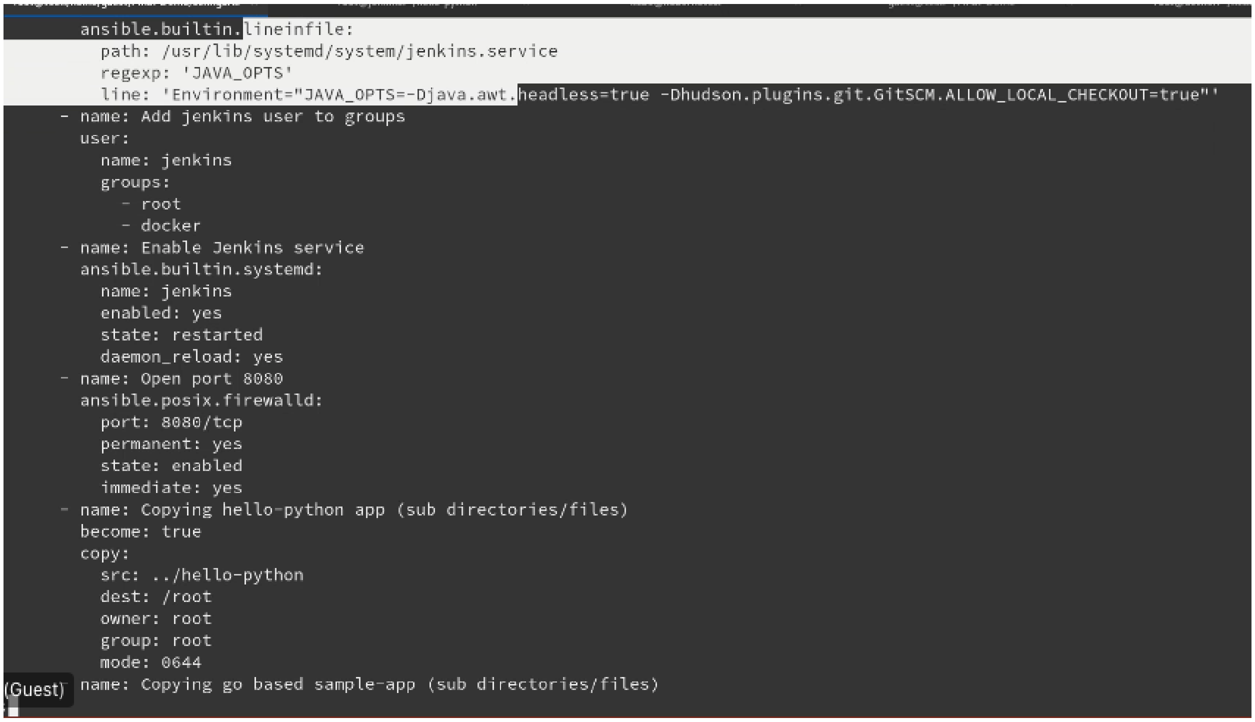
```
- hosts: all
  vars:
    jenkins_packages:
      - jenkins
      - java-11-openjdk
      - git
  tasks:
    - name: Display inventory hostname
      command: echo "{{inventory_hostname}}"
    - name: Change the hostname to our standard
      ansible.builtin.hostname:
        name="{{inventory_hostname}}"
    - name: Install Docker
      yum:
        name: docker
        state: present
    - name: Enable and start docker engine
      ansible.builtin.systemd:
        name: docker
        enabled: yes
        state: started
    - block:
      - name: Add repository
        ansible.builtin.yum_repository:
          name: jenkins
          description: Jenkins Repo
          baseurl: http://pkg.jenkins.io/redhat-stable

      - name: Install Jenkins
        yum:
          name: '{{ jenkins_packages}}'
```

second task changes hostname of the machine to what we have specified in inventory file, i.e jenkins and kubernetes

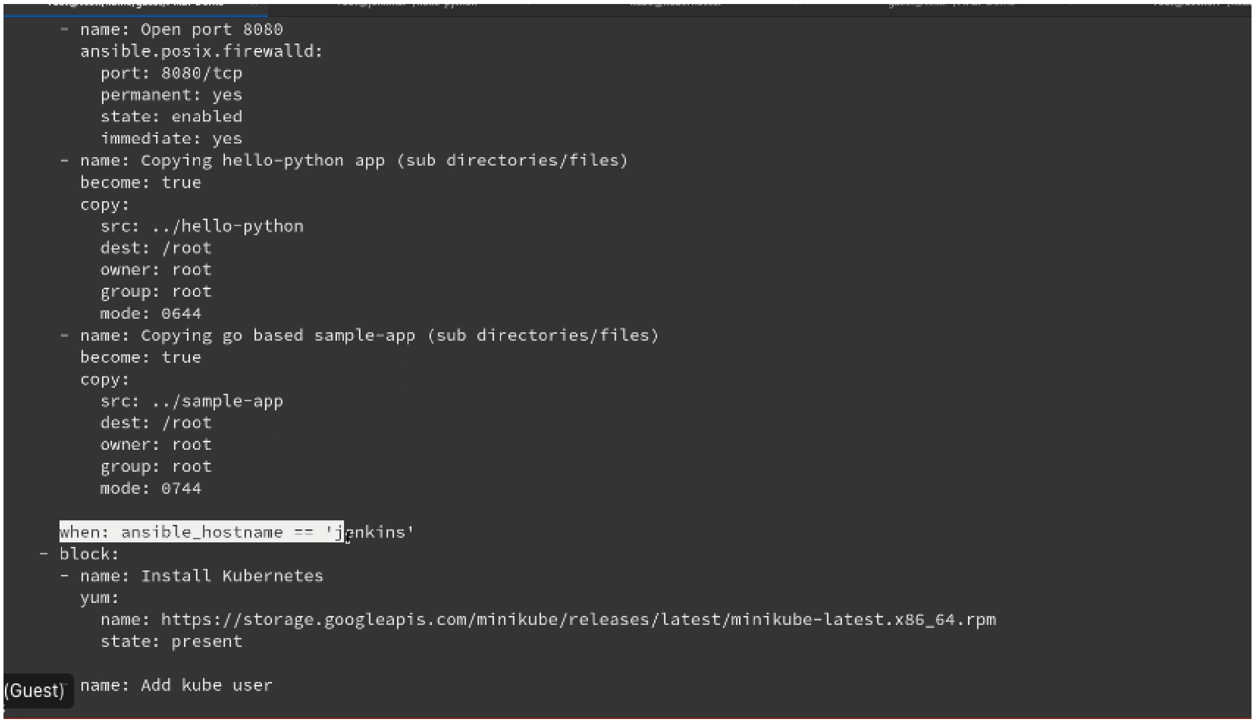


```
      enabled: yes
      state: started
  - block:
    - name: Add repository
      ansible.builtin.yum_repository:
        name: jenkins
        description: Jenkins Repo
        baseurl: http://pkg.jenkins.io/redhat-stable

    - name: Install Jenkins
      yum:
        name: '{{ jenkins_packages}}'
        state: present
    - name: Enable local git repository in Jenkins
      ansible.builtin.lineinfile:
        path: /usr/lib/systemd/system/jenkins.service
        regexp: 'JAVA_OPTS'
        line: 'Environment="JAVA_OPTS=-Djava.awt.headless=true -Dhudson.plugins.git.GitSCM.ALLOW_LOCAL_CHECKOUT=true"'
    - name: Add jenkins user to groups
      user:
        name: jenkins
        groups:
          - root
          - docker
    - name: Enable Jenkins service
      ansible.builtin.systemd:
        name: jenkins
        enabled: yes
        state: restarted
        daemon_reload: yes
      name: Open port 8080
```

regexp matches the ‘JAVA_OPTS’ and replaces this line with the provided line. It is used to allow local git repository.

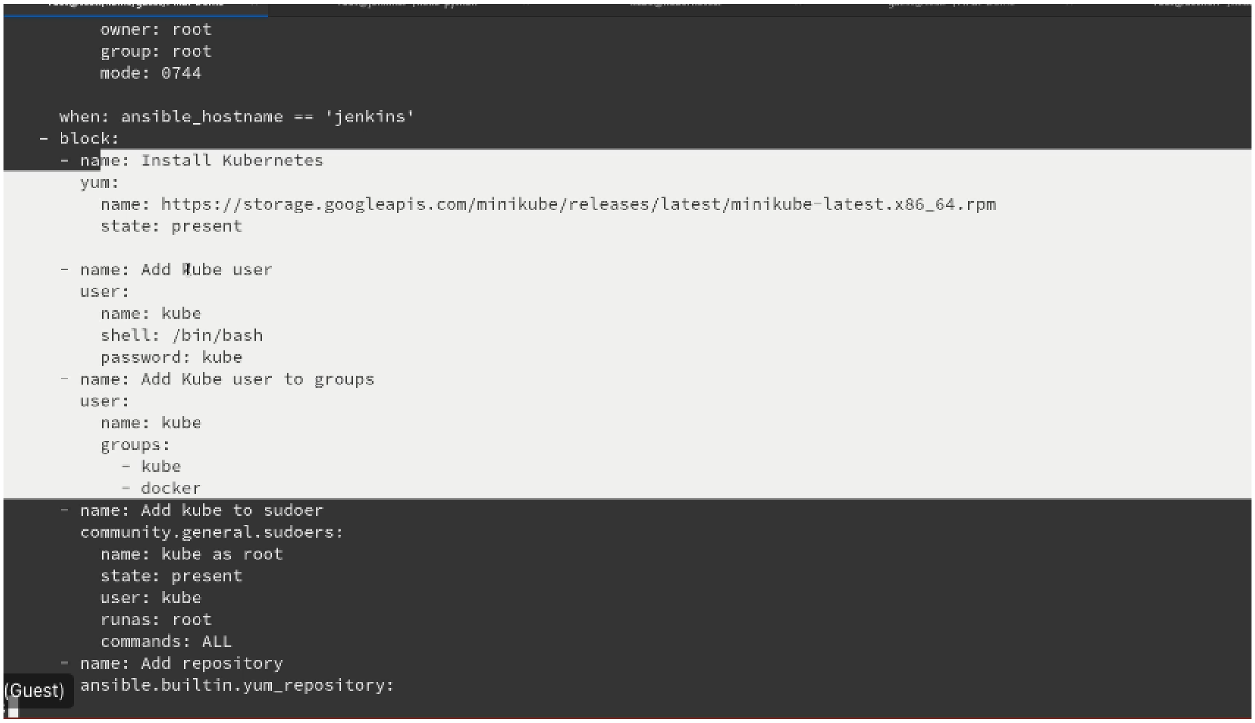


```
      ansible.builtin.lineinfile:
        path: /usr/lib/systemd/system/jenkins.service
        regexp: 'JAVA_OPTS'
        line: 'Environment="JAVA_OPTS=-Djava.awt.headless=true -Dhudson.plugins.git.GitSCM.ALLOW_LOCAL_CHECKOUT=true"'
    - name: Add jenkins user to groups
      user:
        name: jenkins
        groups:
          - root
          - docker
    - name: Enable Jenkins service
      ansible.builtin.systemd:
        name: jenkins
        enabled: yes
        state: restarted
        daemon_reload: yes
    - name: Open port 8080
      ansible.posix.firewalld:
        port: 8080/tcp
        permanent: yes
        state: enabled
        immediate: yes
    - name: Copying hello-python app (sub directories/files)
      become: true
      copy:
        src: ../hello-python
        dest: /root
        owner: root
        group: root
        mode: 0644
      name: Copying go based sample-app (sub directories/files)
```

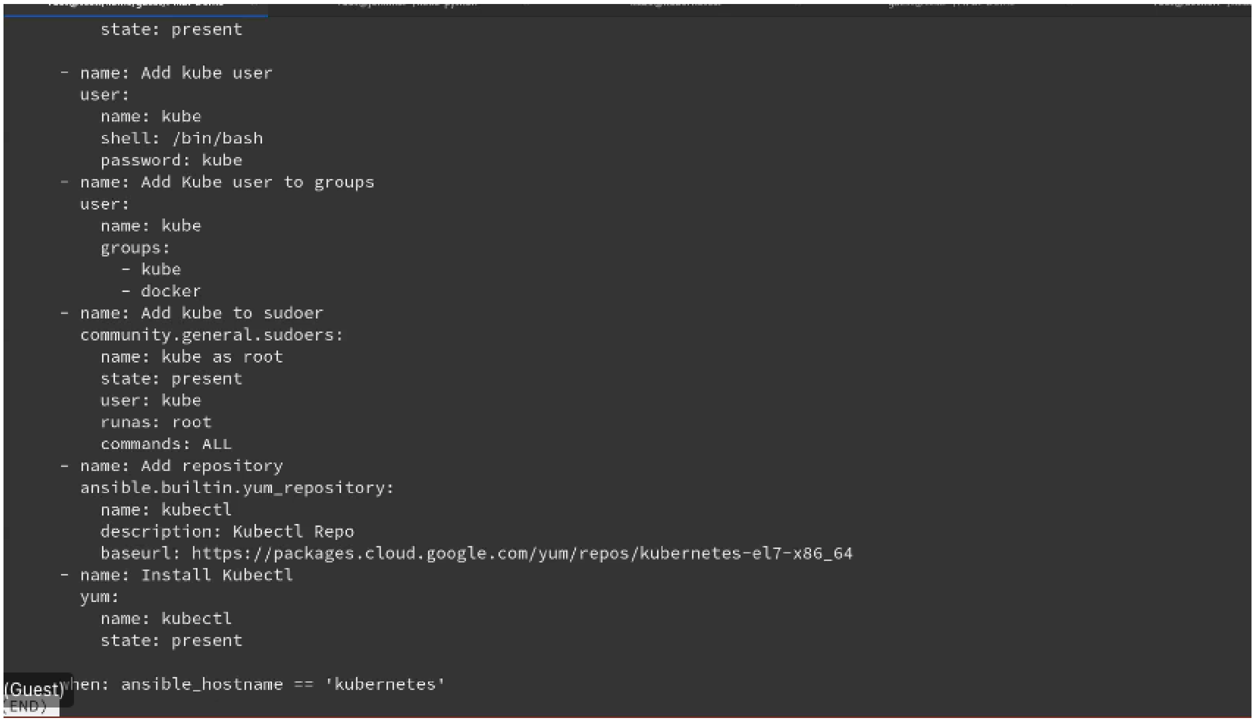


```
  - name: Open port 8080
    ansible.posix.firewalld:
      port: 8080/tcp
      permanent: yes
      state: enabled
      immediate: yes
  - name: Copying hello-python app (sub directories/files)
    become: true
    copy:
      src: ../hello-python
      dest: /root
      owner: root
      group: root
      mode: 0644
  - name: Copying go based sample-app (sub directories/files)
    become: true
    copy:
      src: ../sample-app
      dest: /root
      owner: root
      group: root
      mode: 0744

  when: ansible_hostname == 'jenkins'
- block:
  - name: Install Kubernetes
    yum:
      name: https://storage.googleapis.com/minikube/releases/latest/minikube-latest.x86_64.rpm
      state: present

    name: Add kube user
```

here, this when: ansible_hostname ==’jenkins’, so all the tasks of the block is performed only on jenkins hostname.

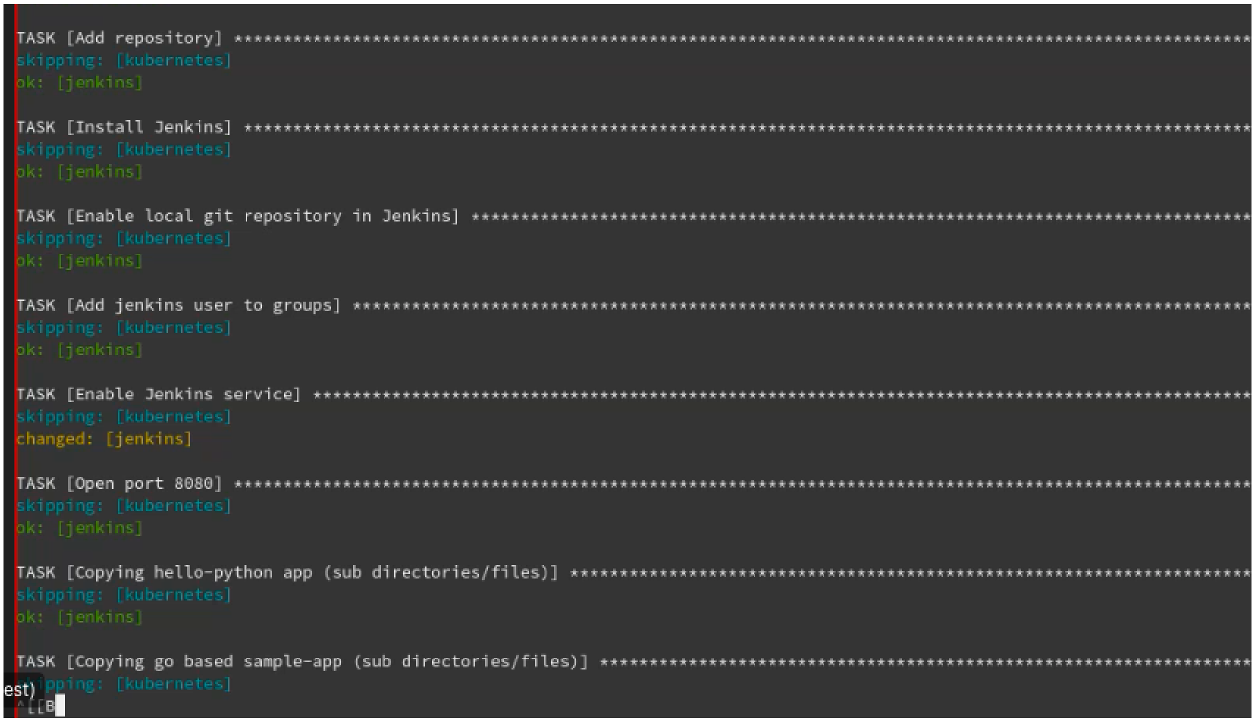


```
      owner: root
      group: root
      mode: 0744

  when: ansible_hostname == 'jenkins'
- block:
  - name: Install Kubernetes
    yum:
      name: https://storage.googleapis.com/minikube/releases/latest/minikube-latest.x86_64.rpm
      state: present

  - name: Add kube user
    user:
      name: kube
      shell: /bin/bash
      password: kube
  - name: Add Kube user to groups
    user:
      name: kube
      groups:
        - kube
        - docker
  - name: Add kube to sudoer
    community.general.sudoers:
      name: kube as root
      state: present
      user: kube
      runas: root
      commands: ALL
  - name: Add repository
    ansible.builtin.yum_repository:
```



```
        state: present

    - name: Add kube user
      user:
        name: kube
        shell: /bin/bash
        password: kube
    - name: Add Kube user to groups
      user:
        name: kube
        groups:
          - kube
          - docker
    - name: Add kube to sudoer
      community.general.sudoers:
        name: kube as root
        state: present
        user: kube
        runas: root
        commands: ALL
    - name: Add repository
      ansible.builtin.yum_repository:
        name: kubectl
        description: Kubectl Repo
        baseurl: https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
    - name: Install Kubectl
      yum:
        name: kubectl
        state: present

    when: ansible_hostname == 'kubernetes'
(END)
```

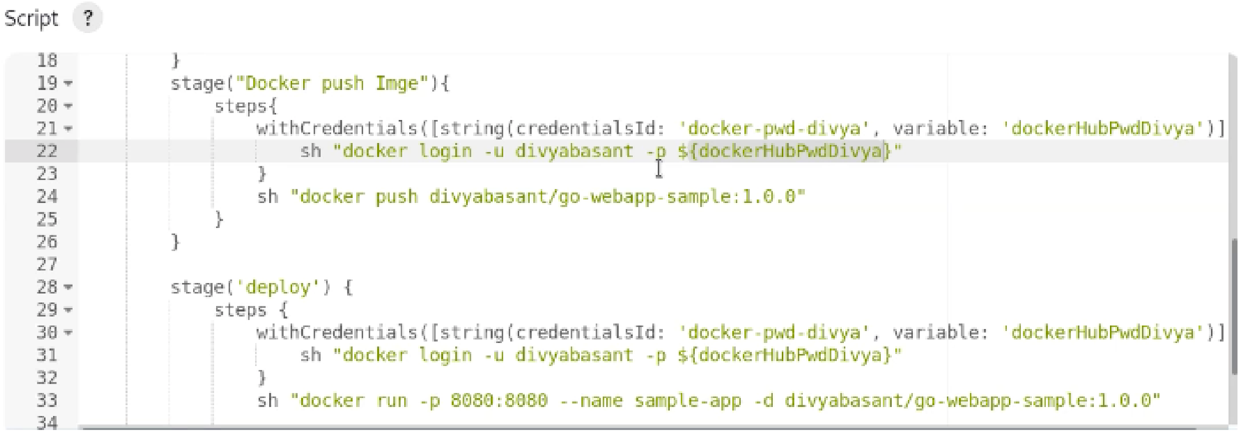
skipping means, this task has some condition and it skipped for kubernetes VM, and did it on jenkins VM.



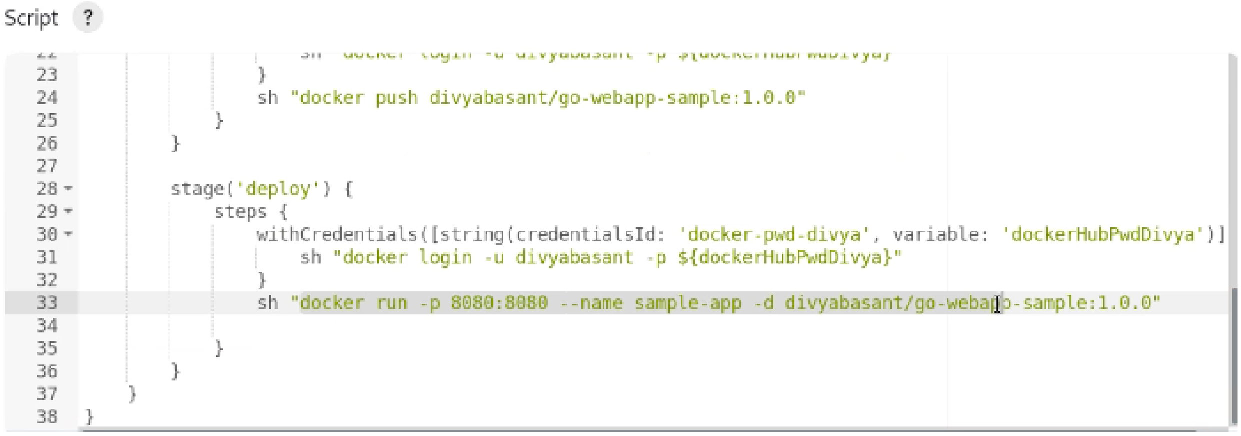
```
TASK [Add repository] ******************************************************************************************************
skipping: [kubernetes]
ok: [jenkins]

TASK [Install Jenkins] *****************************************************************************************************
skipping: [kubernetes]
ok: [jenkins]

TASK [Enable local git repository in Jenkins] ******************************************************************************
skipping: [kubernetes]
ok: [jenkins]

TASK [Add jenkins user to groups] ******************************************************************************************
skipping: [kubernetes]
ok: [jenkins]

TASK [Enable Jenkins service] **********************************************************************************************
skipping: [kubernetes]
changed: [jenkins]

TASK [Open port 8080] ******************************************************************************************************
skipping: [kubernetes]
ok: [jenkins]

TASK [Copying hello-python app (sub directories/files)] ********************************************************************
skipping: [kubernetes]
ok: [jenkins]

TASK [Copying go based sample-app (sub directories/files)] *****************************************************************
skipping: [kubernetes]
^[[B
```

If running command manually works, but it doesn't work in jenkins, its probably due to you need a plugin related to the application, install it.

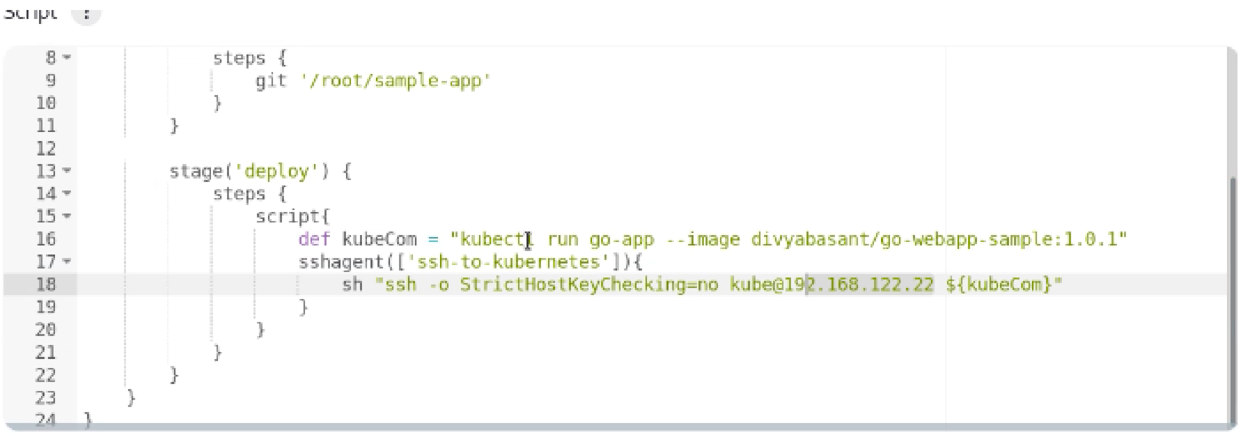
Now, we are using docker to build a image from git repo, pushing it to dockerhub from the jenkins pipeline, and then running it on docker



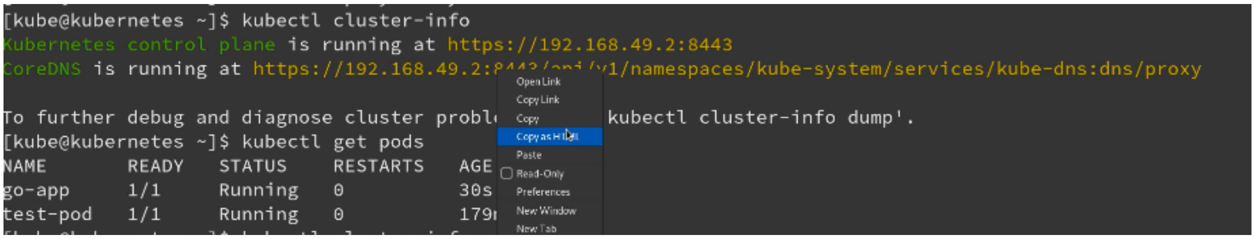
```
        }
        stage("Docker push Imge"){
            steps{
                withCredentials([string(credentialsId: 'docker-pwd-divya', variable: 'dockerHubPwdDivya')]
                    sh "docker login -u divyabasant -p ${dockerHubPwdDivya}"
                }
                sh "docker push divyabasant/go-webapp-sample:1.0.0"
            }
        }

        stage('deploy') {
            steps {
                withCredentials([string(credentialsId: 'docker-pwd-divya', variable: 'dockerHubPwdDivya')]
                    sh "docker login -u divyabasant -p ${dockerHubPwdDivya}"
                }
                sh "docker run -p 8080:8080 --name sample-app -d divyabasant/go-webapp-sample:1.0.0"
```



```
                    sh "docker login -u divyabasant -p ${dockerHubPwdDivya}"
                }
                sh "docker push divyabasant/go-webapp-sample:1.0.0"
            }
        }

        stage('deploy') {
            steps {
                withCredentials([string(credentialsId: 'docker-pwd-divya', variable: 'dockerHubPwdDivya')]
                    sh "docker login -u divyabasant -p ${dockerHubPwdDivya}"
                }
                sh "docker run -p 8080:8080 --name sample-app -d divyabasant/go-webapp-sample:1.0.0"

            }
        }
    }
}
```



```
            steps {
                git '/root/sample-app'
            }
        }

        stage('deploy') {
            steps {
                script{
                    def kubeCom = "kubectl run go-app --image divyabasant/go-webapp-sample:1.0.1"
                    sshagent(['ssh-to-kubernetes']){
                        sh "ssh -o StrictHostKeyChecking=no kube@192.168.122.22 ${kubeCom}"
                    }
                }
            }
        }
    }
}
```



```
[kube@kubernetes ~]$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster probl            kubectl cluster-info dump'.
[kube@kubernetes ~]$ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
go-app      1/1     Running   0          30s
test-pod    1/1     Running   0          179
```

.kube folder has a config file.