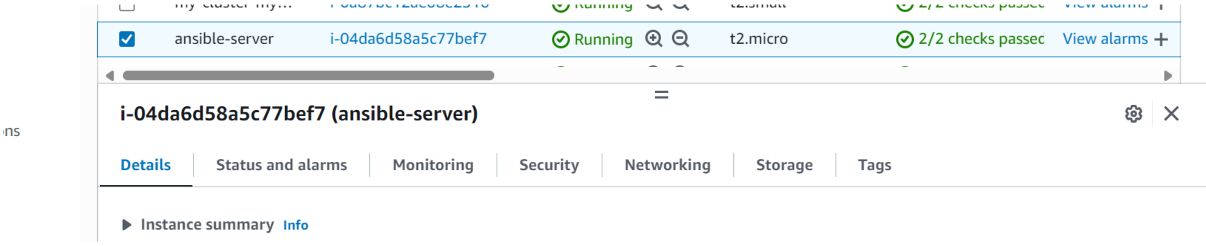
Eks hosting with ansible

Create an instance



Exchange keys between ansible and Jenkins

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10748211> cd .\Downloads\
PS C:\Users\10748211\Downloads> ssh -i "key1.pem" ec2-user@ec2-3-108-62-81.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-3-108-62-81.ap-south-1.compute.amazonaws.com (3.108.62.81)' can't be established.
ED25519 key fingerprint is SHA256:HEOW2OxLoMuN4xZGBd5PKv2ho8xYO5oOJNaQEvdySuE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-108-62-81.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-37-115 ~]$ sudo su -
[root@ip-172-31-37-115 ~]# hostnamectl set-hostname ansible
[root@ip-172-31-37-115 ~]# bash
[root@ansible ~]# |
```

```
[root@ansible ~]# yum install ansible* -y
Last metadata expiration check: 0:02:52 ago on Mon Sep 30 12:09:22 2024.
Dependencies resolved.
========================================================================================================================
 Package                          Architecture          Version                              Repository            Size
========================================================================================================================
Installing:
 ansible                          noarch                8.3.0-1.amzn2023.0.1                 amazonlinux           32 M
 ansible-core                     x86_64                2.15.3-1.amzn2023.0.4                amazonlinux          2.5 M
 ansible-packaging                noarch                1-11.amzn2023.0.1                    amazonlinux           14 k
 ansible-packaging-tests          noarch                1-11.amzn2023.0.1                    amazonlinux          9.7 k
 ansible-srpm-macros              noarch                1-11.amzn2023.0.1                    amazonlinux           22 k
Installing dependencies:
 ansible-test                     x86_64                2.15.3-1.amzn2023.0.4                amazonlinux          705 k
 git-core                         x86_64                2.40.1-1.amzn2023.0.3                amazonlinux          4.3 M
 python3-apipkg                   noarch                1.5-12.amzn2023.0.2                  amazonlinux           19 k
 python3-execnet                  noarch                1.7.1-5.amzn2023.0.2                 amazonlinux          333 k
 python3-iniconfig                noarch                1.1.1-2.amzn2023.0.2                 amazonlinux           17 k
 python3-packaging                noarch                21.3-2.amzn2023.0.2                  amazonlinux           72 k
```

```
[root@ansible ~]# ansible --version
ansible [core 2.15.3]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.9.16 (main, Jul  5 2024, 00:00:00) [GCC 11.4.1 20230605 (Red Hat 11.4.1-2)] (/usr/bin/python3.9)
  jinja version = 3.1.4
  libyaml = True
```

```
[root@ansible ansible]# vim ansible.cfg
[root@ansible ansible]# ansible --version
ansible [core 2.15.3]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/usr/share/ansible']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.9.16 (main, Jul  5 2024, 00:00:00) [GCC 11.4.1 20230605 (Red Hat 11.4.1-2)] (/usr/bin/python3.9)
  jinja version = 3.1.4
  libyaml = True
[root@ansible ansible]#
```

```
# execute a module on the remote server. This can result in a significant

# performance improvement when enabled, however when using "sudo:" you must

# first disable 'requiretty' in /etc/sudoers

#

# By default, this option is disabled to preserve compatibility with

# sudoers configurations that have requiretty (the default on many distros).

#

#pipelining = False

# if True, make ansible use scp if the connection type is ssh

# (default is sftp)

#scp_if_ssh = True

[accelerate]

accelerate_port = 5099

accelerate_timeout = 30

accelerate_connect_timeout = 5.0
```

Paste the ip from cluster terminal

```
root@cluster:~# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enX0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:9d:32:1b:fc:23 brd ff:ff:ff:ff:ff:ff
    inet 172.31.41.252/20 metric 100 brd 172.31.47.255 scope global dynamic enX0
       valid_lft 2761sec preferred_lft 2761sec
    inet6 fe80::9d:32ff:fe1b:fc23/64 scope link
       valid_lft forever preferred_lft forever
root@cluster:~#
```

Paste the IP in vim hosts file in ansible terminal

Add the ansible IP first and the cluster IP

```
[cluster-server]

172.31.37.115

[cluster-server1]

172.31.41.252
~
~
~
~
~
~
~
```

```
[root@ansible ansible]# ansible all --list-hosts
  hosts (1):
    172.31.41.252
[root@ansible ansible]#
```

Go to eks cluster inside vim /etc/ssh/sshd_config and make the following changes

```
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
```

Exchange the keys between ansible and cluster to establish SSH connection

```
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ubuntu\" rather than the user \"root\".';echo
;sleep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDM8RVJyTNLOaAhHXMzNUg09+axPTU+hwWM7l7MUTjHhsUVoBPYl96xTSlA7L2cAHe5cju2mD/tXyYsXNTqDTTM
EZhVxzYZIWiSxFHHDqcEDsSZ1U3w9MO1/BDqr+J0eBRk7myJ9Jrgr1EXuxJCc05R1LibU4qeoRf4WKTRbfcHC8ouoAbK3dSB/PufiDqQZOvS0M+uthDttzFmjfrsMF40Y8Y5BD3G3dMEpDx8
3KX5ykFaVExmrOckbuCI4GoTmRSFIuDTQk4kYobMJbf5yr8lXO79j9+6+k5V5w4fHT0R3cpv29PyQOyEKPJ0YYE65tQbNE3L9rTFyFw9abLm7hjL key1

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC3n9JnLVTcUVLKPY7LCJNUzfNWOUAJFRXlwZ80ll7B1CivuUceTIdyXDsKq9Fni94w9MEXMQQqmARPMWDxJxp+yDxbAlxhA3dSIEmjYLoX
eM9WzZcrgWOnOrMJg1SLvySjRZ7/o8IJ73Vs5ZVVKdTfqfGjDcK6xhwH8oL07ZMJRShCy/WLldx3K4kqxRqy74lZ704ztQZNE1WLRKaiTeYOCzR/GhCiukhC8YhxslPCI76YohUkuXN/mOBK
xEYjawv/fKZB8XsY0nYNrOrYO5y1qPD4D15M+uhNIiMtDiH/iEmUNpVHRemcTr7Bpan+DC6u36jixwLk5Z+f9ZaJLyBAEfnDk4mgz4E6HYmRUSmM3/Vy2vcCa9pa7jnfD6S1pbd9Yr0N93Vu
hKVSQ8ESkkfX4j6kqnAwFhNcqOBa96rYXDAgY0CWdJcZOuWbS4JW7fjpqi6LBLCCOt/FJwQ1TIpBWN2WLXOccVlAhQcoEsJ4W///BIz9Zb3SQgW1lH6TPF0= root@ansible
~
~
~
~
~
~
```

```
[root@ansible ~]# cd .ssh
[root@ansible .ssh]# vim etc/ssh/sshd_config
[root@ansible .ssh]# vim /etc/ssh/sshd_config
```

```
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none


# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# Explicitly disable PasswordAuthentication. By presetting it, we
# avoid the cloud-init set_passwords module modifying sshd_config and
# restarting sshd in the default instance launch configuration.
PasswordAuthentication yes
PermitEmptyPasswords no

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes

-- INSERT --
```

```
[root@ansible ~]# ansible 172.31.41.252 -m ping
172.31.41.252 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
[root@ansible ~]#
```

Inside cd /etc/ansible directory

```
[root@ansible ansible]# vim cluster.yml
[root@ansible ansible]# vim service.yml
[root@ansible ansible]#
```

Inside the vim cluster.yml in hosts add the ip of cluster and change the image url from

docker

```yaml
---
- hosts: 172.31.41.252
  gather_facts: no
  tasks:
    - name: Deploy application in Kubernetes
      k8s:
        state: present
        definition:
          apiVersion: apps/v1
          kind: Deployment
          metadata:
            name: my-app
            namespace: default
            labels:
              app: my-app
          spec:
            replicas: 3
            selector:
              matchLabels:
                app: my-app
            template:
              metadata:
                labels:
                  app: my-app
              spec:
                containers:
                - name: my-app
                  image: 626635426238.dkr.ecr.ap-south-1.amazonaws.com/docker_ecr:latest
                  imagePullPolicy: Always
                  ports:
                  - containerPort: 8080
            strategy:
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: regapp-service
  labels:
    app: my-app
spec:
  selector:
    app: my-app

  ports:
    - port: 8080
      targetPort: 8080

  type: LoadBalancer
~
~
~
~
~
~
```

```
[root@ansible ansible]# ansible-playbook cluster.yml --syntax-check
[WARNING]: Could not match supplied host pattern, ignoring: cluster-server

playbook: cluster.yml
```

On Kubernetes cluster terminal paste these commands for ubuntu server

Update the package list

sudo apt update

Install OpenShift and Kubernetes Python client libraries

sudo apt install pipx

pipx install openshift

pipx install Kubernetes

In ansible terminal inside cd /etc/ansible

```
[root@ansible ~]# cd /etc/ansible
[root@ansible ansible]# ls
ansible.cfg  cluster.yml  hosts  roles  service.yml
[root@ansible ansible]#
```

```
[root@ansible ansible]# vim hosts
[root@ansible ansible]# vim cluster.yml
[root@ansible ansible]# ansible-playbook cluster.yml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [172.31.41.252] ***********************************************************************************************************

TASK [Deploy application in Kubernetes] ****************************************************************************************
ok: [172.31.41.252]

TASK [Copy service file to target] *********************************************************************************************
ok: [172.31.41.252]

TASK [Create Kubernetes Service] ***********************************************************************************************
changed: [172.31.41.252]

PLAY RECAP *********************************************************************************************************************
172.31.41.252              : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@ansible ansible]#
```

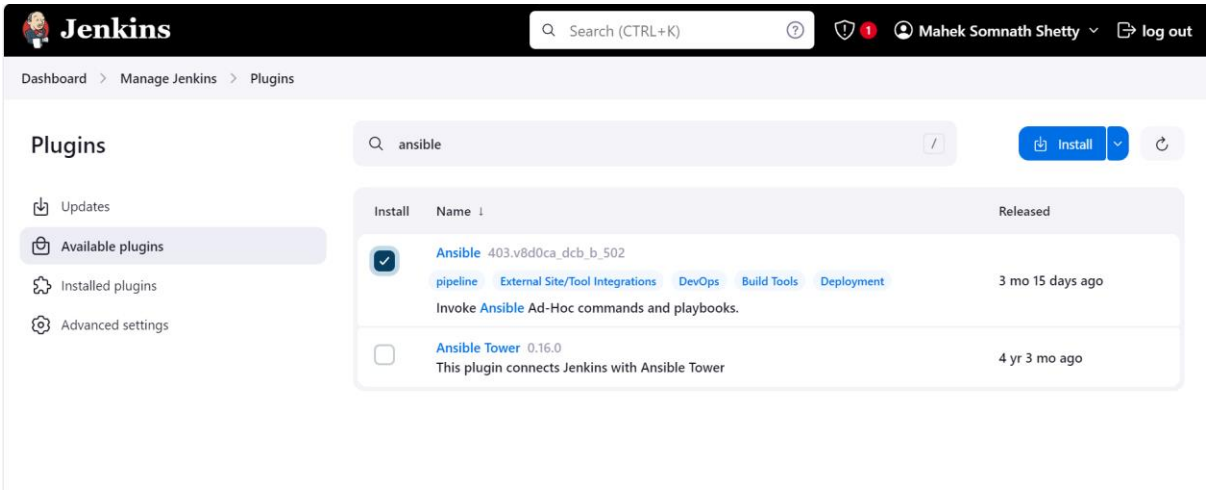
In the cluster terminal use kubectl get svc and copy paste the external ip link on the

browser

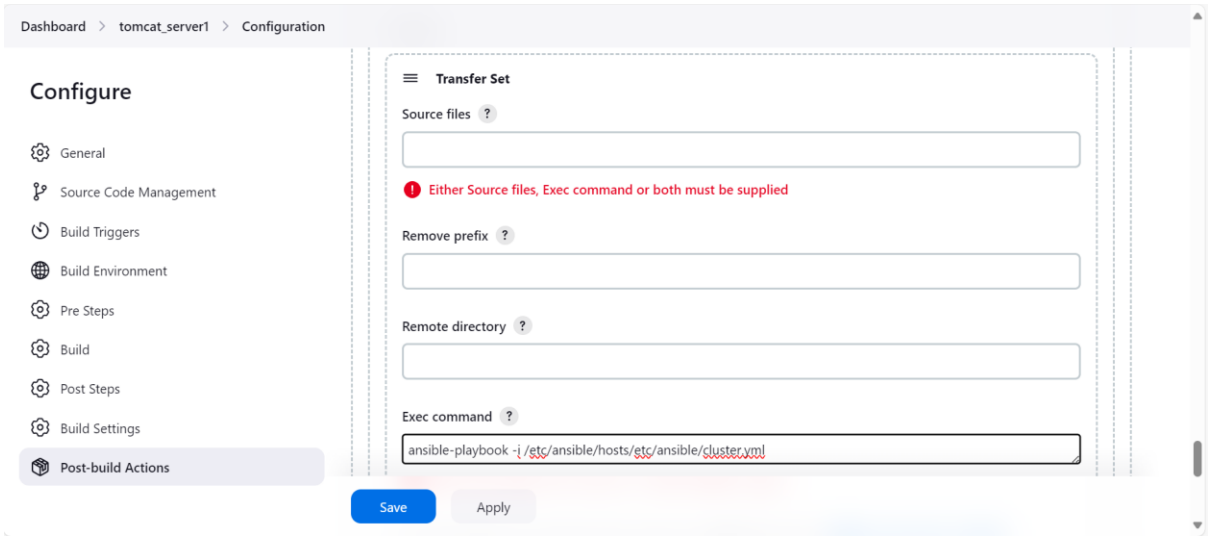
Go to the cluster terminal

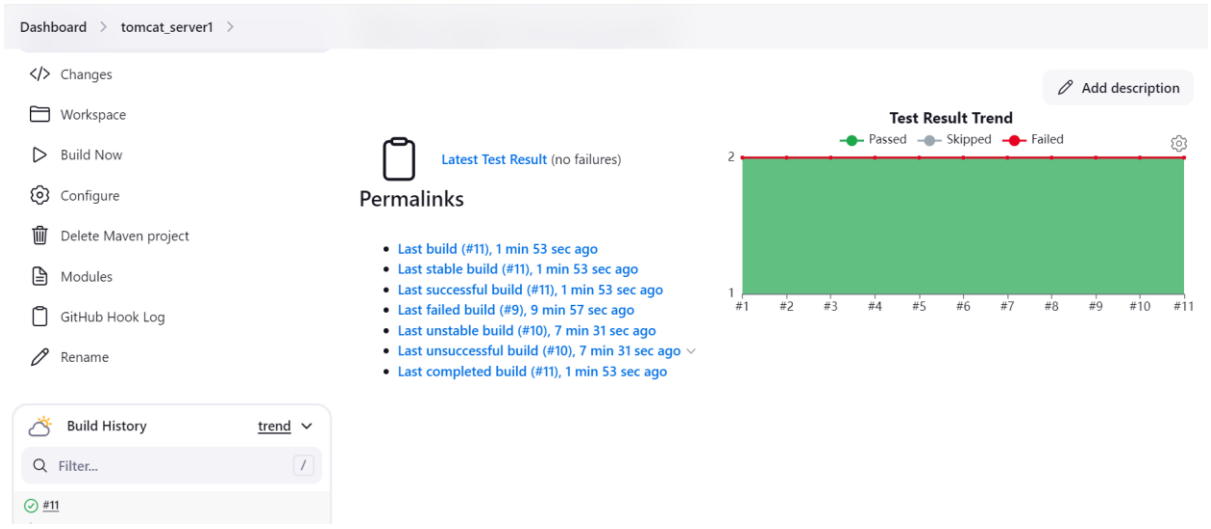
```
root@cluster:~# kubectl delete deployment my-app
deployment.apps "my-app" deleted
root@cluster:~# kubectl delete service regapp-service
service "regapp-service" deleted
```

Go to Jenkins >> manage Jenkins >> available plugins >> ansible >> install



Go to Jenkins >> system





Go to cluster and write kubectl get svc and paste the key