Name: Chaudhari Ganesh Dadabhau

Configuration of K8s Multinode Cluster over AWS by integrating ansible and terraform with dynamic inventory.

Task No.: 5

Integration of terraform, ansible, AWS and k8s. lets understand what is terraform, ansible, AWS and k8s.

Terraform: Terraform is an open-source infrastructure as code software tool. It is developed in Go programming language. The terraform is used to set up infrastructure on Cloud like AWS, google ,azure etc. It supports to multiple cloud providers. It is used for building, changing, and versioning infrastructure safely and efficiently. It allows users to manage cloud services through a language named HashiCorp Configuration Language (HCL).Terraform quickly deploys Infrastructure as Code, due to which environment installation and development is faster with Terraform.

Ansible: Ansible is open-source configuration management tool. ansible is agentless tool. Ansible is built on the top of python language. The node which actually run playbook is called control node and the target node means the node on which playbook will apply is called managed node.

AWS: Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.

Kubernetes: Kubernetes is an open-source container-orchestration system for automating computer application deployment, scaling, and management. Actually k8s manage the container engines like Docker, CRIO, Podman etc. k8s is developed in Go programming language. Kubernetes is a portable, extensible, platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem.

First lets provision AWS ec2 instances using terraform. create folder ec2_provision. then inside that folder create file ec2.tf then write code to

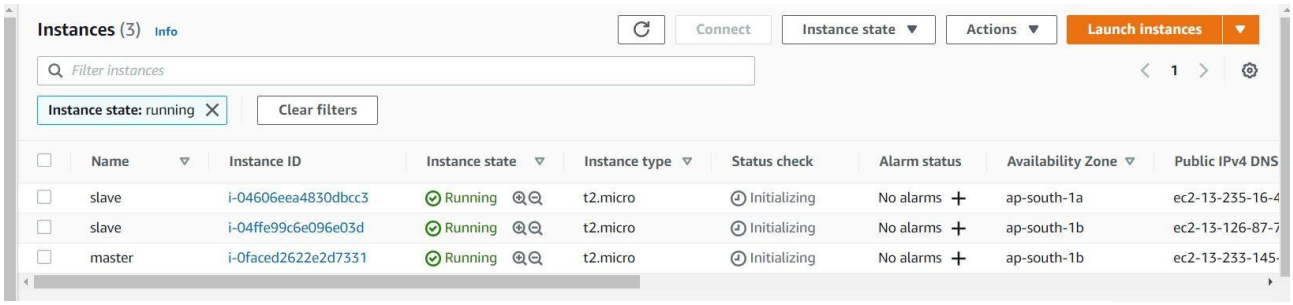
launch EC2 instance. but here AWS AWS credentials required. The Terraform by default taking Credentials from home directory of user (/home/.aws/credentials). I have written code to launch three instances i.e. one for master node, and two for slave nodes in default VPC and using Amazon2 AMI.

```
[root@control_name ec2_provision]# cat /root/.aws/credentials
[default]
aws_access_key_id: [redacted]
aws_secret_access_key: [redacted]
[root@control_name ec2_provision]#
```

```
[root@control_name k8s]# cd ec2_provision/
[root@control_name ec2_provision]# cat ec2.tf
provider "aws" {
  region = "ap-south-1"

}

data "aws_vpc" "default" {
  default = true
}

data "aws_subnet_ids" "all" {
  vpc_id = data.aws_vpc.default.id
}



resource "aws_instance" "master" {

  ami = "ami-08e0ca9924195beba"
  instance_type = "t2.micro"
  key_name= "vpc"
  count= 1

  tags={
  Name="master"

  }
}
resource "aws_instance" "slave" {

  ami = "ami-08e0ca9924195beba"
  instance_type = "t2.micro"
  key_name= "vpc"
  count= 2

  tags={
  Name="slave"

  }

}
```

after completing above code we have initialize current directory using terraform init then we have to use terraform apply -auto-approve. then you will see result

```
[root@control_name ~]# cd /k8s/
[root@control_name k8s]# ls
ansible.cfg  ec2_provision  host.py  k8s.yml  master  slave  vpc.pem
[root@control_name k8s]# cd ec2_provision/
[root@control_name ec2_provision]# ls
ec2.tf  terraform.tfstate  terraform.tfstate.backup
[root@control_name ec2_provision]#
```

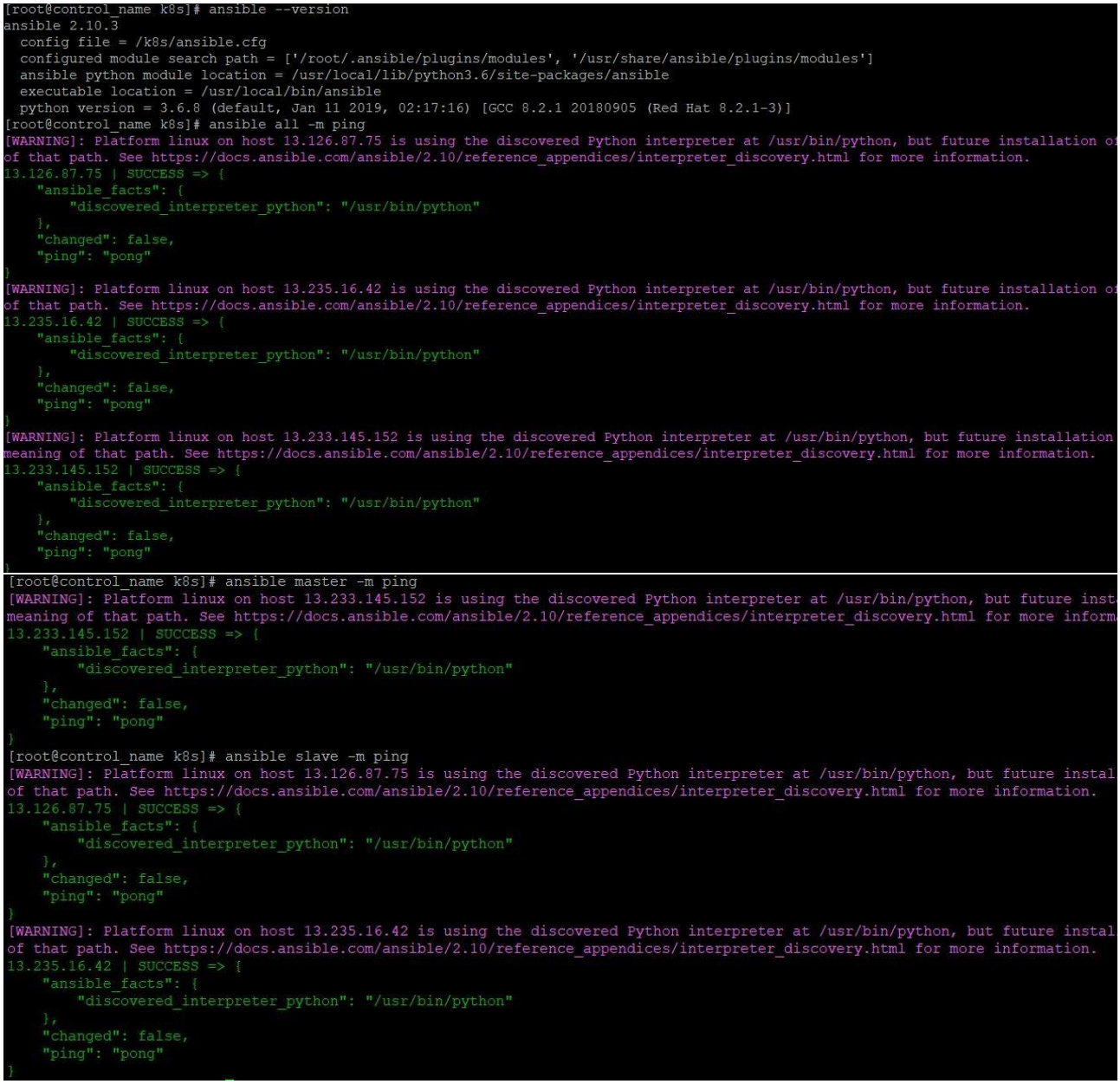


We done with infrastructure part. Now lets move towards configuration of K8s master and K8s slave. First we have to write ansible configuration file and dynamic inventory file using python like

```
[root@control_name k8s]# cat ansible.cfg
[defaults]
inventory= host.py
remote_user= ec2-user
private_key_file=vpc.pem
host_key_checking=false
[privilege_escalation]
become= yes
become_method= sudo
become_user=root
ask_become_pass=no
```

```
[root@control_name k8s]# cat host.py
#!/usr/bin/python3
import json  #ansible can understand easily

try:
    import boto3       #used to connect aws
except Exception as e:
    print("please check boto module is installed or not\n if not then use 'pip3 install boto3' ")
    print(e)
# we are getting IPs and appending in list
def get_hosts(ec2_ob, fv):
    f={"Name":"tag:Name", "Values": [fv]}
    hosts=[]

    for each in ec2_ob.instances.filter(Filters=[f]):
        hosts.append(each.public_ip_address)
    return hosts



def main():
    ec2_ob=boto3.resource("ec2","ap-south-1")   #checking ec2 info in ap-south-1 region


    k8s_master=get_hosts(ec2_ob, 'master')   # sending tag like master o find public IP
    k8s_slave=get_hosts(ec2_ob,'slave')


    all_IPs={'master': k8s_master,'slave': k8s_slave }  # converting list into json so that ansible can understand

    print(json.dumps(all_IPs))



if __name__=="__main__":
    main()
```

after above setup of configuration file and dynamic host file we can check connectivity with target node



now its time to create role for master and slave configurations. first create role for k8s master using ansible-galaxy init master. there are many way to install k8s but we are using kubeadm. Kubeadm repo doesnot present bydefault so we have to configure repo for it. after that we have to install kubeadm, docker, iproute-tc. Here kubeadm install with kubelet etc. iproute-tc we have to install for networking. The k8s doesnot support to cgroup driver so we have to change docker conf. file like below or k8s support systemd driver so we have to add systemcd driver in docker conf. file.

```
---
# tasks file for master
- name: "yum configuration for kubeadm installation"
  yum_repository:
   name: Kubernetes
   baseurl: https://packages.cloud.google.com/yum/repos/kubernetes-el7-\$basearch
   enabled: yes
   description: "k8s repo"
   gpgcheck: yes
   repo_gpgcheck: yes
   gpgkey:
   - https://packages.cloud.google.com/yum/doc/yum-key.gpg
   - https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

- name: "installation of kubeadm,docker engine and iproute-tc"
  package:
   name:
   - kubeadm
   - docker
   - iproute-tc
   state: latest
- name: "changing docker default driver cgroup to systemd driver"
  copy:
    content: >
            {
             "exec-opts": ["native.cgroupdriver=systemd"]
            }
    dest: /etc/docker/daemon.json
- name: "start docker service"
  service:
       name: docker
       state: started
       enabled: true
```

after starting docker service we have to change Iptable file and also pull the docker images because all programs of master node run in container like kube-proxy etc. and after that we have to start and enable kubelet program because it is actually agent program that helps master node to connect docker engine.

```
- name: "setting call-iptables to 1"
  shell: "echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables"
  changed_when: false

- name: "pulling k8s master programs"
  command: "kubeadm config images pull"
  changed_when: false

- name: "starting and enabling kubelete service"
  package:
      name: kubelet
      state: started
      enable: true
  ignore_errors: true
  failed_when: false
```

Then we have to run kubeadm init command and it gives the hardware error then we have to use ignore-preflight option. Now our k8s master node setup is done. but if slave node want to connect master node then slave node should know token hence to generate token we need to

execute kubeadm token create --print-join-command. then this command will give tokens. This token we are storing in one file like below

```
- name: "kubernetes setup"
  file:
     path: /root/.kube
     state: directory
- name: "kubernetes setup"
  file:
      path: /root/.kube/config
      state: touch
- name: ""
  command: "cp /etc/kubernetes/admin.conf /root/.kube/config"

  ignore_errors: true
- name: "solving challeng of resources"
  block:
  - command: "kubeadm init"
  rescue:
  - command: "kubeadm init --ignore-preflight-errors=Mem --ignore-preflight-errors=NumCPU"
    ignore_errors: true
  always:
    - command: "kubeadm token create --print-join-command"
      register: out
    - file:
       path: /cred
       state: touch
    - name: "used for templating token"
      template:
            src: master/templates/cred.j2
            dest: /cred.j2
    - debug:
        var: out.stdout
    - name: "fetching token to join slave node"
      fetch:
          src: /cred.j2
          dest: /k8s/
          flat: yes
```

The template file for tokens

```
[root@control_name k8s]# cat master/
defaults/    files/       handlers/    meta/        README.md    tasks/       templates/   tests/       .travis.yml  vars/
[root@control_name k8s]# cat master/t
tasks/     templates/ tests/
[root@control_name k8s]# cat master/templates/cred.j2
{{out.stdout}}
[root@control_name k8s]#
```

lets move towards to slave node configuration. we have to create role for slave using ansible-galaxy-init slave . and we need conatiner engine, kubelet program. also we need to do change in docker file for systemd driver.then we have to start docker and kubelet service then we have to execute token file on slave node.

```
---
# tasks file for master
- name: "yum configuration for kubeadm installation"
  yum_repository:
   name: Kubernetes
   baseurl: https://packages.cloud.google.com/yum/repos/kubernetes-el7-\$basearch
   enabled: yes
   description: "k8s repo"
   gpgcheck: yes
   repo_gpgcheck: yes
   gpgkey:
   - https://packages.cloud.google.com/yum/doc/yum-key.gpg
   - https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

- name: "installation of kubeadm,docker engine and iproute-tc"
  package:
   name:
   - kubeadm
   - docker
   - iproute-tc
   state: latest
- name: "changing docker default driver cgroup to systemd driver"
  copy:
    content: >
            {
             "exec-opts": ["native.cgroupdriver=systemd"]
            }
    dest: /etc/docker/daemon.json
- name: "start docker service"
  service:
       name: docker
       state: started
       enabled: true

- name: "setting call-iptables to 1"
  shell: "echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables"
  changed_when: false

- name: "starting and enabling kubelete service"
  service:
      name: kubelet
      state: started
      enabled: true
  ignore_errors: true
- name: copying file
  copy:
   src: cred.j2
   dest: /cred.sh
- name: "joining slave node to master node"
```

Upto this we have setup infra-structure on AWS using and created roles for configuration of master and slave nodes. we have to run execute that two roles

```
[root@control_name k8s]# cat k8s.yml
- hosts: master
  gather_facts: no
  roles:
          - role: master

- hosts: slave
  gather_facts: no
  roles:
          - role: slave

[root@control_name k8s]# ansible-playbook --syntax-check k8s.yml

playbook: k8s.yml
[root@control_name k8s]#
```

ansible-playbook k8s.yml

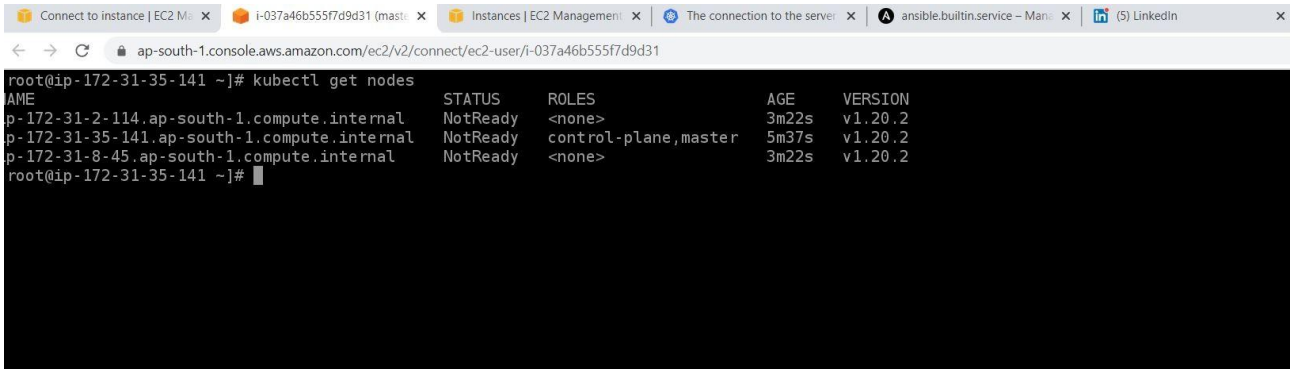
Output:

```
[root@control_name k8s]# ls
ansible.cfg  cred.j2  ec2_provision  host.py  k8s.yml  master  slave  vpc.pem
[root@control_name k8s]# ansible-playbook k8s.yml

PLAY [master] ******************************************************************************************************************************************************

TASK [master : yum configuration for kubeadm installation] *********************************************************************************************************
[WARNING]: Platform linux on host 65.1.94.12 is using the discovered Python interpreter at /usr/bin/python, but future installation of anothe
of that path. See https://docs.ansible.com/ansible/2.10/reference_appendices/interpreter_discovery.html for more information.
changed: [65.1.94.12]

TASK [master : installation of kubeadm,docker engine and iproute-tc] ************************************************************************************************
changed: [65.1.94.12]

TASK [master : changing docker default driver cgroup to systemd driver] *********************************************************************************************
changed: [65.1.94.12]

TASK [master : start docker service] ********************************************************************************************************************************
changed: [65.1.94.12]

TASK [master : setting call-iptables to 1] **************************************************************************************************************************
ok: [65.1.94.12]

TASK [pulling k8s master programs] **********************************************************************************************************************************
ok: [65.1.94.12]

TASK [master : starting and enabling kubelete service] **************************************************************************************************************
ok: [65.1.94.12]
```

```
TASK [master : kubernetes setup] ************************************************************************************************************************************
changed: [65.0.3.74]

TASK [master : command] *********************************************************************************************************************************************
fatal: [65.0.3.74]: FAILED! => {"changed": true, "cmd": ["cp", "/etc/kubernetes/admin.conf", "/root/.kube/"], "delta": "0:00:00.042613", "end": "2021-02-06 07:54:01.197376", "msg": "non-zer
o return code", "rc": 1, "start": "2021-02-06 07:54:01.154763", "stderr": "cp: cannot stat '/etc/kubernetes/admin.conf': No such file or directory", "stderr_lines": ["cp: cannot stat '/etc/
kubernetes/admin.conf': No such file or directory"], "stdout": "", "stdout_lines": []}
...ignoring

TASK [master : command] *********************************************************************************************************************************************
fatal: [65.0.3.74]: FAILED! => {"changed": true, "cmd": ["kubeadm", "init"], "delta": "0:00:02.048604", "end": "2021-02-06 07:54:09.570037", "msg": "non-zero return code", "rc": 1, "start":
 "2021-02-06 07:54:07.521433", "stderr": "\t[WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'\nerror execution phase preflight: [prefl
ight] Some fatal errors occurred:\n\t[ERROR NumCPU]: the number of available CPUs 1 is less than the required 2\n\t[ERROR Mem]: the system RAM (983 MB) is less than the minimum 1700 MB\n[pr
eflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`\nTo see the stack trace of this error execute with --v=5 or higher", "stderr_lin
es": ["\t[WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'", "error execution phase preflight: [preflight] Some fatal errors occurred:
", "\t[ERROR NumCPU]: the number of available CPUs 1 is less than the required 2", "\t[ERROR Mem]: the system RAM (983 MB) is less than the minimum 1700 MB", "[preflight] If you know what y
ou are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`", "To see the stack trace of this error execute with --v=5 or higher"], "stdout": "[init] Using Kubernetes
version: v1.20.2\n[preflight] Running pre-flight checks", "stdout_lines": ["[init] Using Kubernetes version: v1.20.2", "[preflight] Running pre-flight checks"]}

TASK [master : command] *********************************************************************************************************************************************
changed: [65.0.3.74]

TASK [master : ceating tokens] **************************************************************************************************************************************
changed: [65.0.3.74]

TASK [master : file] ************************************************************************************************************************************************
changed: [65.0.3.74]

TASK [master : used for templating token] ****************************************************************************************************************************
changed: [65.0.3.74]

TASK [master : debug] ***********************************************************************************************************************************************
ok: [65.0.3.74] => {
    "out.stdout": "kubeadm join 172.31.35.141:6443 --token ki12n1.xmztzy4ky3ttag4d     --discovery-token-ca-cert-hash sha256:58510be3ec6510ea2c58aa077657b56de8224e776d3e7c42adc1aa25fbe0cd8a
"
}
```

```
TASK [master : fetching token to join slave node] ********************************************************************************************************************
changed: [65.0.3.74]

PLAY [slave] ********************************************************************************************************************************************************

TASK [slave : yum configuration for kubeadm installation] ***********************************************************************************************************
[WARNING]: Platform linux on host 15.206.165.34 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python
meaning of that path. See https://docs.ansible.com/ansible/2.10/reference_appendices/interpreter_discovery.html for more information.
changed: [15.206.165.34]
[WARNING]: Platform linux on host 52.66.250.181 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python
meaning of that path. See https://docs.ansible.com/ansible/2.10/reference_appendices/interpreter_discovery.html for more information.
changed: [52.66.250.181]

TASK [slave : installation of kubeadm,docker engine and iproute-tc] **************************************************************************************************
changed: [15.206.165.34]
changed: [52.66.250.181]

TASK [slave : changing docker default driver cgroup to systemd driver] ***********************************************************************************************
changed: [15.206.165.34]
changed: [52.66.250.181]

TASK [slave : start docker service] **********************************************************************************************************************************
changed: [15.206.165.34]
changed: [52.66.250.181]

TASK [slave : setting call-iptables to 1] ****************************************************************************************************************************
ok: [15.206.165.34]
ok: [52.66.250.181]
```

```
TASK [slave : starting and enabling kubelete service] ****************************************************************************************************************
changed: [52.66.250.181]
changed: [15.206.165.34]

TASK [slave : copying file] ******************************************************************************************************************************************
changed: [52.66.250.181]
changed: [15.206.165.34]

TASK [joining slave node to master node] *****************************************************************************************************************************
changed: [52.66.250.181]
changed: [15.206.165.34]

PLAY RECAP ***********************************************************************************************************************************************************
15.206.165.34              : ok=8    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
52.66.250.181              : ok=8    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
65.0.3.74                  : ok=15   changed=11   unreachable=0    failed=0    skipped=0    rescued=1    ignored=1
```



Then we have to set funnel CNI network after that lets launch multitier application means wordpress and mysql. first we have create ansible role then we need to put yaml files of frondend and backend in files folder then we have write task.

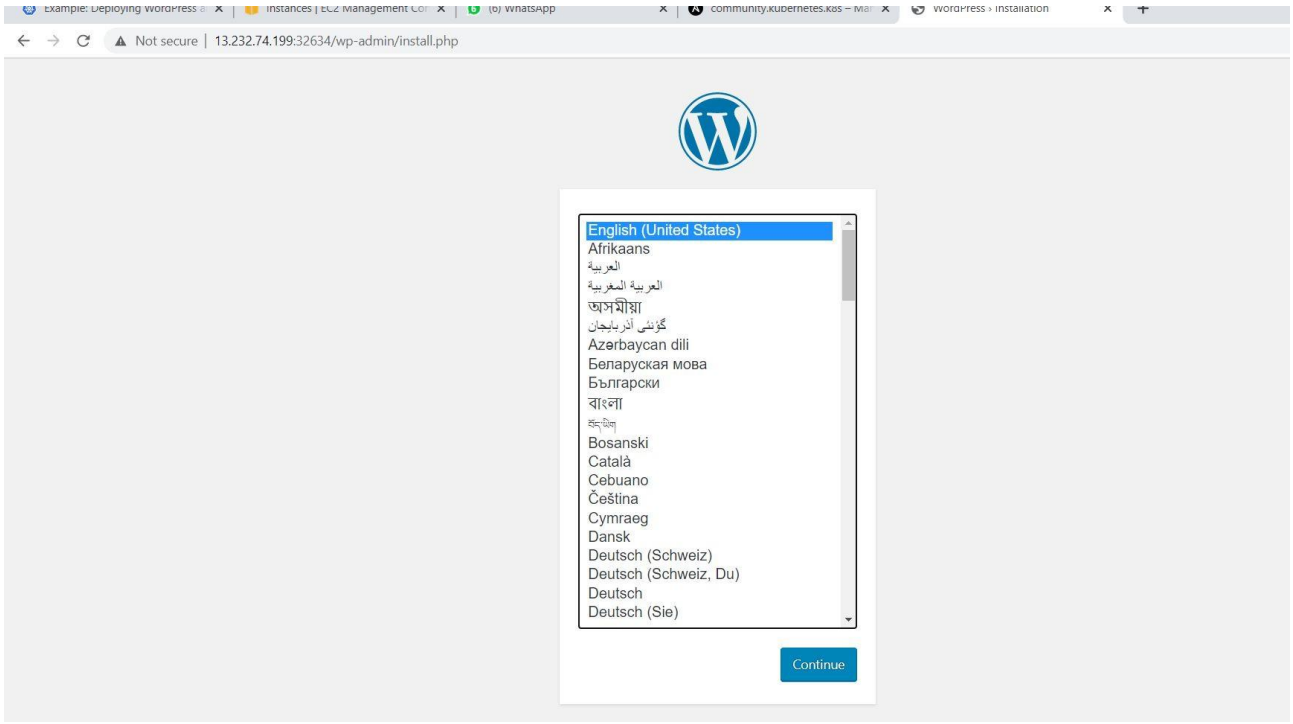
```
[root@control_name k8s]# ls
ansible.cfg  cred.j2  ec2_provision  host.py  k8s.yml  master  multi_tier  mult-tier.yml  slave  vpc.pem
[root@control_name k8s]# tree multi_tier/
multi_tier/
├── defaults
│   └── main.yml
├── files
│   ├── db.yml
│   └── wordpress.yml
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

8 directories, 10 files
[root@control_name k8s]# cat multi_tier/tasks/main.yml
---
# tasks file for multi_tier

- name: copying file
  copy:
    src: db.yml
    dest: /db.yml
- name: copying file
  copy:
    src: wordpress.yml
    dest: /wordpress.yml
- name: Create a Deployment by reading the definition from a local file
  shell: kubectl apply -f /db.yml
  ignore_errors: true


- name: Create a Deployment by reading the definition from a local file
  shell: kubectl apply -f /wordpress.yml
  ignore_errors: true
[root@control_name k8s]#
```

```
[root@control_name k8s]# cat multi_tier/files/
db.yml         .db.yml.swp    wordpress.yml
[root@control_name k8s]# cat multi_tier/files/db.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql
  labels:
    app: db
spec:
  selector:
    matchLabels:
      app: db
  template:
    metadata:
      labels:
        app: db
    spec:
      containers:
      - image: mysql:5.6
        name: mysql1
        env:
        - name: MYSQL_ROOT_PASSWORD
          value: passwd
        - name: MYSQL_DATABASE
          value: db
        ports:
        - containerPort: 3306
          name: mysql

[root@control_name k8s]#
```

```
[root@control_name k8s]# cat multi_tier/files/wordpress.yml
apiVersion: v1
kind: Service
metadata:
  name: wordpress
  labels:
    app: word
spec:
  ports:
    - port: 80
  selector:
    app: wordp
  type: NodePort
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress
  labels:
    app: wordp
spec:
  selector:
    matchLabels:
      app: wordp
  template:
    metadata:
      labels:
        app: wordp
    spec:
      containers:
      - image: wordpress:4.8-apache
        name: wordpress
        env:
         - name: WORDPRESS_DB_HOST
           value:  10.244.2.3

         - name: WORDPRESS_DB_USER
           value: root
         - name: WORDPRESS_DB_PASSWORD
           value: passwd
         - name: WORDPRESS_DB_NAME
           value: db

[root@control_name k8s]#
```

```
[root@control_name k8s]# ansible-playbook multi
multi_tier/     multi-tier.yml
[root@control_name k8s]# ansible-playbook multi-tier.yml

PLAY [master] ******************************************************************************************************************************************************

TASK [multi_tier : copying file] ***********************************************************************************************************************************
[WARNING]: Platform linux on host 13.232.74.199 is using the discovered Python interpreter at /usr/bin/python3.7, but future installation of another Python in
meaning of that path. See https://docs.ansible.com/ansible/2.10/reference_appendices/interpreter_discovery.html for more information.
ok: [13.232.74.199]

TASK [multi_tier : copying file] ***********************************************************************************************************************************
ok: [13.232.74.199]

TASK [multi_tier : Create a Deployment by reading the definition from a local file] ********************************************************************************
changed: [13.232.74.199]

TASK [multi_tier : Create a Deployment by reading the definition from a local file] ********************************************************************************
changed: [13.232.74.199]

PLAY RECAP *********************************************************************************************************************************************************
13.232.74.199              : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@control_name k8s]#
```

finally after completion of playbook you will get but here we need to add service port and public IP of instance.



Thank you !!

Thus I have successfully completed task 5.