

Exp 2: -Kubernetes

Steps: -

- 1) Create an Instance of Ubuntu on AWS
- 2) Connect to the Instance

Commands: -

1. `sudo su`(Super User)
2. `apt-get update && apt-get install -y apt-transport-https`
3. `curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -`
4. `sudo vim /etc/apt/sources.list.d/kubernetes.list`
After that a VI editor is opened....Add this in editor
`deb http://apt.kubernetes.io/ kubernetes-xenial main`

Save and exit the editor

5. `apt-get update`
6. `apt-get install docker.io kubeadm kubectl kubelet kubernetes-cni`
7. `kubeadm init '--ignore-preflight-errors=all'`
OR
`kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all`

It will generate output like below

```
mkdir -p $HOME/.kube  
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

8. `export KUBECONFIG=/etc/kubernetes/admin.conf`
9. `kubectl get nodes`
10. `kubectl get pods --all-namespaces`

Now Create a Pod using command `touch pod.yml`

And add this code:-

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: sample-pod  
  labels:
```

```
    zone: prod
    version: v1
spec:
  containers:
  - name: sample-ctr
    image: devopstrainer/deploy:v1
    ports:
    - containerPort: 80
```

Save and exit the editor:-

1. `kubectl create -f pod.yml`
2. `kubectl get nodes`
3. `kubectl get pods --all-namespaces`

Now update the code on pod.yml

Replace the code with:-

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: rss-site
  labels:
    app: web
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
      - name: front-end
        image: nginx
        ports:
        - containerPort: 80
```

Now type commands:-

1. `kubectl create -f pod.yml`
2. `kubectl describe deploy rss-site`

Exp 3: -Terraform Installation

Steps: -

(Use Fedora 38 for this)

Download this file: -

https://releases.hashicorp.com/terraform/1.0.5/terraform_1.0.5_linux_amd64.zip

Commands: -

1. unzip terraform_1.0.5_linux_amd64.zip (**Unzip the file**)
2. sudo mv terraform /usr/local/bin/
3. which terraform
4. terraform -v

Exp 4: -Terraform Operations

After installing Terraform we have to use 4 commands: -

1. terraform init
2. terraform plan
3. terraform apply
4. terraform destroy

But before that create 3 files: -

1. **main.tf** (In this file change the region to(ap-south-1), access key and secret key, key-name)

For the keys create a user on AWS: -

While creating User “**Don’t give console access**” and attach policy “**AdminFullAccess**”

Generate access key: -

IAM->Users->Username->Security credentials->Create access key
You will get 2 keys: -access and secret key

2. **InstallJenkins.sh** (No change)
3. **Variables.tf** (Change the “ami-id”)

Create these files in a single folder and open the terminal on that folder .

Now run those 4 commands.

Exp 5: -SonarQube

Steps: -

1. Download the SonarQube Community Edition zip file.
2. Unzip the file
3. Download Sonnar Scanner
4. Unzip the file
5. Open the terminal in -> sonarqube-9.0.1.46107/bin/linux-x86-64
6. Run the command: ./sonar.sh console
7. If JDk problem occurs use the command:
8. sudo dnf install java-latest-openjdk.x86_64
9. After that execute step 6.
10. It will start running on url <http://127.0.0.1:9000>

Exp 6: -Nagios Installation

Steps: -

Commands: -

1. dnf install httpd httpd-tools php gcc glibc glibc-common gd gd-devel make net-snmp openssl-devel -y
2. useradd nagios
3. usermod -G nagios nagios
4. usermod -G nagios apache
5. mkdir /root/nagios
6. cd /root/nagios
7. wget <https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.9.tar.gz>
8. wget <https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz>
9. tar -xf nagios-4.4.9.tar.gz
10. tar -xf nagios-plugins-2.3.3.tar.gz
11. cd nagios-4.4.9/
12. ./configure --with-command-group=nagios
13. make all
14. make install
15. make install-init
16. make install-commandmode

```
17. make install-config
18. make install-webconf
19. htpasswd -s -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
    New password:
    Re-type new password:
    Adding password for user nagiosadmin
20. systemctl restart httpd
21. cd /root/nagios/nagios-plugins-2.3.3
22. ./configure --with-nagios-user=nagios --with-nagios-group=nagios
23. make
24. make install
25. /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
26. systemctl enable nagios
27. systemctl enable httpd
28. systemctl restart nagios
29. systemctl status nagios
```

Exp 7: -Nagios for service monitoring

Steps: -

1. Open Nagios on port ...
2. Open Service Status Totals
3. Start SSH Service by commands: -
4. Systemctl start sshd
5. Systemctl stop sshd
6. Systemctl status sshd