

Setup WordPress Application Server with PostgreSQL Database Engine and pgAdmin Management Interface

-Ec2 instance for Jenkins.

```
sudo apt update
```

```
sudo apt install openjdk-21-jdk -y
```

```
sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

```
echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
sudo apt-get update
```

```
sudo apt-get install jenkins -y
```

-Give URL,

```
http://abc.xyz
```

```
=====
```

-Install Docker on Ec2 where Jenkins is installed.

-Install Docker-

```
sudo apt update
```

```
sudo apt install -y docker.io
```

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

-Gives Jenkins permission to run Docker commands.

```
sudo usermod -aG docker jenkins
```

-Also add your default ubuntu user (optional but useful):

```
sudo usermod -aG docker ubuntu
```

Restart Docker and Jenkins Services-

```
sudo systemctl restart docker
```

```
sudo systemctl restart jenkins
```

-Verify Docker Works Inside Jenkins-

Open Jenkins UI → Manage Jenkins → Script Console

-Run this Groovy script:

```
def proc = 'docker --version'.execute()
proc.waitFor()
println proc.text
```

-This will return-

Docker version 24.0.7, build afdd53b

-Account created on Dockerhub.

Docker hub user-

username- vmaurya2008

-Create dockerhub token to creter credentials in Jenkins-

token name- jenkins-docker

Valid -30days

permission -read/write

token password - give your token

id - dockerhub-creds

-Created public repository for push image-

vmaurya2008/wordpress

-Prerequisites for setup Kops.

-Ec2 instances

-Domain for Kubernetes DNS records.

- I used GoDaddy

-Kops setup on Ec2 required.

1-Install kubectl

- follow the url --

<https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/#install-kubectl-binary-with-curl-on-linux>.

- Download the latest release: `curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"`

- `sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl`

- Confirm by command -- `kubectl version --client`

2-Install Kops

1- follow the Url -- https://kops.sigs.k8s.io/getting_started/install/

2- Create kops cluster by command

- `kops create cluster --name=wordpress.vmauryaaws.xyz`

`--state=s3://kopsstate569 --zones=us-east-1a,us-east-1b --node-count=2`

`--node-size=t3.small --control-plane-size=t3.medium --dns-zone=wordpress.vmauryaaws.xyz`

`--node-volume-size=12 --control-plane-volume-size=12 --ssh-public-key`

`~/ssh/id_ed25519.pub`

3- Apply all configuration by this command.

- `kops update cluster --name=wordpress.vmauryaaws.xyz`

`--state=s3://kopsstate569 --yes --admin`

- 4- `kops update cluster --name=wordpress.vmauryaaws.xyz --state=s3://kopsstate569 --yes --admin`
- 5- `kops validate cluster --name=wordpress.vmauryaaws.xyz --state=s3://kopsstate569`
- 6- For deleting cluster via kops: `kops delete cluster --name=wordpress.vmauryaaws.xyz --state=s3://kopsstate569 --yes`
- 4- Create hosted zone in Route53 where we have all the records.
- S3 bucket type (general purpose) store the information of this cluster.

3-ssh key

-CMD

`ssh-keygen -- enter -- enter -- enter`

4-awscli

- Login to AWS account and setup
 - S3 bucket
 - IAM User for AWSCli
 - Create IAM user (xyz) --> attach policy (AdministratorAccess) --> Generate Access keys (CLI) --> Keep is safe.
 - Configure AWSCLI
 - CMD
 - `snap install aws-cli --classic`
 - `aws configure`
 - Route53 Hosted Zone
- Go to Route53 and create hosted zone name (wordpress.vmauryaaws.xyz) and type (public hosted zone).
- This hosted zone gives you four DNS servers record.
- Login to Domain Registor(GoDaddy) and Creates DNS records type (NS) with name (wordpress) pointing to Route53 hosted zone NS server

- Run Jenkins pipeline

For communicating jenkins with kubernetes, we need to copy /kube/config file to the jenkins ec2 server.

-On Kops Ec2 instance-

`kubectl config view --flatten --minify > kubeconfig-jenkins.yaml`

-Copy this file on jenkins ec2

`mkdir -p ~/.kube`

`mv ~/kubeconfig-jenkins.yaml ~/.kube/config`

`chmod 600 ~/.kube/config`

- Upload same file into Jenkins

Go to: Jenkins → Manage Jenkins → Credentials → Global

Add → Secret File

ID: kubeconfig

Upload: kubeconfig-jenkins.yaml

AND run pipeline