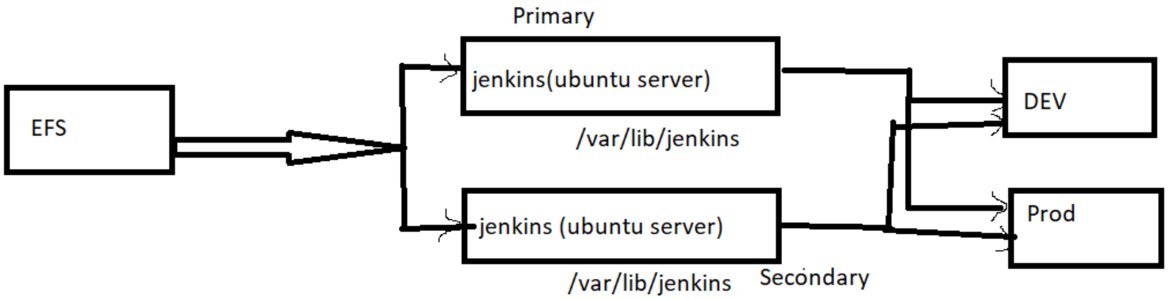
Jenkins High-Availability using AWS EFS:

Flow Diagram:



Step1: To make Jenkins high availability we can use AWS EFS (Elastic File System) to replicate data in /var/lib/Jenkins from Jenkins primary to Jenkins secondary.

Launch 2 instances in EC2 by selecting ubuntu O.S and Security group if it is custom open 22,443,8080 and nfs port (2049) and select availability Zones (one for one Jenkins and one for second Jenkins).

Step2: Access instances using putty and go to root user and make dir for Jenkins and install nfs package in both servers

[]sudo su

[root@/home/ubuntu or root] sudo apt-get update

[root] **sudo apt install nfs-kernel-server**

[root]**mkdir /var/lib/Jenkins**

Now give jenkins install cmmnds upto java installation, don’t install Jenkins

```
[root]sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

Then add a Jenkins apt repository entry:

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

Update your local package index, then finally install Jenkins:

```
[root]  sudo apt-get update
```

```
[root] sudo apt-get install fontconfig openjdk-17-jre
```

Step3: Now go to jenkins folder in Primary and create file for checking purpose

```
[] touch f1
```

```
[]ls
```

F1 and now check Jenkins folder in secondary. no files are there. Delete f1.

Create **EFS in AWS** by Selecting Our VPC which we gave for 2 instances

Click on created EFS > GO to Networks and and once we get available their change Security Group and if we want give Pvt i/p’s our own i/ps and click save

Step4: Now Click on EFS Attach and **Select Mount via DNS , then CP NFS client**

[]df –Th (check there is no nfs mount)

[] sudo mount –t nfts –o safsfsafsfafaffafa-eeuiee-dd-com:/ **/var/lib/jenkins**

Inplace of /efs we should keep /var/lib/jenkins , because we need to mount that folder.

```
[]df –Th
```

```
Fndsfkwndksandkdnsadnma.com/ nfs mount
```

We got mount successful . Do the same in second server also and make permanent mount

```
[]vi /etc/fstab/
```

```
Fdsfdsfdsfdsfdsfsdf.com /var/lib/jenkins nfs defaults 0 0
```

```
:wq!
```

Do same in 2nd server…

Now mounted perminently… we can check by restarting server also…

Now if we create or delete files folders in any server will replicate in other server.

Step5: install jenkins in first primary server now,

[] apt install jenkins

Now if we give ip:8080 in dashboard we get login dashboard .

[] Cat /var/lib/jenkins/secrets/initialAdminPassword

Fffewfewfewfewfefewfefdewfdewde

If we see second server, we can see files in jenkins folder, if we give cat /var/lib/jenkins/secrets/initialAdminPassword   we get same password, because of EFS, now install Jenkins in second server also n, because Jenkins installation happens in diff path

Give second server i/p and port

Give same psswd in both dashboards and skin create and login as admin....

Now login sucessful , if we face any issue while creating job, restart jenkins in both servers.

Now creat job1 in Jenkins primary dashoard, and build it, once build completes, if we see in Jenkins folder in server, we see workspace dir in primary and secondary as well.

So now if we refresh secondary jenk dashboard we can't see job created in Jenkins server

Restart Jenkins Dashboard or restart Jenkins in server, now we got job1 created in jenk primary with build history, we can build same job here, we get build2 and if we want this change to reflect in Jenkins server1, restart Jenkins..

Now stop Jenkins second server, that data replicates sucessfully in stop state, once we start instance, we can see all updated data,,

So, if we get server issue in Jenkins primary, we simply stop jenk instance1 and we start second one.

And in Godaddy (domain service we change i/p of jenk1 to jenk2)

So, developers use the same domain, and they get the same dashboard.