

pOC on Jenkins HA

Document for Jenkins High Availability



April 15, 2019

Dark MAtter

**Index**

[**Jenkins High Availability** 2](#_Toc6218061)

[High Availability 2](#_Toc6218062)

[Jenkins High Availability 2](#_Toc6218063)

[High Availability Cluster Configurations 3](#_Toc6218064)

[HAProxy setup 3](#_Toc6218065)

[**Running Jenkins Behind HAProxy** 4](#_Toc6218066)

[**Jenkins Master Primary Server Setup** 5](#_Toc6218067)

[Jenkins Secondary Replication Server Setup 6](#_Toc6218068)

[**NFS Share Setup** 7](#_Toc6218069)

[Configuring Jenkins Master Primary and Secondary for NFS mount point. 8](#_Toc6218070)

# **Jenkins High Availability**

## High Availability

HA is a characteristic of a system design that minimizes the downtime and maximizes the time the system is available. It can be achieved by following possibilities.

1. Implement multiple application servers.
2. Scaling and slaves matters.
3. Spread out physically.
4. Maintain a recurring online backup system along with hardware.
5. Use of a virtualized server for zero-downtime recovery.

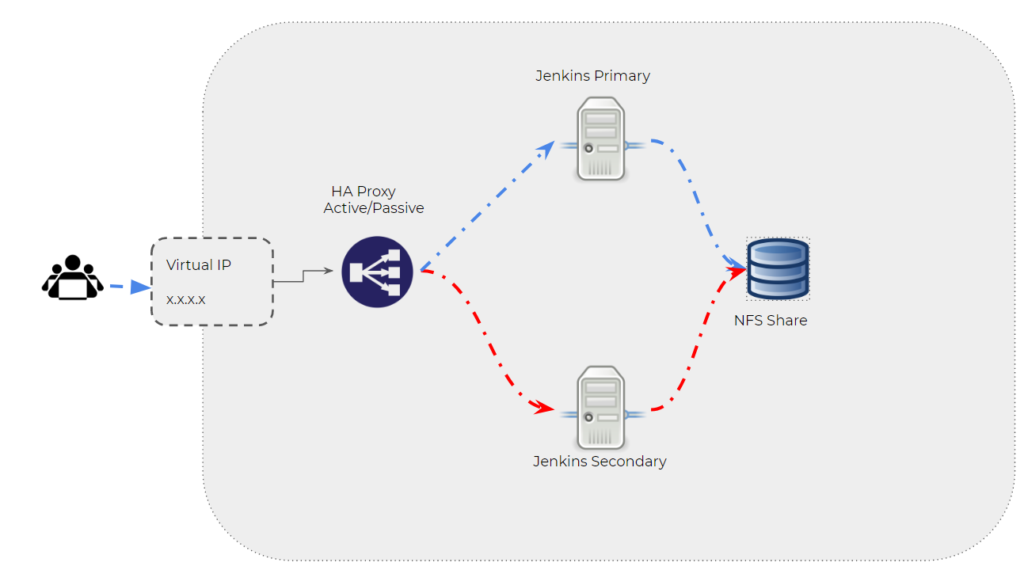
## Jenkins High Availability

Jenkins High Availability used to overcome the downtime caused by any of these two situations.

* Reaching load capacity from an instance running some services
* Having an instance or service malfunction or stop running completely

Introducing redundancy to the equation is a common approach to address those cases. It is done by adding instances to the system that have the same capabilities as the existing one. The result is usually referred to as a cluster.

Jenkins HA with NFS architecture diagram shown below.



From the above architecture diagram components involved are

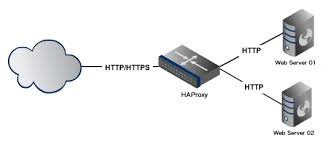
* HAProxy server
* Jenkins Primary (Master Server)
* Jenkins Secondary (Replication Server)
* NFS share

In Jenkins HA architecture with NFS (Network File System), will require four servers out of which, two servers will be served as Jenkins Master Primary & Jenkins Mater Secondary. One HAProxy server which will be load balancing between Jenkins Master primary and secondary. NFS server plays an important role, which mounts the shared file system on each of the Jenkins masters on the location of Jenkins home directory, this will keep the both nodes in sync, and all the Jenkins Master Primary configuration files will be in Jenkins Master Secondary. Now, whenever any job runs in Jenkins Master primary, will be replicated in Jenkins secondary.

## High Availability Cluster Configurations

### HAProxy setup

HAProxy, which stands for High Availability Proxy, is a popular open source software TCP/HTTP Load Balancer and proxying solutions. Its most common use is to improve the performance and reliability of a server environment by distributing the workload across multiple servers (e.g. web, application, and database). Below is the diagram, which gives an idea how HAProxy works.



# **Running Jenkins Behind HAProxy**

Running Jenkins Primary server behind the layer of load balancer.

Below are the following steps to configure HAProxy for Jenkins Master primary.

* Install HAProxy in Linux server

# yum install haproxy –y //install ha proxy

* Important configuration files.

# /etc/haproxy/haproxy.conf

* Configurations

To setup Load Balancing for Jenkins Master follow the steps below.

Update the Jenkins Master server details with port in /etc/haproxy/haproxy.conf as below.

defaults

mode http

option http-server-close

timeout client 20s

timeout server 20s

timeout connect 4s

frontend ft\_app

bind <haproxy server IP>:80 name app

default\_backend bk\_app

backend bk\_app

server s1 <Jenkins primary IP>:8080 check

server s2 <Jenkins secomdary IP>:8080 check backup

Save and quit the file.

* Start haproxy service

# systemctl start haproxy //start haproxy service

# systemctl status haproxy //verify haproxy is active running

Now, HAProxy is installed and configured

# **Jenkins Master Primary Server Setup**

Installing standalone Jenkins in Linux (RHEL 7), follow the steps below.

* Installing Java

# yum install java-1.8.0-openjdk-deve1 –y //install java

* Installing Jenkins

# curl –silent –location <http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo> | sudo tee /etc/yum.repos.d/jenkins.repo-ci.org.key //download jenkins repo

# sudo rpm –import <https://jenkins-ci.org/redhat/jenkins-ci.org.key> //import jenkins keys

# yum install jenkins –y //install jenkins

# systemctl start jenkins //start the Jenkins service

# systemctl status Jenkins //check the status of Jenkins

# sudo systemctl enable Jenkins //enable the Jenkins service

* Adjust the Firewall

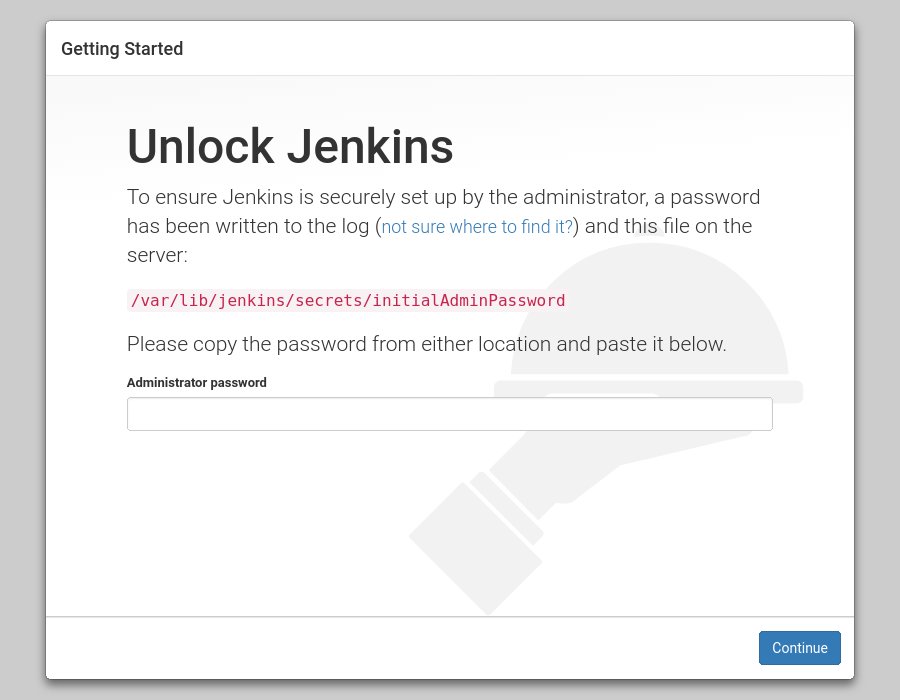
If Jenkins is installed on remote server which is protected by firewall, need to port to 8080

# firewall-cmd --permanent --zone=public --add-port=8080/tcp

//to add port 8080

# firewalld-cmd –reload //update firewalld

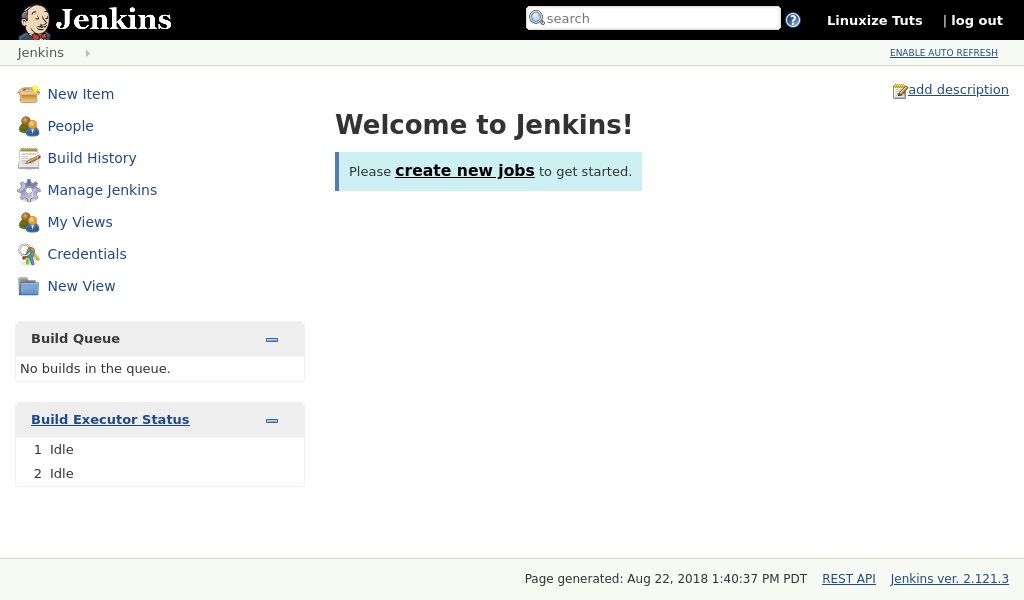
* Setting up Jenkins
  + In browser give the IP address followed by port 8080
  + <http://your_ip_address:8080>
  + Screen similar to the following will appear, prompting to enter the Administrator password that is created during the installation.



# cat /var/lib/jenkins/secrets/initialAdminPassword

//to see 32-character long alphanumeric password

* + Copy the password and paste it into the Administrator password field and click continue.
  + Click on the Install suggested plugins box, and the installation process will start immediately.
  + Once the installation is complete, it will be prompted to set up the first administrative user. Fill out all required information and click Save and Continue.
  + On the next page, it will be asked to set the URL for the Jenkins instance. The URL field will be populated with an automatically generated URL.
  + To complete the setup confirm the URL by clicking on the Save and Finish button



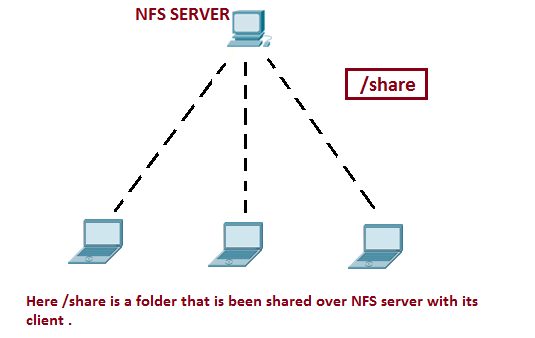
Finally, click on the Start using Jenkins button and you will be redirected to the Jenkins dashboard logged in as the admin user you have created in one of the previous steps.

## Jenkins Secondary Replication Server Setup

Follow the same steps, which were used to configure and install the Jenkins Master Primary server.

# **NFS Share Setup**

Network File System is a client/server application that allows all network users to access shared files stored on computers of different types. NFS provides access to shared files through an interface called the Virtual File System (VFS) that runs on top of TCP/IP. Below is the diagram, which shows NFS share server and client server.



Sharing files through NFS is simple and involves two steps:

On the Linux system that runs the NFS server, export (share) one or more directories by listing them in the /etc/exports file and by running the exportfs command. In addition, start the NFS server.

On each client system, use the mount command to mount the directories, which the server exported.

* installing NFS server

# yum install nfs-utils nfs-utils-lib //installing required NFS packages

# systemctl enable nfs-server.service //enable the NFS service

# systemctl start nfs-server.service // start the NFS service

* Important files for NFS configuration

/etc/exports: It is a main configuration file of NFS, all exported files and directories are defined in this file at the NFS Server end.

/etc/fstab: To mount a NFS directory on your system across the reboots, we need to make an entry in /etc/fstab.

* Configuring NFS server

Make a directory for mount point

# mkdir /jenkins\_home //stoage directory to mount with Jenkins Master Primary

Update the Jenkins primary and secondary ip address in /etc/export file as follows

/jenkins\_home <I.P>(rw,sync,no\_root\_squash) //Jenkins Primary

/Jenkins\_home <I.P>(rw,sync,no\_root\_squash) //Jenkins Secondary

## Configuring Jenkins Master Primary and Secondary for NFS mount point.

Update the mount point in /etc/fstab file as follow in both Jenkins Nodes

<NFS I.P>:/jenkins\_home /var/lib/jenkins\_home nfs defaults 0 0 //update in jenkins Primary server

Save and exit the file

# mount –a //mount all drives updated in /etc/fstab

# mount //to verify the mount point

NFS server is now set, if any jobs running in Jenkins Primary will be in sync with NFS shared directory and Jenkins Secondary.

Finally, Jenkins HA setup is completed.

**Note:**

1. All the data (Jenkins configurations) is stored in NFS shared directory.
2. In case of fail over HAProxy server will automatically redirect to Jenkins Secondary server.
3. Need to reload the configurations from the disk to synchronize the data from the disc in secondary server to reflect all the configurations from disc in Jenkins console.