**DevOps Tools Installation, and Configuration**

**--Gitlab Installation and configuration**

**1. Install and configure the necessary dependencies**

If you install Postfix to send email please select 'Internet Site' during setup. Instead of using Postfix you can also use Sendmail or configure a custom SMTP server and configure it as an SMTP server.

On Centos 6 and 7, the commands below will also open HTTP and SSH access in the system firewall.

#sudo yum install curl policycoreutils openssh-server openssh-clients

#sudo systemctl enable sshd

#sudo systemctl start sshd

#sudo yum install postfix

#sudo systemctl enable postfix

#sudo systemctl start postfix

#sudo firewall-cmd --permanent --add-service=http

#sudo systemctl reload firewalld

**2. Add the GitLab package server and install the package**

#curl -sS https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.rpm.sh | sudo bash

#sudo yum install gitlab-ce

OR

Install manually download the package from (https://packages.gitlab.com/gitlab/gitlab-ce) and install with below commands

#curl -LJO https://packages.gitlab.com/gitlab/gitlab-ce/packages/el/7/gitlab-ce-XXX.rpm/download

#rpm -i gitlab-ce-XXX.rpm

OR

#curl -s https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.rpm.sh | sudo bash

#sudo yum install gitlab-ce-8.13.0-ce.0.el7.x86\_64

**3. Configure and start GitLab**

#sudo gitlab-ctl reconfigure

**4. Browse to the hostname and login**

On your first visit, you'll be redirected to a password reset screen to provide the password for the initial administrator account. Enter your desired password and you'll be redirected back to the login screen.

The default account's username is root. Provide the password you created earlier and login. After login you can change the username if you wish.

**--Jenkins Installation and configuration**

**Step 1: Update your CentOS 7 system**

#sudo yum install epel-release

#sudo yum update

#sudo reboot

When the reboot finishes, login with the same sudo user.

**Step 2: Install Java**

Before you can install Jenkins, you need to setup a Java virtual machine on your system. Here, let's install the latest OpenJDK Runtime Environment 1.8.0 using YUM:

#sudo yum install java-1.8.0-openjdk.x86\_64

After the installation, you can confirm it by running the following command:

#java -version

In order to help Java-based applications locate the Java virtual machine properly, you need to set two environment variables: "JAVA\_HOME" and "JRE\_HOME".

#sudo cp /etc/profile /etc/profile\_backup  
#echo 'export JAVA\_HOME=/usr/lib/jvm/jre-1.8.0-openjdk' | sudo tee -a /etc/profile

#echo 'export JRE\_HOME=/usr/lib/jvm/jre' | sudo tee -a /etc/profile

#source /etc/profile

Finally, you can print them for review:

#echo $JAVA\_HOME

#echo $JRE\_HOME

**Step 3: Install Jenkins**

Use the official YUM repo to install the latest stable version of Jenkins, which is 1.651.2 at the time of writing:

#cd ~

#sudo wget -O /etc/yum.repos.d/jenkins.repo <http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo>

#sudo rpm --import <http://pkg.jenkins-ci.org/redhat-stable/jenkins-ci.org.key>

#sudo yum install jenkins

Start the Jenkins service and set it to run at boot time:

#sudo systemctl start jenkins.service

#sudo systemctl enable jenkins.service

In order to allow visitors access to Jenkins, you need to allow inbound traffic on port 8080:

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

#sudo firewall-cmd –reload

Now, test Jenkins by visiting the following address from your web browser:

http://<your-Vultr-server-IP>:8080

**Step 4: Install Nginx (optional)**

In order to facilitate visitors' access to Jenkins, you can setup an Nginx reverse proxy for Jenkins, so visitors will no longer need to key in the port number 8080 when accessing your Jenkins application.

Install Nginx using YUM:

#sudo yum install nginx

Modify the configuration of Nginx:

#sudo vi /etc/nginx/nginx.conf

Find the two lines below:

location / {

}

Insert the six lines below into the { } segment:

proxy\_pass http://127.0.0.1:8080;

proxy\_redirect off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

The final result should be:

location / {

proxy\_pass http://127.0.0.1:8080;

proxy\_redirect off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

Save and quit:

:wq

Start and enable the Nginx service:

#sudo systemctl start nginx.service

#sudo systemctl enable nginx.service

Allow traffic on port 80:

#sudo firewall-cmd --zone=public --permanent –add-service=http

#sudo firewall-cmd –reload

Finally, visit the following address from your web browser to confirm your installation:

http://<your-Vultr-server-IP>

**--Nexus 3 Installation and configuration**

**Step 1: Login to your Linux server and update it.**

#sudo yum update -y

**Step 2: Install OpenJDK 1.8**

#sudo yum install java-1.8.0-openjdk.x86\_64

**Step 3: Create a directory named app and cd into the directory.**

#sudo mkdir /app && cd /app

**Step 4: Download the latest nexus. You can get the latest download links fo for nexus from here. Here I am downloading nexus 3.**

#sudo wget <https://sonatype-download.global.ssl.fastly.net/nexus/3/nexus-3.0.2-02-unix.tar.gz>

Untar the downloaded file.

#sudo tar -xvf nexus-3.0.2-02-unix.tar.gz

Rename the untared file to nexus.

#sudo mv nexus-3.0.2-02 nexus

**Step 5: As a good security practice, it is not advised to run nexus service with any sudo user. So create a new user named nexus.**

#sudo adduser nexus

Change the ownership of nexus file to nexus user.

#sudo chown -R nexus:nexus /app/nexus

Open /app/nexus/bin/nexus.rc file, uncomment run\_as\_user parameter and set it as following.

run\_as\_user="nexus"

**Step 6: If you want to change the default nexus data directory, open nexus properties file and change the data directory “-Dkaraf.data” parameter to a preferred location as shown below.**

#sudo vi /app/nexus/bin/nexus.vmoptions

An example configuration is shown below.

-Xms1200M

-Xmx1200M

-XX:+UnlockDiagnosticVMOptions

-XX:+UnsyncloadClass

-Djava.net.preferIPv4Stack=truer

-Dkaraf.home=.

-Dkaraf.base=.

-Dkaraf.etc=etc

-Djava.util.logging.config.file=etc/java.util.logging.properties

-Dkaraf.data=/nexus/nexus-data

-Djava.io.tmpdir=data/tmp

-Dkaraf.startLocalConsole=false

Running Nexus As A Service

It is better to have a init.d entry to manage nexus service using the Linux service command. Follow the steps given below for the setup.

**Step 1: Create a symbolic link for nexus service script to /etc/init.d folder.**

#sudo ln -s /app/nexus/bin/nexus /etc/init.d/nexus

**Step 2: Execute the following commands to add nexus service to boot.**

#sudo chkconfig --add nexus

#sudo chkconfig --levels 345 nexus on

Manage Nexus Service

Now we have all the configurations in place. To start the Nexus service, use the following command.

#sudo service nexus start

The above command will start the nexus service on port 8081. To access the nexus dashboard, visit http://:8081. You will be able to see the nexus homepage as shown below.