yum install –y git

How to Install Jenkins on CentOS 7

Prerequisites

Before proceeding, you must have:

* Deployed a Vultr CentOS 7 server instance from scratch.
* Logged into your machine as a non-root user with sudo privileges.

Step 1: Update your CentOS 7 system

One of the Linux system administrator's best practices is keeping a system up to date. Install the latest stable packages, then reboot.

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

This command will tell you about the Java runtime environment that you have installed:

openjdk version "1.8.0\_91"

OpenJDK Runtime Environment (build 1.8.0\_91-b14)

OpenJDK 64-Bit Server VM (build 25.91-b14, mixed mode)

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>

### Installing Docker (OperatingSystem : CentOs 7)

There are two method to Install Docker in CentOS 7

## [METHOD1]: INSTALL WITH YUM

$ cd /etc/yum.repos.d

yum.repos.d]$ ls -la

CentOs-Base.repo,CentOs-CR.repo,CentOs-Debuginfo.repo,

CentOs-fasttrack.repo,CentOs-Media.repo,CentOs-Soureces.repo

CentOs-Vault.repo,epel.repo, epel-testing.repo

$ sudo vim docker.repo

[dockerrepo]

name=Docker Repository

baseurl=https://yum.dockerproject.org/repo/main/centos/7/

enable=1

gpgcheck=1

gpgkey=https://yum.dockerproject.org/gpg

$sudo yum update

$sudo yum install docker-engine

$sudo systemctl enable docker //To enable our docker service

$sudo systemctl start docker //start the docker

$sudo systemctl status docker //check the status of the docker

By default the docker create a file in "/var/run"

directory that own by the docker group called docker.sock,

if you are not the member of that group or you are not root

by default you dont have access to the docker daemon.

$ sudo usermod -aG docker your\_username //add your user to docker group

$ docker images

$ docker run hello-world // pull the image and create a Container

## [METHOD2]: INSTALL WITH SCRIPT

$ sudo yum update //make sure your existing yum packages are up-to-date

$ curl -fsSL https://get.docker.com/| sh //run the docker installation script

This Script adds the docker.repo repository and installs Docker.

$ sudo service docker start

## UNINSTALL DOCKER:

$ yum l

ist installed |grep docker //list the package you have installed

$ sudo yum -y remove docker-engine.x86\_64 //remove the package

This command does not remove images, containers,

volumes or user-created configuration files on host.

$ rm -rf /var/lib/docker //delete all your images,containers,volumes

<https://docs.docker.com/engine/admin/systemd/#manually-create-the-systemd-unit-files>

Change default container image loc:

vim /etc/docker/daemon.json

{

"graph": "/mnt/docker-data",

"storage-driver": "overlay"

}

140 cat /etc/docker/daemon.json

141 sudo systemctl daemon-reload

142 sudo systemctl restart docker

DOCKER MACHINE

$ curl -L https://github.com/docker/machine/releases/download/v0.12.2/docker-machine-`uname -s`-`uname -m` >/tmp/docker-machine &&

chmod +x /tmp/docker-machine &&

sudo cp /tmp/docker-machine /usr/local/bin/docker-machine

DOCKER COMPOSE:

1. Run this command to download the latest version of Docker Compose:
2. sudo curl -L https://github.com/docker/compose/releases/download/1.16.1/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose

**Use the latest Compose release number in the download command.**

The above command is an example, and it may become out-of-date. To ensure you have the latest version, check the [Compose repository release page on GitHub](https://github.com/docker/compose/releases).

If you have problems installing with curl, see [Alternative Install Options](https://docs.docker.com/compose/install/#alternative-install-options).

1. Apply executable permissions to the binary:
2. sudo chmod +x /usr/local/bin/docker-compose
3. Optionally, install [command completion](https://docs.docker.com/compose/completion/) for the bash and zsh shell.
4. Test the installation.
5. $ docker-compose --version

docker-compose version 1.16.1, build 1719ceb

GITLAB CENTOS7 Install

|  |  |
| --- | --- |
| accepted | Problem is solved! I contated the GitLab by your official page in Facebook and theys answer.  I am using GitLab in a Desktop and it was using ~700MB. If you too want turn off GitLab on startup, just execute in a terminal:  Stop gitlab at startup  sudo systemctl disable gitlab-runsvdir.service |

## ****Step 4: Change External URl****

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

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

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

sudo systemctl enable sshd

sudo systemctl start sshd

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

sudo systemctl reload firewalld

Next, install Postfix to send notification emails. If you want to use another solution to send emails please skip this step and [configure an external SMTP server](https://docs.gitlab.com/omnibus/settings/smtp.html) after GitLab has been installed.

sudo yum install postfix

sudo systemctl enable postfix

sudo systemctl start postfix

During Postfix installation a configuration screen may appear. Select 'Internet Site' and press enter. Use your server's external DNS for 'mail name' and press enter. If additional screens appear, continue to press enter to accept the defaults.

#### 2. Add the GitLab package repository and install the package

Add the GitLab package repository.

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

Next, install the GitLab package.

sudo yum install -y gitlab-ce

## ****Step 4: Change External URl****

If you want to change the external url the use follow below steps:

# vim /etc/gitlab/gitlab.rb

external\_url '[http://host.domain.com](http://host.domain.com/)'

#### 3. Configure and start GitLab

sudo gitlab-ctl reconfigure

yum install -y vim

yum update kernel

yum install -y gcc kernel-devel kernel-headers dkms make bzip2 perl

reboot

**otherwise issue:**

Failed to set up service vboxadd, please check the log file

/var/log/VBoxGuestAdditions.log for details.

[root@devopsserver mnt]# cat /var/log/VBoxGuestAdditions.log

vboxadd.sh: failed: Look at /var/log/vboxadd-install.log to find out what went wrong.

vboxadd.sh: failed: Look at /var/log/vboxadd-install.log to find out what went wrong.

vboxadd.sh: failed: modprobe vboxguest failed.

[root@devopsserver mnt]# cat /var/log/vboxadd-install.log

/tmp/vbox.0/Makefile.include.header:112: \*\*\* Error: unable to find the sources of your current Linux kernel. Specify KERN\_DIR=<directory> and run Make again. Stop.

mount -r -t iso9660 /dev/sr0 /mnt

yum install –y git

Issue:

[root@devopsserver ~]# sudo systemctl status jenkins.service

● jenkins.service - LSB: Jenkins Automation Server

Loaded: loaded (/etc/rc.d/init.d/jenkins; bad; vendor preset: disabled)

Active: active (exited) since Tue 2017-09-19 20:15:38 IST; 25s ago

Docs: man:systemd-sysv-generator(8)

Process: 9591 ExecStop=/etc/rc.d/init.d/jenkins stop (code=exited, status=0/SUCCESS)

Process: 9601 ExecStart=/etc/rc.d/init.d/jenkins start (code=exited, status=0/SUCCESS)

Sep 19 20:15:38 devopsserver systemd[1]: Starting LSB: Jenkins Automation Server...

Sep 19 20:15:38 devopsserver runuser[9607]: pam\_unix(runuser:session): session opened for user jenkins by (uid=0)

Sep 19 20:15:38 devopsserver jenkins[9601]: Starting Jenkins [ OK ]

Sep 19 20:15:38 devopsserver systemd[1]: Started LSB: Jenkins Automation Server.

[root@devopsserver ~]# /etc/rc.d/init.d/jenkins restart

Restarting jenkins (via systemctl): [ OK ]

[root@devopsserver ~]# ps -aef | grep -in jenkins

150:jenkins 9808 1 95 20:16 ? 00:00:02 /etc/alternatives/java -Dcom.sun.akuma.Daemon=daemonized -Djava.awt.headless=true -DJENKINS\_HOME=/var/lib/jenkins -jar /usr/lib/jenkins/jenkins.war --logfile=/var/log/jenkins/jenkins.log --webroot=/var/cache/jenkins/war --daemon --httpPort=8080 --debug=5 --handlerCountMax=100 --handlerCountMaxIdle=20

153:root 9830 1567 0 20:16 pts/0 00:00:00 grep --color=auto -in Jenkins

-🡪 Unable to install gitlab and Jenkins on same server.

Gitlab 502 error

Jenkins service not starting…

Sol: vim /etc/sysconfig/Jenkins

[root@devopsserver default]# cat /etc/sysconfig/jenkins | grep -in 8080

50:## Default: 8080

56:JENKINS\_PORT="8080"

Chnage Port to 9080

[root@devopsserver default]# /etc/init.d/jenkins restart

Restarting jenkins (via systemctl): [ OK ]

[root@devopsserver default]# !ps

ps -aef | grep -in jenkins

151:jenkins 25241 1 99 21:02 ? 00:00:03 /etc/alternatives/java -Dcom.sun.akuma.Daemon=daemonized -Djava.awt.headless=true -DJENKINS\_HOME=/var/lib/jenkins -jar /usr/lib/jenkins/jenkins.war --logfile=/var/log/jenkins/jenkins.log --webroot=/var/cache/jenkins/war --daemon --httpPort=9080 --debug=5 --handlerCountMax=100 --handlerCountMaxIdle=20

154:root 25276 5284 0 21:02 pts/1 00:00:00 grep --color=auto -in jenkins