**Install Docker CE on an Offline CentOS 7 Machine**

[Docker](https://www.docker.com/) is a computer software, that performs operating-system-level virtualization, also known as **Containerization**. Docker is developed by [Docker Inc](https://www.docker.com/company). Docker uses predefined and custom images to create and run containers. Images are provided through Docker Hub (an online Docker registry).

Containers are isolated from each other having their own set of binaries, libraries and configuration files. Unlike virtual machines, containers may or may not have their own operating system. The containers are run by the host operating system Kernel, therefore, a container is much light weight as compare to a virtual machine.

Docker is a **freemium** software. Because a **Community Edition (CE)**  with limited features is available free while an **Enterprise Edition (EE)** is available with more features and a license cost. Also Docker is provided with Red Hat Atomic Host 7, which is a RHEL 7 based operating system optimized for hosting containers.

|  |  |
| --- | --- |
| Read also: | [Installing Red Hat Atomic Host 7](https://ahmermansoor.blogspot.com/2019/01/installing-red-hat-atomic-host-7.html) |

In this article, we will install Docker-CE (Community Edition) on an Offline CentOS 7 machine and then we will download and run Docker images from Docker Hub on the same offline CentOS 7 machine.

**System Specification:**

In this article, we are using two isolated CentOS 7 servers. One server has access to Internet, while the other server isn’t connected with Internet.

|  |  |  |
| --- | --- | --- |
|  | **Offline Server** | **Online Server** |
| **Hostname:** | docker-offline.example.com | docker-online.example.com |
| **Operating System:** | CentOS 7.6 | CentOS 7.6 |
| **Internet:** | No | Yes |

Operating systems on both systems must be same, otherwise the package versions may be different and won’t work.

**Download Docker-CE including Dependent Packages:**

Connect with docker-online.example.com using **ssh**.

**Docker-CE** requires some packages that are available in EPEL (Extra Packages for Enterprise Linux) yum repository. Therefore, we have to install EPEL yum repository before installing the Docker-CE.

[root@docker-online ~]# yum install -y epel-release.noarch

Loaded plugins: fastestmirror

Loading mirror speeds from cached hostfile

\* base: repo.inara.pk

\* extras: repo.inara.pk

\* updates: repo.inara.pk

base | 3.6 kB 00:00

extras | 3.4 kB 00:00

updates | 3.4 kB 00:00

Resolving Dependencies

--> Running transaction check

---> Package epel-release.noarch 0:7-11 will be installed

--> Finished Dependency Resolution

Dependencies Resolved

================================================================================

Package Arch Version Repository Size

================================================================================

Installing:

epel-release noarch 7-11 extras 15 k

Transaction Summary

================================================================================

Install 1 Package

Total download size: 15 k

Installed size: 24 k

Downloading packages:

epel-release-7-11.noarch.rpm | 15 kB 00:01

Running transaction check

Running transaction test

Transaction test succeeded

Running transaction

Installing : epel-release-7-11.noarch 1/1

Verifying : epel-release-7-11.noarch 1/1

Installed:

epel-release.noarch 0:7-11

Complete!

EPEL yum repository has been installed. Install [Docker-CE yum repository for CentOS](https://docs.docker.com/install/linux/docker-ce/centos/) now.

[root@docker-online ~]# yum-config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo

Loaded plugins: fastestmirror

adding repo from: https://download.docker.com/linux/centos/docker-ce.repo

grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo

repo saved to /etc/yum.repos.d/docker-ce.repo

Enable **Docker-CE (Nightly)** yum repository.

[root@docker-online ~]# yum-config-manager --enable docker-ce-nightly

Loaded plugins: fastestmirror

=========================== repo: docker-ce-nightly ============================

[docker-ce-nightly]

async = True

bandwidth = 0

base\_persistdir = /var/lib/yum/repos/x86\_64/7

baseurl = https://download.docker.com/linux/centos/7/x86\_64/nightly

cache = 0

cachedir = /var/cache/yum/x86\_64/7/docker-ce-nightly

check\_config\_file\_age = True

compare\_providers\_priority = 80

cost = 1000

deltarpm\_metadata\_percentage = 100

deltarpm\_percentage =

enabled = 1

enablegroups = True

exclude =

failovermethod = priority

ftp\_disable\_epsv = False

gpgcadir = /var/lib/yum/repos/x86\_64/7/docker-ce-nightly/gpgcadir

gpgcakey =

gpgcheck = True

gpgdir = /var/lib/yum/repos/x86\_64/7/docker-ce-nightly/gpgdir

gpgkey = https://download.docker.com/linux/centos/gpg

hdrdir = /var/cache/yum/x86\_64/7/docker-ce-nightly/headers

http\_caching = all

includepkgs =

ip\_resolve =

keepalive = True

keepcache = False

mddownloadpolicy = sqlite

mdpolicy = group:small

mediaid =

metadata\_expire = 21600

metadata\_expire\_filter = read-only:present

metalink =

minrate = 0

mirrorlist =

mirrorlist\_expire = 86400

name = Docker CE Nightly - x86\_64

old\_base\_cache\_dir =

password =

persistdir = /var/lib/yum/repos/x86\_64/7/docker-ce-nightly

pkgdir = /var/cache/yum/x86\_64/7/docker-ce-nightly/packages

proxy = False

proxy\_dict =

proxy\_password =

proxy\_username =

repo\_gpgcheck = False

retries = 10

skip\_if\_unavailable = False

ssl\_check\_cert\_permissions = True

sslcacert =

sslclientcert =

sslclientkey =

sslverify = True

throttle = 0

timeout = 30.0

ui\_id = docker-ce-nightly/x86\_64

ui\_repoid\_vars = releasever,

basearch

username =

We have added two yum repositories in our CentOS 7 server. Therefore, we should build yum cache before installing from these repositories.

[root@docker-online ~]# yum makecache fast

Loaded plugins: fastestmirror

Loading mirror speeds from cached hostfile

epel/x86\_64/metalink | 9.1 kB 00:00

\* base: repo.inara.pk

\* epel: sg.fedora.ipserverone.com

\* extras: repo.inara.pk

\* updates: repo.inara.pk

base | 3.6 kB 00:00

docker-ce-nightly | 3.5 kB 00:00

docker-ce-stable | 3.5 kB 00:00

extras | 3.4 kB 00:00

updates | 3.4 kB 00:00

Metadata Cache Created

Create a directory to download Docker-CE and dependent packages.

[root@docker-online ~]# mkdir ~/docker

[root@docker-online ~]# cd ~/docker

Download Docker-CE and dependent packages using **yumdownloader** command.

[root@docker-online docker]# yumdownloader --resolve docker-ce

Loaded plugins: fastestmirror

Loading mirror speeds from cached hostfile

\* base: repo.inara.pk

\* epel: mirrors.aliyun.com

\* extras: repo.inara.pk

\* updates: repo.inara.pk

--> Running transaction check

---> Package docker-ce.x86\_64 3:18.09.1-3.el7 will be installed

--> Processing Dependency: container-selinux >= 2.9 for package: 3:docker-ce-18.09.1-3.el7.x86\_64

--> Processing Dependency: libseccomp >= 2.3 for package: 3:docker-ce-18.09.1-3.el7.x86\_64

--> Processing Dependency: containerd.io for package: 3:docker-ce-18.09.1-3.el7.x86\_64

--> Processing Dependency: docker-ce-cli for package: 3:docker-ce-18.09.1-3.el7.x86\_64

--> Processing Dependency: libcgroup for package: 3:docker-ce-18.09.1-3.el7.x86\_64

--> Processing Dependency: libseccomp.so.2()(64bit) for package: 3:docker-ce-18.09.1-3.el7.x86\_64

--> Running transaction check

---> Package container-selinux.noarch 2:2.74-1.el7 will be installed

--> Processing Dependency: policycoreutils-python for package: 2:container-selinux-2.74-1.el7.noarch

---> Package containerd.io.x86\_64 0:1.2.2-3.el7 will be installed

---> Package docker-ce-cli.x86\_64 1:18.09.1-3.el7 will be installed

---> Package libcgroup.x86\_64 0:0.41-20.el7 will be installed

---> Package libseccomp.x86\_64 0:2.3.1-3.el7 will be installed

--> Running transaction check

---> Package policycoreutils-python.x86\_64 0:2.5-29.el7\_6.1 will be installed

--> Processing Dependency: policycoreutils = 2.5-29.el7\_6.1 for package: policycoreutils-python-2.5-29.el7\_6.1.x86\_64

--> Processing Dependency: setools-libs >= 3.3.8-4 for package: policycoreutils-python-2.5-29.el7\_6.1.x86\_64

--> Processing Dependency: libsemanage-python >= 2.5-14 for package: policycoreutils-python-2.5-29.el7\_6.1.x86\_64

--> Processing Dependency: audit-libs-python >= 2.1.3-4 for package: policycoreutils-python-2.5-29.el7\_6.1.x86\_64

--> Processing Dependency: python-IPy for package: policycoreutils-python-2.5-29.el7\_6.1.x86\_64

--> Processing Dependency: libqpol.so.1(VERS\_1.4)(64bit) for package: policycoreutils-python-2.5-29.el7\_6.1.x86\_64

--> Processing Dependency: libqpol.so.1(VERS\_1.2)(64bit) for package: policycoreutils-python-2.5-29.el7\_6.1.x86\_64

--> Processing Dependency: libapol.so.4(VERS\_4.0)(64bit) for package: policycoreutils-python-2.5-29.el7\_6.1.x86\_64

--> Processing Dependency: checkpolicy for package: policycoreutils-python-2.5-29.el7\_6.1.x86\_64

--> Processing Dependency: libqpol.so.1()(64bit) for package: policycoreutils-python-2.5-29.el7\_6.1.x86\_64

--> Processing Dependency: libapol.so.4()(64bit) for package: policycoreutils-python-2.5-29.el7\_6.1.x86\_64

--> Running transaction check

---> Package audit-libs-python.x86\_64 0:2.8.4-4.el7 will be installed

---> Package checkpolicy.x86\_64 0:2.5-8.el7 will be installed

---> Package libsemanage-python.x86\_64 0:2.5-14.el7 will be installed

---> Package policycoreutils.x86\_64 0:2.5-29.el7 will be updated

---> Package policycoreutils.x86\_64 0:2.5-29.el7\_6.1 will be an update

---> Package python-IPy.noarch 0:0.75-6.el7 will be installed

---> Package setools-libs.x86\_64 0:3.3.8-4.el7 will be installed

--> Finished Dependency Resolution

Delta RPMs disabled because /usr/bin/applydeltarpm not installed.

(1/13): container-selinux-2.74-1.el7.noarch.rpm | 38 kB 00:01

(2/13): audit-libs-python-2.8.4-4.el7.x86\_64.rpm | 76 kB 00:01

(3/13): checkpolicy-2.5-8.el7.x86\_64.rpm | 295 kB 00:02

(4/13): libseccomp-2.3.1-3.el7.x86\_64.rpm | 56 kB 00:00

(5/13): libsemanage-python-2.5-14.el7.x86\_64.rpm | 113 kB 00:01

(6/13): libcgroup-0.41-20.el7.x86\_64.rpm | 66 kB 00:02

(7/13): policycoreutils-python-2.5-29.el7\_6.1.x86\_64.rpm | 456 kB 00:03

(8/13): python-IPy-0.75-6.el7.noarch.rpm | 32 kB 00:02

(9/13): policycoreutils-2.5-29.el7\_6.1.x86\_64.rpm | 916 kB 00:14

(10/13): setools-libs-3.3.8-4.el7.x86\_64.rpm | 620 kB 00:12

warning: /root/docker/docker-ce-cli-18.09.1-3.el7.x86\_64.rpm: Header V4 RSA/SHA512 Signature, key ID 621e9f35: NOKEY

Public key for docker-ce-cli-18.09.1-3.el7.x86\_64.rpm is not installed

(11/13): docker-ce-cli-18.09.1-3.el7.x86\_64.rpm | 14 MB 01:37

(12/13): docker-ce-18.09.1-3.el7.x86\_64.rpm | 19 MB 02:23

Public key for containerd.io-1.2.2-3.el7.x86\_64.rpm is not installed 00:00 ETA

(13/13): containerd.io-1.2.2-3.el7.x86\_64.rpm | 22 MB 02:26

Docker-CE and dependent packages has been downloaded.

[root@docker-online docker]# ls

audit-libs-python-2.8.4-4.el7.x86\_64.rpm

checkpolicy-2.5-8.el7.x86\_64.rpm

containerd.io-1.2.2-3.el7.x86\_64.rpm

container-selinux-2.74-1.el7.noarch.rpm

docker-ce-18.09.1-3.el7.x86\_64.rpm

docker-ce-cli-18.09.1-3.el7.x86\_64.rpm

libcgroup-0.41-20.el7.x86\_64.rpm

libseccomp-2.3.1-3.el7.x86\_64.rpm

libsemanage-python-2.5-14.el7.x86\_64.rpm

policycoreutils-2.5-29.el7\_6.1.x86\_64.rpm

policycoreutils-python-2.5-29.el7\_6.1.x86\_64.rpm

python-IPy-0.75-6.el7.noarch.rpm

setools-libs-3.3.8-4.el7.x86\_64.rpm

The above packages versions may be different on your Linux server due to the Linux versions your are using.

Now, we need to transfer it to docker-offline.example.com. Therefore, we create a tarball of all downloaded packages.

[root@docker-online docker]# tar cvzf ~/docker.tar.gz \*

audit-libs-python-2.8.4-4.el7.x86\_64.rpm

checkpolicy-2.5-8.el7.x86\_64.rpm

containerd.io-1.2.2-3.el7.x86\_64.rpm

container-selinux-2.74-1.el7.noarch.rpm

docker-ce-18.09.1-3.el7.x86\_64.rpm

docker-ce-cli-18.09.1-3.el7.x86\_64.rpm

libcgroup-0.41-20.el7.x86\_64.rpm

libseccomp-2.3.1-3.el7.x86\_64.rpm

libsemanage-python-2.5-14.el7.x86\_64.rpm

policycoreutils-2.5-29.el7\_6.1.x86\_64.rpm

policycoreutils-python-2.5-29.el7\_6.1.x86\_64.rpm

python-IPy-0.75-6.el7.noarch.rpm

setools-libs-3.3.8-4.el7.x86\_64.rpm

Now, we have a tarball of Docker-CE and dependent packages.

[root@docker-online docker]# ls ~ -lh

total 58M

-rw-------. 1 root root 1.5K Dec 22 11:29 anaconda-ks.cfg

drwxr-xr-x. 2 root root 4.0K Feb 8 21:32 docker

-rw-r--r--. 1 root root 58M Feb 8 21:39 docker.tar.gz

**Installing Docker-CE on Offline System:**

Transfer the docker.tar.gz tarball to docker-offline.example.com and copy at home directory of **root** user.

[root@docker-offline ~]# ls -lh

total 58M

-rw-------. 1 root root 1.5K Dec 22 12:23 anaconda-ks.cfg

-rw-r--r--. 1 root root 58M Feb 8 21:39 docker.tar.gz

Extract the docker.tar.gz tarball.

[root@docker-offline ~]# mkdir docker

[root@docker-offline ~]# tar xvf docker.tar.gz -C ~/docker

audit-libs-python-2.8.4-4.el7.x86\_64.rpm

checkpolicy-2.5-8.el7.x86\_64.rpm

containerd.io-1.2.2-3.el7.x86\_64.rpm

container-selinux-2.74-1.el7.noarch.rpm

docker-ce-18.09.1-3.el7.x86\_64.rpm

docker-ce-cli-18.09.1-3.el7.x86\_64.rpm

libcgroup-0.41-20.el7.x86\_64.rpm

libseccomp-2.3.1-3.el7.x86\_64.rpm

libsemanage-python-2.5-14.el7.x86\_64.rpm

policycoreutils-2.5-29.el7\_6.1.x86\_64.rpm

policycoreutils-python-2.5-29.el7\_6.1.x86\_64.rpm

python-IPy-0.75-6.el7.noarch.rpm

setools-libs-3.3.8-4.el7.x86\_64.rpm

Install all packages in the ~/docker directory.

[root@docker-offline ~]# cd docker

[root@docker-offline docker]# rpm -ivh --replacefiles --replacepkgs \*.rpm

warning: containerd.io-1.2.2-3.el7.x86\_64.rpm: Header V4 RSA/SHA512 Signature, key ID 621e9f35: NOKEY

Preparing... ################################# [100%]

Updating / installing...

1:policycoreutils-2.5-29.el7\_6.1 ################################# [ 8%]

2:libcgroup-0.41-20.el7 ################################# [ 15%]

3:setools-libs-3.3.8-4.el7 ################################# [ 23%]

4:python-IPy-0.75-6.el7 ################################# [ 31%]

5:libsemanage-python-2.5-14.el7 ################################# [ 38%]

6:libseccomp-2.3.1-3.el7 ################################# [ 46%]

7:docker-ce-cli-1:18.09.1-3.el7 ################################# [ 54%]

8:containerd.io-1.2.2-3.el7 ################################# [ 62%]

9:checkpolicy-2.5-8.el7 ################################# [ 69%]

10:audit-libs-python-2.8.4-4.el7 ################################# [ 77%]

11:policycoreutils-python-2.5-29.el7################################# [ 85%]

12:container-selinux-2:2.74-1.el7 ################################# [ 92%]

13:docker-ce-3:18.09.1-3.el7 ################################# [100%]

Start and enable Docker service.

[root@docker-offline docker]# systemctl enable docker.service

Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.

[root@docker-offline docker]# systemctl start docker.service

We have successfully installed Docker-CE on an offline CentOS 7 server.

**Download and Run Docker images on an Offline CentOS 7 Server:**

Before using Docker we required Docker images, that are provided by Docker-Hub and other Docker Registry Servers.

Since, our CentOS 7 server is not connected to Internet, therefore, we cannot pull Docker images directly from Docker Hub.

The workaround of this problem is to download Docker images on an Online system and then transfer and load these images to our Offline CentOS 7 server.

**Note:** You must have Docker-CE installed on the online system first. Because we will use the **docker** command to download and save images from Docker Hub.

[root@docker-online ~]# docker pull jenkins/jenkins

Using default tag: latest

latest: Pulling from jenkins/jenkins

741437d97401: Pull complete

34d8874714d7: Pull complete

0a108aa26679: Pull complete

7f0334c36886: Pull complete

aa29d9cbdbf5: Pull complete

e54d29f74413: Pull complete

eb5b24cf4e1f: Pull complete

5edfd6c9b475: Pull complete

b00dabba5e89: Pull complete

9f51dff87c48: Pull complete

0544e8830903: Pull complete

dd2464419c60: Pull complete

78125f701da6: Pull complete

5e3b2221f1a0: Pull complete

8700b2d54fbc: Pull complete

4613d2e35dec: Pull complete

08320da45709: Pull complete

8f947c5bbe77: Pull complete

51cf55002ec2: Pull complete

9537066ae19a: Pull complete

e156275467ac: Pull complete

Digest: sha256:20981c20164347728fca4774b3c45f5d24a73d857e8b9b8e6faf4100cfc0812d

Status: Downloaded newer image for jenkins/jenkins:latest

Similarly, pull more images according to your requirement.

Save Jenkins image in a tar file.

[root@docker-online ~]# docker save jenkins/jenkins > ~/jenkins.tar

[root@docker-online ~]# ls -lh

total 690M

-rw-------. 1 root root 1.5K Dec 22 11:29 anaconda-ks.cfg

-rw-r--r--. 1 root root 690M Feb 12 22:07 jenkins.tar

Transfer jenkins.tar to docker-offline.example.com.

Load jenkins.tar image into docker.

[root@docker-offline ~]# docker load < jenkins.tar

13d5529fd232: Loading layer 105.6MB/105.6MB

abc3250a6c7f: Loading layer 24.07MB/24.07MB

578414b395b9: Loading layer 8.005MB/8.005MB

6257fa9f9597: Loading layer 146.4MB/146.4MB

364be905de1c: Loading layer 2.332MB/2.332MB

57eab9d93a79: Loading layer 3.584kB/3.584kB

ad6eaafe7ab3: Loading layer 1.536kB/1.536kB

b98fdbf8cf7f: Loading layer 356.3MB/356.3MB

596ecd570594: Loading layer 1.698MB/1.698MB

f5ee7c2ae54f: Loading layer 338.9kB/338.9kB

349fe2545d85: Loading layer 3.584kB/3.584kB

b2b9702adfd1: Loading layer 9.728kB/9.728kB

ad16984b47fb: Loading layer 868.9kB/868.9kB

c469c008fbc0: Loading layer 4.608kB/4.608kB

5a8ce619bb31: Loading layer 77.33MB/77.33MB

8eae0810e454: Loading layer 4.608kB/4.608kB

5924ca705d38: Loading layer 9.216kB/9.216kB

852edd42bb1e: Loading layer 4.608kB/4.608kB

03b3a4ed2e5a: Loading layer 3.072kB/3.072kB

37dfb8384dfe: Loading layer 7.168kB/7.168kB

8f65ce1dc902: Loading layer 12.29kB/12.29kB

Loaded image: jenkins/jenkins:latest

jenkins.tar image has been loaded into Docker. Use following command to verify this.

[root@docker-offline ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

jenkins/jenkins latest 9b74eda1c268 32 hours ago 704MB

Now, we can create and run Docker containers from the Jenkins/Jenkins image.

Docker-CE has been installed on our Offline CentOS 7 server. To start using Docker you should refer to Docker Documentation or from beginner to advance level reference we recommend you should read Docker Deep Dive.