# Install Docker

## Install EPEL Repository

Create local yum repos directory, and import the EPEL repo.

|  |
| --- |
| $ pbrun bash  # mkdir -p /opt/mount1/yum/yum.repos.d  # mkdir -p /opt/mount1/yum/yum.repos.d/disabled\_repos  # cp -f /etc/yum.repos.d/\* /opt/mount1/yum/yum.repos.d/  # mv /opt/mount1/yum/yum.repos.d/epel.repo /opt/mount1/yum/yum.repos.d/disabled\_repos/  # cp -f /etc/yum.conf /opt/mount1/yum/  # echo "reposdir=/opt/mount1/yum/yum.repos.d" >> /opt/mount1/yum/yum.conf  # echo "http\_caching=none" >> /opt/mount1/yum/yum.conf  # echo "proxy=http://web-proxy.houston.hp.com:8080" >> /opt/mount1/yum/yum.conf  # export http\_proxy=http://web-proxy.houston.hp.com:8080  # export https\_proxy=http://web-proxy.houston.hp.com:8080  # cd /tmp  # wget --no-check-certificate https://fedoraproject.org/static/0608B895.txt  # rpm --import 0608B895.txt  # rm -f 0608B895.txt  # vim /opt/mount1/yum/yum.repos.d/epel.repo  [epel]  name=Extra Packages for Enterprise Linux 6 - $basearch  #baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch  mirrorlist=http://mirrors.fedoraproject.org/metalink?repo=epel-6&arch=$basearch  failovermethod=priority  enabled=1  gpgcheck=1  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6  [epel-debuginfo]  name=Extra Packages for Enterprise Linux 6 - $basearch - Debug  #baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch/debug  mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-debug-6&arch=$basearch  failovermethod=priority  enabled=0  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6  gpgcheck=1  [epel-source]  name=Extra Packages for Enterprise Linux 6 - $basearch - Source  #baseurl=http://download.fedoraproject.org/pub/epel/6/SRPMS  mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-source-6&arch=$basearch  failovermethod=priority  enabled=0  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6  gpgcheck=1  # yum --config=/opt/mount1/yum/yum.conf repolist |

## Install Docker

Docker requires a 64-bit installation regardless of your Red Hat version. Docker requires that your kernel must be 3.10 at minimum. Red Hat 7 runs the 3.10 kernel, 6.6 does not. We make an exception for Red Hat 6.6. To run Docker on Red Hat-6.6 or later, you need kernel 2.6.32-431 or higher.

|  |
| --- |
| $ uname -r  2.6.32-504.12.2.el6.x86\_64 |

|  |
| --- |
| $ pbrun bash  # yum -y --config=/opt/mount1/yum/yum.conf install docker-io  # mkdir -p /opt/mount2/var/docker  # vim /etc/sysconfig/docker  other\_args="-g /opt/mount2/var/docker"  HTTP\_PROXY=http://web-proxy.houston.hp.com:8080  http\_proxy=$HTTP\_PROXY  HTTPS\_PROXY=$HTTP\_PROXY  https\_proxy=$HTTP\_PROXY  export HTTP\_PROXY HTTPS\_PROXY http\_proxy https\_proxy  # chkconfig docker on  # service docker start  # docker info  # docker version |

# Install Docker Registry

<https://github.com/docker/docker-registry>

<https://pypi.python.org/pypi/docker-registry>

## Install Nginx

|  |
| --- |
| $ pbrun bash  # vim /etc/yum.repos.d/nginx.repo  [nginx]  name=nginx repo  baseurl=http://nginx.org/packages/rhel/6/$basearch/  gpgcheck=0  enabled=1  # vim /etc/yum.conf  proxy=http://web-proxy.houston.hp.com:8080  # yum -y install nginx  # service nginx start |

## Setup Python Environment

pip, virtualenv, virtualenvwrapper

|  |
| --- |
| $ pbrun bash  # export http\_proxy=http://web-proxy.houston.hp.com:8080  # export https\_proxy=http://web-proxy.houston.hp.com:8080  # cd /tmp  # wget --no-check-certificate <https://raw.github.com/pypa/pip/master/contrib/get-pip.py>  # python get-pip.py  # rm -f get-pip.py  # pip --proxy http://web-proxy.houston.hp.com:8080 install virtualenv  # pip --proxy http://web-proxy.houston.hp.com:8080 install virtualenvwrapper  # exit |

Activate virtual environment by virtualenvwrapper

|  |
| --- |
| $ export WORKON\_HOME=$HOME/.virtualenvs  $ source /usr/bin/virtualenvwrapper.sh  $ lsvirtualenv  $ workon <environment> |

## Install Supervisord

|  |
| --- |
| $ pbrun bash  # export http\_proxy=http://web-proxy.houston.hp.com:8080  # export https\_proxy=http://web-proxy.houston.hp.com:8080  # pip --proxy http://web-proxy.houston.hp.com:8080 install supervisor  # echo\_supervisord\_conf > /etc/supervisord.conf  # wget --no-check-certificate -O /etc/init.d/supervisor https://raw.githubusercontent.com/Supervisor/initscripts/master/redhat-init-mingalevme  # chmod +x /etc/init.d/supervisor  # chkconfig --add supervisor  # chkconfig supervisor on  # service supervisor start |

## Install Docker Registry

python-devel, openssl-devel, xz-devel

|  |
| --- |
| $ pbrun bash  # yum -y install python-devel  # yum -y install opensll-devel  # yum -y install xz-devel  # exit |

Install docker registry by pip

|  |
| --- |
| $ export WORKON\_HOME=$HOME/.virtualenvs  $ source /usr/bin/virtualenvwrapper.sh  $ mkvirtualenv docker-registry  (docker-registry)$ export http\_proxy=http://web-proxy.houston.hp.com:8080  (docker-registry)$ export https\_proxy=http://web-proxy.houston.hp.com:8080  (docker-registry)$ pip --proxy http://web-proxy.houston.hp.com:8080 install git+https://github.com/martinpaljak/M2Crypto.git  (docker-registry)$ pip --proxy http://web-proxy.houston.hp.com:8080 install docker-registry |

## Configure Docker Registry

|  |
| --- |
| $ pbrun bash  # mkdir /opt/mount2/docker-registry  # chown jiand:ldap /opt/mount2/docker-registry  # exit  $ cd ~/.virtualenvs/docker-registry/lib/python2.6/site-packages/config/  $ cp config\_sample.yml config.yml  $ vim config.yml  common: &common  # By default, the registry acts standalone (eg: doesn't query the index)  ~~standalone: \_env:STANDALONE:true~~  standalone: \_env:STANDALONE:false  # The default endpoint to use (if NOT standalone) is index.docker.io  ~~index\_endpoint: \_env:INDEX\_ENDPOINT: https://index.docker.io~~  index\_endpoint: \_env:INDEX\_ENDPOINT:http://c1t13203.itcs.hp.com:5000  local: &local  <<: \*common  storage: local  ~~storage\_path: \_env:STORAGE\_PATH:/tmp/registry~~  storage\_path: /opt/mount2/docker-registry/registry  $ mkdir /opt/mount2/docker-registry/log  $ mkdir /opt/mount2/docker-registry/registry  $ gunicorn --access-logfile /opt/mount2/docker-registry/log/access.log --error-logfile /opt/mount2/docker-registry/log/error.log --max-requests 100 -k gevent --graceful-timeout 3600 -t 3600 -w 4 -b 0.0.0.0:5000 docker\_registry.wsgi:application |

## Configure Supervisor

|  |
| --- |
| $ pbrun bash  # mkdir /opt/mount1/docker-registry  # chown jiand:ldap /opt/mount1/docker-registry  # exit  $ mkdir /opt/mount1/docker-registry/bin  $ vim /opt/mount1/docker-registry/bin/start.sh  #!/bin/bash  NAME="docker\_registry"  VENVDIR=/home/jiand/.virtualenvs/docker-registry/  USER=jiand  GROUP=ldap  NUM\_WORKERS=3  echo "Starting $NAME as `whoami`"  # activate the virtualenv  cd $VENVDIR  source bin/activate  gunicorn --access-logfile /opt/mount2/docker-registry/log/access.log --error-logfile /opt/mount2/docker-registry/log/error.log --max-requests 100 -k gevent --graceful-timeout 3600 -t 3600 -b 127.0.0.1:5000 --name $NAME --workers $NUM\_WORKERS --user=$USER --group=$GROUP docker\_registry.wsgi:application  $ chmod +x /opt/mount1/docker-registry/bin/start.sh  $ pbrun bash  # vim /etc/supervisord.conf  [program:docker\_registry]  command=/opt/mount1/docker-registry/bin/start.sh  # service supervisor restart |

## Configure Nginx

|  |
| --- |
| $ pbrun bash  # vim /etc/nginx/conf.d/docker\_registry.conf  server {  listen 80;  server\_name localhost;  location / {  proxy\_pass http://127.0.0.1:5000;  client\_max\_body\_size 0;  }  }  # service nginx restart |

## Windows File Share

Full control:

|  |
| --- |
| $ pbrun bash  # yum -y install cifs-utils  # mkdir /opt/docker-registry-storage  # chown jiand:ldap /opt/docker-registry-storage/  # mount -t cifs //americas.cpqcorp.net/EG/119954\_hpit-w-hp-bugzilla-dev\_docker-registry-storage -o username=da-sheng.jian@hp.com,password=HPpass234 /opt/docker-registry-storage |

# Install Docker Index

## Install Nginx

|  |
| --- |
| $ pbrun bash  # vim /etc/yum.repos.d/nginx.repo  [nginx]  name=nginx repo  baseurl=http://nginx.org/packages/rhel/6/$basearch/  gpgcheck=0  enabled=1  # vim /etc/yum.conf  proxy=http://web-proxy.houston.hp.com:8080  # yum -y install nginx  # service nginx start |

## Setup Python Environment

pip, virtualenv, virtualenvwrapper

|  |
| --- |
| $ pbrun bash  # export http\_proxy=http://web-proxy.houston.hp.com:8080  # export https\_proxy=http://web-proxy.houston.hp.com:8080  # cd /tmp  # wget --no-check-certificate <https://raw.github.com/pypa/pip/master/contrib/get-pip.py>  # python get-pip.py  # rm -f get-pip.py  # pip --proxy http://web-proxy.houston.hp.com:8080 install virtualenv  # pip --proxy http://web-proxy.houston.hp.com:8080 install virtualenvwrapper  # exit |

Activate virtual environment by virtualenvwrapper

|  |
| --- |
| $ export WORKON\_HOME=$HOME/.virtualenvs  $ source /usr/bin/virtualenvwrapper.sh  $ lsvirtualenv  $ workon <environment> |

## Install Supervisord

|  |
| --- |
| $ pbrun bash  # export http\_proxy=http://web-proxy.houston.hp.com:8080  # export https\_proxy=http://web-proxy.houston.hp.com:8080  # pip --proxy http://web-proxy.houston.hp.com:8080 install supervisor  # echo\_supervisord\_conf > /etc/supervisord.conf  # wget --no-check-certificate -O /etc/init.d/supervisor https://raw.githubusercontent.com/Supervisor/initscripts/master/redhat-init-mingalevme  # chmod +x /etc/init.d/supervisor  # chkconfig --add supervisor  # chkconfig supervisor on  # service supervisor start |

## Install Docker Index

python-devel

|  |
| --- |
| $ pbrun bash  # yum -y install python-devel  # exit |

Prepare the environment

|  |
| --- |
| $ pbrun bash  # mkdir /opt/mount1/docker-index  # chown jiand:ldap /opt/mount1/docker-index  # mkdir /opt/mount2/docker-index  # chown jiand:ldap /opt/mount2/docker-index  # exit  $ mkdir /opt/mount2/docker-index/db  $ export WORKON\_HOME=$HOME/.virtualenvs  $ source /usr/bin/virtualenvwrapper.sh  $ mkvirtualenv docker-index |

Install docker index source code to /opt/mount1/docker-index

|  |
| --- |
| $ …  (docker-install)$ pip --proxy http://web-proxy.houston.hp.com:8080 install -r /opt/mount1/docker-index/requirements/main.txt |

## Configure Docker Index

|  |
| --- |
| $ vim /opt/mount1/docker-index/config.py  DEBUG = False  TESTING = False  SECRET\_KEY = 'hard to guess string'  SQLALCHEMY\_DATABASE\_URI = 'sqlite:////opt/mount2/docker-index/db/data.sqlite'  SQLALCHEMY\_COMMIT\_ON\_TEARDOWN = True  MAIL\_SERVER = 'localhost'  MAIL\_PORT = 25  DI\_APPLICATION\_NAME = 'Docker Index'  DI\_MAIL\_SUBJECT\_PREFIX = '[Docker Index] '  DI\_MAIL\_SENDER = 'jiand@c1t13203.itcs.hp.com'  DI\_REGISTRY\_ENDPOINT = 'c1t13202.itcs.hp.com'  DI\_REGISTRY\_VERSION = '0.9.1' |

## Configure Supervisor

|  |
| --- |
| $ vim /opt/mount1/docker-index/bin/gunicorn\_start.sh  #!/bin/bash  NAME="docker\_index"  INDEXDIR=/opt/mount1/docker-index  VENVDIR=/home/jiand/.virtualenvs/docker-index/  USER=jiand  GROUP=ldap  NUM\_WORKERS=3  echo "Starting $NAME as `whoami`"  # activate the virtualenv  cd $VENVDIR  source bin/activate  export PYTHONPATH=$INDEXDIR:$PYTHONPATH  gunicorn --access-logfile /opt/mount1/docker-index/log/access.log --error-logfile /opt/mount1/docker-index/log/error.log --max-requests 100 -k gevent --graceful-timeout 3600 -t 3600 -b 127.0.0.1:5000 --name $NAME --workers $NUM\_WORKERS --user=$USER --group=$GROUP run:app  $ chmod +x /opt/mount1/docker-index/bin/gunicorn\_start.sh  $ pbrun bash  # vim /etc/supervisord.conf  [program:docker\_index]  command=/opt/mount1/docker-index/bin/gunicorn\_start.sh  # service supervisor restart |

## Configure Nginx

|  |
| --- |
| $ pbrun bash  # vim /etc/nginx/conf.d/docker\_index.conf  server {  listen 80;  server\_name localhost;  location / {  proxy\_pass http://127.0.0.1:5000;  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;  }  }  # service nginx restart |