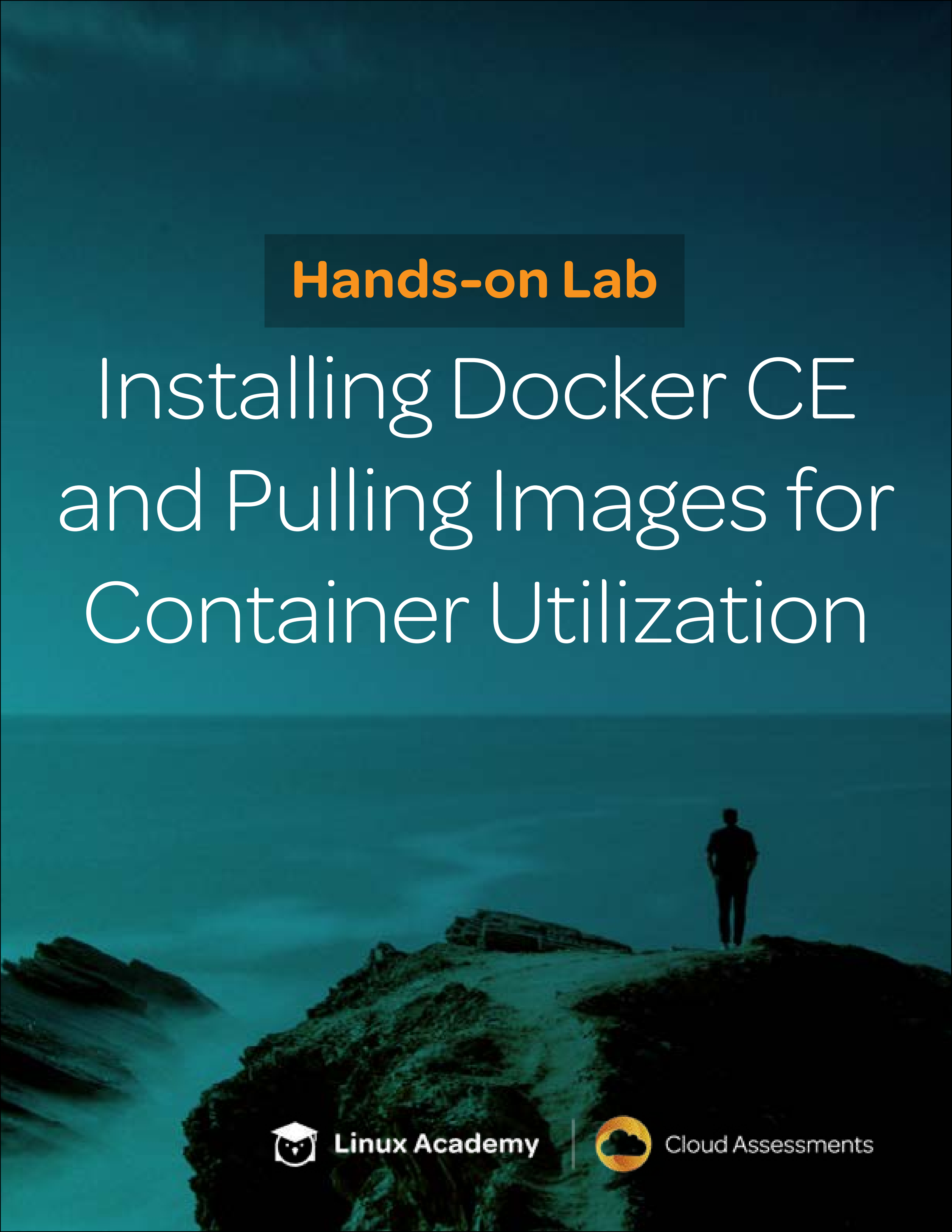

Hands-on Lab

Installing Docker CE and Pulling Images for Container Utilization

Linux Academy | Cloud Assessments

Contents

Our development team is beginning development of a new web-based API. With our organization moving to a Docker Swarm implementation, they need some time to get used to containers and how to use them.

They have provided us with the credentials and connection information for a CentOS 7 system in the development environment. We have been asked to install the Docker Community Edition on this server. We need to configure it so that it starts automatically after a reboot, that is uses the latest version of Docker, and that it has all of the current dependencies.

We must also configure Docker CE so that our users can execute Docker CE commands without needing to do so with sudo privileges.

Also, the following images need to be locally available:

- CentOS 7 (latest)
- CentOS 6 (latest)
- Ubuntu 14.04 (latest)
- HTTPD

Install Docker CE on a Single Node

Log in to the server using the credentials provided on the hands-on lab page:

```
[linux@linuxacademy ~]# cloud_user@IP$
```

When prompted, enter the provided password.

Once logged in, install the needed yum prerequisite packages. You may be asked for a password; this is the same password you logged in with:

```
[cloud_user@IP ~]$ sudo yum install -y yum-utils device-mapper-
persistent-data lvm2
```

Install Docker CE

Now that we have the yum package installed, we can create our Docker repository and add it to our repo list:

```
[cloud_user@IP ~]$ sudo yum-config-manager --add-repo https://
download.docker.com/linux/centos/docker-ce.repo
```

With the repository in place, we can install Docker CE. Whenever prompted, enter y to agree to the installation:

```
[cloud_user@IP ~]$ sudo yum install docker-ce
```

Make sure this is done after you add the Docker repository; we want the most recent version of Docker, not the version that is located in the default repositories.

Add a User to a User Group

We also need to add our user, cloud_user, to the Docker user group. This lets our user execute Docker commands without using the sudo program. Once we set this up, we will need to log out and then back into the server:

```
[cloud_user@IP ~]$ sudo usermod -aG docker cloud_user
[cloud_user@IP ~]$ exit
```

Log back into the server using the same credentials you signed in with.

Enable Sudo Privileges

With Docker CE installed, and our user group set up, we can set up our Docker Daemon.

As tasked, we need to enable sudo privileges on the Docker Daemon. While doing so, you will be asked for the password you used to log in:

```
[cloud_user@IP ~]$ sudo systemctl enable docker && sudo systemctl
start docker && sudo systemctl status docker
[sudo] password for cloud_user:
```

Information over the docker Daemon appears. We can get more specific by looking at our docker image repository:

```
[cloud_user@IP ~]$ docker images
REPOSITORY      TAG      IMAGE      ID      CREATED      SIZE
```

Our current repository is empty.

Running docker info provides us with a large amount of information over the docker. In this guide, we will only be showing the command, not the information that follows:

```
[cloud_user@IP ~]$ docker info
```

Lastly, let's check our version. At the time of our lab recording, the most recent version was 17.09. As long as you have that version or higher, then you will have the correct docker:

```
[cloud_user@IP ~]$ docker --version
Docker version 17.09.1-ce, build 19e2cf6
```

If you have more than one node, then complete this process for each additional node.

Pull and Confirm Image Downloads

With Docker CE set up, we can pull our images into it. First, run docker images to check what we have so far:

```
[cloud_user@IP ~]$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
```

Our repository is empty, for now.

Pull in the Needed Images

Let's get our Centos image first. Using docker again, pull the Centos image. We will be using centos:latest, though you can specify a version, such as centos:7, as well:

```
[cloud_user@IP ~]$ docker pull centos:latest
```

The latest version of the image is pulled in.

Use the command docker images again to make sure we have the image:

```
[cloud_user@IP ~]$ docker images
```

An item called centos appears in our repository. Great! Now, we have three more images we need to pull in. Note that after each pull, information over the image will appear. This information will not be reflected in the guide; only the command will be shown.

Using docker, pull in the images centos: 6, ubuntu: 14.04, and HTTPD:

```
[cloud_user@IP ~]$ docker pull centos:6

[cloud_user@IP ~]$ docker pull ubuntu:14.04

[cloud_user@IP ~]$ docker pull httpd
```

Using the docker images command once again, let's review and make sure all of our images appear.

```
[cloud_user@IP ~]$ docker images
```

The repository now holds all our needed images.

Review

Congratulations! You are now able to install and configure Docker CE, as well as use it to pull images into your local repository. Feel free to keep practicing in the lab environment even after completing the lab.