

This guide covers the installation of Docker and Docker Compose on **CentOS** and **Amazon Linux**. Follow the respective instructions based on your environment.

## Docker Installation for CentOS

### Prerequisites

- CentOS 8 or higher
- Root or sudo access
- Minimum 2GB of RAM recommended, 8GB of RAM recommended for all Liberty Framework Services.

### Step 1: Update System Packages

Before starting the installation, update your system to ensure all packages are up-to-date.

```
1 sudo yum update -y
```

if Podman is installed, remove all packages, artifacts and containers storage

```
1 yum remove buildah skopeo podman containers-common atomic-registries docker container-tools
2 rm -rf /etc/containers/* /var/lib/containers/* /etc/docker /etc/subuid* /etc/subgid*
3 cd ~ && rm -rf /.local/share/containers/
```

### Step 2: Install Required Dependencies

Install the necessary packages required to set up the Docker repository.

```
1 sudo yum install -y yum-utils
```

### Step 3: Set Up the Docker Repository

Add the Docker repository to your CentOS system.

```
1 sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

### Step 4: Install Docker

Install Docker Engine, CLI, and Containerd.

```
1 sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

### Step 5: Start and Enable Docker

Start the Docker service and enable it to start on boot.

```
1 sudo systemctl start docker
2 sudo systemctl enable docker
```

Verify the installation by running a test Docker container.

```
1 | sudo docker run hello-world
```

If the container runs and displays a welcome message, Docker is installed correctly.

### Step 7: Adding Your User to the Docker Group (Optional)

To run Docker commands without `sudo`, add your user to the Docker group.

```
1 | sudo usermod -aG docker $(whoami)
```

Log out and log back in to apply the group changes.

### Uninstall Docker

To remove Docker, the CLI, Containerd, and Docker Compose, use the following commands:

```
1 | sudo yum remove docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin docker-  
2 | ce-rootless-extras  
3 | sudo rm -rf /var/lib/docker  
   | sudo rm -rf /var/lib/containerd
```

## Docker Installation for Amazon Linux OS

### Prerequisites

- Amazon Linux or Amazon Linux 2
- Root or sudo access
- Minimum 2GB of RAM recommended, 8GB of RAM recommended for all Liberty Framework Services.

### Step 1: Update System Packages

Before starting the installation, update your system to ensure all packages are up-to-date.

```
1 | sudo yum update -y
```

### Step 2: Install Docker

Install Docker using the Amazon Linux Extras & yum package manager.

```
1 | sudo amazon-linux-extras install docker -y
```

### Step 3: Start and Enable Docker

Start the Docker service and enable it to start on boot.

```
1
```

```
2 | sudo systemctl start docker
   | sudo systemctl enable docker
```

## Step 4: Verify Docker Installation

Verify the installation by running a test Docker container.

```
1 | sudo docker run hello-world
```

If the container runs and displays a welcome message, Docker is installed correctly.

## Step 5: Install Docker Compose

Download the current stable release of Docker Compose:

```
1 | sudo curl -L "https://github.com/docker/compose/releases/download/$(curl -s
   | https://api.github.com/repos/docker/compose/releases/latest | grep -Po '"tag_name": "\K.*?(?=")')/docker-
   | compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Apply executable permissions to the binary:

```
1 | sudo chmod +x /usr/local/bin/docker-compose
```

Verify that the installation was successful:

```
1 | docker-compose --version
```

## Step 6: Adding Your User to the Docker Group (Optional)

To run Docker commands without `sudo`, add your user to the Docker group.

```
1 | sudo usermod -aG docker $(whoami)
```

Log out and log back in to apply the group changes.

## Uninstall Docker

To remove Docker, the CLI, Containerd, and Docker Compose, use the following commands:

```
1 | sudo yum remove docker
2 | sudo rm -rf /var/lib/docker
3 | sudo rm /usr/local/bin/docker-compose
```

## Post installation Tasks

If you want to set a custom directory for docker and if you are running behind a proxy, the docker service must be modified

Edit the service: `/lib/systemd/system/docker.service`

```
1 [Service]
2 Type=notify
3 # the default is not to use systemd for cgroups because the delegate issues still
4 # exists and systemd currently does not support the cgroup feature set required
5 # for containers run by docker
6 ExecStart=/usr/bin/dockerd --data-root <CUSTOM_DIRECTORY> -H fd:// --
7 containerd=/run/containerd/containerd.sock
8 ExecReload=/bin/kill -s HUP $MAINPID
9 TimeoutStartSec=0
10 RestartSec=2
11 Restart=always
12 Environment="HTTP_PROXY=<PROXY_URL>"
   Environment="HTTPS_PROXY=<PROXY_URL>"
```

If you want to change the default IP range (172.17.x.x) for docker Edit the file: /etc/docker/daemon.json

```
1 # Set the ip range according to your requirements
2 # bip is for the internal interface
3 # default-address-pools is for all new networks
4 {
5   "bip": "172.26.0.1/16",
6   "default-address-pools": [
7     { "base": "172.27.0.0/16", "size": 24 }
8   ]
9 }
```

## Conclusion

You have successfully installed Docker and Docker Compose on your CentOS or Amazon Linux OS system. You can now begin deploying and managing your Docker containers for Liberty Framework.

## References

- [Docker Documentation](#)
- [AWS Documentation](#)