

# Basic docker commands

- [Intro](#)
- [Requirements](#)
- [Docker installation](#)
- [Login](#)
- [Working with images](#)
  - [Pull image from official public docker repository](#)
  - [Pull image from specific repository](#)
  - [Tagging image with specific tag or version](#)
  - [Push image to specific repository](#)
  - [Images list](#)
  - [Image remove](#)
- [Working with containers](#)
  - [Containers list](#)
  - [Running container](#)
  - [Stopping container](#)
- [Other helpful commands](#)
  - [Clearing local docker data](#)

## Intro

Basic docker commands page has been introduced to help docker newcomers to start working with docker with few basic commands.

## Requirements

- You know the Artifactory docker repository URL where your desired image currently is or will be located
- Valid credentials with access to desired repository
- Your machine where you will work with Docker.



It can be virtual machine, server, testbed, laptop, workstation - this is up to you and your possibilities.

- Docker client installed on your machine according to [Docker installation](#)
- Proper connectivity from your machine to repository URL

## Docker installation

There is multiple ways how to install docker client. You need to have sudo access on your machine to be able to setup Docker.

We recommend this:

- on CentOS/Fedora

```
sudo yum install docker -y
```

- on Ubuntu

```
sudo apt install docker -y
```

For using docker without "sudo" user must be member of "docker" group on testbed to be able to utilize docker properly without need to type sudo all the time:

```
sudo groupadd docker
sudo usermod -a -G docker $USER
sudo systemctl restart docker
docker version
```

Docker daemon unix socket is owned by root user, standard user cannot operate docker by default. However, docker daemon can be controlled by members of "docker" group, so we are utilising this possibility.

## Login

To be able to work with private docker registry/repository, users must login first. More information can be found at <https://docs.docker.com/engine/reference/commandline/login/>

```
docker login <REPOSITORY_URL>

# Or with providing username to avoid asking username repeatedly
docker login -u <USER_NAME> <REPOSITORY_URL>

# Or with providing username and password to avoid asking credentials - do not use in shared environments as
other users could be able to find your password in plaintext in command history
docker login -u <USER_NAME> -p <USER_PASSWORD> <REPOSITORY_URL>
```

## Working with images

### Pull image from official public docker repository

```
docker pull <IMAGE>:<TAG>
```

### Pull image from specific repository

```
docker pull <REPOSITORY_KEY>.artifactory.shared.tds.CUSTOMERX.com/<IMAGE>:<TAG>
```

### Tagging image with specific tag or version

```
docker tag <IMAGE>:<TAG> <REPOSITORY_KEY>.artifactory.shared.tds.CUSTOMERX.com/<IMAGE>:<TAG>
```

### Push image to specific repository

```
docker push <REPOSITORY_KEY>.artifactory.shared.tds.CUSTOMERX.com/<IMAGE>:<TAG>
```

### Images list

```
docker images
```

### Image remove

```
docker rm <IMAGE_ID>
```

## Working with containers

### Containers list

```
docker container ls --all
```

### Running container

```
docker run <REPOSITORY_KEY>.artifactory.shared.tds.CUSTOMERX.com/<IMAGE>:<TAG>
```

### Stopping container

```
docker stop <CONTAINER>
```

## Other helpful commands

### Clearing local docker data



This command removes all unused images, containers and other data from docker locally. For more info, refer to official documentation.

It is helpful for easy starting with the "clear table".

```
docker system prune -a -f
```