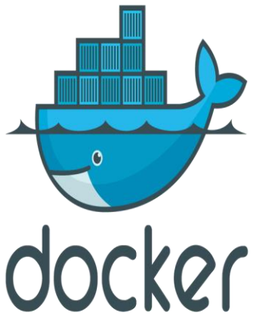
Docker

You Already Know

Course(s):

Docker for the Absolute Beginner - Hands-On



Recap

- Containerize your web application
  - Docker architecture
  - Setting up Docker on virtual machines
  - Docker Hub repository

- Manage a Docker container
  - Docker volume
  - Docker network

- Configure and deploy pipeline
  - Jenkins pipeline and stages
  - Deployment with Jenkins pipeline

Recap

- Explain Docker Compose, Docker Swarm, and Docker Images

  - What is Docker?
  - Docker Architecture
  - Containers vs. Virtual Machines
  - Docker Command Line
  - Docker Commands
  - Exporting a port
  - Docker run
  - Docker Images
  - Docker compose
  - Docker swarm

A Day in the Life of a Full Stack Developer

Joe, the Full Stack Developer for Abq Inc., has been working very hard to deliver all projects adhering to the expected quality.

Joe has been assigned a new task during sprint planning. He needs to configure a Jenkins job to perform Docker image build and configure Swarm service on Swarm master. He needs to to deploy the Docker image to Docker Swarm finally.

In this lesson, we will learn how to solve this real-world scenario and help Joe effectively complete his task.

Learning Objectives

By the end of this lesson, you will be able to:

- Build an image and run it as a container
- Push an image to Docker Hub
- Deploy a Docker container on Docker Swarm
- Scale a Docker container on Docker Swarm

Build a Custom Docker Image

Duration: 15 min.

Problem Statement:

Demonstrate how to build a Docker image and run it as a container.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to create a custom docker image:

1. Build a custom Docker image
2. Initialize a custom container on Docker host
3. Validate connectivity using the curl command
4. Push the code to GitHub repositories

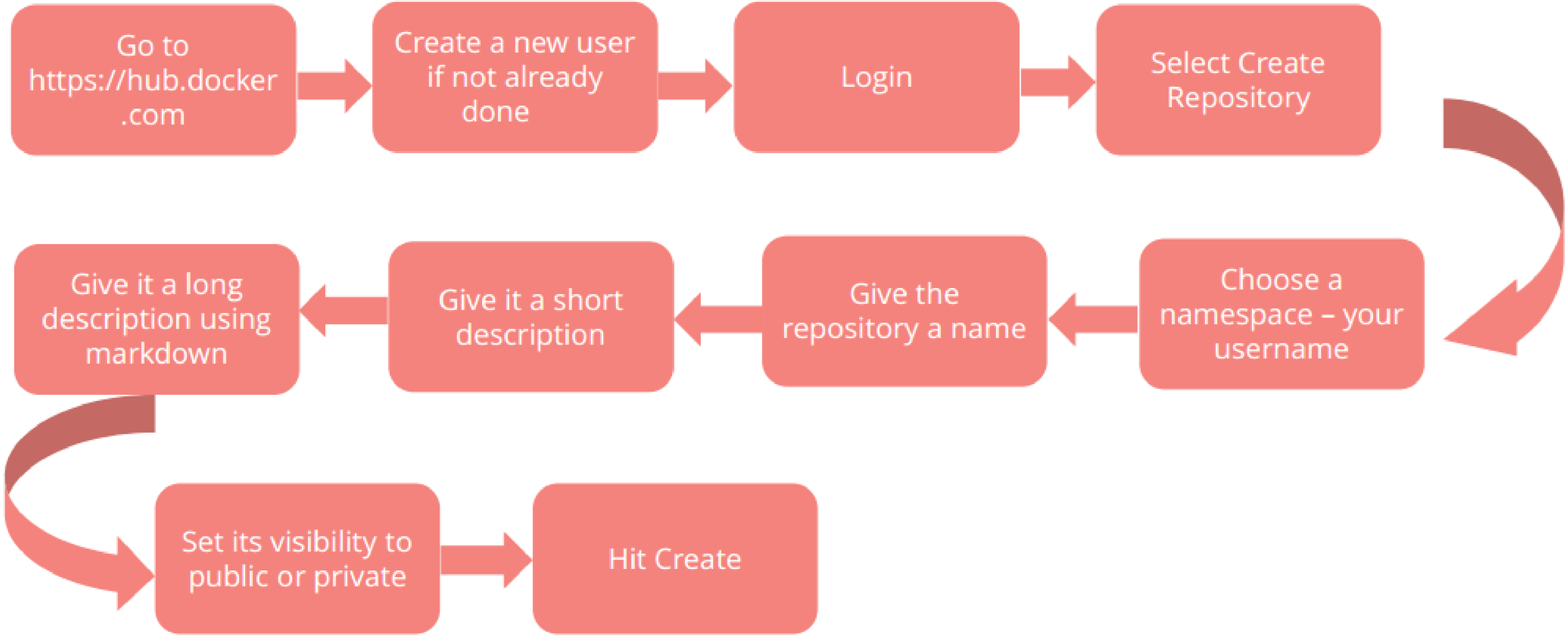
Introduction to Docker Hub

Docker Hub

Docker Hub is a hosted repository service provided by Docker to find and share container images with your team. Key features include:

- Private Repositories: Push and pull container images
- Automated Builds: Automatically build container images from GitHub and Bitbucket and push them to Docker Hub
- Teams & Organizations: Manage access to private repositories
- Official Images: Pull and use high-quality container images provided by Docker
- Publisher Images: Pull and use high-quality container images provided by external vendors. Certified images also include support and guarantee compatibility with Docker Enterprise
- Webhooks: Trigger actions after a successful push to a repository to integrate Docker Hub with other services

Docker Hub Configuration: Step-by-Step Procedure



Docker Hub Login

Docker login command is used to connect to Docker Hub from Docker client CLI. Before doing this, you must tag existing image with Docker image concatenated with Docker username.

docker tag <image_name><tag> <docker_hub_username>/ <image_name><tag>
docker push <docker_hub_username>/ <image_name><tag>

```
root@docker:~# docker tag docker_apache anujsharma1990/docker_apache
root@docker:~# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: anujsharma1990
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@docker:~# docker push anujsharma1990/docker_apache
The push refers to repository [docker.io/anujsharma1990/docker_apache]
e7e6e790a930: Pushed
735b1c81b582: Pushed
76c033092e10: Mounted from library/ubuntu
2146d867acf3: Mounted from library/ubuntu
ae1f631f14b7: Mounted from library/ubuntu
102645f1cf72: Mounted from library/ubuntu
latest: digest: sha256:7d8cccf25ef3b9f1578f0efb2d26a0c67cacc7c168709c12310c0ad47ad821e9 size: 1776
root@docker:~#
```

Push an Image to Docker Hub

Duration: 20 min.

Problem Statement:
Demonstrate how to push a Docker image to Docker Hub.

ASSISTED PRACTICE

Assisted Practice:Guidelines

Steps to push an image to docker hub:

1. Prepare a custom Docker image

2. Push the Docker image to Docker Hub

3. Push the code to GitHub repositories

Container Deployment Using Docker Swarm

Duration: 20 min.

Problem Statement:
Demonstrate how to deploy a Docker container on Docker swarm.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to deploy container in docker swarm:

1. Set up the Docker instance
2. Build a custom Docker image
3. Initialize the Docker swarm cluster and deploy a container to the cluster
4. Push the code to GitHub repositories

Container Scaling with Docker Swarm

Duration: 15 min.

Problem Statement:
Demonstrate how to scale a Docker container on Docker swarm.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to scale docker container:

1. Create a Docker service

2. Scale the service up and down to increase and decrease Docker containers

3. Push the code to GitHub repositories

Distribute Your App Across a Cluster

Duration: 15 min.

Problem Statement:
Demonstrate how to distribute your app across a swarm cluster.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to distribute app across a cluster:

1. Set up the Docker instance
2. Set up Docker swarm with multiple nodes
3. Deploy a custom Docker image to Docker swarm cluster
4. Push the code to GitHub repositories

Setting Up Jenkins Pipeline

Duration: 15 min.

Problem Statement:
Demonstrate how to set up Jenkins pipeline and connect it to docker cloud.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to set up Jenkins pipeline with Docker:

1. Install Docker plugin
2. Configure Docker cloud
3. Configure Jenkins job
4. Push the code to GitHub repositories

Key Takeaways

- Docker is a tool designed to create, deploy, and run applications easily using containers.
- A container is a standard unit of software that packs up code and its dependencies.
- Docker Hub is a hosted repository service provided by Docker to find and share container images with your team.
- Swarm utilizes the same command line from Docker.

Setting Up Jenkins Pipeline to Deploy Docker Swarm

Duration: 30 min.

Problem Statement:
You are given a project to accomplish the following:

- Configure a Jenkins job to perform Docker image build
- Configure Swarm service on Swarm master for Docker image
- Deploy image to Docker Swarm

LESSON-END PROJECT