Danielle Ussery
Professor Chintan Thakkar
October 17, 2021

# Week 3 Live

This week we went over how to store, distribute, and manage docker images and how to create a docker registry on the docker hub. We first need to build the docker image before we can store the image on docker hub. For example, if a client deploys images, we can publish those in our registry. From there we can give access to the client so they can deploy images on their end. To push images we need to specify the server name, org ID, and the tag name in order to push the image properly.

ECR (Elastic Container Registry) and the ECS (Elastic Container Service) pipeline, deals with the code commit process. It manages containers by making it easy to deploy, manage, and scale applications. We're unable to store the data on our computer because that's how the images are being deployed originally. Depending on the company, if they use a short cloud then they'll go with a short container. If they're using Amazon cloud then they would most likely work with Amazon's elastic container registry.

Digest uniquely identifies content and is a cryptographically strong hash. All content is now part of a named repository, images are no longer a secret, and the namespace is opened up to allow more than two components.

The docker image is the entire file that's being used to execute code in the docker container. Whenever you are deploying, you're going to grab the image tag. We can create a repository, which will provide a pull command.

Docker compose allows us to run multiple containers at d time in a compose ".yml" file. It uses a single command to deploy the entire app, handles container dependencies, works with Docker Swarm, Networking, Volumes, and Universal Control Plane.There is a 3-step process: define environment with a docker file, define the services that make up your app in a Docker compose file, then run the CLI: "$ docker-compose up".