# ASE Fall 22

Docker Presentation

# What is Docker?

It’s a container, but what’s a container

- A container is a way to package applications with all the necessary dependencies and configurations
- Can be thought of as an executable component that combines source code and OS dependencies required for the application
- Built using an image - a **portable** artifact that can easily be shared
- Images are usually stored in some forms of **repositories**

# What is a Docker container?

Now that we know docker is a container let’s talk about Docker container

- A docker container is a specific type of container (the most popular)
- It is built using **docker** images
- Docker images are stored in **dockerhub** repository

# Why does Docker matter?

- Every configuration of the application as well as all dependencies are already packaged in a container image. You can leverage the image to build the container and run your application
- The application is running in a isolated environment;
- Any changes in the container FS will be destroyed upon termination
- A development environment using containers is consistent with the production environment

# Is Docker a virtual machine?

- Unlike VMs, containers don’t carry the payload of an entire OS instance and hypervisor. They include only the OS processes and dependencies necessary to execute the code. Container sizes are measured in megabytes (vs. gigabytes for some VMs), make better use of hardware capacity, and have faster startup times.
- Containerized applications can be written once and run anywhere. And compared to VMs, containers are faster and easier to deploy, provision and restart. This makes them ideal for use in continuous integration and continuous delivery (CI/CD) pipelines and a better fit for development teams adopting Agile and DevOps practices.
- With containers, developers can run several times as many copies of an application on the same hardware as they can using VMs. This can reduce cloud spending.

Source: IBM https://www.ibm.com/cloud/learn/docker

# How do dockers container work?

- The Linux Kernel has process isolation and virtualization capabilities
- For instance, Cgroups (Control groups) allocate resources among processes so that access and visibility into other areas of the system are restricted
- Docker leverages this technology that’s why its containers are isolated environment

# Demo

- Start a java application that collects information about a person and stores it into a MongoDB database
- We’ll host the MongoDB server in a docker container
- Running the DB server in Docker ensures consistency throughout the team as the production environment and development will be consistent
- Can create new isolated instances of MongoDB server and you can connect the volume associated with one instance to another

# Resources

- https://www.mongodb.com/compatibility/docker
- https://www.ibm.com/cloud/learn/docker
- https://docs.docker.com/engine/install/
- https://hub.docker.com/