# Docker Introduction

* It provides the ability to run one or more application in an isolated environment called a container
* It is a containerization ecosystem

## What is Containerization?

* Involved bundling an application together with all the configuration files, libraries, and dependencies required
* We will be using **Docker** and **Kubernetes**
* You are hosting a virtual OS
* When you create a container, the allocation of resources is dynamic
  + It will only use as many resources as possible the container needs to run

## What is Virtualization?

* It is a creation of a virtual machine that stimulates a real computer with an operating system
* When you create virtual machines, the allocation of resources is static
  + Meaning once you start a virtual machine, you cannot change the resources it currently has
* It is heavy to run

# What is the purpose of Docker?

* It allows developers to work in standardize environments using containers
  + Meaning they can work on any computer with any OS they are comfortable with and still be able to share their application to everyone
* Makes it perfect for CI/CD workflow
* It makes scaling and deployment way easier
  + Docker containers can run on most things (cloud providers, data center, virtual/physical machine)
  + You can scale up or tear down application as business dictates

# Docker artifacts/terminology

## Docker Images

* They are standalone package that includes everything we need to run an application such as code itself, runtime environment, libraries, etc.
* They are immutable file and represents an application and its virtual environment at a specific point in time
  + Great for consistency when sharing your application everyone
  + Immutable meaning once you make it you can’t change it, you’ll have to create a new version of your image if you want to update it
* Essentially, they are a virtual file storage that has a copy of your files and other development platform it needs to run those files

## Docker Container

* **An image cannot run on its own,** it needs a docker container to run the image
* Docker container is the runnable instance of a docker image

## Docker ignore

* It will ignore certain files and not copy them to the image
* Like gitignore

## Docker Registry

* It is a server-side or cloud application where you can store your images and make it easy to distribute to everyone else
* Think of github but just for docker images
* **Docker hub**

## Docker Configuration

* Contains all the information to tell the docker container how to run the image
* We used the docker run command to already configure this (mostly just changing what port the container should run on)

# Docker instructions

## From

* Initialize our build stage and sets the base image
* Essentially this is where we indicate what we need to be able create/run our application

## Workdir

* Sets the working directory
* Just creates a folder and this is where we will copy and paste our files into that folder

## Copy

* Copy and pastes files into the image’s virtual file system

## Run

* It will run what you put in the terminal

# Docker CLI

* This will build the docker image
  + Docker build -t [username]/[imageName]:[imageVersion] .
* This will run the docker image in a container
  + Docker run -d -p 5001:80 -t [username]/[imageName]:[imageVersion]