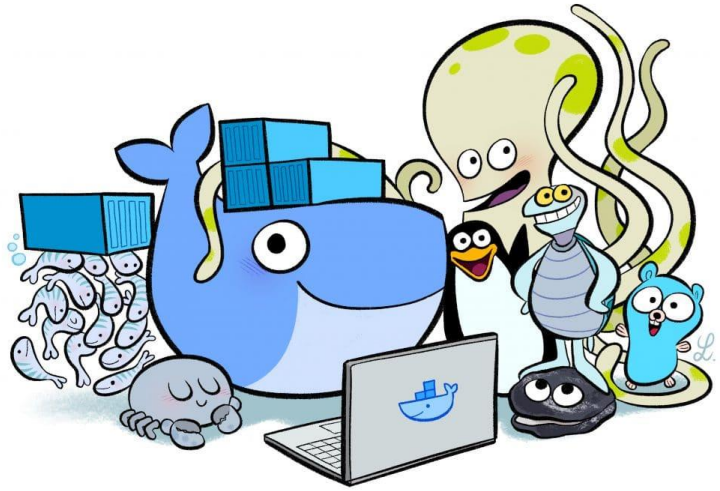
**CSCI 5409:** Adv. Topic In Cloud Computing

TA: Purvesh Rathod



# Tutorial 2: Web apps running in Docker

# Tutorial Summary

- Learning outcomes
  - Image Layers
  - Dockerize node application
  - Networking in Docker
  - Docker Volume

# Image Layers

- Images are made up of file systems changes and metadata.
- Each layers is uniquely identified and only stored once on a host.
- It saves storage space on the host and transfer time on pull/push.
- A container is just a single read/write layer on top of image.

Steps for building an image

- FROM (base image)
- ENV (environment variables)
- RUN (any shell command)
- EXPOSE (Open port from a container)
- CMD (Execute when container starts)
- docker image build (creating image)

# Docker Network

- Each container connects to a private virtual network called “bridge” by default.
- The virtual network routes through the NAT firewall on the host IP address.
- The containers in the same virtual network can communicate with each other without “-p”.
- Docker best practice:
  - Create a new virtual network for each app.
  - E.g. “flask_web_app” for mysql and flask container
  - E.g. “api_net” for nodejs and mongo container.
- Use Docker **DNS** for inter network communication (Comes built-in with custom networks or use “**—net-alias**” flag).

**Commands:**

- Show network: docker network ls
- Inspect nw: docker network inspect
- Create a nw: docker network create –driver <>
- Attach a nw: docker network connect
- Detach a nw: docker network disconnect

# Docker Data Volumes & Bind Mount

- Container are **usually** immutable and ephemeral.
- Containers can have unique data or database data which should be related to container but the host.
- Docker gives features to ensure “Separation of concerns”
  - Update the container/version of app while preserving the data.
  - **Note:** Data remains persistent even if you stop/restart the container. It gets removed when you delete the container.
  - Docker Solutions: Volumes and Bind Mounts
- Volumes: Assign a location outside the UFS of container.
- Bind Mounts: Link the container path to host path.

Commands

- Show volumes: docker volume ls
- Create volume: docker volume create <->
- Named volume: [-v] <name>:<path>
- E.g., docker run **–v mysql-db:/var/lib/mysql** mysql
- Bind mount can’t use in Dockerfile (must be container run)
- Bind mount: … run –v **//c/Users/../:/path/container (windows)**

  **/User/../:/path/container (mac/linux)**

# Questions?

# Thank You!