# Demos

## Demo - Docker Installation

* + Purpose
    - Docker Installation, Basic commands execution
  + Steps
  + EC2 Launch
  + EC2 Connect
  + Within the terminal
    - “sudo apt update”
      * To update all the repos
    - Install Docker
      * “sudo apt install docker.io”
        + Installs both docker-client and docker-server in one go
      * “docker version”
        + Note

Permission Settings - Issuing the command “docker version” results in permission error, however issuing “sudo docker version” works

Resolution

Execute the below command to make sure that the user “ubuntu” is not part of the “docker” group

“sudo usermod -aG docker ubuntu”

## Demo - Whalesay

* Purpose
  + Launching a container
* Steps
  + Execute the command
    - “docker run docker/whalesay cowsay hello”
      * Notes
        + The version is not provided, hence the default version “:latest” will be taken out
    - Execute the following command will list down the ‘whalesay’ image
      * “docker images”
    - Execute the following command to list down the containers
      * “docker ps -a” (“docker ps --all)

## Demo -MySql

* Purpose
  + Containzering MySql application
* Steps
  + Pull down the mysql image
    - “docker pull mysql”
  + Context
    - When a MySql based container has to be launched, the credentials username/password somehow need to be specified as well
    - Command
      * “docker run -d -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=password\_123 mysql”
      * -d option is for detached mode - background mode
  + Login to the mysql container
    - “docker exec -it <my\_sql\_container\_name> bash”
  + Connect to the mysql client
    - mysql -u root -p
      * Note
        + There will be a prompt to enter the password. Enter the value as “password\_123”
        + Here “root” is the root username
  + Create sample database and tables
    - Database creation
      * CREATE DATABASE SAMPLE;
    - Use the database
      * USE SAMPLE;
    - Table creation
      * CREATE TABLE CUSTOMER (
      * CUS\_ID INT NOT NULL PRIMARY KEY,
      * CUS\_NAME VARCHAR(20) NOT NULL
      * );
    - Insert Records
      * INSERT INTO CUSTOMER VALUES (1, 'Sachin Tendulkar');
      * INSERT INTO CUSTOMER VALUES (2, 'Saurav Ganguly');
    - View records
      * select \* from CUSTOMER;
  + “exit”
    - This comes out from the mysql shell and comes to Docker shell
  + “exiit”
    - This comes out from the Docker shell and comes to Ubuntu shell

## Demo - Cleanup

* Purpose
  + Bulk Cleanup
* Steps
  + For cleanup, the following docker elements need to be removed
    - Images
    - Containers
  + Dependency
    - Because Containers are launched from images, the containers need to be cleanup first followed by images
  + To do a bulk clean, lets issue the following commands first
    - Docker images
    - Docker ps -a
  + For Containers
    - The following command “docker ps -a -q” shows all the ids of the containers
    - The sequence is to stop the containers first and then to remove the container
      * Stop all the containers first
        + “docker stop $(docker ps -a -q)”

Issuing the above command removes all the containers in 1 go

* + - * Remove all the containers next
        + docker rm $(docker ps -a -q)
    - Verify using “docker ps -a” to see that there are no containers
  + For Images
    - The below command list down the ids of all images
      * “docker images -q”
    - Remove all the images
      * docker rmi $(docker images -q)
    - Verify using “docker images” to see that there are no images