**Docker:**

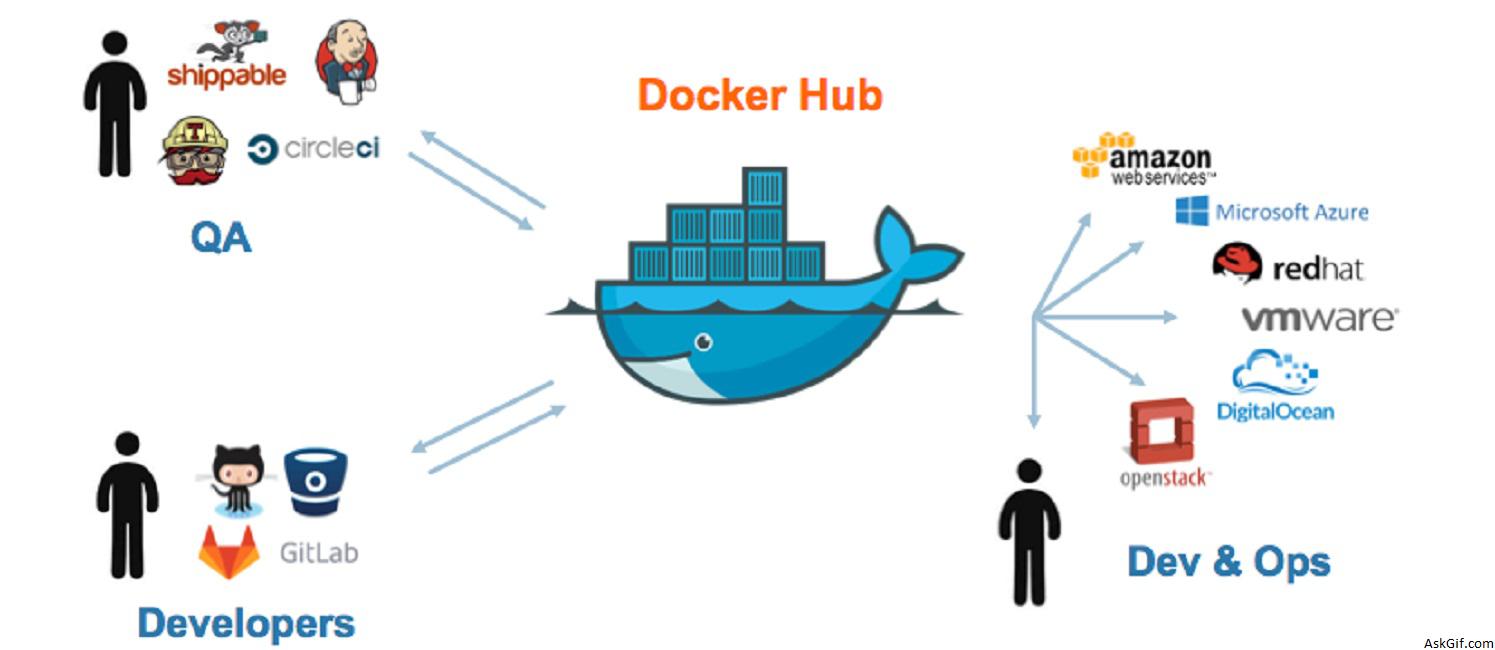
* Docker is the world’s leading software container platform.
* Docker makes the process of application deployment very easy and efficient and resolves a lot of issues related to deploying applications.
* Docker is a tool designed to make it easier to deploy and run applications by using containers.
* Docker gives you a standard way of packaging your application with all its dependencies in a container.
* Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

[**Virtualization**](https://www.google.com/search?rlz=1C1GCEA_enIN866IN866&sxsrf=ALeKk00BQPzK7gfEOOSc6kIuoJ_YWlGHjg:1587953338116&q=virtualization&spell=1&sa=X&ved=2ahUKEwjQno2Vw4fpAhWPxjgGHYfDBdAQkeECKAB6BAgWECc): This is a technology where will create Virtual Machines with necessary configurations (OS + Applications + Bins + Libs).

* This is very heavy: Takes lot of memory and costly.
* We need to take care of all required configurations and their versions.
* Older techniques.

**Containerization**: This is a technology where will create container on the top of container engines.

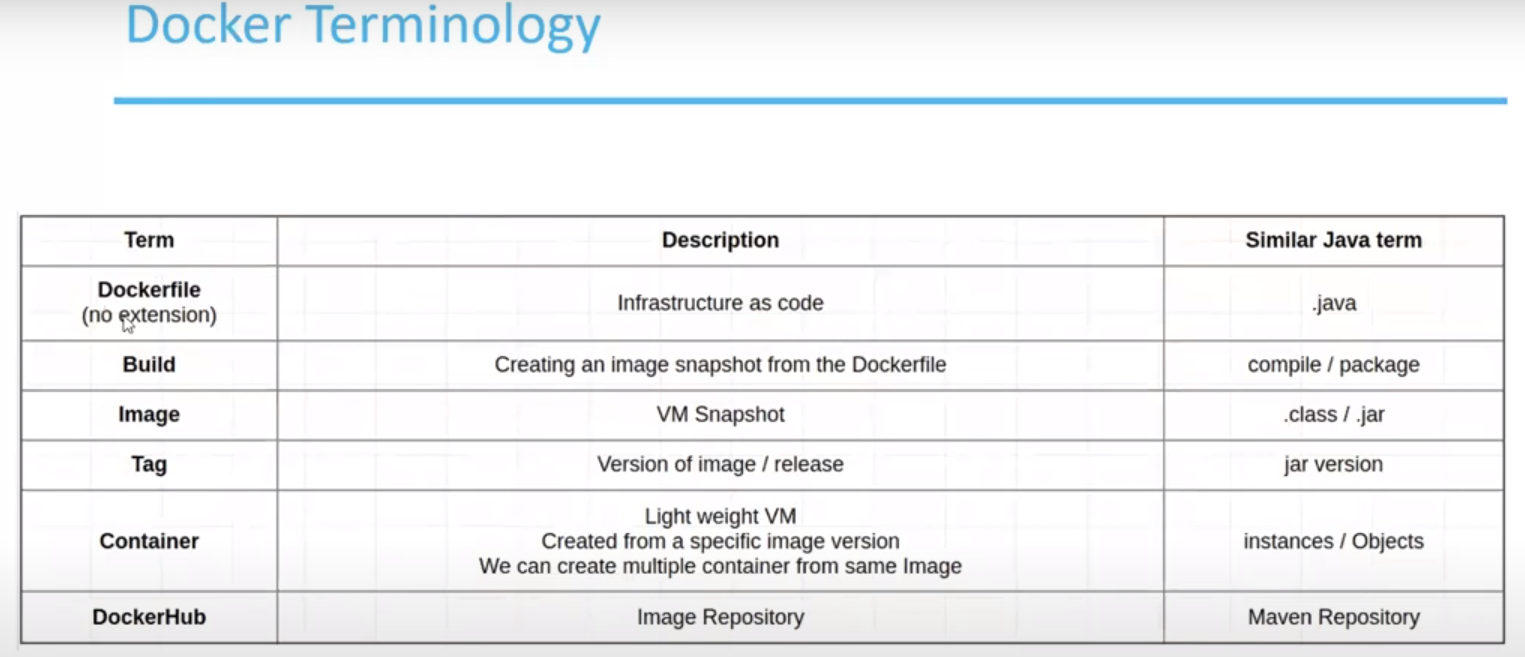
* Very light weight and required only 1 VM (we can create multiple container including Hub and Nodes in single VM).
* No OS+ Applications + Libs +Bins
* Will created container with the help of docker images.

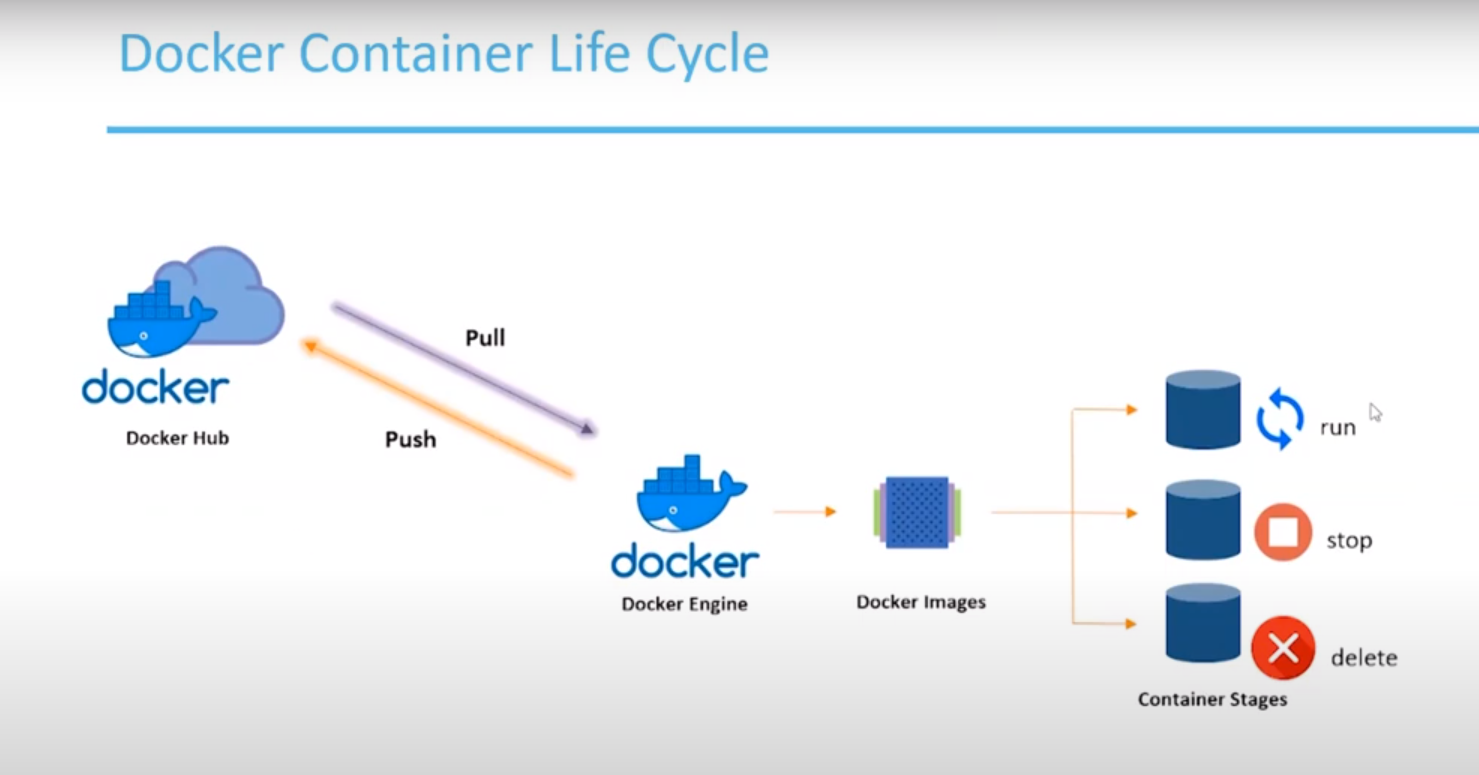


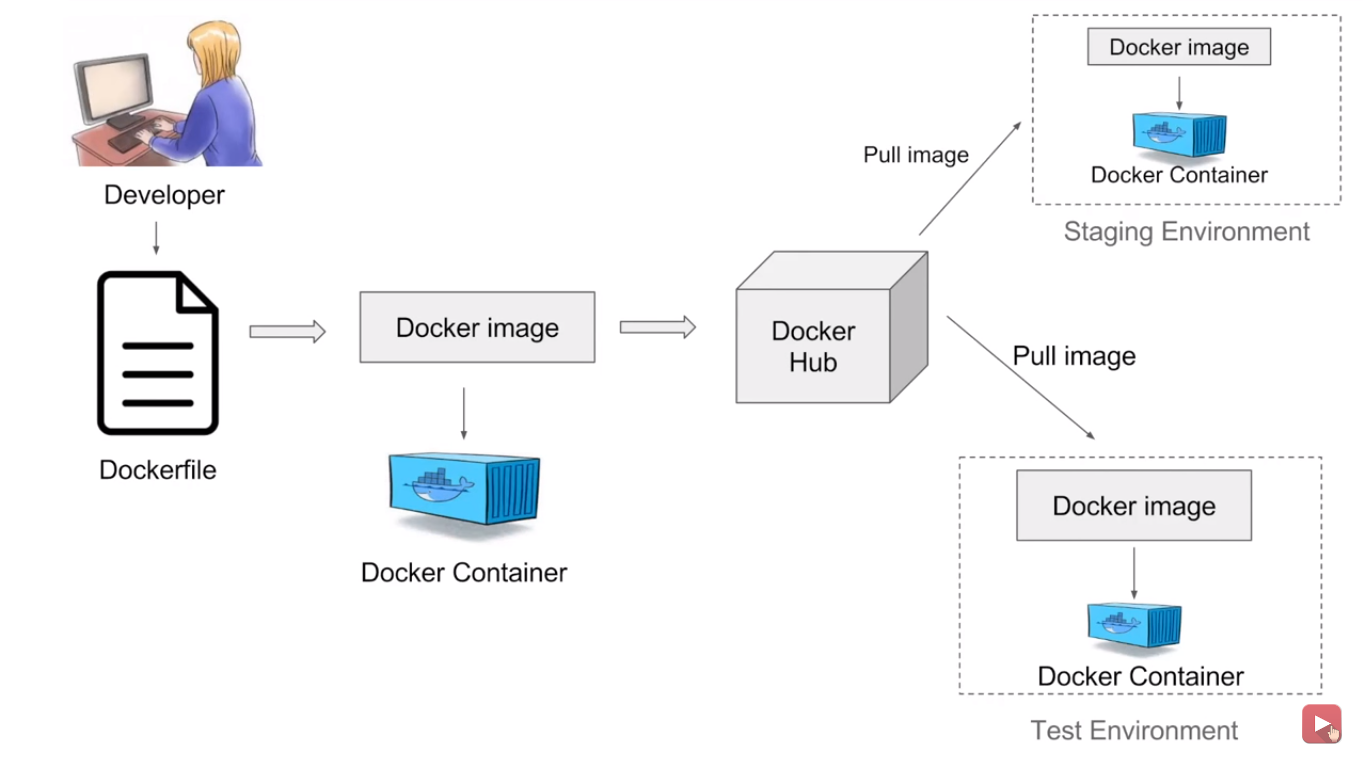
**Docker Hub**: It is a central repository for docker where all the images are available, and we can pull and push the images.

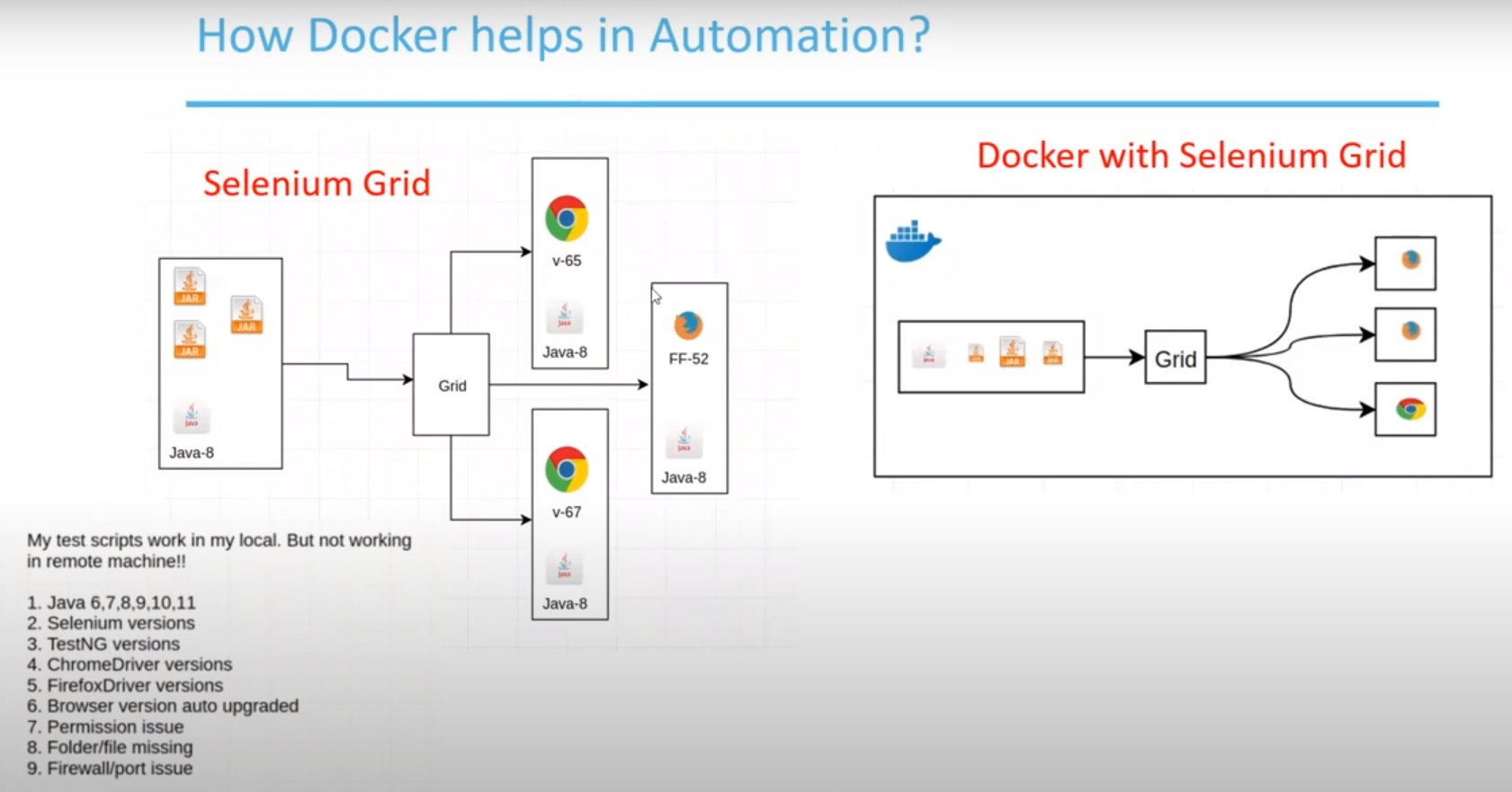
**Docker container**: A container is a standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another.

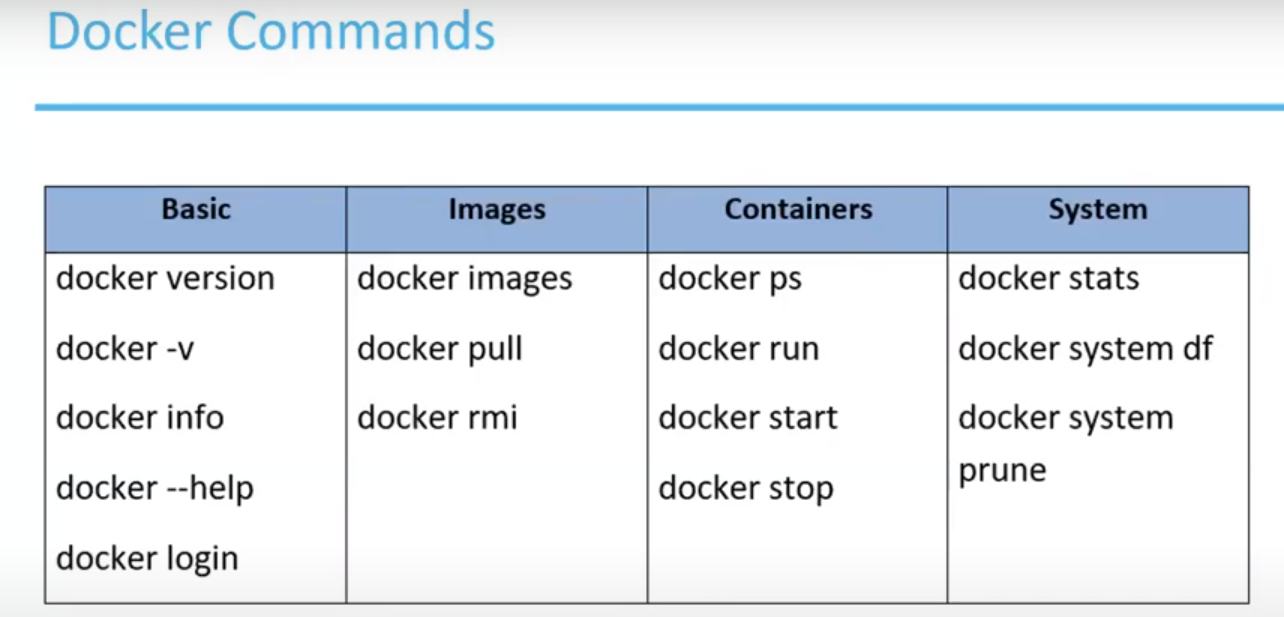
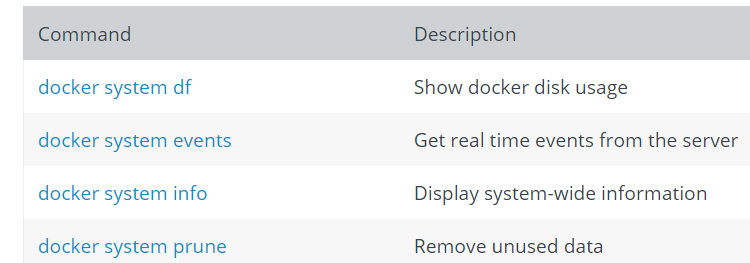
**Docker Images**: A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.









docker run -it imageName : it indicates interactable

ps- process status

**Pull the image: selenium/standalone-chrome**

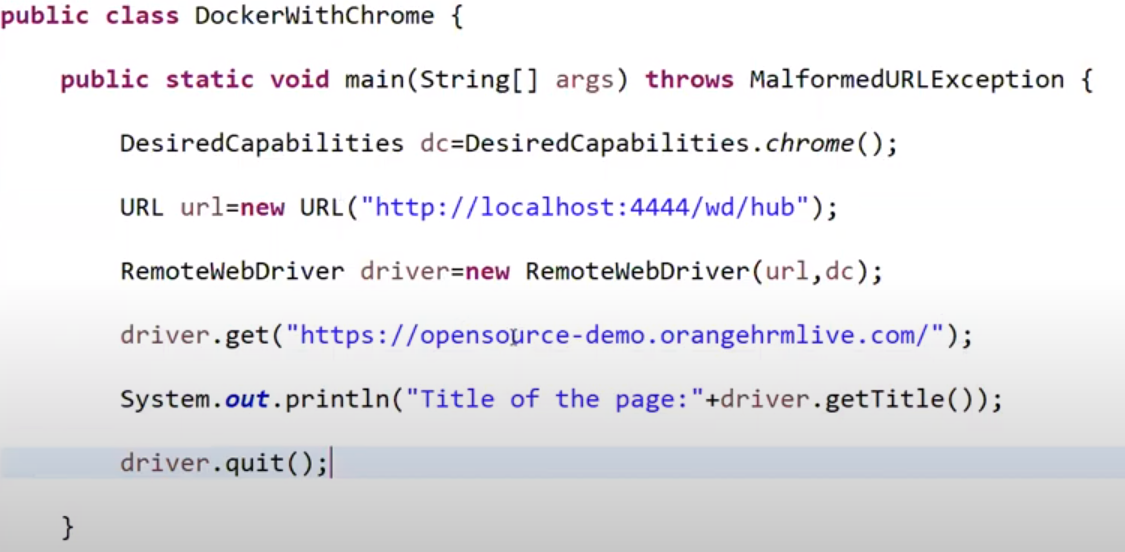
**Run the container:**

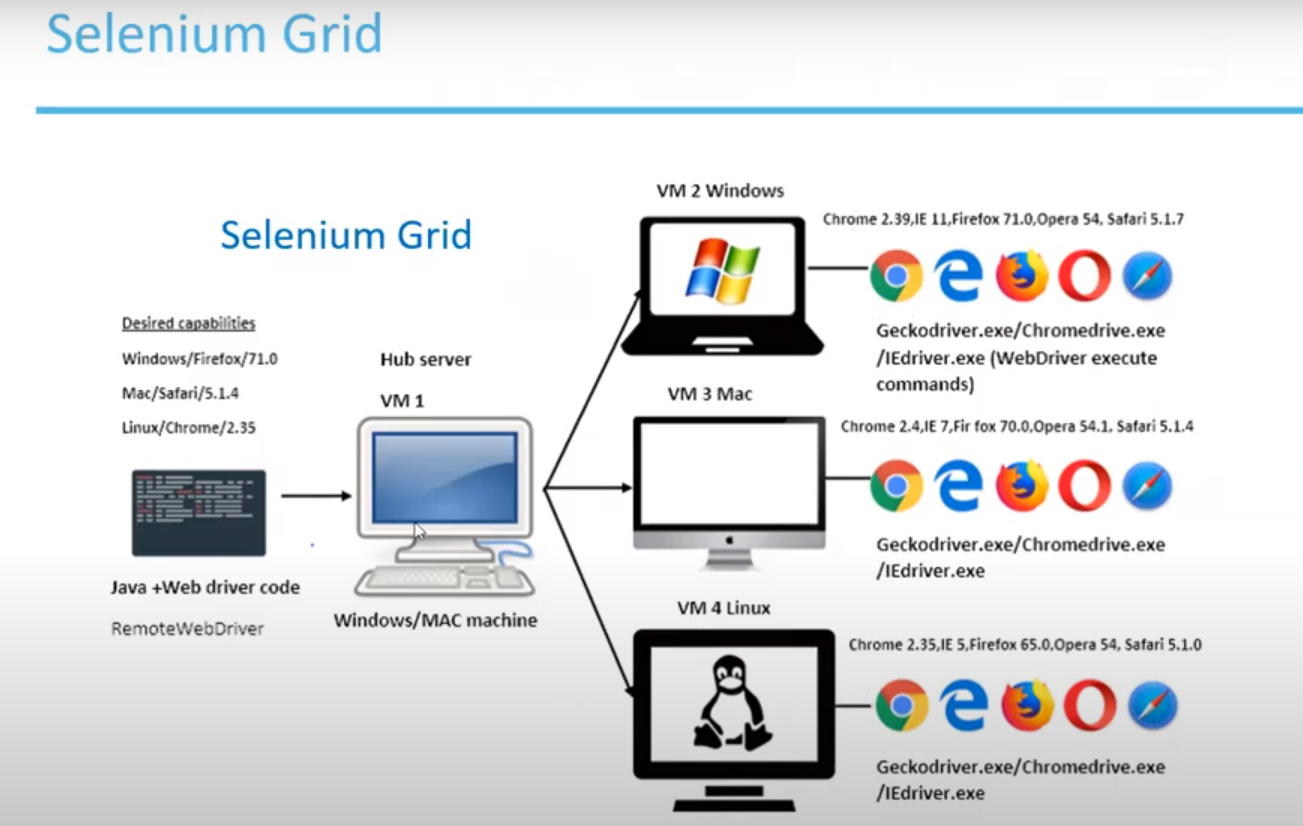
docker run -d -p 4444:4444 -v /dev/shm:/dev/shm selenium/standalone-chrome:latest

-d indicates: will run in backend

-p: port number

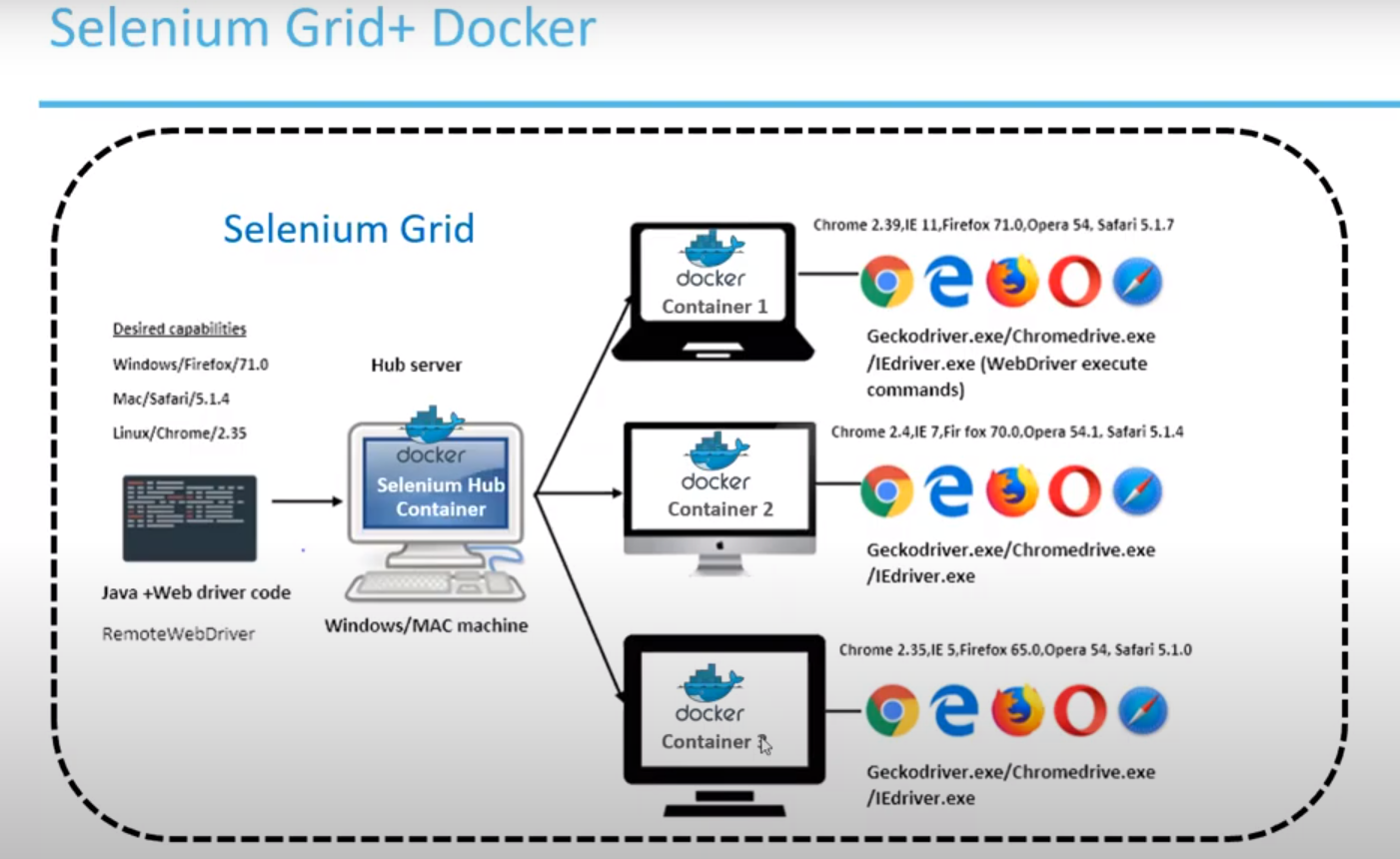
**Note**: Container should up and running for execution.





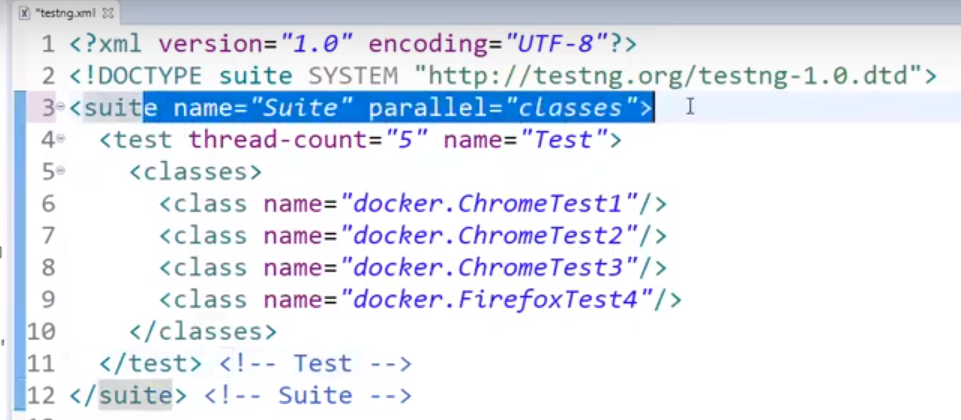
As per the desired capabilities in our code, hub will routed our execution to respective node.We have 4 VM’s (1- Hub, 3-Nodes) and each VM should have proper configuration and installations then only our script will execute, and this is the biggest disadvantages.

­



Here all the containers are created by images and this is in single VM only.

* Now we need to create this Hub-Node environment, and this will do using **docker-compose.yaml** file and this will create container and attach nodes to hub.
* Go to file location and run this file using: **docker-compose up**
* Go to grid console and check grid is running and nodes are associates to hub.
  + <http://localhost:4444/grid/console>
* We can increase number of nodes: **docker-compose scale chrome/firefox=3**
* Execute your test.
* To stop the grid and cleanup the container: **docker-compose down**



But here also we are starting and stopping docker manually.

Now we will create bat files and execute through code using java.

We need 3 files:

* docker-compose.yaml
* start\_docker.bat:
  + go to location of compose file and run **docker-compose up**
* stop\_docker.bat:
  + go to location of compose file and run **docker-compose down**

