Dev Ops

Dev Ops is neither a technology nor a tools, it is a concept which helps development team and Operation team (production team) to communicate to develop & deploy the applications, these teams would use some tools to develop & deploy

* Git: Version controlling
* Jenkins/Bamboo: CI - CD pipeline
* Docker: It is a container that can ship your application to any environment
* AWS: Cloud system where you deploy the application

Git:

It is centralized version controlling system, used to share and collaborate other people work, Git provides a Repository where you can have all your work, the repository will track all the changes each person does with some unique id i.e., called as snapshot will have hash number.

If you use GIT you can get updates done on any time, i.e., if developers had written some code and updated the GIT later they write more lines of code & feel that the old code is required, then they can get that old code back if its committed.

Git has many commands

git init: To create local repository

git pull: To pull the changes to local repo from the remote repo

git push: To push the local changes to the remove repo

git branch branchName: To create a new branch

git checkout branchName: To get inside the branch

git merge destinationBranchName: To merge the destinationBranch with the currently checked in branch

git add: to add the changes that needs to be committed

git commit: to commit the changes

git log: To see history of commits

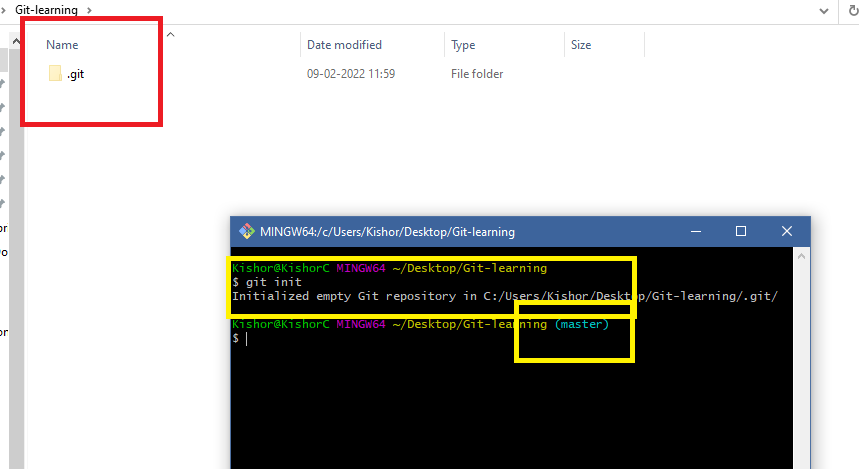
git clone: To clone the remote repository

Ground Rules when you are working in GIT

1. Create a custom branch if you want to update your repository, even a small modification also you have to do in custom branch not in the master branch
2. From local repository push the custom branch to the remote repository not the master branch
3. In Remote repository merge the master with the custom branch, if git doesn’t give conflict error you are free to merge, if you get a conflict close the merge request and update the local repository with the remote repository through pull request
4. Merge the remote repository changes with local repository changes if there’s a conflict using pull command and then push the changes to the remote.

There are two ways you can create local repository in GIT

1. git clone: Creates local repository from remote repository, it will have link to the remote by default
2. git init: Creates local repository manually, but will not have link to the remote repository by default, you must link it through some git command.



By default you have a branch called master branch.

Important commands

git branch branchName

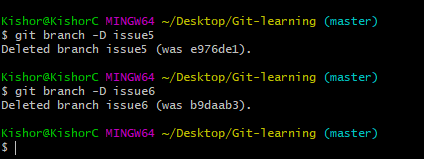
git checkout branchName

git merge branchName

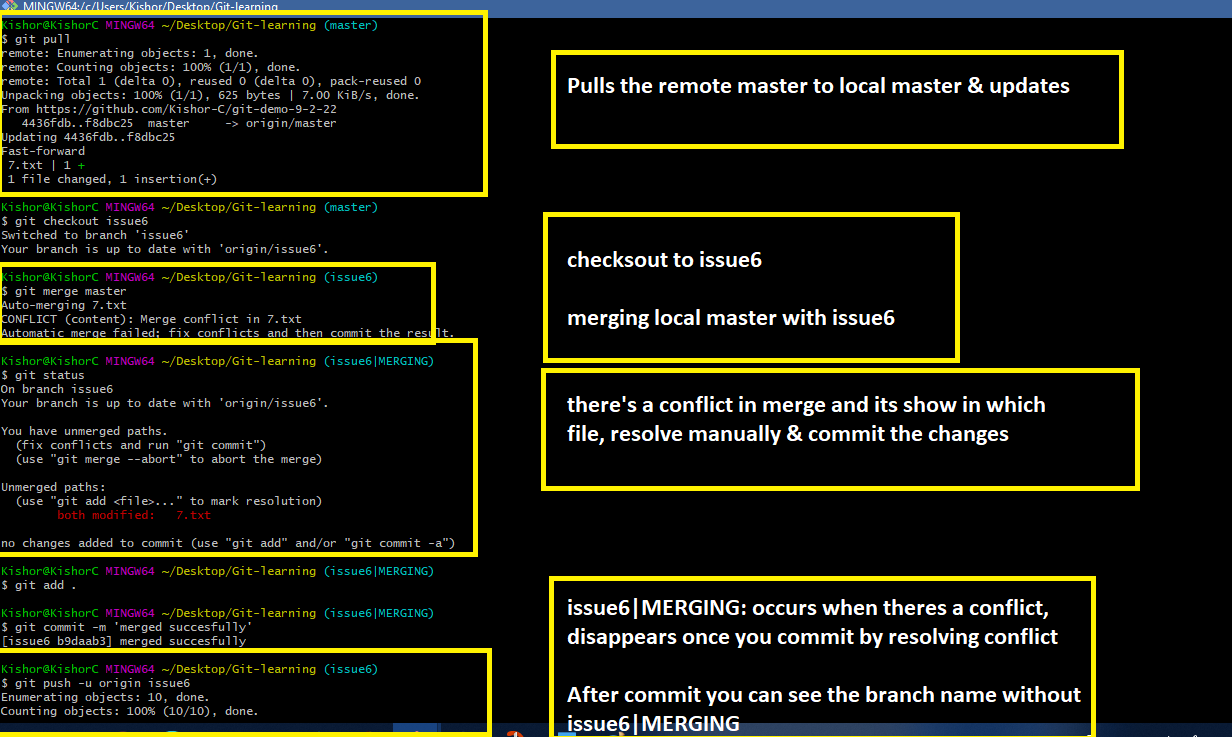
git pull

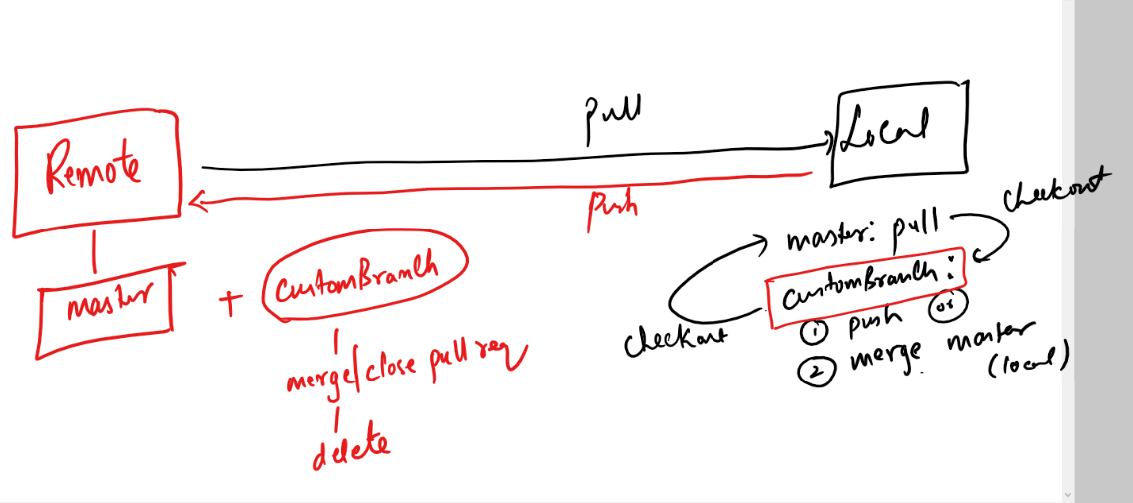
git push -u origin branchName

git branch -d branchName: Deletes the branch



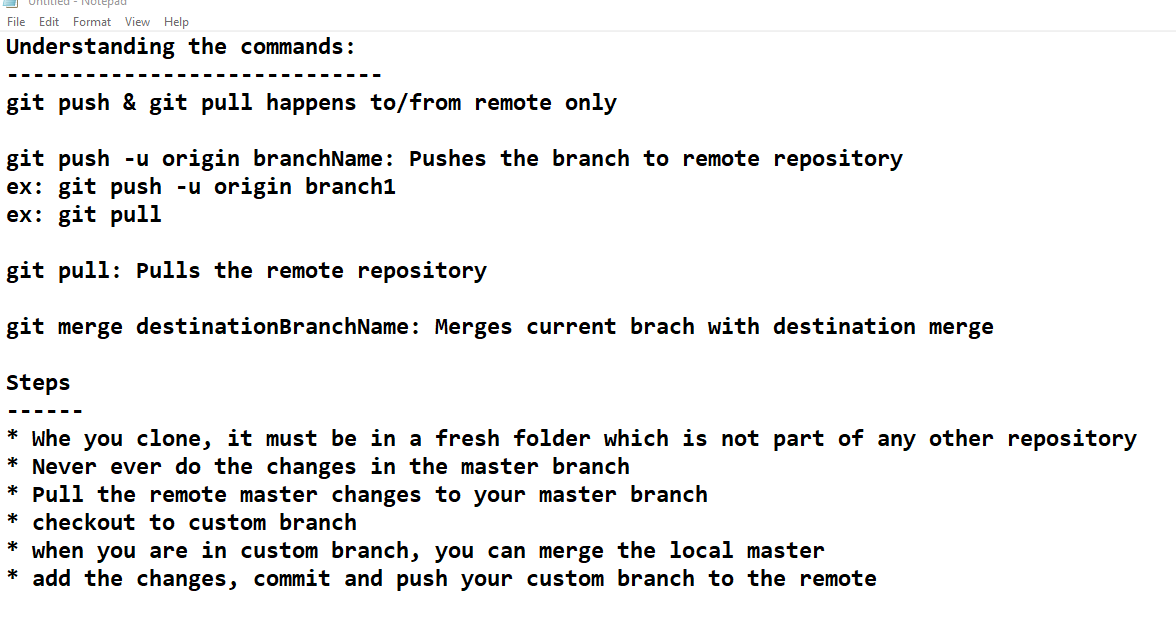
When conflicts to occur

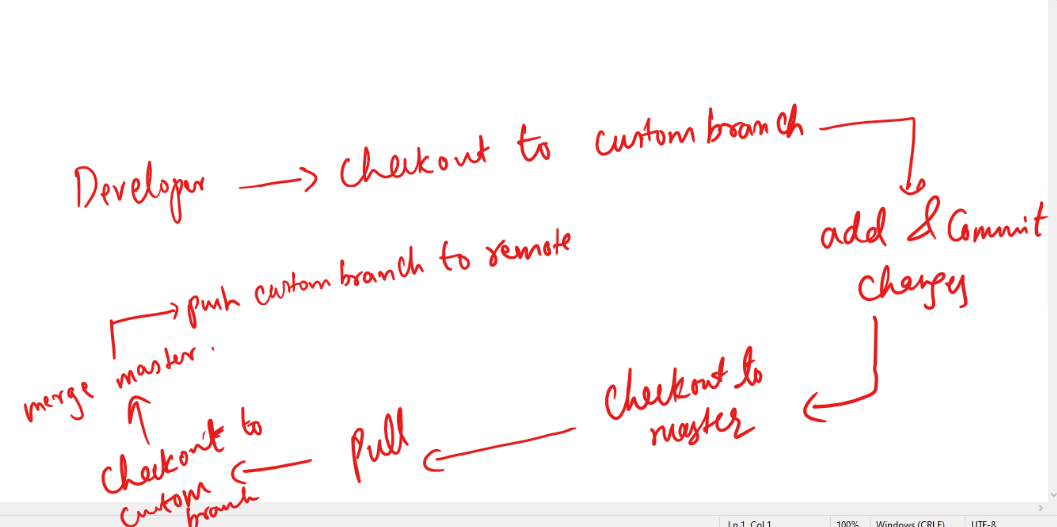




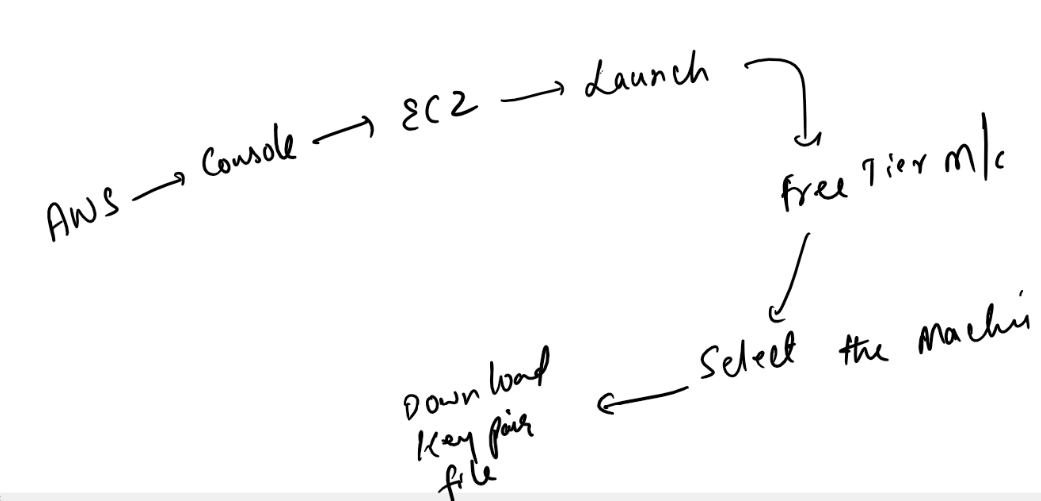
Git Organization:

Here you can create a repository where multiple members can collaborate to the project





AWS



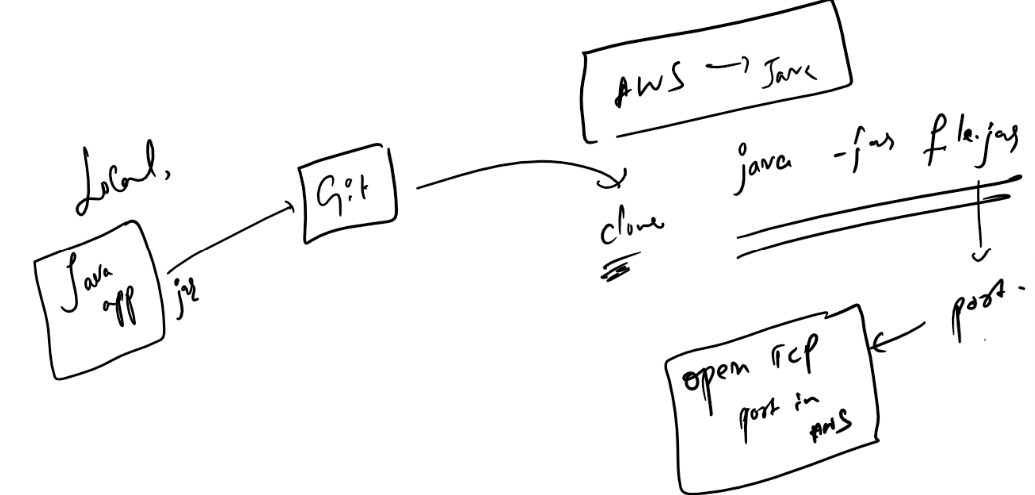
How to install git in AWS EC2 linux

>> sudo yum install git -y

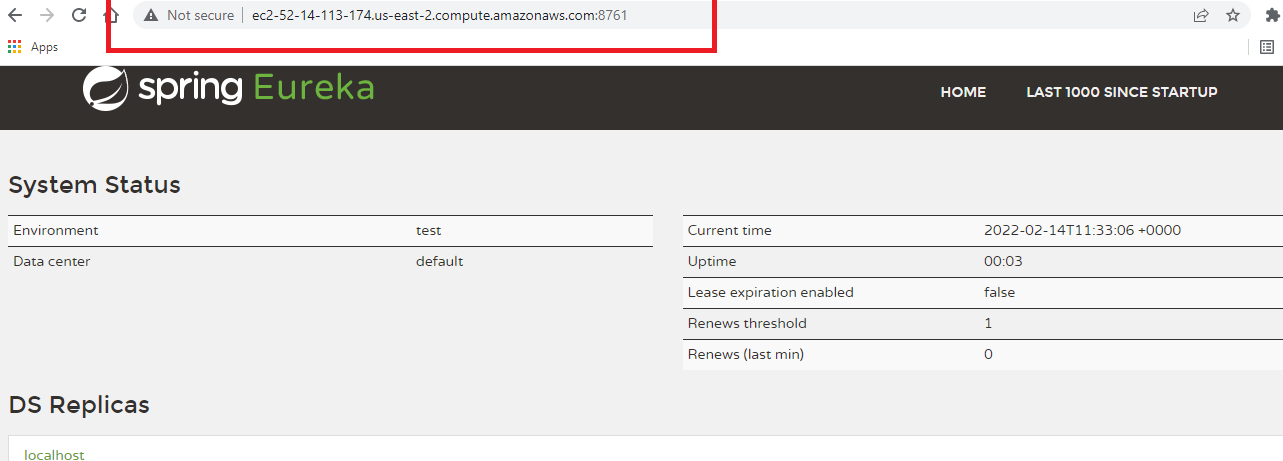
How to install java in AWS EC2 linux

>> sudo yum install java-1.8.0-devel -y

You can create an executable jar and run on your aws ec2, firstly you must push the jar to the git from your local machine, then you need to pull the jar from git from the cloud machine



Note: AWS machines port are not available to public by default, hence you must configure TCP port in the security of the AWS instance to access that port.



Docker:

It is a way to ship and run applications in a container without installing any external softwares.

Containers: Docker containers are the environment that will have the applications & the softwares to run the application, these are self-contained containers, which are independent from other containers

Dockerfile: This is a text file that will have the instructions to run the application in the docker container.

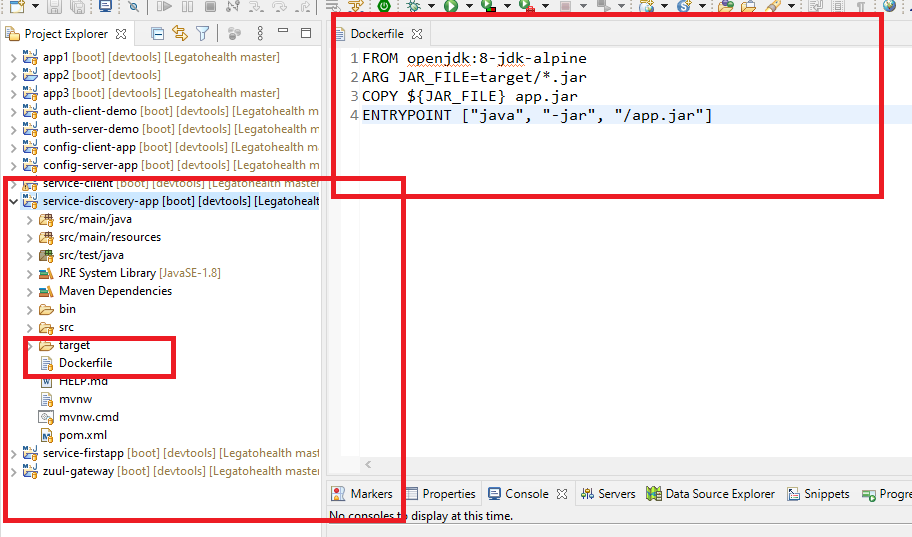
Docker Image: This is an executable file that is a template for the docker container, this will instantiate the container, this is created form the Dockerfile

Docker Hub: This is the public website that maintains the docker images, you must have an account to push / pull docker images

Note: If you have a docker, you don’t have to install any softwares

Docker provides some commands to build docker images, run docker images, push/pull docker images

Dockerfile: it must be created in the project folder



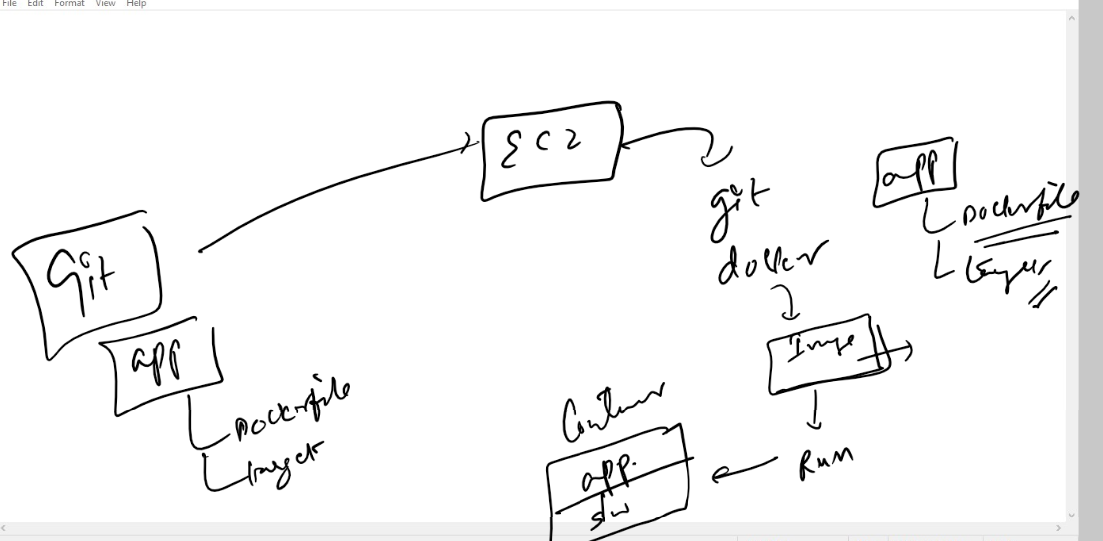
Now the Dockerfile has the instruction that what is the software required to run the application, and the command to run the application i.e., java -jar app.jar,

app.jar: it is the file created when you build the docker image, from the target folder

Steps:

1. Create the application
2. Create a Dockerfile (No extensions)
3. Write the instructions in the Dockerfile
4. Build the docker image
5. Run the docker image

Note: To build & run the docker image we need Docker installed, since in windows it takes more than 20mins to install docker, we can use EC2 linux to install docker.



Now if you have an Docker image you can push to Docker hub and get that image in multiple machines and run that image, so that in none of the machines you are installing any softwares except Docker.

How to install docker in the AWS ec2

>> sudo yum install docker -y

>> sudo service docker start

The start would start the docker software

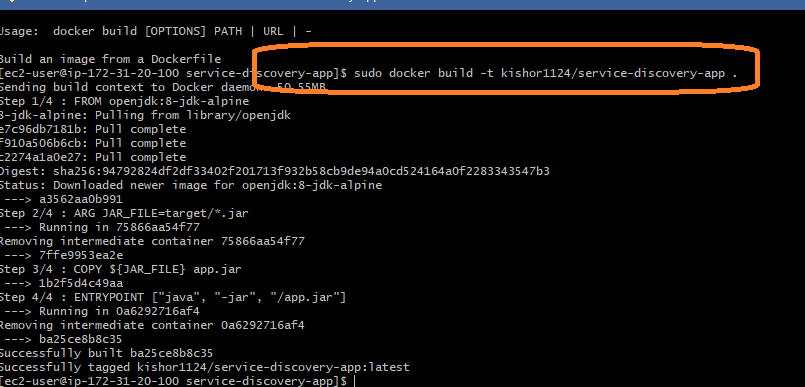
>> sudo docker info

The info would provide docker information

Note: To build the image you must be inside the folder that has Dockerfile

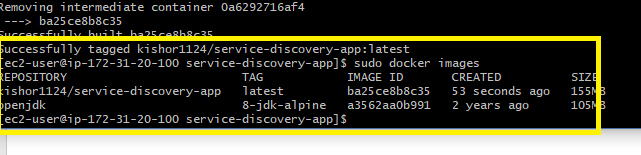
>> sudo docker build -t kishor1124/service-discovery-app

The above command builds the docker image, you can push or run or pull this image



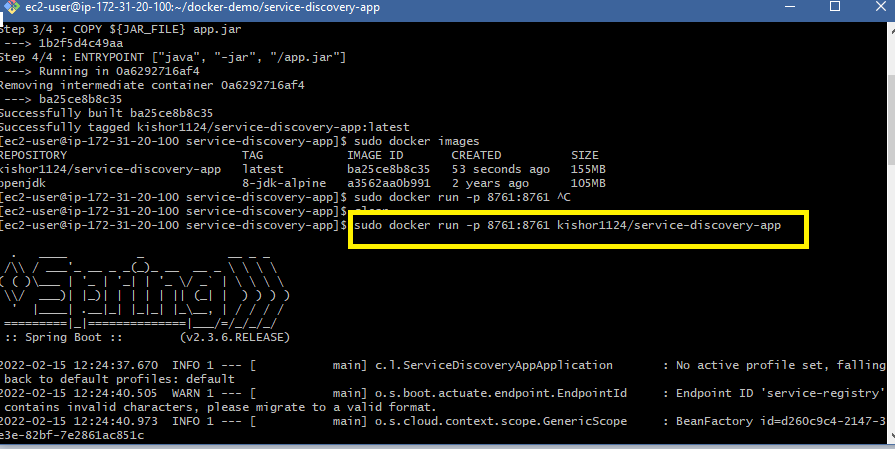
Now you can see the docker image created using sudo docker images

>> sudo docker images



How to run the image

>> sudo docker run -p publicPort : privatePort imageName



Here 8761:8761 means the 1st port is to expose the application to the public, the 2nd is the port that is running inside the container of the docker, because of application.properties

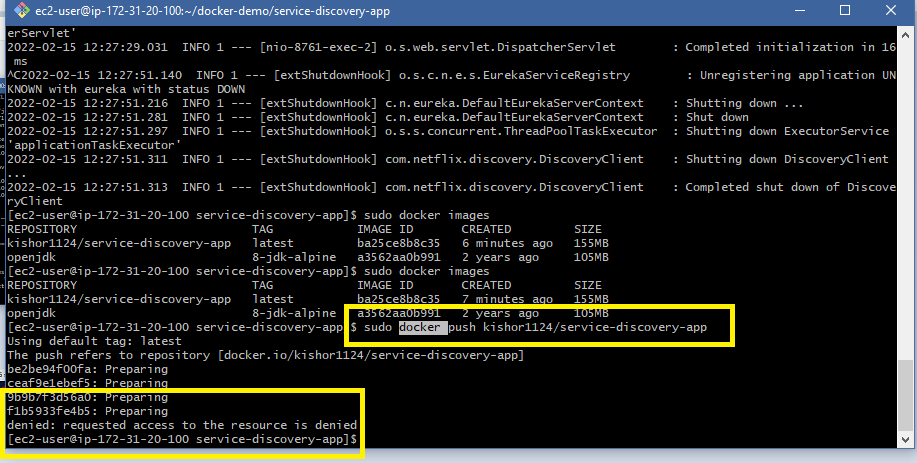
How to push the docker image

>> sudo docker push image-Name

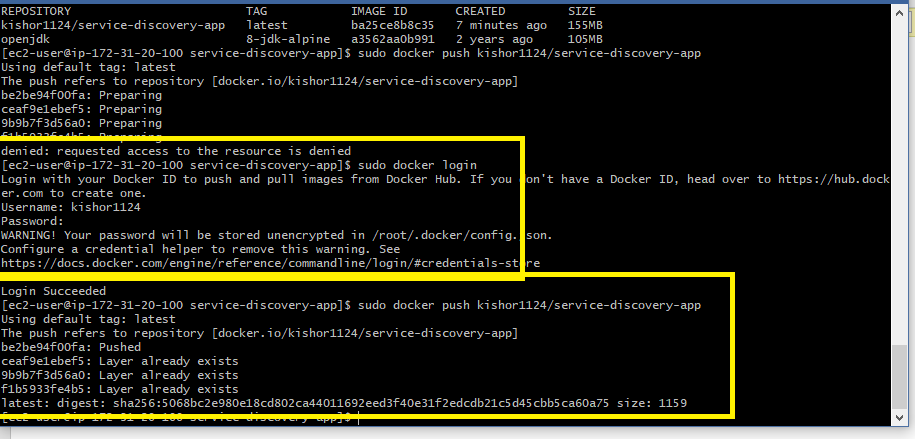
i.e.,

>> sudo docker push kishor1124/service-dicovery-app

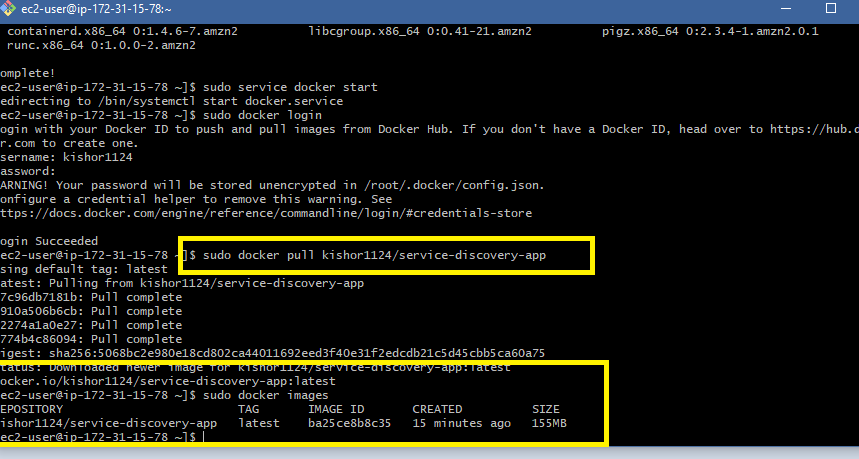
Note: To push the docker image you must login to docker hub from the terminal



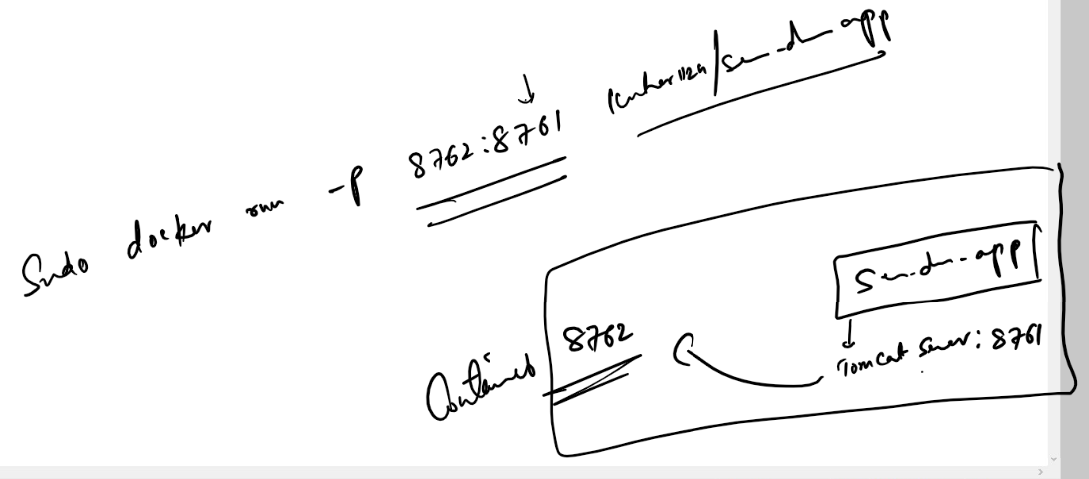
Logging in to docker account

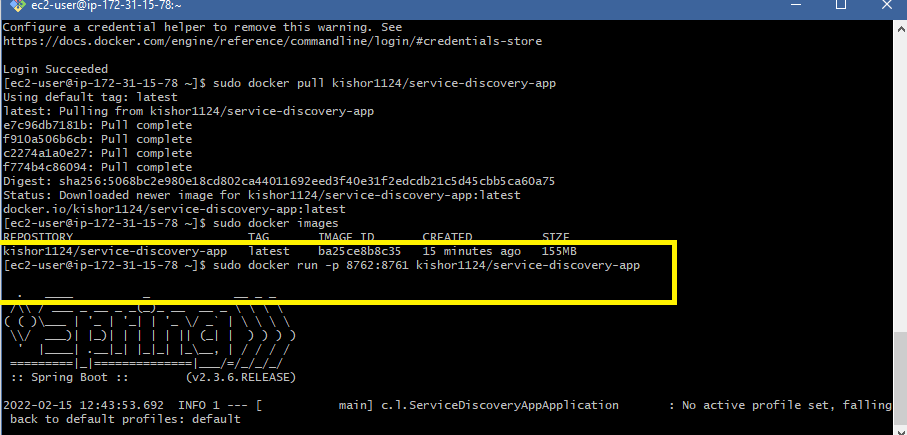


Now you can pull the docker image run in multiple machines



Running the docker image





Now you have exposed your private port to 8762,

