Cloud computing:

Agility

High Availability and reliability

High Scalability

Multi-Sharing

Device and Location independence

Maintenance

Low cost

Services in the pay-per-use mode

Advantages:

Back-up and restore data

Improved collaboration

Excellent accessibility

Low maintenance cost

Mobility

Services in the pay-per-use model

Unlimited storage capacity

Data Security

Disadvantages:

Internet connectivity

Vendor lock-in (different vendors, different services)

Limited control (control is with the service provider)

Security (you will be sending all sensitive information to a third party service provider)

History:

In 1999, Salesforce.com started delivering of applications to users using a simple website. The applications were delivered to enterprises over the Internet, and this way the dream of computing sold as utility were true.

In 2002, Amazon started Amazon Web Services, providing services like storage, computation and even human intelligence. However, only starting with the launch of the Elastic Compute Cloud in 2006 a truly commercial service open to everybody existed.

In 2009, Google Apps also started to provide cloud computing enterprise applications.

In 2009, Microsoft launched Windows Azure, and companies like Oracle and HP have all joined the game

Cloud computing architecture:

<https://www.javatpoint.com/cloud-computing-architecture>

Cloud enabled Technologies

DevOps

GIT

JIRA

Jenkins

Maven

<https://aws.amazon.com/what-is/cloud-native/>

Docker:

We can create containerized applications that can run in any computer which has docker.

Container provides all the necessary components like runtime, drivers, etc to run an application in any computer where docker is present.

Demo:

Create a spring boot application and create a docker image of that application.

That image can be run locally.

That image can be deployed in a hub.docker.com or aws ecr or any cloud

To create a docker image of an application:

1. Create a Dockerfile in the project
2. What this Dockerfile should contain?

If it is a spring boot project, an easy step to create docker image is using mvn command (no need of Dockerfile)

mvn spring-boot:build-image

1. Generate .jar file

Go to cmd where pom.xml is present:

mvn clean install

1. Check targer folder that it contains a .jar file now
2. Create a docker image of this project:

mvn spring-boot:build-image

we have got a docker image: 07-jun-h2-demo:0.0.1-SNAPSHOT

docker images

docker image ls

docker container ls

docker ps

docker rm <<container id>> -f

docker rmi <<image id>>

How to create docker image of an application using docker file:

1. Create a spring boot project
2. Create a Dockerfile in location where pom.xml is present. If you right click the project and create file, it will be there only.
3. Inside Dockerfile:

FROM java:8

EXPOSE 5000

ADD target/13-rest-4-docker-1-0.0.1-SNAPSHOT.jar 13-rest-4-docker-1-0.0.1-SNAPSHOT.jar

ENTRYPOINT ["java","-jar","13-rest-4-docker-1-0.0.1-SNAPSHOT.jar"]

1. Note the .jar file names in 3 locations are the .jar file we generated using

mvn clean install (in cmd) or

right click project Run As -> Maven Build (goals: clean install)

1. Now, lets create docker image: go to cmd where Dockerfile is present

docker build -t 3-rest-4-docker-1 .

now, the docker image is created.

1. To push this image to hub.docker.com,

We need to create a repository (same name is good)

Tag the local image to docker hub repository:

docker tag 3-rest-4-docker-1 jagindia/jag-ust-demo

push the image to repository:

docker push jagindia/jag-ust-demo

1. docker images
2. run docker image:

docker run -p 5000:5000 jagindia/jag-ust-demo:latest

1. go to browser:

localhost:5000/hi

Introduction to Cloud computing – Cloud concepts, SaaS / PaaS / IaaS, Private / Public / Hybrid Clouds  
Concepts of Cloud Native Development, Virtualization, Cloud Deployment

SaaS

PaaS

IaaS

SAAS

Software As A Service

Ex: Office365, Zoho, Google Workspace

PAAS

Platform As A Service

Ex: Oracle Cloud Platform, Microsoft Azure, Google App Engine

IAAS

Infrastructure As A Service

GCP, AWS EC2, ECS, EKS