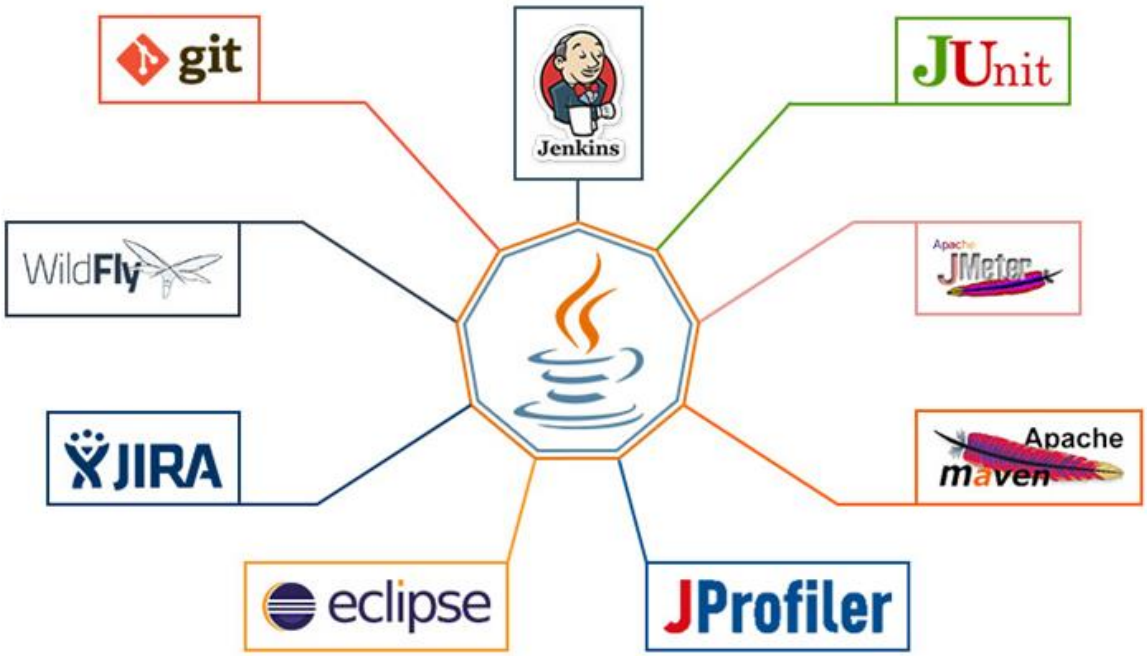

git
Jenkins
JUnit
WildFly
Apache JMeter
JIRA
Apache maven
eclipse
JProfiler

INDEX

Tools Used in Modern Build and Deployment ----------

1. Introduction 04
2. Required Tools during Development 04
3. Diagram 05
4. Different Roles during Development 06
5. Java Full Stack Developer 07

Tools Used in Modern Build and Deployment

Prepared By - Nirmala Kumar Sahu

Introduction

- Keeping our code ready to execute is called Building the application/ project.
- Deploying in the server and keeping in cloud environment and registering with DNS to make project/ application available for all users is called Deployment.
- DNS: Domain Naming Server/ Service
  E.g., GoDaddy, Jelastic, Cloud Flare, Heroku and etc.

Two types of Software Projects

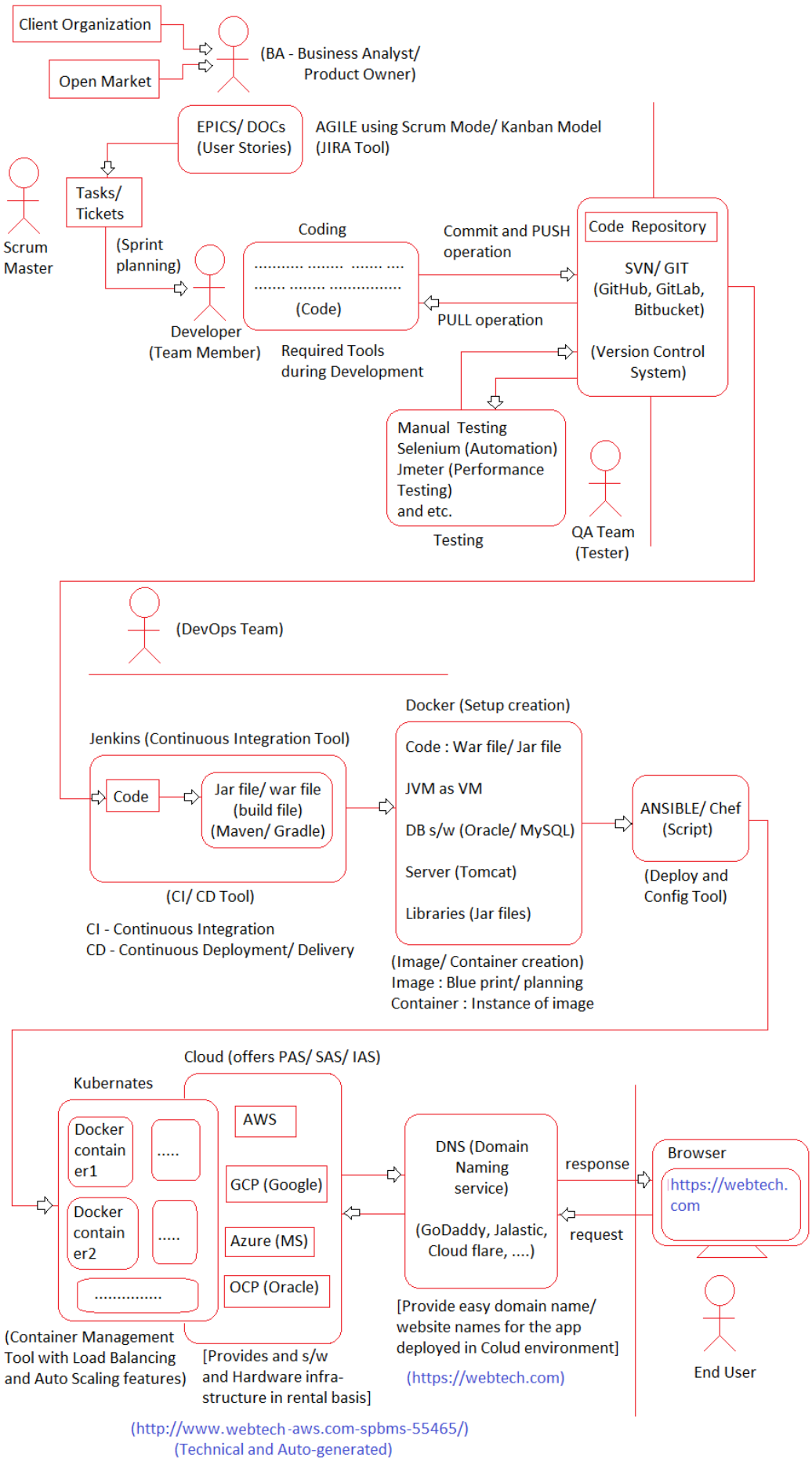
a. Services Projects (The Project will be developed according to the requirements of certain client organization)
   E.g., Citi bank project for Citi Bank by Polaris, Amex project for American Express bank by CTS

b. Products Projects (The project will be developed by software company directly based on the gap/ need available in market and will be kept for sale for all the client organizations)
   E.g., Tally, MS Office, Zoom, Class Plus, Postman and etc.

Required Tools during Development

- IDE (Eclipse/ STS/ IntelliJ) - For writing the code
- Build Tool (Maven/ Gradle) - For getting jars/ Project structures
- Logging (SLF4J/ Log4J/ Common Logging) - For logging activity, to keep track code flow
- Debugging (Eclipse Debugger)
- Unit Testing (JUnit, HttpUnit, Mockito)
- Code Coverage (Jacoco) - Keeps track of how much percentage of code involved in the given situation.
- Sonar Lint - To check standards and quality code
- POSTMAN - For webservices/ API testing
- Swagger - For web services/ API documentation
- Local server or Embedded server (tomcat/ jetty) - To run application locally
- lnMemory DB software or Local DB software
- PuTTY - To connect to Remote servers available in Linux OS
  and etc.

Diagram


Client Organization
Open Market
(BA - Business Analyst/ Product Owner)
EPICS/ DOCs (User Stories)
AGILE using Scrum Mode/ Kanban Model (JIRA Tool)
Tasks/ Tickets
Scrum Master
(Sprint planning)
Developer (Team Member)
Coding
(Code)
Commit and PUSH operation
PULL operation
Code Repository
SVN/ GIT (GitHub, GitLab, Bitbucket)
(Version Control System)
Required Tools during Development
Manual Testing
Selenium (Automation)
Jmeter (Performance Testing)
and etc.
Testing
QA Team (Tester)
(DevOps Team)
Jenkins (Continuous Integration Tool)
Code
Jar file/ war file (build file) (Maven/ Gradle)
(CI/ CD Tool)
CI - Continuous Integration
CD - Continuous Deployment/ Delivery
Docker (Setup creation)
Code : War file/ Jar file
JVM as VM
DB s/w (Oracle/ MySQL)
Server (Tomcat)
Libraries (Jar files)
(Image/ Container creation)
Image : Blue print/ planning
Container : Instance of image
ANSIBLE/ Chef (Script)
(Deploy and Config Tool)
Cloud (offers PAS/ SAS/ IAS)
Kubernates
Docker container1
Docker container2
AWS
GCP (Google)
Azure (MS)
OCP (Oracle)
(Container Management Tool with Load Balancing and Auto Scaling features)
[Provides and s/w and Hardware infra-structure in rental basis]
DNS (Domain Naming service)
(GoDaddy, Jalastic, Cloud flare, ...)
response
request
Browser
https://webtech.com
End User
[Provide easy domain name/ website names for the app deployed in Colud environment]
(https://webtech.com)
(http://www.webtech-aws.com-spbms-55465/)
(Technical and Auto-generated)

Different Roles during Development

Role 1: (Business Analyst/ Product owner)

- By talking to client organization or by open market outsource prepares EPIC documents representing the requirements.

Role 2: (Scrum Master)

- Based on the user-stories of EPIC docs, he creates tasks/ tickets and gives to team member as part Sprint planning, also conducts daily standup meetings, spring reviews and etc.

Role 3: (Developer)

- Develops the based on given tasks/ tickets using IDEs.
- Creates the project directories and gets jar files using Maven/ Gradle build tools and uses the required tools as given above.

Role 4: Tester

- Collects the code from Performs both Manual testing and Automation testing and generates Test reports.

Role 5: DevOps Team

- Creates repositories, accounts in GIT/ SVN code repository.
- Builds the war/ jar file representing project using Maven/ Gradle tool.
- Configures the Jenkins tool for Continuous Integration to prepare build file for every code change that takes place in GIT/ SVN repository.
- Creates the Docker Image specifying environment required to execute the project like DB, VM, jars, Server and etc.
- Keep Cloud infrastructure ready to deploy the project using AWS/ Azure and etc.
- Uses the Kubernetes tool for creating Docker Containers based Docker images having features of load balancing and auto scaling (increasing or decreasing Docker containers availability based on number of users/ requests).
- Writes Ansible scripts as Deploy and Configuration scripts telling what Docker image should be used and where it has to be deployed in cloud server like specifying port numbers, server names, memory details and etc.
- Uses DNS to provide easy domain name/ website name for accessing Cloud deployed application.

Role 6: End-user

- Uses the browser or other tools to request application that is deployed cloud through DNS server.

Java Full Stack Developer

- To become a Java full stack developer, we must have strong knowledge on following things.
  - Java language, technology and frameworks, tools
  - Testing
  - DB s/w
  - UI technologies
  - DevOPs
  - AWS (cloud)

---------- The END ----------