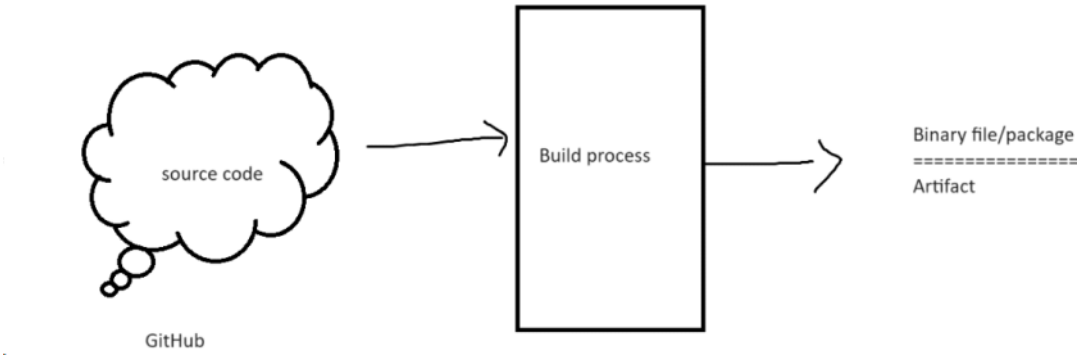
Jfrog-Day-1

29 November 2023 14:55

- In this session we are going to talk about
  - what is an Artifact?
  - what is an Artifact repository?
  - Types of Artifact repositories?
  - Jfrog Artifactory?
  - How to install the JFROG Artifactory on ec2
- **First of all can you tell me what is an Artifact?**



  - It's a binary file or package which is generated after build process completed on the source code.

Java ---> Maven(Build) --> .jar/.war/.ear

.Net --> MSBuild --> .exe/.msi

  - Suppose just assume your Java source-code is available on the GitHub after the build process the binary files will get created & the format of the files like JAR/WAR/EAR files.
  - Similarly the .Net source code will generate the binary files in the format like .exe/.msi packages after the build process gets completed.
  - In general the binary files that are created after the build process created we can refer them as artifacts.
  - So now the files like JAR/WAR/EAR/MSI/EXE we can call as a artifacts.
  - Normally whenever there is a change in the suits that is presented in the GitHub an Artifact will get generated.
  - **What we will do with this Artifact?**
    we will deploy this Artifact in DEV/QA/PROD environments.
  - Mostly the developers will check-in latest code into GitHub many times in a day, for every latest check-in made into GitHub repository an Artifact will get generated. The Artifact whatever is created will do deployment in to the DEV environment.
  - Just Imagine the application was working without any issues before the deployment. But after the latest Artifact deployment application is not working correctly.
  - **How do you fix it?**
    - We have to revert the changes in code & get the Artifact for reverted changes & deploy it. This is one way and its time taking process.
    - Maintain storage for artifacts & deploy previous artifacts whatever needed.
  - **Which way we will normally after in this to options obviously will go with the second option only correct or not? -** Correct
- Now Artifact repository tool will help us to take a backup of artifacts that are created as part of the build process for ever latest check-in in GitHub. So if there is a requirement to rollback your environment to the previous Artifact version, you can simply pull that Artifact from the Artifact

repository and you can make it your application working successfully.

- Artifact repository is not only used for storing the binaries but it is also provide dependence libraries that are required to complete build process successfully.
  - Normally when we run mvn commands the binaries are downloaded from internet instead of that we can download it from Jfrog.
  - Now let's see this practically
    - When you run **mvn install** command the Maven command connects to the central repo which is on the Internet & download the libraries that are needed to build your application.
    - In some organisations it is not allowed to connect directly to the internet and download the required packages.
    - In those situations in order to get the binaries the Maven command will connect to the Jfrog artifactory and Jfrog will get will get in sync with the central repository which is stored on the internet.

- In the market there are number of Artifact repository tools are there
  - Helix
  - Pulp
  - Nuget
  - Jfrog
  - Nexus
  - Docker registry

- In this set of tools we are going to discuss about the JFROG artifactory tool.

- Jfrog artifactory tool mainly available in two editions
  - Pro-edition
  - Open source solution

- Now we'll see **how to install the open source solution model Jfrog artifactory on Amazon Linux ec2 machine.**
  - **Prerequisites**
    - Create EC2 with instance type of t2.small
    - Allow the ports
      - 8081
      - 8082
    - Java
    - Maven
    - Git
  - **Installation steps**
    - There two types of installable available for Jfrog
      - pro installer --> It's Licensed version
      - oss installer --> It's a free open source solution
    - Login to EC2 with root user & install java
      - yum install java-1.8* -y
    - Download the artifactory installers into /opt
      There two types of installable available for Jfrog
      - pro installer --> It's Licensed version
      - OSS installer --> It's a free open source solution

      Now let's install OSS installer

      cd /opt
      wget -O https://releases.jfrog.io/artifactory/bintray-artifactory/org/artifactory/oss/jfrog-artifactory-oss/6.23.42/jfrog-artifactory-oss-6.23.42.zip
    - Extract the archive
      unzip jfrog-artifactory-oss-6.23.42.zip
      mv jfrog-artifactory-oss-6.23.42 jfrog
    - set **JFROG_HOME** variable
      echo "export JFROG_HOME=/opt/jfrog" > /etc/profile.d/jfrog.sh
      source /etc/profile/jfrog.sh
    - Run a start-up script
      cd /opt/jfrog/app/bin/
      ./artifactory.sh start
    - Access the artifactory from browser

http://<dnshostname>:8081

- Default credentials to login jfrog artifactory
  user: admin
  password: password

- Change the default password

- Here we can create different type artifact repositories based on the source code,
  - If we have **java** source code we can create **maven** repository
  - If we have **php** source code we can create **php** repository
  - If we have **python** source code we can create **py** repository
  - If we are **not sure about the source code** we can create **Generic** repository

- In Jfrog main we will have 3 types of repositories
  - **Local** --> Local repositories are the place to store artifacts generated as part of build
  - **Remote** --> This repository get in sync with central maven repository
  - **Virtual** --> It's a combination of local + remote repositories

  In Local repositories we can see
  libs-snapshot-local
  libs-release-local

  In Remote repositories we can see
  jcenter

  In Virtual repositories we can see
  libs-snapshot
  libs-release

- **UC: Create a maven local repository with name "online-bookstore-local" for online-bookstore java application**.
  - This will help us to store different versions of artifacts like
    - jar
    - war
    - ear

- **UC: Create maven remote repository with name "jcenter"**
  - This repository will help to download the dependencies from jcenter repo from Jfrog instead of the central repository in Internet

- **UC: Create maven virtual repository with name "online-bookstore"**
  - online-bookstore-local + jcenter

- Check Artifact repository browser & we don't see any artifacts on those repositories, those are empty.