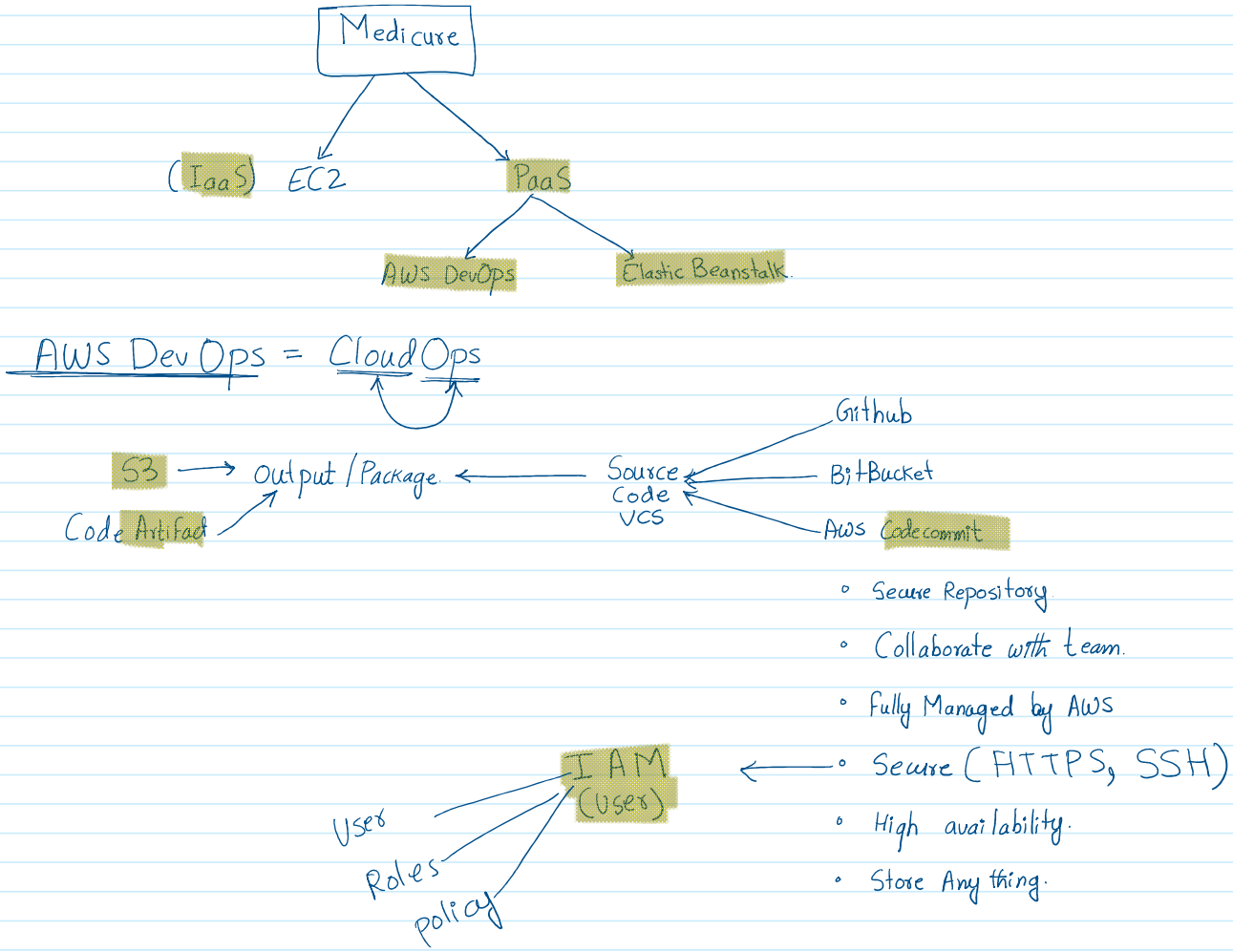
64.04Jul.Project(CloudOps - Medicure)

Tuesday, July 4, 2023 7:12 PM



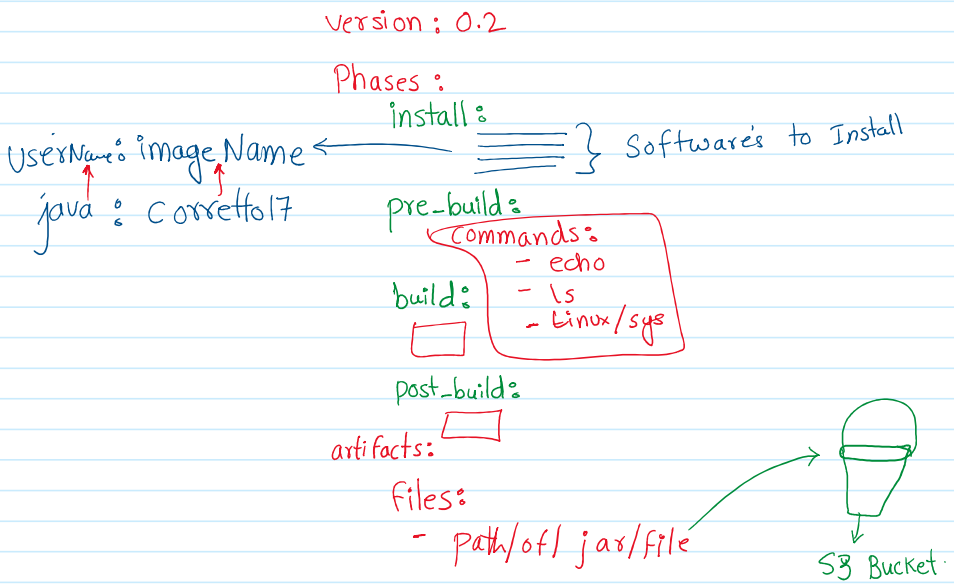
1. Create IAM User
    a. User name:
    b. Attach Policy Directly:
        i. AmazonEC2FullAccess
        ii. AmazonS3FullAccess
        iii. AWSCodeBuildAdminAccess
        iv. AWSCodeCommitFullAccess
        v. AWSCodeDeployFullAccess
    c. CodeCommit Generate User Credentials
        i. Select dev2 user---> Security Credentials --> HTTPS Git credentials for CodeCommit
        ii. UserName : dev2-at-738131768609
        iii. Password : VBPDy6gwvpWyuLeAKrDFa7MQOdmuZBtAhoxcVMe6Sc8=
2. Create Repository in AWS CodeCommit
3. Clone in your machine
    a. $ git init

b. $ git clone URL(HTTPS/SSH)

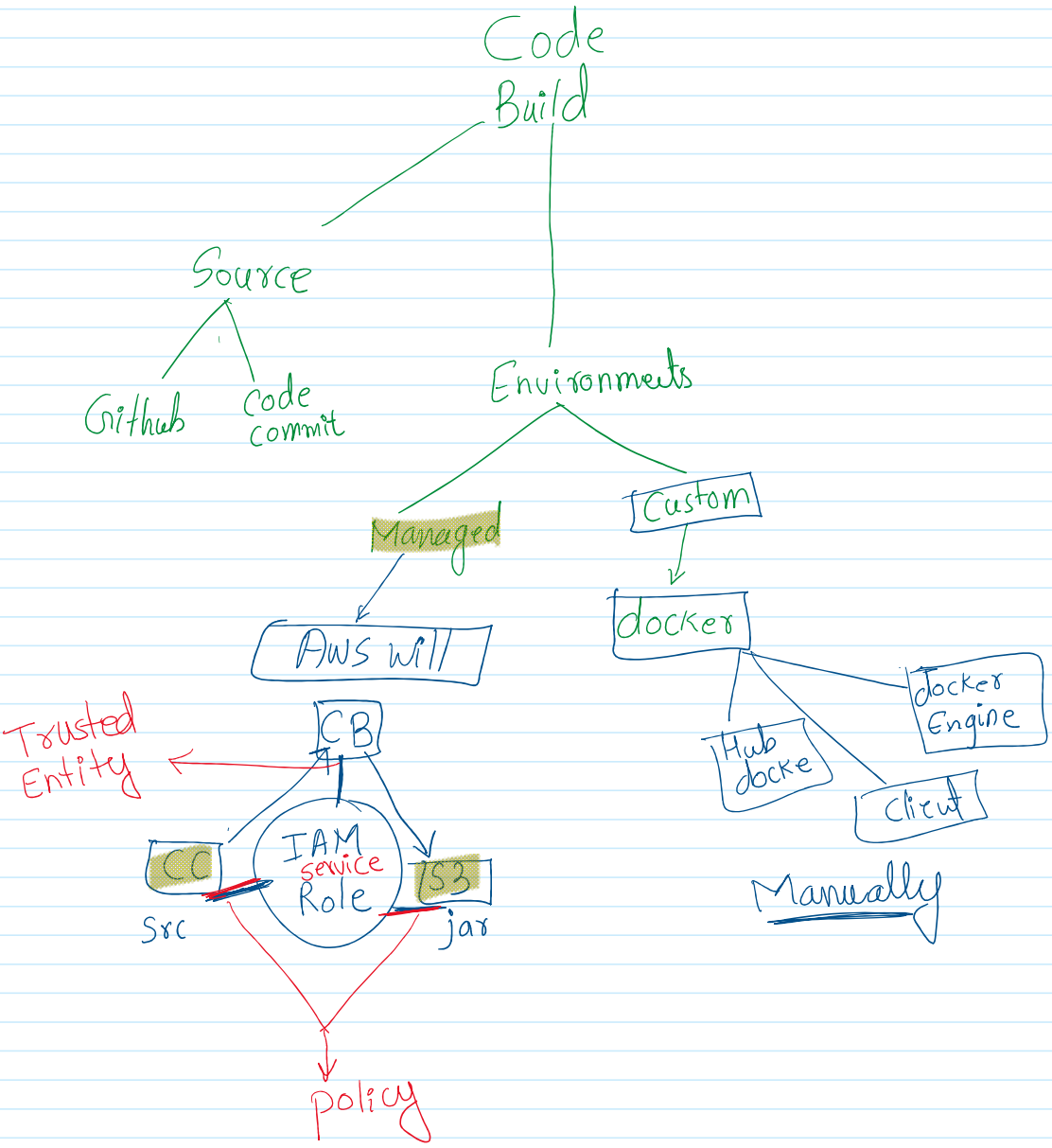
CodeBuild

- Building Project → buildspec.yml
  - Understandable By CB service.
- Recommended to store into root directory.

```
D:\PRACTICES\TEST
|   .gitignore
|   buildspec.yml
|   mvnw
|   mvnw.cmd
|   pom.xml
|   README.md
|
+---.mvn
|   \---wrapper
|           maven-wrapper.jar
|           maven-wrapper.properties
|           MavenWrapperDownloader.java
|
\---src
    +---main
    |   +---java
    |   |   \---io
    |   |       \---nubeera
```

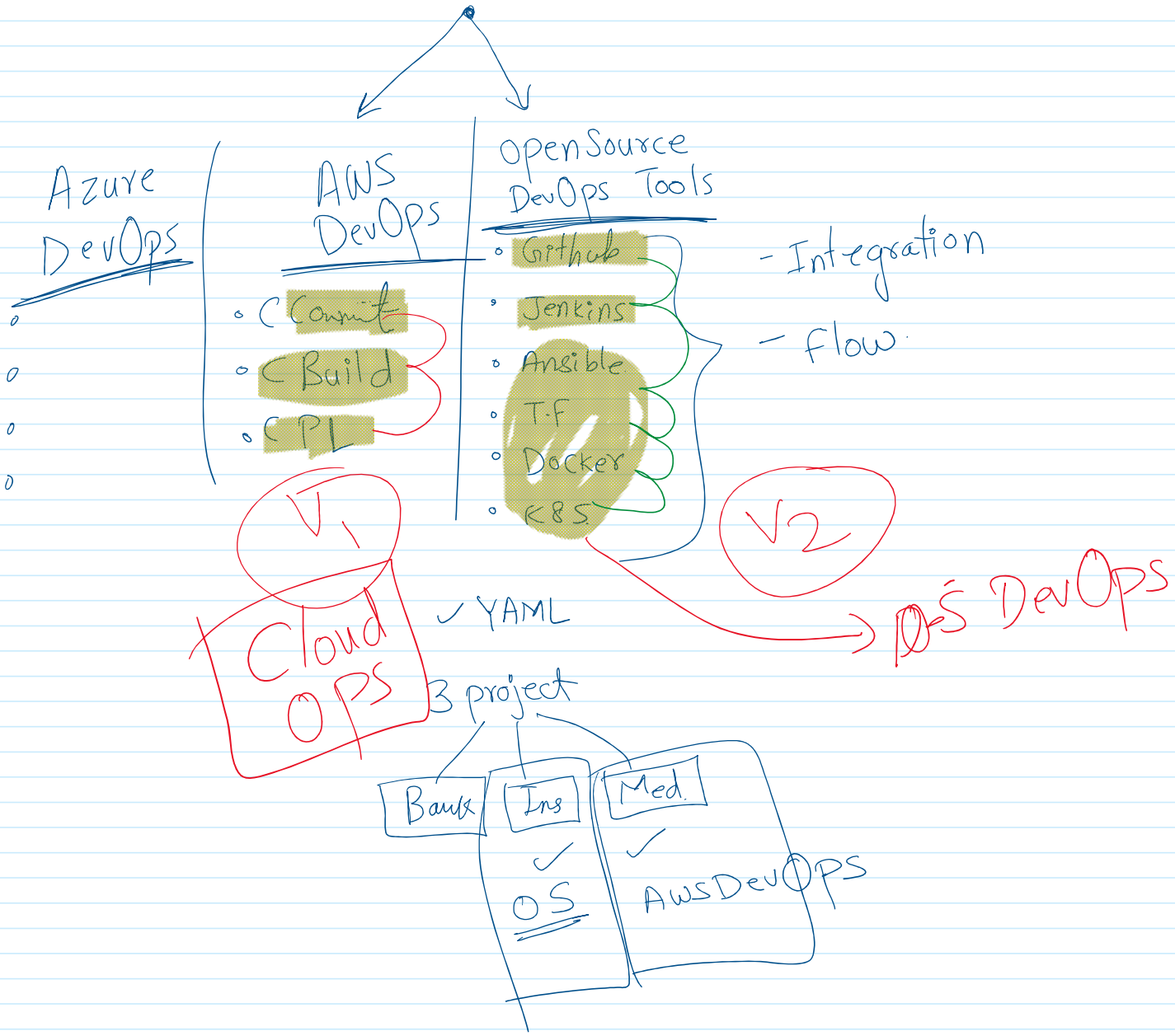


1. Source: CodeCommit --> Repository
2. Processing: CB --> buildspec.yaml
3. Output: Create S3 Bucket



```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto17
  pre_build:
    commands:
      - echo Nothing to do in the pre_build phase...
  build:
    commands:
      - echo Build started on `date`
      - mvn install
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  files:
    - target/*.jar
```



1. A microservice which exposes below mentioned endpoints as APIs and uses in memory h2 database to store the data.
   a. /registerDoctor (HTTP Method : POST) (Request Body : JSON)
   b. /updateDoctor/{doctorRegNo} (HTTP Method : PUT) (Request Body : JSON)
   c. /searchDoctor/{doctorName} (HTTP Method : GET) (No Request Body)
   d. /deletePolicy/{doctorRegNo} (HTTP Method : DELETE) (No Request Body)
2. Write necessary Junit testcase.
3. Generate HTML report using TestNG.
4. Push your code into your GitHub Repository.