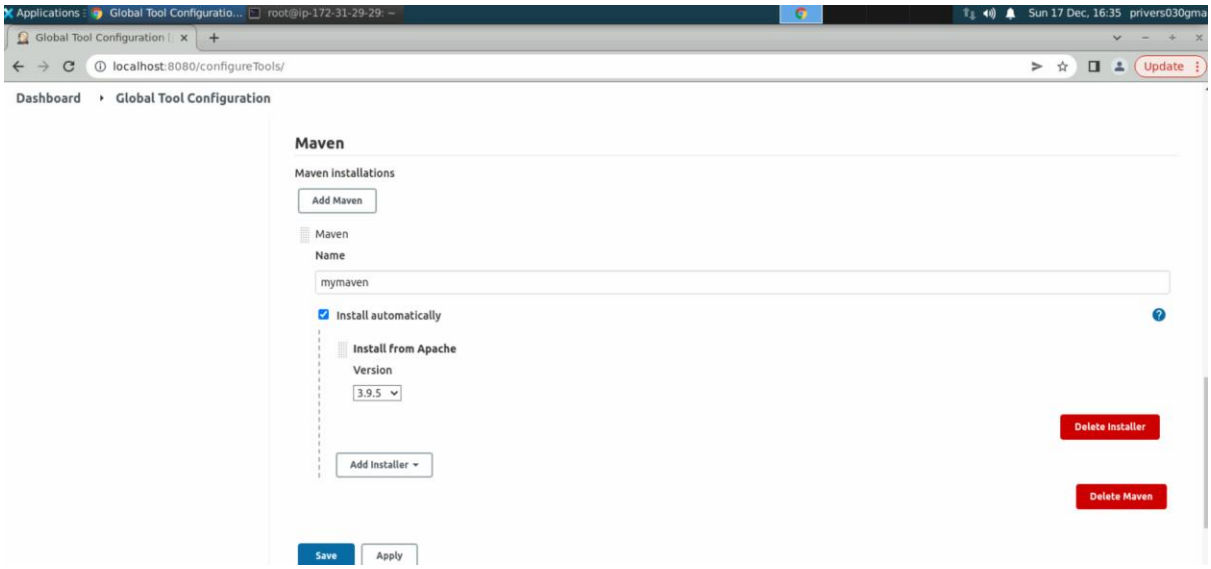
Write details about the definition of the Project:

Solution:

Continuous Integration:

Step 1: Launch Jenkins and set up Maven tool configuration as we have to perform Continuous Integration:

Add the screenshot.



Step 2: Create a Pipeline job in which you will write steps to clone the repo and package the code

Take screenshot of Trigeers

```
pipeline{
    tools {
        maven 'mymaven'
    }
    agent any
```

```
  stages{
    stage('clone the repo')
    {
      steps{
      git 'https://github.com/Sonal0409/DevOpsCodeDemo.git'
    }
    }

    stage('Build the code')
    {
      steps{
        sh 'mvn package'
      }
    }
  }
}
```

Save the pipeline and build it. You will find the package in workspace/target folder

You need to copy the path where the application package is present

/var/lib/jenkins/workspace/CICD_CourseEndPorject2/target/addressbook.war

In the above path, pls note and make change to your Jenkins job name

Continuous Delivery:

Step 1: Use the linux command to copy the addressbook.war file in to the same directory as that of dockerfile

Step 2. We have to take the above build and write a docker file that will create custom Image

New stages are added to the pipeline

```
pipeline{
    tools {
        maven 'mymaven'
    }
    agent any

    stages{
        stage('clone the repo')
        {
            steps{
            git 'https://github.com/Sonal0409/DevOpsCodeDemo.git'
        }
        }

        stage('Build the code')
        {
            steps{
                sh 'mvn package'
            }
        }

        stage('copy the build to workspace')
        {
            steps{
            // copy the build file from target folder to workspace folder where dockerfile is also pressent

            sh 'cp /var/lib/jenkins/workspace/CICD_CourseEndPorject2/target/addressbook.war .'

        }
        }
```

```
    stage('build the Image'){
      steps{
        sh 'docker build -t project2image .'
      }
    }
  }
}
```

==========

By default Jenkins user cannot run docker commands

We have to give permissions to run docker commands

Execute the below command:

chmod 777 /var/run/docker.sock

Run the pipeline and add the screenshot.

Add the new stage to run the image and deploy the application on container

```
  pipeline{
    tools {
      maven 'mymaven'
    }
    agent any

    stages{
      stage('clone the repo')
      {
        steps{
        git 'https://github.com/Sonal0409/DevOpsCodeDemo.git'
      }
```

```
        }

        stage('Build the code')
        {
           steps{
              sh 'mvn package'
           }
        }

        stage('copy the build to workspace')
        {
           steps{
           // copy the build file from target folder to workspace folder where dockerfile is also
pressent

           sh 'cp /var/lib/jenkins/workspace/CICD_CourseEndPorject2/target/addressbook.war .'

        }
        }

        stage('build the Image'){
           steps{
              sh 'docker build -t project2image .'
           }
        }

        stage('Deploy on Container')
        {
           steps{
           sh 'docker run -d -P project2image'
        }
```

```
        }
      }
   }
```

Build the job and add screenshot.

A new container must be created

docker ps

Add the stage to push the image to docker Hub

```
   pipeline{
      tools {
         maven 'mymaven'
      }
      agent any

      stages{
         stage('clone the repo')
         {
            steps{
            git 'https://github.com/Sonal0409/DevOpsCodeDemo.git'
         }
         }

         stage('Build the code')
         {
            steps{
               sh 'mvn package'
            }
         }
```

```
        stage('copy the build to workspace')
        {
           steps{
           // copy the build file from target folder to workspace folder where dockerfile is also
pressent

           sh 'cp /var/lib/jenkins/workspace/CICD_CourseEndPorject2/target/addressbook.war .'

        }
        }

        stage('build the Image'){
           steps{
              sh 'docker build -t project2image .'
           }
        }



        stage('Push Image dockerhub')
        {
           steps{
           sh 'docker login -u sonal04 -p abc@123'
           sh 'docker tag project2image sonal04/project2image'
             sh 'docker push sonal04/project2image'
           }
        }
```

```
        stage('Deploy on Container')
        {
            steps{
            sh 'docker run -d -P project2image'
        }

        }
    }
}
```

Final pipeline

```
pipeline{
    tools {
        maven 'mymaven'
    }
    agent any

    stages{
        stage('clone the repo')
        {
            steps{
            git 'https://github.com/Sonal0409/DevOpsCodeDemo.git'
        }
        }

        stage('Build the code')
        {
            steps{
                sh 'mvn package'
            }
```

```
        }


        stage('copy the build to workspace')
        {
           steps{
           // copy the build file from target folder to workspace folder where dockerfile is also
pressent


           sh 'cp /var/lib/jenkins/workspace/CICD_CourseEndPorject2/target/addressbook.war .'


        }
        }


        stage('build the Image'){
           steps{
              sh 'docker build -t project2image:$BUILD_NUMBER .'
           }
        }




        stage('Push Image dockerhub')
        {
           steps{
           sh 'docker login -u sonal04 -p abc@123'
           sh 'docker tag project2image:$BUILD_NUMBER sonal04/project2image:$BUILD_NUMBER'
             sh 'docker push sonal04/project2image:$BUILD_NUMBER'
           }
        }
```

```
        stage('Deploy on Container')
        {
           steps{
           sh 'docker run -d -P sonal04/project2image:$BUILD_NUMBER'
        }


        }
      }
   }
```

Add the screenhots.