Phase 2 Implementation:

Step 1: Create OneDrive as a local private Git repository.

git init

git add--all

git commit -m "init"

git init --bare ~/OneDrive/repository/EbsBackend.git

git remote add OneDrive ~/ OneDrive/git/project1.git

git push -u OneDrive master

Step 2: Jenkins install

First we need to install JDK 11.

Then visit Jenkins install page à https://www.jenkins.io/download/

Then follow along this wizard (if need help follow this website for complete installation à https://www.blazemeter.com/blog/how-to-install-jenkins-on-windows)

Step 3: Docker Desktop install

Follow this link for installation à https://docs.docker.com/desktop/windows/install/

[WSL 2 must be selected at the starting of the installation/ if windows 10/11 Pro you can ignore]

Step 4: SonarQube install

Follow this link à https://www.sonarqube.org/downloads/

[Download the free version]

If installation complete start sonar in cmd and visit Localhost:9000

[Id: admin pw: admin]à you need to change the pw for the first time

Step 5: Open Jenkins in à http://localhost:8080/

1. Go to à Manage Jenkins
2. Click à Manage Plugins
3. Install Plugins for Docker and Sonarqube
   1. Docker Pipeline
   2. Docker Plugin
   3. Docker-build-step
   4. SonarQube Scanner for Jenkins
4. Go back to Dashboard à Manage Jenkins à click configuration
   1. Under the section à SonarQube servers à Select the Environment variables
5. Go back to Dashboard à Manage Jenkins à click Manage Credentials
   1. For Docker
      1. Give your Username
      2. And the generated token from Docker
   2. For SonarQube
      1. Same for sonarqube like Docker and for generating the token you need to visit Localhost:9000
6. Go back to Dashboard à Manage Jenkins à click Global Tool Configuration
   1. Under the section à SonarScanner for MSBuild select the name and installation path

Step 6: Creating the Jenkins Pipeline

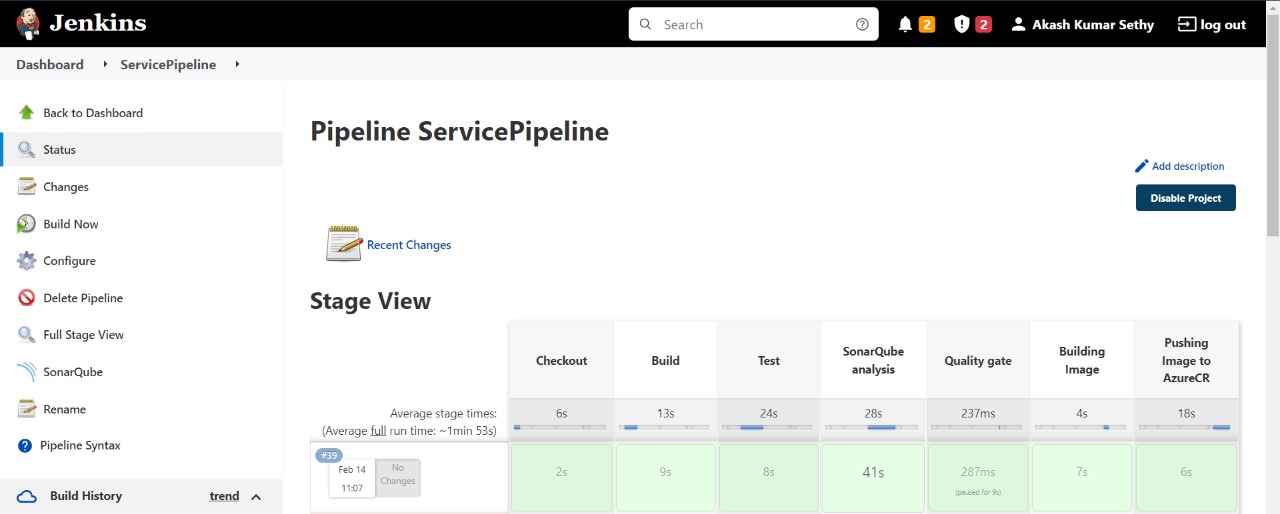
Dashboard: Click **new item** and select **pipeline** and give a **Pipeline name**

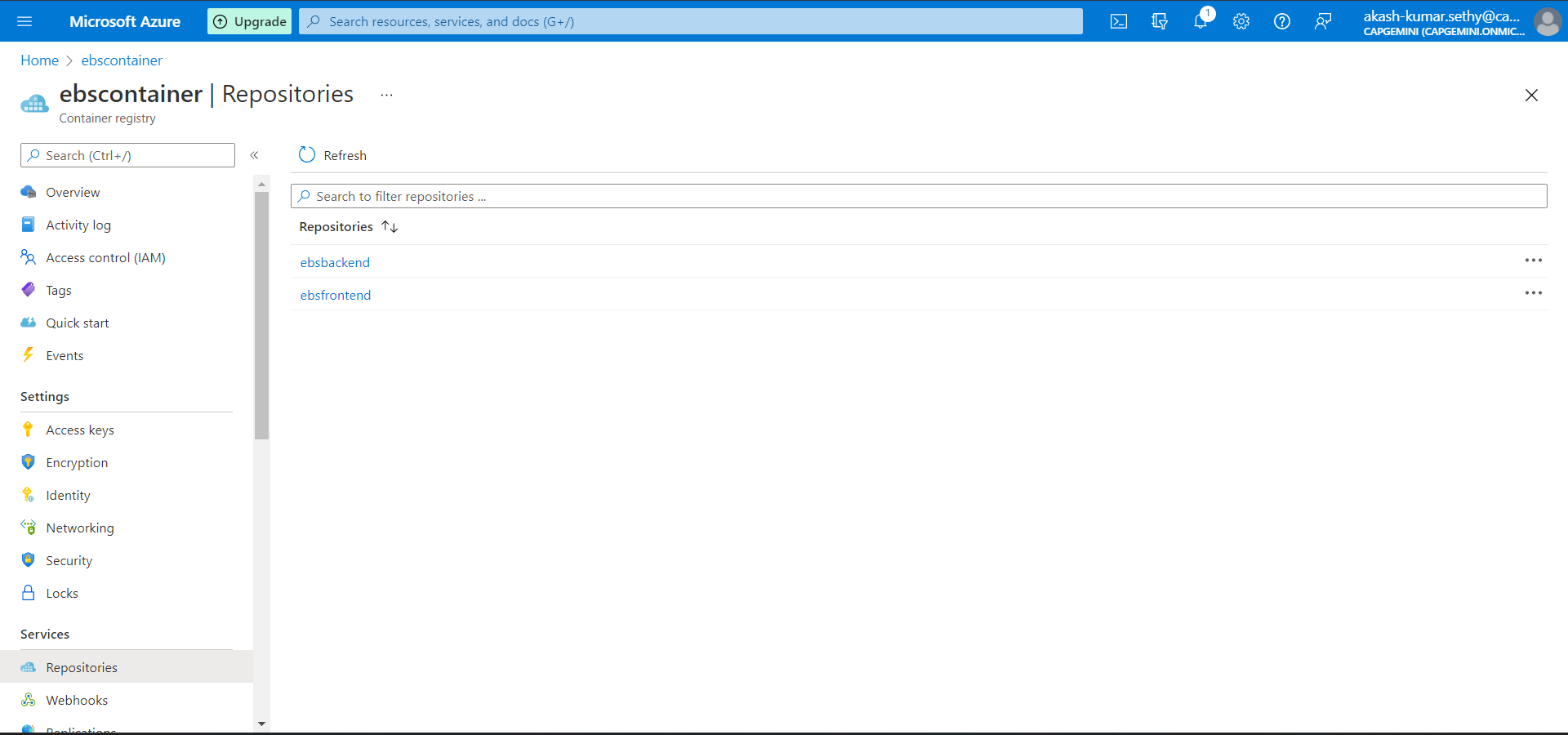
Now write the Pipeline Script: Backend Pipeline Script

|  |
| --- |
| pipeline {  agent any  environment {  dotnet = 'C:\\Program Files\\dotnet\\dotnet.exe'  sonarscanner='C:\\ProgramData\\Jenkins\\.jenkins\\tools\\hudson.plugins.sonar.MsBuildSQRunnerInstallation\\SonarScannerforMSBuild\\SonarScanner.MSBuild.dll'  }  stages {  stage('Checkout') {  steps {  // Get some code from a GitHub repository  git 'C:\\Users\\DELL\\OneDrive\\ebsProject.git'  }  }  stage('Build') {  steps {  // Building the backend  bat 'cd EBSBackend && dotnet build EBS.sln'  }  }  stage('Test') {  steps {  bat 'cd EBSBackend && cd EBSTest && dotnet test EBSTest.csproj'  }  }  stage('SonarQube analysis') {  steps {  withSonarQubeEnv('sonarqube') {  bat 'cd EBSBackend && dotnet C:\\ProgramData\\Jenkins\\.jenkins\\tools\\hudson.plugins.sonar.MsBuildSQRunnerInstallation\\SonarScannerforMSBuild\\SonarScanner.MSBuild.dll begin /k:"ebsProject" /d:sonar.host.url="http://localhost:9000" /d:sonar.login="\*\*\*\*\*\*\*\*[Provide the sonarqube generated token]\*\*\*\*\*\*\*\*"'  bat 'cd EBSBackend && dotnet build EBS.sln'  bat 'cd EBSBackend && dotnet C:\\ProgramData\\Jenkins\\.jenkins\\tools\\hudson.plugins.sonar.MsBuildSQRunnerInstallation\\SonarScannerforMSBuild\\SonarScanner.MSBuild.dll end /d:sonar.login="\*\*\*\*\*\*\*\*[Provide the sonarqube generated token]\*\*\*\*\*\*\*\*\*"'  }  }  }  stage("Quality gate") {  steps{  waitForQualityGate abortPipeline: true  }  }    stage("Building Image") {  steps{  bat 'cd EBSBackend && cd EBS\_Backend && docker build -t (docker Username)/ebsbackend:v1 .'  }  }  stage("Pushing Image to Docker Hub") {  steps{  bat 'cd EBSBackend && cd EBS\_Backend && docker push (docker Username)/ebsbackend:v1'  }  }  }  } |

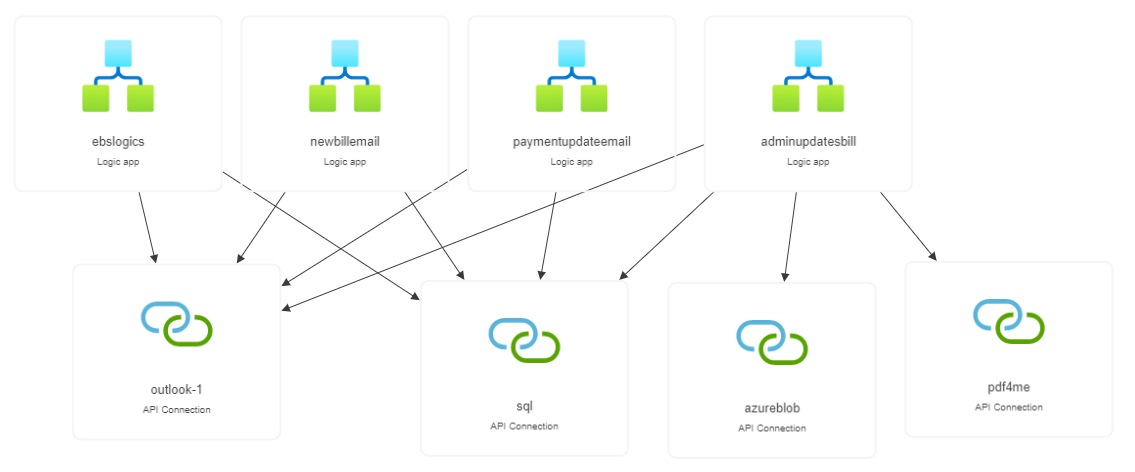
For Frontend Pipeline:

|  |
| --- |
| pipeline {  agent any  stages {  stage('Checkout') {  steps {  // Get some code from a GitHub repository  git 'C:\\Users\\DELL\\OneDrive\\ebsProject.git'  }  }  stage("Building Image") {  steps{  bat 'cd Frontend\\eBillSolution && docker build -t dockerelon/ebsfrontend:v1 .'  }  }  stage("Pushing Image to Docker Hub") {  steps{  bat 'cd Frontend\\eBillSolution && docker push dockerelon/ebsfrontend:v1'  }  }  }  } |





**Logic Apps:**



**SonarQube:**

