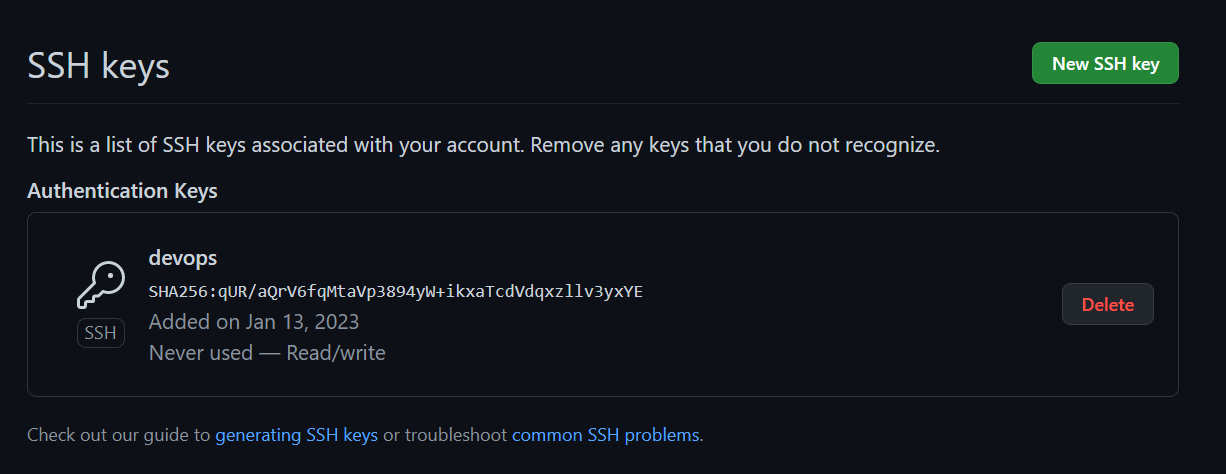
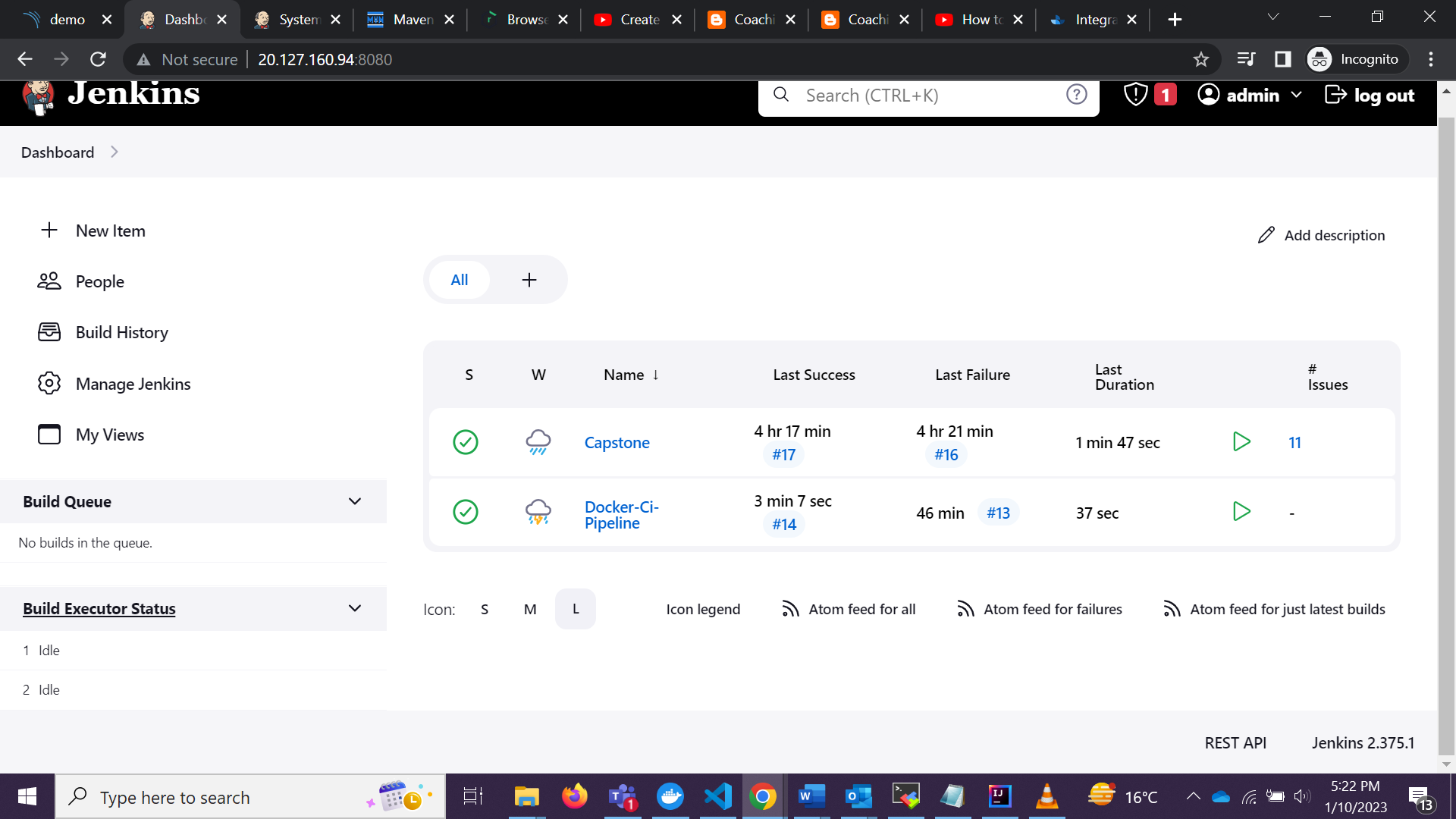
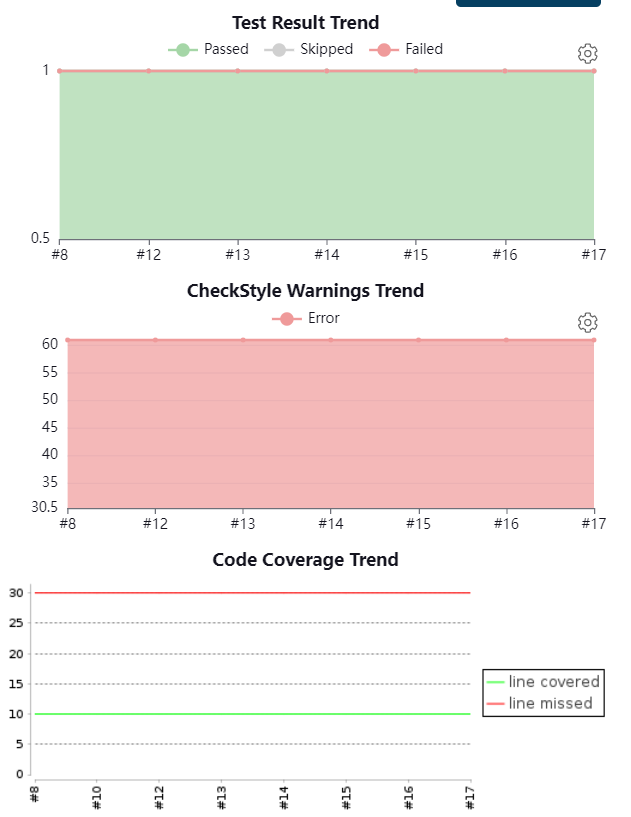
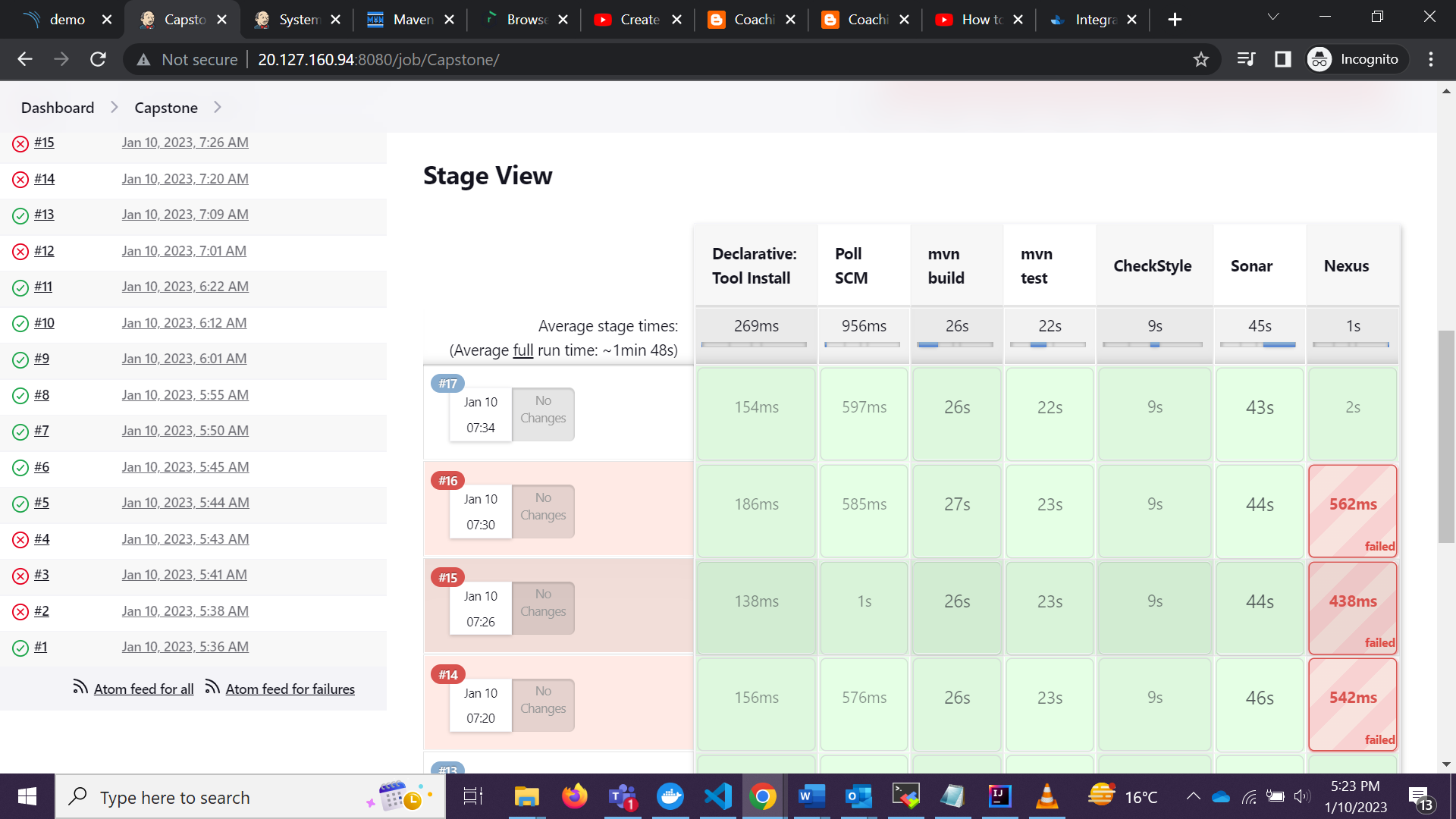
****



****



pipeline {

agent any

tools{

maven "maven3"

}

stages {

stage('Poll SCM') {

steps {

checkout scmGit(branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/anoopraj467/Capston']])

}

}

stage('mvn build'){

steps{

sh 'mvn -B -DskipTest clean package'

}

}

stage('mvn test'){

steps{

sh 'mvn test'

junit 'target/surefire-reports/\*.xml'

}

}

stage('CheckStyle'){

steps{

sh 'mvn checkstyle:checkstyle'

recordIssues(tools: [checkStyle(pattern: '\*\*/checkstyle-result.xml')])

}

}

stage('Sonar'){

steps{

sh 'mvn clean verify sonar:sonar \

-Dsonar.projectKey=Capstone \

-Dsonar.host.url=http://20.127.160.165:9000 \

-Dsonar.login=sqp\_f8b760311be4e965dffe3dee0e4fa53dbcfd4483'

}

}

stage('Nexus'){

steps{

script{

pom = readMavenPom file: "pom.xml";

filesByGlob = findFiles(glob: "target/\*.${pom.packaging}");

echo "${filesByGlob[0].name} ${filesByGlob[0].path} ${filesByGlob[0].directory} ${filesByGlob[0].length} ${filesByGlob[0].lastModified}"

artifactPath = filesByGlob[0].path;

}

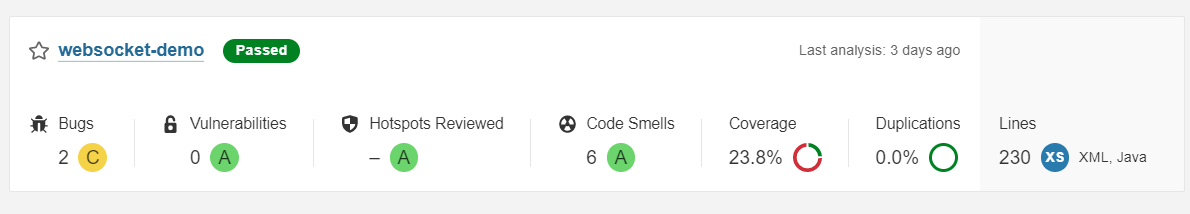
nexusArtifactUploader artifacts: [[artifactId: 'demo', classifier: '', file: 'target/demo-0.0.1-SNAPSHOT.jar', type: 'jar']], credentialsId: 'nexus-cred', groupId: 'demo', nexusUrl: '20.127.187.74:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'maven-snapshots', version: '0.0.1-SNAPSHOT'

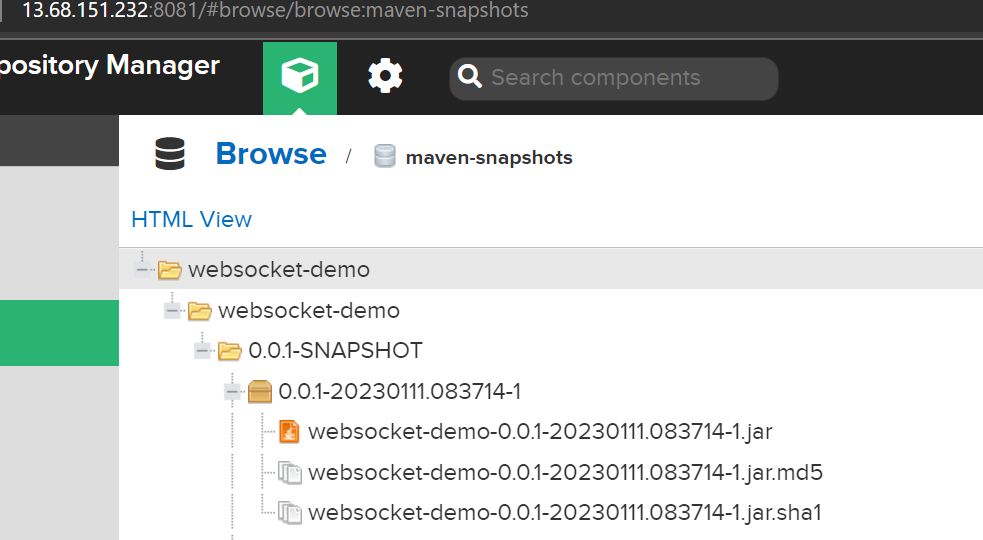
}

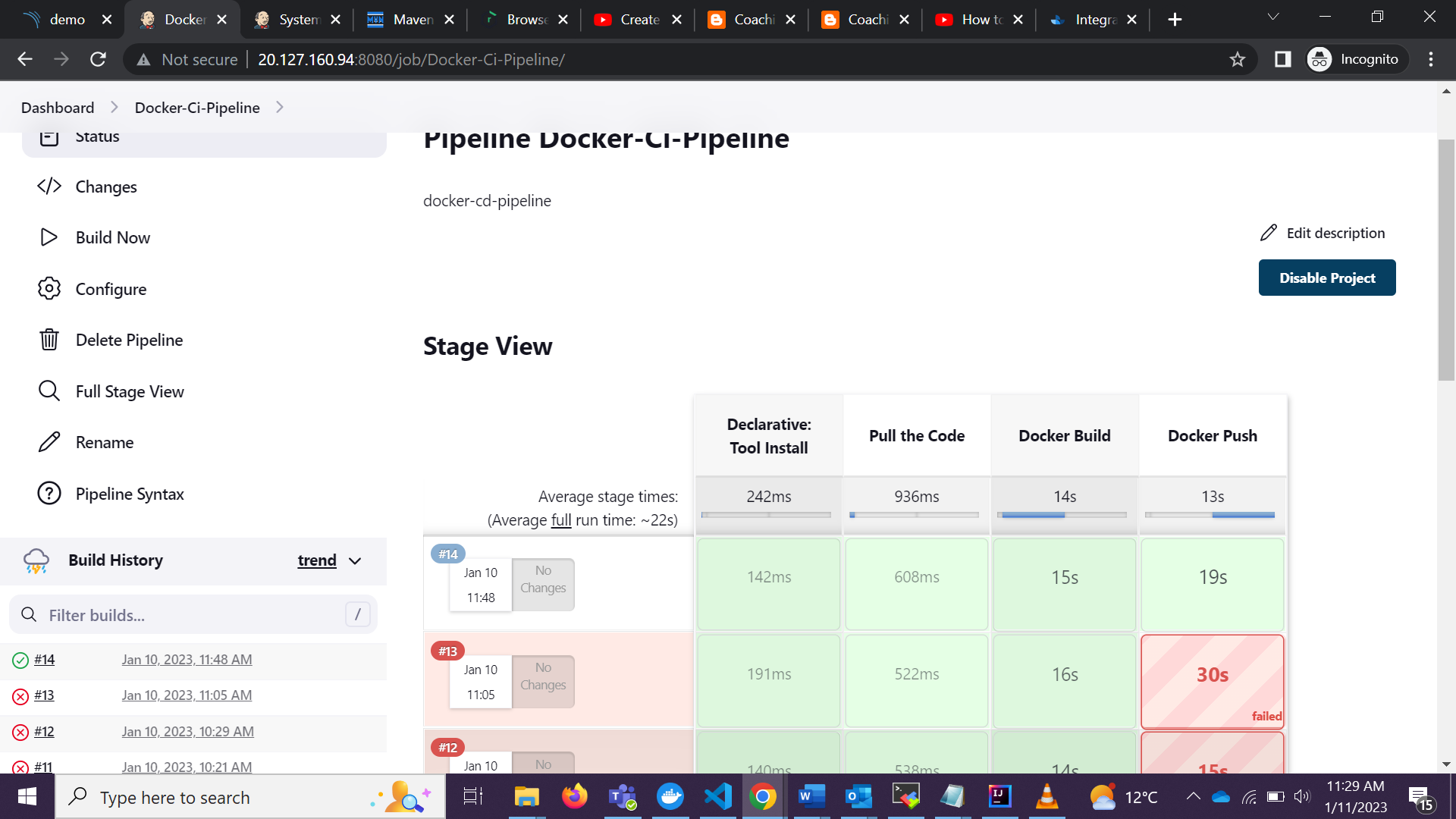
}

}

}

****

****



pipeline {

agent any

environment{

dockerImage = ''

}

tools{

maven "maven3"

}

stages {

stage('Pull the Code') {

steps {

checkout scmGit(branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/anoopraj467/Capston']])

}

}

stage('Docker Build') {

steps {

script{

sh 'mvn compile jib:dockerBuild'

}

}

}

stage('Docker Push'){

steps{

// This step should not normally be used in your script. Consult the inline help for details.

withDockerRegistry(credentialsId: 'nexus-cred', url: 'http://20.127.187.74:8085') {

// some block

sh 'mvn compile jib:build -Djib.allowInsecureRegistries=true -DsendCredentialsOverHttp'

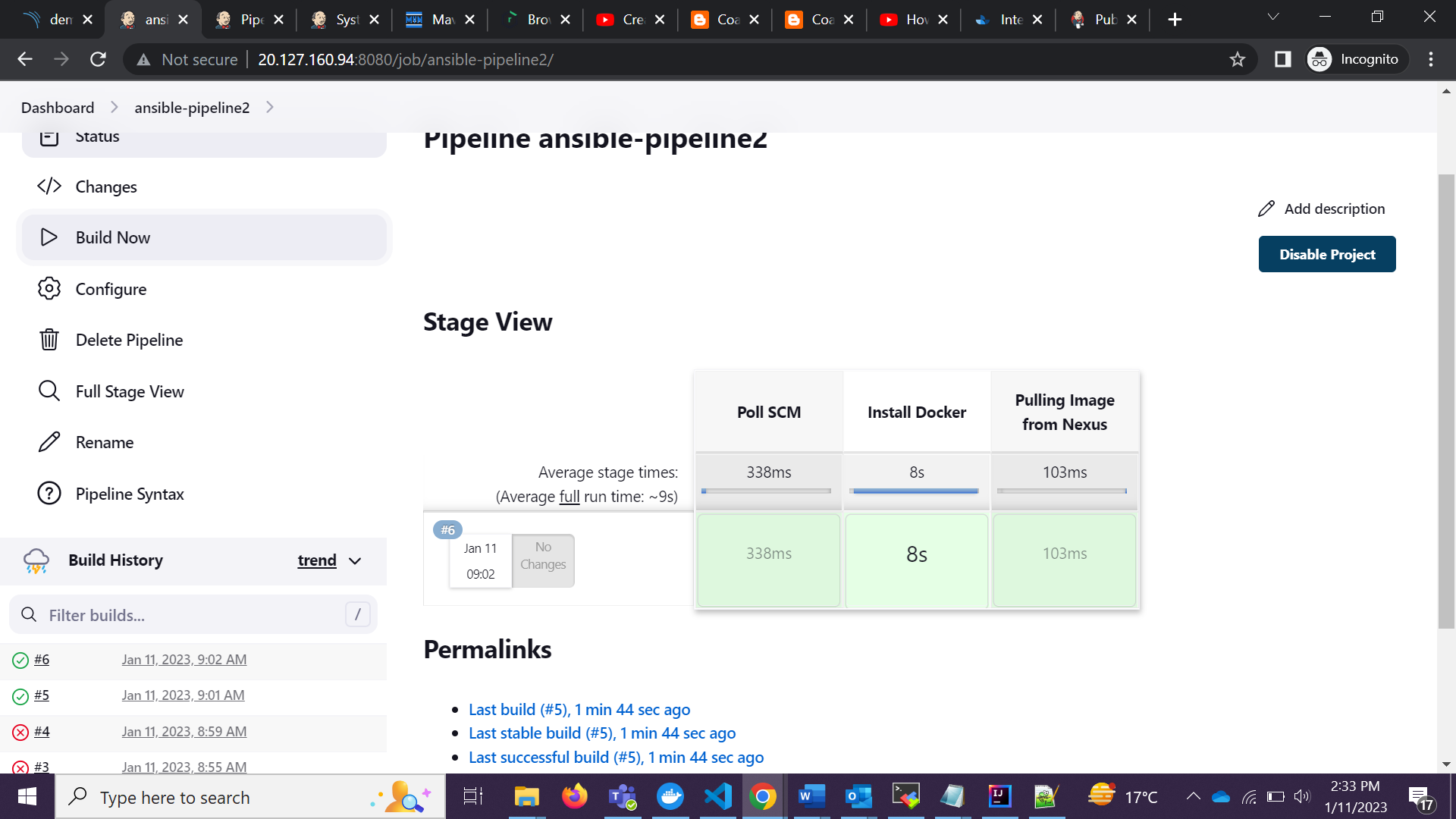
}

}

}

}

}



pipeline {

agent any

stages {

stage('Poll SCM') {

steps {

checkout scmGit(branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/anoopraj467/ansible']])

}

}

stage('Install Docker'){

steps{

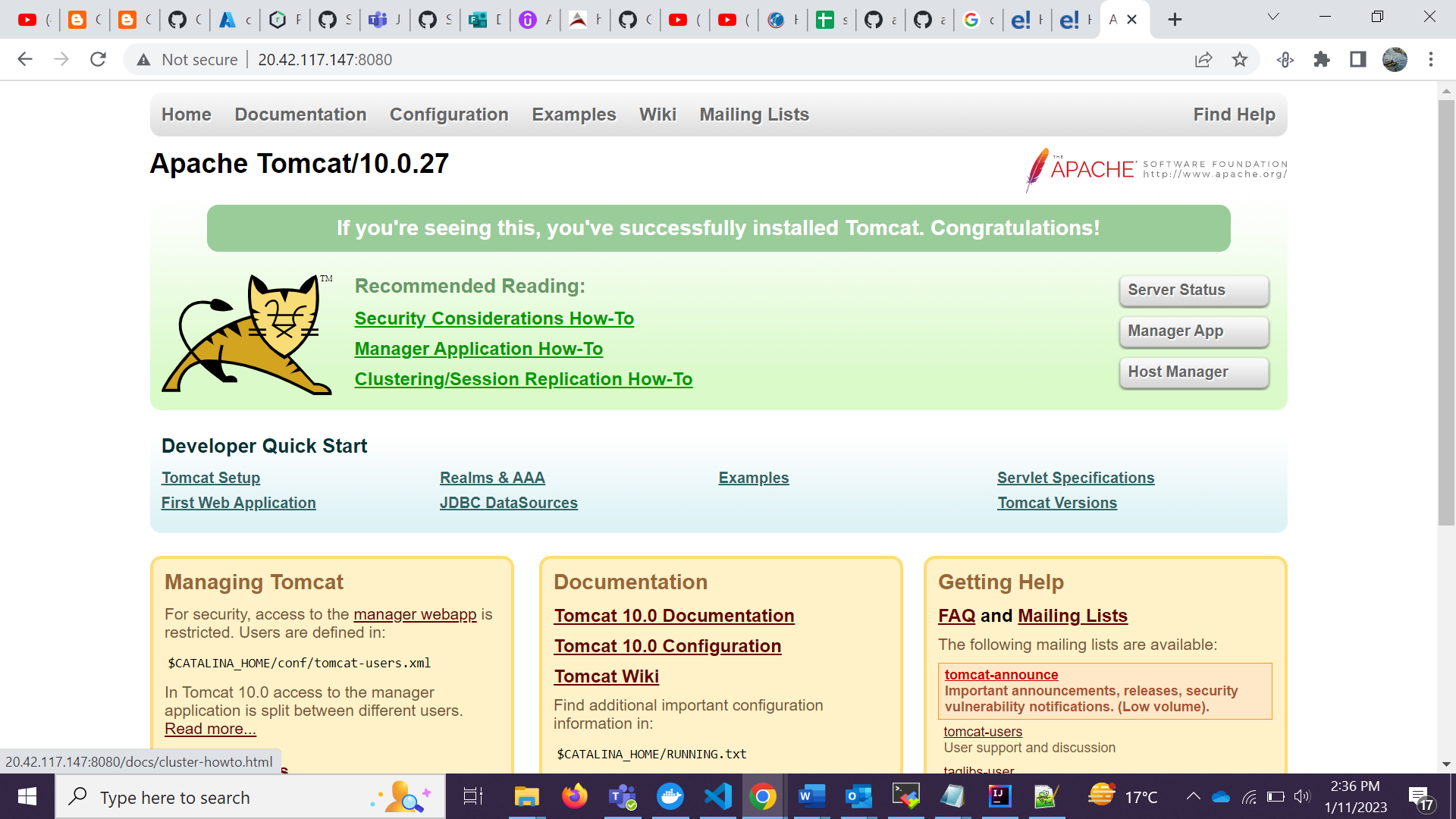
ansiblePlaybook credentialsId: 'root', disableHostKeyChecking: true, installation: 'ansible2', inventory: 'dev.inv', playbook: 'docker.yml'

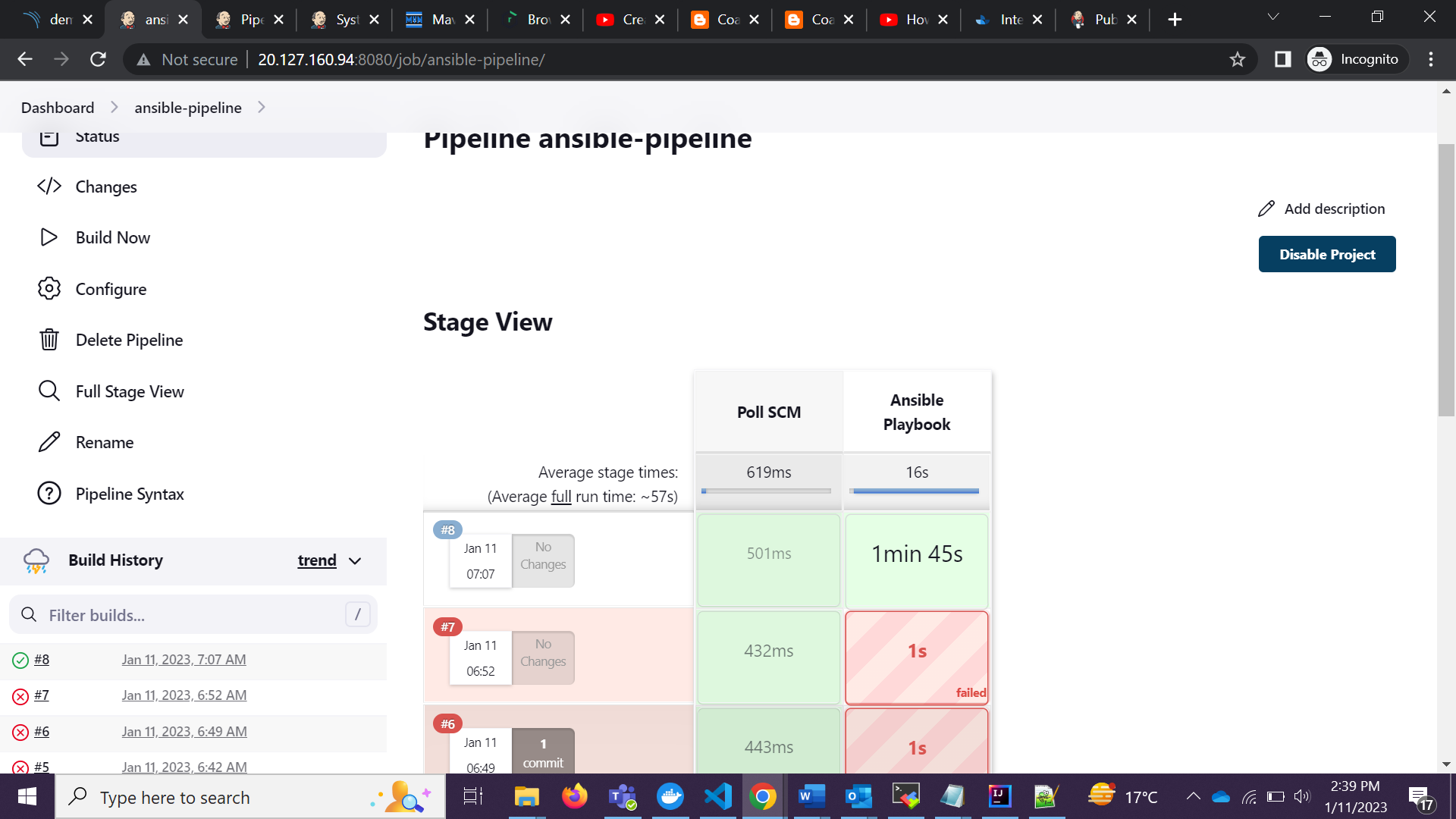
}

}

}

}





pipeline {

agent any

stages {

stage('Poll SCM') {

steps {

checkout scmGit(branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/anoopraj467/ansible']])

}

}

stage('Ansible Playbook'){

steps{

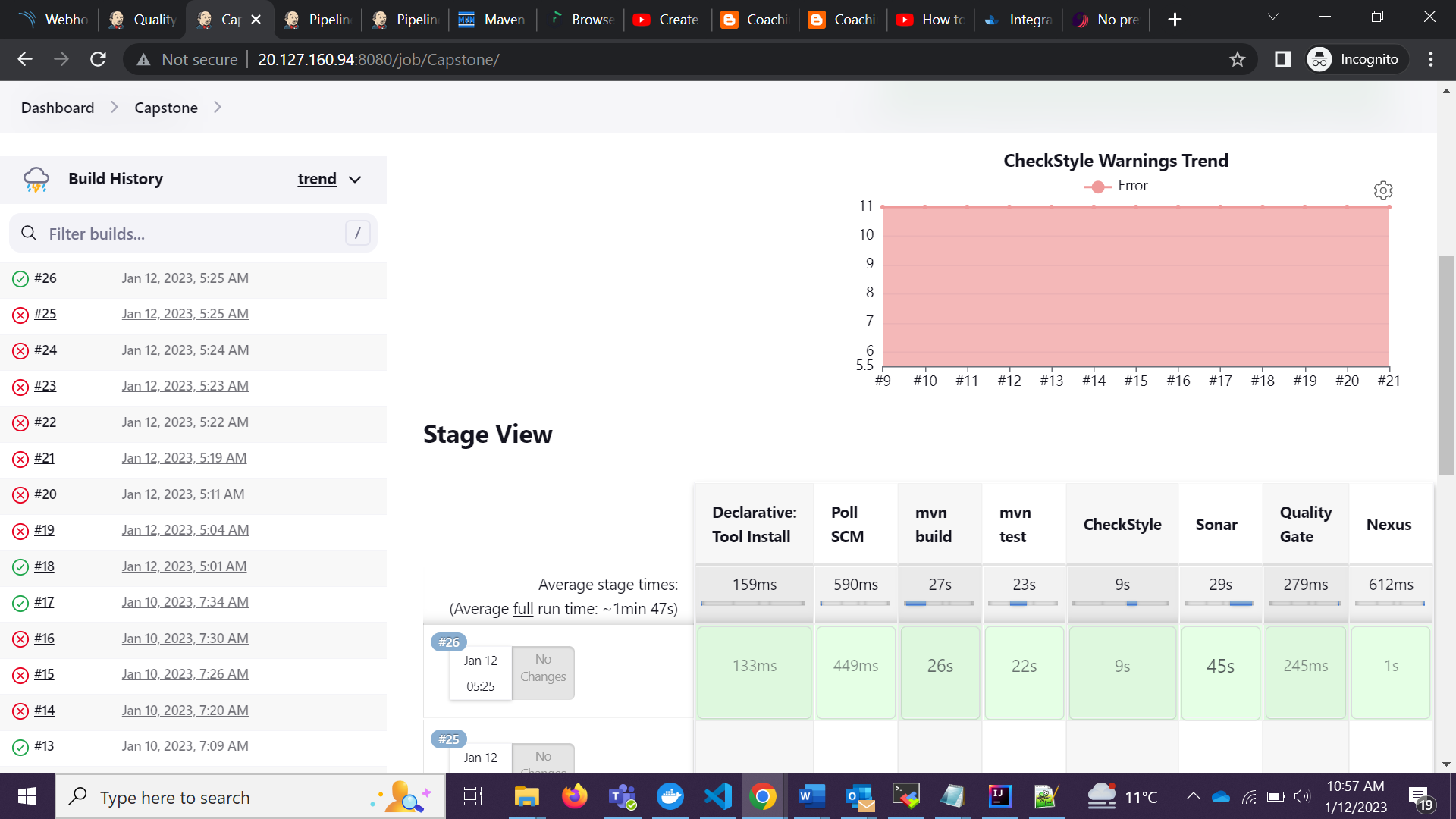
ansiblePlaybook credentialsId: 'root', disableHostKeyChecking: true, installation: 'ansible2', inventory: 'dev.inv', playbook: 'tomcat.yml'

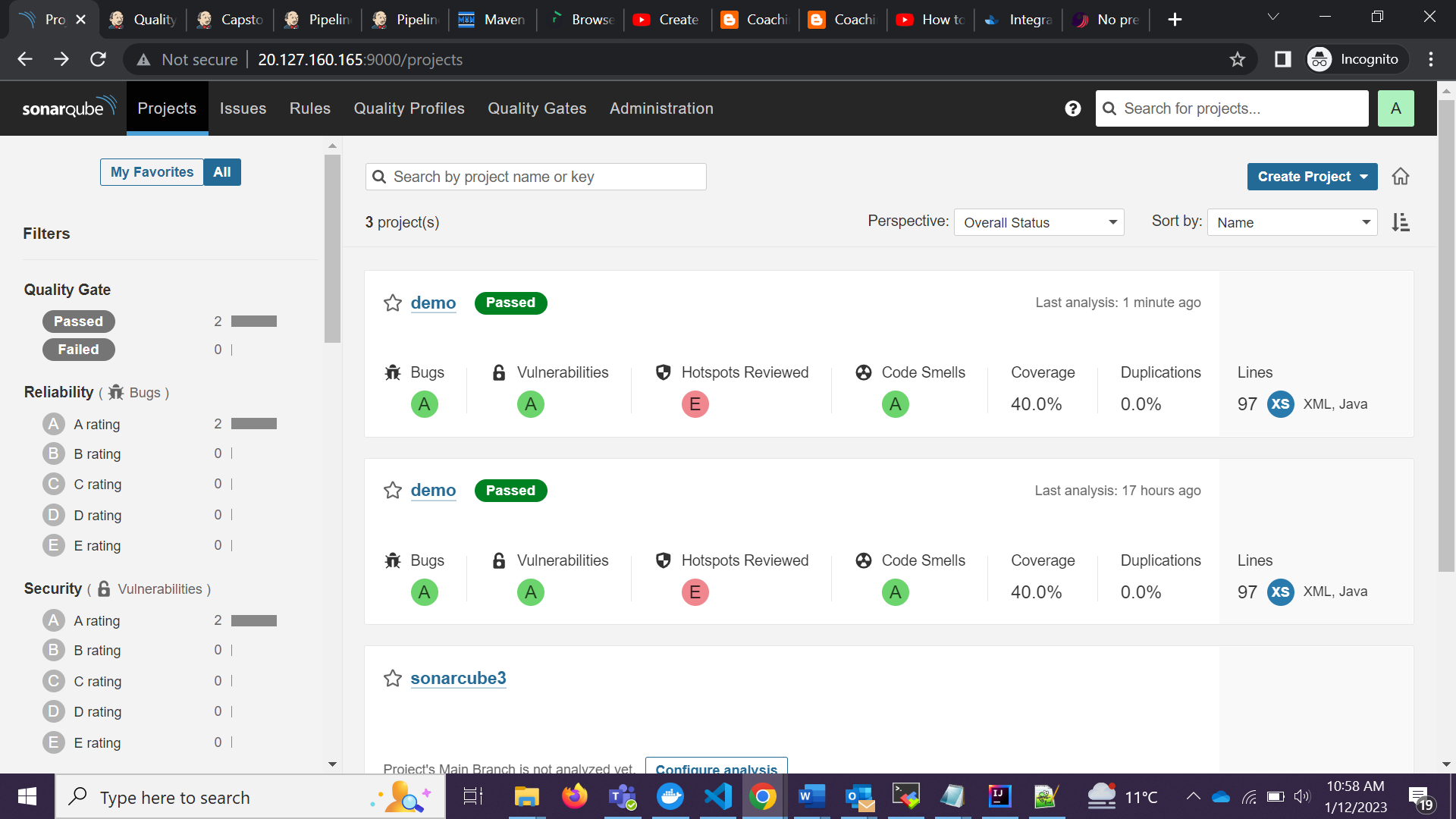
}

}

}

}





pipeline {

agent any

tools{

maven "maven3"

}

stages {

stage('Poll SCM') {

steps {

checkout scmGit(branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/anoopraj467/Capston']])

}

}

stage('mvn build'){

steps{

sh 'mvn -B -DskipTest clean package'

}

}

stage('mvn test'){

steps{

sh 'mvn test'

junit 'target/surefire-reports/\*.xml'

}

}

stage('CheckStyle'){

steps{

sh 'mvn checkstyle:checkstyle'

recordIssues(tools: [checkStyle(pattern: '\*\*/checkstyle-result.xml')])

}

}

stage('Sonar'){

steps{

sh 'mvn clean verify sonar:sonar \

-Dsonar.projectKey=Capstone \

-Dsonar.host.url=http://20.127.160.165:9000 \

-Dsonar.login=sqp\_f8b760311be4e965dffe3dee0e4fa53dbcfd4483'

}

}

stage ("Quality Gate ") {

steps {

echo "Hello Quality Gate"

}

}

stage('Nexus'){

steps{

script{

pom = readMavenPom file: "pom.xml";

filesByGlob = findFiles(glob: "target/\*.${pom.packaging}");

echo "${filesByGlob[0].name} ${filesByGlob[0].path} ${filesByGlob[0].directory} ${filesByGlob[0].length} ${filesByGlob[0].lastModified}"

artifactPath = filesByGlob[0].path;

}

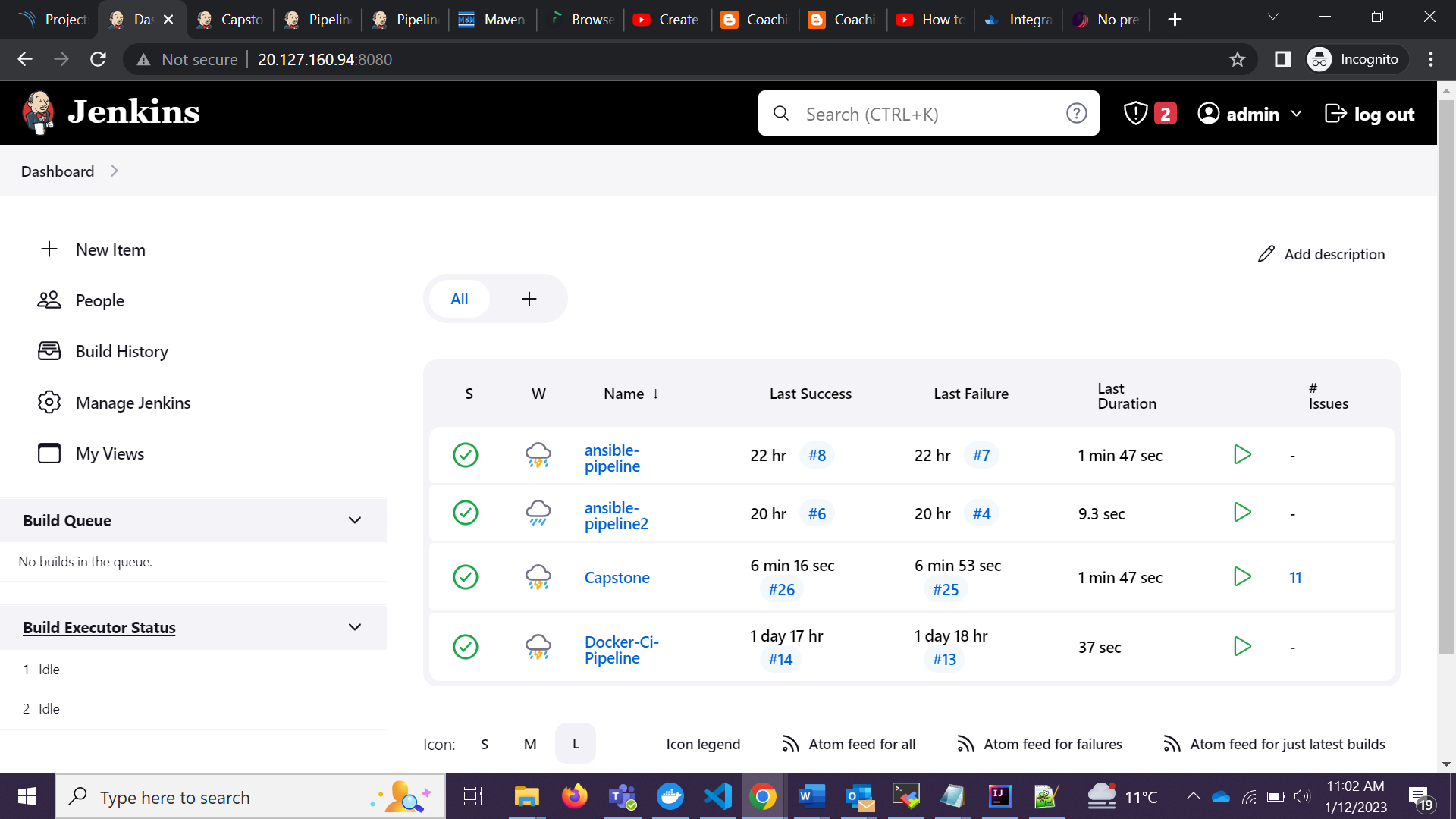
nexusArtifactUploader artifacts: [[artifactId: 'demo', classifier: '', file: 'target/demo-0.0.1-SNAPSHOT.jar', type: 'jar']], credentialsId: 'nexus-cred', groupId: 'demo', nexusUrl: '20.127.187.74:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'maven-snapshots', version: '0.0.1-SNAPSHOT'

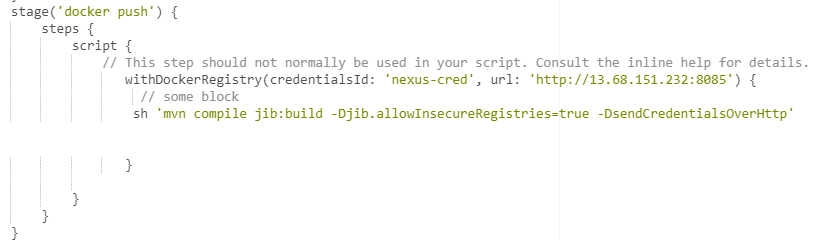
}

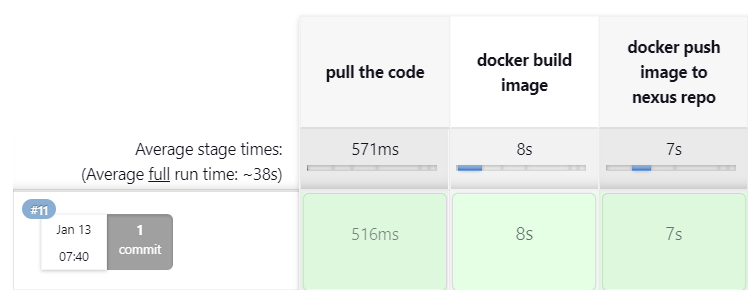
}

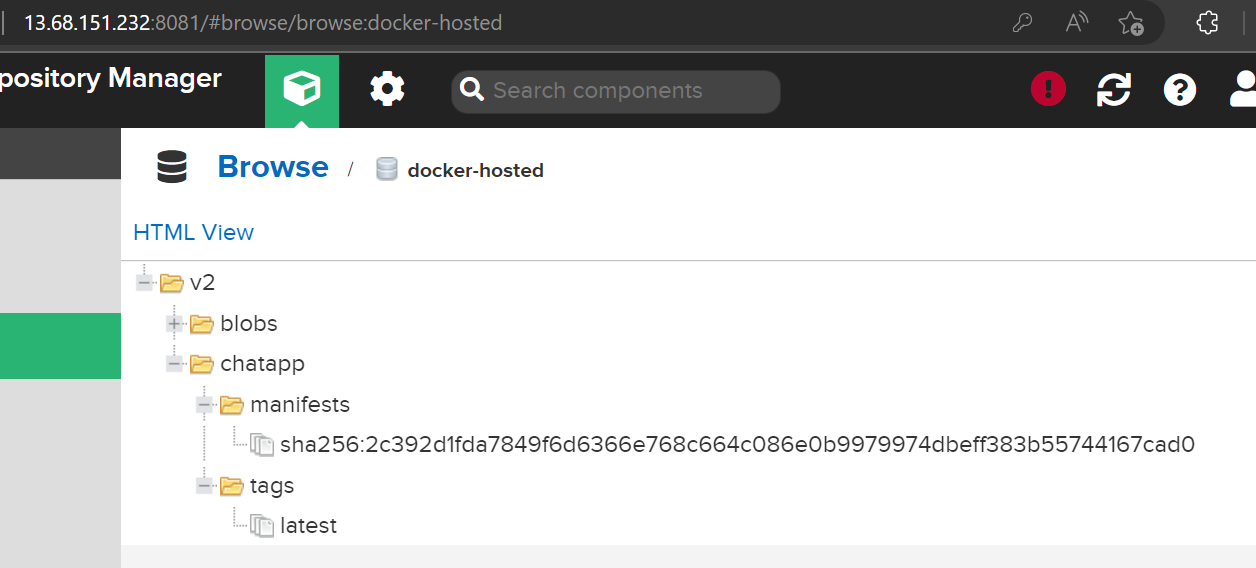
}

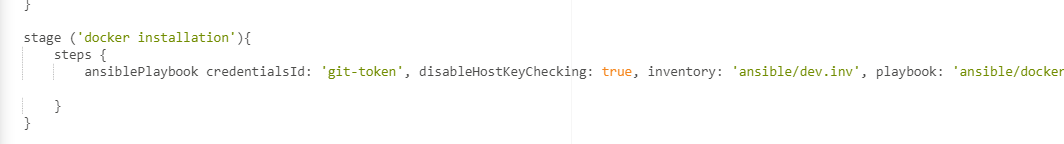
}

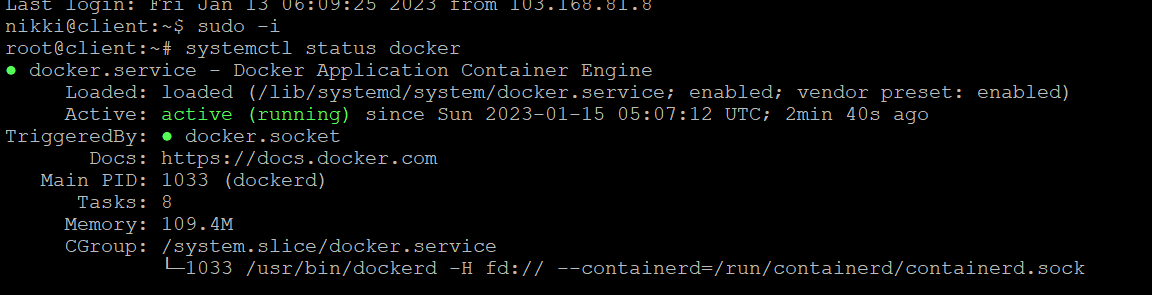


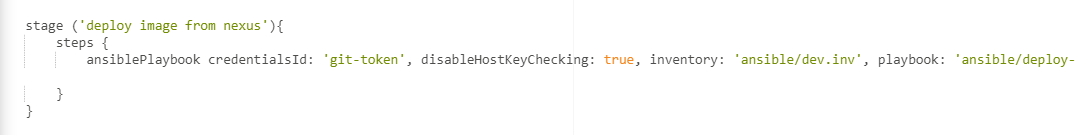
****

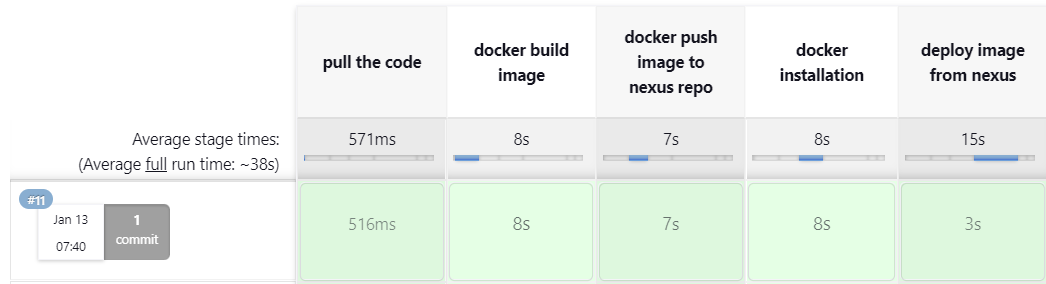
****

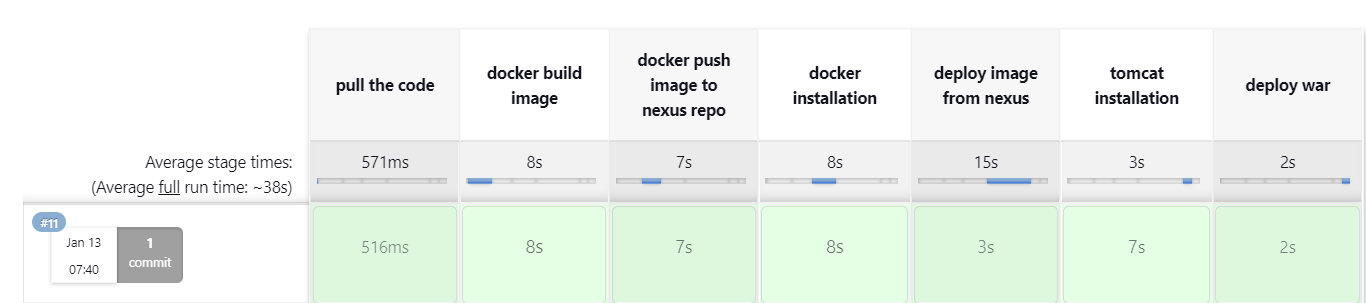
****

****

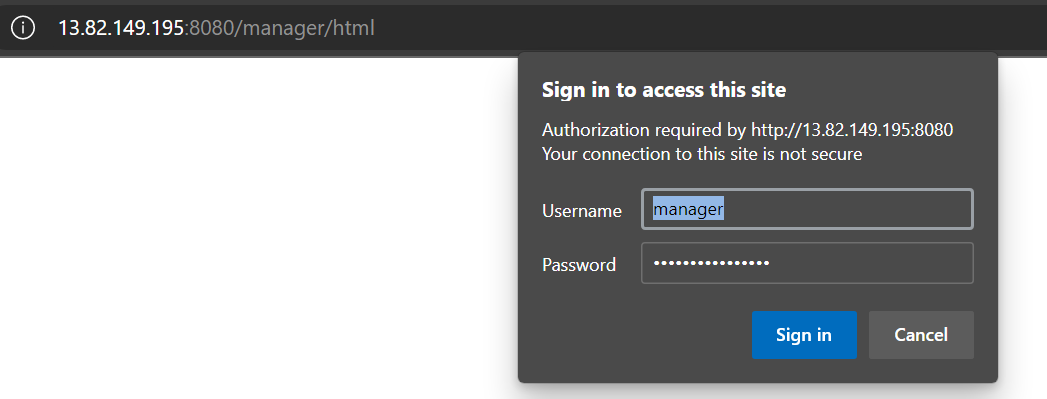
****

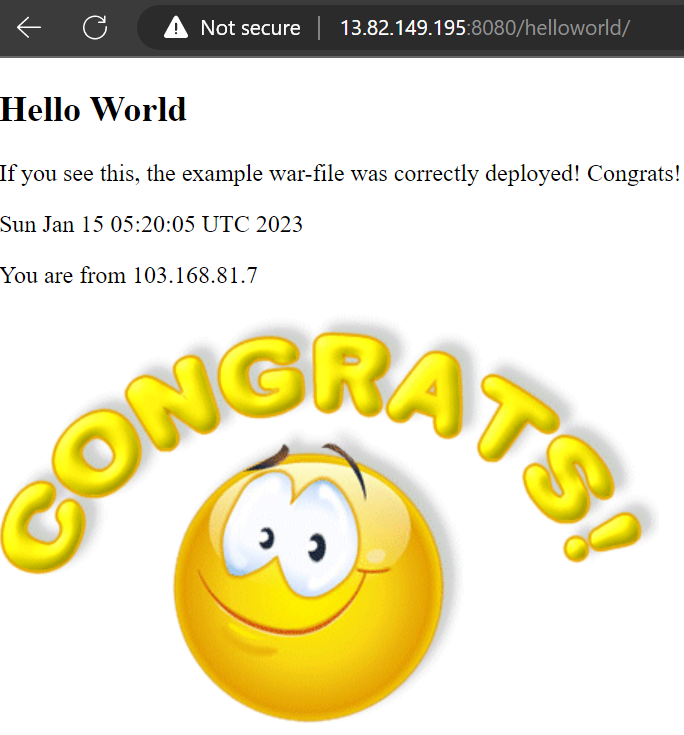
****

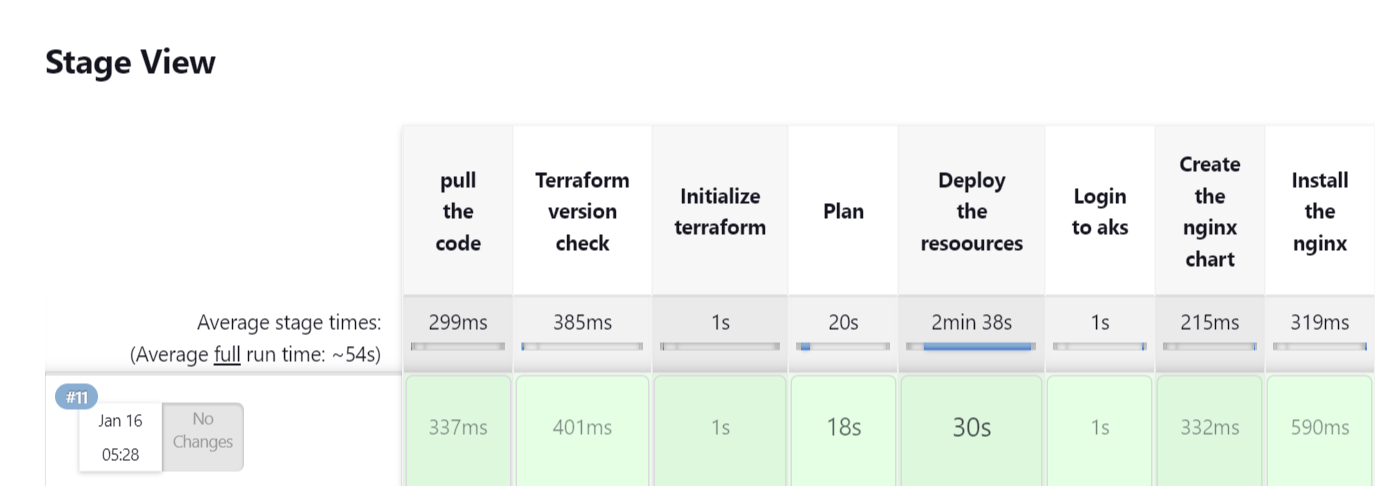
****

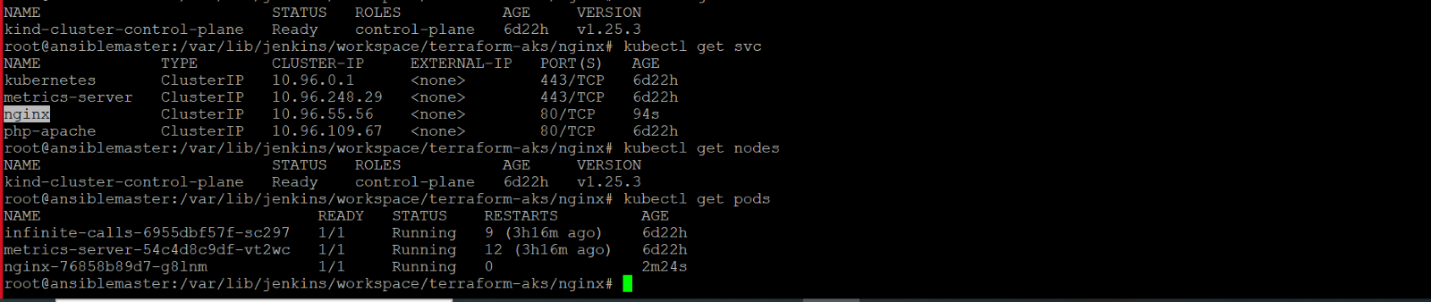
****

****

****

****

****

****