1. **Bitbucket code commit flow(if solution file present in .Net project)**

(1)cd "C:\Users\s.e.kumar.pradhan\git\_clone\ebex\_pr\_trans\Ebid" ==== go to the root file

(2)MSBuild.SonarQube.Runner.exe begin /k:"Ebid" /n:"Ebid" /v:"4.4" =====

(3)"C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\MSBuild\15.0\Bin\MSBuild.exe" "C:\Users\s.e.kumar.pradhan\git\_clone\ebex\_pr\_trans\Ebid\Source\e-auction.sln"

(4)MSBuild.SonarQube.Runner.exe end ===== (This will end the analysis and return the result to SonarQube server)

(2) java project pipeline code

node {

//stage('SCM-checkout') {

//git 'https://github.com/sambit54321/practice.git'

stage('SCM-checkout') {

def mvnHome = checkout([$class: 'GitSCM', branches: [[name: '\*/master']], doGenerateSubmoduleConfigurations: false, extensions: [], submoduleCfg: [], userRemoteConfigs: [[credentialsId: 'ae9aebd3-f496-4892-b313-e01a0ffa64f3', url: 'https://github.com/sambit54321/practice.git']]])

}

stage('Build') {

bat '''"C:\\apache-maven-3.6.0\\bin\\mvn" clean install " -f "C:\\Jenkins\\workspace\\pipeline\_maven\_test\\maven-simple-master\\pom.xml"'''

}

stage('Code-Analysis') {

bat "cd C:\\Jenkins\\workspace\\pipeline\_maven\_test"

bat "mvn sonar:sonar -f C:\\Jenkins\\workspace\\pipeline\_maven\_test\\maven-simple-master\\pom.xml"

}

}

1. **.Net project pipeline with script:**

ssss

**Net Application compiler execute:**

C:\New folder\ebex\_pr\_trans\MyPayMyChoice>C:\Windows\Microsoft.NET\Framework64\v4.0.30319\aspnet\_compiler.exe -p "C:\New folder\ebex\_pr\_trans\MyPayMyChoice\Source" /v /commercesite /u /f "C:\Myoutput"

* Type this command then see the created folder.
* First you have to create one folder then you have to store the code in that folders.

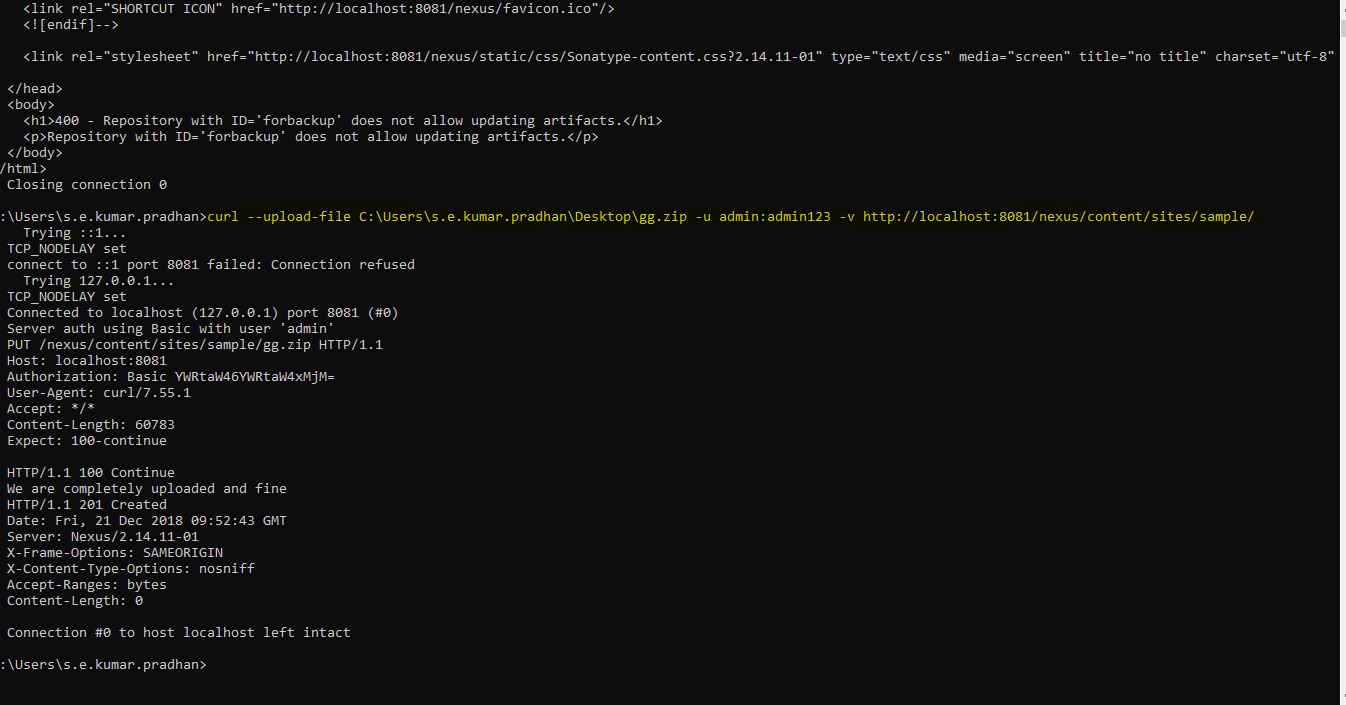
**FxCop:(Code Analysis)**

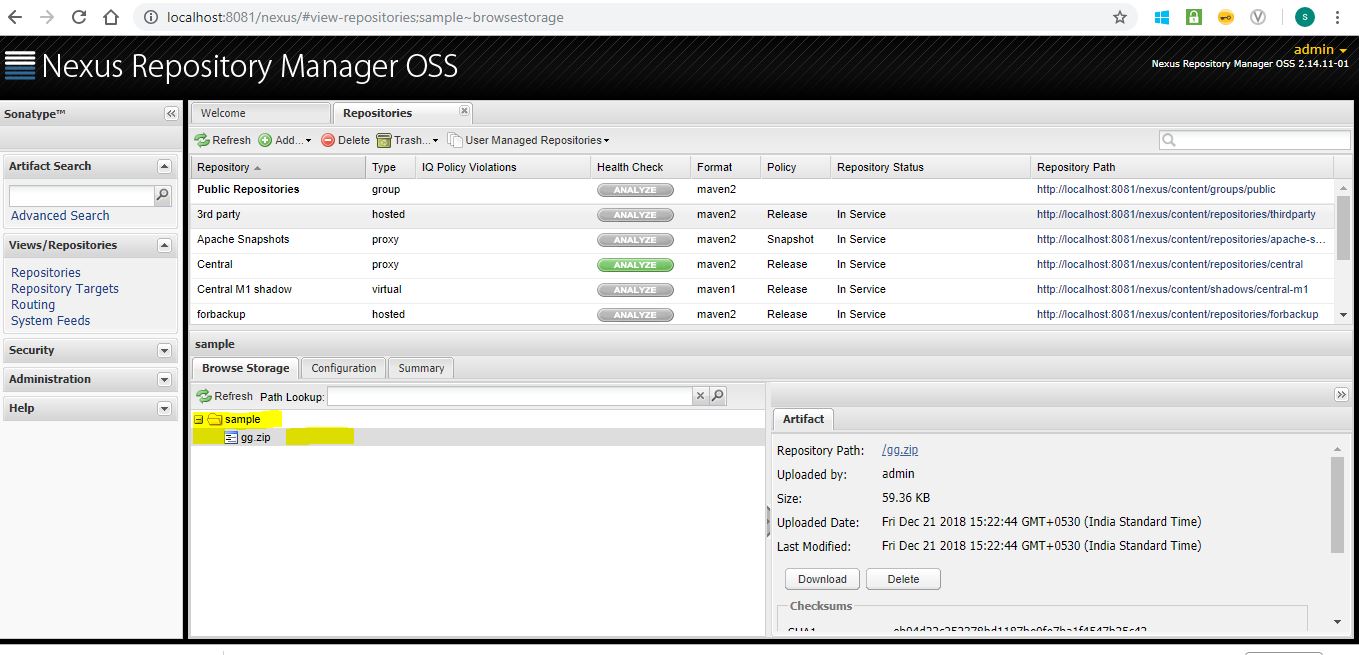
1. Cd <C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\Team Tools\Static Analysis Tools\FxCop> // cd FxCop folder path
2. FxCopCmd.exe /file:C:\Myoutput\bin\App\*.dll /rule:"C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\Team Tools\Static Analysis Tools\FxCop\Rules" /out:C:\Myoutput\fxcopreport.xml
3. Check myoutput folder you can see xml file

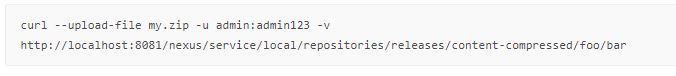
**UPLOAD ARTIFACT TO NEXUS BY COMMAND PROMPT:**

Go to command prompt and type the above command

curl -v -u admin:admin123 --upload-file pom.xml http://localhost:8081/nexus/content/repositories/releases/org/foo/1.0/foo-1.0.pom







**Flyway**

Flyway is a database migration tool.

Download flyway from <https://flywaydb.org>

Install h2 database.

(1) Go to flyway folder -> conf -> open flyway conf by notepad++

(2) then go to h2 database -> bin -> open h2 jar (it will open through internet explorer)

(3) go to flyway folder -> then go to sql folder -> write sql query and the you have go to cmd prompt and execute some command like

Flyway –version

Flyway migration

Flyway info

Flyway validates

Flyway clean

flyway repair

**sample code model:**

flyway.driver=org.hsqldb.jdbcDriver

flyway.url=jdbc:hsqldb:file:/db/flyway\_sample

flyway.user=SA

flyway.password=mySecretPwd

flyway.connectRetries=10

flyway.initSql=SET ROLE 'myuser'

flyway.schemas=schema1,schema2,schema3

flyway.table=schema\_history

flyway.locations=classpath:com.mycomp.migration,database/migrations,filesystem:/sql-migrations

flyway.sqlMigrationPrefix=Migration-

flyway.undoSqlMigrationPrefix=downgrade

flyway.repeatableSqlMigrationPrefix=RRR

flyway.sqlMigrationSeparator=\_\_

flyway.sqlMigrationSuffixes=.sql,.pkg,.pkb

flyway.encoding=ISO-8859-1

flyway.placeholderReplacement=true

flyway.placeholders.aplaceholder=value

flyway.placeholders.otherplaceholder=value123

flyway.placeholderPrefix=*#[*

flyway.placeholderSuffix=]

flyway.resolvers=com.mycomp.project.CustomResolver,com.mycomp.project.AnotherResolver

flyway.skipDefaultResolvers=false

flyway.callbacks=com.mycomp.project.CustomCallback,com.mycomp.project.AnotherCallback

flyway.skipDefaultCallbacks=false

Command-line: validate

Validate applied migrations against resolved ones (on the filesystem or classpath) to detect accidental changes that may prevent the schema(s) from being recreated exactly.

Validation fails if

* differences in migration names, types or checksums are found
* versions have been applied that aren’t resolved locally anymore
* versions have been resolved that haven’t been applied yet