

* CI & CD (Continuous Integration and Continuous Delivery, Continuous Deployment)
* Jenkins is an open-source Continuous Integration server written in Java for orchestrating a chain of actions to achieve the Continuous Integration process in an automated fashion.
* Jenkins supports the complete development life cycle of software from building, testing, documenting the software, deploying, and other stages of the software development life cycle.

----------------------------------------------------------------------------------------------------------------------------------------------------------------

**CONTINUOUS INTEGRATION:**

* Continuous Integration is a process of integrating code changes from multiple developers in a single project many times.
* The software is tested immediately after a code commit.
* With each code commit, code is built and tested. If the test is passed, the build is tested for deployment.
* If the deployment is successful, the code is pushed to production.
* This commit, build, test, and deploy is a continuous process and hence the name continuous integration/deployment.

=> Continuous Integration Server (Jenkins, Bamboo, Cruise Control, TeamCity, and others)

=> Source Control Tool (e.g., CVS, SVN, GIT, Mercurial, Perforce, ClearCase and others)

=> Build tool (Make, ANT, Maven, Ivy, Gradle, and others)

=> Automation testing framework (Selenium, Appium, Test Complete, UFT, and others)

----------------------------------------------------------------------------------------------------------------------------------------------------------------

**CMD COMMENT TO RUN THE JAR or WAR FILE:**

--> For Jar file

**java -jar jarname.jar**

--> for War file

**java -jar jarname.war**

----------------------------------------------------------------------------------------------------------------------------------------------------------------

**TO OPEN JENKINS:**

Type this in browser

**localhost:8080**

UserName: **admin**

PassWord: **531a436ee87f49cc83723fe209047c44**

"C:\Users\SUMAHALI\.jenkins\secrets\initialAdminPassword" (In this file you can see the password)

**CREATE A PROJECT:**

Step 1 – In Dashboard click on New Item.

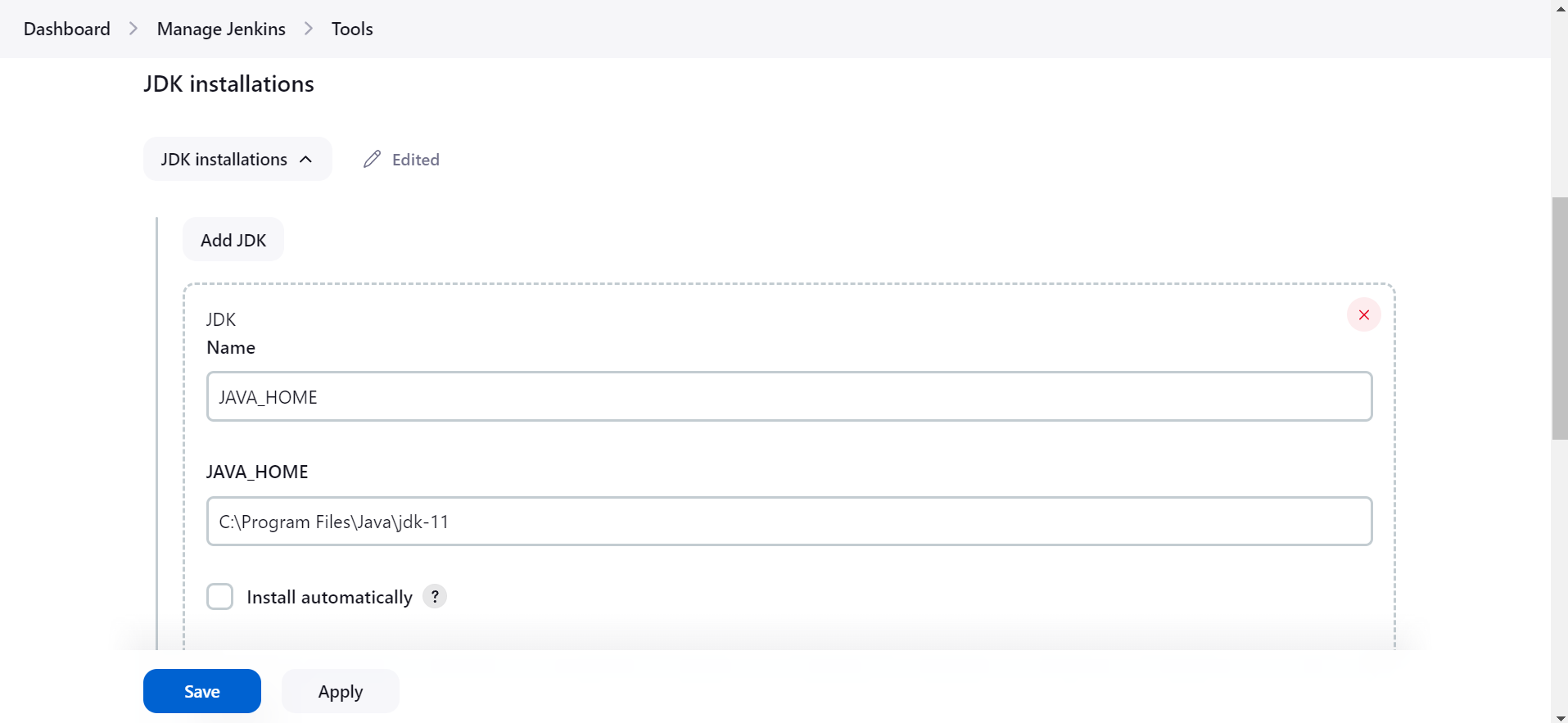
Step 2 – Give the name to the Project and select any one of available project templates and click ok.

Step 3 – Click on Save.

----------------------------------------------------------------------------------------------------------------------------------------------------------------

**ADD JDK IN JENKINS:**

Manage Jenkins >> Tools >> JDK Installations



----------------------------------------------------------------------------------------------------------------------------------------------------------------

**RUN A JAVA PROGRAM:**

Step 1 – Write a java Program in Notepad and save this file with **.java** extension.

Step 2 – Open the already created project.

Step 3 – Click on **Configure.**

Step 4 – build Steps >> Add Build Step >>Execute window Batch Command

Step 5 – Type the commands for run the java program.

javac HelloWorld.java

java HelloWorld

Step 6 – Click on **save.**

Step 7 – place the java file in the workspace of the project.

**"C:\Users\SUMAHALI\.jenkins\workspace\First Project\HelloWorld.java"**

If workspace folder is not visible. Open your project in Jenkins and click on **Build Now**. You can see the workspace

Step 8 – Click on **Build Now.**

Step 9 – See the log in **Console Output.**

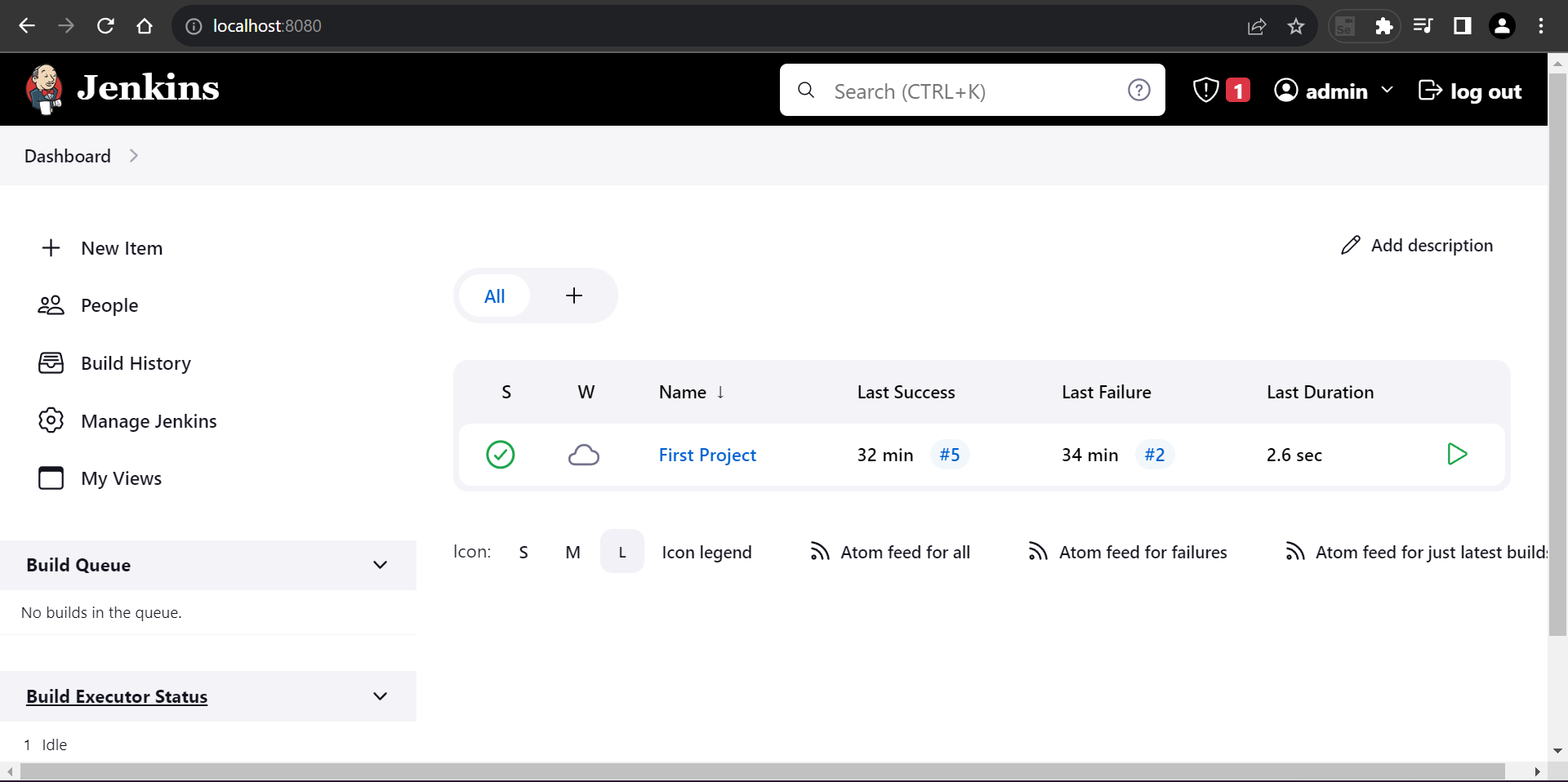
***public class HelloWorld{***

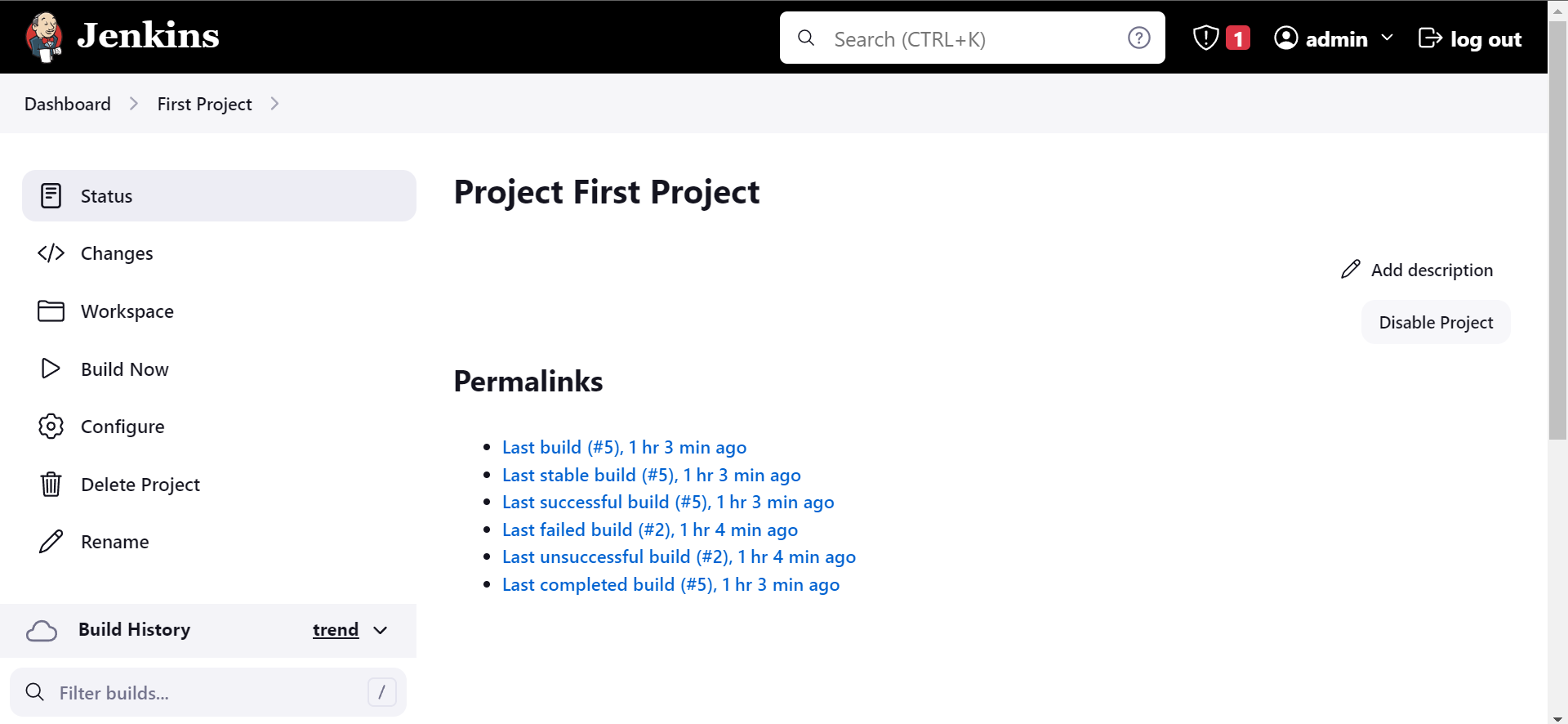
***public static void main(String[] args){***

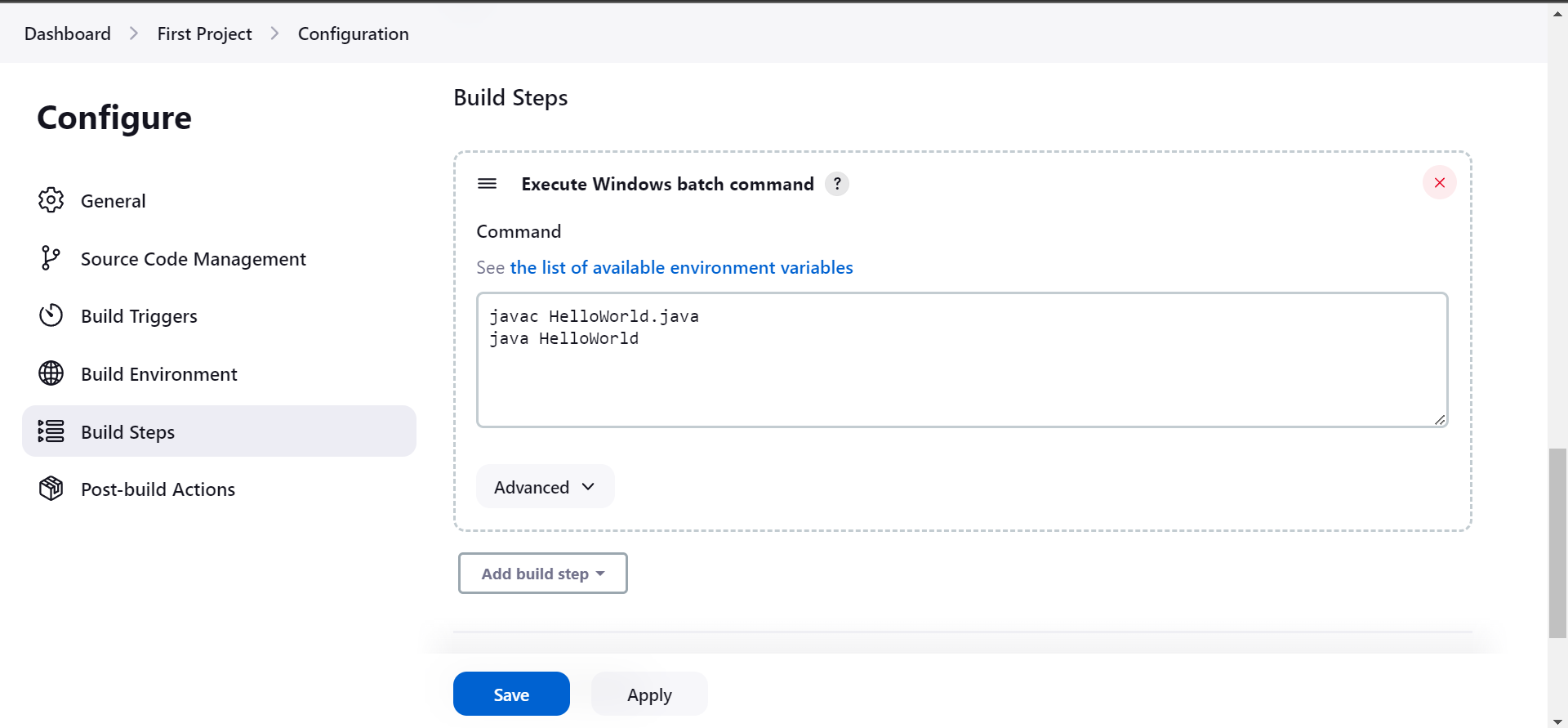
***System.out.println("Hello World!!!!");***

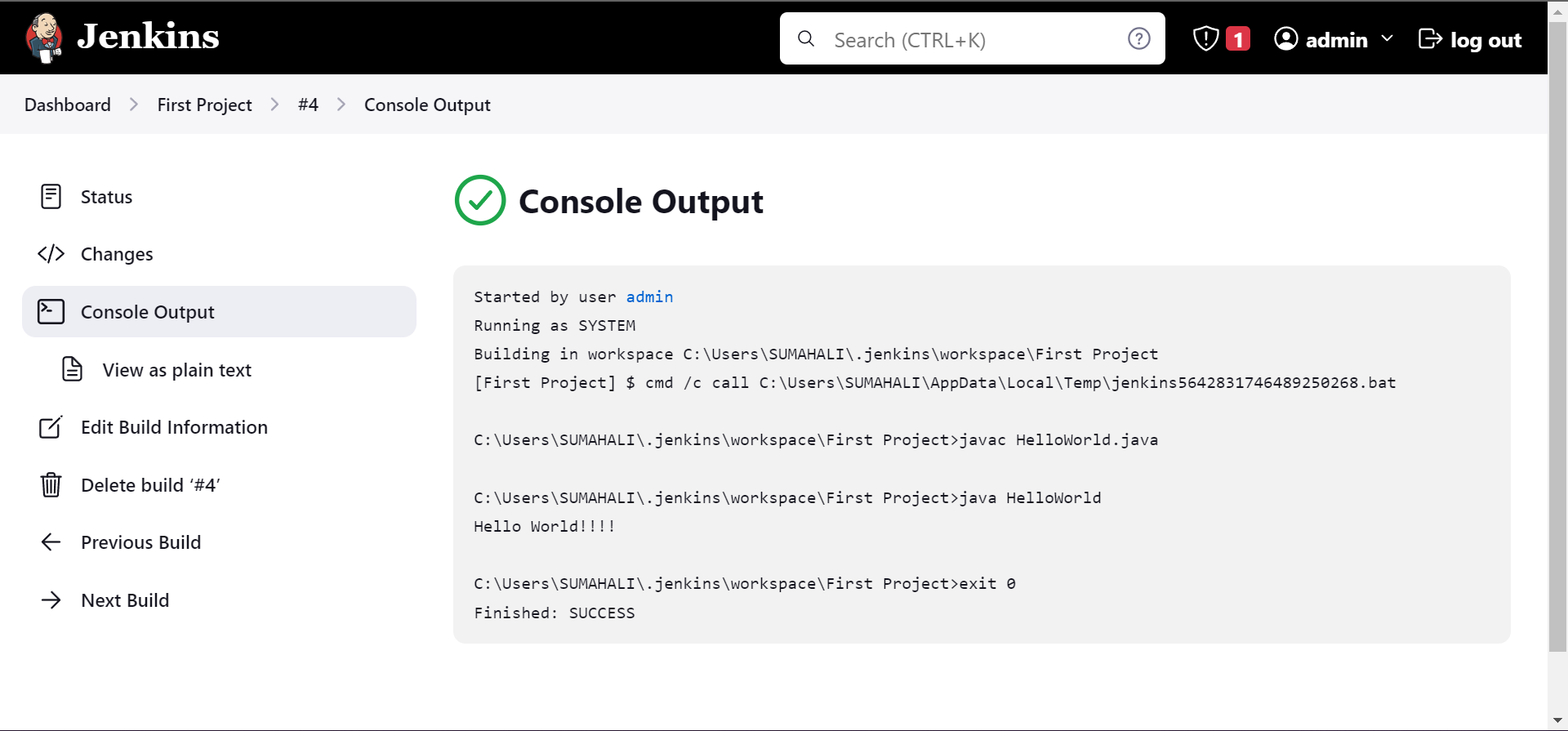
***}***

***}***









----------------------------------------------------------------------------------------------------------------------------------------------------------------

**BUILD TRIGGERS:**

**Schedule (Periodically):**

Step 1 – Click on **Configuration.**

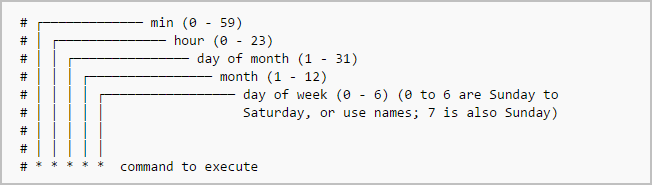
Step 2 – Click **Build Triggers.**

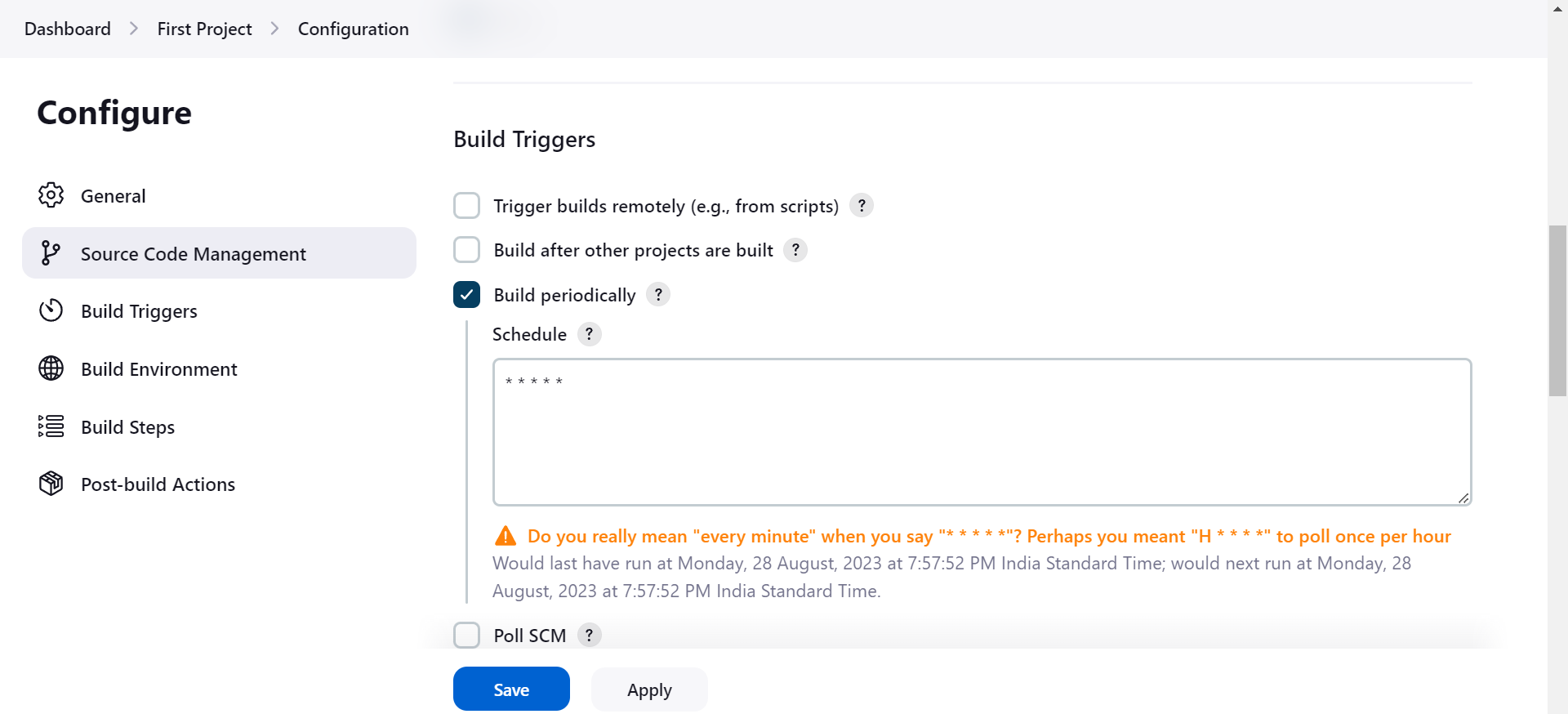
Step 3 – Select **Build Periodically.**

Step 4 – Enter schedule (Crontab). <https://crontab.guru/> - for more information

\* \* \* \* \* (Run Every minutes)

**Crontab:**



****

**Build After Other Project are Build:**

Step 1 – Create another one project.

Step 2 – Open the project and click **Configure**.

Step 3 – Build Triggers >> Build after other projects are built

Step 4 – Give the project name (followed by this project)

Step 5 – Run the previous project.

