Jenkins Tutorial -DevOps Tool

Jenkins Tutorial

1. Introduction

Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks such as building, testing, and deploying software. Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with the Java Runtime Environment installed. Jenkins is simply the old **Hudson** with a new name.

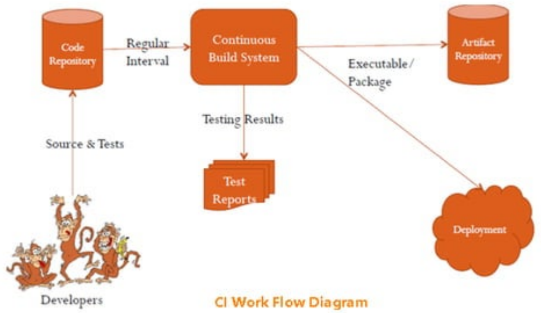
Jenkins is a software that allows **continuous integration**. Jenkins will be installed on a server where the central build will take place. The following flowchart demonstrates a very simple workflow of how Jenkins works.

1.1 Continuous Integration Workflow

Martin Fowler said "Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily – leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible"

At a regular frequency (ideally at every commit), the system is:

1. **Integrated –** All changes up until that point are combined into the project
2. **Built –** The code is compiled into an executable or package
3. **Tested –** Automated test suites are run
4. **Archived –** Versioned and stored so it can be distributed as is, if desired
5. **Deployed –** Loaded onto a system where the developers can interact with it



CI Work Flow Diagram

Continuous Integration Tools

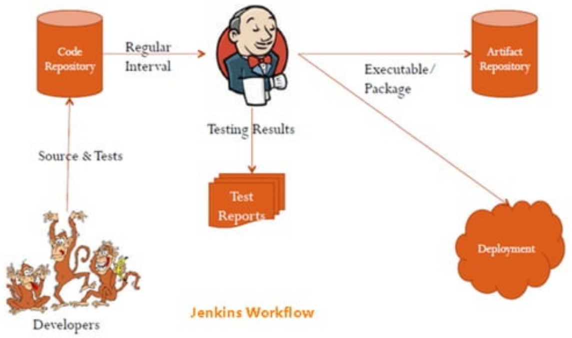
- Code Repositories : **SVN, Mercurial, Git**
- Continuous Build Systems : **Jenkins, Bamboo, Cruise Control**
- Test Frameworks : **JUnit, Cucumber, CppUnit**
- Artifact Repositories : **Nexus, Artifactory, Archiva**

1.2 Jenkins Tool Workflow

Jenkins is a **Java based Continuous Build System** Branched from Hudson, **Runs in servlet container (Glassfish, Tomcat).** It is supported by over 400 plugins like SCM, Testing, Notifications, Reporting, Artifact Saving, Triggers, and External Integration etc.

In 2005 – Hudson was first release by Kohsuke Kawaguchi of Sun Microsystems. **2010 – Oracle** bought Sun Microsystems Due to a naming dispute, **Hudson was renamed to Jenkins**

Oracle continued development of Hudson (as a branch of the original)



Key Features of Jenkins

- It will Generate test reports
- Jenkins can Integrate with many different Version Control Systems
- Jenkins will Deploys directly to production or test environments and many more

2. Jenkins Installation

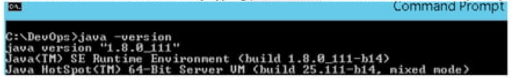
Requirements

- Java 7 or Java 8 must be installed.
- Requires minimum RAM of 512MB

2.1 Installing Jenkins in Windows

1.Download Jenkins.war.**(if it is downloaded in .zip format rename/change extension to .war)**

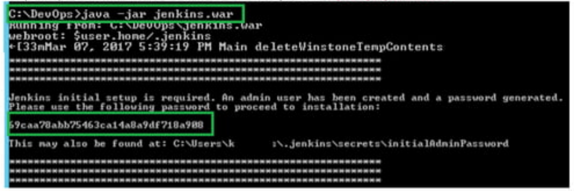
2.Check Java is installed or not by typing java -version command in command prompt.



Command Prompt

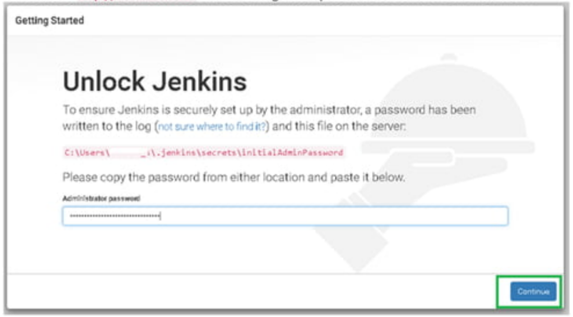
```
C:\DevOps>java -version
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_111-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.111-b14, mixed mode)
```

3.Open up a terminal in the download directory and run java -jar jenkins.war



```
C:\DevOps>java -jar jenkins.war
Running from: C:\DevOps\jenkins.war
webroot: $user.home/.jenkins
←[33mMar 07, 2017 5:39:19 PM Main deleteWinstoneTempContents
*************************************************************
*************************************************************
*************************************************************

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

69caa78abb75463ca14a8a9df718a908

This may also be found at: C:\Users\k      :\.jenkins\secrets\initialAdminPassword

*************************************************************
*************************************************************
*************************************************************
```

4.Browse to http://localhost:8080 and enter highlated password to continue the installation



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

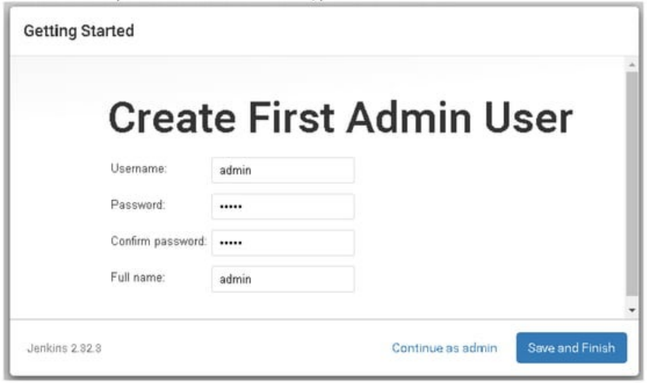
`C:\Users\        _:\.jenkins\secrets\initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

`································`

5.Next it will takes you to **“Create First Admin User”,** provide details & **finish**



Getting Started

Create First Admin User

Username: admin

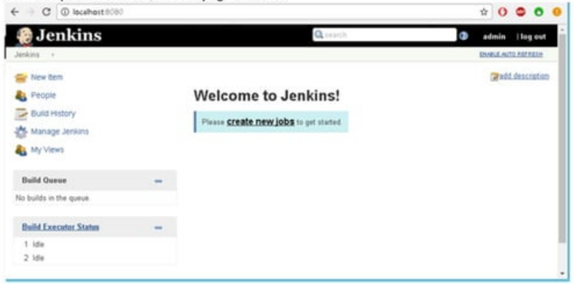
Password: •••••

Confirm password: •••••

Full name: admin

Jenkins 2.32.3

6.It will opens the Jenkins Home page as below



localhost:8080

Welcome to Jenkins!

Please **create new jobs** to get started

Build Queue

No builds in the queue

Build Executor Status

1 Idle

2 Idle

We can also install Jenkins using Apache Tomcat also. Just download & Start the tomcat. Upload the Jenkins.war in tomcat from admin panel. You can access by using http://localhost:8080/jenkins

2.2 How to Change Jenkins Port number

Some times 8080 is busy with some other services. In that case we can change port to some other number by using following steps

1.Press Ctrl+C on Jenkins command line to Stop the Service

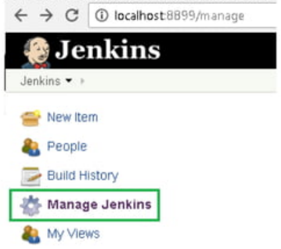
2.It it is Jenkins.war Installation, Start Jenkins from cmd line using :

```
java -jar jenkins.war --httpPort=8081
```

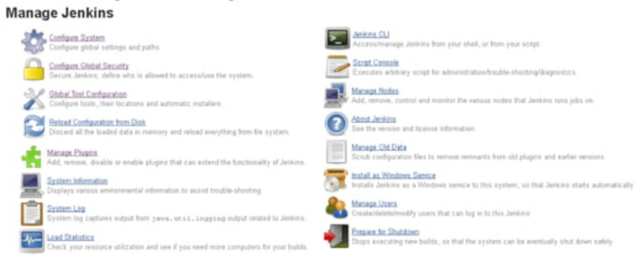
3.If is tomcat Installation, open **xml** & change `"--httpPort=8080"` with new port number

3. Jenkins Configuration

We can configure Jenkins jobs based up on our requirement. For doing any configuration we have to go to **Manage Jenkins** on the left menu of the Dashboards.



It contains following modules for configuration



Manage Jenkins

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

System Information
Displays various environmental information to assist trouble-shooting.

System Log
System log captures output from `java.util.logging` output related to Jenkins.

Load Statistics
Check your resource utilization and see if you need more computers for your builds.

Jenkins CLI
Access/manage Jenkins from your shell, or from your script.

Script Console
Executes arbitrary script for administration/trouble-shooting/diagnostics.

Manage Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

About Jenkins
See the version and license information.

Manage Old Data
Scrub configuration files to remove remnants from old plugins and earlier versions.

Install as Windows Service
Installs Jenkins as a Windows service to this system, so that Jenkins starts automatically.

Manage Users
Create/delete/modify users that can log in to this Jenkins

Prepare for Shutdown
Stops executing new builds, so that the system can be eventually shut down safely.

We have to use the above configuration as per our requirement. As of now we don't all of them.we need some basic configuration required for working with Jenkins.

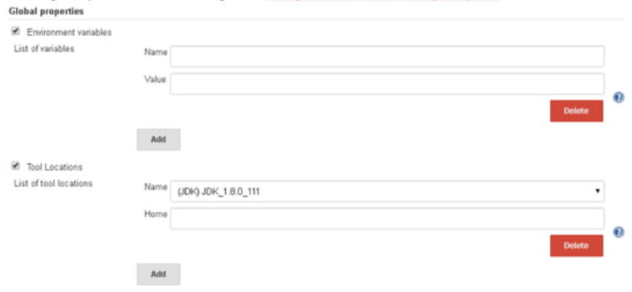
For that we need to Install and Configure below Tools/ Softwares

- Install **Java**
- Install **Git** (just download & install as normal software)
- Install **Maven**
- Install **Ant**

3.1 Configure System

Here we can manage paths to the various tools to use in builds, such as the JDKs, the versions of Ant and Maven, as well as security options, email servers, and other system-wide configuration details. When plugins are installed. Jenkins will add the required configuration fields dynamically after the plugins are installed.

For Configure System we have to navigate to **Manage Jenkins →Select Configure System**


Global properties

✓ Environment variables

List of variables

Name

Value

✓ Tool Locations

List of tool locations

Name: (JDK) JDK_1.8.0_111

Home

3.2 Configure Global Security

It is used for SecuringJenkins & define who is allowed to access/use the system.

The ***Configure Global Security*** page has two sections in which you:

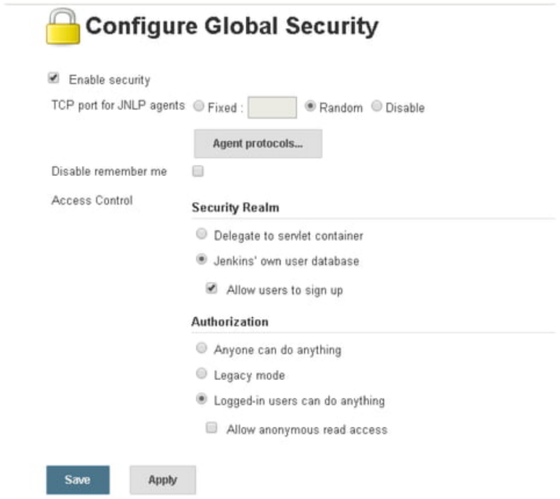
- Set the security realm to determine who is allowed access
- Set the authorization to determine what each user is allowed to do

-

Jenkins' Own User Database

This is the simplest authentication scheme–Jenkins maintains its own independent user database. People can sign up for their own accounts, and you as the administrator decide who can do what in Jenkins.

1. Go to the Jenkins **dashboard**, usu ally http://server:8080 or http://server/jenkins:8080, where server is the host on which Jenkins is running
2. Select **Manage Jenkins**, then **Configure Global Security**
3. Click **Enable Securit** The page will expand to offer a choice of access control.
4. Select **Jenkins' own user database & check** mark next **to Allow users to sign up**
5. Continue with Authorization, below. In particular, do not forget to press the Save button at the bottom of the page.


Configure Global Security

[✓] Enable security

TCP port for JNLP agents: ○ Fixed : [ ] ● Random ○ Disable

Agent protocols...

Disable remember me [ ]

Access Control

Security Realm

○ Delegate to servlet container

● Jenkins' own user database

[✓] Allow users to sign up

Authorization

○ Anyone can do anything

○ Legacy mode

● Logged-in users can do anything

[ ] Allow anonymous read access

Save Apply

3.3 Global Tool Configuration

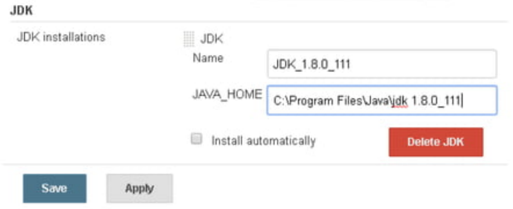
The Global Tool Configuration lets you define variables that can be managed centrally but used in all of your build jobs. You can add as many properties as you want here, and use them in your build jobs. Jenkins will make them available within your build job environment, so you can freely use them within your Ant and Maven build scripts. Note that you shouldn't put periods **(".") in** the property names, as they won't be processed correctly.

3.3.1 Java Configuration in Jenkins

1.Navigate to Manage Jenkins → Global Tool Configuration→JDK Installations →Add JDK

2.Set JDK Name. For Ex: JDK_1.8.0_111 (uncheck Install automatically)

3.Set JAVA_HOME directory path. For Ex: C:\Program Files\Java\jdk 1.8.0_111



Name: JDK_1.8.0_111

JAVA_HOME: C:\Program Files\Java\jdk 1.8.0_111

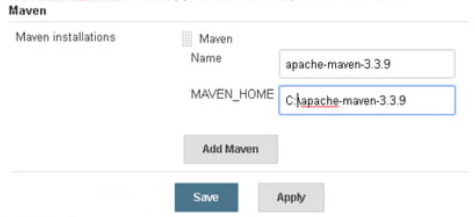
4.Apply & Save.

3.3.2 Apache Maven Configuration in Jenkins

1.Navigate to **Manage Jenkinks→ Global Tool Configuration→ Maven Installations→ Add Maven**

2.Set Maven Name. For Ex: apache-maven-3.3.9 (uncheck Install automatically)

3.Set MAVEN_HOME directory path. For Ex: C: \apache-maven-3.3.9



Name: apache-maven-3.3.9

MAVEN_HOME: C:\apache-maven-3.3.9

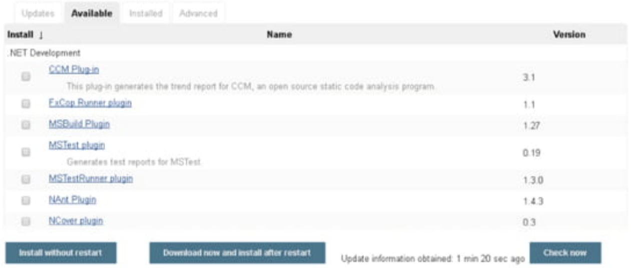
4.Apply & Save.

3.4 Reload Configuration from Disk

This Option will Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk. Be careful using this, it won't display any page directly deletes the data when you click on OK button on alert message!!

3.5 Manage Plugins

Here we can Add, remove, disable or enable plugins that can extend the functionality of Jenkins.



Install ↓	Name	Version
.NET Development		
☐	CCM Plug-in This plug-in generates the trend report for CCM, an open source static code analysis program.	3.1
☐	FxCop Runner plugin	1.1
☐	MSBuild Plugin	1.27
☐	MSTest plugin Generates test reports for MSTest.	0.19
☐	MSTestRunner plugin	1.3.0
☐	NAnt Plugin	1.4.3
☐	NCover plugin	0.3

Update information obtained: 1 min 20 sec ago

We can install the plugins directly from Available plugins tab, and uninstall when ever you want

How to add Jenkins Plugins manually

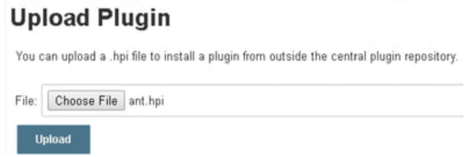
Some time proxy settings won't allow us to contact Jenkins CI server directly. In those cases we can install Jenkins plugins by downloading them manually. Follow below Steps to do so

1.Go to https://plugins.jenkins.io/

2.Search the plugin you want & click on plugin page

3.on the right side of the page, click on Archives, select latest ,it will download plugin in **.hpi format**

4.Now go to Manage Plugins→ Advanced **tab** come down & **upload plugin**



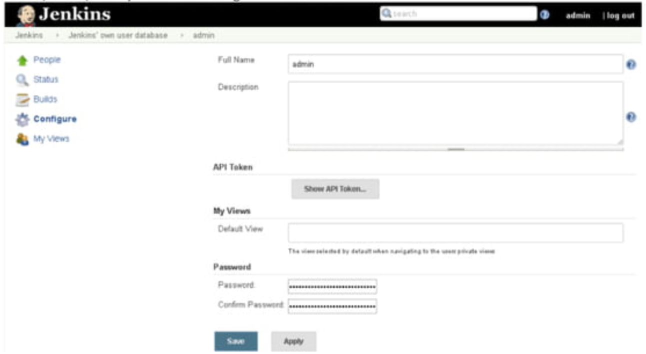
Upload Plugin

You can upload a .hpi file to install a plugin from outside the central plugin repository.

File: ant.hpi

3.6 Manage Users

Create/delete/modify users that can log in to this Jenkins



Jenkins › Jenkins' own user database › admin

Full Name: admin

Description:

API Token

My Views

Default View:

The view selected by default when navigating to the users private views

Password

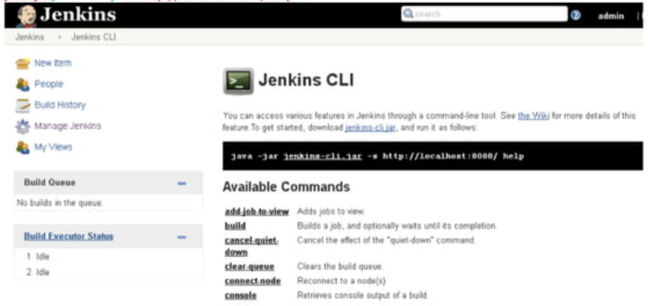
Password: ••••••••••••••••••••••••••••

Confirm Password: ••••••••••••••••••••••••••••

3.7 Jenkins CLI

Access/manage Jenkins from your shell, or from your script. You can access various features in Jenkins through a command-line tool. See the Wiki for more details of this feature.To get started, download jenkins-cli.jar, and run it as follows:

```
java -jar jenkins-cli.jar -s http://localhost:8080/ help
```



Jenkins CLI

You can access various features in Jenkins through a command-line tool. See the Wiki for more details of this feature.To get started, download jenkins-cli.jar, and run it as follows:

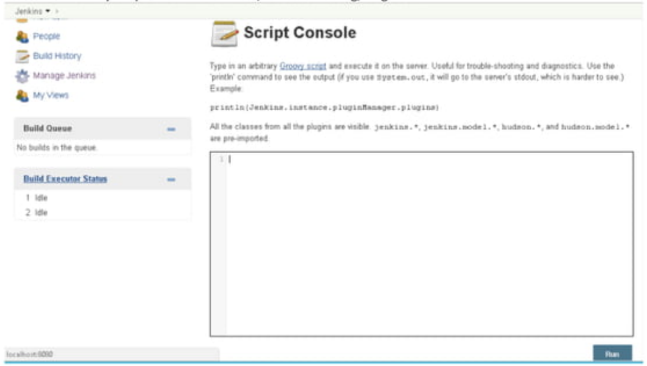
```
java -jar jenkins-cli.jar -s http://localhost:8080/ help
```

Available Commands

add-job-to-view	Adds jobs to view.
build	Builds a job, and optionally waits until its completion.
cancel-quiet-down	Cancel the effect of the "quiet-down" command.
clear-queue	Clears the build queue.
connect-node	Reconnect to a node(s)
console	Retrieves console output of a build.

3.8 Script Console

Executes arbitrary script for administration/troubleshooting/diagnostics.



Script Console

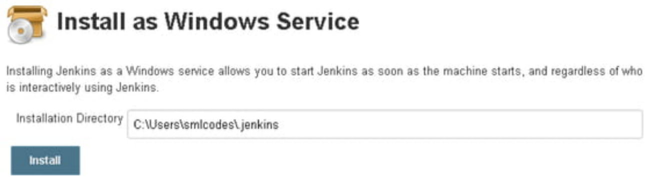
Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use `System.out`, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

3.9 Install as Windows Service

Installs Jenkins as a Windows service to this system, so that Jenkins starts automatically when the machine boots.



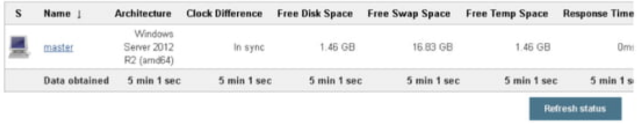
Install as Windows Service

Installing Jenkins as a Windows service allows you to start Jenkins as soon as the machine starts, and regardless of who is interactively using Jenkins.

Installation Directory: C:\Users\smlcodes\.jenkins

3.10 Manage Nodes

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.



S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
🖥	master	Windows Server 2012 R2 (amd64)	In sync	1.46 GB	16.83 GB	1.46 GB	0m
	Data obtained	5 min 1 sec	5 min 1 sec	5 min 1 sec	5 min 1 sec	5 min 1 sec	5 min 1 s

And also we have some other things in manage Jenkins, they are

- **Manage Old Data**: remove remnants from old plugins and earlier versions.
- **System Information:** Displays various environmental information to assist trouble-shooting.
- **System Log**: system log captures output from java.util.logging output related to Jenkins.
- **Load Statistics :** Check your resource utilization
- **Prepare for Shutdown:** Stops executing new builds, so that the system can be shut down safely.

3.11Jenkins Pipeline

Jenkins Pipeline is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins

Typically, this "Pipeline as Code" would be written to a **Jenkinsfile** and checked into a project's source control repository, for example:

Jenkinsfile (Declarative Pipeline)

```
pipeline {
  agent any

  stages {
    stage('Build') {
      steps {
        echo 'Building..'
      }
    }
    stage('Test') {
      steps {
        echo 'Testing..'
      }
```

```
        }
        stage('Deploy') {
            steps {
                echo 'Deploying....'
            }
        }
    }
}
```