

# Continuous delivery pipeline configuration

This file explains how to configure a continuous delivery pipeline for our project using NetBeans, Jenkins, Maven and GlassFish.

We will create three GlassFish servers that represent development, testing and production environments, our application will be deployed to these servers.

## MySQL connector

For MySQL to work on the GlassFish servers, download mysql-connector-java-5.1.47.jar on this site: <https://mvnrepository.com/artifact/mysql/mysql-connector-java/5.1.47>

Put this file in `C:\Program Files\glassfish-4.1.1\glassfish\lib` (the GlassFish installation directory may be different).

## System configuration

The following environment variables must be configured (paths may be a bit different):

- GLASSFISH\_HOME = `C:\Program Files\glassfish-4.1.1`
- MAVEN\_HOME = `C:\Program Files\NetBeans 8.2\java\maven`

Also add the followings to your Path variable:

- `%GLASSFISH_HOME%\bin`
- `%MAVEN_HOME%\bin`

## Creating GlassFish domains

A GlassFish domain must be created for each environment.

To create the development domain, open the console and use the following command (**change the path to the location where you want to create the domains**):

```
asadmin create-domain --domainindir "C:\Users\vinro\OneDrive\Documents\Efrei\M1\Java EE\Glassfish domains" --adminport 5050 --instanceport 5051 development
```

When prompted, enter "admin" for both user name and password.

```
PS C:\> asadmin create-domain --domainid "C:\Users\vinro\OneDrive\Documents\Efrei\M1\Java EE\Glassfish domains" --
adminport 5050 --instanceport 5051 development
Enter admin user name [Enter to accept default "admin" / no password]>admin
Enter the admin password [Enter to accept default of no password]>
Enter the admin password again>
Using port 5050 for Admin.
Using port 5051 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using default port 8181 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Using default port 6666 for OSGI_SHELL.
Using default port 9009 for JAVA_DEBUGGER.
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=VINCENT-ASUS,OU=GlassFish,O=Oracle Corporation,L=Santa Clara,ST=California,C=US]
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=VINCENT-ASUS-instance,OU=GlassFish,O=Oracle Corporation,L=Santa Clara,ST=California,C=US]
Domain development created.
Domain development admin port is 5050.
Domain development admin user is "admin".
Command create-domain executed successfully.
```

The development domain is now created.

Use the same command to create domains for testing and production environments:

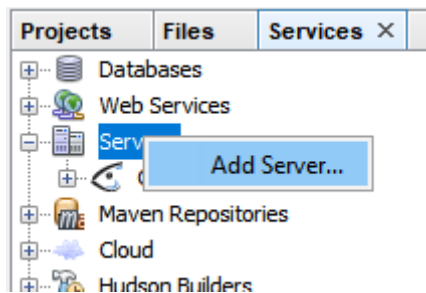
```
asadmin create-domain --domainid "C:\Users\vinro\OneDrive\Documents\Efrei\M1\Java
EE\Glassfish domains" --adminport 6060 --instanceport 6061 testing
```

```
asadmin create-domain --domainid "C:\Users\vinro\OneDrive\Documents\Efrei\M1\Java
EE\Glassfish domains" --adminport 7070 --instanceport 7071 production
```

## GlassFish server configuration

We will first create the development server.

In NetBean's Services tab, right click "Servers", and add a server:



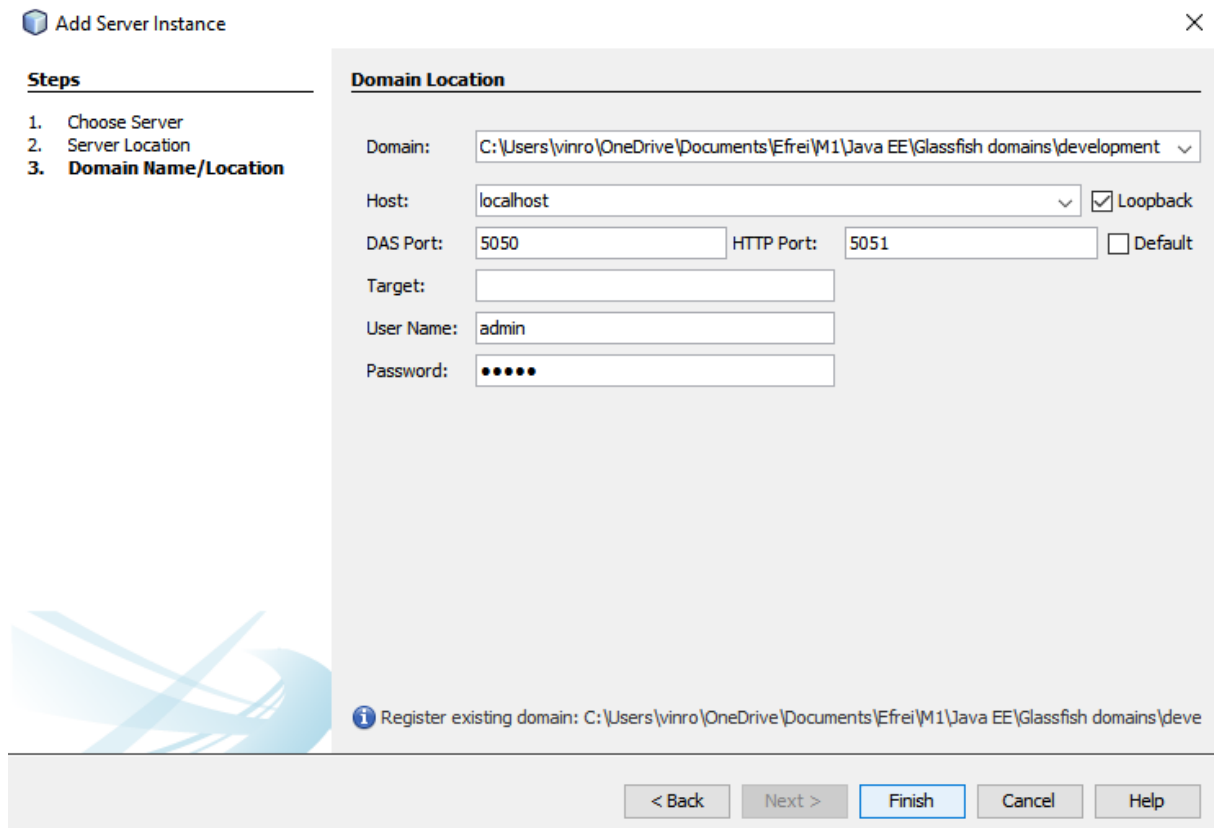
Select “GlassFish Server”, give your server a name such as “GlassFish Server (development)”, and click “Next”:

The screenshot shows the 'Add Server Instance' wizard window. On the left, under the 'Steps' section, step 1 is 'Choose Server' and step 2 is '...'. The main panel is titled 'Choose Server' and contains a list of server options: 'Apache Tomcat or TomEE', 'GlassFish Server' (which is highlighted with a blue selection bar), 'JBoss Application Server', 'Oracle WebLogic Server', and 'WildFly Application Server'. Below the list is a text field labeled 'Name:' containing the text 'GlassFish Server (development)'. At the bottom of the window are five buttons: '< Back', 'Next >' (highlighted with a blue border), 'Finish', 'Cancel', and 'Help'.

On the next page, set your GlassFish installation location and click “Next”:

The screenshot shows the 'Add Server Instance' wizard window at the 'Server Location' step. The left 'Steps' section shows step 1 as 'Choose Server', step 2 as 'Server Location' (highlighted), and step 3 as 'Domain Name/Location'. The main panel is titled 'Server Location' and features an 'Installation Location:' label above a text field containing 'C:\Program Files\glassfish-4.1.1', with a 'Browse...' button to its right. Below this are two radio buttons: 'Local Domain' (selected) and 'Remote Domain'. There is a 'Download Now...' button and a checkbox labeled 'I have read and accept the license agreement... (click)'. At the bottom left, a yellow warning icon is followed by the text: 'No usable default domain. Use Next to create a personal domain.' The bottom of the window contains the same set of navigation buttons as the previous step: '< Back', 'Next >' (highlighted), 'Finish', 'Cancel', and 'Help'.

On the next page, set the path to the GlassFish domain you created before, set “DAS Port” to 5050, “HTTP Port” to 5051, and both “User Name” and “Password” to “admin”:



The screenshot shows the 'Add Server Instance' dialog box with the 'Domain Location' tab selected. The 'Steps' list on the left indicates the current step is '3. Domain Name/Location'. The 'Domain Location' tab contains the following fields:

- Domain: C:\Users\vinro\OneDrive\Documents\Efrei\M1\Java EE\Glassfish domains\development
- Host: localhost (with a checked 'Loopback' checkbox)
- DAS Port: 5050 (with an unchecked 'Default' checkbox)
- HTTP Port: 5051
- Target: (empty field)
- User Name: admin
- Password: (masked with dots)

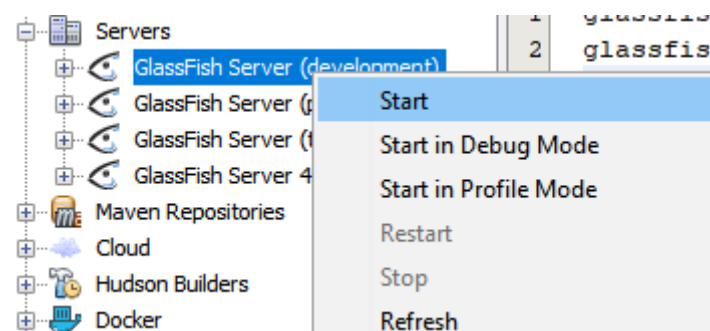
At the bottom, there is a status bar with the text: 'Register existing domain: C:\Users\vinro\OneDrive\Documents\Efrei\M1\Java EE\Glassfish domains\deve'. The bottom of the dialog has buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Click finish and a new GlassFish server will be created.

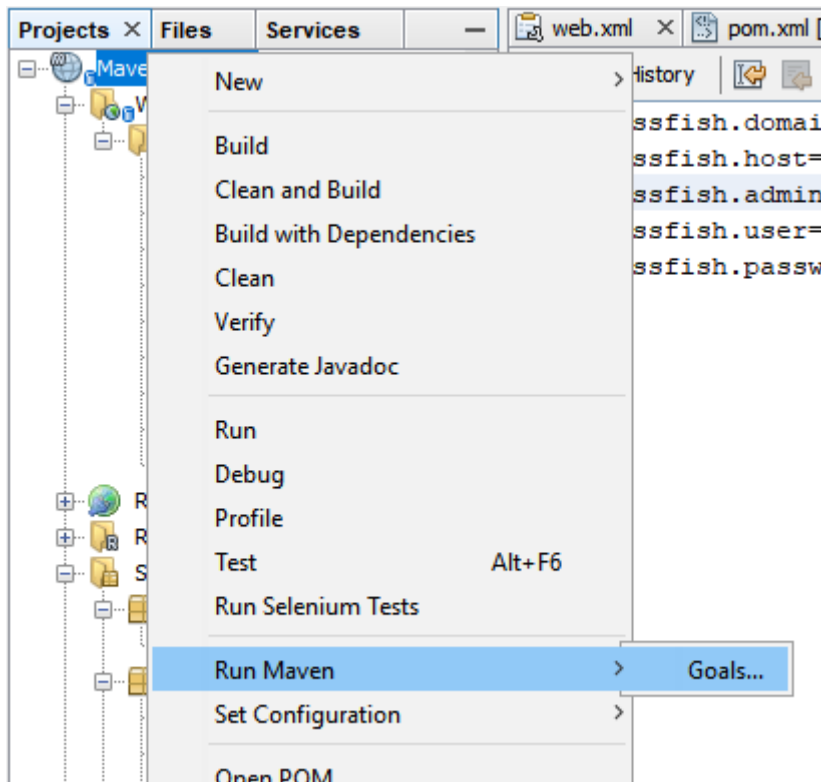
Repeat these steps to create two other servers for the testing and production environments. Use these settings:

- Testing server:
  - Name: “testing”
  - DAS Port: 6060
  - HTTP Port: 6061
- Production server:
  - Name: “production”
  - DAS Port: 7070
  - HTTP Port: 7071

We will try the development server. Right-click it and select “Start”:



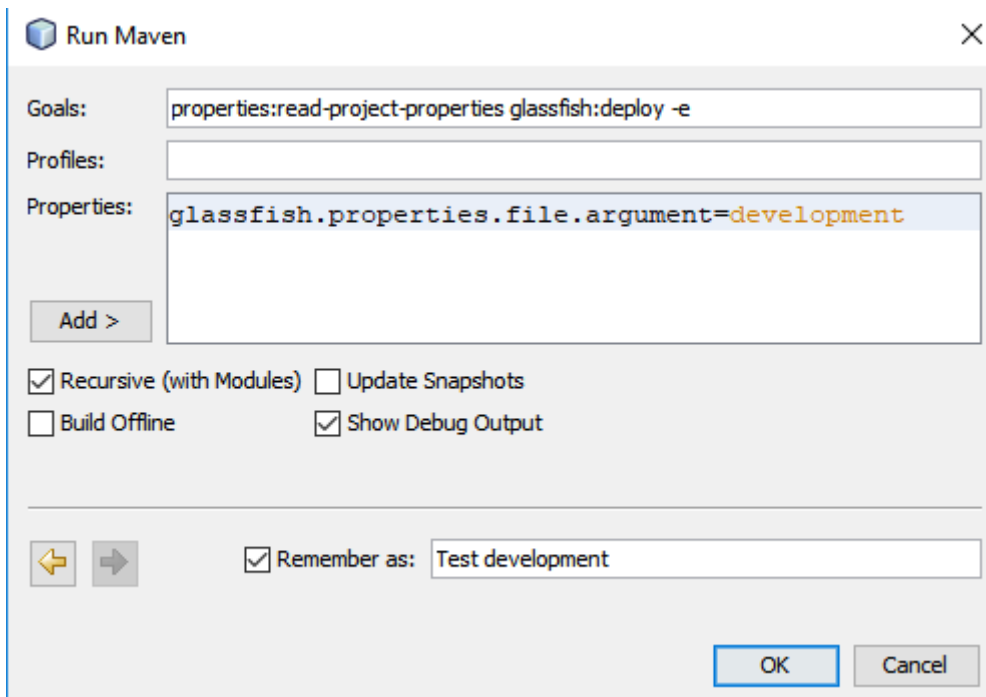
Test your configuration by running a Maven Goal:



Enter the following and click OK:

Goals: *properties:read-project-properties glassfish:deploy -e*

Properties: *glassfish.properties.file.argument=development*



After the build ends, connect to your server console at <http://localhost:5050/> and log in using “admin” as both user name and password.

Click “List Deployed Applications”:



Then, click “Launch” on the right of “MavenPROJECT”:

## Applications

Applications can be enterprise or web applications, or various kinds of modules. Restart an application or module by clicking on the reload link, this action will apply only to the targets that the application or module is enabled on.

Deployed Applications (1)					
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">Deploy...</a>	<a href="#">Undeploy</a>	<a href="#">Enable</a>	<a href="#">Disable</a>
		Filter:	<input type="text"/>		
Select	Name	Deployment Order	Enabled	Engines	Action
<input type="checkbox"/>	MavenPROJECT	100	✓	ejb, web	<a href="#">Launch</a> <a href="#">Redeploy</a>   <a href="#">Reload</a>

GlassFish gives you two links to visit the application, click on the first one and you should be redirected to the employee management application:

## Web Application Links

If the server or listener is not running, the link may not work. In this event, check the status of the server instance.

**Application Name:** MavenPROJECT

**Links:**  
[server] <http://VINCENT-ASUS:5051/MavenPROJECT-1.0-SNAPSHOT>  
[server] <https://VINCENT-ASUS:8181/MavenPROJECT-1.0-SNAPSHOT>

## Jenkins configuration

### Plugins

Make sure the following Jenkins plugins are installed:

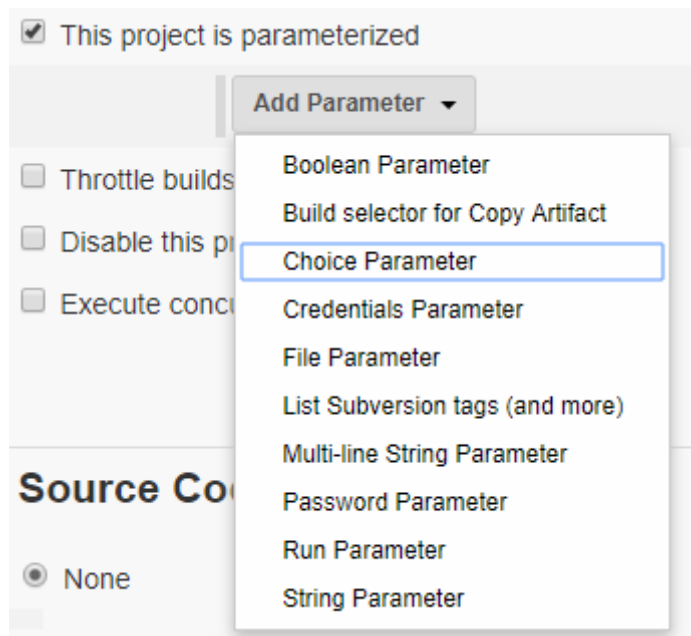
- Build with parameters
- Copy artifact
- GIT plugin
- Parameterized trigger plugin
- Maven integration plugin
- GitHub plugin

### Configure the Jenkins job

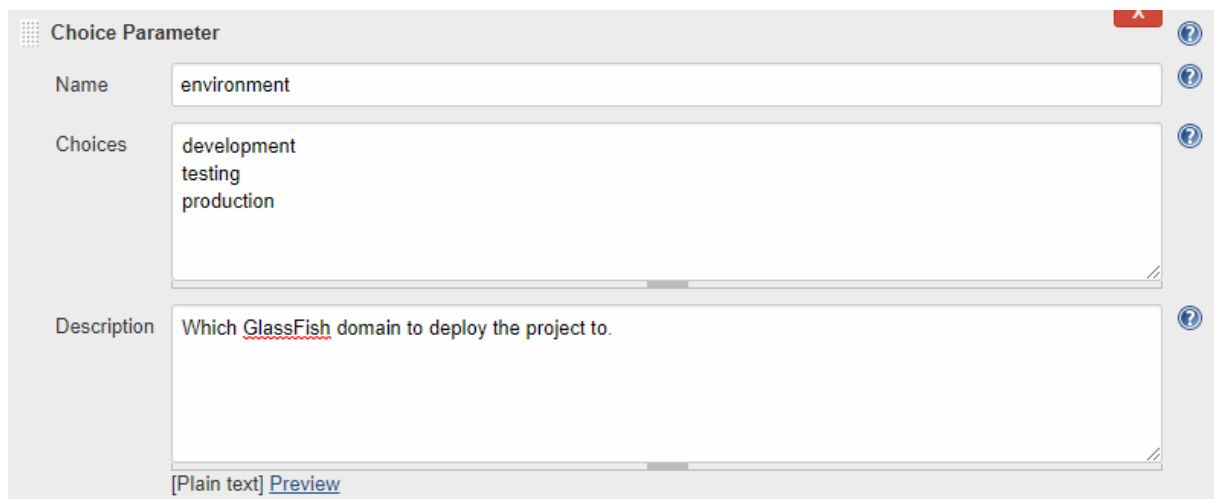
Create a new job, give it a name, and select “Freestyle project”.

To select the environment we want to deploy the project to, we add a parameter to the build.

Under “General”, check the “This project is parameterized option” and add a “Choice Parameter”:



Configure the parameter:



The image shows the 'Choice Parameter' configuration window in Jenkins. It has a title bar with a red close button and a help icon. The window contains three main sections: 'Name' with a text field containing 'environment', 'Choices' with a text area containing 'development', 'testing', and 'production', and 'Description' with a text area containing 'Which GlassFish domain to deploy the project to.'. At the bottom, there is a '[Plain text] Preview' link.

**Choice Parameter**

Name: environment

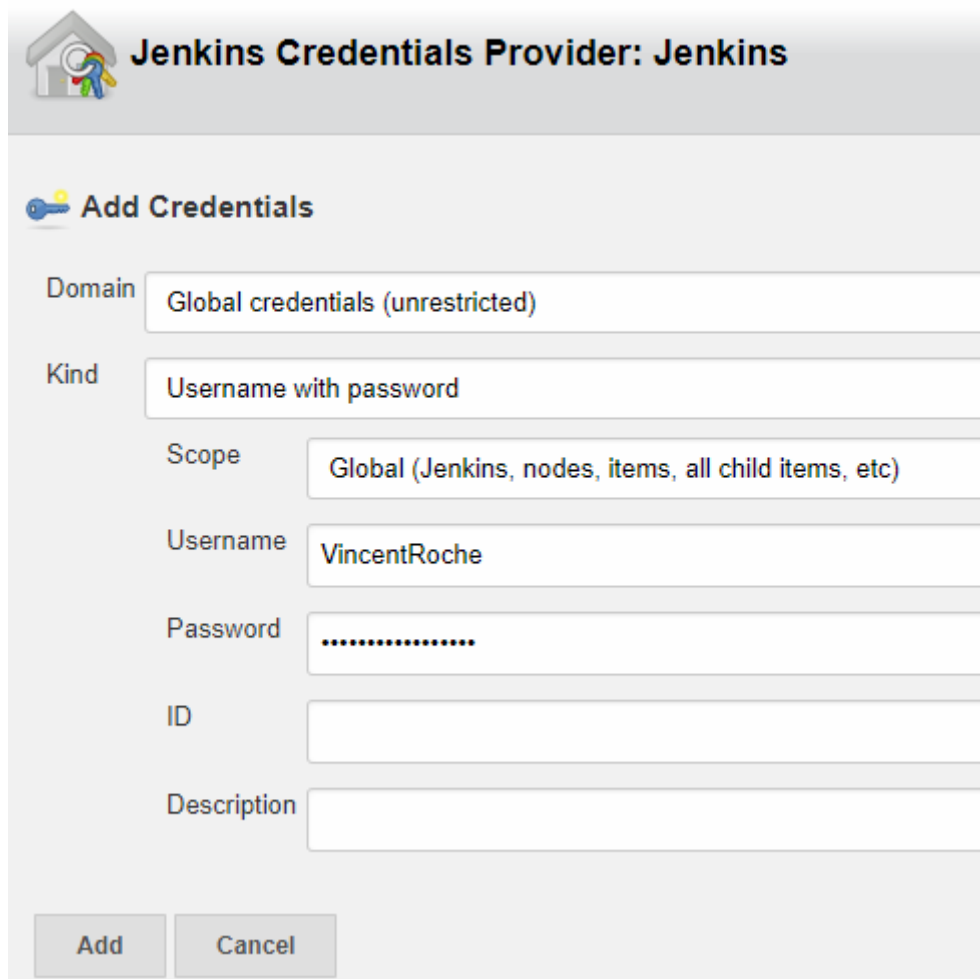
Choices: development, testing, production

Description: Which GlassFish domain to deploy the project to.

[Plain text] [Preview](#)

In the “Source Code Management” section, select “Git”.

Set <https://github.com/ClaireH97/JEE2.git> as the repository URL, and click *Add* → *Jenkins* to add your GitHub credentials:



The image shows the 'Jenkins Credentials Provider: Jenkins' dialog. It has a title bar with a house icon and the title 'Jenkins Credentials Provider: Jenkins'. Below the title bar is a section 'Add Credentials' with a key icon. The form contains several fields: 'Domain' with a dropdown menu showing 'Global credentials (unrestricted)', 'Kind' with a dropdown menu showing 'Username with password', 'Scope' with a dropdown menu showing 'Global (Jenkins, nodes, items, all child items, etc)', 'Username' with a text field containing 'VincentRoche', 'Password' with a text field containing dots, 'ID' with an empty text field, and 'Description' with an empty text field. At the bottom, there are 'Add' and 'Cancel' buttons.

**Jenkins Credentials Provider: Jenkins**

**Add Credentials**

Domain: Global credentials (unrestricted)

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: VincentRoche

Password: .....

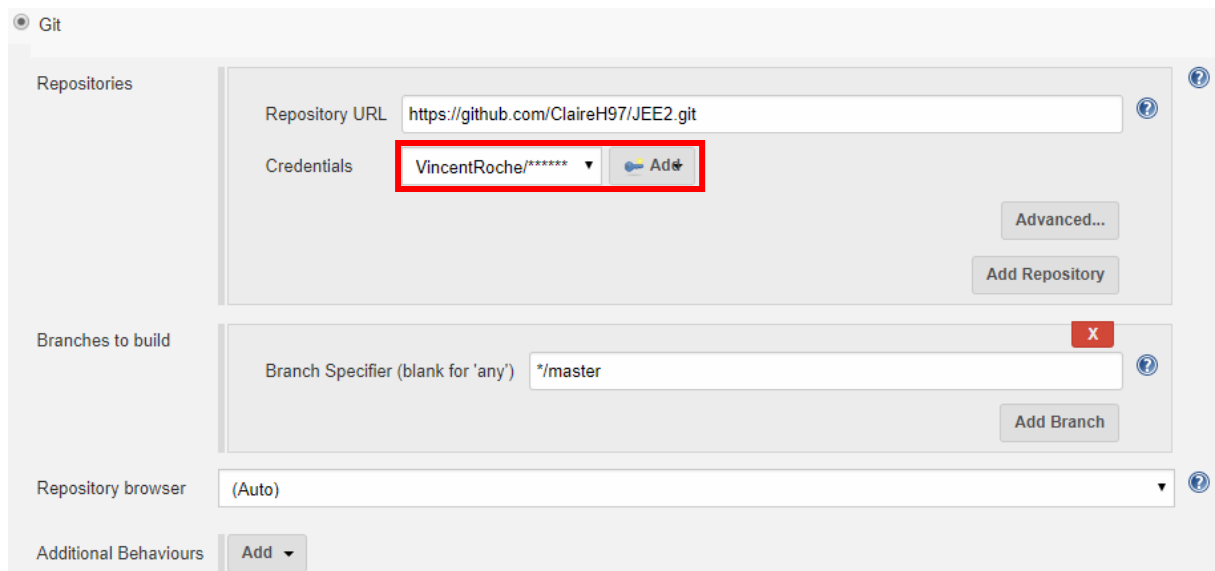
ID:

Description:

**Add** **Cancel**

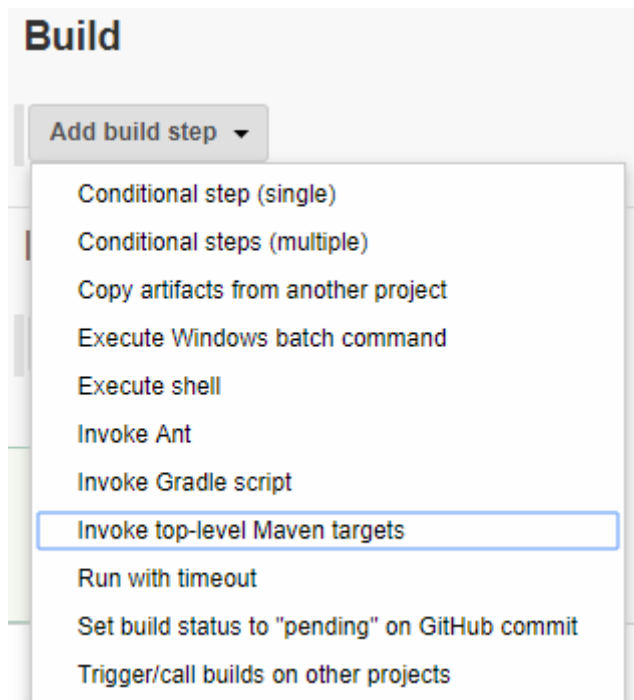


Select the credentials you just created in the dropdown menu and no error should be displayed:



The screenshot shows the 'Git' configuration page in Jenkins. The 'Repositories' section contains a 'Repository URL' field with the value 'https://github.com/ClaireH97/JEE2.git' and a 'Credentials' dropdown menu with 'VincentRoche/\*\*\*\*\*\*' selected. The 'Advanced...' button is visible. The 'Branches to build' section contains a 'Branch Specifier (blank for 'any')' field with the value '\*/master'. The 'Repository browser' is set to '(Auto)'. The 'Additional Behaviours' section has an 'Add' button.

In the “Build” section, click “Add build step” and select “Invoke top-level Maven targets”:



The screenshot shows the 'Build' section in Jenkins. The 'Add build step' dropdown menu is open, showing a list of build steps. 'Invoke top-level Maven targets' is highlighted.

Click “Advanced...” to view all the settings, and give them these values:

Goals: *properties:read-project-properties glassfish:redploy -e*

POM: *MavenPROJECT\pom.xml*

Properties: *glassfish.properties.file.argument=\${environment}*

Invoke top-level Maven targets

Goals: properties:read-project-properties glassfish:redeploy -e

POM: MavenPROJECT\pom.xml

Properties: glassfish.properties.file.argument=\${environment}

JVM Options:

Inject build variables: ☐

Use private Maven repository: ☐

Settings file: Use default maven settings

Global Settings file: Use default maven global settings

The Jenkins job is now configured! Save and you can now build and deploy the project by clicking “Build with parameters”.

Select the server to deploy the project to (it must be running in NetBeans) and wait for the build to end.

## Project Employees

This build requires parameters:

environment

Which GlassFish domain to deploy the project to.

Build

The project has been deployed by Jenkins on the development server!

Visit <http://localhost:5050/> as done before to test the project.

For the testing server, the URL is <http://localhost:6060/>, and for the production server it is <http://localhost:7070/>.