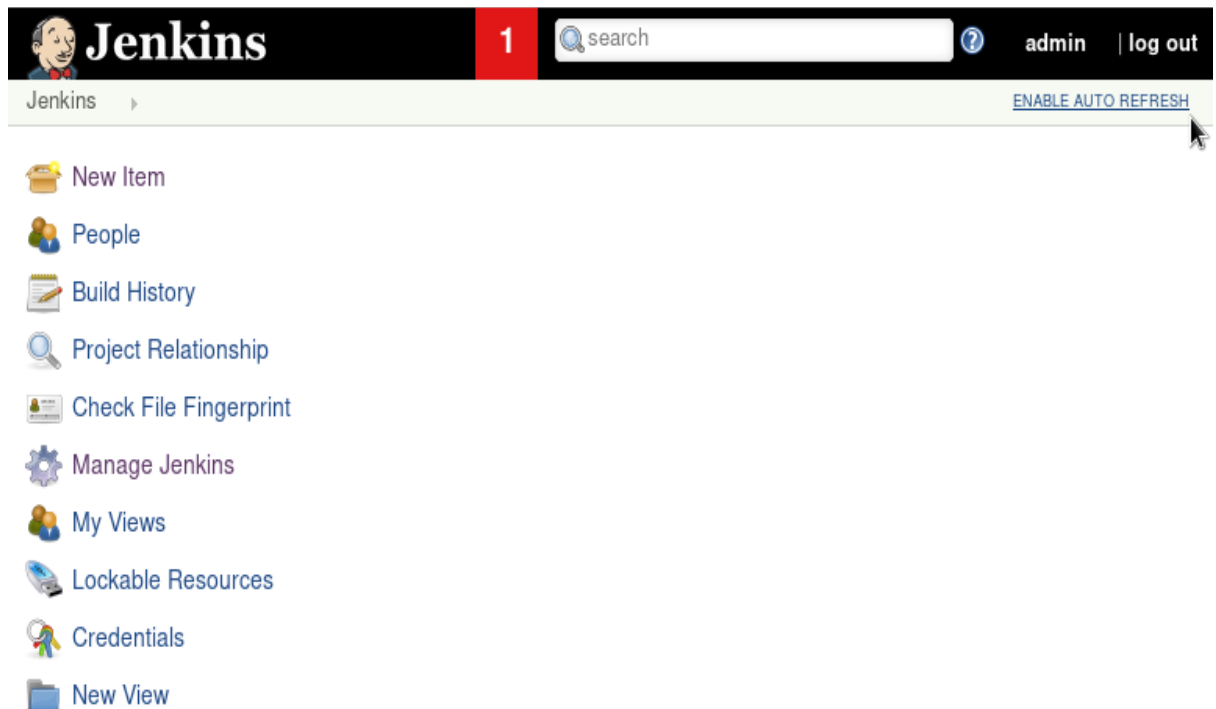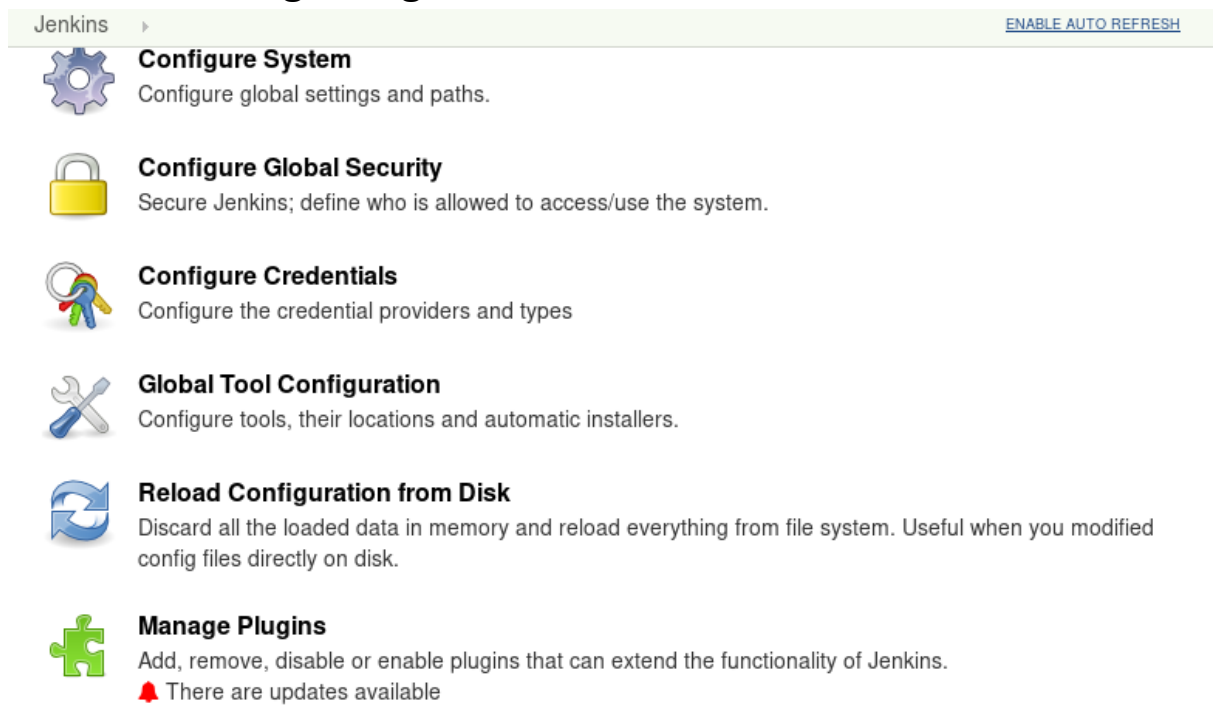# Maven-Jenkins-Github_Integration

1. Login to your Jenkins.

2. Go inside Manage Jenkins.

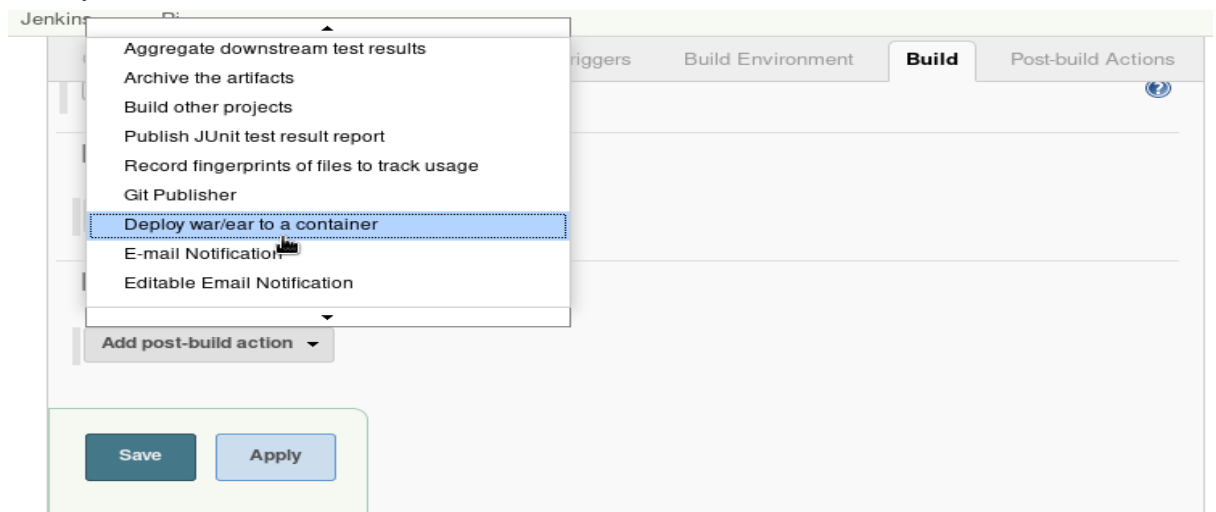3. Click on Manage Plugins.

4. Download Deploy war to container plugin.

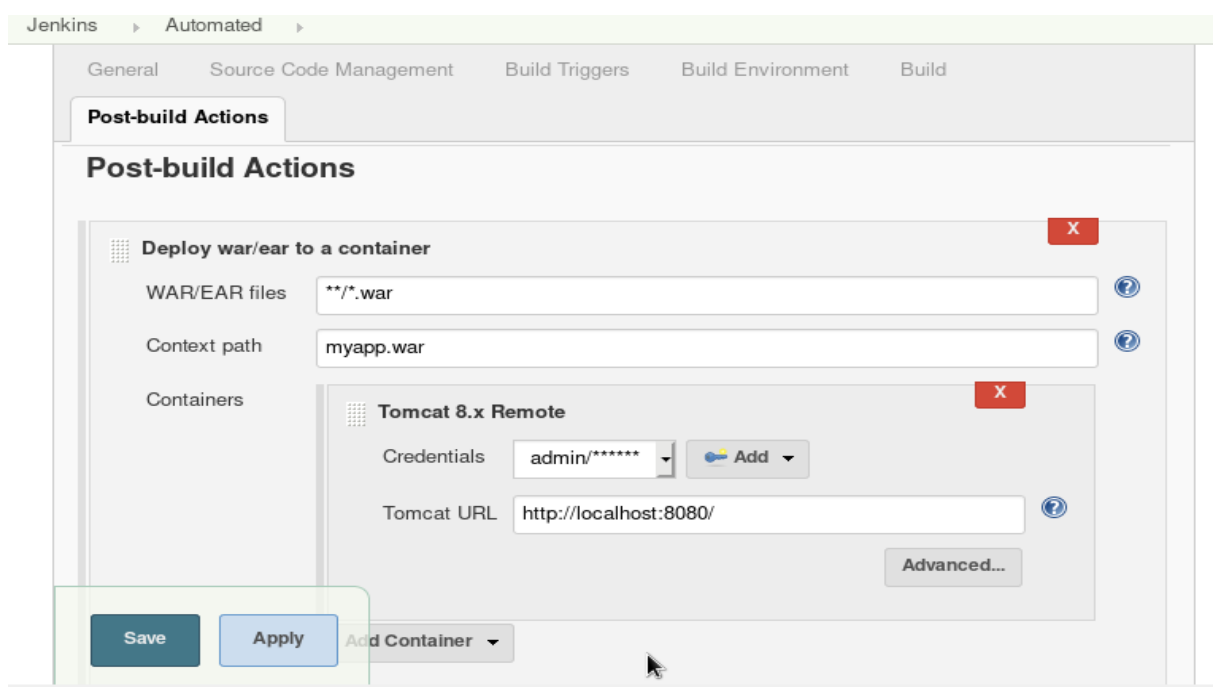|			
☑	**Command Agent Launcher Plugin** Allows agents to be launched using a specified command.	1.4	Uninstall
☑	**Credentials Binding Plugin** Allows credentials to be bound to environment variables for use from miscellaneous build steps.	1.20	Uninstall
☑	**Credentials Plugin** This plugin allows you to store credentials in Jenkins.	2.3.1	Uninstall
☑	**Deploy to container Plugin** This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment	1.15	Uninstall
☑	**Display URL API** Provides the DisplayURLProvider extension point to provide alternate URLs for use in notifications	2.3.2	Uninstall
☑	**Docker Commons Plugin** Provides the common shared functionality for various Docker-related plugins.	1.16	Uninstall
☑	**Docker Pipeline** Build and use Docker containers from pipelines.	1.21	Uninstall
	Durable Task Plugin		

5. Now create a new job and select Freestyle project.

Enter an item name

|
» *Required field*

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

OK

6. Add the action that you want in your project. In our case none. And go to post build action. And select Deploy war/ear to container.

7. Add the name and type of your file in our case anything that contain war. Now enter your credential and tomcat url.

8. Click on save and run the build.