

# Jenkins

## Configuração

Pré-requisito: O docker deve já estar pré instalado antes da configuração do Jenkins.

1- Acessar a pagina <https://jenkins.io/doc/book/installing/>

2- Copiar comando do seu SO, executar no terminal do Docker:

Windows

```
$ docker run ^  
-u root ^  
--rm ^  
-d ^  
-p 8080:8080 ^  
-p 50000:50000 ^  
-v jenkins-data:/var/jenkins_home ^  
-v /var/run/docker.sock:/var/run/docker.sock ^  
jenkins/jenkins
```

O comando acima irá baixar e executar a imagem, ele cria o container do Jenkins, após a conclusão da etapa anterior o jenkins deve estar disponível na url

<http://192.168.99.100:8080/>.

Ainda é necessário a senha de Admin que está dentro do docker.

O container do Jenkins deve estar com um nome aleatório, vamos renomear,

Listar todos os containers:

```
$ docker ps -a
```

3- Renomear o container, pegar o ID do container com o comando:

```
$ docker rename 123455543 jenkins
```

4- Acessar o jenkins para pegar a senha de admin, executar o shell dentro do container:

```
$ docker exec -i -t jenkins sh
```

5- Pegar o valor do arquivo "initialAdminPassword":

```
$ cat /var/jenkins_home/secrets/initialAdminPassword
```

6- Logar no Jenkins com a chave obtida do comando anterior e customizar jenkins:

Selecionar opção "Instalar plug-ins recomendados".

7- Criar o usuário

usuário: administrator

senha: administrator

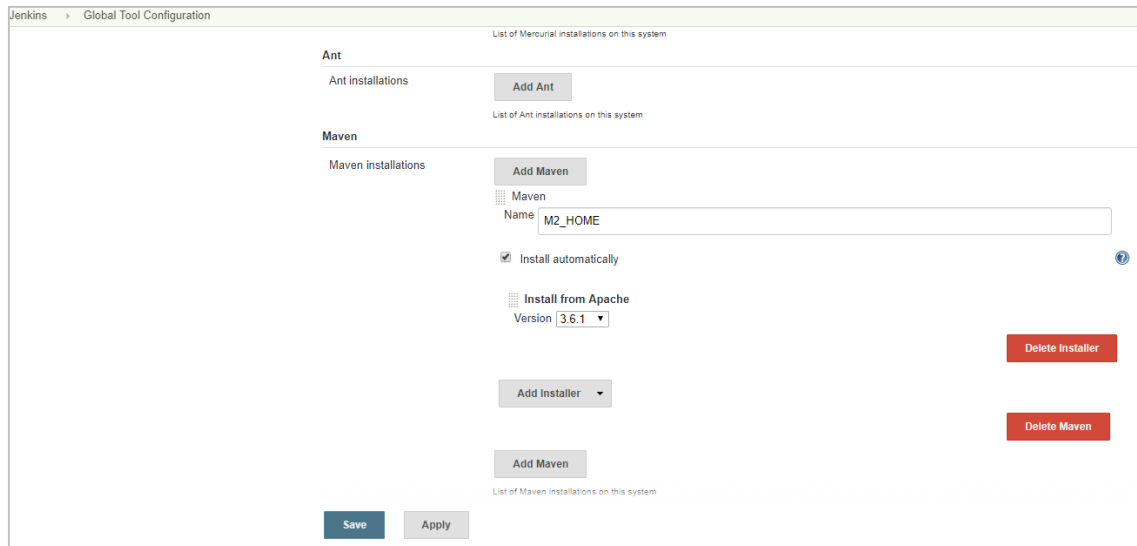
Nome: administrator

email: [email@email.com](mailto:email@email.com)

Setup Jenkins:

*Jenkins>> Manage Jenkins>>Global Tool Configuration*

Add Maven: Nome "*M2\_HOME*", e marcar check "*Instalar Automaticamente*";

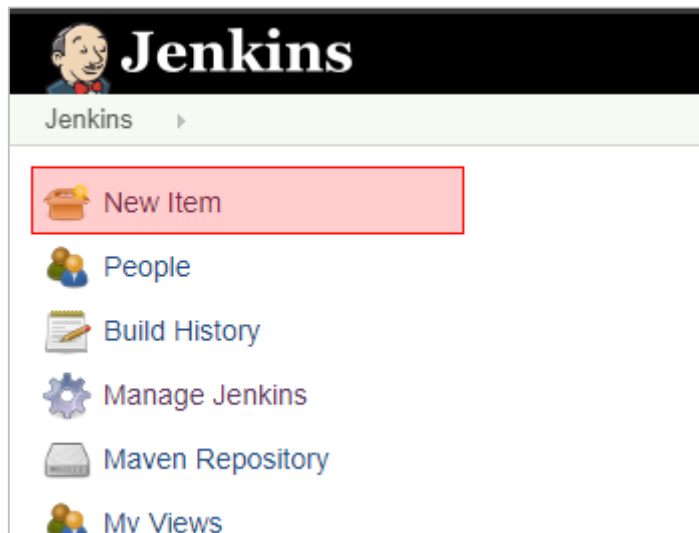


The screenshot shows the Jenkins 'Global Tool Configuration' page for Maven. The page has a header 'Jenkins > Global Tool Configuration'. Below it, there's a section for 'Ant' with 'Ant Installations' and an 'Add Ant' button. The main section is for 'Maven' with 'Maven Installations'. It includes a table with one entry: 'Maven' with 'Name' 'M2\_HOME'. Below the table, there's a checkbox 'Install automatically' which is checked, and a dropdown 'Install from Apache' with 'Version' '3.6.1'. There are buttons 'Delete Installer' and 'Delete Maven' on the right. At the bottom, there are 'Save' and 'Apply' buttons.

Criando o job que irá iniciar os testes:

Criar um novo “job”

Clicar em “New Item”



Colocar Nome do job, selecionar “*Freestyle Project*”, e confirmar:


**Jenkins** search


Jenkins > All >


### Enter an item name


TestarLogin


» 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.


**External Job**  
 This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.


**Multi-configuration project**  
 Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

Preencha os campos necessários:

Descrição, projeto do git hub, comando maven, gatilho de agendamento.

**Jenkins** search adm

Jenkins > TestarLogin >

**General** Source Code Management Build Triggers Build Environment Build Post-build Actions

Description: Esse job executa os teste no login.

[Plain text] [Preview](#)

JIRA site

☐ Discard old builds

☒ GitHub project

Project url: <https://github.com/gerrysousa/desafio-automacao-selenium-webdriver.git>

[Advanced...](#)

☐ Disable Automated Maven Repository Cleanup

☐ This build requires lockable resources

☐ This project is recommended

Jenkins > TestarLogin >

**Source Code Management** General Build Triggers Build Environment Build Post-build Actions [Advanced...](#)

### Source Code Management

☐ None

☒ Git

Repositories

Repository URL: <https://github.com/gerrysousa/desafio-automacao-selenium-webdriver.git>

Credentials: [- none -](#) [Add](#)

[Advanced...](#)

[Add Repository](#)

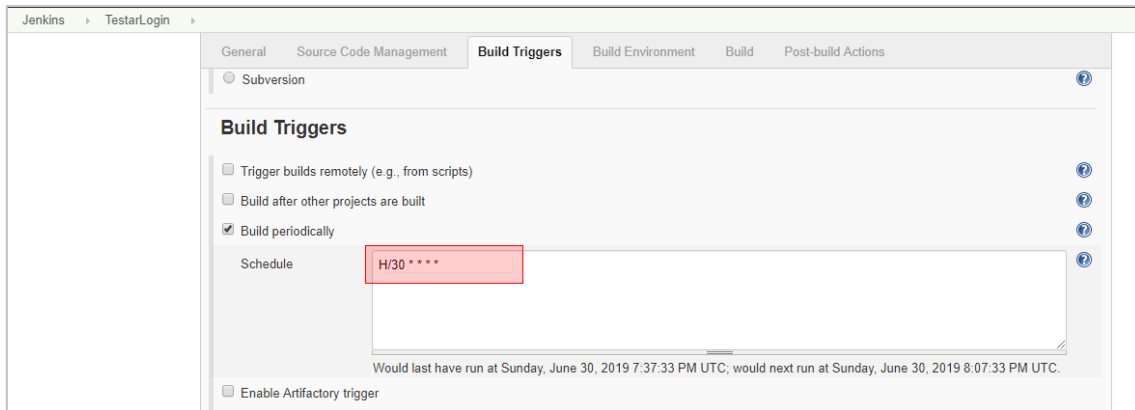
Branches to build

Branch Specifier (blank for 'any'): \*/master

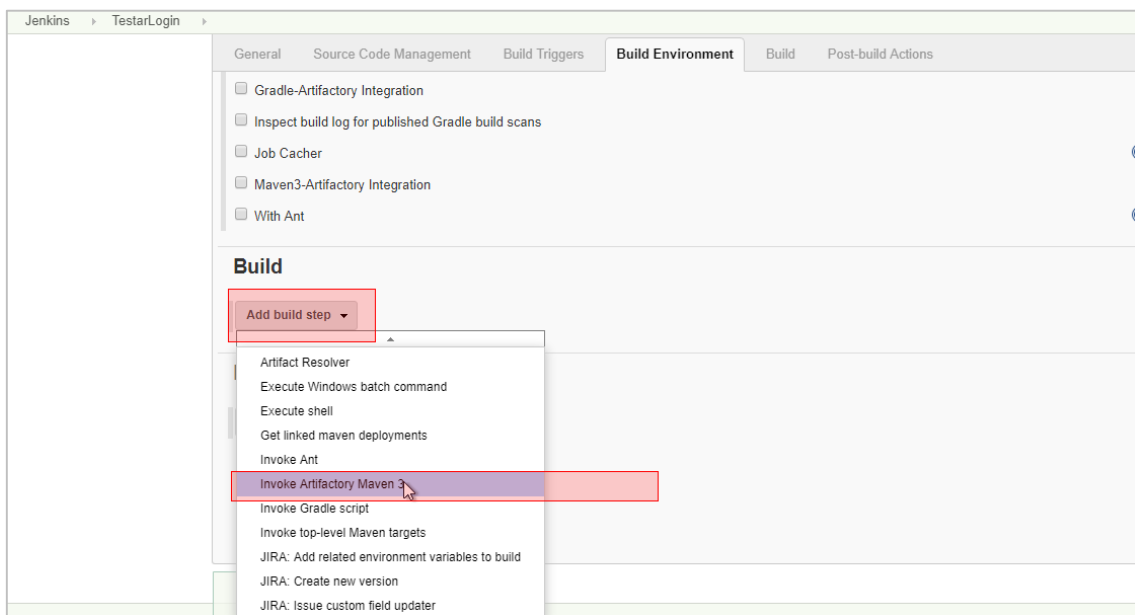
[Add Branch](#)

Repository browser: (Auto)

Agendar pra rodar de 30 em 30 minutos "H/30 \* \* \* \* "



Add comando Maven no Build



Selecionar o Maven adicionado anteriormente, colocar o arquivo "pom.xml". e colocar o parâmetro do comando que será executado ao iniciar o job `test - DsuiteXmlFile=src/main/java/tests/xml/Test_LoginLocal.xml`.

O comando executado é o seguinte:

`"mvn test -DsuiteXmlFile=src/main/java/tests/xml/Test_LoginLocal.xml"`

The screenshot shows the Jenkins configuration page for a job named 'TestarLogin'. The 'Build Environment' tab is selected. Under the 'Build' section, the 'Invoke Artifactory Maven 3' step is configured with the following values:

- Maven Version: M2\_HOME
- Root POM: pom.xml
- Goals and options: test -DsuiteXmlFile=src/main/java/tests/xml/Test\_LoginLocal.xml

At the bottom, there are 'Save' and 'Apply' buttons. The 'Post-build Actions' tab is also visible below.

Agora o job está pronto para ser executado:

The screenshot shows the Jenkins job overview page for 'TestarLogin'. The left sidebar contains navigation links: Back to Dashboard, Status, Changes, Workspace, Build Now (highlighted with a red box), Delete Project, Configure, Dependency Graph, Favorite, GitHub, Open Blue Ocean, and Rename. The main content area displays the project name 'Project TestarLogin' and the description 'Esse job executa os teste no login.' Below this are links for 'Workspace' and 'Recent Changes'. A 'Permalinks' section lists various build links, including 'Last build (#8), 2 min 56 sec ago'. At the bottom, the 'Build History' section shows a table of recent builds:

Build Number	Timestamp
#8	Jun 30, 2019 9:11 PM
#7	Jun 30, 2019 8:29 PM

Podemos ver os logs da execução clicando no build executado:

```
Jenkins > TestarLoginGrid > #1

Starting firefox on grid
Jun 30, 2019 9:24:47 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Starting firefox on grid
Jun 30, 2019 9:25:07 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Starting firefox on grid
Jun 30, 2019 9:25:27 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Starting firefox on grid
Jun 30, 2019 9:25:47 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Starting firefox on grid
Jun 30, 2019 9:26:08 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
[INFO] Tests run: 10, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 177.749 s - in TestSuite
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 10, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 03:11 min
[INFO] Finished at: 2019-06-30T21:26:24Z
[INFO] -----
Build step 'Invoke Artifactory Maven 3' changed build result to SUCCESS
Finished: SUCCESS
```

Foi criado um job que executa o projeto local, e outro que executa do projeto no hub selenium grid.

search

administrator | log out

Jenkins >

ENABLE AUTO REFRESH

add description

New Item

People

Build History

Manage Jenkins

Maven Repository

My Views

Leader board

Open Blue Ocean

Lockable Resources

Credentials

Dependency Graph

New View

All

S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
		Test-jenkins-1	10 days - #22	9 days 22 hr - #26	0.39 sec	
		TestarLogin	24 min - #8	1 hr 6 min - #7	1 min 34 sec	
		TestarLoginGrid	12 min - #1	N/A	3 min 24 sec	
		TestComGit	5 days 17 hr - #50	1 hr 15 min - #51	46 sec	

Icon: S M L

Legend RSS for all RSS for failures RSS for just latest builds