

Primeiramente para utilizarmos o jenkins precisamos instalar ele através de um sistema linux, no meu caso como utilizo Windows, foi usado o WSL 2.



Com o Ubuntu aberto vamos agora partir para a instalação do jenkins na nossa máquina com os seguintes comandos.

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

Then add a Jenkins apt repository entry:

```
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
```

Update your local package index, then finally install Jenkins:

```
sudo apt-get update
sudo apt-get install fontconfig openjdk-17-jre
sudo apt-get install jenkins
```

Com o jenkins instalado vamos agora acessar o endereço padrão do jenkins <http://localhost:8080/>.

Assim abrindo a seguinte tela, para que prosseguimos com o login.

Começando

Abrir o Jenkins

Para garantir que o Jenkins está configurado de forma segura pelo administrador, uma senha foi escrita no arquivo de registro ([não sabe onde encontrar?](#)) e neste arquivo no servidor:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Por favor copie a senha de qualquer uma das localizações e cole abaixo.

Senha do administrador

Pegamos o seguinte diretório [/var/lib/jenkins/secrets/initialAdminPassword](#) e então utilizamos o comando cat no ubuntu para que possamos pegar a chave necessária para abrir o Jenkins

```
hask@Hask: ~  
hask@Hask:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
16eabd90995a41139b4847aeea146999  
hask@Hask:~$
```

Após inserir a senha, vai abrir uma tela para que possamos baixar os plugins necessários. Clicamos para baixar os plugins sugeridos.

Iniciando

Iniciando

✓ Folders	✓ OWASP Markup Formatter	⌚ Build Timeout	⌚ Credentials Binding	** Ionicons API
⌚ Timestamper	⌚ Workspace Cleanup	⌚ Ant	⌚ Gradle	Folders
⌚ Pipeline	⌚ GitHub Branch Source	⌚ Pipeline: GitHub Groovy Libraries	⌚ Pipeline: Stage View	OWASP Markup Formatter
⌚ Git	⌚ SSH Build Agents	⌚ Matrix Authorization Strategy	⌚ PAM Authentication	** Struts
⌚ LDAP	⌚ Email Extension	⌚ Mailer		** bouncycastle API

Depois de baixados os plugins necessários, vamos agora criar um usuário administrador.

Iniciando

Criar o primeiro usuário administrativo

Nome de usuário

Senha

Confirmar a senha

Nome completo

Endereço de e-mail

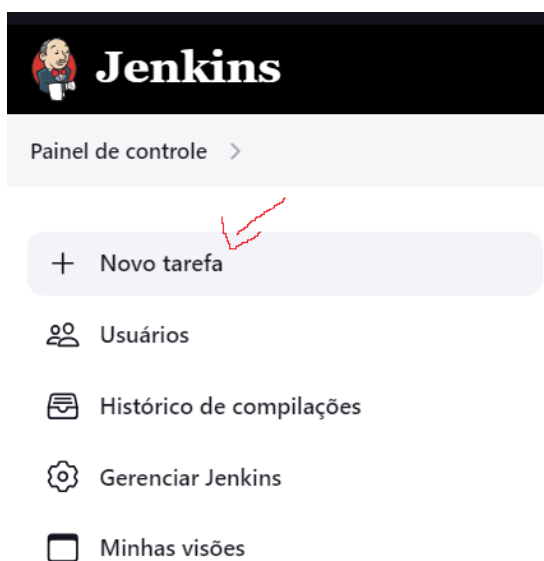
Jenkins 2.426.1

[Pular e continuar como administrador](#)

[Salvar e continuar](#)

Depois de realizado todos esses passos agora estamos prontos para começar a utilizar a ferramenta.

Vamos criar uma nova tarefa.




Inserir um nome para a tarefa, e selecionar Pipeline.


Entre com um nome de item

Teste


» Campo requerido

**Construir um projeto de software de estilo livre.**


Esta é a central de funcionalidades do Jenkins. Ela construirá seu projeto e você pode combinar qualquer SCM com qualquer sistema de construção, e ele até mesmo pode ser usado para outras tarefas diferentes de construções de *software*.

**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.

**Construir projeto de múltiplas configurações**


Apropriado para projetos que necessitam de grande número de diferentes configurações, como teste em múltiplos ambientes, builds para plataformas específicas, etc.

**Folder**

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**

Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**

Creates a set of multibranch project subfolders by scanning for repositories.

Podemos colocar uma descrição para a nossa tarefa, descendo vamos ter um campo de script, onde vamos colocar a nossa pipeline.

Pipeline

Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Hello') {
6       steps {
7         echo 'Teste Disciplina DevOps'
8       }
9     }
10  }
11 }
12
```

Hello World

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

Depois de salvar, podemos agora clicar para construir a nossa tarefa, assim executando-a e a visualizando.

Painel de controle > Teste >

Status

</> Changes

▶ Construir agora

⚙ Configurar

🗑 Excluir Pipeline

🔍 Full Stage View

✎ Renomear

❓ Pipeline Syntax

Teste

Teste para disciplina DevOps

Stage View

Average stage times:
(Average full run time: ~285ms)

#1	nov. 28 22:55	No Changes
----	------------------	---------------

Hello

58ms

58ms

Links permanentes

🌞 Histórico de construções Tendência ▾

🔍 Filtro de construções... /

#1
| 28 de nov. de 2023 22:55

📡 Atom feed para todos 📡 Atom feed por falhas

Conseguimos também ver a saída do console da nossa tarefa.

✓ Saída do console

🔍 </> Changes or falhas

📄 Console Output

📄 Edit Build Information

🗑 Apagar a construção {0}

🔄 Restart from Stage

🔄 Replay

☰ Pipeline Steps

📁 Workspaces

```
Started by user Lucas Lourenço
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Teste
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Teste Disciplina DevOps
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```