

Introdução ao Jenkins

DigitalHouse>



**Certified
Developer**

The Ultimate Tech Degree

Índice

- 1. O que é Jenkins?**
- 2. Instalação**
- 3. Configuração**
- 4. Distributiva**
- 5. Plugins**
- 6. Jenkinsfile**

1 | O que é Jenkins?

Jenkins

Jenkins é um servidor projetado para integração contínua (CI). É gratuito e de código aberto. E se tornou o software mais usado para essa tarefa!

O que podemos fazer em Jenkins? Permite-nos organizar uma cadeia de ações que ajuda a concretizar o processo de integração contínua (e muito mais!) de forma automatizada.

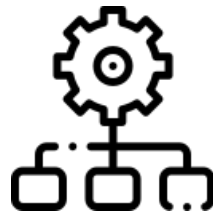


Algumas de suas características



Automação

É o software de automação mais utilizado para integração contínua.



Tarefas

Jenkins pode orquestrar qualquer tipo de processo e executar tarefas manuais, periódicas ou automatizadas.



Comunidade

A comunidade desenvolveu mais de 15.000 plug-ins para agilizar as equipes de desenvolvimento.



Fácil de usar!

Seu uso é simples e você pode aumentar sua capacidade de computação adicionando novos agentes ou servidores.

2 | Instalação

Como instalamos o Jenkins?

A instalação varia de acordo com o sistema operacional. Ambos são válidos para servidores e nossos computadores pessoais. Os mais comuns são:

- Execute o Jenkins como um contêiner do Docker ou use a imagem oficial para executar no Kubernetes.
- Baixe um MSI para instalar no Windows 10 ou Windows Server.
- Pode ser usado em distribuições baseadas em Red Hat como RHEL, CentOS ou Fedora via RPM.
- Implante-o no Ubuntu / Debian por meio de um pacote ".deb".

Por exemplo, no docker é implementado assim:

```
profesor@digitalhouse: ~  
profesor@digitalhouse:~$ docker run -d -p 8081:8080 -p 50000:50000 jenkins/jenkins:lts-jdk11  
fe8224ba9b3397f50d908f698c5ef7f264566d75224ca9fe75aeb7f9ae236c8c  
profesor@digitalhouse:~$ docker images  
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE  
jenkins/jenkins     lts-jdk11      ea470c80c15d   2 weeks ago    680MB  
profesor@digitalhouse:~$ docker ps  
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS  
fe8224ba9b33   jenkins/jenkins:lts-jdk11          "/sbin/tini -- /usr/..." 12 seconds ago Up 11 seconds 0.0.0.0:50000->50000/tcp, 0.0.0.0:8081->8080/tcp  
goofy_lewin  
profesor@digitalhouse:~$
```


3 | Configuração

Configuração Jenkins

Na seção de configuração, podemos personalizar várias opções:

SMTP e outros canais

Podemos configurar o Jenkins para se comunicar conosco por e-mail, enviando informações de nosso SMTP ou WebHooks para usar canais como o Slack.

variáveis ambientais

Em nossa instância do Jenkins, podemos salvar informações confidenciais (por exemplo, os dados de conexão com nossos servidores como um par de chaves para ssh).

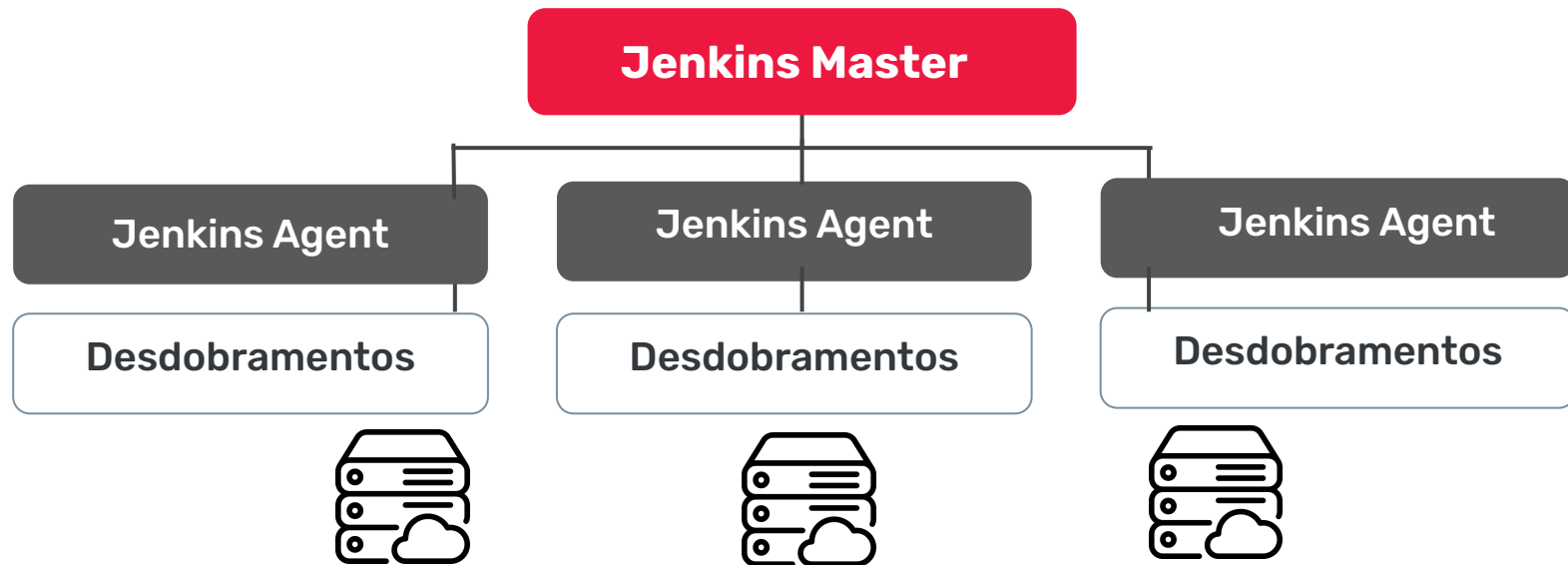
Segurança em primeiro lugar!

Podemos configurar as funções e permissões de nosso Jenkins para limitar o acesso do usuário. É importante que cada usuário tenha as permissões mínimas para realizar suas tarefas.

4 | Distributiva

Múltiplos nós

Jenkins pode distribuir o trabalho em várias máquinas, ajudando você a construir, testar e implantar em várias plataformas com mais rapidez.



5 | Plugins

Plugins

Com centenas de complementos no Centro de Atualização, o Jenkins se integra a virtualmente todas as ferramentas na integração contínua e na cadeia de entrega. Alguns exemplos são...



Docker



**Slack
Notifications**



Pipelines



Azure

6 | Jenkinsfile

Jenkinsfile

Ao criar um arquivo chamado Jenkinsfile, podemos definir nosso pipeline por meio de código escrito em Groovy (um derivado de Java).

Isso nos dá alguns benefícios:

- Ampliar o uso de plug-ins.
- Personalizar alguns usos que não são encontrados em plug-ins.
- Que nosso pipeline tenha a mesma lógica de um aplicativo por meio de estruturas de controle.

{ }

```
pipeline {  
    // Podemos usar informações criptografadas como chaves por meio de código  
    environment {  
        AWS_ACCESS_KEY_ID      = credentials('jenkins-aws-secret-key-id')  
        AWS_SECRET_ACCESS_KEY = credentials('jenkins-aws-secret-access-key')  
    }  
}
```

Jenkinsfile: Dividimos em etapas

Os pipelines são divididos em diferentes etapas, encadeadas para obter o resultado esperado no final do pipeline. Os chamamos de **Stages**.

```
pipeline {  
  stages {  
    stage('Stage 1') {  
    }  
    stage('Stage 2') {  
    }  
  }  
}
```

```
pipeline {  
  stages {  
    stage('Stage 1') {  
      steps {  
        sh("kubectl --kubeconfig  
$MY_KUBECONFIG get pods")  
      }  
    }  
  }  
}
```

DigitalHouse>