

Alessandra Avelino



Jenkins

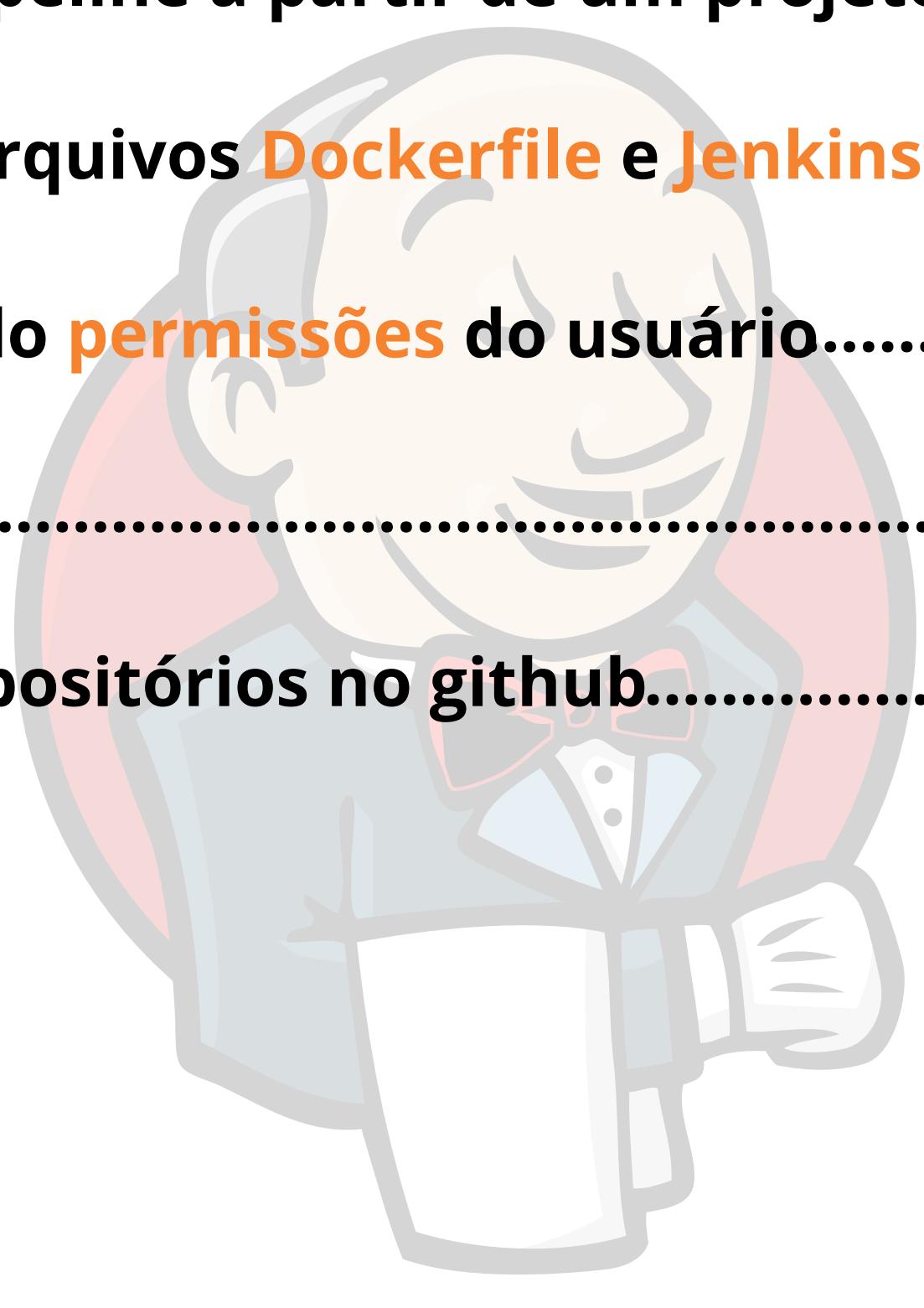
APRENDA A CONFIGURAR UMA PIPELINE DO ZERO

Através deste tutorial, criaremos uma pipeline com jenkins desde a criação dos arquivos Docker até o build.

Gerência e Config. de Servidor para Internet

ÍNDICE

- Criando os arquivos **Docker** e **Dockerfile** 2
- Configurando o **Jenkins** 4
- Criando um **usuário**..... 5
- Criando a pipeline a partir de um projeto no **github**6
- Criando os arquivos **Dockerfile** e **Jenkinsfile**..... 9
- Configurando **permissões** do usuário..... 11
- Deploy..... 12
- Link dos repositórios no **github**.....12



Jenkins

CRIANDO OS ARQUIVOS DOCKER E DOCKERFILE

1

Criar arquivo Dockerfile

Através desse Dockerfile, iremos configurar a imagem do Jenkins, especificando alguns comandos, desde a imagem base, a versão do jdk já que o Jenkins é baseado no Java. Ainda no arquivo, é criado um grupo e adicionado o usuário do jenkins, ao mesmo, para que seja viável a execução do jenkins sem precisar de root. Por fim, é utilizando o plugin do blueocean, que iremos entender melhor mais a frente.

```
# DOCKERFILE

FROM jenkins/jenkins:latest-jdk11
USER root
RUN apt-get update && apt-get install -y lsb-release
RUN curl -fsSLo /usr/share/keyrings/docker-archive-keyring.asc \
  https://download.docker.com/linux/debian/gpg
RUN echo "deb [arch=$(dpkg --print-architecture) \
  signed-by=/usr/share/keyrings/docker-archive-keyring.asc] \
  https://download.docker.com/linux/debian \
  $(lsb_release -cs) stable" > /etc/apt/sources.list.d/docker.list
RUN apt-get update && apt-get install -y docker-ce-cli
RUN groupadd docker && usermod -aG docker jenkins
RUN usermod -aG root jenkins
USER jenkins
RUN jenkins-plugin-cli --plugins "blueocean:1.25.8 docker-workflow:521.v1a_a_dd2073b_2e"
```

2

Criar arquivo docker-compose.yml

Neste arquivo, iremos descrever a construção da imagem e as configurações do container para utilizarmos o Jenkins.

Especificamos a versão do Docker Compose e os serviços que estarão presente, no nosso caso, iremos utilizar o jenkins, bem como será o nome do nosso contêiner. O build é as configurações para a construção, no context especificamos o contexto para a construção da imagem. Em ports definimos a porta que o Jenkins estará em execução. Por fim, definimos o volume para montar o contêiner.

Atente-se para o caminho de acordo com o seu sistema operacional, veja abaixo um exemplo de como ficará o caminho do volume.

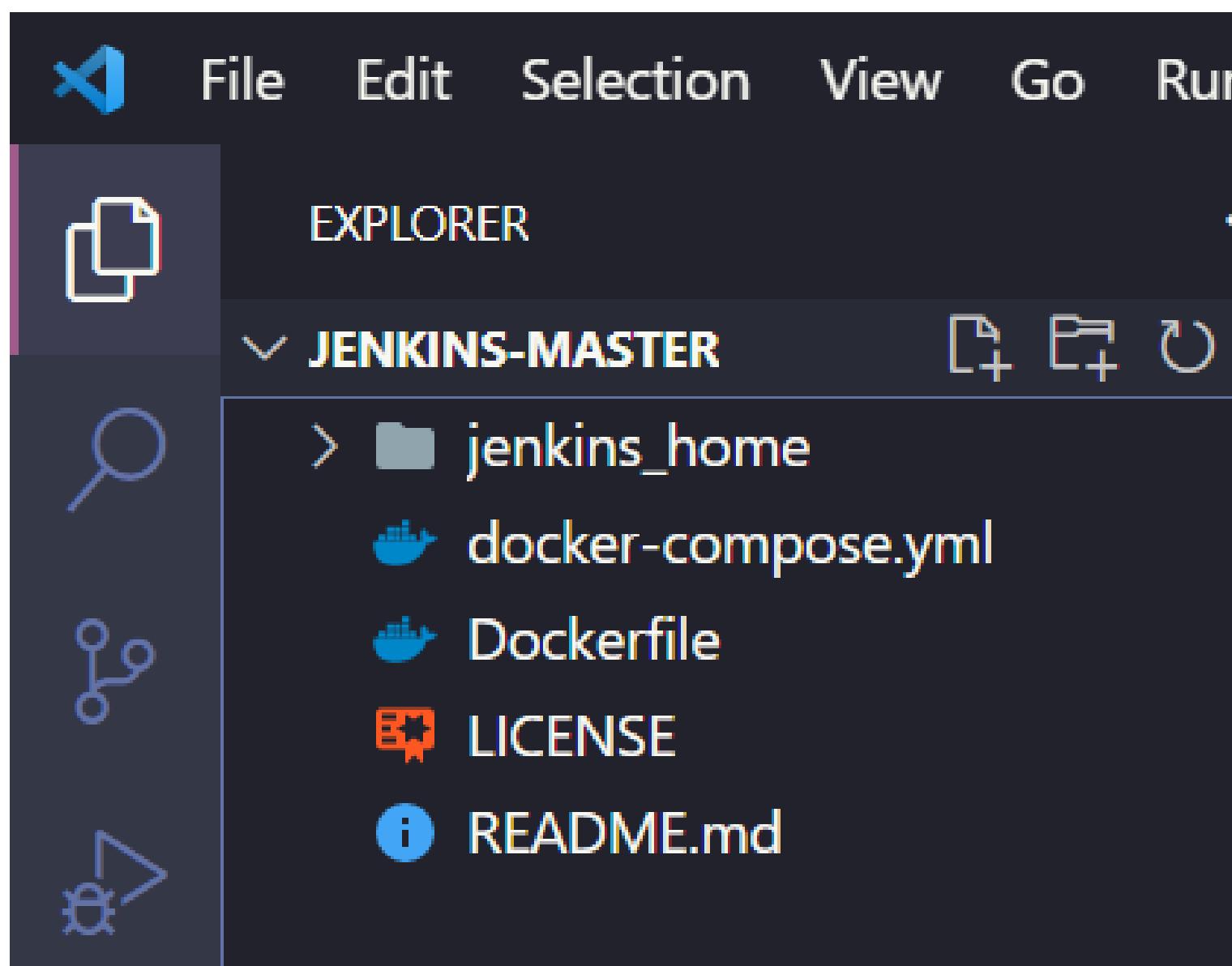
```
# DOCKERCOMPOSE.YML

version: '3'
services:
  jenkins:
    container_name: jenkins
    build:
      context: .
    ports:
      - 8080:8080
    volumes:
      - /home/ifpb/Downloads/jenkins/jenkins_home/:/var/jenkins_home
      - /var/run/docker.sock:/var/run/docker.sock
```

3

Criar pasta `jenkins_home`

Nesta etapa, iremos criar uma pasta com o nome `jenkins_home`, a qual irá conter todos os arquivos necessários referente a configurações de plugins, segurança, trabalho, logs, etc..

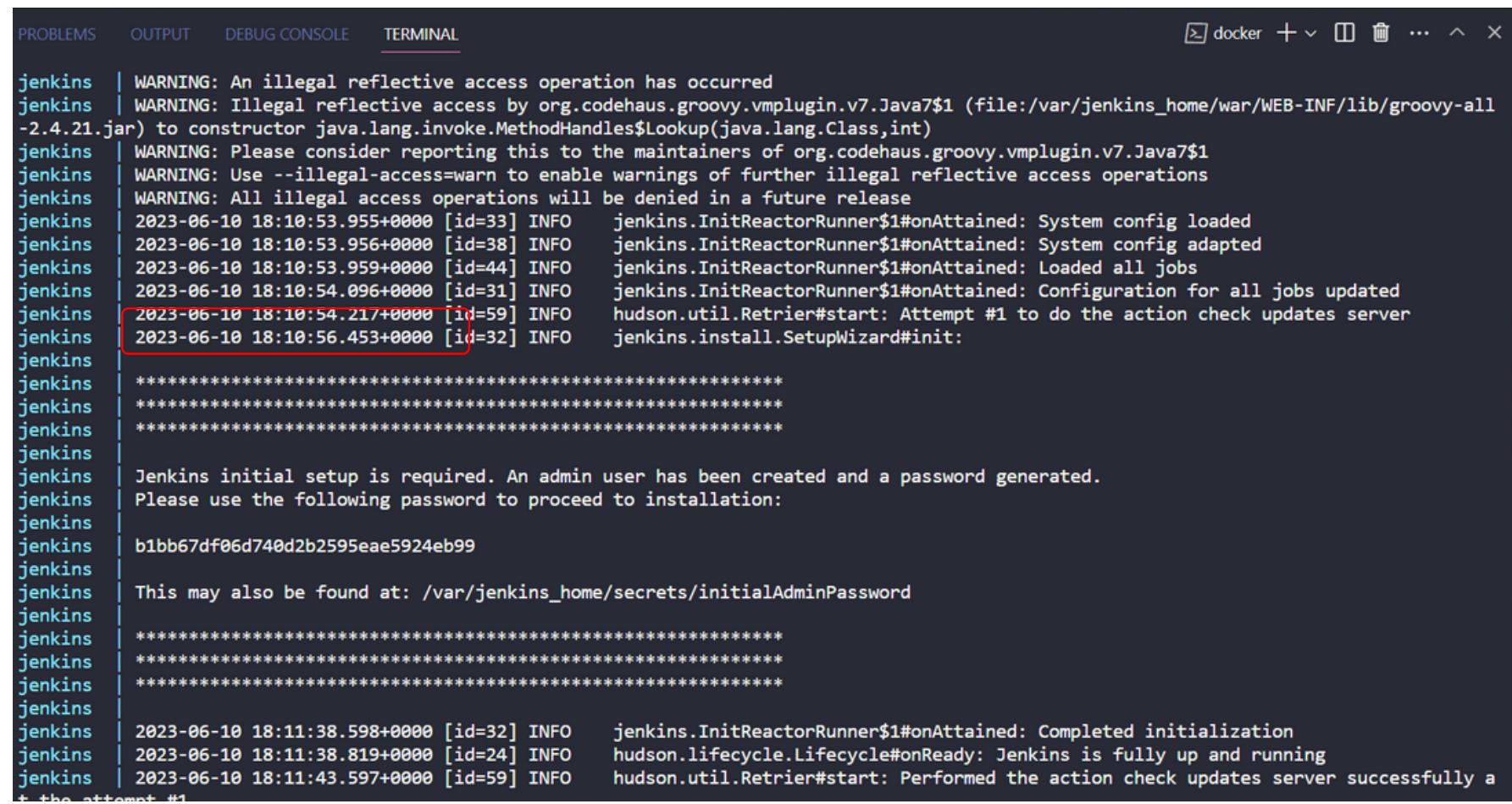


4

Subir a imagem do contêiner

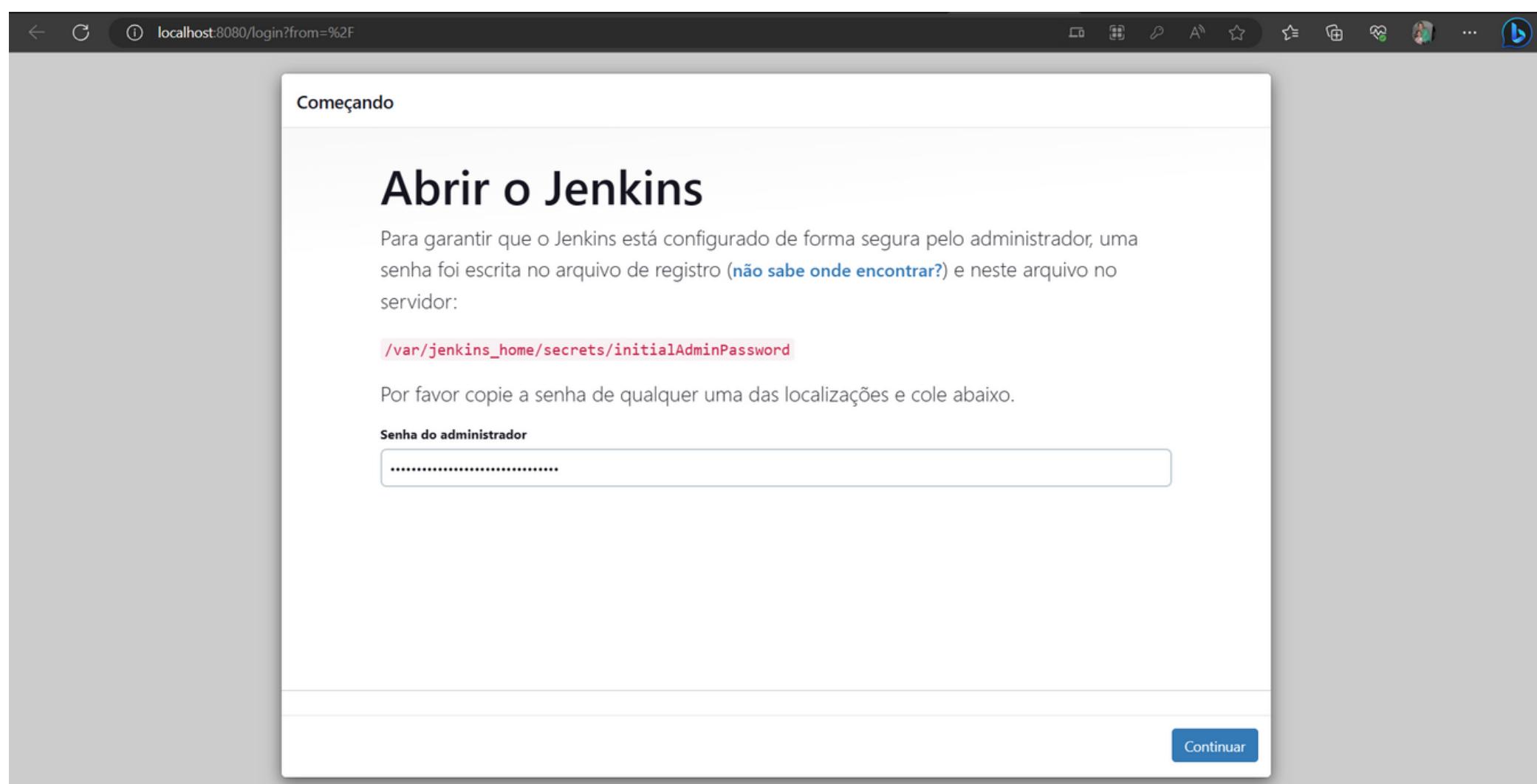
No seu terminal, execute o seguinte comando: `docker compose up`. Caso dê tudo certo, será exibido a senha do administrador no terminal que será necessário posteriormente, então lembre-se de anotar. Em seguida, poderá acessar a interface do Jenkins em <http://localhost:8080/>.

Terminal



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
jenkins | WARNING: An illegal reflective access operation has occurred
jenkins | WARNING: Illegal reflective access by org.codehaus.groovy.vmplugin.v7.Java7$1 (file:/var/jenkins_home/war/WEB-INF/lib/groovy-all-2.4.21.jar) to constructor java.lang.invoke.MethodHandles$Lookup(java.lang.Class,int)
jenkins | WARNING: Please consider reporting this to the maintainers of org.codehaus.groovy.vmplugin.v7.Java7$1
jenkins | WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
jenkins | WARNING: All illegal access operations will be denied in a future release
jenkins | 2023-06-10 18:10:53.955+0000 [id=33] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
jenkins | 2023-06-10 18:10:53.956+0000 [id=38] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
jenkins | 2023-06-10 18:10:53.959+0000 [id=44] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
jenkins | 2023-06-10 18:10:54.096+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
jenkins | 2023-06-10 18:10:54.217+0000 [id=59] INFO hudson.util.Retriger#start: Attempt #1 to do the action check updates server
jenkins | 2023-06-10 18:10:56.453+0000 [id=32] INFO jenkins.install.SetupWizard#init:
jenkins |
jenkins ****
jenkins ****
jenkins ****
jenkins ****
jenkins Jenkins initial setup is required. An admin user has been created and a password generated.
jenkins Please use the following password to proceed to installation:
jenkins b1bb67df06d740d2b2595eae5924eb99
jenkins This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
jenkins ****
jenkins ****
jenkins ****
jenkins 2023-06-10 18:11:38.598+0000 [id=32] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
jenkins 2023-06-10 18:11:38.819+0000 [id=24] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
jenkins 2023-06-10 18:11:43.597+0000 [id=59] INFO hudson.util.Retriger#start: Performed the action check updates server successfully a
+ the attempt #1
```

Interface do Jenkins



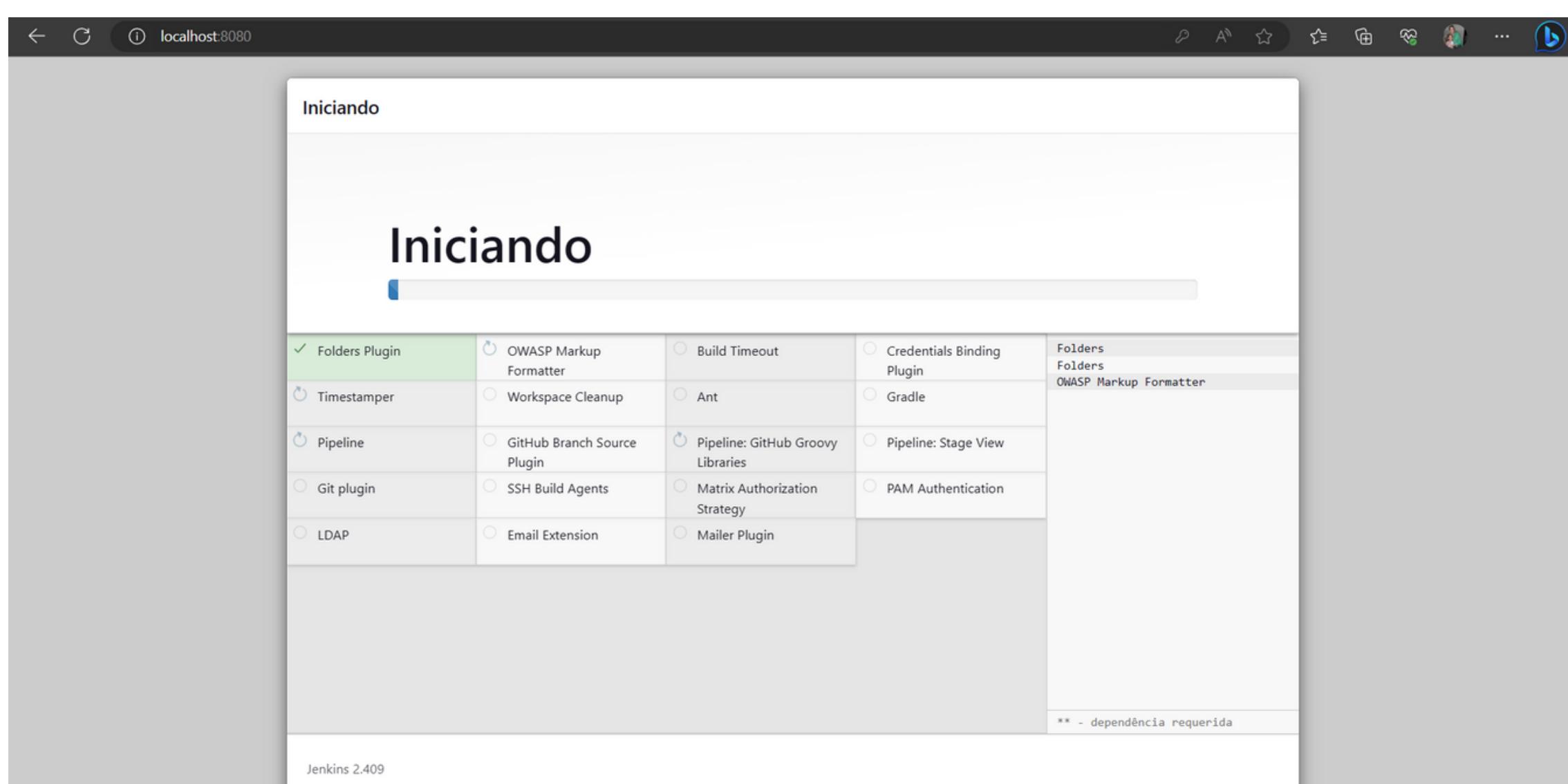
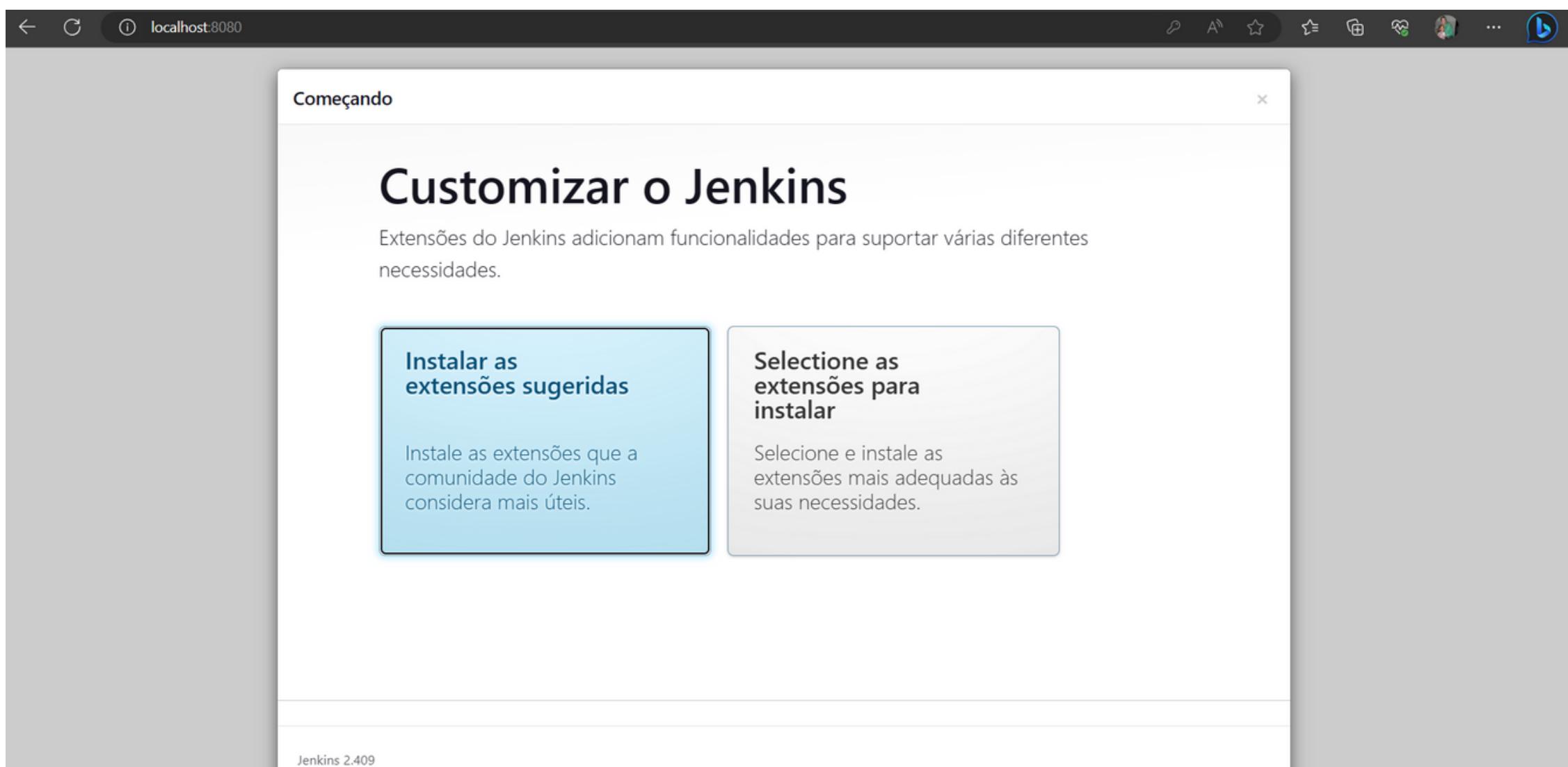
4

Configurando o Jenkins

Cole a senha que foi exibida no seu terminal na página inicial onde ele solicita a **Senha do Administrador** e clique em **continuar**.

Após isso, aparecerá a seguinte tela, a qual podemos clicar em Instalar **Extensões Sugeridas**. Depois disso, ele começará a instalar as extensões.

Customização



5

Criando um usuário

Após finalizar a instalação dos plugins, o Jenkins solicitará que você crie um usuário. Preencha as informações e lembre-se de anota-las, pois você utilizará para fazer login. Após isso, ele solicitará que você configure uma instância, porém, pode manter o padrão e continuar. Na página final, quando estiver tudo certo, basta clicar em [Começar a usar o Jenkins](#)

Iniciando

Criar o primeiro usuário administrativo

Nome de usuário

Senha

Confirmar a senha

Nome completo

Jenkins 2.409 Pular e continuar como administrador Salvar e continuar

Iniciando

Configuração da instância

URL do Jenkins:

A URL do Jenkins é usada para prover a URL raiz para links absolutos para vários recursos do Jenkins. Isto significa que este valor é requerido para a operação apropriada de muitas funcionalidades do Jenkins incluindo notificações por correio eletrônico, atualização de estado de PR e a variável de ambiente BUILD_URL provida pelos passos de construção.

O valor proposto padrão mostrado é **ainda não salvo** sendo gerado da solicitação atual, se possível. A melhor prática é configurar este valor para a URL que se espera que os usuários utilizem. Isto evita confusão quando compartilhando ou vendendo links.

Jenkins 2.409 Agora não Gravar e Concluir

Começar

O Jenkins está pronto!

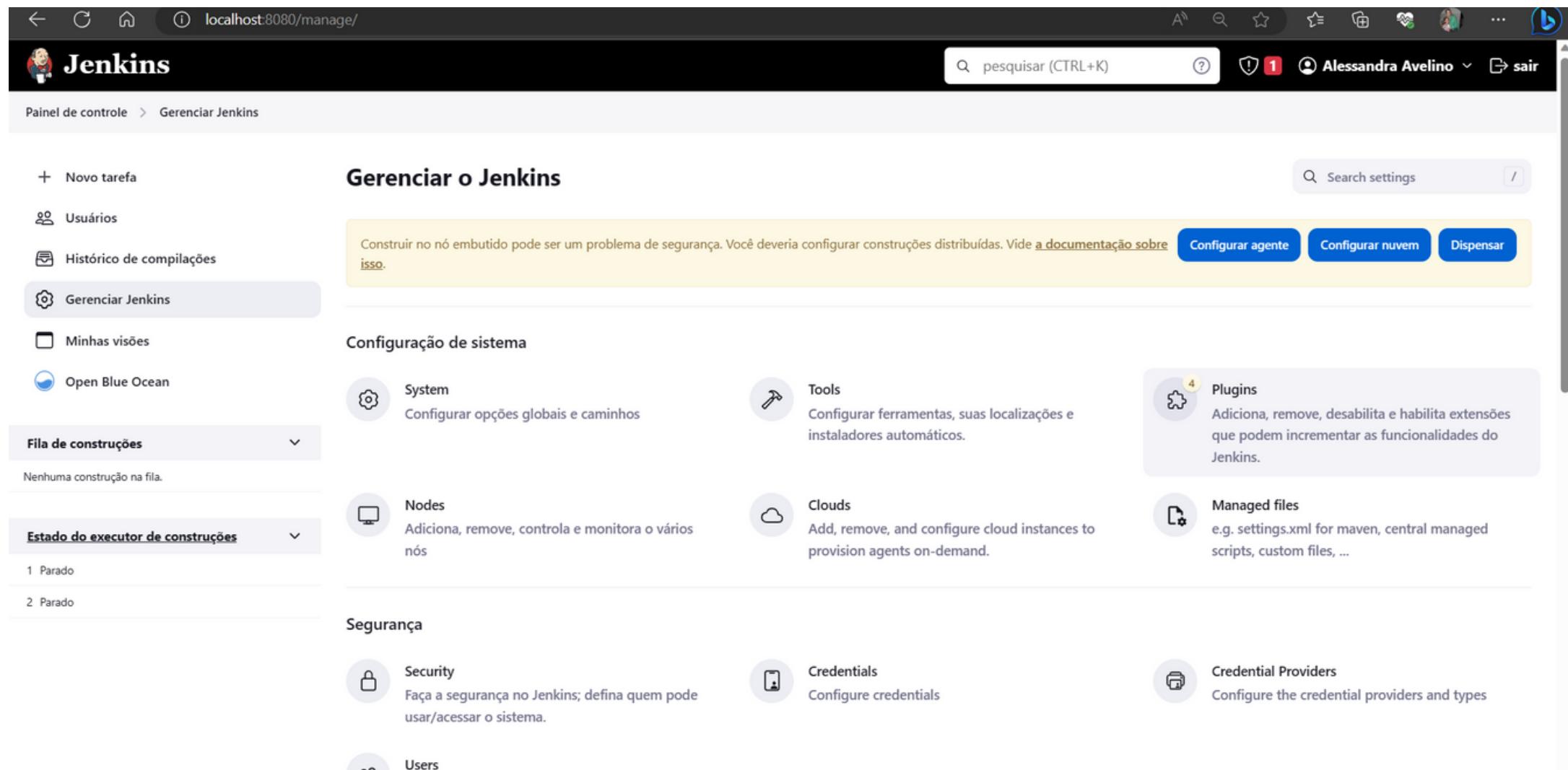
A configuração do seu Jenkins está completa.

Começar a usar o Jenkins

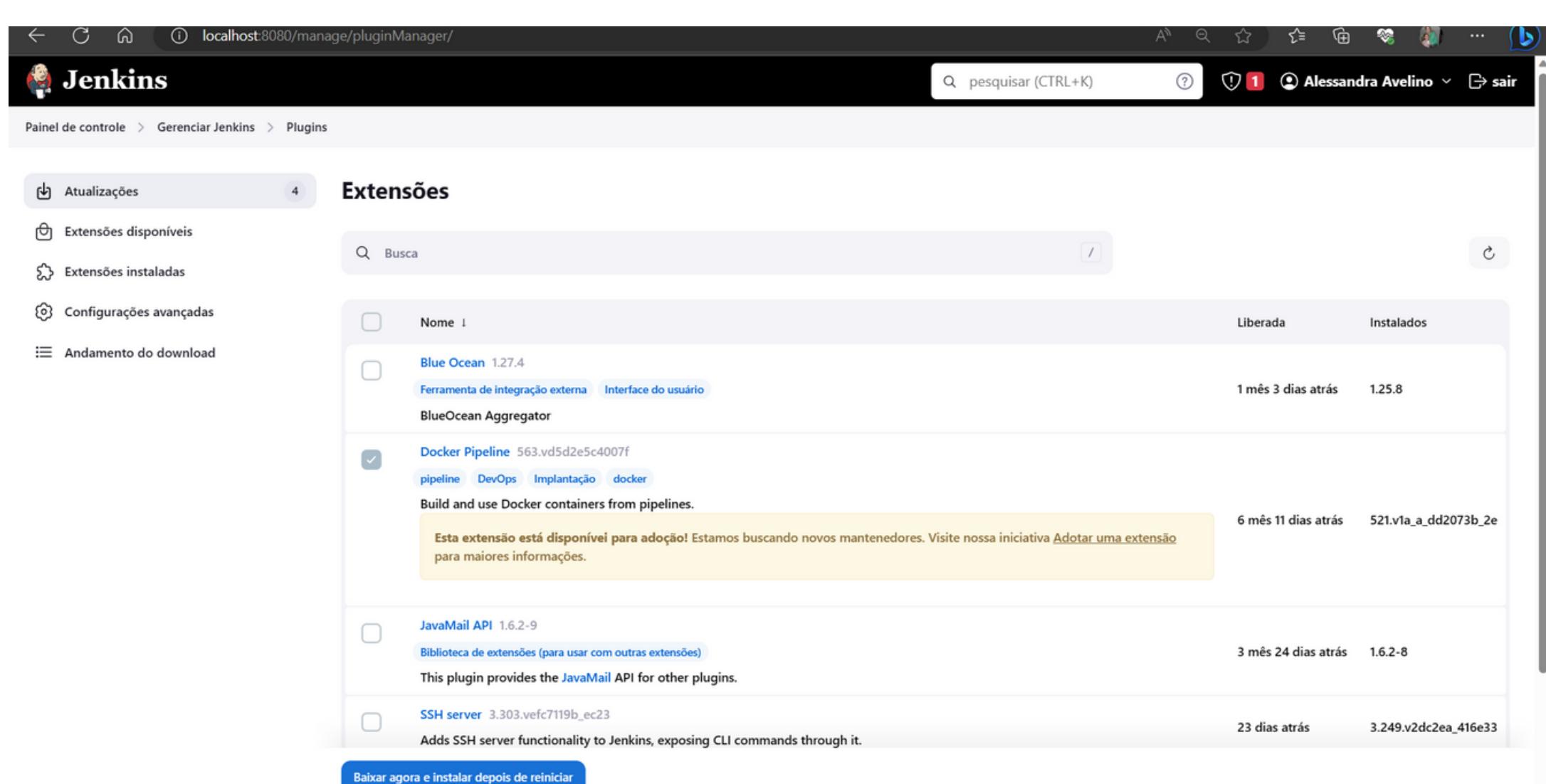
Jenkins 2.409

Pesquisar PTB2 18:01 10/06/2023

Navegue até Gerenciar Jenkins > Plugins e procure por Docker Pipeline, para baixar o plugin necessário para a integração entre o Docker e o Jenkins.



The screenshot shows the Jenkins Manage Jenkins interface. In the left sidebar, under 'Gerenciar Jenkins', the 'Plugins' option is selected. The main content area is titled 'Gerenciar o Jenkins' and contains sections for 'Configuração de sistema' and 'Segurança'. The 'Plugins' section is highlighted with a yellow background and shows four items: 'System', 'Nodes', 'Clouds', and 'Plugins'. The 'Plugins' item has a sub-section with a warning message: 'Construir no nó embutido pode ser um problema de segurança. Você deveria configurar construções distribuídas. Vide a documentação sobre isso.' Below this, there are buttons for 'Configurar agente', 'Configurar nuvem', and 'Dispensar'. On the left, there are dropdown menus for 'Fila de construções' and 'Estado do executor de construções', both currently set to 'Nenhum'.



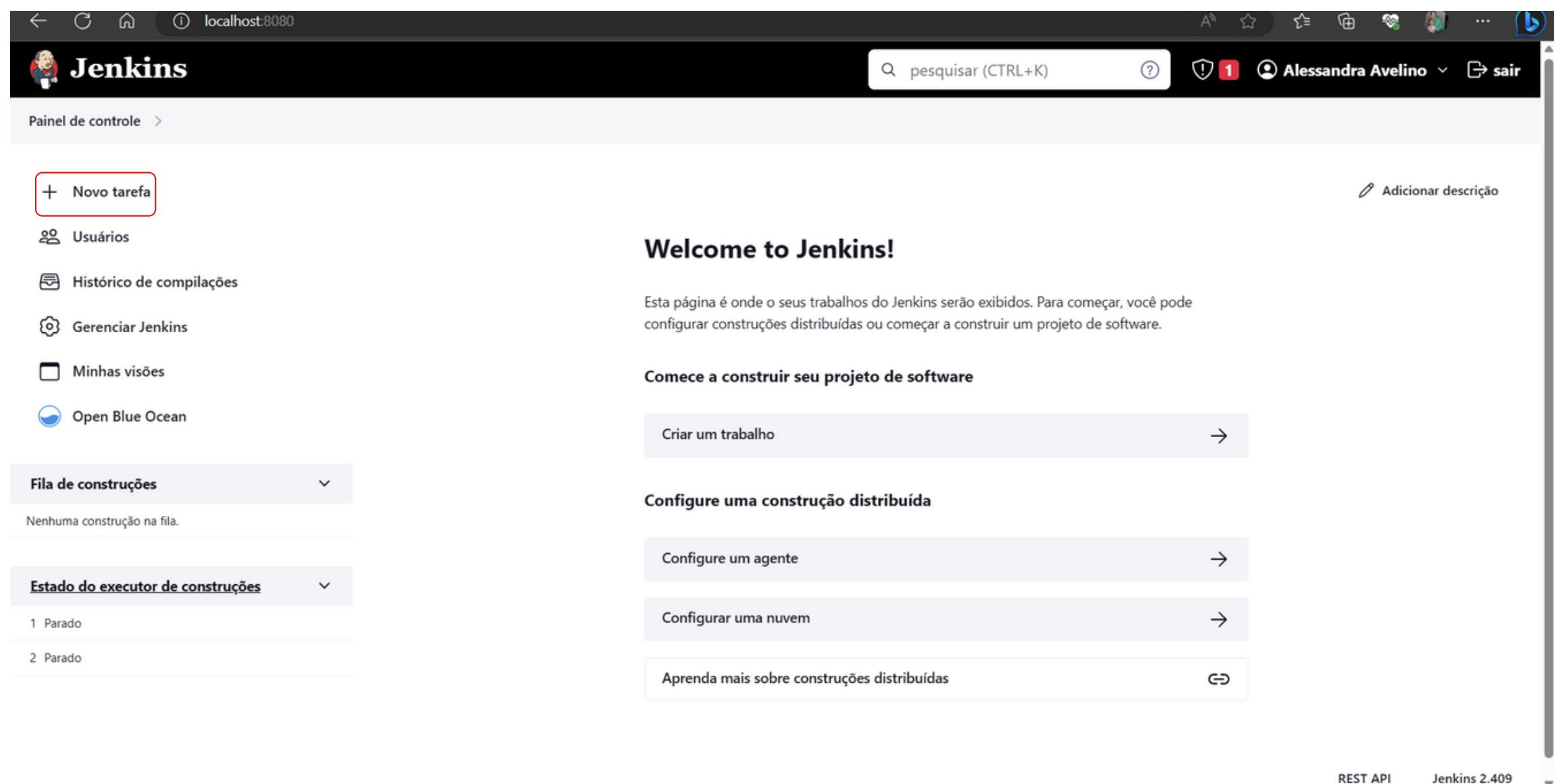
The screenshot shows the Jenkins Plugin Manager interface. The left sidebar shows 'Atualizações' with 4 available updates. The main content area is titled 'Extensões' and lists several available plugins: 'Blue Ocean', 'Docker Pipeline', 'JavaMail API', and 'SSH server'. The 'Docker Pipeline' plugin is selected, indicated by a checked checkbox. A tooltip for this plugin states: 'Esta extensão está disponível para adoção! Estamos buscando novos mantenedores. Visite nossa iniciativa [Adotar uma extensão](#) para maiores informações.' At the bottom of the list is a blue button labeled 'Baixar agora e instalar depois de reiniciar'.

6

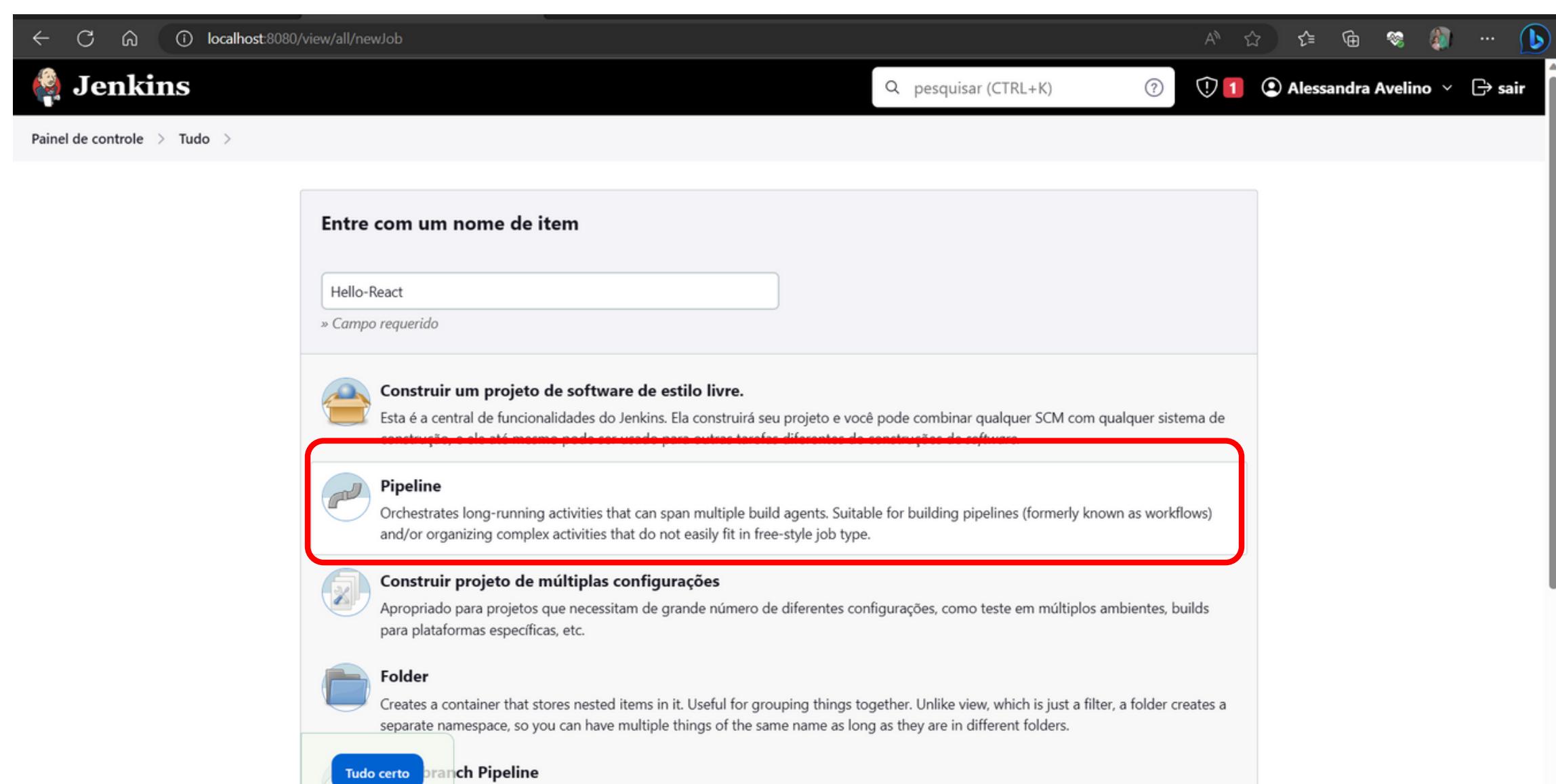
Criando a Pipeline a partir de um projeto do GitHub

Nesta etapa, finalmente criaremos a nossa Pipeline. O primeiro passo será você subir a sua aplicação (no nosso caso, será em ReactJS) para o Github.

Após isso, no **Painel Principal** do Jenkins, clique em **Nova Tarefa**. Em seguida, insira um nome e clique em **Pipeline**. Conforme feito nas imagens abaixo:



The screenshot shows the Jenkins dashboard at localhost:8080. The sidebar on the left has a red box around the '+ Novo tarefa' button. The main content area displays the 'Welcome to Jenkins!' message and various configuration options like 'Criar um trabalho', 'Configure uma construção distribuída', and 'Pipeline'.



The screenshot shows the 'New Item' creation page at localhost:8080/view/all/newJob. A red box highlights the 'Pipeline' option under the 'Construir um projeto de software de estilo livre' section. Other options shown include 'Construir projeto de múltiplas configurações' and 'Folder'.

Criando os arquivos Dockerfile e Jenkinsfile

Antes de seguirmos na pipeline, iremos criar os seguintes arquivos na raíz do projeto hello-react.

Em resumo, no Dockerfile especificamos a imagem e a versão que iremos utilizar, o diretório de trabalho que contem a aplicação a qual será copiada e executada, também é instalada as dependências e finalmente um comando para iniciar a aplicação.

No jenkinsfile, definimos um agente o qual será responsável por executar a pipeline, definimos também a imagem docker, o mapeamento das portas, os estágios da aplicação, que no nosso caso há apenas o 'Build', e em steps definimos as ações que serão executadas. Por fim, inicializamos a instalação das dependências e no fim, é realizad o comando npm run build, para que a aplicação seja executada.

```
// JENKINSFILE

pipeline {
    agent {
        docker {
            image 'node:20.2.0-alpine3.17'
            args '-p 3000:3000'
        }
    }
    stages {
        stage('Build') {
            steps {
                sh 'npm install'
                sh 'CI=false npm run build'
            }
        }
    }
}
```

```
# DOCKERFILE

# Imagem de Origem
FROM node:20.2.0-alpine3.17

# Diretório de trabalho(é onde a aplicação ficará dentro do container).
WORKDIR /app

# Adicionando `/app/node_modules/.bin` para o $PATH
ENV PATH /app/node_modules/.bin:$PATH

# Instalando dependências da aplicação e armazenando em cache.
COPY package.json /app/package.json
RUN npm install --silent
RUN npm install react-scripts@3.3.1 -g --silent

# Inicializa a aplicação
CMD ["npm", "start"]
```

8

Clone do projeto no Jenkins

Retomando do ponto em que paramos no Jenkins, nesta etapa iremos especificar o projeto que queremos dar o build a partir do github, inserindo as seguintes informações:

Após isso, será redirecionado pra essa página, basta clicar em **Construir agora** e você verá a pipeline executando. Clique em "**Open Blue Ocean**", para ver os logs da execução.

9

Configurando permissões do usuário

É provável que ocorra erro de permissão negada quando o Jenkins tenta se comunicar com o Docker. Para solucionar isto, abra o seu terminal e insira os seguintes comandos:



The screenshot shows the Jenkins Pipeline interface for a job named 'Hello-React 1'. The pipeline step 'docker pull' has failed with the following error message:

```
+ docker pull node:20.2.0-alpine3.17
2  permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://<2>var<2>run<2>docker.sock/v1.24/images/create?fromImage=node&tag=20.2.0-alpine3.17": dial unix
3 /var/run/docker.sock: connect: permission denied
script returned exit code 1
```



```
# 1º Navegar para a pasta
$ cd /var/run

# 2º Lista todos os arquivos e diretórios no diretório atual
$ ls -a

# 3º Altera as permissões do arquivo conceder acesso total
$ sudo chmod 777 /var/run/docker.sock

# 4º Retornará uma lista de nomes de usuários do sistema
$ cut -d: -f1 /etc/passwd

# 5º Cria um novo usuário chamado "jenkins" no sistema
$ sudo user add jenkins

# 6º Novamente etornará uma lista de nomes de usuários do sistema.
$ cut -d: f1 /etc/passwd

# 7º Mosta os usuários que fazem parte do grupo docker
$ grep docker /etc/group

# 8º Mosta os usuários que fazem parte do grupo root
$ grep root /etc/group

# 9º Adiciona o usuário "jenkins" ao grupo "docker"
$ sudo usermod -aG docker jenkins

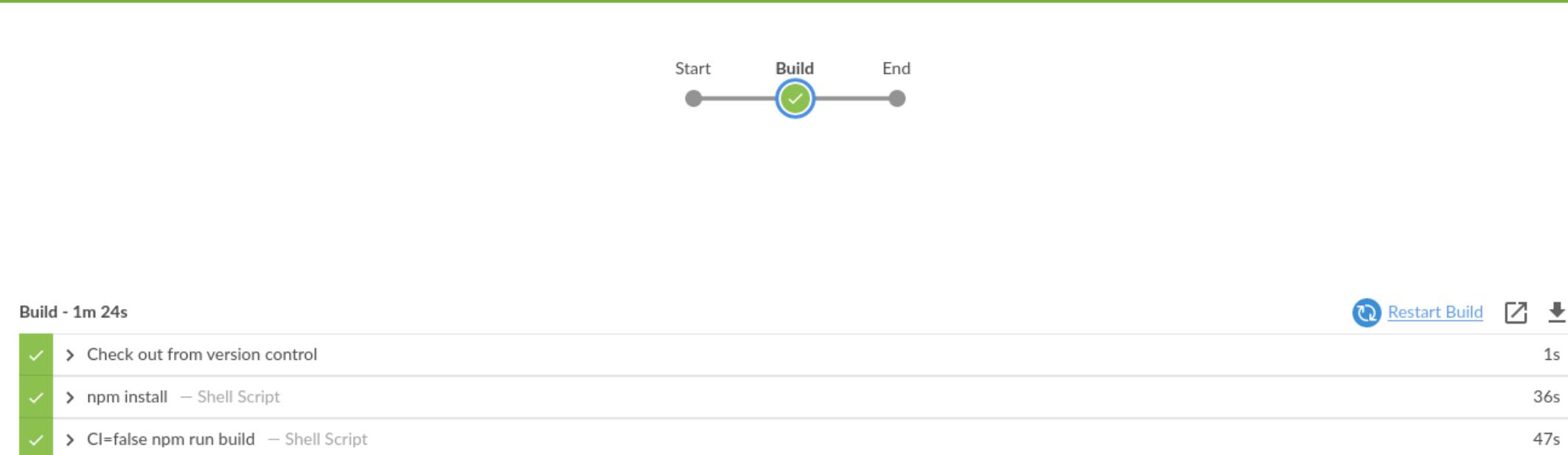
# 10º Adiciona o usuário "jenkins" ao grupo "root"
$ sudo usermod -a -G root jenkins

# 11º Adiciona o usuário atual ao grupo "docker"
$ sudo usermod -a -G docker $USER

#12º Verifica se o usuário "jenkins" foi adicionado ao grupo "root"
$ grep root /etc/group
```

10 Deploy

Após seguir os comandos, construa novamente a Pipeline, e acompanhe os logs no Blue Ocean, e verá que tudo ocorrerá corretamente dessa vez. Ficando da seguinte maneira:



Repositórios dos projetos utilizado neste tutorial

Contêiner: <https://github.com/alessandraavelino/jenkins>

Projeto hello-react: <https://github.com/alessandraavelino/hello-react>



Jenkins é uma ferramenta que permite criar fluxos de trabalho automatizados para agilizar o desenvolvimento de software. Ele automatiza várias etapas do processo, desde a verificação contínua do código até a disponibilização da versão final do software. Isso elimina a necessidade de execução manual dessas tarefas e ajuda a reduzir erros, acelerar o tempo de lançamento e melhorar a qualidade geral do software.