

Instalação e Configuração do GitLab com Docker e Automatização de Pipelines

Neste guia, vamos abordar o processo de instalação e configuração do GitLab Runner utilizando Docker, além de mostrar como automatizar a execução de pipelines no GitLab. A importância dessa configuração reside no fato de que, ao rodar o GitLab Runner em contêineres Docker, você consegue obter um ambiente altamente escalável e flexível para executar seus pipelines de integração contínua (CI). Com Docker, é possível gerenciar as dependências de forma isolada, garantir maior segurança e realizar a automação de testes e implantações de maneira eficiente. A automatização de pipelines também assegura a execução contínua de tarefas críticas sem intervenção manual, otimizando os processos de desenvolvimento.

1. Atualização do Sistema

Antes de começar, é importante garantir que todos os pacotes do sistema estejam atualizados:

```
sudo apt update && sudo apt upgrade -y
```

- **apt update:** Atualiza a lista de pacotes disponíveis.
- **apt upgrade -y:** Instala as atualizações disponíveis para os pacotes já instalados.

2. Instalação do Docker

O GitLab Runner será configurado para rodar dentro de contêineres Docker. Primeiramente, precisamos instalar o Docker:

2.1. Adicionar o Repositório do Docker

```
sudo apt-get install apt-transport-https ca-certificates curl software-properties-common  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor  
or -o /usr/share/keyrings/docker-archive-keyring.gpg  
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

- **apt-transport-https, ca-certificates, curl, software-properties-common:** Pacotes necessários para adicionar repositórios externos via HTTPS.
- **curl -fsSL:** Baixa a chave GPG oficial do Docker.
- **echo "deb [arch...:** Adiciona o repositório do Docker às fontes do apt.

2.2. Instalação do Docker

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

- **docker-ce**: Instala o Docker Community Edition.
- **docker-ce-cli**: Instala a CLI do Docker.
- **containerd.io**: Instala o runtime do container.

2.3. Habilitar e Iniciar o Docker

```
sudo systemctl enable docker
sudo systemctl start docker
```

- **systemctl enable docker**: Configura o Docker para iniciar automaticamente junto com o sistema.
- **systemctl start docker**: Inicia o serviço Docker imediatamente.

3. Instalação e Configuração do GitLab Runner

3.1. Adicionar o Repositório Oficial do GitLab Runner

```
curl -L https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh | sudo bash
```

- **curl -L**: Baixa e executa o script de instalação do repositório oficial do GitLab Runner.

3.2. Instalar o GitLab Runner

```
sudo apt-get install gitlab-runner
```

- **gitlab-runner**: Instala o GitLab Runner, que será utilizado para executar os jobs das pipelines.

3.3. Configurar o GitLab Runner como um Serviço

```
sudo gitlab-runner install --user=gitlab-runner --working-
directory=/home/gitlab-runner
sudo gitlab-runner start
```

- **gitlab-runner install**: Instala o GitLab Runner como um serviço no sistema.
- **gitlab-runner start**: Inicia o serviço do GitLab Runner.

3.4. Adicionar o Usuário GitLab Runner ao Grupo Docker

```
sudo usermod -aG docker gitlab-runner
```

- **usermod -aG docker**: Adiciona o usuário gitlab-runner ao grupo docker, permitindo que ele execute comandos Docker.

3.5. Reiniciar os Serviços

```
sudo systemctl restart gitlab-runner  
sudo systemctl restart docker
```

- **systemctl restart:** Reinicia os serviços do GitLab Runner e Docker para aplicar as configurações.

4. Registro e Configuração do GitLab Runner

4.1. Registrar o GitLab Runner

```
sudo gitlab-runner register --url http://cicd.com.br/ --registration-  
token SEU_TOKEN_DE_REGISTRO
```

- **gitlab-runner register:** Comando para registrar o GitLab Runner com a instância do GitLab usando o URL e o token de registro fornecido.

4.2. Configuração de Pipeline Automática

Para automatizar as pipelines, utilize o seguinte comando:

```
curl -X POST --fail -F token=$TRIGGER_TOKEN -F ref=main http://cicd.com.br  
/api/v4/projects/3/trigger/pipeline
```

- **curl -X POST:** Envia uma requisição POST para o GitLab para disparar uma pipeline.
- **--fail:** Faz o curl retornar um erro caso a requisição falhe.
- **-F token=\$TRIGGER_TOKEN:** Passa o token da trigger da pipeline.
- **-F ref=main:** Especifica o branch "main" para a pipeline.

Este comando pode ser automatizado através de um script shell ou integrado diretamente no processo de CI/CD para garantir que as pipelines sejam executadas conforme o esperado.

5. Conclusão

Seguindo estes passos, você terá configurado o GitLab Runner para executar pipelines em contêineres Docker, com suporte completo para automação. Esta configuração é ideal para ambientes de CI/CD robustos e escaláveis.