

Acesso a aplicação

endereço: 98.82.14.50/zabbix

User: Admin

Senha: zabbix

Para a proposta em questão foi utilizada a VPC default da AWS para evitar custos com IPV4 públicos (este que foi anunciado no início de 2024). Mais abaixo descrevo como deveria estar montado o ambiente a fim de adequá-lo às melhores práticas

1 - Servidor Zabbix

Para realização da atividade foram utilizadas 1 instância EC2 para montar o servidor de aplicação com o Zabbix e uma segunda instância com o banco de dados postgresql.

Abaixo os comandos utilizados para instalar o zabbix:

#download do zabbix

```
wgethttps://repo.zabbix.com/zabbix/7.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_7.0-2+ubuntu24.04_all.deb
```

#instalação dos pacotes

```
dpkg -i zabbix-release_7.0-2+ubuntu24.04_all.deb
```

```
apt update
```

Instalação do frontend zabbix

```
apt install zabbix-server-pgsql zabbix-frontend-php php8.3-pgsql zabbix-apache-conf  
zabbix-sql-scripts zabbix-agent
```

Instalação do cliente Postgresql

```
sudo apt install postgresql postgresql-contrib -y
```

Importando o esquema inicial e os dados no servidor zabbix utilizando o ip interno da instância com o banco de dados

```
zcat /usr/share/zabbix-sql-scripts/postgresql/server.sql.gz | psql -h 172.31.86.51 -u zabbix -d  
zabbix
```

#editar arquivo nano /etc/zabbix/zabbix_server.conf para que a conexão do banco com a aplicação seja possível.

```
sudo nano /etc/zabbix/zabbix_server.conf
```

```
DBUser = DBUser=zabbix
```

```
DBPassword = DBPassword=5ffc0a32
```

```
DBHost=local = DBHost=172.31.86.51
```

#startando o serviço - Nesse passo iniciamos o serviço do zabbix

- systemctl restart zabbix-server zabbix-agent apache2
- systemctl enable zabbix-server zabbix-agent apache2

O comando enable garante que o serviço reinicie junto caso a instância seja desligada

Por fim para acessar o zabbix usamos o endereço da instancia/zabbix (dessa maneira 98.82.14.50/zabbix). Lembrando de permitir as portas 443 e 80 no security group associado a instancia

2 - Banco de Dados

O banco de dados Postgresql foi instalado em uma segunda instância EC2 utilizando o seguinte script:

Para instalação

```
sudo apt-get -y install postgresql
```

#Acessando o banco

```
sudo -u postgres psql
```

#Instalação do repositório postgresql

```
sudo apt install postgresql postgresql-contrib
```

#Startando o banco

```
sudo systemctl restart postgresql
```

#Verifica o status

```
sudo systemctl status postgresql
```

#Garante que o banco inicie junto com a instância caso seja desligada

```
sudo systemctl enable postgresql
```

#Dentro do banco criamos o usuário zabbix

```
CREATE USER zabbix WITH PASSWORD '5ffc0a32';
```

Também criamos o banco de dados zabbix que utilizaremos na instância com a aplicação para importar o esquema inicial

```
CREATE DATABASE zabbix OWNER zabbix;
```

Dentro do banco também precisamos dar as permissões de acesso a partir da rede da vpc dentro do arquivo pg_hba.conf localizado no destino /etc/postgresql/16/main, basta adicionar o endereço da rede ao campo ipv4 connections e testamos a conexão com o comando

```
psql -h <host> -U postgres -d <database>
```

Também é necessário verificar se o banco está escutando em todas as interfaces, no arquivo localizado no destino /etc/postgresql/<version>/main/postgresql.conf editamos o seguinte campo listen_addresses = ''

Dentro do banco de dados também fazemos a instalação do zabbix agent para monitorar o SO como solicitado

Download do agente

```
wget https://repo.zabbix.com/zabbix/7.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_7.0-2+ubuntu24.04_all.deb
```

```
dpkg -i zabbix-release_7.0-2+ubuntu24.04_all.deb
```

```
apt update
```

```
apt install zabbix-agent
```

#editar config do agente

```
sudo nano/etc/zabbix/zabbix_agent2.conf
```

Procuramos pelos seguintes parâmetros e alteramos para que o zabbix possa coletar as informações

server 127.0.0.1 e substitua pelo ip do seu zabbix server

ServerActive=zabbix_server_ip_address # substitua 127.0.0.1 pelo ip do seu zabbix server

Hostname #define um nome para o servidor a ser monitorado

como ta na mesma rede eu posso pegar o ip privado do server-

#reinicia o agente

- sudo systemctl restart zabbix-agent2
- systemctl enable zabbix-agent ---- garante que reinicie junto com a máquina caso a mesma seja desligada em algum momento
- systemctl status zabbix-agent - serve para verificar se o agente está rodando normalmente

Já dentro da aplicação Zabbix

Fazemos o cadastro do host para ser monitorado, na aba hosts criamos um novo, adicionamos o ip, o nome definido e o template que será usado para monitorar o SO.

Para esta instância foram adicionadas liberadas as portas de acesso 5432 para o postgresql e 10050/10051 para que o zabbix possa coletar informações

Em seguida rodamos o script solicitado para teste

```
CREATE DATABASE testdb;
```

```
CREATE TABLE table1 (
```

```
column1 integer,
```

```
column2 integer);
```

```
INSERT INTO table1 (column1)
```

```
SELECT a.column1 FROM generate_series(1, 1000000) AS a (column1);
```

3 - Backup

Para criação do backup foi optado por utilizar a linguagem Python, este que será explicado em outro arquivo. Para que seja possível rodar o script na EC2 com o Postgresql primeiro criamos o arquivo backup_postgres.py. e damos permissão para rodar com o comando `chmod +x backup_postgres.py`

Para rodar esse arquivo precisamos fazer a instalação do cli da AWS e a biblioteca boto3

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

```
unzip awscliv2.zip
```

```
sudo ./aws/install
```

```
sudo apt-get update
```

```
sudo apt-get install python3-boto3
```

Em seguida criamos o bucket S3 em que serão armazenados os scripts com o nome backuppostgresabbix e adicionamos a role de acesso à instância EC2 com o banco de dados.

Fazendo isso entramos com nossa accesskey e secret access key e rodamos o scrip com o comando `python3 backup_postgres.py` com isso podemos verificar o arquivo [testdb_2024-08-31.backup](#) no S3.

Se quisermos restaurar o banco precisamos seguir os seguintes passos.

1. Transferir o Arquivo de Backup do S3 para a Instância EC2

Se o backup está armazenado no S3, você pode baixá-lo de volta para a instância EC2 usando o comando `aws s3 cp`:

```
aws s3 cp s3://renan-dias-2024/testdb_2024-08-31.backup /path/to/destination/
```

Substituir o `/path/to/destination/` pelo diretório onde deseja salvar o arquivo na instância EC2.

2. Restaurar o Backup Usando `pg_restore`

O comando `pg_restore` é usado para restaurar bancos de dados a partir de backups gerados com o `pg_dump` em formato custom (-F c no comando de backup).

Passos para restaurar o backup:

1. Conectar ao PostgreSQL:

Certifique-se de que o servidor PostgreSQL está rodando na instância onde você quer restaurar o backup.

```
pg_restore -h localhost -U postgres -d testdb -v /path/to/destination/testdb_2024-08-31.backup
```