

Acesso a aplicação:
endereço: 44.204.74.220/zabbix
user: Admin
senha: 5ffc0a32

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 instancia EC2 para montar o servidor de aplicação com o Zabbix e uma segunda instância com o agente do Zabbix instalado e a porta 10050 aberta a fim de coletar as informações de monitoramento (lembrando que o zabbix já monitora o host em que está instalado também).

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 Cliente mysql

apt install -y zabbix-server-mysql install zabbix-frontend-php install zabbix-apache-conf zabbix-sql-scripts zabbix-agent

#conectando no banco - Aqui utilizamos o DB endpoint gerado ao criar a instancia RDS.

mysql -h zabbix-database.chqsa662obh9.us-east-1.rds.amazonaws.com -u_zabbix -p5ffc0a32

#criando Banco de dados

create database zabbix character set utf8 collate utf8_bin;

#Importando o esquema inicial e os dados no servido zabbix

zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -h zabbix-database.chqsa662obh9.us-east-1.rds.amazonaws.com -uzabbix -p5ffc0a32 zabbix

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

sudo nano /etc/zabbix/zabbix_server.conf

DBUser=zabbix | DBUser=<DB_USER>

#DBPassword | #DBHost=local |

#editar arquivo apache - nessa parte apenas selecionar a timezone (em nosso caso America/Sao Paulo)

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

#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 44.204.74.220/zabbix)

Subindo a segunda instância e instalando o agente Zabbix para monitorar

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_agent.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-agent
- 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 do Zabbix precisamos cadastrar o nosso host a ser monitorado, indo até a aba Host -> opção create host-> cadastrar o ip e selecionar o template que em nosso caso utilizamos um template default para monitorar máquinas Linux.

2 - Banco de Dados

Para o banco de dados foi optado por utilizar uma instância RDS com Mysql e importamos o esquema inicial e os dados do servidor zabbix pois ao utilizarmos o serviço gerenciado da AWS tornamos o ambiente mais fácil de operar automatizando backups, atualizações de software, replicação e dimensionamento utilizando o storage autoscaling por exemplo.

Para criar a instância usamos a console da aws e definimos as opções desejadas. Depois de criar a instância gera o db endpoint que usamos nos passos anteriores para conectar a mesmo

3 - Com relação a Segurança

Com relação a segurança nossa instância com o zabbix fica em uma subnet pública para que possa ser acessado via internet com um security group liberando as portas 80 e 443 para o endereço em questão 45.7.128.158/32. Já o banco de dados fica sem um endereço de acesso público, podendo ser acessado apenas pela porta padrão do mysql 3306 aberta para o endereço do servidor Zabbix.

4- Referente ao backup

Como mencionado no item 2 utilizando o o RDS podemos automatizar os backups com um período de retenção de 1 a 35 dias e caso seja necessário um período maior precisamos exportar esse backup para o s3. Esse backup faz um snapshot do banco e armazena os logs de transação, isso nos permite fazer o point in time recovery rapidamente dentro do período de retenção.

Os dados são armazenados no S3 e o cliente recebe storage gratuita igual o espaço do tamanho do banco de dados.

Esse backup é feito dentro de uma janela de manutenção definida pelo operador e pode ser habilitado tanto ao criar a instância quanto depois selecionando a instancia em questão, selecionando modify e navegando até a aba referente a backup.

OBS: Se em algum momento restaurarmos o banco, será criada uma nova instância do mesmo, ou seja, precisa fazer a aplicação apontar para o novo endpoint.

Abaixo segue a arquitetura proposta e também recomendações para um ambiente de produção:

- 1 - Utilizar uma VPC personalizada para a aplicação
- 2 - Colocar o banco de dados em uma rede privada exposto apenas para a aplicação zabbix



Region us-east-1

