

Insper, 2024-1

Computação em Nuvem

Projeto IaC

Jerônimo de Abreu Afrange

Passo a passo de implantação: README.md

Repositório: <https://github.com/jeronimo-a/projeto-cloud-2024-1>

Introdução

Neste projeto, foi feita a implementação de uma infraestrutura de servidores *web* utilizando o recurso de IaC *CloudFormation*, da AWS. O *template* do projeto gera uma *stack* com servidores *WordPress* com escalabilidade automática, uma única base de dados compartilhada, e balanceamento de carga de acesso aos servidores.

Etapas de implementação

1 – Base de dados

O primeiro item necessário para a criação da infraestrutura é a base de dados. Optou-se por fazer uso de uma base de dados MySQL ao invés de *DynamoDB* da AWS, porque o WordPress não funciona, por padrão, com a *DynamoDB*, simplificando o processo de implantação.

É necessário implementar a base de dados primeiro porque é durante a criação da imagem do servidor que ela é populada com os dados da aplicação *web*.

2 – Imagem aplicação *web*

O segundo item necessário é a imagem de servidor com a aplicação *web* instalada. Para a criação deste item, é necessário

desenvolver um *script* de instalação que será executado em uma máquina feita a partir de uma imagem base. Neste projeto, a imagem base é a do Ubuntu Server 24.04 x86-64.

O *script* de instalação faz cinco (5) coisas: instala as dependências do *WordPress* na máquina, instala o *WordPress* em si, instala a interface de linha de comando do *WordPress*, configura o *apache* para disponibilizar o *WordPress* e faz a população da base de dados com os dados do *WordPress*.

3 – *Launch configuration e auto-scaling group*

A configuração de inicialização (*launch configuration*) é a parte principal do sistema de auto escalamento (*auto-scaling*). Nela, é definida como serão inicializadas as instâncias do servidor *web*.

Neste projeto, elas são inicializadas a partir da imagem criada no passo anterior, e reconfiguram a base de dados para a URL do *site* no momento de inicialização para a URL do *load balancer* (quando as instâncias forem criadas, todos os itens abaixo já estarão prontos).

O *autoscaling group* faz referência ao *launch configuration* e agrupa todas as instâncias criadas a partir dele. É nessa etapa que se definem também as políticas de aumento e redução de capacidade. Optou-se por aumentar a quantidade de instâncias disponíveis em um (1) quando a utilização média das CPUs das instâncias ultrapassar os 70%, e diminuir em um (1) quando a mesma métrica estiver abaixo de 20%.

4 – *Load balancer*

O *load balancer* é o recurso responsável por receber todo o tráfego destinado aos servidores *web* e redirecioná-lo às instâncias mais adequadas. A definição de se uma instância é ou não é adequada é determinada a partir dos *health checks*, que são verificações de funcionalidade dos servidores. Neste projeto a verificação é simplesmente o código de resposta do servidor, código 200 indica que a instância está apta a receber tráfego.

Infraestrutura completa

As etapas acima foram descritas de forma simplificada, para cada uma delas houve a criação de outros recursos necessários para a sua implementação. Segue abaixo a lista de todos os recursos inclusos no *template*:

- **Project Virtual Private Cloud:** a VPC é a unidade organizacional mais alta de todas, ela é uma abstração de uma nuvem física, ou seja, de um *datacenter* físico. Nela estão contidos todos os demais recursos.

- **Project Subnet:** é a sub rede principal da infraestrutura, todos os recursos pertencem a ela.

- **Database Redundancy Subnet:** é a sub rede de redundância, definida em uma zona de disponibilidade diferente da sub rede principal. Essa sub rede é utilizada pela instância da base de dados e pelo *load balancer*, mas ambos fazem parte também da sub rede principal (*project subnet*).

- **Project Internet Gateway:** é o recurso necessário para conectar a VPC à internet.

- **Project VPC Gateway Attachment:** é o recurso que liga o *project internet gateway* à VPC.

- **Project Route Table:** é a tabela de rotas, cada rota relaciona uma rede de destino a um *gateway*. É similar a um roteador.

- **Project Route:** é a única rota definida na tabela acima, relaciona qualquer endereço de destino ao *gateway* da internet.

- **Project Subnet Route Table Association:** associa a tabela de rotas à sub rede principal do projeto.

- **Project Key Pair:** define o par de chaves que será incluído nas instâncias EC2, é usado apenas para *debugging*.

- **MySQL Security Group:** é o grupo de segurança que faz com que os seus membros aceitem conexões na porta 3306 vindas de qualquer lugar de dentro da VPC. O único membro é a instância da base de dados.

- **Webserver Security Group:** é o grupo de segurança que faz com que os seus membros aceitem conexões na porta 80 vindas de qualquer lugar. Os membros desse grupo são as instâncias do servidor *web* e o *load balancer*.

- **SSH Security Group:** é o grupo de segurança que faz com que os seus membros aceitem conexões na porta 22 vindas da sub rede definida como parâmetro. É usado somente para *debugging*.

- **Project DB Subnet Group:** é o grupo de sub redes que serão utilizadas pela base de dados. É necessário ter um grupo de mais de uma sub rede, em diferentes zonas de disponibilidade, por redundância.

- **Project DB Instance:** é a instância da base de dados. Nela ficam todos os dados do *site*.

- **Webserver Instance Role:** é a função (*role*) necessária para a instância de criação de configuração do *webserver* poder criar uma nova imagem.

- **Webserver Infrastructure Configuration:** define como é a instância utilizada para a criação da imagem do *webserver*.

- **Webserver Instance Profile:** define o perfil da instância de criação da imagem, associa o *role* a configuração de infraestrutura acima.

- **Webserver Image Component:** é o componente (nesse caso único) da receita de criação da imagem, contém todos os *scripts* necessários.

- **Webserver Image Recipe:** é a receita de criação da imagem, agrupa todos os componentes da receita (apenas um, nesse caso).

- **Webserver Distribution Configuration:** configura como que a imagem gerada será distribuída. Nesse caso, não importa muito.
- **Webserver Image:** é a imagem em si, depende da configuração de distribuição, da receita da imagem e da configuração da infraestrutura da instância de criação da imagem.
- **Webserver Load Balancer:** é o balanceador de carga em si.
- **Webserver Target Group:** configura o grupo alvo do balanceador de carga, que é o grupo de instâncias às quais será distribuído o tráfego web.
- **HTTP Listener:** define as políticas de encaminhamento de tráfego web do load balancer.
- **Webserver Launch Configuration:** configura como que é feita a inicialização das instâncias do servidor web. Aqui está contido o script de reconfiguração da URL do site.
- **Webserver Auto Scaling Group:** configurações do grupo de escalabilidade automática.
- **Webserver High CPU Alarm:** alarme que aumenta o número de instâncias no *auto scaling group* quando a utilização de CPU média passa de certo ponto.
- **Webserver Low CPU Alarm:** alarme que diminui o número de instâncias no *auto scaling group* quando a utilização de CPU média fica abaixo de certo ponto.
- **Webserver Scale Up Policy:** é a política de aumento da capacidade, depende somente do alarme de alta utilização de CPU.
- **Webserver Scale Down Policy:** é a política de redução da capacidade, depende somente do alarme de baixa utilização de CPU.

