**Termos gerais importantes:**

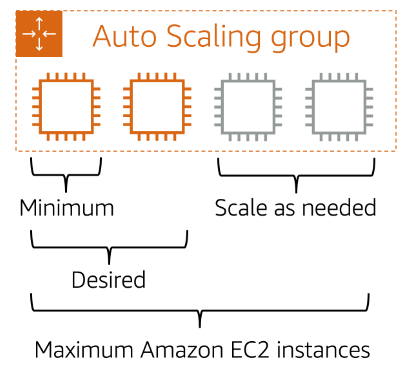
* **On-demand:** Serviços em nuvem disponíveis quando você precisar. Tais serviços não pertencem ao consumidor, e sim ao fornecedor.
* **On-premises:** Serviços físicos, como um servidor físico. O consumidor é responsável por instalar e manter os equipamentos.
* **Scalability:** É a capacidade do serviço se “moldar” à sua quantidade de demanda. Iniciamos o serviço somente com os recursos necessários, e vamos aumentando-os conforme a demanda aumenta. Podemos escalar verticalmente (ou seja, aumentar hardware) ou horizontalmente (ou seja, criando novas instâncias de nossas máquinas para atender à demanda adicional).
* **Serverless:** Quando, apesar do código ainda rodar em um servidor, não podemos ver nem acessar a infraestrutura.
* **Cluster:** Uma série de instâncias EC2 (contêineres) rodando juntos.

**Amazon EC2:**

**Termos importantes:**

* **Amazon EC2 Auto-Scaling:** Utilizado para escalar as instâncias de EC2 horizontalmente. Podemos utilizar o **dynamic scaling**, que responde dinamicamente às novas demandas, ou o **predictive scaling**, que agenda um número de instâncias a serem upadas em determinados horários, baseado em previsão.

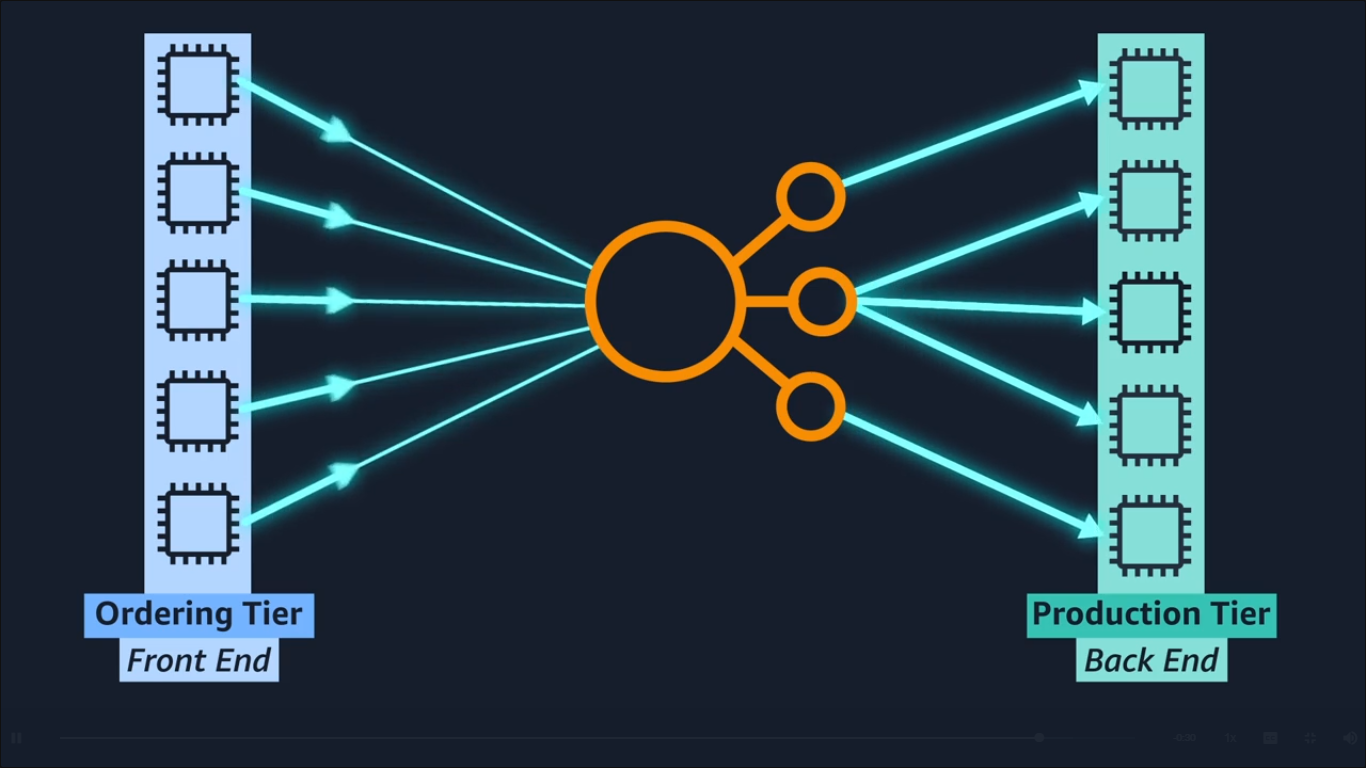
Ao iniciar um **Auto-Scaling Group**, definimos o número mínimo, o número “desejado” e o número máximo de instâncias. Só pagamos pelas instâncias que usamos, quando usamos.

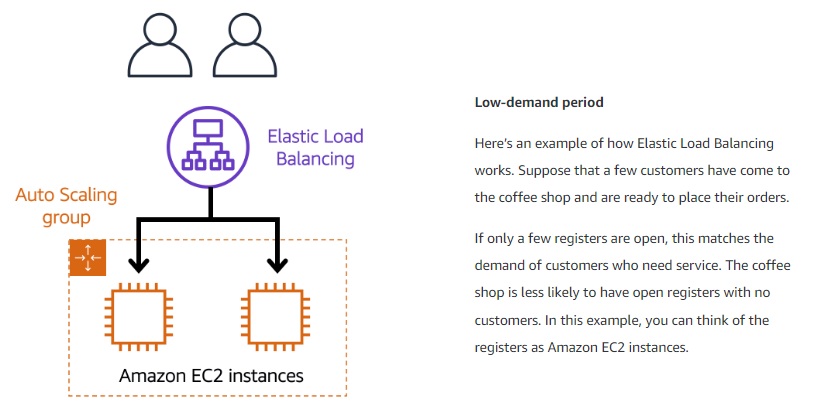
****

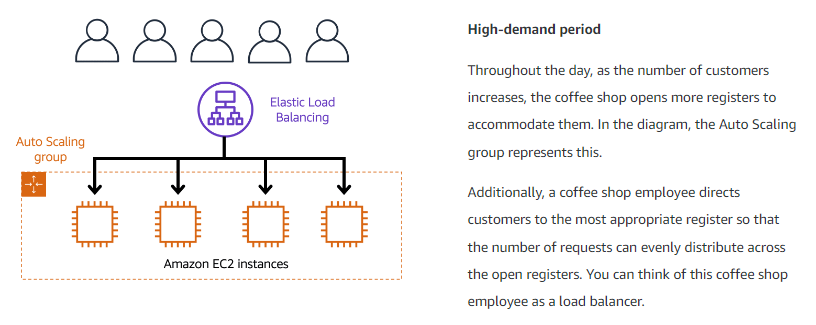
**Amazon ELB (Elastic Load Balancing)**

**Utilidade geral:**

* Serve para direcionar o tráfego entre as instâncias de EC2, por exemplo. Sua finalidade é não permitir que uma instância seja overloaded e outra underloaded.



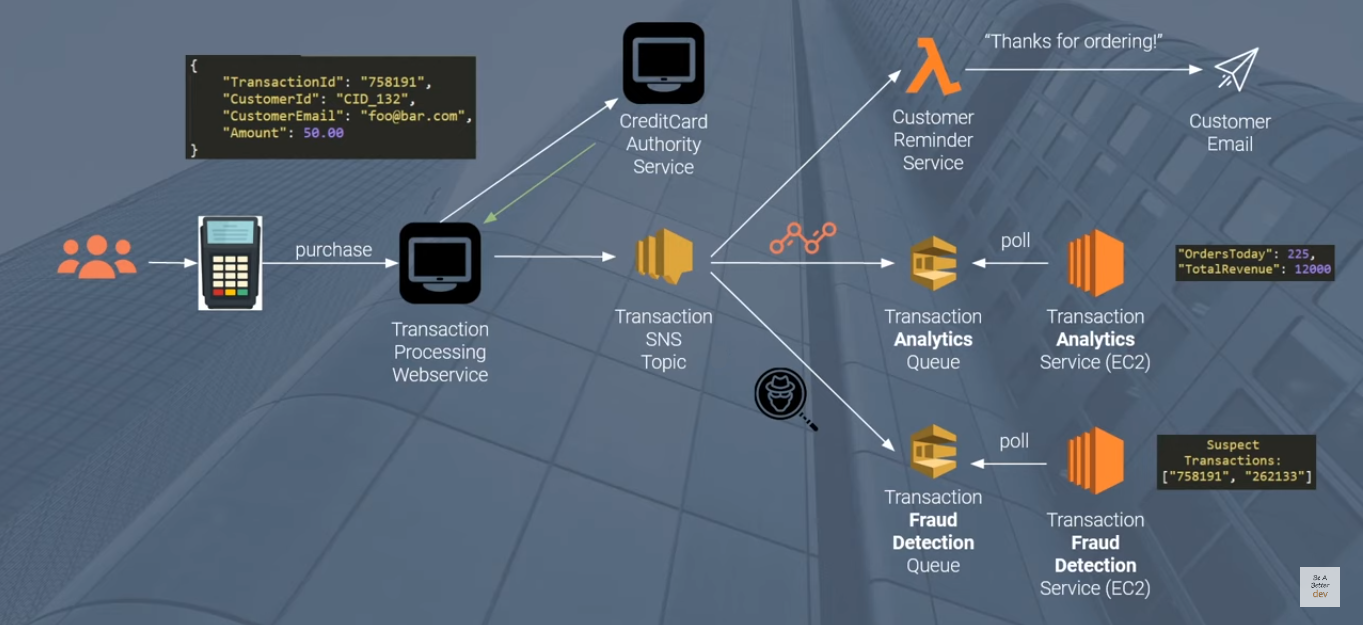
****

****

**Amazon SNS (Simple Notification Service)**

**Utilidade geral:**

* Serviço de publish/subscribe. Sistemas producers podem publicar mensagens para sistemas consumers (como o SQS, por exemplo), através do SNS Topic, que é responsável por realizar o fanout (direcionar a mesma mensagem para diversos consumers). Abaixo, segue uma boa ilustração de seu use-case ([SNS vs SQS Comparison? Whats the difference? | Learn with a practical example - YouTube](https://www.youtube.com/watch?v=mXk0MNjlO7A)):



**Amazon SQS (Simple Queue Service)**

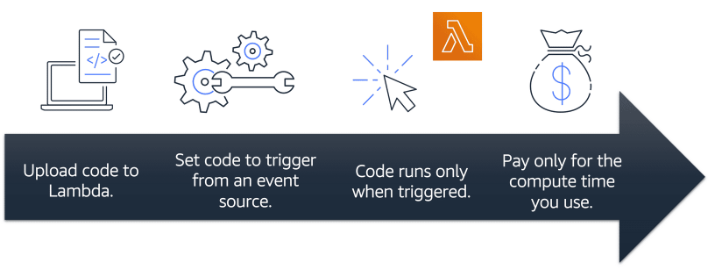
**Utilidade geral:**

* Uma fila, onde mensagens são publicadas e consumidas. Para melhor entendimento, consultar explicação do SNS acima.

**AWS Lambda**

**Utilidade geral:**

* Serviço onde subimos um código em uma **lambda function**, que será rodada de forma Serverless em um ambiente escalável e gerenciado pela AWS após a ativação de um trigger configurado. Designado para processamento rápido, que levam menos que 15 minutos. Não recomendado para deep-learning, por exemplo.





**Amazon ECS (Elastic Container Service)**

**Utilidade geral:**

* Serve para gerenciar aplicações contêinerizadas de forma escalável, sem que precisemos lidar com algum software próprio de orquestração de contêineres.

**Amazon EKS (Elastic Kubernetes Service)**

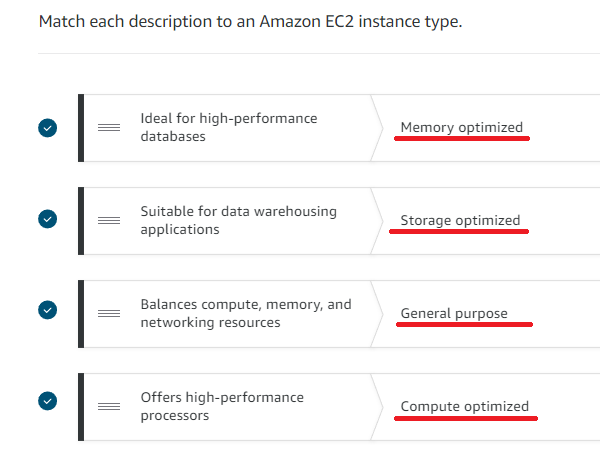
**Utilidade geral:**

**AWS FARGATE**

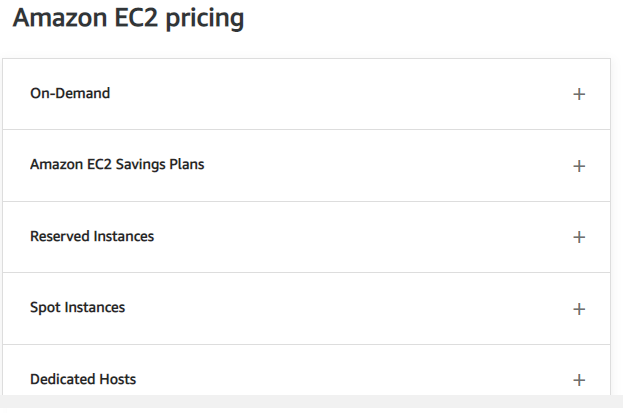
**Utilidade geral:**

* [**AWS Fargate**](https://aws.amazon.com/fargate/)is a serverless compute engine for containers. It works with both Amazon ECS and Amazon EKS. When using AWS Fargate, you do not need to provision or manage servers. AWS Fargate manages your server infrastructure for you.

**Tipos de instâncias EC2 (Lesson 7)**



**Tipos de pagamentos EC2 (Lesson 8)**

****