



# Computação em AWS

Willyan Guimarães - @willyancaetanodev

# Objetivo Geral

Conhecer os principais serviços de computação na AWS.

# Percurso

## **Etapa 1**

Elastic Compute Cloud – EC2

## **Etapa 2**

Amazon EC2 AutoScaling

## **Etapa 3**

Elastic Load Balancing – ELB

# Percurso

**Etapa 4** Serviços de mensageria

**Etapa 5** Computação sem servidor

**Etapa 6** Containers em AWS

# Pré-requisitos

- Nenhum pré-requisito específico

## Etapa 1

# Elastic Compute Cloud – EC2

# Um cenário real

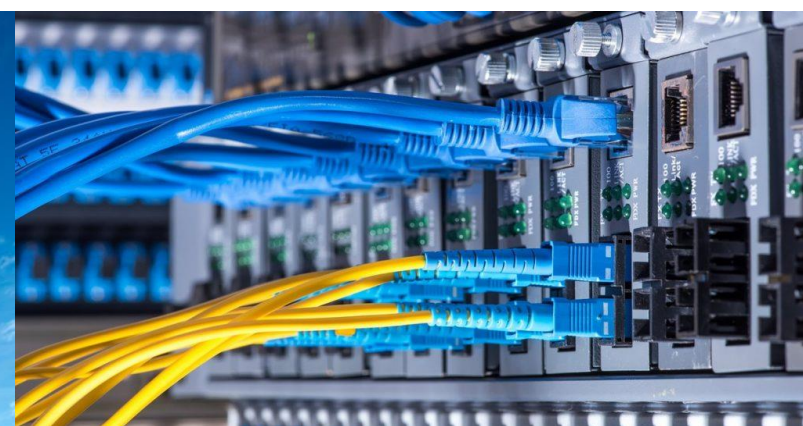
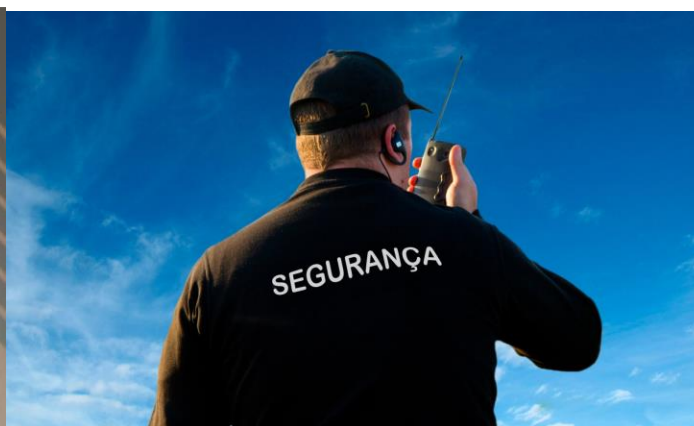




# Vou montar esse data center!



# Vou montar esse data center!





\$\$\$\$\$\$....



# E se...

- Todo o investimento nesse data center não for necessário dada sua demanda ?
- Um dia seu negócio crescer e a infraestrutura não suportar a demanda de usuários ?

# Então seria ideal

- Economia de recursos e custo
- Escalabilidade
- Elasticidade
- Disponibilidade

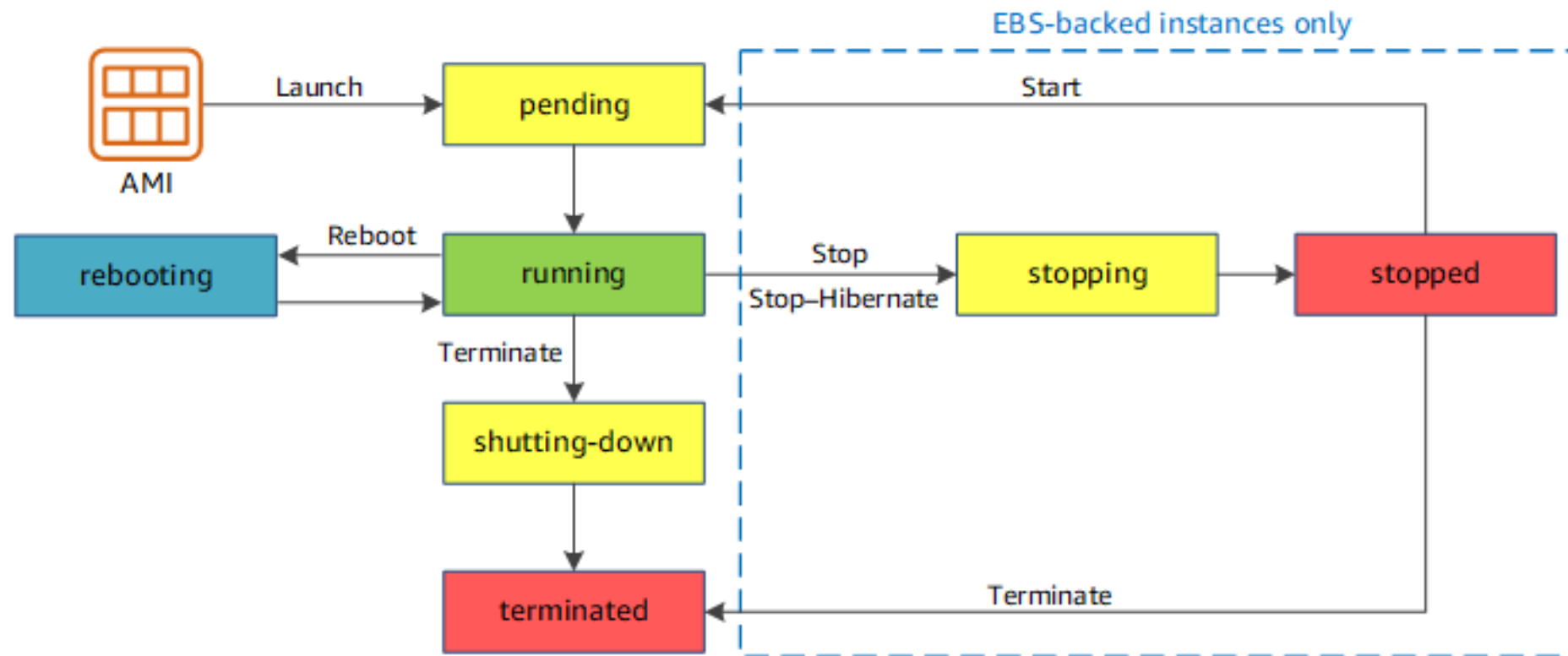
# EC2 - Elastic Compute Cloud

- Capacidade computacional segura e redimensionável
- Computação: CPU, Memória, Rede, Armazenamento, Sistema operacional
- Definição de preço conforme uso e modalidades específicas a necessidade
- Instâncias com tipos otimizados para sua atividade

# Conceito chave - Instância

- Servidor virtual na nuvem AWS
- Possui configurações de memória, CPU, disco, rede e sistema operacional

# Ciclo de vida





# Tipos de instância

- Diferentes tipos para casos de uso diferentes
- Várias combinações diferentes de configurações

# Tipos de instância

- Uso geral
- Otimizadas para computação
- Otimizadas para memória
- Computação acelerada
- Otimizadas para armazenamento

# Uso Geral

- Equilíbrio de recursos de computação, memória e rede
- Indicado para servidores de aplicativo, jogos, backend, banco de dados pequenos

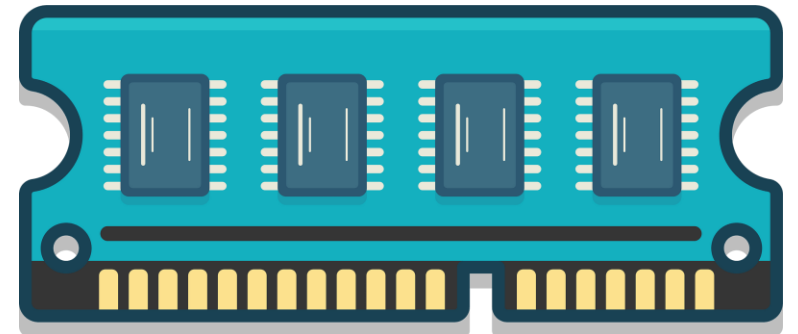


# Otimizadas para computação

- Ideal para cargas de trabalho que exigem processadores de alto desempenho
- Pode ser usado para os mesmos casos de uso da categoria de uso geral mas quando se deseja um melhor desempenho
- Ideal também para processamento em lote

# Otimizadas para memória

- Projeto para alto desempenho no processamento de grandes quantidades de informações na memória
- Exemplo: Bancos de dados de alto desempenho, processamento em tempo real de dados



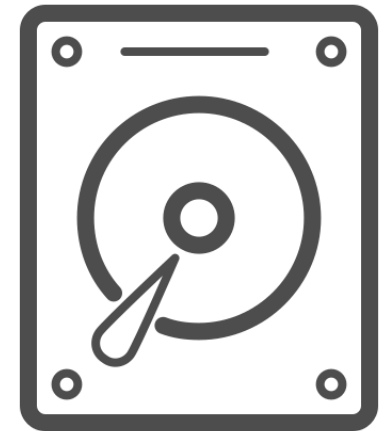
# Computação acelerada

- Usa aceleração de hardware ou coprocessadores para executar alguns funções de forma mais eficiente do que em um software executado direto na CPU
- Exemplos de casos de uso: Cálculo de ponto flutuante, processamento de gráficos e correspondência de padrões de dados.



# Otimizadas para armazenamento

- Ideal para cargas de trabalho que exigem acesso de leitura e gravação com grande volume de dados.
- Casos de uso: Sistemas de arquivos distribuídos, Data warehouse, sistema de processamento de transações on-line.



# Para saber mais

- <https://aws.amazon.com/pt/ec2/>
- [https://docs.aws.amazon.com/pt\\_br/AWSEC2/latest/UserGuide/concepts.html](https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/concepts.html)
- <https://aws.amazon.com/pt/ec2/instance-types/>



# Dúvidas?

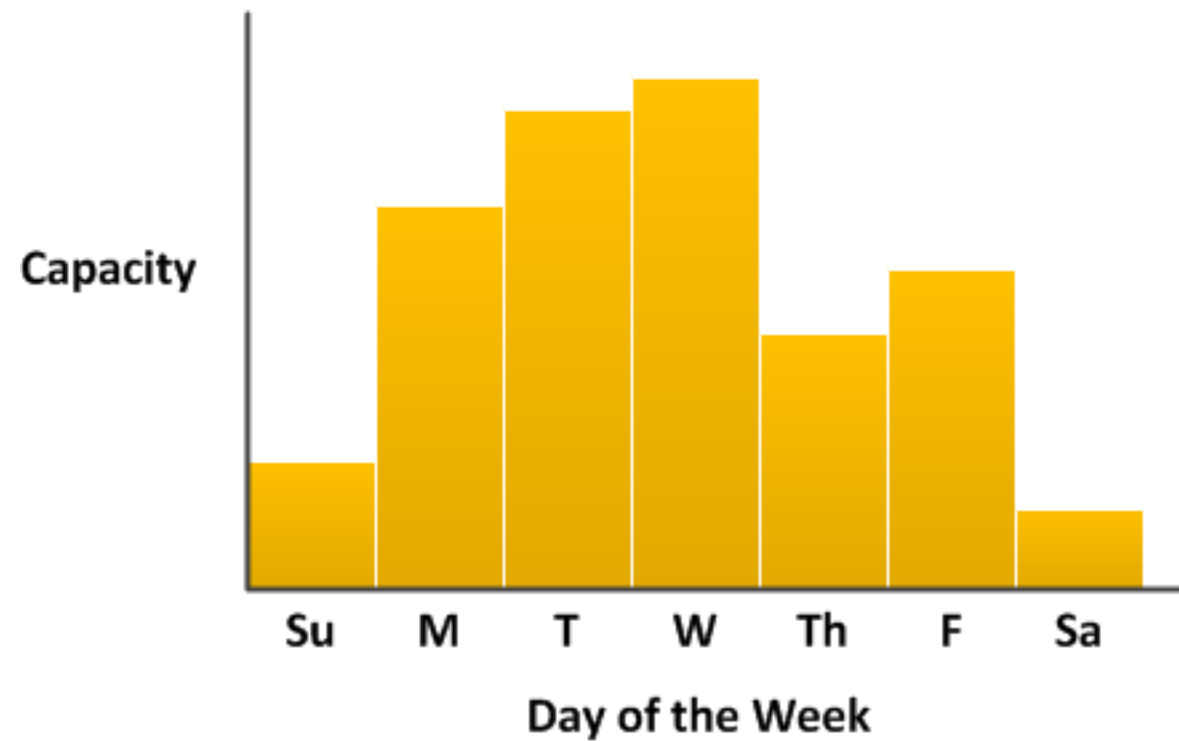
- > Fórum/Artigos
- > Comunidade Online (Discord)



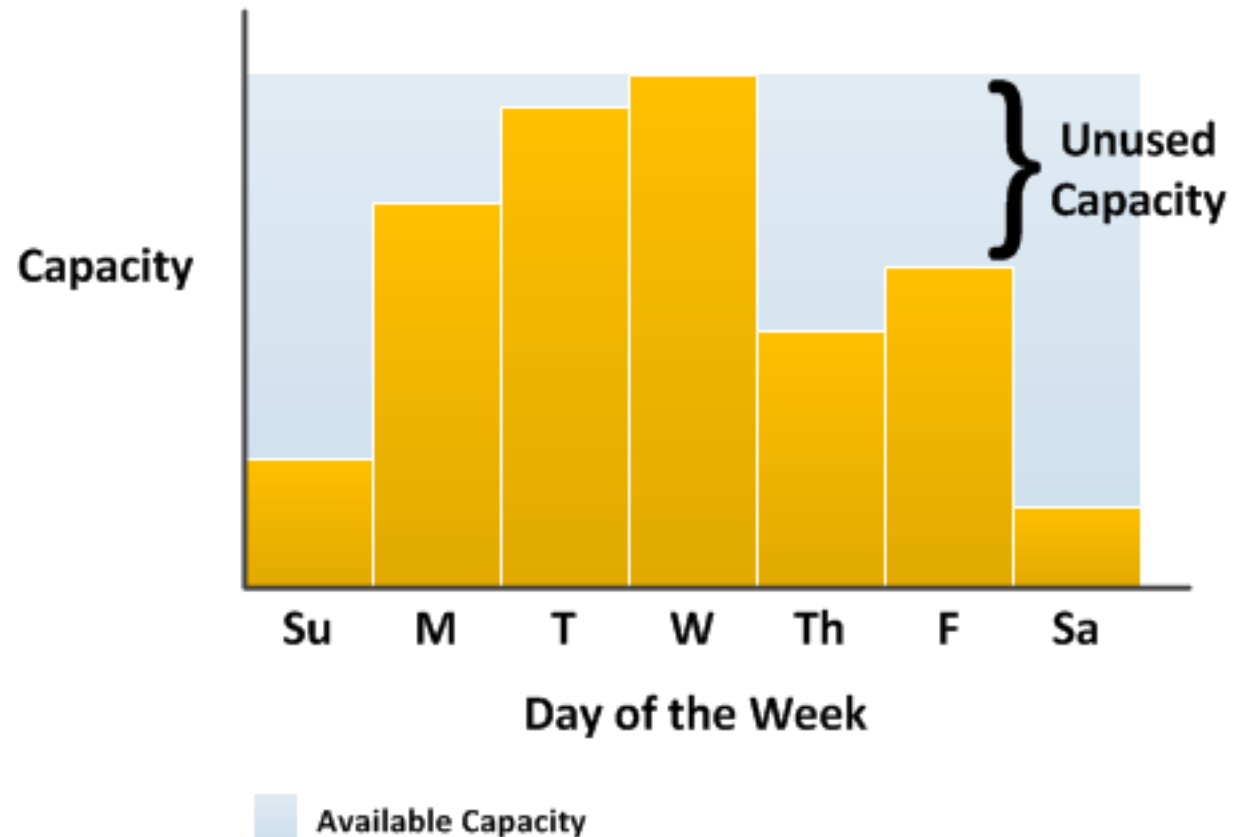
Etapa 2

# Amazon EC2 AutoScaling

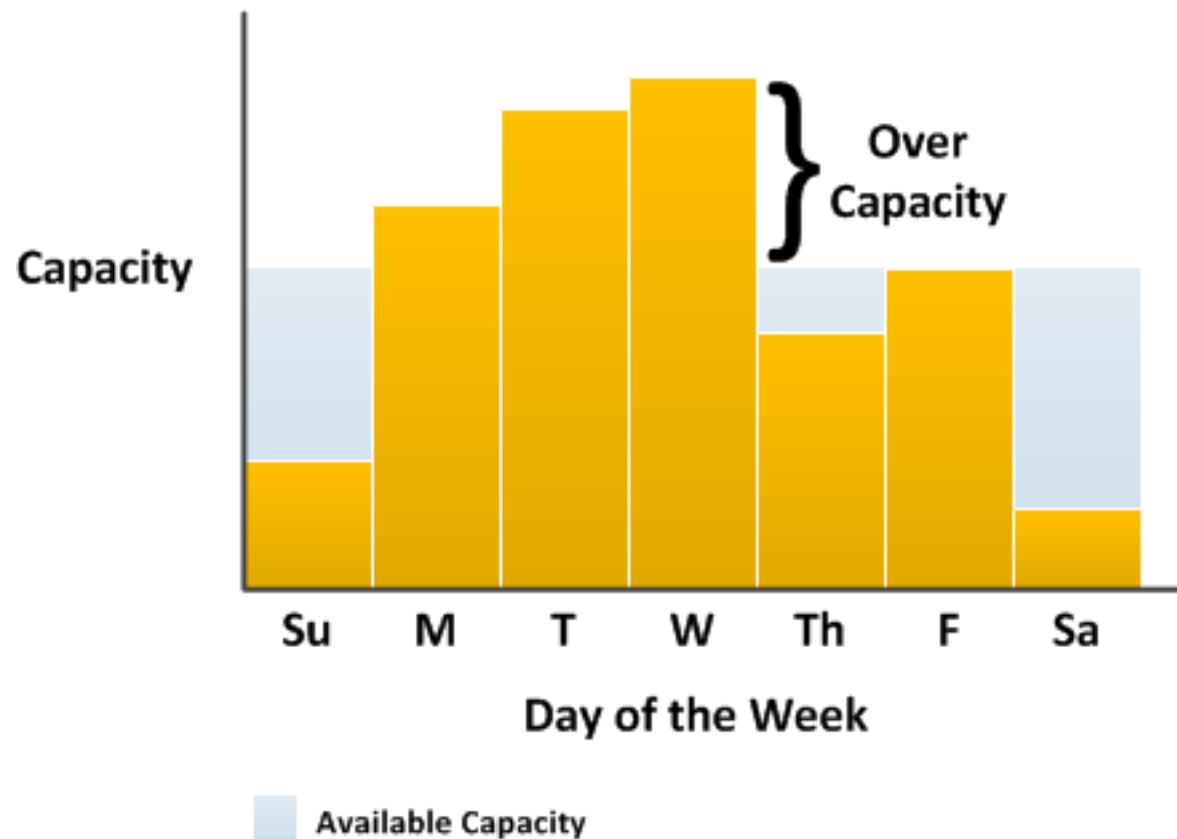
# Um cenário com EC2



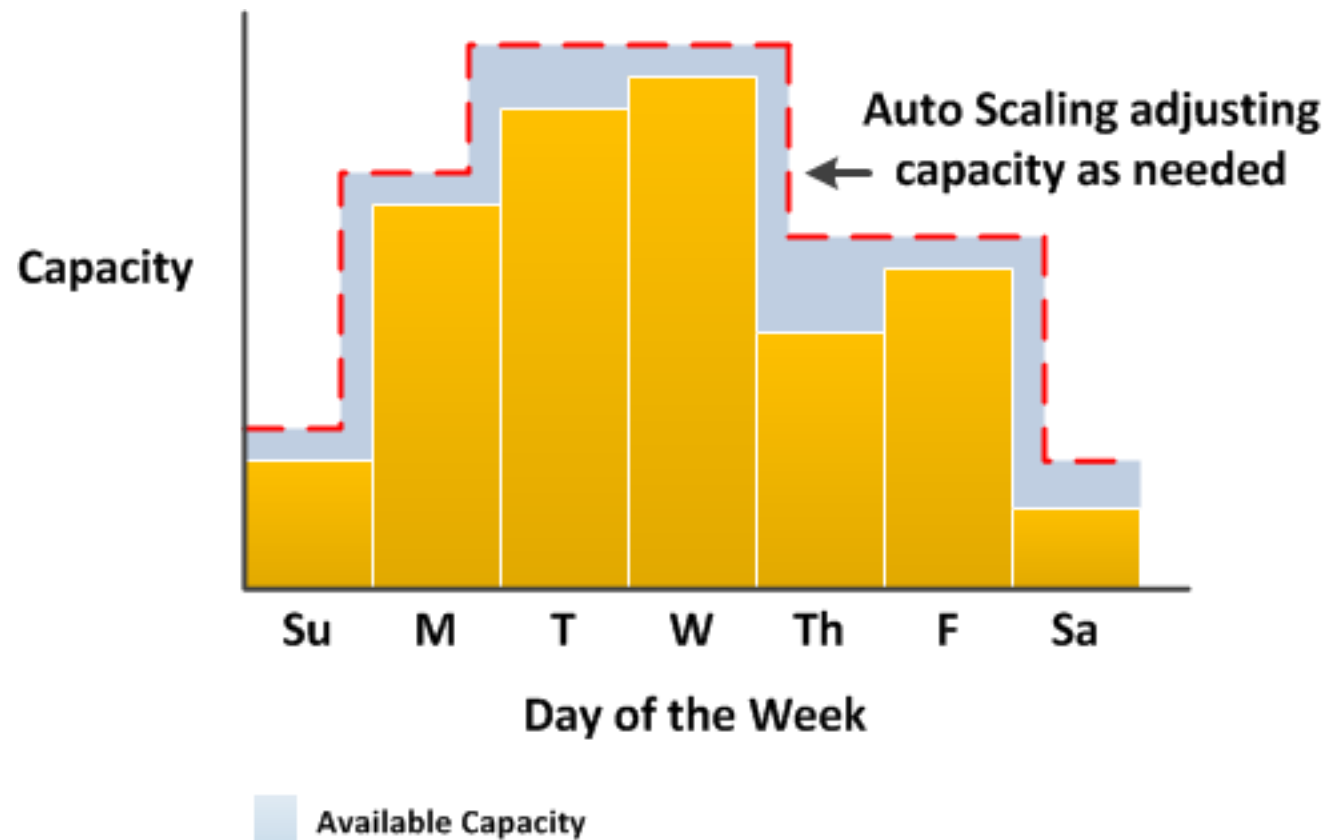
# Escalar para capacidade total



# Escalar para capacidade média



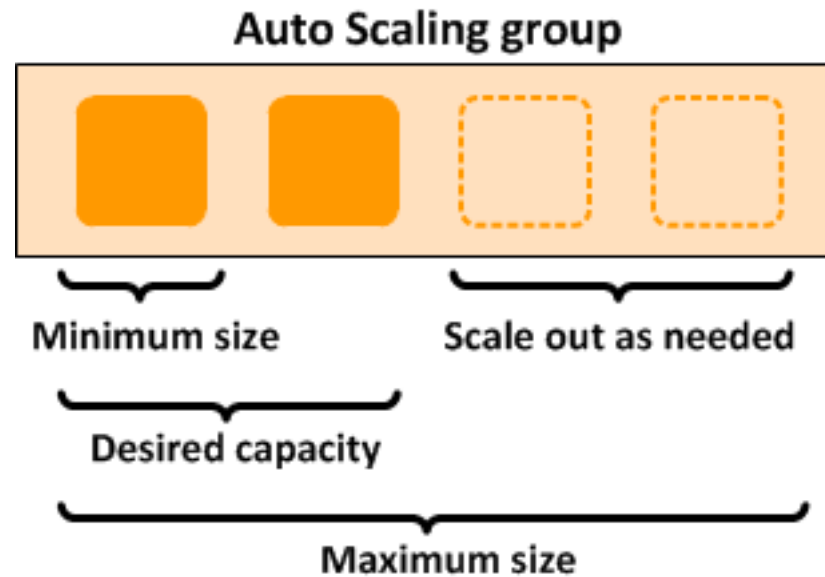
# Escalar conforme necessidade



# Amazon EC2 Auto Scaling

- Provê escalabilidade horizontal para seus serviços
- Melhora a tolerância a falhas com identificação de instâncias indisponíveis e implantação multi-AZ
- Melhor gerenciamento de custos

# Como é a configuração





# Abordagem

- Scaling Preditivo
- Scaling Dinâmico
- É possível combinar os dois

# Para saber mais

- <https://aws.amazon.com/pt/ec2/autoscaling/>
- <https://pt.wikipedia.org/wiki/Escalabilidade>
- [https://docs.aws.amazon.com/pt\\_br/AWSEC2/latest/UserGuide/concepts.html](https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/concepts.html)
- [https://docs.aws.amazon.com/pt\\_br/autoscaling/ec2/userguide/ec2-auto-scaling-predictive-scaling.html](https://docs.aws.amazon.com/pt_br/autoscaling/ec2/userguide/ec2-auto-scaling-predictive-scaling.html)
- [https://docs.aws.amazon.com/pt\\_br/autoscaling/ec2/userguide/as-scale-based-on-demand.html](https://docs.aws.amazon.com/pt_br/autoscaling/ec2/userguide/as-scale-based-on-demand.html)

# Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



## Etapa 3

# Elastic Load Balancing - ELB

# ELB – Elastic Load Balancing

- Balanceamento de carga de aplicação, gateway e rede
- Escopo regional
- Escala de forma automática, sem custos
- Junto ao EC2 AutoScaling permite criar aplicações altamente disponíveis

# Para saber mais

- <https://aws.amazon.com/pt/elasticloadbalancing/?nc=sn&loc=0>
- [https://docs.aws.amazon.com/pt\\_br/elasticloadbalancing/latest/application/introduction.html](https://docs.aws.amazon.com/pt_br/elasticloadbalancing/latest/application/introduction.html)

# Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



## Etapa 4

# Serviços de mensageria



# Amazon Simple Queue Service

## – SQS

- Sistema de enfileiramento de mensagens
- Um usuário envia uma mensagem para fila, o outro usuário lê, processa e a exclui da fila

# Amazon Simple Notification Service – SNS

- Sistema pub/sub
- Utiliza tópicos como estrutura
- Usuário publica mensagens no tópico e assinantes escutam

# Para saber mais

- <https://aws.amazon.com/pt/messaging/>
- <https://aws.amazon.com/pt/sqs/>
- <https://aws.amazon.com/pt/sns/>

# Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



## Etapa 5

# Computação sem servidor

# O que é computação sem servidor ?

- Também chamado pelo termo “Serverless”
- O termo “sem servidor” significa que o código é executado em servidores sem que você precise provisionar ou gerenciar esses servidores
- Capacidade automaticamente ajustada pelo serviço, sem necessidade de nenhuma configuração

# AWS Lambda

- Execução de código sem provisionar servidores
- Código organizado em funções
- Você pode escolher a linguagem de programação de sua preferência
- Executa a partir de eventos ou chamadas diretas a API do Lambda



## How Lambda works



Upload your code to AWS Lambda or write code in Lambda's code editor



Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity



Lambda runs your code only when triggered, using only the compute resources needed



Just pay for the compute time you use



# Para saber mais

- <https://aws.amazon.com/pt/lambda/getting-started/>
- [https://docs.aws.amazon.com/pt\\_br/lambda/?id=docs\\_gateway](https://docs.aws.amazon.com/pt_br/lambda/?id=docs_gateway)
- [https://docs.aws.amazon.com/pt\\_br/lambda/latest/dg/welcome.html](https://docs.aws.amazon.com/pt_br/lambda/latest/dg/welcome.html)

# Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



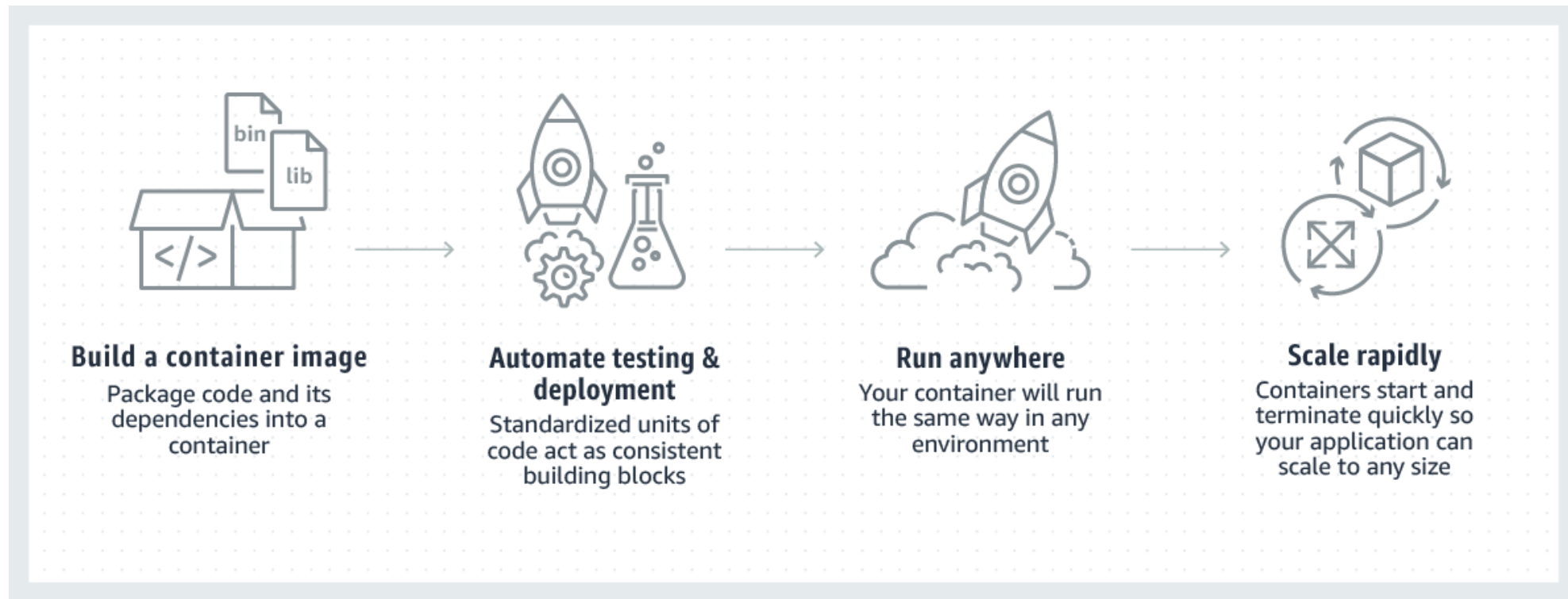
Etapa 6

# Containers em AWS

# Containers

- Forma padrão de empacotar seu aplicativo em um único objeto
- Executados como processos isolados
- Docker

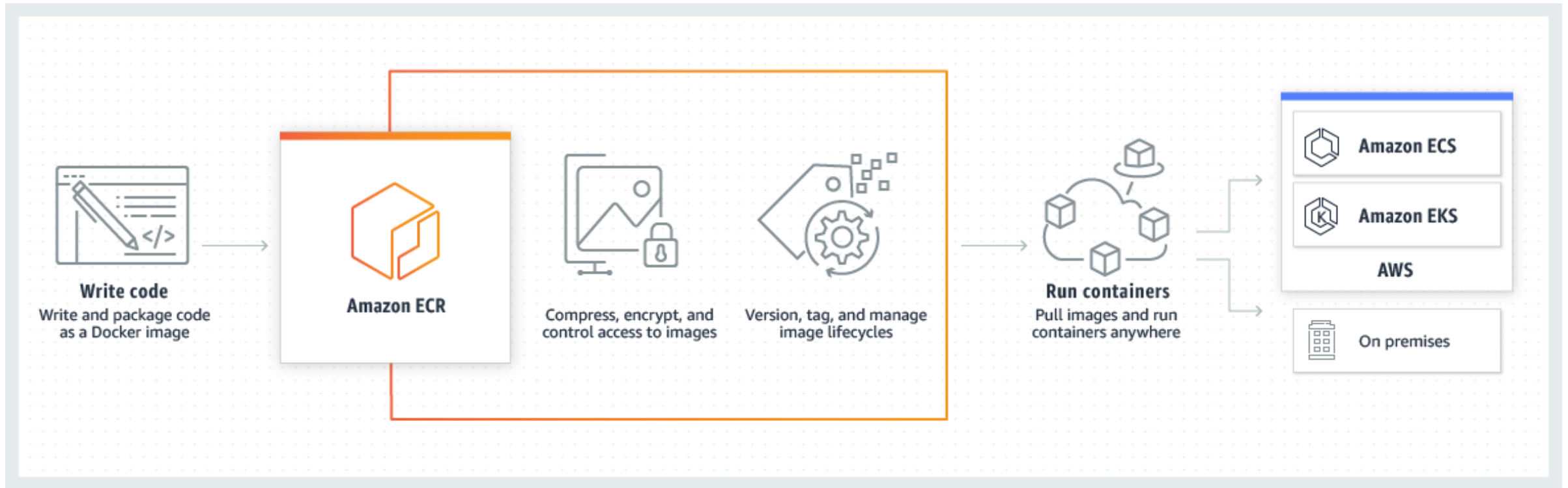
# Containers



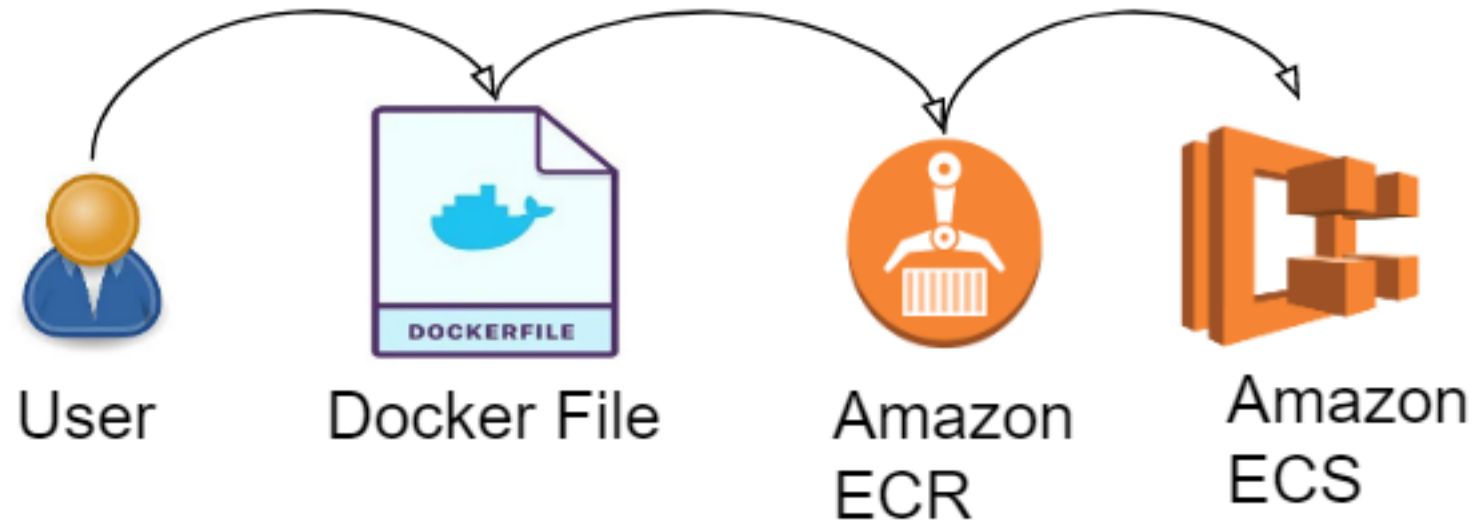
# Serviços AWS

- ECR – Elastic Container Registry
- ECS – Elastic Container Service
- EKS – Elastic Kubernetes Service
- AWS Fargate

# ECR – Elastic Container Registry



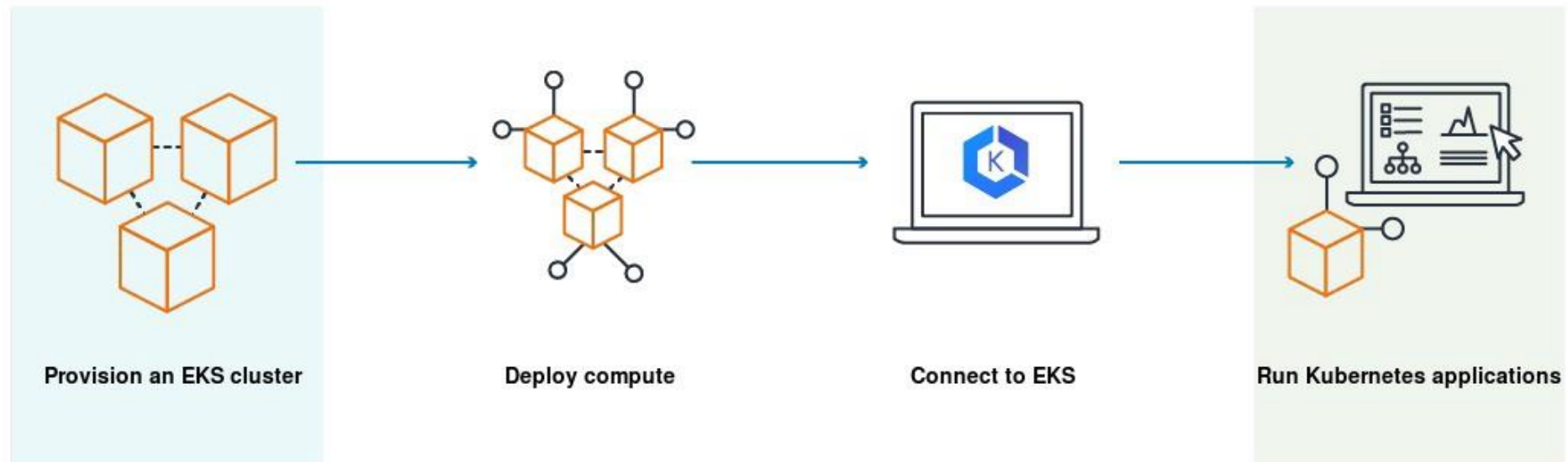
# ECS – Elastic Container Service





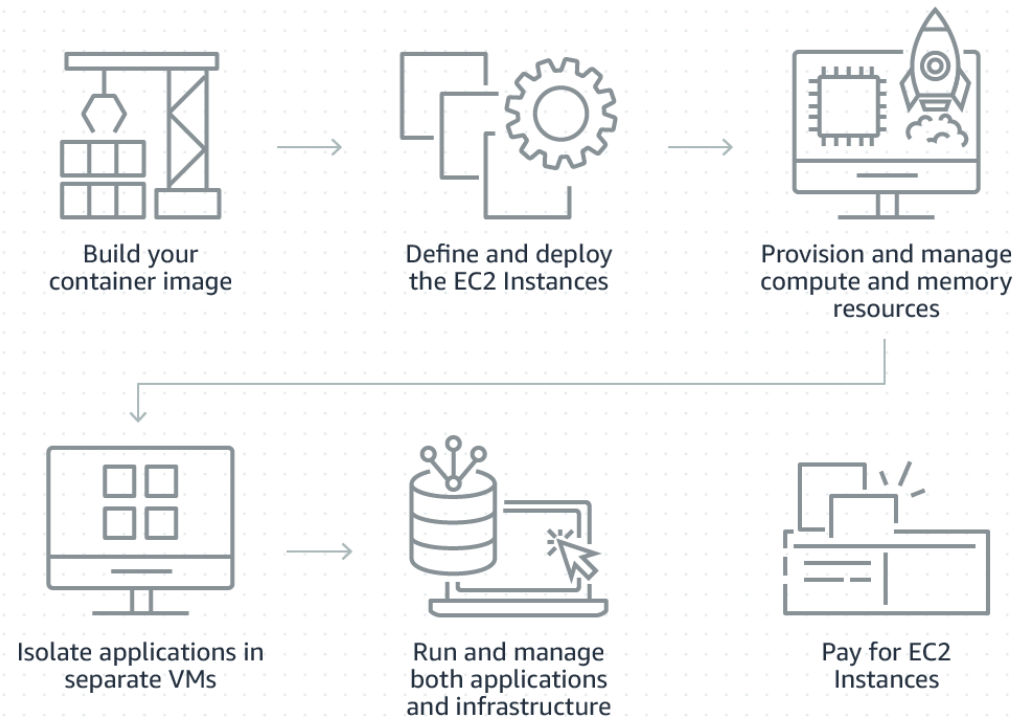
# EKS – Elastic Kubernetes Service

How does Amazon EKS work?

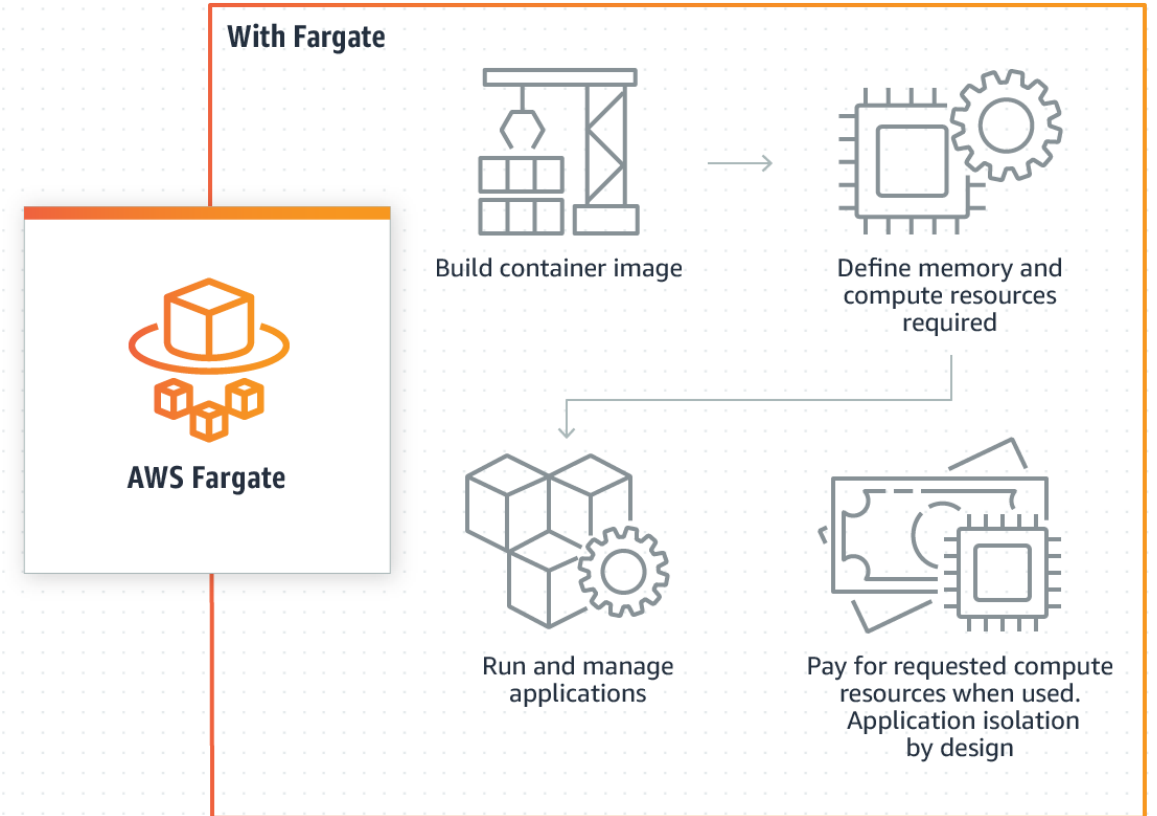


# AWS Fargate

## Without Fargate



## With Fargate



# Para saber mais

- <https://aws.amazon.com/pt/containers/>
- <https://aws.amazon.com/pt/ecr/>
- <https://aws.amazon.com/pt/ecs/>
- <https://aws.amazon.com/pt/eks/>

# Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)

