Microservices & Kubernetes em um ambiente de alta demanda

iFood em números

3
países

500+
cidades

100%
crescimento

2000+
pedidos/minuto

17.000.000+
pedidos/mês

iFood em números (nerds)

1.000+ instâncias
no horário de pico

140+ serviços

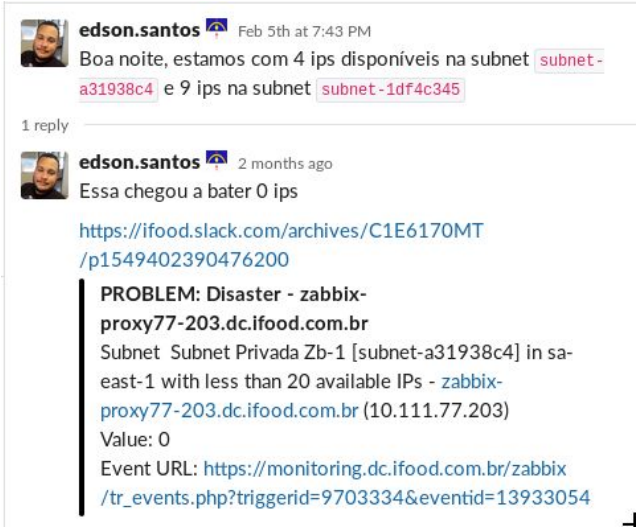
1.8M de requests
durante load test

500+ deployments/mês

3 trilhões de eventos/mês

Dores do crescimento

IPs na conta AWS (SP) acabando



Dores do crescimento

Throttling do SNS

com.amazonaws.services.sns.model.AmazonSNSException: Rate exceeded (Service: AmazonSNS; Status Code: 400; Error Code: Throttling; Request ID: 75d4b748-27dd-5851-b686-e255e704e768)

Dores do crescimento

Throttling do Lambda

Limite de execução simultânea no nível da conta

Por padrão, o AWS Lambda limita o total de execuções simultâneas entre todas as funções dentro de uma determinada região a 1000. Você pode visualizar a configuração do nível da conta usando a API GetAccountSettings e exibindo o objeto `AccountLimit`. Esse limite pode ser aumentado conforme descrito abaixo:

Para solicitar um aumento de limite para execuções simultâneas

1. Abra a página AWS Support Center, faça login se necessário e escolha **Create case (Criar caso)**.
2. Em **Referente**, selecione **Aumento de Service Limit**.
3. Em **Limit Type (Tipo de limite)**, escolha **Lambda**, preencha os campos necessários no formulário e escolha o botão na parte inferior da página para seu método preferido de contato.

Como subimos serviços?

HashiCorp
Terraform

+

Desafios

Diminuir tempo de deploy

Escalar mais rápido

Otimizar recursos

kubernetes

+

Voluntários?

Serviço novo

Potencial de grande volume de dados

Account Metadata

Armazena diferentes perspectivas do cliente

Flexível de acordo com as necessidades de cada área do iFood

Tabelão

amazon
DynamoDB

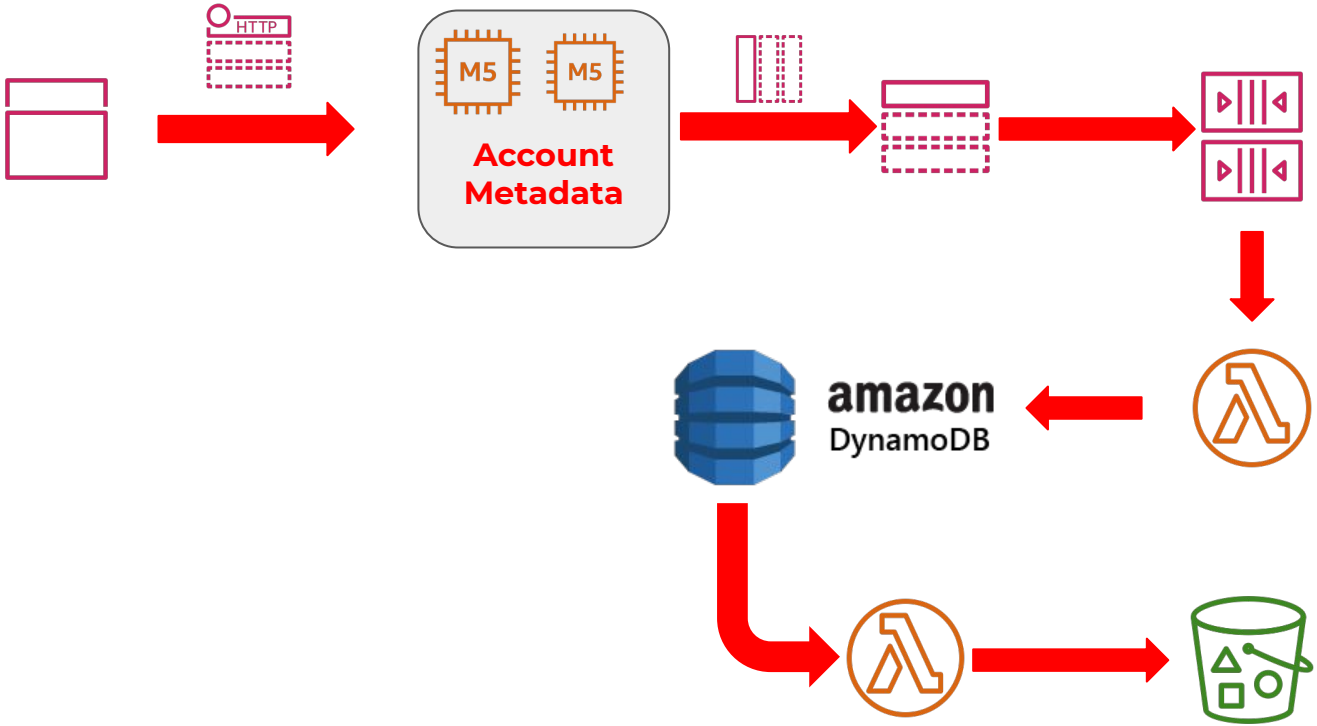
account_id	namespace	frequency	last_order	last_login	contact_type	tags
7db77c3e-519e-497d-99e4-efa7a03eddb0	public-data	>10	2019-04-07T19:22:57.057Z			
7db77c3e-519e-497d-99e4-efa7a03eddb0	public-tags					[LANCHEIRO, PIZZEIRO, WEEKEND]
7db77c3e-519e-497d-99e4-efa7a03eddb0	public-preferences				[EMAIL, PHONE]	
7db77c3e-519e-497d-99e4-efa7a03eddb0	private-security			2019-03-12T18:00:00.057Z		

Configuração

Garantir que apps não quebrem a cada mudança

namespace	schema
public-data	{"definitions":{},"$schema":"http://json-schema.org/draft-07/schema#","$id":"http://example.com/root.json","type":"object","title":"The Data public","properties":{"status":{"$id":"#/properties/status","type":"string","title":"The Status…….

Ingestão de dados



Exposição dados públicos

Como escalar em K8s?

Horizontal Pod Autoscaler

CPU

Memória

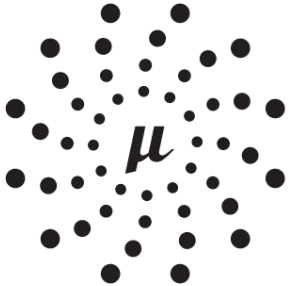
Número de requisições

Como escalar em K8s?

```
apiVersion: v1
items:
  - apiVersion: autoscaling/v1
    kind: HorizontalPodAutoscaler
    metadata:
    name: account-metadata
    namespace: account-metadata
    spec:
    maxReplicas: 20
    minReplicas: 5
    scaleTargetRef:
    apiVersion: extensions/v1beta1
    kind: Deployment
    name: account-metadata
    targetCPUUtilizationPercentage: 50
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: account-metadata
  namespace: account-metadata
spec:
        resources:
          requests:
                memory: "1024Mi"
                cpu: 1000m
```

E o serviço?



MICRONAUT™

Funciona com Java, Groovy, Kotlin

Injeção de Dependência em tempo de compilação

Netty como servidor HTTP

Fácil de aprender?

Sim, curva de aprendizado suave

```
import static io.micronaut.runtime.Micronaut.run;

public class Application {

    public static void main(String... args) {
        run(Application.class, args);
    }
}
```

Controller Reativo

Funciona com CompletableFuture, Reactor & RxJava

```
@Validated
@Controller
public class TagController {

@Inject
private TagService tagService;

@Post("/accounts/{accountId}/tags")
public CompletableFuture<HttpResponse> process(UUID accountId, @Body
@Valid TagRequest tagRequest) {
  return tagService.process(accountId, tagRequest)
        .thenApplyAsync(response -> created(response));
}
```

Controller

```
@Post(value = "/namespaces")
@Status(OK)
public void create(@Body @Valid NamespaceCreateRequest namespace) {
  namespaceService.create(namespace.getNamespace(), namespace.getSchema());
}
```

Definição Bean

```
@Singleton
class TagService {

@Inject
private TagDAO tagDAO;

……..

}
```

Demo

Melhorando startup - 1

Gerar lista de classes usada no boot da aplicação:

```
-XX:+UseAppCDS -XX:DumpLoadedClassList=classes.lst
```

Melhorando startup - 2

Gerar arquivo preparado para boot:

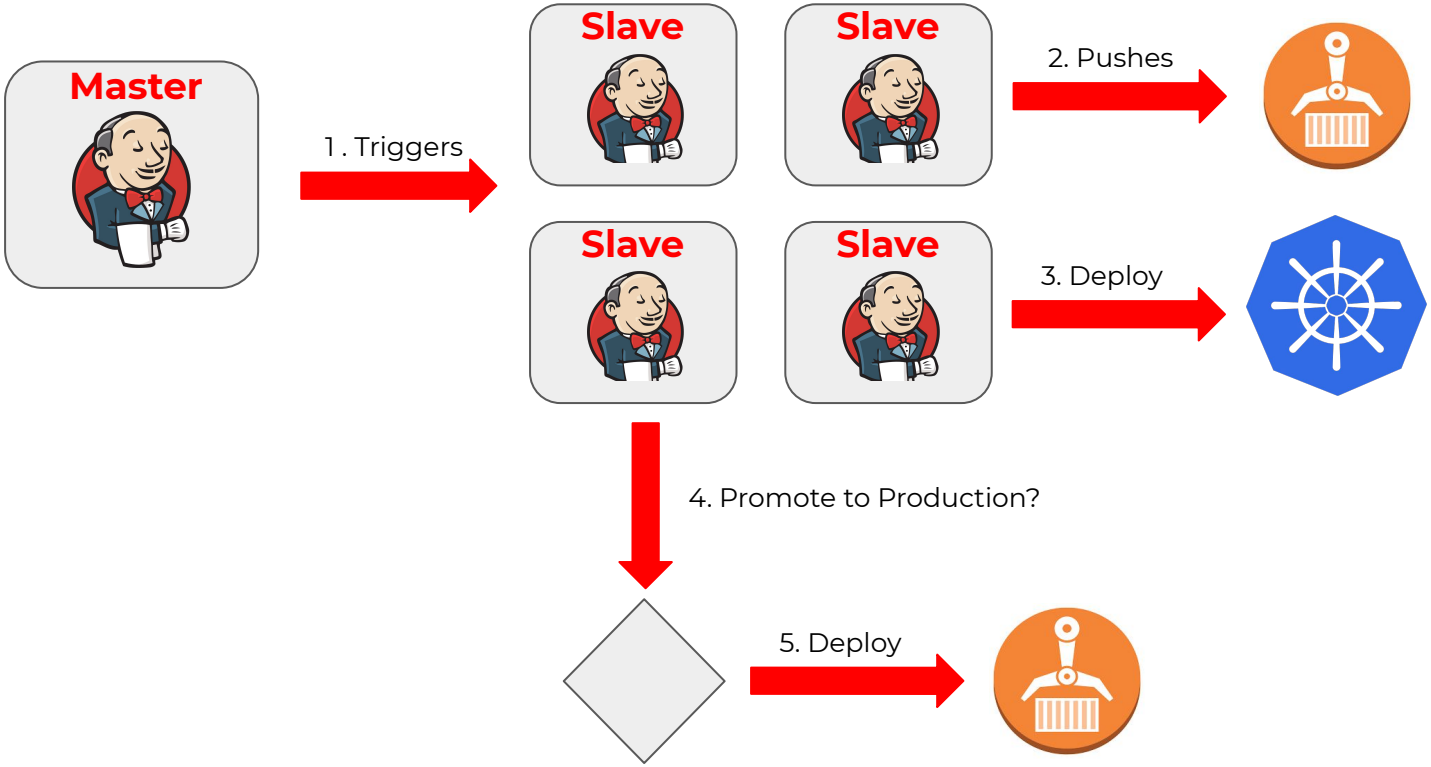
```
-Xshare:dump -XX:+UseAppCDS -XX:MetaspaceSize=128m
-XX:SharedClassListFile=classes.lst
-XX:SharedArchiveFile=app-cds.jsa
```

Melhorando startup - 3

Execução:

```
java -Xshare:on -XX:SharedArchiveFile=app-cds.jsa -jar
helloworld.jar
```

Jenkins pipeline



Build via Docker

```
FROM gradle:jdk11 as builder

COPY --chown=gradle:gradle . /home/gradle/src

WORKDIR /home/gradle/src

RUN gradle clean build

FROM openjdk:11-jre-slim-sid

EXPOSE 8080

COPY --from=builder /home/gradle/src/build/libs/*.jar account-metadata.jar

RUN /usr/bin/java -XX:+UseAppCDS -XX:DumpLoadedClassList=classes.lst -jar account-metadata.jar &
sleep 5 && exit

RUN /usr/bin/java -Xshare:dump -XX:+UseAppCDS -XX:SharedClassListFile=classes.lst
-XX:SharedArchiveFile=app-cds.jsa --class-path account-metadata.jar

ENTRYPOINT /usr/bin/java -Xshare:on -XX:SharedArchiveFile=app-cds.jsa -jar "account-metadata.jar"
```

O iFood é hoje o melhor lugar para se trabalhar com dev porque...


ifood