

THIAGO ADRIANO

POSTECH

SOFTWARE ARCHITECTURE

KUBERNETES

AULA 05

SUMÁRIO

O QUE VEM POR AÍ?	3
HANDS ON.....	4
SAIBA MAIS	5
O QUE VOCÊ VIU NESTA AULA?	10
REFERÊNCIAS	11

O QUE VEM POR AÍ?

Nesta aula, você vai aprender a como dar os primeiros passos com escalabilidade utilizando as ReplicaSets e o Deployment do Kubernetes. Por meio de alguns exemplos práticos, você vai firmar o conhecimento sobre este assunto e ficar pronto(a) para utilizá-lo em seu dia-a-dia.

EMAND

HANDS ON

Na primeira videoaula, os professores apresentaram o ReplicaSets por meio de exemplos práticos, utilizando um projeto desenvolvido em aula.

Na videoaula seguinte, os docentes apresentaram o Deployment por meio de exemplos práticos, demonstrando a diferença entre Deployment e Replicasets.

EMANIP

SAIBA MAIS

REPLICASETS

Conforme aprendemos em aulas anteriores, o Kubernetes é uma plataforma de orquestração de containers que permite gerenciar e escalar aplicativos distribuídos.

Nesta aula, aprenderemos a como garantir a disponibilidade e a confiabilidade do aplicativo, utilizando ReplicaSets e Deployments.

ReplicaSets é um objeto do Kubernetes que garante que um número específico de réplicas de um conjunto de Pods esteja em execução a qualquer momento. Ele monitora o status das réplicas de um Pod e, se alguma réplica falhar, inicia uma nova para substituí-la.

Temos a figura 1 – “Exemplo de cluster com Nodes parte 1”, ilustrando como funciona o fluxo de replicação de containers com o ReplicaSet. Note que, nesta imagem, nós temos dois containers funcionando corretamente e um com crash, e em ReplicaSet temos o número 2, que demonstra que precisamos de dois containers funcionando.

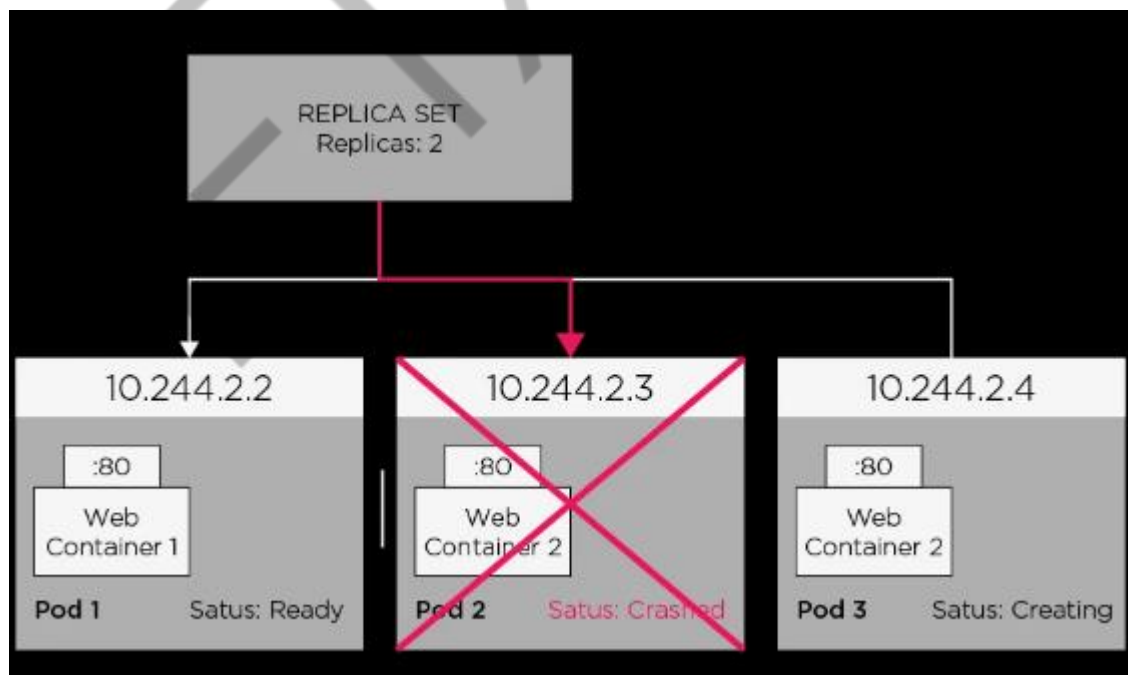


Figura 1 – Exemplo de cluster com nodes - parte 1

Fonte: <https://5i4i.short.gy/NDNuC2> (2019), adaptado por FIAP (2023)

A ideia da figura 1 – “Exemplo de cluster com nodes – parte 1”, seria ilustrar que tivemos um container que ficou down (caiu), e logo em seguida tivemos um novo tomando o seu lugar, para que o fluxo continue com o número de réplicas configurados, conforme a figura 2 – “Exemplo de cluster com nodes – parte 2”.

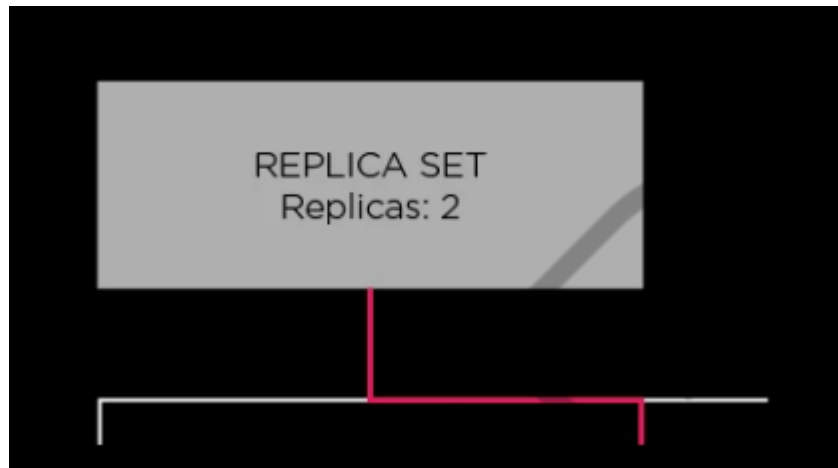


Figura 2 – Exemplo de Cluster com nodes - parte 2
 Fonte: <https://5i4i.short.gy/NDNuC2> (2019)

Conforme vimos no exemplo anterior, os ReplicaSets são responsáveis por manter a integridade do estado do aplicativo, garantindo que haja um número mínimo de réplicas em execução em todos os momentos. Eles são uma extensão do conceito de Deployment que também aprenderemos nesta aula, e que permitem gerenciar o número de réplicas de um conjunto de Pods e atualizar o aplicativo para uma nova versão.

Os ReplicaSets também são responsáveis pela seleção dos Pods que fazem parte do conjunto. Eles usam um seletor de rótulo (que aprendemos na aula anterior), para definir quais Pods pertencem ao conjunto.

Por exemplo, um ReplicaSet pode selecionar todos os Pods com o rótulo "app = nginx", para criar um conjunto de réplicas do servidor web nginx.

Para melhor entendimento, observe a figura 3 – “Exemplo de arquivo ReplicaSet”, que é um exemplo do arquivo de criação de ReplicaSets:

```
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: nginx-fiap
5  spec:
6    template:
7      metadata:
8        name: nginx-app
9        labels:
10         app: nginx-app
11      spec:
12        containers:
13         - name: nginx-container-fiap
14           image: nginx:latest
15     replicas: 3
16     selector:
17       matchLabels:
18         app: nginx-app
```

Figura 3 – Exemplo de arquivo ReplicaSet
Fonte: Elaborado pelo autor (2023)

Analisando este arquivo, note que nós estamos criando um ReplicaSet chamado "nginx-fiap" com 3 réplicas, utilizando o seletor "nginx-app" para a criação dos nossos containers.

Deployments

Um Deployment é um objeto Kubernetes que gerencia a implantação de um conjunto de réplicas de um aplicativo. Ele fornece recursos para gerenciar a atualização de aplicativos, rollbacks e dimensionamento horizontal.

O Deployment é responsável por garantir que um conjunto específico de réplicas esteja em execução em um determinado momento.

O número de réplicas é especificado pelo usuário, e o Kubernetes monitora constantemente o estado dessas réplicas. Se uma réplica falhar ou ficar indisponível, o Kubernetes iniciará automaticamente uma nova réplica para substituí-la.

Além disso, o Deployment permite que os usuários atualizem o aplicativo de forma controlada e automática. Quando uma atualização é implantada, o Kubernetes cria um novo conjunto de réplicas com a nova versão do aplicativo e, em seguida, gradualmente substitui as réplicas antigas pelas novas. Esse processo é chamado de “Rolling Update”, e permite que os usuários atualizem o aplicativo sem interromper o serviço.

Se ocorrer algum problema durante uma atualização, o Kubernetes oferece a possibilidade de reverter para a versão anterior do aplicativo. Isso é chamado de rollback e é uma funcionalidade importante para garantir a disponibilidade contínua do serviço.

Outra funcionalidade importante do Deployment é o dimensionamento horizontal. Isso permite que os usuários aumentem ou diminuam o número de réplicas do aplicativo com base na demanda do serviço. Isso é particularmente útil para aplicativos que experimentam flutuações na carga de trabalho.

Na figura 4 – “Exemplo de arquivo Deployment”, temos um exemplo de arquivo YAML criado para definir um deployment.


```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: nginx-fiap
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        app: nginx-app
10   template:
11     metadata:
12       labels:
13         app: nginx-app
14     spec:
15       containers:
16       - name: nginx-container-fiap
17         image: nginx:latest
```

Figura 4 – Exemplo de arquivo Deployment
Fonte: Elaborado pelo autor (2023)

Neste arquivo de exemplo, nós estamos criando um Deployment chamado "nginx-fiap" com 3 réplicas. Este Deployment está usando um seletor para selecionar os Pods que devem ser gerenciados por ele, neste caso, todos os Pods com o rótulo "nginx-app".

Note que o arquivo de Deployment não tem muita diferença de um ReplicaSet, mas no dia-a-dia, acabamos utilizando mais o Deployment, por conta do seu controle de versão.

O QUE VOCÊ VIU NESTA AULA?

Nesta aula, aprendemos como escalar nossos aplicativos utilizando dois recursos do Kubernetes na prática: os ReplicaSets e os Deployments. Para testar o fluxo de escala, os professores apresentaram também uma forma de se fazer um teste de carga utilizando o K6.

O que achou do conteúdo? Conte-nos no Discord! Estamos disponíveis na comunidade para tirar dúvidas, fazer networking, enviar avisos e muito mais.

REFERÊNCIAS

DOBIES, J. **Operadores do Kubernetes: Automatizando a Plataforma de Orquestração de Contêineres**. [s.l.]: Novatec, 2020.

EMEND

PALAVRAS-CHAVE

Deployments. Kubernetes. ReplicaSets.

EMEND



POSTECH