

Série de Certificação GCP: 4.2

Gerenciando recursos do Kubernetes Engine



Prashanta Paudel

7 de novembro de 2018 · 5 minutos de leitura

Já aprendemos funções introdutórias e básicas no tópico anterior, de modo que iremos diretamente fazer as tarefas mencionadas no plano de estudos.

Visualizando o inventário atual do cluster em execução (nós, pods, serviços)

Depois que os clusters do kubernetes estiverem ativos e em execução, você poderá verificar o status dos nós, pods e serviços.

Primeiro vamos fazer dois clusters na Europa norte 1 e asia east1, ambos agrupamentos regionais.

```
prashantagcppaudel @ cloudshell: ~ (fourpointtwo-221807)
clusters de contêiner $ gcloud criam cluster Ásia-leste -
tipo-maquina = f1-micro --num-nodes = 2 --region = asia-
east1
AVISO: A partir de 1.12, novo os clusters terão a
autenticação básica desabilitada por padrão. A autenticação
básica pode ser ativada (ou desativada) manualmente usando o
sinalizador `-[no-] enable-basic-auth`.
AVISO: A partir de 1.12, novos clusters não terão um
certificado de cliente emitido. Você pode ativar (ou
desativar) manualmente a emissão do certificado de cliente
usando o sinalizador `-[no-] issue-client-certificate`.
AVISO: Atualmente, o VPC-native não é o modo padrão durante
a criação do cluster. No futuro, isso se tornará o modo
padrão e poderá ser desativado usando o sinalizador `--no-
enable-ip-alias`. Use o sinalizador `-[no-] enable-ip-
alias` para suprimir este aviso.
AVISO: a partir de 1.12, os pools de nós padrão em novos
clusters terão seus terminais de metadados da instância
legada do Compute Engine desativados por padrão. Para criar
um cluster com pontos de extremidade de metadados da
instância legada desabilitados no conjunto de nós padrão,
execute `clusters create` com o sinalizador - metadados
disable-legacy-endpoints = true`.
Isso ativará o recurso de acompanhamento automático para
```

nós. Consulte <https://cloud.google.com/kubernetes-engine/docs/node-auto-repair> para mais informações sobre os autorepairs do nó.

ATENÇÃO: A partir do Kubernetes v1.10, os novos clusters não mais terão os escopos compute-rw e storage-ro adicionados ao especificado em escopos (embora o último permaneça incluído nos escopos padrão). Para usar esses escopos, adicione-os explicitamente a --scopes. Para usar o novo comportamento, defina a propriedade container / new_scopes_behavior (gcloud config set container / new_scopes_behavior true).

Criando cluster asia-east-cluster em asia-east1 ... pronto.
Criado

[<https://container.googleapis.com/v1/projects/fourpointtwo-221807/zones/asia-east1/clusters/asia-east-cluster>].

Para inspecionar o conteúdo de seu cluster, acesse:
https://console.cloud.google.com/kubernetes/workload_/gcloud/asia-east1/asia-east-cluster?project=fourpointtwo-221807

entrada kubeconfig gerada para o asia- cluster leste.

NOME	LOCALIZAÇÃO	MASTER_VERSION	MASTER_IP	MACHINE_TYPE	NODE_VERSION	NUM_NODES	STATUS
asia-east-cluster	asia-east1	1.9.7-gke.6	35.221.167.78	f1-micro	1.9.7-gke.6	6	RUNNING

```
prashantagcppaudel @ cloudshell: ~ (fourpointtwo-221807) Os
clusters de contêineres $ gcloud criam europe-cluster --
machine-type = n1-standard-1 --num-nodes = 2 --region =
europe-north1
```

AVISO: A partir de 1.12, novo os clusters terão a autenticação básica desabilitada por padrão. A autenticação básica pode ser ativada (ou desativada) manualmente usando o sinalizador --enable-basic-auth.

AVISO: A partir de 1.12, novos clusters não terão um certificado de cliente emitido. Você pode ativar (ou desativar) manualmente a emissão do certificado de cliente usando o sinalizador --issue-client-certificate.

AVISO: Atualmente, o VPC-native não é o modo padrão durante a criação do cluster. No futuro, isso se tornará o modo padrão e poderá ser desativado usando o sinalizador --no-enable-ip-alias. Use o sinalizador --enable-ip-alias para suprimir este aviso.

AVISO: a partir de 1.12, os pools de nós padrão em novos clusters terão seus terminais de metadados da instância legada do Compute Engine desativados por padrão. Para criar um cluster com pontos de extremidade de metadados da instância legada desabilitados no conjunto de nós padrão, execute clusters create com o sinalizador --disable-legacy-endpoints = true.

Isso ativará o recurso de acompanhamento automático para nós. Consulte <https://cloud.google.com/kubernetes-engine/docs/node-auto-repair> para mais informações sobre os autorepairs do nó.

ATENÇÃO: A partir do Kubernetes v1.10, os novos clusters não mais terão os escopos compute-rw e storage-ro adicionados ao especificado em escopos (embora o último permaneça incluído nos escopos padrão). Para usar esses escopos, adicione-os explicitamente a --scopes. Para usar o novo comportamento, defina a propriedade container / new_scopes_behavior (gcloud config set container / new_scopes_behavior true).

Criando cluster europa-cluster na europa-north1 ... feito.
Criado

```
[https://container.googleapis.com/v1/projects/fourpointtwo-221807/zones/europe-north1/clusters/europe-cluster].
Para inspecionar o conteúdo do seu cluster, acesse:
https://console.cloud.google.com/kubernetes/workload_/gcloud/europe-north1/europe-cluster?project=fourpointtwo-221807
entrada kubeconfig gerada para o cluster Europa.
NOME LOCALIZAÇÃO MASTER_VERSION MASTER_IP MACHINE_TYPE
NODE_VERSION NUM_NODES STATUS
europe-cluster europa-norte1 1.9.7-gke.6 35.228.64.238 n1-standard-1 1.9.7-gke.6 6 RUNNING
```

Para listar os clusters em execução no Kubernetes

```
$ gcloud lista de clusters de contêineres
NOME LOCALIZAÇÃO MASTER_VERSION MASTER_IP MACHINE_TYPE
NODE_VERSION NUM_NODES STATUS
Ásia-leste-cluster Ásia-Leste1 1.9.7-gke.6 35.221.167.78 f1-micro 1.9.7-gke.6 6 RUNNING
europe-cluster europa-norte1 1.9.7-gke.6 35.228.64.238 n1-standard-1 1.9.7-gke.6 6 RUNNING
```

Agora adicione algumas cargas de trabalho a este cluster

Aplicativo de execução \$ kubectl - image gcr.io/google-samples/hello-app:1.0

Para ver os pods

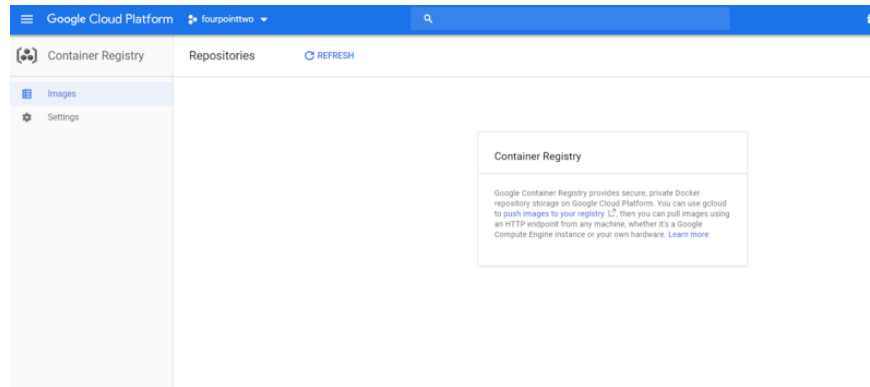
#kubectl obter pods

```
gcloud compute instances list

prashantagcppaudel @ cloudshell: ~ (fourpointtwo-221807) $
kubectl obter implementação
NOME DESEJADO ATUAL ATUALIZADO DISPONÍVEL IDA
frontend 3 3 3 3 2h
redis-master 1 1 1 1 2h
redis-escravo 2 2 2 2 2h
prashantagcppaudel @ cloudshell: ~ (fourpointtwo-221807) $
```

Navegando no Registro de Imagens do Contêiner

Primeiro, vá para o Console e, em seguida, para o Container Registry, onde você pode ver as imagens do contêiner.



Registro de Contêiner

Trabalhando com os nós

Você adiciona mais réplicas ao recurso de implantação do seu aplicativo usando o `kubectl scale` comando. Para adicionar duas réplicas adicionais à sua Implantação (para um total de três), execute o seguinte comando:

```
implementação em escala kubectl hello-web --replicas = 3
```

Você pode ver as novas réplicas em execução no seu cluster executando os seguintes comandos:

```
kubectl obter implantação hello-web
```

Saída:

```
NOME DESEJADO ATUAL ACIMA DISPONÍVEL AGE  
hello-web 3 3 3 2 1m
```

```
kubectl obter pods
```

Saída:

```
NOME READY STATUS REESTARES Idade
hello-web-4017757401-ntgdb 1/1 Corrida 0 9s
hello-web-4017757401-pc4j9 1/1 Corrida 0 9s
hello-web-4017757401-px7tx 1/1 Corrida 0 1m
```

Agora, você tem várias instâncias do seu aplicativo sendo executadas independentemente umas das outras e você pode usar o `kubectl scale` comando para ajustar a capacidade do seu aplicativo.

O balanceador de carga que você provisionou na etapa anterior começará a rotear o tráfego para essas novas réplicas automaticamente.

Trabalhando com Pods

Quando os kubernetes estiverem ativos e funcionando, você poderá visualizar e editar os pods.

```
prashantagcppaudel @ cloudshell: ~ (fourpointtwo-221807) $
kubectl obter pods
NOME PRONTO ESTADO REESTAR ATRASO
frontend-67f65745c-c7j8q 1/1 Rodando 0 2h
frontend-67f65745c-q7hnl 1/1 Rodando 0 2h
frontend-67f65745c-rm2s8 1/1 Rodando 0 2h
redis-master-585798d8ff-9mbfz 1/1 Rodando 0 2h
redis-escravo-865486c9df-m6gwd 1/1 Rodando 0 2h
redis-escravo-865486c9df-wjft6 1/1 Rodando 0 2h
```

Trabalhando com Serviços

Você pode ver todos os serviços em execução no Kubernetes acessando o `Kubernetes engine> services`

Navigation menu

Kubernetes Engine

Clusters

Workloads

Services

Applications

Configuration

Storage

Google Cloud Platform

fourpointtwo

Services

REFRESH

Kubernetes services

Brokered services

BETA

Services are sets of pods with a network endpoint that can be used for discovery and load balancing. Ingresses are collections of rules for routing external HTTP(S) traffic to services.

is system object: False

Filter resources

X

Name	Status	Service Type	Endpoints	Pods	Namespace	Cluster
frontend	Ok	Load balancer	35.228.127.160:80 L	3 / 3	default	europa-cluster
nginx-1-nginx-svc	Ok	Load balancer	35.194.247.21:80 L 35.194.247.21:443 L	3 / 3	default	asia-east-cluster
redis-master	Ok	Cluster IP	10.27.240.58	1 / 1	default	europa-cluster
redis-slave	Ok	Cluster IP	10.27.241.209	2 / 2	default	europa-cluster

Serviços

