Les bases de StatefulSet (StatefulSet Basics)

Cet exercice fournit une introduction à la gestion d'applications avec **StatefulSets**. Il montre comment **créer, supprimer, mettre à l'échelle** et **mettre à jour les pods de StatefulSets**.

Objectifs

Les StatefulSets sont destinés à être utilisés avec des applications avec état et des systèmes distribués. Cependant, l'administration d'applications dynamiques et de systèmes répartis sur Kubernetes est un sujet vaste et complexe. Afin de démontrer les fonctionnalités de base d'un StatefulSet et de ne pas confondre le premier sujet avec le second, vous allez déployer une application Web simple à l'aide d'un **StatefulSet**.

Après cet exercice, vous serez familiarisé avec ce qui suit.

- Comment créer un StatefulSet
- Comment un StatefulSet gère ses pods
- Comment supprimer un StatefulSet
- Comment redimensionner un StatefulSet
- Comment mettre à jour les pods d'un StatefulSet

Cet exercice suppose que votre cluster est configuré pour provisionner de manière dynamique **PersistentVolumes**. Si votre cluster n'est pas configuré pour le faire, vous devrez provisionner manuellement deux volumes de 1 Gio avant de commencer.

Créer un StatefulSet

Commencez par créer un StatefulSet en utilisant l'exemple ci-dessous. Il est similaire à l'exemple présenté dans le concept StatefulSets. Il crée un service headless, nginx, pour publier les adresses IP des pods dans le StatefulSet, Web.

web.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
  - port: 80
    name: web
  clusterIP: None
  selector:
    app: nginx
```

web.yaml

```
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  serviceName: "nginx"
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: k8s.gcr.io/nginx-slim:0.8
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: www
          mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
  - metadata:
      name: www
    spec:
      accessModes: [ "ReadWriteOnce" ]
      resources:
        requests:
          storage: 1Gi
```

Enregistrez l'exemple dans un fichier nommé **web.yaml**.
Vous devrez utiliser deux fenêtres de terminal. Dans le premier terminal, utilisez **kubectl get** pour regarder la création des pods de StatefulSet.

sudo kubectl get pods -w -l app=nginx

Dans le second terminal, utilisez **kubectl create** pour créer le service Headless et le StatefulSet définis dans web.yaml.

sudo kubectl create -f web.yaml

service/nginx created
statefulset.apps/web created

La commande ci-dessus crée deux pods, chacun exécutant un serveur Web NGINX. Obtenez le service nginx et le web StatefulSet pour vérifier qu'ils ont été créés avec succès.

sudo kubectl get service nginx

```
NAME    TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
nginx   ClusterIP   None         <none>        80/TCP    12s
```

sudo kubectl get statefulset web

```
NAME    DESIRED   CURRENT   AGE
web     2         1         20s
```

Création de pod ordonnée

Pour un StatefulSet avec N répliques, lors du déploiement des pods, ils sont créés séquentiellement, dans l'ordre de {0..N-1}. Examinez le résultat de la commande **kubectl get** dans le premier terminal. Finalement, la sortie ressemblera à l'exemple ci-dessous.

sudo kubectl get pods -w -l app=nginx

```
NAME    READY   STATUS              RESTARTS   AGE
web-0   0/1     Pending             0          0s
web-0   0/1     Pending             0          0s
web-0   0/1     ContainerCreating   0          0s
web-0   1/1     Running             0          19s
web-1   0/1     Pending             0          0s
web-1   0/1     Pending             0          0s
web-1   0/1     ContainerCreating   0          0s
web-1   1/1     Running             0          18s
```

Notez que le **pod Web-1** n'est pas lancé tant que le **Pod web-0** il n'est pas en cours d'exécution et prêt.

Pods dans un StatefulSet

Les pods d'un StatefulSet ont un index ordinal unique et une identité de réseau stable.

Examen de l'index ordinal du pod

Récupérez les pods de StatefulSet.

sudo kubectl get pods -l app=nginx

```
NAME      READY     STATUS    RESTARTS   AGE
web-0     1/1       Running   0          1m
web-1     1/1       Running   0          1m
```

Comme indiqué dans le concept StatefulSets, les pods d'un StatefulSet ont une identité unique et collante. Cette identité est basée sur un index ordinal unique attribué à chaque pod par le contrôleur StatefulSet. Les noms des pods se présentent sous la forme **<statefulset name> - <index ordininal>**. StatefulSet ayant deux réplicas sur le Web, il crée deux pods, Web-0 et Web-1.

Utilisation d'identités de réseau stables

Chaque pod a un nom d'hôte stable basé sur son index ordinal. Utilisez **kubectl exec** pour exécuter la commande hostname dans chaque pod.

for i in 0 1; do kubectl exec web-$i -- sh -c 'hostname'; done

A remplacer si nécessaire par :
sudo kubectl exec web-0 -- sh -c 'hostname'
sudo kubectl exec web-1 -- sh -c 'hostname'

```
web-0
web-1
```

Utilisez **kubectl run** pour exécuter un conteneur fournissant la commande **nslookup** à partir du package **dnsutils**. En utilisant nslookup sur les noms d'hôte des pods, vous pouvez examiner leurs adresses DNS dans le cluster.

sudo kubectl run -i --tty --image busybox dns-test --restart=Never --rm /bin/sh

```
nslookup web-0.nginx
Server:    10.0.0.10
Address 1: 10.0.0.10 kube-dns.kube-system.svc.cluster.local

Name:      web-0.nginx
Address 1: 10.244.1.6

nslookup web-1.nginx
Server:    10.0.0.10
Address 1: 10.0.0.10 kube-dns.kube-system.svc.cluster.local

Name:      web-1.nginx
Address 1: 10.244.2.6
```

Le CNAME du service headless pointe vers les services SRV (un pour chaque pod en cours d'exécution et prêt). Les services SRV pointent vers les entrées de service A contenant les adresses IP des pods.

Dans un terminal, regardez les pods de StatefulSet.

sudo kubectl get pod -w -l app=nginx

Dans un deuxième terminal, utilisez kubectl delete pour supprimer tous les pods du StatefulSet.

sudo kubectl delete pod -l app=nginx
pod "web-0" deleted
pod "web-1" deleted

Attendez que StatefulSet les redémarre et que les deux pods passent à Running et Ready.

sudo kubectl get pod -w -l app=nginx

```
NAME     READY     STATUS              RESTARTS   AGE
web-0    0/1       ContainerCreating   0          0s
NAME     READY     STATUS    RESTARTS   AGE
web-0    1/1       Running   0          2s
web-1    0/1       Pending   0          0s
web-1    0/1       Pending   0          0s
web-1    0/1       ContainerCreating   0          0s
web-1    1/1       Running   0          34s
```

Utilisez **kubectl exec** et **kubectl run** pour afficher les noms d'hôte des pods et les entrées DNS dans le cluster.

for i in 0 1; do kubectl exec web-$i -- sh -c 'hostname'; done

A remplacer si nécessaire par:
sudo kubectl exec web-0 -- sh -c 'hostname'
sudo kubectl exec web-1 -- sh -c 'hostname'

web-0
web-1

sudo kubectl run -i --tty --image busybox dns-test --restart=Never --rm /bin/sh

```
nslookup web-0.nginx
Server:    10.0.0.10
Address 1: 10.0.0.10 kube-dns.kube-system.svc.cluster.local

Name:      web-0.nginx
```

Address 1: 10.244.1.7

nslookup web-1.nginx
Server: 10.0.0.10
Address 1: 10.0.0.10 kube-dns.kube-system.svc.cluster.local

Name: web-1.nginx
Address 1: 10.244.2.8

Les ordinaux des pods, les noms d'hôte, les services SRV et les noms d'enregistrement A des pods n'ont pas changé, mais les adresses IP associées aux pods peuvent avoir changé. Dans le cluster utilisé pour cet exercice, ils ont changé. C'est pourquoi il est important de ne pas configurer d'autres applications pour qu'elles se connectent aux pods dans un StatefulSet par adresse IP.

Si vous devez rechercher et vous connecter aux membres actifs d'un StatefulSet, vous devez interroger le CNAME du service Headless (**nginx.default.svc.cluster.local**). Les enregistrements SRV associés au CNAME ne contiendront que les pods du StatefulSet en cours d'exécution et prêts.

Si votre application implémente déjà une logique de connexion qui teste la vivacité (**liveness**) et la disponibilité (**readiness**), vous pouvez utiliser les services SRV des pods (**web-0.nginx.default.svc.cluster.local, web-1.nginx.default.svc.cluster. local**), car elles sont stables, et votre application pourra découvrir les adresses des pods lorsqu'elles passeront à **Running** et **Ready**.

Écrire dans un stockage stable
Obtenez les PersistentVolumeClaims pour web-0 et web-1.

sudo kubectl get pvc -l app=nginx

NAME	STATUS	VOLUME	CAPACITY	ACCESSMODES	AGE
www-web-0	Bound	pvc-15c268c7-b507-11e6-932f-42010a800002	1Gi	RWO	48s
www-web-1	Bound	pvc-15c79307-b507-11e6-932f-42010a800002	1Gi	RWO	48s

Le contrôleur StatefulSet a créé deux PersistentVolumeClaim qui sont liés à deux PersistentVolumes. Le cluster utilisé dans cet exercice étant configuré pour provisionner dynamiquement PersistentVolumes, ces derniers ont été créés et liés automatiquement.

Les serveurs Web NGINX, par défaut, serviront un fichier d'index à **/usr/share/nginx/html/index.html**. Le champ volumeMounts de la spécification StatefulSets garantit que le répertoire **/usr/share/nginx/html** est sauvegardé par un PersistentVolume.

Ecrivez les noms d'hôte des pods dans leurs fichiers index.html et vérifiez que les serveurs Web NGINX servent les noms d'hôte.

for i in 0 1; do kubectl exec web-$i -- sh -c 'echo $(hostname) > /usr/share/nginx/html/index.html'; done

A remplacer si nécessaire par :
sudo kubectl exec web-0 -- sh -c 'echo $(hostname) > /usr/share/nginx/html/index.html'

sudo kubectl exec web-1 -- sh -c 'echo $(hostname) > /usr/share/nginx/html/index.html'

for i in 0 1; **do** kubectl exec -it web-$i -- curl localhost; **done**

A remplacer si nécessaire par :
sudo kubectl exec -it web-0 -- curl localhost
sudo kubectl exec -it web-1 -- curl localhost

web-0
web-1

Remarque: Si vous voyez à la place 403 Forbidden responses pour la commande curl ci-dessus, vous devrez corriger les autorisations du répertoire monté par volumeMounts (en raison d'un bogue lors de l'utilisation de volumes hostPath) avec:

for i in 0 1; do kubectl exec web-$i -- chmod 755 /usr/share/nginx/html; done

A remplacer si nécessaire par :

sudo kubectl exec web-0 -- chmod 755 /usr/share/nginx/html

sudo kubectl exec web-1 -- chmod 755 /usr/share/nginx/html

avant de réessayer la commande curl ci-dessus.

Dans un terminal, regardez les pods de StatefulSet.

sudo kubectl get pod -w -l app=nginx

Dans un deuxième terminal, supprimez tous les pods de StatefulSet.
sudo kubectl delete pod -l app=nginx

pod "web-0" deleted
pod "web-1" deleted

Examinez le résultat de la commande **kubectl get** dans le premier terminal et attendez que tous les pods passent à Running et Ready.

sudo kubectl get pod -w -l app=nginx

```
NAME     READY     STATUS              RESTARTS   AGE
web-0    0/1       ContainerCreating   0          0s
NAME     READY     STATUS    RESTARTS   AGE
web-0    1/1       Running   0          2s
```

```
web-1   0/1   Pending   0   0s
web-1   0/1   Pending   0   0s
web-1   0/1   ContainerCreating   0   0s
web-1   1/1   Running   0   34s
```

Vérifiez que les serveurs Web continuent à servir leurs noms d'hôte.

for i in 0 1; do kubectl exec -it web-$i -- curl localhost; done
A remplacer si nécessaire par:

sudo kubectl exec -it web-0 -- curl localhost

sudo kubectl exec -it web-1 -- curl localhost

```
web-0
web-1
```

Même si **web-0** et **web-1** ont été reprogrammés, ils continuent de servir leurs noms d'hôte car les volumes persistants associés à leurs **PersistentVolumeClaims** sont remontés sur leurs volumes de montage. Quels que soient les nœuds Web-0 et Web-1 programmés, leurs volumes persistants seront montés sur les points de montage appropriés.

Mise à l'échelle (scaling) d'un StatefulSet
La mise à l'échelle d'un StatefulSet fait référence à l'augmentation ou la diminution du nombre de réplicas. Ceci est accompli en mettant à jour le champ réplicas. Vous pouvez utiliser **kubectl scale** ou **kubectl patch** pour redimensionner un StatefulSet.

Mise à l'échelle
Dans une fenêtre de terminal, regardez les pods dans le StatefulSet.

sudo kubectl get pods -w -l app=nginx

Dans une autre fenêtre de terminal, utilisez **kubectl scale** pour redimensionner le nombre de répliques à 5.

sudo kubectl scale sts web –replicas=5

```
statefulset.apps/web scaled
```

Examinez le résultat de la commande **kubectl get** dans le premier terminal et attendez que les trois pods supplémentaires passent à **Running** et **Ready**.

sudo kubectl get pods -w -l app=nginx

```
NAME    READY   STATUS    RESTARTS   AGE
web-0   1/1     Running   0          2h
web-1   1/1     Running   0          2h
NAME    READY   STATUS    RESTARTS   AGE
```

```
web-2   0/1   Pending   0   0s
web-2   0/1   Pending   0   0s
web-2   0/1   ContainerCreating   0   0s
web-2   1/1   Running   0   19s
web-3   0/1   Pending   0   0s
web-3   0/1   Pending   0   0s
web-3   0/1   ContainerCreating   0   0s
web-3   1/1   Running   0   18s
web-4   0/1   Pending   0   0s
web-4   0/1   Pending   0   0s
web-4   0/1   ContainerCreating   0   0s
web-4   1/1   Running   0   19s
```

Le contrôleur StatefulSet a mis à l'échelle le nombre de réplicas. Comme pour la création de StatefulSet, le contrôleur StatefulSet a créé chaque pod séquentiellement par rapport à son index ordinal. Il a attendu que le prédécesseur de chaque pod soit en cours d'exécution et prêt avant de lancer le pod suivant.

Réduire (Scaling Down)

Dans un terminal, regardez les pods de StatefulSet.

sudo kubectl get pods -w -l app=nginx

Dans un autre terminal, utilisez le **kubectl patch** pour réduire le StatefulSet à trois réplicas.

sudo kubectl patch sts web -p '{"spec":{"replicas":3}}'

statefulset.apps/web patched

Attendez que Web-4 et Web-3 passent à Terminating.

sudo kubectl get pods -w -l app=nginx

```
NAME    READY   STATUS            RESTARTS   AGE
web-0   1/1     Running           0          3h
web-1   1/1     Running           0          3h
web-2   1/1     Running           0          55s
web-3   1/1     Running           0          36s
web-4   0/1     ContainerCreating 0          18s
NAME    READY   STATUS    RESTARTS   AGE
web-4   1/1     Running   0          19s
web-4   1/1     Terminating   0      24s
web-4   1/1     Terminating   0      24s
web-3   1/1     Terminating   0      42s
web-3   1/1     Terminating   0      42s
```

Arrêt de pod ordonné

Le contrôleur a supprimé un pod à la fois, dans l'ordre inverse de son index ordinal, et il a attendu qu'ils s'éteignent complètement avant de supprimer le suivant.

Affichez les PersistentVolumeClaims de StatefulSet.

sudo kubectl get pvc -l app=nginx

NAME	STATUS	VOLUME	CAPACITY	ACCESSMODES	AGE
www-web-0	Bound	pvc-15c268c7-b507-11e6-932f-42010a800002	1Gi	RWO	13h
www-web-1	Bound	pvc-15c79307-b507-11e6-932f-42010a800002	1Gi	RWO	13h
www-web-2	Bound	pvc-e1125b27-b508-11e6-932f-42010a800002	1Gi	RWO	13h
www-web-3	Bound	pvc-e1176df6-b508-11e6-932f-42010a800002	1Gi	RWO	13h
www-web-4	Bound	pvc-e11bb5f8-b508-11e6-932f-42010a800002	1Gi	RWO	13h

Il y a toujours cinq PersistentVolumeClaims et cinq PersistentVolumes. Lors de l'exploration du stockage stable d'un pod, nous avons constaté que les volumes persistants montés sur les pods d'un StatefulSet ne sont pas supprimés lorsque les pods de StatefulSet sont supprimés. Cela est toujours vrai lorsque la suppression du pod est provoquée par la réduction (scale down) de StatefulSet.

Mise à jour des StatefulSets

Dans Kubernetes 1.7 et versions ultérieures, le contrôleur StatefulSet prend en charge les mises à jour automatisées. La stratégie utilisée est déterminée par le champ **spec.updateStrategy** de l'objet **API StatefulSet.** Cette fonctionnalité peut être utilisée pour mettre à niveau les images de conteneur, les demandes de ressources et / ou les limites, les étiquettes et les annotations des pods dans un StatefulSet. Il existe deux stratégies de mise à jour valides, RollingUpdate et OnDelete.

La stratégie de mise à jour RollingUpdate est la stratégie par défaut pour StatefulSets.

Rolling Update

La stratégie de mise à jour RollingUpdate mettra à jour tous les pods d'un StatefulSet, dans l'ordre ordinal inverse, tout en respectant les garanties de StatefulSet.

Corrigez (patch) le composant Web StatefulSet pour appliquer la stratégie de mise à jour RollingUpdate.

sudo kubectl patch statefulset web -p
'{"spec":{"updateStrategy":{"type":"RollingUpdate"}}}'
statefulset.apps/web patched

Dans une fenêtre de terminal, patch le Web StatefulSet pour modifier à nouveau l'image du conteneur.

sudo kubectl patch statefulset web --type='json' -p='[{"op": "replace", "path": "/spec/template/spec/containers/0/image", "value":"gcr.io/google_containers/nginx-slim:0.8"}]'
statefulset.apps/web patched

Dans un autre terminal, observez les pods dans le StatefulSet.

sudo kubectl get po -l app=nginx -w

```
NAME    READY   STATUS    RESTARTS   AGE
web-0   1/1     Running   0          7m
web-1   1/1     Running   0          7m
web-2   1/1     Running   0          8m
web-2   1/1     Terminating   0          8m
web-2   1/1     Terminating   0          8m
web-2   0/1     Terminating   0          8m
web-2   0/1     Terminating   0          8m
web-2   0/1     Terminating   0          8m
web-2   0/1     Terminating   0          8m
web-2   0/1     Pending   0          0s
web-2   0/1     Pending   0          0s
web-2   0/1     ContainerCreating   0          0s
web-2   1/1     Running   0          19s
web-1   1/1     Terminating   0          8m
web-1   0/1     Terminating   0          8m
web-1   0/1     Terminating   0          8m
web-1   0/1     Terminating   0          8m
web-1   0/1     Pending   0          0s
web-1   0/1     Pending   0          0s
web-1   0/1     ContainerCreating   0          0s
web-1   1/1     Running   0          6s
web-0   1/1     Terminating   0          7m
web-0   1/1     Terminating   0          7m
web-0   0/1     Terminating   0          7m
web-0   0/1     Terminating   0          7m
web-0   0/1     Terminating   0          7m
web-0   0/1     Terminating   0          7m
web-0   0/1     Pending   0          0s
web-0   0/1     Pending   0          0s
web-0   0/1     ContainerCreating   0          0s
web-0   1/1     Running   0          10s
```

Les pods de StatefulSet sont mis à jour dans l'ordre ordinal inverse. Le contrôleur StatefulSet termine chaque pod et attend sa transition vers Running et Ready avant de mettre à jour le prochain pod. Notez que, même si le contrôleur StatefulSet ne procède pas à la mise à jour du prochain pod tant que son successeur ordinal est en cours d'exécution et prêt, il restaure tout pod qui a échoué lors de la mise à jour à sa version actuelle. Les pods qui ont déjà reçu la mise à jour seront restaurés dans la version mise à jour et les pods qui n'ont pas encore reçu la mise à jour seront restaurés dans la version précédente. De cette manière, le contrôleur tente de continuer à maintenir l'application en bonne santé et la mise à jour cohérente en présence de défaillances intermittentes.

Obtenez les pods pour voir leurs images de conteneur.

for p in 0 1 2; do kubectl get po web-$p --template '{{range $i,
$c := .spec.containers}}{{$c.image}}{{end}}'; echo; done

A remplacer si nécessaire par:

sudo kubectl get po web-0 --template '{{range $i,
$c := .spec.containers}}{{$c.image}}{{end}}'; echo

sudo kubectl get po web-1 --template '{{range $i,
$c := .spec.containers}}{{$c.image}}{{end}}'; echo

sudo kubectl get po web-2 --template '{{range $i,
$c := .spec.containers}}{{$c.image}}{{end}}'; echo

k8s.gcr.io/nginx-slim:0.8
k8s.gcr.io/nginx-slim:0.8
k8s.gcr.io/nginx-slim:0.8

Tous les pods de StatefulSet exécutent maintenant l'image de conteneur précédente.

A noter que vous pouvez également utiliser **kubectl rollout status sts / <name>** pour afficher le statut d'une mise à jour de rolling update.

Mise en place d'une mise à jour (Staging an Update)

Vous pouvez créer une mise à jour d'un **StatefulSet** à l'aide du paramètre partition de la stratégie de mise à jour **RollingUpdate**. Une mise à jour progressive conservera tous les pods dans le StatefulSet à la version actuelle tout en autorisant des mutations dans le fichier **.spec.template** de **StatefulSet**.

Patcher le Web StatefulSet pour ajouter une partition au champ **updateStrategy**.

sudo kubectl patch statefulset web -p
'{"spec":{"updateStrategy":{"type":"RollingUpdate","rollingUpdate":{"partition":3}}}}'
statefulset.apps/web patched

Patcher à nouveau le StatefulSet pour modifier l'image du conteneur.

sudo kubectl patch statefulset web --type='json' -p='[{"op": "replace", "path":
"/spec/template/spec/containers/0/image", "value":"k8s.gcr.io/nginx-slim:0.7"}]'
statefulset.apps/web patched

Supprimer un pod dans le StatefulSet.

sudo kubectl delete po web-2
pod "web-2" deleted

Attendez que le pod soit en cours d'exécution et prêt.

sudo kubectl get po -l app=nginx -w

```
NAME    READY   STATUS              RESTARTS   AGE
web-0   1/1     Running             0          4m
web-1   1/1     Running             0          4m
web-2   0/1     ContainerCreating   0          11s
web-2   1/1     Running   0         18s
```

Récupérez le conteneur du pod.

sudo kubectl get po web-2 --template '{{range $i, $c := .spec.containers}}{{$c.image}}{{end}}'
k8s.gcr.io/nginx-slim:0.8

Notez que, même si la stratégie de mise à jour est RollingUpdate, le contrôleur StatefulSet a restauré le pod avec son conteneur d'origine. En effet, l'ordinal du pod est inférieur à la partition spécifiée par **updateStrategy**.

Déployer un canari (Rolling Out a Canary)

Vous pouvez déployer un canari pour tester une modification en décrémentant la partition que vous avez spécifiée ci-dessus.

Patcher le StatefulSet pour décrémenter la partition.

sudo kubectl patch statefulset web -p '{"spec":{"updateStrategy":{"type":"RollingUpdate","rollingUpdate":{"partition":2}}}}'
statefulset.apps/web patched

Attendez que le Web-2 soit en cours d'exécution et prêt.

sudo kubectl get po -l app=nginx -w

```
NAME    READY   STATUS              RESTARTS   AGE
web-0   1/1     Running             0          4m
web-1   1/1     Running             0          4m
web-2   0/1     ContainerCreating   0          11s
web-2   1/1     Running   0         18s
```

Récupérez le conteneur du pod.

sudo kubectl get po web-2 --template '{{range $i, $c := .spec.containers}}{{$c.image}}{{end}}'
k8s.gcr.io/nginx-slim:0.7

Lorsque vous avez modifié la partition, le contrôleur StatefulSet a automatiquement mis à jour le pod Web-2 car son ordinal était supérieur ou égal à la partition.

Supprimez le pod Web-1.

sudo kubectl delete po web-1
pod "web-1" deleted

Attendez que le pod Web-1 soit en cours d'exécution et prêt.

sudo kubectl get po -l app=nginx -w

```
NAME    READY   STATUS        RESTARTS   AGE
web-0   1/1     Running       0          6m
web-1   0/1     Terminating   0          6m
web-2   1/1     Running       0          2m
web-1   0/1     Terminating   0          6m
web-1   0/1     Terminating   0          6m
web-1   0/1     Terminating   0          6m
web-1   0/1     Pending       0          0s
web-1   0/1     Pending       0          0s
web-1   0/1     ContainerCreating   0          0s
web-1   1/1     Running       0          18s
```

Get the web-1 Pods container.

Obtenez le conteneur de Pods Web-1.

sudo kubectl get po web-1 --template '{{range $i, $c := .spec.containers}}{{$c.image}}{{end}}'
k8s.gcr.io/nginx-slim:0.8

La configuration initiale de Web-1 a été restaurée car l'ordinal du pod était inférieur à la partition. Lorsqu'une partition est spécifiée, tous les pods dont l'ordinal est supérieur ou égal à la partition seront mis à jour lors de la mise à jour du fichier **.spec.template** de StatefulSet. Si un pod dont le nombre ordinal est inférieur à celui de la partition est supprimé ou terminé, il sera restauré dans sa configuration d'origine.

Déploiement par étapes (Phased Roll Outs)
Vous pouvez effectuer un déploiement par étapes (par exemple, un déploiement linéaire, géométrique ou exponentiel) à l'aide d'une mise à jour rolling update partitionnée de la même manière que vous avez déployé un canari. Pour effectuer un déploiement par étapes, définissez la partition sur l'ordinal auquel vous souhaitez que le contrôleur suspende la mise à jour.

La partition est actuellement définie sur 2. Définissez la partition sur 0.

sudo kubectl patch statefulset web -p
'{"spec":{"updateStrategy":{"type":"RollingUpdate","rollingUpdate":{"partition":0}}}}'
statefulset.apps/web patched

Attendez que tous les pods de StatefulSet soient en cours d'exécution et prêts.

sudo kubectl get po -l app=nginx -w

```
NAME    READY   STATUS              RESTARTS   AGE
web-0   1/1     Running             0          3m
web-1   0/1     ContainerCreating   0          11s
web-2   1/1     Running             0          2m
web-1   1/1     Running             0          18s
```

```
web-0   1/1   Terminating   0   3m
web-0   1/1   Terminating   0   3m
web-0   0/1   Terminating   0   3m
web-0   0/1   Terminating   0   3m
web-0   0/1   Terminating   0   3m
web-0   0/1   Terminating   0   3m
web-0   0/1   Pending   0   0s
web-0   0/1   Pending   0   0s
web-0   0/1   ContainerCreating   0   0s
web-0   1/1   Running   0   3s
```

Récupérez les conteneurs du Pod.

for p in 0 1 2; do kubectl get po web-$p --template '{{range $i,
$c := .spec.containers}}{{$c.image}}{{end}}'; echo; done

A remplacer si nécessaire par:

sudo kubectl get po web-0 --template '{{range $i,
$c := .spec.containers}}{{$c.image}}{{end}}'; echo

sudo kubectl get po web-1 --template '{{range $i,
$c := .spec.containers}}{{$c.image}}{{end}}'; echo

sudo kubectl get po web-2 --template '{{range $i,
$c := .spec.containers}}{{$c.image}}{{end}}'; echo

```
k8s.gcr.io/nginx-slim:0.7
k8s.gcr.io/nginx-slim:0.7
k8s.gcr.io/nginx-slim:0.7
```

En déplaçant la partition sur 0, vous avez autorisé le contrôleur StatefulSet à poursuivre le processus de mise à jour.

Mise à jour avec suppression (On Delete)

La stratégie de mise à jour **OnDelete** implémente le comportement hérité – legacy- (1.6 et versions antérieures). Lorsque vous sélectionnez cette stratégie de mise à jour, le contrôleur StatefulSet ne met pas à jour automatiquement les pods lorsqu'une modification est apportée au champ **.spec.template** de StatefulSet. Cette stratégie peut être sélectionnée en définissant le **.spec.template.updateStrategy.type** sur **OnDelete**.

Suppression de StatefulSets

StatefulSet prend en charge la suppression de non en cascade et en cascade. Dans une suppression non en cascade, les pods StatefulSet ne sont pas supprimés lorsque StatefulSet est supprimé. Dans une suppression en cascade, le StatefulSet et ses pods sont supprimés.

Suppression non en cascade (Non-Cascading Delete)

Dans une fenêtre de terminal, regardez les pods dans le StatefulSet.

sudo kubectl get pods -w -l app=nginx

Utilisez **kubectl delete** pour supprimer le StatefulSet. Assurez-vous de fournir le paramètre **--cascade = false** à la commande. Ce paramètre indique à Kubernetes de ne supprimer que le StatefulSet et de ne supprimer aucun de ses pods.

sudo kubectl delete statefulset web --cascade=false
statefulset.apps "web" deleted

Demandez aux pods d'examiner leur statut.

sudo kubectl get pods -l app=nginx

```
NAME    READY   STATUS    RESTARTS   AGE
web-0   1/1     Running   0          6m
web-1   1/1     Running   0          7m
web-2   1/1     Running   0          5m
```

Même si Web a été supprimé, tous les pods sont toujours en cours d'exécution et prêts. Supprimer web-0.

sudo kubectl delete pod web-0
pod "web-0" deleted

Récupérez les pods de StatefulSet.

sudo kubectl get pods -l app=nginx

```
NAME    READY   STATUS    RESTARTS   AGE
web-1   1/1     Running   0          10m
web-2   1/1     Running   0          7m
```

Web StatefulSet ayant été supprimé, Web-0 n'a pas été relancé.

Dans un terminal, regardez les pods de StatefulSet.

sudo kubectl get pods -w -l app=nginx

Dans un deuxième terminal, recréez le StatefulSet. Notez que, sauf si vous avez supprimé le service nginx (ce que vous ne devriez pas avoir), vous verrez une erreur indiquant que le service existe déjà.

sudo kubectl create -f web.yaml
statefulset.apps/web created
Error from server (AlreadyExists): error when creating **"web.yaml":** services **"nginx"** already exists

Ignorer l'erreur. Cela indique uniquement qu'une tentative de création du service nginx Headless a été effectuée, même si ce service existe déjà.

Examinez le résultat de la commande **kubectl get** en cours d'exécution dans le premier terminal.

sudo kubectl get pods -w -l app=nginx

```
NAME    READY   STATUS    RESTARTS   AGE
web-1   1/1     Running   0          16m
web-2   1/1     Running   0          2m
NAME    READY   STATUS    RESTARTS   AGE
web-0   0/1     Pending   0          0s
web-0   0/1     Pending   0          0s
web-0   0/1     ContainerCreating   0          0s
web-0   1/1     Running   0          18s
web-2   1/1     Terminating   0          3m
web-2   0/1     Terminating   0          3m
web-2   0/1     Terminating   0          3m
web-2   0/1     Terminating   0          3m
```

Lorsque le StatefulSet Web a été recréé, il a d'abord relancé Web-0. Comme web-1 était déjà en cours d'exécution et prêt, lorsque ce dernier est passé à Running et Ready, il a simplement adopté ce pod. Depuis que vous avez recréé StatefulSet avec des répliques égales à 2, une fois que Web-0 a été recréé et une fois que Web-1 a été déterminé comme étant déjà en cours d'exécution et prêt, Web-2 a été arrêté.

Examinons à nouveau le contenu du fichier index.html servi par les serveurs Web des pods.

for i in 0 1; do kubectl exec -it web-$i -- curl localhost; done
A remplacer si nécessaire par :

sudo kubectl exec -it web-0 -- curl localhost

sudo kubectl exec -it web-1 -- curl localhost

```
web-0
web-1
```

Même si vous avez supprimé les composants StatefulSet et web-0, celui-ci conserve le nom d'hôte saisi à l'origine dans son fichier index.html. En effet, StatefulSet ne supprime jamais les volumes persistants associés à un pod. Lorsque vous avez recréé StatefulSet et que Web-0 a été relancé, son PersistentVolume d'origine a été remonté.

Cascading Delete

Dans une fenêtre de terminal, regardez les pods dans le StatefulSet.

sudo kubectl get pods -w -l app=nginx

Dans un autre terminal, supprimez à nouveau StatefulSet. Cette fois, omettez le paramètre **--cascade = false**.

sudo kubectl delete statefulset web
statefulset.apps "web" deleted

Examinez le résultat de la commande **kubectl get** qui s'exécute dans le premier terminal et attendez que tous les pods soient transférés vers **Terminating**.

sudo kubectl get pods -w -l app=nginx

```
NAME    READY   STATUS    RESTARTS   AGE
web-0   1/1     Running   0          11m
web-1   1/1     Running   0          27m
NAME    READY   STATUS        RESTARTS   AGE
web-0   1/1     Terminating   0          12m
web-1   1/1     Terminating   0          29m
web-0   0/1     Terminating   0          12m
web-0   0/1     Terminating   0          12m
web-0   0/1     Terminating   0          12m
web-1   0/1     Terminating   0          29m
web-1   0/1     Terminating   0          29m
web-1   0/1     Terminating   0          29m
```

Comme vous l'avez vu dans la section Réduire (Scaling down), les pods sont terminés un par un, en respectant l'ordre inverse de leurs index ordinaux. Avant de terminer un pod, le contrôleur StatefulSet attend que son successeur soit complètement terminé.

Notez que, bien qu'une suppression en cascade supprime le StatefulSet et ses pods, elle ne supprimera pas le service headless associé au StatefulSet. Vous devez supprimer le service nginx manuellement.

sudo kubectl delete service nginx
service "nginx" deleted

Recréez le service StatefulSet et Headless une fois de plus.

sudo kubectl create -f web.yaml
service/nginx created
statefulset.apps/web created

Lorsque tous les pods de StatefulSet passent à Running et Ready, récupérez le contenu de leurs fichiers **index.html**.

for i in 0 1; do kubectl exec -it web-$i -- curl localhost; done

A remplacer si nécessaire par :

sudo kubectl exec -it web-0 -- curl localhost

sudo kubectl exec -it web-1 -- curl localhost

web-0
web-1

Même si vous avez complètement supprimé StatefulSet et tous ses pods, ceux-ci sont recréés avec leur PersistentVolumes monté, et web-0 et web-1 serviront toujours leurs noms d'hôte.

Enfin, supprimez le Web StatefulSet et le service nginx.

sudo kubectl delete service nginx
service **"nginx"** deleted

sudo kubectl delete statefulset web
statefulset **"web"** deleted

Politique de gestion des pods (Pod Management Policy)
Pour certains systèmes distribués, les garanties de commande StatefulSet sont inutiles et/ou indésirables. Ces systèmes ne requièrent que l'unicité et l'identité. Pour résoudre ce problème, dans Kubernetes 1.7, nous avons ajouté .**spec.podManagementPolicy** à l'objet d'API StatefulSet.

Gestion des pods commandés (OrderedReady Pod Management)
La gestion du pod **OrderedReady** est la valeur par défaut pour StatefulSets. Il indique au contrôleur StatefulSet de respecter les garanties de commande démontrées ci-dessus.

Gestion des pods parallèles (Parallel Pod Management)
La gestion des pods parallèles indique au contrôleur StatefulSet de lancer ou de terminer tous les pods en parallèle et de ne pas attendre que les pods soient en cours d'exécution et prêts ou soient complètement terminés avant de lancer ou de terminer un autre pod.

web-parallel.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
  - port: 80
    name: web
  clusterIP: None
  selector:
    app: nginx
```

web-parallel.yaml

```
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  serviceName: "nginx"
  podManagementPolicy: "Parallel"
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: k8s.gcr.io/nginx-slim:0.8
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: www
          mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
  - metadata:
      name: www
    spec:
      accessModes: [ "ReadWriteOnce" ]
      resources:
        requests:
          storage: 1Gi
```

Sauvegarder le fichier **web-parallel.yaml**

Ce manifeste est identique à celui que vous avez récupéré ci-dessus, à la différence que le **.spec.podManagementPolicy** du Web StatefulSet est défini sur Parallèle.

Dans un terminal, observez les pods dans le StatefulSet.

sudo kubectl get po -l app=nginx -w

Dans un autre terminal, créez le StatefulSet et le service dans le manifeste.

sudo kubectl create -f web-parallel.yaml
service/nginx created
statefulset.apps/web created

Examinez le résultat de la commande **kubectl get** que vous avez exécutée dans le premier terminal.

sudo kubectl get po -l app=nginx -w

```
NAME    READY   STATUS              RESTARTS   AGE
web-0   0/1     Pending             0          0s
web-0   0/1     Pending             0          0s
web-1   0/1     Pending             0          0s
web-1   0/1     Pending             0          0s
web-0   0/1     ContainerCreating   0          0s
web-1   0/1     ContainerCreating   0          0s
web-0   1/1     Running             0          10s
web-1   1/1     Running             0          10s
```

Le contrôleur StatefulSet a lancé Web-0 et Web-1 en même temps.

Laissez le deuxième terminal ouvert et, dans une autre fenêtre de terminal, mettez à l'échelle (scale) le StatefulSet.

sudo kubectl scale statefulset/web --replicas=4
statefulset.apps/web scaled

Examinez la sortie du terminal où la commande **kubectl get** est en cours d'exécution.

```
web-3   0/1     Pending             0          0s
web-3   0/1     Pending             0          0s
web-3   0/1     Pending             0          7s
web-3   0/1     ContainerCreating   0          7s
web-2   1/1     Running             0          10s
web-3   1/1     Running             0          26s
```

Le contrôleur StatefulSet a lancé deux nouveaux pods et il n'a pas attendu que le premier soit lancé et prêt avant de lancer le second.

Laissez ce terminal ouvert et, dans un autre terminal, supprimez le web StatefulSet.

sudo kubectl delete sts web

Encore une fois, examinez le résultat de la commande **kubectl get** qui s'exécute dans l'autre terminal.

```
web-3   1/1     Terminating   0          9m
web-2   1/1     Terminating   0          9m
web-3   1/1     Terminating   0          9m
web-2   1/1     Terminating   0          9m
```

```
web-1   1/1   Terminating   0   44m
web-0   1/1   Terminating   0   44m
web-0   0/1   Terminating   0   44m
web-3   0/1   Terminating   0   9m
web-2   0/1   Terminating   0   9m
web-1   0/1   Terminating   0   44m
web-0   0/1   Terminating   0   44m
web-2   0/1   Terminating   0   9m
web-2   0/1   Terminating   0   9m
web-2   0/1   Terminating   0   9m
web-1   0/1   Terminating   0   44m
web-1   0/1   Terminating   0   44m
web-1   0/1   Terminating   0   44m
web-0   0/1   Terminating   0   44m
web-0   0/1   Terminating   0   44m
web-0   0/1   Terminating   0   44m
web-3   0/1   Terminating   0   9m
web-3   0/1   Terminating   0   9m
web-3   0/1   Terminating   0   9m
```

Le contrôleur StatefulSet supprime tous les pods simultanément, il n'attend pas que le successeur ordinal du pod se termine avant de supprimer ce pod.

Fermez le terminal où la commande **kubectl get** est en cours d'exécution et supprimez le service nginx.

sudo kubectl delete svc nginx

Nettoyer

Vous devrez supprimer le support de stockage persistant pour **PersistentVolumes** utilisé dans cet exercice. Suivez les étapes nécessaires, en fonction de votre environnement, de la configuration de stockage et de la méthode de provisioning, pour vous assurer que tout le stockage est récupéré.