(/)

- Documentation (/docs/home/)
- Blog (/blog/)
- Partners (/partners/)
- Community (/community/)
- Case Studies (/case-studies/)
- English
    - Chinese (/cn/)
- v1.11
    - v1.11 (https://kubernetes.io)
    - v1.10 (https://v1-10.docs.kubernetes.io)
    - v1.9 (https://v1-9.docs.kubernetes.io)
    - v1.8 (https://v1-8.docs.kubernetes.io)
    - v1.7 (https://v1-7.docs.kubernetes.io)

Try Kubernetes (/docs/tutorials/kubernetes-basics/)

Get Started (/docs/tutorials/stateless-application/hello-minikube/)

Ready to get your hands dirty? Build a simple Kubernetes cluster that runs "Hello World" for Node.js.

Documentation (/docs/home/)

Learn how to use Kubernetes with the use of walkthroughs, samples, and reference documentation. You can even help contribute to the docs (/editdocs/)!

Community (/community/)

If you need help, you can connect with other Kubernetes users and the Kubernetes authors, attend community events, and watch video presentations from around the web.

Blog (/blog/)

Read the latest news for Kubernetes and the containers space in general, and get technical how-tos hot off the presses.

Interested in hacking on the core Kubernetes code base?

View On Github (https://github.com/kubernetes/kubernetes)

Explore the community

Twitter (https://twitter.com/kubernetesio) Github (https://github.com/kubernetes/kubernetes) Slack (http://slack.k8s.io/) Stack Overflow (http://stackoverflow.com/questions/tagged/kubernetes) Forum (https://discuss.kubernetes.io) Events Calendar (https://calendar.google.com/calendar/embed?src=nt2tcnbtbied3l6gi2h29slvc0%40group.calendar.google.com)

Kubernetes Blog

Monday, January 30, 2017

Running MongoDB on Kubernetes with StatefulSets (https://kubernetes.io/blog/2017/01/running-mongodb-on-kubernetes-with-statefulsets/)

Editor's note: Today's post is by Sandeep Dinesh, Developer Advocate, Google Cloud Platform, showing how to run a database in a container.

Conventional wisdom says you can't run a database in a container. "Containers are stateless!" they say, and "databases are pointless without state!"

Of course, this is not true at all. At Google, everything runs in a container, including databases. You just need the right tools. Kubernetes 1.5 (https://kubernetes.io/blog/2016/12/kubernetes-1.5-supporting-production-workloads) includes the new StatefulSet (http://kubernetes.io/docs/concepts/abstractions/controllers/statefulsets/) API object (in previous versions, StatefulSet was known as PetSet). With StatefulSets, Kubernetes makes it much easier to run stateful workloads such as databases.

If you've followed my previous posts, you know how to create a MEAN Stack app with Docker (http://blog.sandeepdinesh.com/2015/07/running-mean-web-application-in-docker.html), then migrate it to Kubernetes (https://medium.com/google-cloud/running-a-mean-stack-on-google-cloud-platform-with-kubernetes-149ca81c2b5d) to provide easier management and reliability, and create a MongoDB replica set (https://medium.com/google-cloud/mongodb-replica-sets-with-kubernetes-d96606bd9474) to provide redundancy and high availability.

While the replica set in my previous blog post worked, there were some annoying steps that you needed to follow. You had to manually create a disk, a ReplicationController, and a service for each replica. Scaling the set up and down meant managing all of these resources manually, which is an opportunity for error, and would put your stateful application at risk In the previous example, we created a Makefile to ease the management of of these resources, but it would have been great if Kubernetes could just take care of all of this for us.

With StatefulSets, these headaches finally go away. You can create and manage your MongoDB replica set natively in Kubernetes, without the need for scripts and Makefiles. Let's take a look how.

Note: StatefulSets are currently a beta resource. The sidecar container (https://github.com/cvallance/mongo-k8s-sidecar) used for auto-configuration is also unsupported.

Prerequisites and Setup

Before we get started, you'll need a Kubernetes 1.5+ and the Kubernetes command line tool (http://kubernetes.io/docs/user-guide/prereqs/). If you want to follow along with this tutorial and use Google Cloud Platform, you also need the Google Cloud SDK (http://cloud.google.com/sdk).

Once you have a Google Cloud project created (https://console.cloud.google.com/projectcreate) and have your Google Cloud SDK setup (hint: gcloud init), we can create our cluster.

To create a Kubernetes 1.5 cluster, run the following command:

```
gcloud container clusters create "test-cluster"
```

This will make a three node Kubernetes cluster. Feel free to customize the command (https://cloud.google.com/sdk/gcloud/reference/container/clusters/create) as you see fit.

Then, authenticate into the cluster:

```
gcloud container clusters get-credentials test-cluster
```

Setting up the MongoDB replica set

To set up the MongoDB replica set, you need three things: A StorageClass (http://kubernetes.io/docs/user-guide/persistent-volumes/#storageclasses), a Headless Service (http://kubernetes.io/docs/user-guide/services/#headless-services), and a StatefulSet (http://kubernetes.io/docs/concepts/abstractions/controllers/statefulsets/).

I’ve created the configuration files for these already, and you can clone the example from GitHub:

```
git clone https://github.com/thesandlord/mongo-k8s-sidecar.git

cd /mongo-k8s-sidecar/example/StatefulSet/
```

To create the MongoDB replica set, run these two commands:

```
kubectl apply -f googlecloud\_ssd.yaml

kubectl apply -f mongo-statefulset.yaml
```

That’s it! With these two commands, you have launched all the components required to run an highly available and redundant MongoDB replica set.

At an high level, it looks something like this:

Let’s examine each piece in more detail.

StorageClass

The storage class tells Kubernetes what kind of storage to use for the database nodes. You can set up many different types of StorageClasses in a ton of different environments. For example, if you run Kubernetes in your own datacenter, you can use GlusterFS (https://www.gluster.org/). On GCP, your storage choices (https://cloud.google.com/compute/docs/disks/) are SSDs and hard disks. There are currently drivers for AWS (http://kubernetes.io/docs/user-guide/persistent-volumes/#aws), Azure (http://kubernetes.io/docs/user-guide/persistent-volumes/#azure-disk), Google Cloud (http://kubernetes.io/docs/user-guide/persistent-volumes/#gce), GlusterFS (http://kubernetes.io/docs/user-guide/persistent-volumes/#glusterfs), OpenStack Cinder (http://kubernetes.io/docs/user-guide/persistent-volumes/#openstack-cinder), vSphere (http://kubernetes.io/docs/user-guide/persistent-volumes/#vsphere), Ceph RBD (http://kubernetes.io/docs/user-guide/persistent-volumes/#ceph-rbd), and Quobyte (http://kubernetes.io/docs/user-guide/persistent-volumes/#quobyte).

The configuration for the StorageClass looks like this:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
 name: fast
provisioner: kubernetes.io/gce-pd
parameters:
 type: pd-ssd
```

This configuration creates a new StorageClass called “fast” that is backed by SSD volumes. The StatefulSet can now request a volume, and the StorageClass will automatically create it!

Deploy this StorageClass:

```
kubectl apply -f googlecloud\_ssd.yaml
```

Headless Service

Now you have created the Storage Class, you need to make a Headless Service. These are just like normal Kubernetes Services, except they don’t do any load balancing for you. When combined with StatefulSets, they can give you unique DNS addresses that let you directly access the pods! This is perfect for creating MongoDB replica sets, because our app needs to connect to all of the MongoDB nodes individually.

The configuration for the Headless Service looks like this:

```
apiVersion: v1

kind: Service

metadata:

 name: mongo

 labels:

 name: mongo

spec:

 ports:

 - port: 27017

 targetPort: 27017

 clusterIP: None

 selector:

 role: mongo
```

You can tell this is a Headless Service because the clusterIP is set to “None.” Other than that, it looks exactly the same as any normal Kubernetes Service.

StatefulSet

The pièce de résistance. The StatefulSet actually runs MongoDB and orchestrates everything together. StatefulSets differ from Kubernetes ReplicaSets (http://kubernetes.io/docs/user-guide/replicasets/) (not to be confused with MongoDB replica sets!) in certain ways that makes them more suited for stateful applications. Unlike Kubernetes ReplicaSets, pods created under a StatefulSet have a few unique attributes. The name of the pod is not random, instead each pod gets an ordinal name. Combined with the Headless Service, this allows pods to have stable identification. In addition, pods are created one at a time instead of all at once, which can help when bootstrapping a stateful system. You can read more about StatefulSets in the documentation (http://kubernetes.io/docs/concepts/abstractions/controllers/statefulsets/).

Just like before, this “sidecar” container (https://github.com/cvallance/mongo-k8s-sidecar) will configure the MongoDB replica set automatically. A “sidecar” is a helper container which helps the main container do its work.

The configuration for the StatefulSet looks like this:

```
apiVersion: apps/v1beta1

kind: StatefulSet

metadata:

 name: mongo

spec:

 serviceName: "mongo"

 replicas: 3

 template:

 metadata:

 labels:

   role: mongo

   environment: test

 spec:

 terminationGracePeriodSeconds: 10

 containers:

   - name: mongo

     image: mongo

     command:

       - mongod

       - "--replSet"

       - rs0

       - "--smallfiles"

       - "--noprealloc"

     ports:

       - containerPort: 27017

     volumeMounts:

       - name: mongo-persistent-storage

         mountPath: /data/db

   - name: mongo-sidecar
```

```
    image: cvallance/mongo-k8s-sidecar

    env:

      - name: MONGO\_SIDECAR\_POD\_LABELS

        value: "role=mongo,environment=test"

volumeClaimTemplates:

- metadata:

name: mongo-persistent-storage

annotations:

  volume.beta.kubernetes.io/storage-class: "fast"

spec:

accessModes: ["ReadWriteOnce"]

resources:

  requests:

    storage: 100Gi
```

It’s a little long, but fairly straightforward.

The first second describes the StatefulSet object. Then, we move into the Metadata section, where you can specify labels and the number of replicas.

Next comes the pod spec. The terminationGracePeriodSeconds is used to gracefully shutdown the pod when you scale down the number of replicas, which is important for databases! Then the configurations for the two containers is shown. The first one runs MongoDB with command line flags that configure the replica set name. It also mounts the persistent storage volume to /data/db, the location where MongoDB saves its data. The second container runs the sidecar.

Finally, there is the volumeClaimTemplates. This is what talks to the StorageClass we created before to provision the volume. It will provision a 100 GB disk for each MongoDB replica.

Using the MongoDB replica set

At this point, you should have three pods created in your cluster. These correspond to the three nodes in your MongoDB replica set. You can see them with this command:

```
kubectl get pods

NAME   READY STATUS RESTARTS AGE

mongo-0 2/2  Running 0     3m

mongo-1 2/2  Running 0     3m

mongo-2 2/2  Running 0     3m
```

Each pod in a StatefulSet backed by a Headless Service will have a stable DNS name. The template follows this format: <pod-name>.<service-name>

This means the DNS names for the MongoDB replica set are:

```
mongo-0.mongo

mongo-1.mongo

mongo-2.mongo
```

You can use these names directly in the connection string URI (http://docs.mongodb.com/manual/reference/connection-string) of your app.

In this case, the connection string URI would be:

```
“mongodb://mongo-0.mongo,mongo-1.mongo,mongo-2.mongo:27017/dbname\_?”
```

That’s it!

Scaling the MongoDB replica set

A huge advantage of StatefulSets is that you can scale them just like Kubernetes ReplicaSets. If you want 5 MongoDB Nodes instead of 3, just run the scale command:

```
kubectl scale --replicas=5 statefulset mongo
```

The sidecar container will automatically configure the new MongoDB nodes to join the replica set.

Include the two new nodes (mongo-3.mongo & mongo-4.mongo) in your connection string URI and you are good to go. Too easy!

Cleaning Up

To clean up the deployed resources, delete the StatefulSet, Headless Service, and the provisioned volumes.

Delete the StatefulSet:

```
kubectl delete statefulset mongo
```

Delete the Service:

```
kubectl delete svc mongo
```

Delete the Volumes:

```
kubectl delete pvc -l role=mongo
```

Finally, you can delete the test cluster:

```
gcloud container clusters delete "test-cluster"
```

Happy Hacking!

For more cool Kubernetes and Container blog posts, follow me on Twitter (https://twitter.com/sandeepdinesh) and Medium (https://medium.com/@SandeepDinesh).

-Sandeep Dinesh, Developer Advocate, Google Cloud Platform.

« Prev (/blog/2017/01/how-we-run-kubernetes-in-kubernetes-kubeception/)

Next >> (/blog/2017/01/fission-serverless-functions-as-service-for-kubernetes/)

(https://kubernetes.io/feed.xml)

@Kubernetesio (https://twitter.com/kubernetesio)

View on Github (https://github.com/kubernetes/kubernetes)

#kubernetes-users (http://slack.k8s.io)

Stack Overflow (http://stackoverflow.com/questions/tagged/kubernetes)

Forum (https://discuss.kubernetes.io)

Download Kubernetes (http://get.k8s.io/)

2018

Introducing Kubebuilder: an SDK for building Kubernetes APIs using CRDs (https://kubernetes.io/blog/2018/08/10/introducing-kubebuilder-an-sdk-for-building-kubernetes-apis-using-crds/) Aug 10

Out of the Clouds onto the Ground: How to Make Kubernetes Production Grade Anywhere (https://kubernetes.io/blog/2018/08/03/out-of-the-clouds-onto-the-ground-how-to-make-kubernetes-production-grade-anywhere/) Aug 3

Dynamically Expand Volume with CSI and Kubernetes (https://kubernetes.io/blog/2018/08/02/dynamically-expand-volume-with-csi-and-kubernetes/) Aug 2

KubeVirt: Extending Kubernetes with CRDs for Virtualized Workloads (https://kubernetes.io/blog/2018/07/27/kubevirt-extending-kubernetes-with-crds-for-virtualized-workloads/) Jul 27

Feature Highlight: CPU Manager (https://kubernetes.io/blog/2018/07/24/feature-highlight-cpu-manager/) Jul 24

The History of Kubernetes & the Community Behind It (https://kubernetes.io/blog/2018/07/20/the-history-of-kubernetes--the-community-behind-it/) Jul 20

Kubernetes Wins the 2018 OSCON Most Impact Award (https://kubernetes.io/blog/2018/07/19/kubernetes-wins-2018-oscon-most-impact-award/) Jul 19

11 Ways (Not) to Get Hacked (https://kubernetes.io/blog/2018/07/18/11-ways-not-to-get-hacked/) Jul 18

How the sausage is made: the Kubernetes 1.11 release interview, from the Kubernetes Podcast (https://kubernetes.io/blog/2018/07/16/how-the-sausage-is-made-the-kubernetes-1.11-release-interview-from-the-kubernetes-podcast/) Jul 16

Resizing Persistent Volumes using Kubernetes (https://kubernetes.io/blog/2018/07/12/resizing-persistent-volumes-using-kubernetes/) Jul 12

Dynamic Kubelet Configuration (https://kubernetes.io/blog/2018/07/11/dynamic-kubelet-configuration/) Jul 11

Meet Our Contributors - Monthly Streaming YouTube Mentoring Series (https://kubernetes.io/blog/2018/07/10/meet-our-contributors---monthly-streaming-youtube-mentoring-series/) Jul 10

CoreDNS GA for Kubernetes Cluster DNS (https://kubernetes.io/blog/2018/07/10/coredns-ga-for-kubernetes-cluster-dns/) Jul 10

IPVS-Based In-Cluster Load Balancing Deep Dive (https://kubernetes.io/blog/2018/07/09/ipvs-based-in-cluster-load-balancing-deep-dive/) Jul 9

Airflow on Kubernetes (Part 1): A Different Kind of Operator (https://kubernetes.io/blog/2018/06/28/airflow-on-kubernetes-part-1-a-different-kind-of-operator/) Jun 28

Kubernetes 1.11: In-Cluster Load Balancing and CoreDNS Plugin Graduate to General Availability (https://kubernetes.io/blog/2018/06/27/kubernetes-1.11-release-announcement/) Jun 27

Dynamic Ingress in Kubernetes (https://kubernetes.io/blog/2018/06/07/dynamic-ingress-in-kubernetes/) Jun 7

4 Years of K8s (https://kubernetes.io/blog/2018/06/06/4-years-of-k8s/) Jun 6

Say Hello to Discuss Kubernetes (https://kubernetes.io/blog/2018/05/30/say-hello-to-discuss-kubernetes/) May 30

Introducing kustomize; Template-free Configuration Customization for Kubernetes (https://kubernetes.io/blog/2018/05/29/introducing-kustomize-template-free-configuration-customization-for-kubernetes/) May 29

Kubernetes Containerd Integration Goes GA (https://kubernetes.io/blog/2018/05/24/kubernetes-containerd-integration-goes-ga/) May 24

Getting to Know Kubevirt (https://kubernetes.io/blog/2018/05/22/getting-to-know-kubevirt/) May 22

Gardener - The Kubernetes Botanist (https://kubernetes.io/blog/2018/05/17/gardener/) May 17

Docs are Migrating from Jekyll to Hugo (https://kubernetes.io/blog/2018/05/05/hugo-migration/) May 5

Announcing Kubeflow 0.1 (https://kubernetes.io/blog/2018/05/04/announcing-kubeflow-0.1/) May 4

Current State of Policy in Kubernetes (https://kubernetes.io/blog/2018/05/02/policy-in-kubernetes/) May 2

Developing on Kubernetes (https://kubernetes.io/blog/2018/05/01/developing-on-kubernetes/) May 1

Zero-downtime Deployment in Kubernetes with Jenkins (https://kubernetes.io/blog/2018/04/30/zero-downtime-deployment-kubernetes-jenkins/) Apr 30

Kubernetes Community - Top of the Open Source Charts in 2017 (https://kubernetes.io/blog/2018/04/25/open-source-charts-2017/) Apr 25

Kubernetes Application Survey 2018 Results (https://kubernetes.io/blog/2018/04/24/kubernetes-application-survey-results-2018/) Apr 24

Local Persistent Volumes for Kubernetes Goes Beta (https://kubernetes.io/blog/2018/04/13/local-persistent-volumes-beta/) Apr 13

Migrating the Kubernetes Blog (https://kubernetes.io/blog/2018/04/11/migrating-the-kubernetes-blog/) Apr 11

Container Storage Interface (CSI) for Kubernetes Goes Beta (https://kubernetes.io/blog/2018/04/10/container-storage-interface-beta/) Apr 10

Fixing the Subpath Volume Vulnerability in Kubernetes (https://kubernetes.io/blog/2018/04/04/fixing-subpath-volume-vulnerability/) Apr 4

Kubernetes 1.10: Stabilizing Storage, Security, and Networking (https://kubernetes.io/blog/2018/03/26/kubernetes-1.10-stabilizing-storage-security-networking/) Mar 26

Principles of Container-based Application Design (https://kubernetes.io/blog/2018/03/principles-of-container-app-design/) Mar 15

Expanding User Support with Office Hours (https://kubernetes.io/blog/2018/03/expanding-user-support-with-office-hours/) Mar 14

How to Integrate RollingUpdate Strategy for TPR in Kubernetes (https://kubernetes.io/blog/2018/03/how-to-integrate-rollingupdate-strategy/) Mar 13

Apache Spark 2.3 with Native Kubernetes Support (https://kubernetes.io/blog/2018/03/apache-spark-23-with-native-kubernetes/) Mar 6

Kubernetes: First Beta Version of Kubernetes 1.10 is Here (https://kubernetes.io/blog/2018/03/first-beta-version-of-kubernetes-1-10/) Mar 2

Reporting Errors from Control Plane to Applications Using Kubernetes Events (https://kubernetes.io/blog/2018/01/reporting-errors-using-kubernetes-events/) Jan 25

Core Workloads API GA (https://kubernetes.io/blog/2018/01/core-workloads-api-ga/) Jan 15

Introducing client-go version 6 (https://kubernetes.io/blog/2018/01/introducing-client-go-version-6/) Jan 12

Extensible Admission is Beta (https://kubernetes.io/blog/2018/01/extensible-admission-is-beta/) Jan 11

Introducing Container Storage Interface (CSI) Alpha for Kubernetes (https://kubernetes.io/blog/2018/01/introducing-container-storage-interface/) Jan 10

Kubernetes v1.9 releases beta support for Windows Server Containers (https://kubernetes.io/blog/2018/01/kubernetes-v19-beta-windows-support/) Jan 9

Five Days of Kubernetes 1.9 (https://kubernetes.io/blog/2018/01/five-days-of-kubernetes-19/) Jan 8

2017

2016

2015

Documentation (/docs/home/) Blog (/blog/) Partners (/partners/) Community (/community/) Case Studies (/case-studies/)
twitter (https://twitter.com/kubernetesio) Github (https://github.com/kubernetes/kubernetes)
Slack (http://slack.k8s.io/)
Stack Overflow (http://stackoverflow.com/questions/tagged/kubernetes) Forum (https://discuss.kubernetes.io) Events Calendar (https://calendar.google.com/calendar/embed?src=nt2tcnbtbied3l6gi2h29slvc0%40group.calendar.google.com)

Get Kubernetes (/docs/setup/pick-right-solution/) Contribute

(https://git.k8s.io/community/contributors/guide)

- Documentation (/docs/home/)
- Blog (/blog/)
- Partners (/partners/)
- Community (/community/)
- Case Studies (/case-studies/)
- English
- Chinese (/cn/)
- v1.11
- v1.11 (https://kubernetes.io)
- v1.10 (https://v1-10.docs.kubernetes.io)
- v1.9 (https://v1-9.docs.kubernetes.io)
- v1.8 (https://v1-8.docs.kubernetes.io)
- v1.7 (https://v1-7.docs.kubernetes.io)
- Get Started (/docs/tutorials/stateless-application/hello-minikube/)
- Documentation (/docs/home/)
- Community (/community/)
- Blog (/blog/)
- Twitter (https://twitter.com/kubernetesio)
- Github (https://github.com/kubernetes/kubernetes)
- Slack (http://slack.k8s.io/)
- Stack Overflow (http://stackoverflow.com/questions/tagged/kubernetes)
- Forum (https://discuss.kubernetes.io)
- Events Calendar (https://calendar.google.com/calendar/embed?src=nt2tcnbtbied3l6gi2h29slvc0%40group.calendar.google.com)