Sign in

□ Bookmark

< Previous

Learn Learning Paths \vee Certifications \vee FAQ & Help

Next >

100 XP

Exercise - Deploy MongoDB

15 minutes

Unit 4 of 11 ∨

The Fruit Smoothies' ratings website consists of several components. There's a web frontend, a document database that stores captured data, and a RESTful API that allows the web frontend to communicate with the database. The development team is using MongoDB as the document store database of choice for the ratings website.

In this exercise, you'll deploy MongoDB to the Azure Kubernetes Service (AKS) cluster using Helm. You'll also see how to use a Kubernetes secret to store the MongoDB connection username and password.

This example architecture deploys MongoDB on the cluster for the application to use to store data. While this is acceptable for test and development environments, it's not recommended for production environments. For production, it's recommended to store your application state and data in a scalable data storage platform, such as CosmosDB.

In this exercise, you will:

- ✓ Configure the Helm stable repository
- ✓ Install the MongoDB chart
- Create a Kubernetes secret to hold database credentials

Azure Kubernetes Service ratingsapp ratings-mongodb mongosecret Type: ClusterIP ratings-mongodb.ratingsapp.svc.cluster.local

Add the Helm stable repository

Helm is an application package manager for Kubernetes. It offers a way to easily deploy applications and services using charts.

The Helm client is already installed in the Azure Cloud Shell and can be run with the helm command. Helm provides a standard repository of charts for many different software packages. Helm has a chart for MongoDB that is part of the official Helm *stable* charts repository.

1. Configure the Helm client to use the stable repository by running the helm repo add command below.

bash

helm repo add stable https://kubernetes-charts.storage.googleapis.com/

2. You can now list the charts to install by running the helm search repo command. Notice how you can list all charts from the stable channel in the command below.

helm search repo stable

You'll see a list of the available charts, like this example.

output			🖺 Сору
NAME stable/acs-engine-autoscaler stable/aerospike stable/airflow stable/ambassador	CHART VERSION 2.2.2 0.2.8 4.1.0 4.1.0	APP VERSION 2.1.1 v4.5.0.5 1.10.4 0.81.0	DESCRIPTION DEPRECATED Scales worker nodes with A Helm chart for Aerospike in Kuber Airflow is a platform to programmat A Helm chart for Datawire Ambassado

Install a Helm chart

A Helm chart is a collection of files that describe a related set of Kubernetes resources. You can use a single chart to deploy something simple, like a memcached pod, or something complex, like a full web app stack with HTTP servers, databases, and caches.

Helm charts are stored in Helm chart repositories. The official chart repository is maintained on GitHub. The Helm Hub provides a way to discover and view documentation of such charts.

You're now ready to install the MonogoDB instance. Recall from earlier, that you configured your cluster with a ratingsapp namespace. You'll specify the namespace as part of the helm install command, and a name for the database release. The release is called ratings and is deployed into the ratingsapp namespace.

1. Run the helm install command below. Make sure to replace <username> and <password> with appropriate values, and note them for later use.

Keep in mind that the MongoDB connection string is a URI. You have to escape special characters using a standard URI escape mechanism when choosing special characters in the username or password.

helm install ratings stable/mongodb \
--namespace ratingsapp \
--set mongodbUsername=<username>,mongodbPassword=<password>,mongodbDatabase=ratingsdb

mongodbUsername, mongodbPassword, and mongodbDatabase parameters and their values, which set the username, password, and database name, respectively. The application expects that the database is called **ratingsdb**. The helm install command is a powerful command with many capabilities.

2. After the installation is finished, you should get an output similar to this example. Make a note of the MongoDB

You provide parameters with the —set switch and a comma-separated list of key=value pairs. Pay attention to the

host, which should be ratings-mongodb.ratingsapp.svc.cluster.local, if you used the same parameters.

output

Copy

NAME: ratings
LAST DEPLOYED: Mon Dec 30 23:00:47 2019
NAMESPACE: ratingsapp
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
** Please be patient while the chart is being deployed **

MongoDB can be accessed via port 27017 on the following DNS name from within your cluster:
ratings-mongodb.ratingsapp.svc.cluster.local

Keep in mind that you can easily remove a Helm release by running the helm uninstall command. The full

command is helm uninstall ratings —namespace ratingsapp. In this exercise, uninstalling a chart should only be necessary if you made a mistake specifying a non-escaped username or password.

Create a Kubernetes secret to hold the MongoDB details In the previous step, you installed MongoDB using Helm, with a specified username, password, and database name. Now

you'll store these details in a Kubernetes secret. This step ensures that you don't leak secrets by hard coding them into configuration files.

Kubernetes has a concept of secrets. Secrets let you store and manage sensitive information, such as passwords. Putting

this information in a secret is safer and more flexible than hard coding it in a pod definition or a container image.

The ratings API expects to find the connection details to the MongoDB database in the form of mongodb://<username>:

<password>@<endpoint>:27017/ratingsdb. Replace <username>, <password>, and <endpoint> with the ones you
used when you created the database, for example, mongodb://ratingsuser:ratingspassword@ratingsmongodb.ratingsapp.svc.cluster.local:27017/ratingsdb.

1. Use the kubectl create secret generic command to create a secret called mongosecret in the ratingsapp

namespace. A Kubernetes secret can hold several items and is indexed by a key. In this case, the secret contains only one key, called MONGOCONNECTION. The value is the constructed connection string from the previous step. Replace <username> and <password> with the ones you used when you created the database.

bash

kubectl describe secret mongosecret ——namespace ratingsapp

The output from this command looks similar to this example.

output

Name: mongosecret
Namespace: ratingsapp
Labels: <none>
Annotations: <none>

Type: Opaque

Data
====
MONGOCONNECTION: 98 bytes

You now have an AKS cluster with a configured MongoDB database in a namespace called ratingsapp. In this namespace, you'll find the following resources:

- **Deployment/ratings-mongodb**: A deployment represents one or more identical pods managed by the Kubernetes Deployment Controller. This deployment defines the number of replicas (pods) to create for MongoDB. The Kubernetes Scheduler ensures that if pods or nodes encounter problems, additional pods are scheduled on healthy nodes.
- Pod/ratings-mongodb-{random-string}: Kubernetes uses pods to run an instance of MongoDB.
 Service/ratings-mongodb: To simplify the network configuration, Kubernetes uses services to group
- Service/ratings-mongodb: To simplify the network configuration, Kubernetes uses services to group a set of pods and provide network connectivity logically. Connectivity to the MongoDB database is exposed via this service through the DNS name ratings-mongodb.ratingsapp.svc.cluster.local.
 Secret/mongosecret: A Kubernetes secret is used to inject sensitive data into pods, such as access credentials or
- communicate with MongoDB.

keys. This secret holds the MongoDB connection details. You'll use it in the next unit to configure the API to

In this exercise, you configured the Helm stable repository, then used a Helm chart to deploy MongoDB to your cluster.

Summary

Next, you'll deploy the Fruit Smoothies ratings-api to your AKS cluster.

You then created a Kubernetes secret to hold database credentials.

Next unit: Exercise - Deploy the ratings API

Continue >

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue.

🗅 Сору