Kubernetes with Containers and DevOps Workshop

Hands-on lab step-by-step

Aralık 2018

10-Moving your data services to Hosted Data Solutions (CosmosDB)

In this section we will be creating a CosmosDB instance in your Azure account to migrate/export your MongoDB data to CosmosDB. You can use CosmosDB as a drop in replacement for MongoDB, since CosmosDB uses a MongoDB compatibale API. As such, you are only required to replace/change the MongoDB URI connection string with one supplied by CosmosDB in the dashboard.

**Setup CosmosDB**

You can create a CosmosDB service/instance in one (1) of two (2) ways:

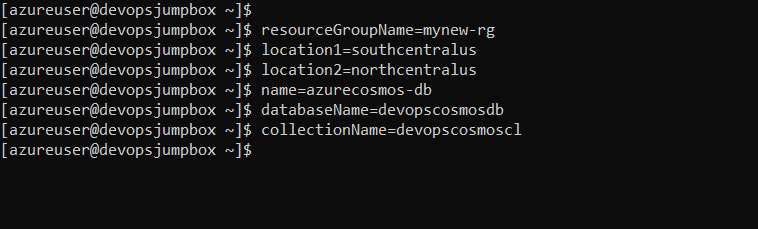
1. Via the [Azure-CLI](https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest) command line tool --**OR**--
2. Via the [Azure Web Portal](https://portal.azure.com/)

**Method 1: via Azure-CLI**

The Azure-CLI Command Line Tool is available and supported for Windows, macOS and Linux. The following uses the cross plaform Azure-CLI in a Linux bash shell to deploy an instance of CosmosDB into your Azure Subscription/Account.

Note:

* We are leveraging Linux bash environment variables to script our deployment. You should change **ALL** the values for each variable listed to fit your preferred deployment. As an example location1 and location2 should be changed to your desired Azure datacentre.
* The \ slashes in the below example are used for escaping new lines for readability purposes (they are not a requirement), as such you may remove them to form a single line command.
  + 1. # !!!Set variables for the new account, database, and collection etc.!!!
    2. resourceGroupName=mynewcosmosdb-rg
    3. location1=southcentralus
    4. location2=northcentralus
    5. name=mstrazurecosmos-db
    6. databaseName=mstrcosmosdb
    7. collectionName=mstrcosmoscl

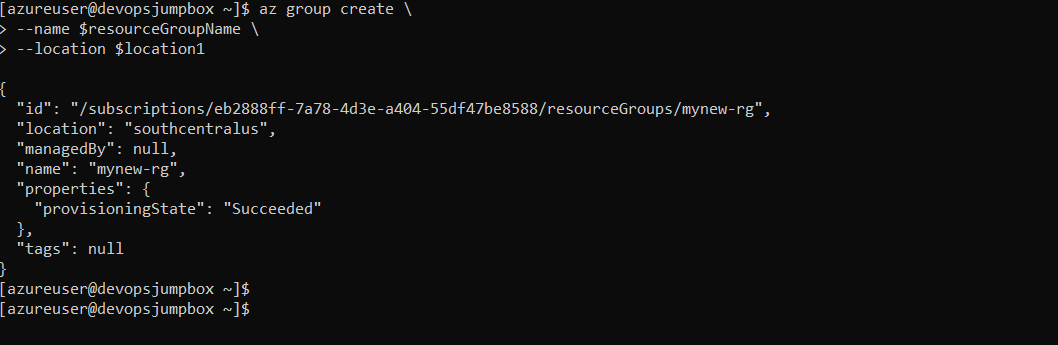


# Create a resource group

* + 1. az group create \

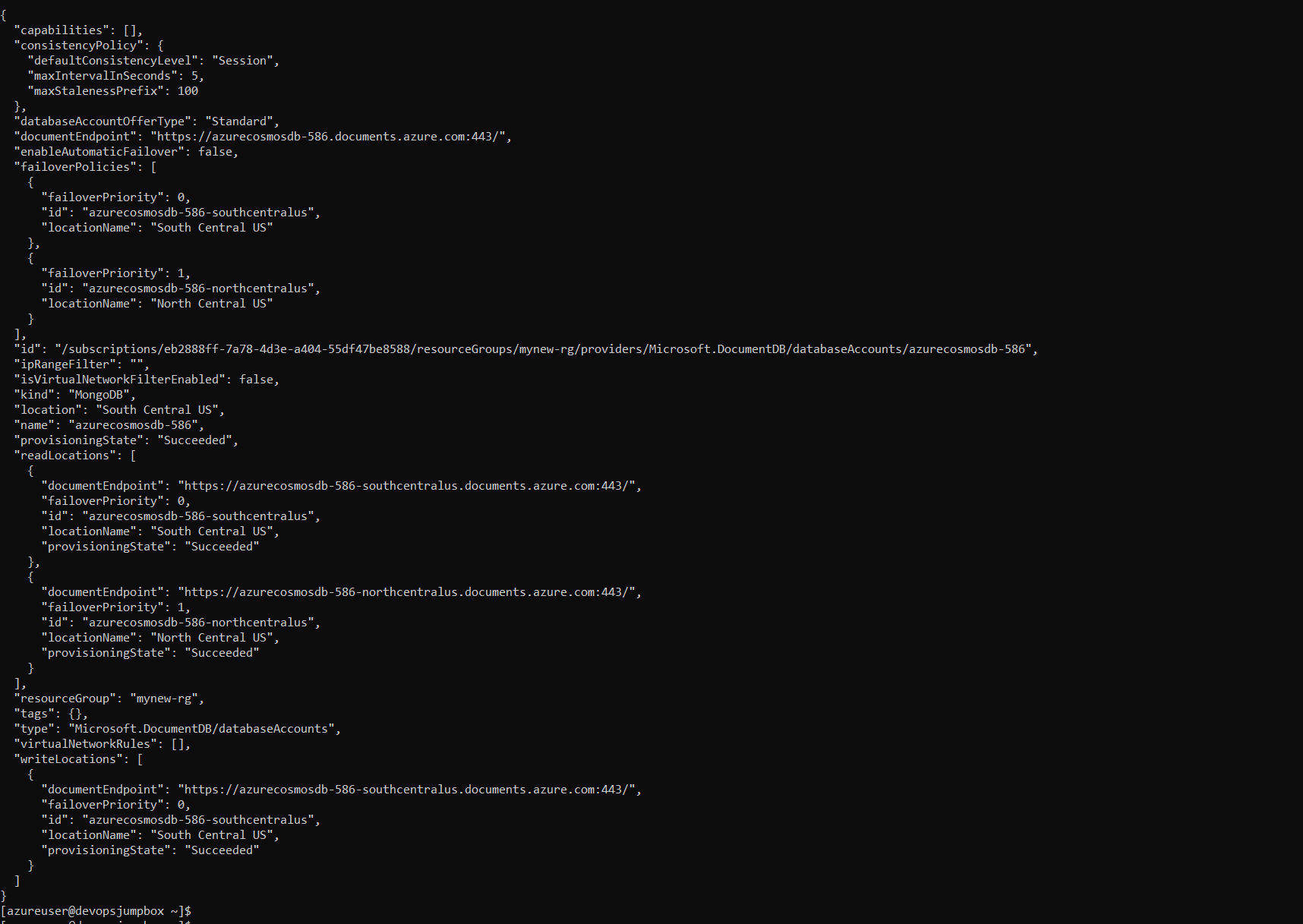
--name $resourceGroupName \

* + 1. --location $location1



# Create a MongoDB API Cosmos DB account

* + 1. az cosmosdb create \
    2. --name $name \
    3. --kind MongoDB \
    4. --locations "$location1"=0 "$location2"=1 \
    5. --resource-group $resourceGroupName \
    6. --max-interval 10 \
    7. --max-staleness-prefix 200



Once you've run the above command, your CosmosDB instance will be provisioned within minutes. When the deployment is sucessful you will see output in your terminal with information about your CosmosDB deployment.

Migrating Data From MongoDB to CosmosDB

In this section we will use the **mongodump** and **mongorestore** commands to export data from MongoDB and then import back into CosmosDB.

The method used in this section is simple by design and may not be the right method for your production migrations. We would recommend working with a MongoDB DBA and Architect who is experienced with MongoDB data migration in a production environment to minimize downtime. Please refer to: [Guide for a successful migration](https://docs.microsoft.com/en-us/azure/cosmos-db/mongodb-migrate#guide-for-a-successful-migration)in the Azure Docs for CosmosDB.

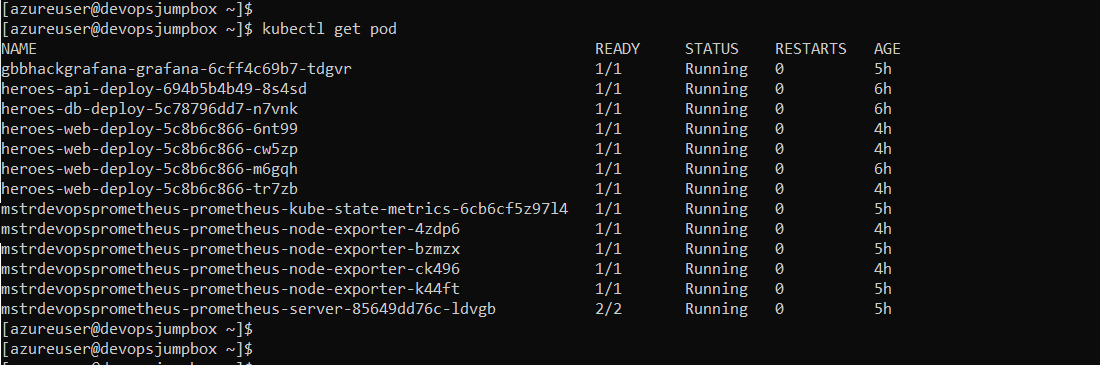
**Export/Import Data**

These commands will be run from inside our MongoDB pod.

1. Exec into mongo database pod and export data

# list pods in the cluster and set the variable to your pod name

kubectl get pod

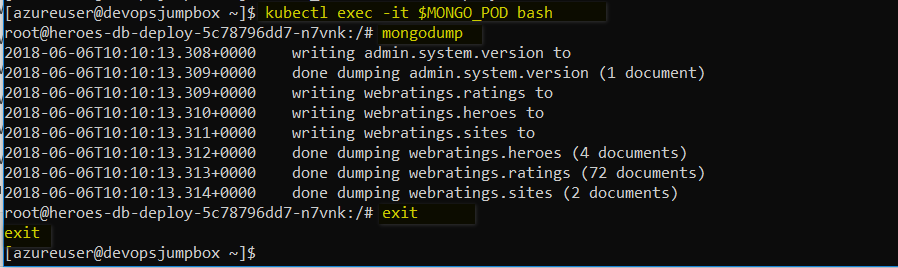


MONGO\_POD=heroes-db-deploy-2357291595-xb4xm



kubectl exec -it $MONGO\_POD bash

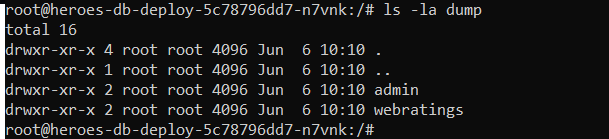
root@heroes-db-deploy-2357291595-xb4xm:/# **mongodump**



validate the the export was successful

root@heroes-db-deploy-6ccb88c7bc-cfmqh:/#

ls -la dump



**Do not exit the pod. Step 3 will be run from the same location.**

1. Import the data using mongorestore

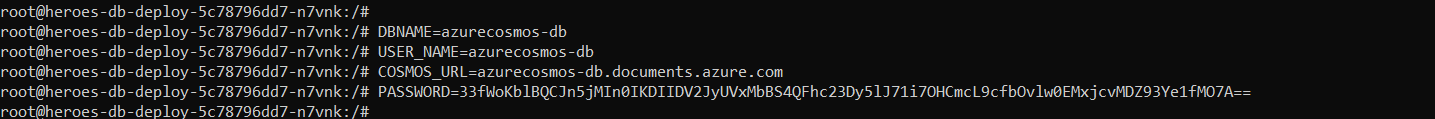
Retrieve the CosmosDB connection details from the Azure Portal

Click on Connection String and make note of **the HOST,USERNAME**, and **PRIMARY PASSWORD**

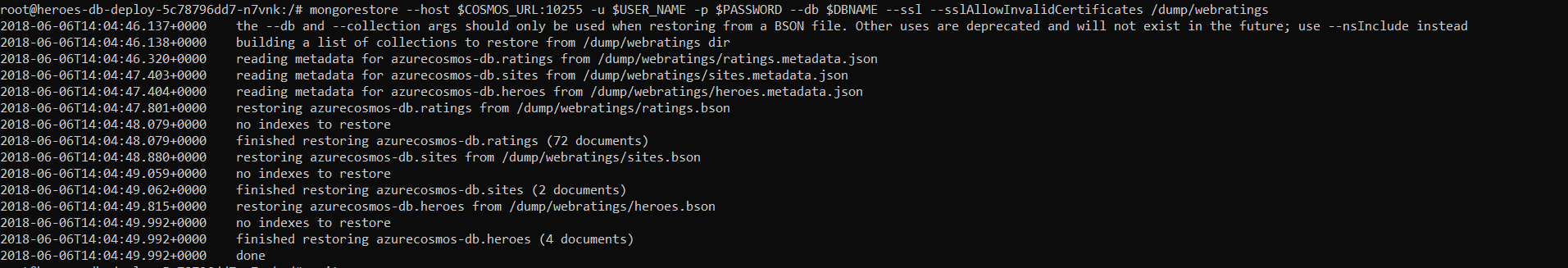
From the same prompt inside the pod, set the variables below and run the command.

# set the below environment variables inside the mongoDB pod

* + 1. DBNAME=mstrazurecosmos-db
    2. USER\_NAME=mstrazurecosmos-db
    3. COSMOS\_URL=mstrazurecosmos-db.documents.azure.com
    4. PASSWORD=ee751qSQdm4MzFfdbTuTgQtOAAb2YdpZuyivegIHYWc9Z9B0rJNEomrkLebISsuYEdRtp4H1nhbdOheC3VyzoQ==



mongorestore --host $COSMOS\_URL:10255 -u $USER\_NAME -p $PASSWORD --db $DBNAME --ssl --sslAllowInvalidCertificates /dump/webratings



1. Exit from pod by typing

exit



**View data the Azure Portal**

When you navigate to your CosmosDB instance in the Azure portal, you can view your Database, Collections and Documents by navigating to the "Data Explorer" section of your CosmosDB instance. You can perform CRUD, Query and other operations on your database here as well.

