



# Contents

1.	About Test Drive .....	3
2.	What is MariaDB? .....	3
3.	About MariaDB Enterprise Cluster in Azure.....	3
3.1.	Architecture .....	3
3.2.	Objective.....	4
3.3.	Administering the cluster .....	4
4.	Getting Started.....	4
4.1.	Install WordPress.....	5
4.2.	Administrating MariaDb Enterprise Cluster.....	7
4.2.1.	Scenario 1: Making one instance MaxScaleVM1 unavailable .....	7
4.2.2.	Scenario 2: Making both the instances of MaxScale unavailable and bring up one instance again.....	11

## 1. About Test Drive

Welcome to the MariaDB test drive on Azure!

In this 2 hours' test drive, we will dive in and see what MariaDB can do. This User Guide walks you through the steps on how the MariaDB Enterprise Cluster + MariaDB MaxScale solution exhibit Highly Available (HA).

## 2. What is MariaDB?

MariaDB is one of the most popular database servers in the world. MariaDB is developed as open source software and as a relational database, it provides an SQL interface for accessing data.

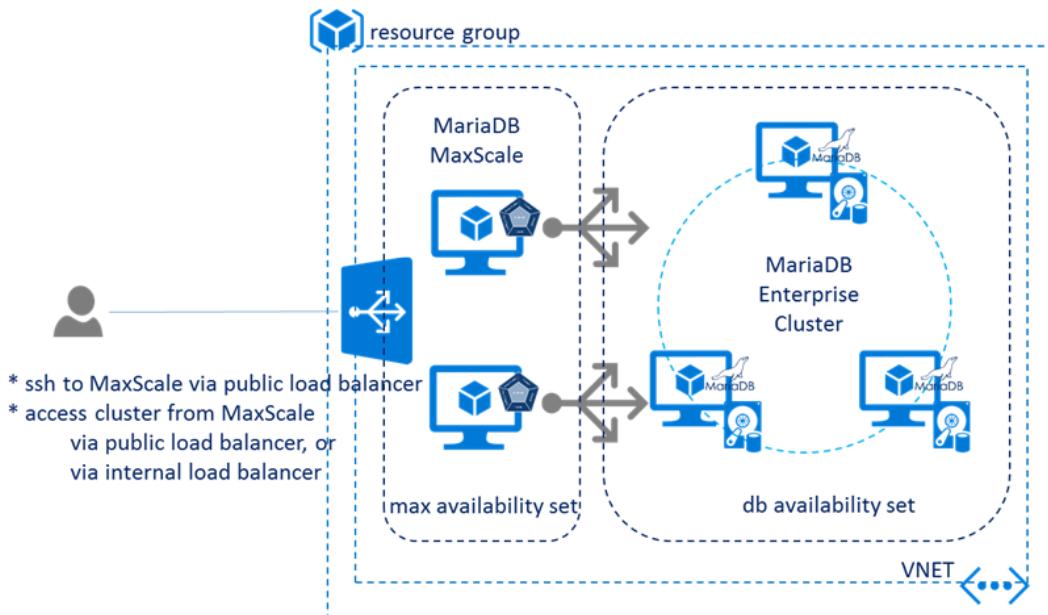
MariaDB turns into structured information in a wide array of applications, ranging from banking to websites. It is an enhanced, drop-in replacement for MySQL. MariaDB is used because it is fast, scalable and robust, with a rich ecosystem of storage engines, plugins and many other tools make it very versatile for a wide variety of use cases.

## 3. About MariaDB Enterprise Cluster in Azure

**MariaDB Enterprise Cluster** extends MariaDB, the widely adopted, MySQL-compatible open source database with Galera clustering technology. **MariaDB MaxScale** offers connection- and statement-based load balancing.

The MariaDB Enterprise Cluster + MariaDB MaxScale solution for Azure consists of a 3-node MariaDB Enterprise Cluster and dual MariaDB MaxScale nodes in a Highly Available (HA) configuration.

### 3.1. Architecture



- Developers and DBAs can now provision a 3-node production-ready MariaDB Enterprise Cluster with MariaDB MaxScale®, using virtual machines of sizes D2/DS2 and up.
- MariaDB Enterprise Cluster with Galera technology is a multi-master cluster that achieves high availability and scalability through features such as synchronous

replication, read and write to any cluster nodes without slave lag, automatic membership control of new nodes joining the cluster and failed nodes dropping from the cluster.

- MariaDB MaxScale is a database gateway that insulates client applications from the complexities of a database cluster.
- In addition to load balancing, MariaDB MaxScale also provides logging, filtering, and monitoring mechanisms.
- The MaxScale VMs of this architecture are two A series virtual machines with no data disks attached. An Azure load balancer is placed in front of the MaxScale VMs to achieve high availability.
- The VMs can be accessed through the load balancers public IP or DNS.

### 3.2. Objective

The MariaDB Enterprise Cluster offering in Microsoft Azure deploys 3 data/Galera nodes and 2 MariaDB MaxScale nodes, all running CentOS 7. MariaDB MaxScale in this solution is automatically configured to provide three services: **RW Split Router** (Read/Write Split) on port 4006, **Write Connection Router** (to the "Master" node) on port 4007, and **Read Connection Router** (to the "Slave" nodes) on port 4008.

A Client VM is also part of this deployment. This VM has been installed and configured with WordPress workload, which communicates with the MariaDB MaxScale VM's.

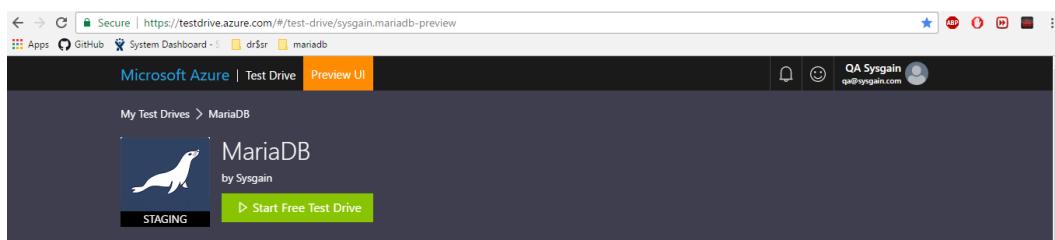
The 2 MaxScale nodes are set up behind an Azure Load Balancer, configure to automatically fail over if one instance of MaxScale becomes unavailable.

### 3.3. Administering the cluster

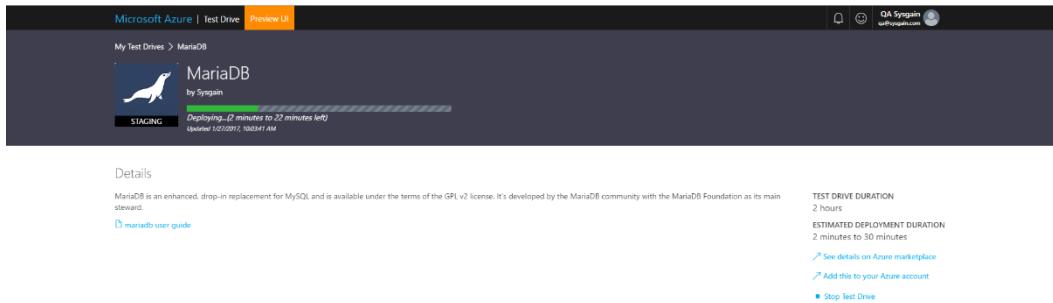
You can use SSH to connect directly to the MaxScale nodes from any host on the Internet. Because there are 2 MaxScale nodes, and both are accessed through the same public-facing hostname, **non-standard ports are used for SSH**. To SSH to the max1 node, use **port 220**, to SSH to the max2 node, use **port 2202**.

## 4. Getting Started

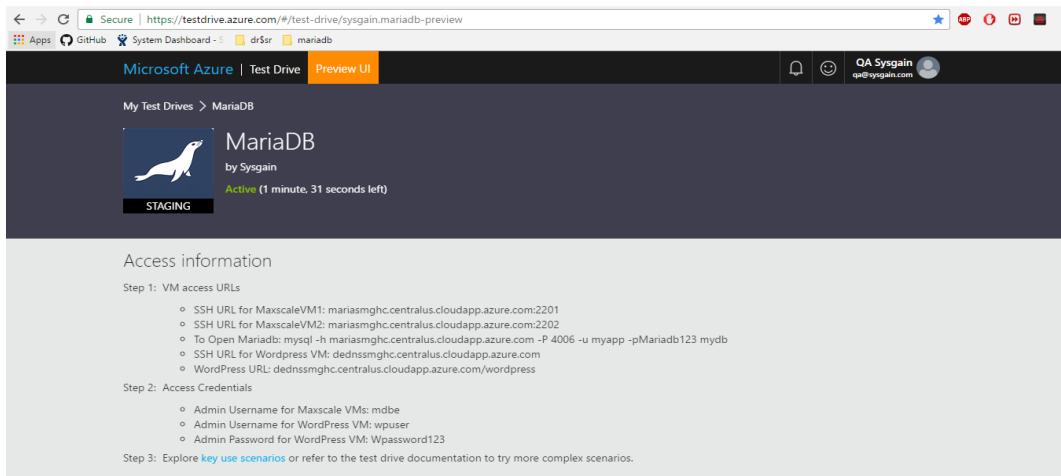
1. Once you have signed in to Microsoft Test Drive Portal you will see the launch screen as shown below. Now you are ready to launch the test drive by clicking the '**Start Free Test Drive**'.



- Once you click on Start Free Test Drive button to deploy the test drive, it takes about 2-30 minutes to launch the test drive.



- Once the Test Drive is ready, required URLs and Login Credentials will be shown on the web page under “Access information” as well as by e-mail. We can find the duration of test drive on the top section of the Test Drive launch page.



## 4.1. Install WordPress

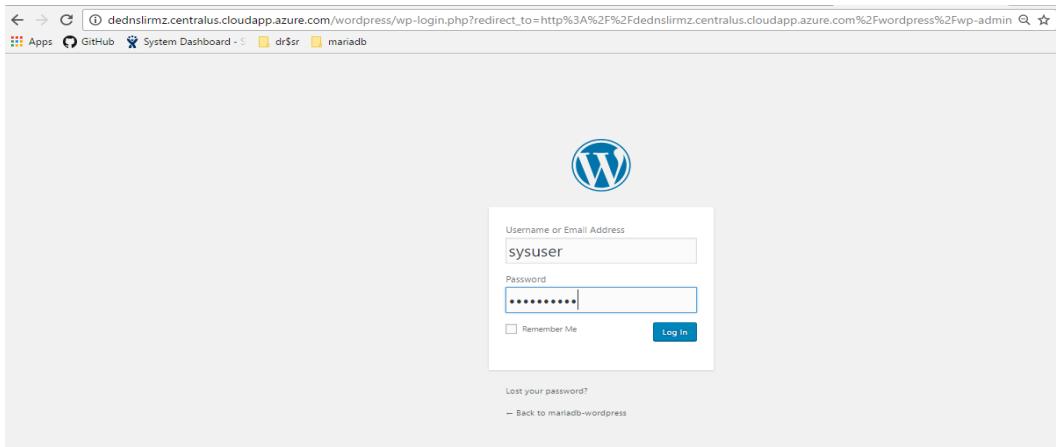
- Copy the WordPress URL found in Access Information section of test-drive and paste it in a browser.

The screenshot shows the initial WordPress installation screen. At the top, the URL is `dednsfirmz.centralus.cloudapp.azure.com/wordpress/wp-admin/install.php`. Below the header, there's a large blue 'W' logo. The main title is 'Welcome' with a sub-instruction: 'Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.' A section titled 'Information needed' asks for Site Title, Username, Password, Your Email, and Search Engine Visibility. The 'Site Title' field contains 'dr\$sr'. The 'Username' field contains 'dr\$sr'. The 'Password' field contains '3dBoxBUZkBV11TtP' and is marked as 'Strong'. The 'Your Email' field contains 'dr\$sr@gmail.com'. Under 'Search Engine Visibility', there's a checkbox for 'Discourage search engines from indexing this site'. At the bottom is a blue 'Install WordPress' button.

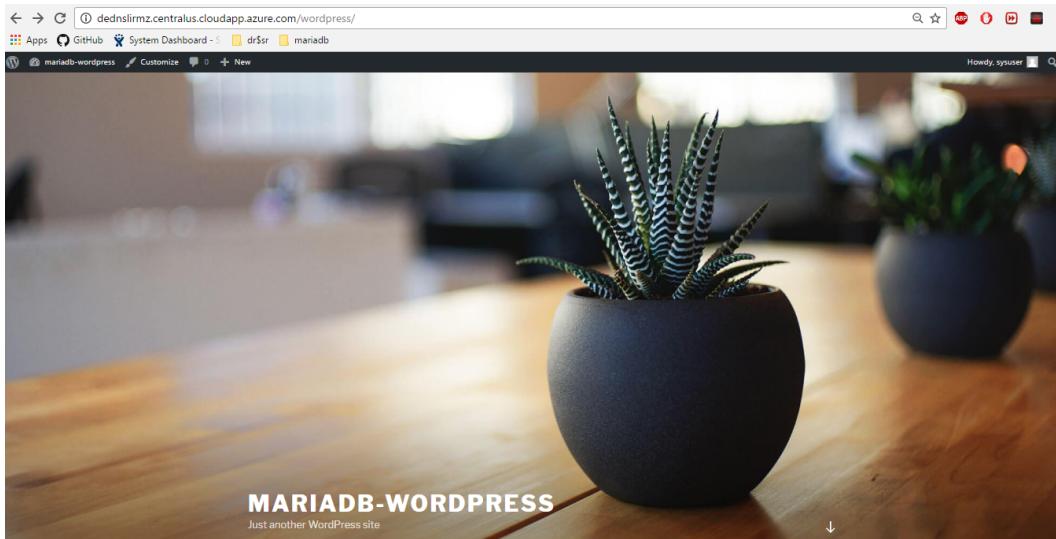
2. To install WordPress, fill in the details and click on 'Install WordPress'.

This screenshot shows the same WordPress installation screen after filling in the details. The 'Site Title' is now 'mariadb-wordpress', the 'Username' is 'sysuser', the 'Password' is 'syspwd@123', and the 'Your Email' is 'sysuser@gmail.com'. All other fields and instructions remain the same as in the first screenshot.

3. Login to your WordPress account with your WordPress Credentials and access the WordPress.



4. Now you can create and edit the WordPress default website as you like.



## 4.2. Administrating MariaDb Enterprise Cluster

To demonstrate the High Availability of MariaDB, we have designed two scenarios

**Scenario 1:** Making one instance of MaxScale unavailable

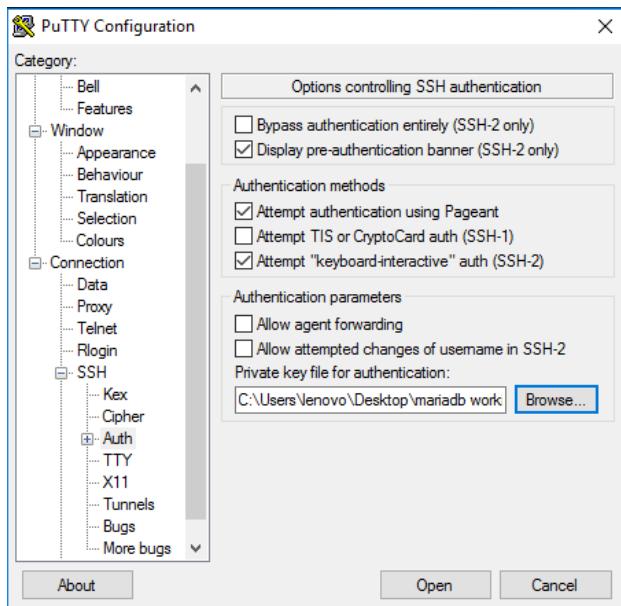
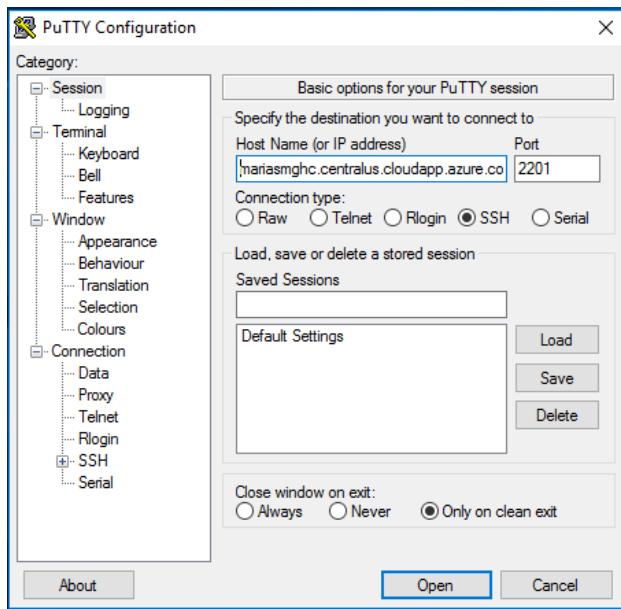
**Scenario 2:** Making both the instances of MaxScale unavailable and bringing up one instance again.

### 4.2.1. Scenario 1: Making one instance MaxScaleVM1 unavailable

#### For Windows

SSH into MaxScaleVM1 using the DNS provided in Test Drive launch page and port 2201 with **ssh Key for Windows** provided on Test Drive launch page.

(Copy, paste the key URL in a new browser, which automatically downloads the key. Save the key on desktop, which can be used during SSH into putty for MaxScale VM's).



## For Mac

SSH into MaxScaleVM1 using the DNS provided in Test Drive launch page and port 2201 with the **ssh Key for Mac** provided on the Test Drive launch page .

Download the key (.pem) from the link provided on Test Drive launch page to your local machine

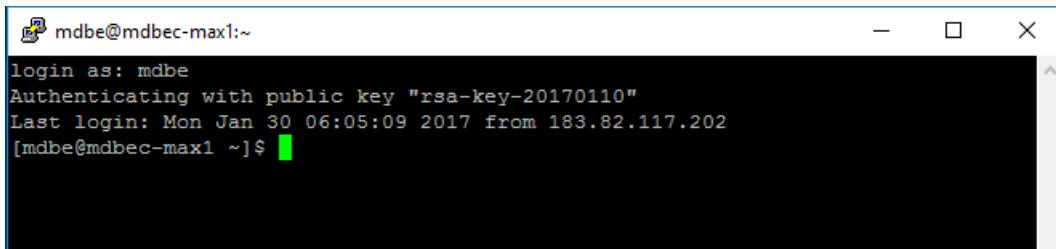
Open the terminal go to the location where the key is downloaded and run following command

**chmod 400 mariadb\_privatekey.pem**

```
Last login: Tue Feb 28 11:12:57 on ttys000
[Udays-MacBook-Pro:~ d-b$ cd Desktop/
[Udays-MacBook-Pro:Desktop d-b$ chmod 400 mariadb_privatekey.pem]
```

1. Login into the MaxScaleVM1 with the given credentials in the Test Drive launch page.

#### For Windows



A screenshot of a Windows Command Prompt window titled "mdbe@mdbec-max1:~". The window shows the user has logged in as "mdbe" and is authenticating with a public key "rsa-key-20170110". The last login was on Monday, January 30, 2017, at 06:05:09 from IP address 183.82.117.202. The prompt "[mdbe@mdbec-max1 ~]\$" is visible at the bottom.

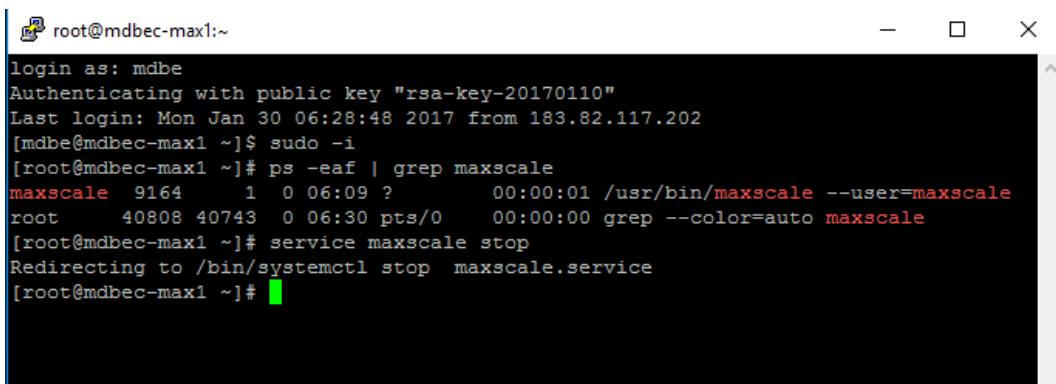
#### For Mac

Enter the following command in the terminal where the key (.pem) is located.

```
ssh -i mariadb_privatekey.pem mdbe@<DNS Name on Launch Page> -p 2201
```

```
[Udays-MacBook-Pro:~ d-b$ cd Desktop/
[Udays-MacBook-Pro:Desktop d-b$ chmod 400 mariadb_privatekey.pem
[Udays-MacBook-Pro:Desktop d-b$ ssh -i mariadb_privatekey.pem mdbe@mariazqeqq.centralus.cloudapp.azure.com -p 2201]
```

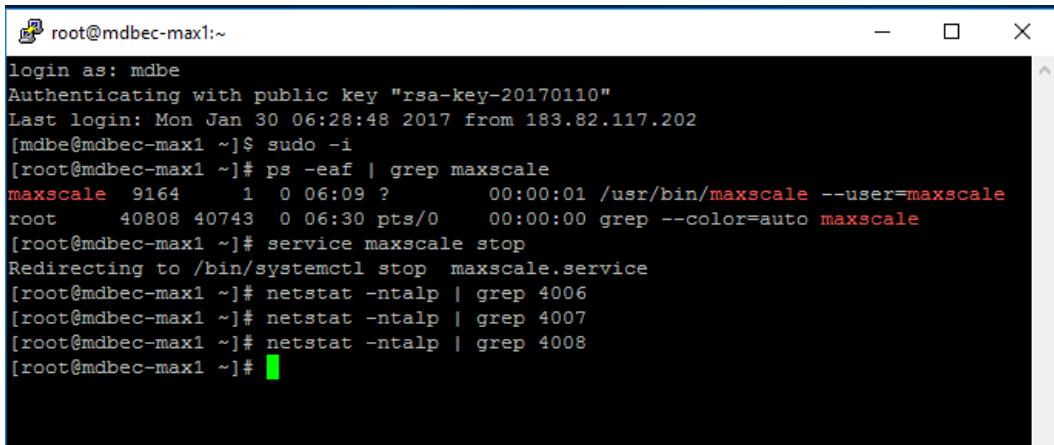
2. Change to root user by using **sudo -i**. Enter **ps -eaf | grep maxscale** command to check for the process status of MaxScale and then stop the MaxScale service by using **service maxscale stop**.



A screenshot of a Mac Terminal window titled "root@mdbec-max1:~". The user has logged in as "root" and is authenticating with a public key "rsa-key-20170110". The last login was on Monday, January 30, 2017, at 06:28:48 from IP address 183.82.117.202. The user runs the command "ps -eaf | grep maxscale" which shows a process named "maxscale" with PID 9164. The user then runs "service maxscale stop" which stops the MaxScale service. The prompt "[root@mdbec-max1 ~]#" is visible at the bottom.

3. The following commands lets us know what ports are open and whether any programs are listening on ports after the service has been stopped.

```
netstat -ntalp | grep 4006  
netstat -ntalp | grep 4007  
netstat -ntalp | grep 4008
```



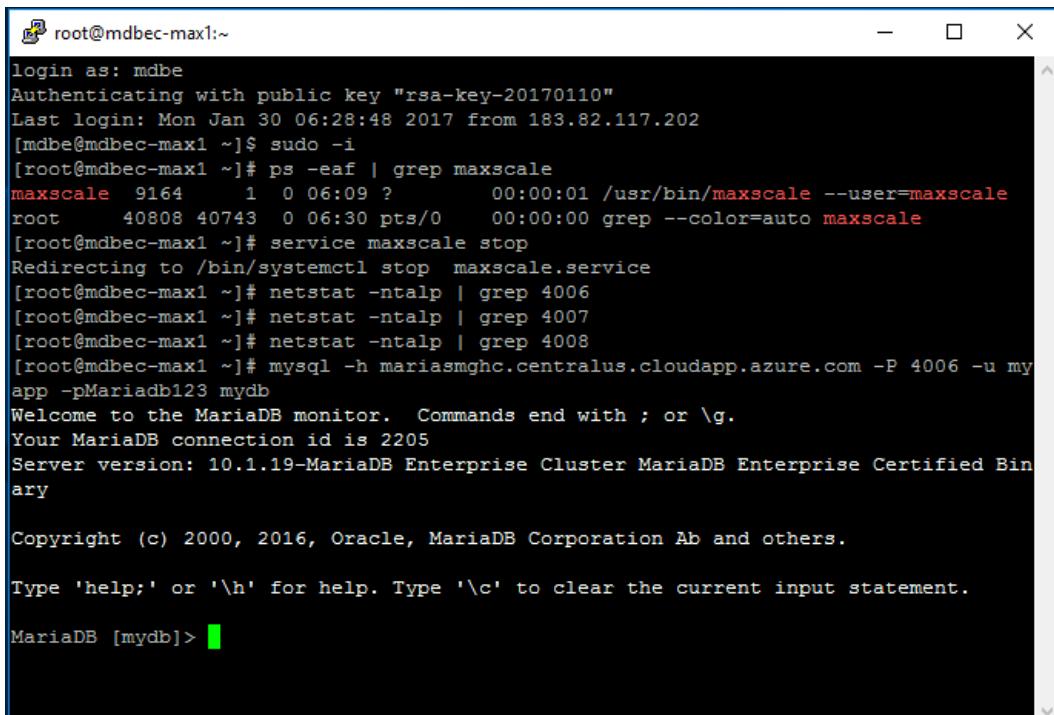
The screenshot shows a terminal window with the following command history:

```
root@mdbec-max1:~  
login as: mdbe  
Authenticating with public key "rsa-key-20170110"  
Last login: Mon Jan 30 06:28:48 2017 from 183.82.117.202  
[mdbec@mdbec-max1 ~]$ sudo -i  
[root@mdbec-max1 ~]# ps -eaf | grep maxscale  
maxscale 9164 1 0 06:09 ? 00:00:01 /usr/bin/maxscale --user=maxscale  
root 40808 40743 0 06:30 pts/0 00:00:00 grep --color=auto maxscale  
[root@mdbec-max1 ~]# service maxscale stop  
Redirecting to /bin/systemctl stop maxscale.service  
[root@mdbec-max1 ~]# netstat -ntalp | grep 4006  
[root@mdbec-max1 ~]# netstat -ntalp | grep 4007  
[root@mdbec-max1 ~]# netstat -ntalp | grep 4008  
[root@mdbec-max1 ~]#
```

4. Though one instance of MaxScale becomes unavailable, we can still access the service using **To Open MariaDB of VM access URLs** from Access information provided in Test Drive which shows High Availability (in this case:

```
mysql -h mariasmghc.centralus.cloudapp.azure.com -P 4006 -u myapp  
-pMariadb123 mydb).
```

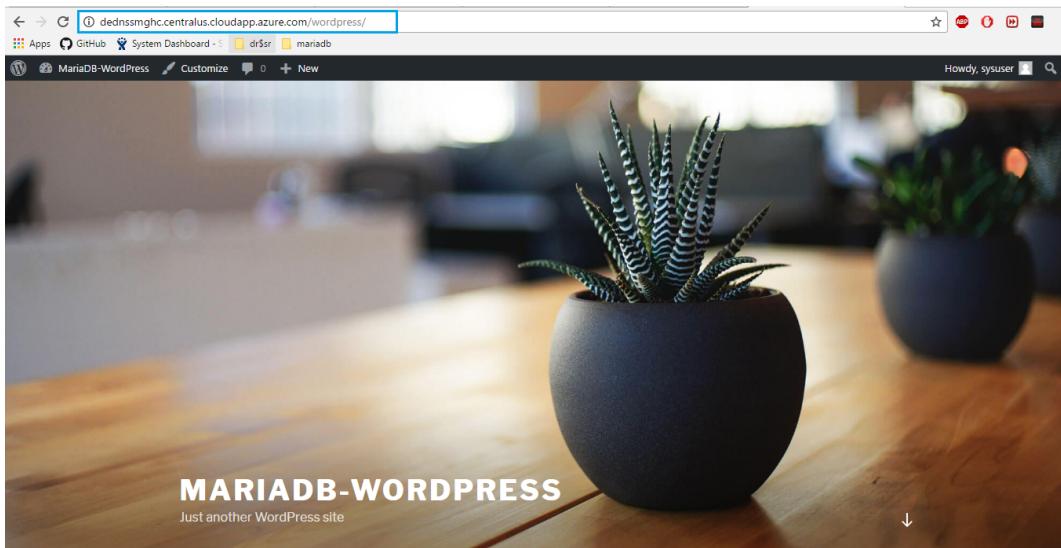
If we get **MariaDB [mydb]** on using the above command, this shows that the service is ON.



The screenshot shows a terminal window with the following command history and MySQL prompt:

```
root@mdbec-max1:~  
login as: mdbe  
Authenticating with public key "rsa-key-20170110"  
Last login: Mon Jan 30 06:28:48 2017 from 183.82.117.202  
[mdbec@mdbec-max1 ~]$ sudo -i  
[root@mdbec-max1 ~]# ps -eaf | grep maxscale  
maxscale 9164 1 0 06:09 ? 00:00:01 /usr/bin/maxscale --user=maxscale  
root 40808 40743 0 06:30 pts/0 00:00:00 grep --color=auto maxscale  
[root@mdbec-max1 ~]# service maxscale stop  
Redirecting to /bin/systemctl stop maxscale.service  
[root@mdbec-max1 ~]# netstat -ntalp | grep 4006  
[root@mdbec-max1 ~]# netstat -ntalp | grep 4007  
[root@mdbec-max1 ~]# netstat -ntalp | grep 4008  
[root@mdbec-max1 ~]# mysql -h mariasmghc.centralus.cloudapp.azure.com -P 4006 -u myapp -pMariadb123 mydb  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 2205  
Server version: 10.1.19-MariaDB Enterprise Cluster MariaDB Enterprise Certified Binary  
  
Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [mydb]>
```

5. Alternatively,we can also check the MariaDB Database connectivity by using the WordPress URL provided in Access Information and run it in the Browser.



6. You can start the service again by using **service maxscale start** command.

```
root@mdbec-max1:~#
Last login: Mon Jan 30 06:28:48 2017 from 183.82.117.202
[mdbec@mdbec-max1 ~]$ sudo -i
[root@mdbec-max1 ~]# ps -eaf | grep maxscale
maxscale 9164 1 0 06:09 ? 00:00:01 /usr/bin/maxscale --user=maxscale
root 40808 40743 0 06:30 pts/0 00:00:00 grep --color=auto maxscale
[root@mdbec-max1 ~]# service maxscale stop
Redirecting to /bin/systemctl stop maxscale.service
[root@mdbec-max1 ~]# netstat -ntalp | grep 4006
[root@mdbec-max1 ~]# netstat -ntalp | grep 4007
[root@mdbec-max1 ~]# netstat -ntalp | grep 4008
[root@mdbec-max1 ~]# mysql -h mariasmghc.centralus.cloudapp.azure.com -P 4006 -u myapp -pMariadb123 mydb
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 2205
Server version: 10.1.19-MariaDB Enterprise Cluster MariaDB Enterprise Certified Binary

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

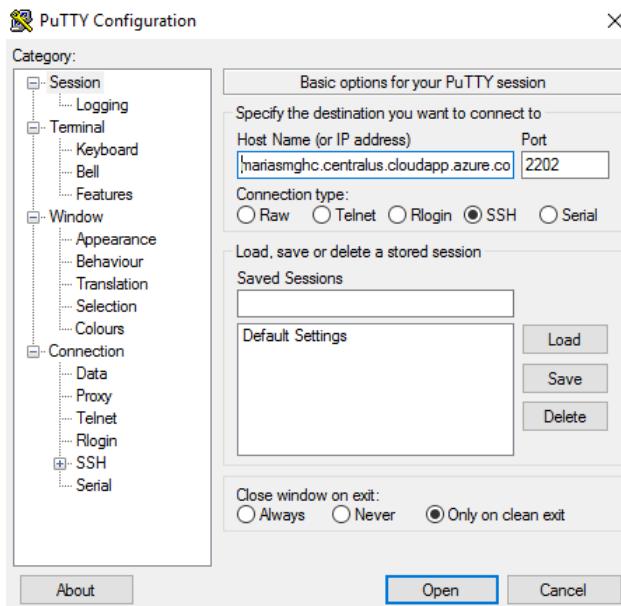
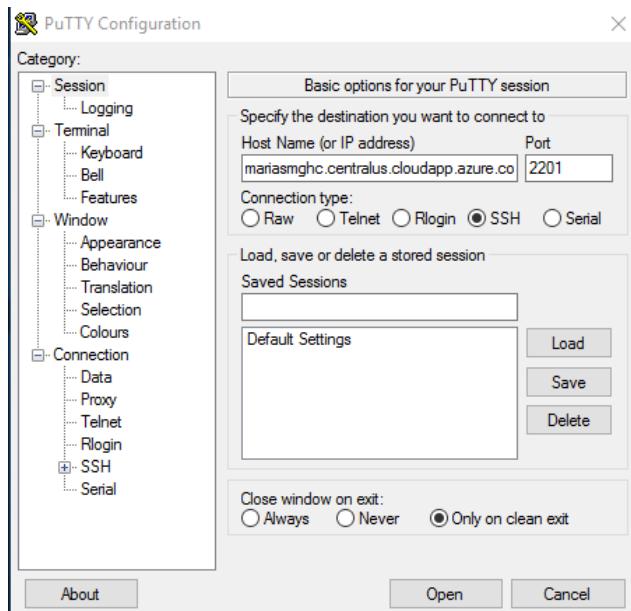
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [mydb]>
[1]+ Stopped                  mysql -h mariasmghc.centralus.cloudapp.azure.com -P 40
[root@mdbec-max1 ~]#
[root@mdbec-max1 ~]# service maxscale start
Redirecting to /bin/systemctl start maxscale.service
[root@mdbec-max1 ~]#
```

4.2.2. Scenario 2: Making both the instances of MaxScale unavailable and bring up one instance again.

#### For Windows

SSH into MaxScaleVM1 (port 2201) and MaxScaleVM2 (port 2202) using the DNS provided in the Access Information from the Test Drive with the key (for Windows).



## For Mac

SSH into MaxScaleVM1 (port 2201) and MaxScaleVM2 (port 2202) using the DNS provided in the Access Information from the Test Drive with the key (.pem for Mac).

```
[Udays-MacBook-Pro:~ d-b$ cd Desktop/
[Udays-MacBook-Pro:Desktop d-b$ chmod 400 mariadb_privatekey.pem
[Udays-MacBook-Pro:Desktop d-b$ ssh -i mariadb_privatekey.pem mdbe@mariazqeinq.centralus.cloudapp.azure.com -p 2201
```

```
[Udays-MacBook-Pro:Desktop d-b$ chmod 400 mariadb_privatekey.pem
[Udays-MacBook-Pro:Desktop d-b$ ssh -i mariadb_privatekey.pem mdbe@mariazqeinq.centralus.cloudapp.azure.com -p 2202
```

1. Change the user to root by using **sudo -i**. Enter **ps -eaf | grep maxscale** command to check for the process status of MaxScale and then stop the MaxScale service by using service stop command i.e. **service maxscale stop** in both the VM's.

```

root@mdbec-max1:~ login as: mdbe
Authenticating with public key "rsa-key-20170110"
Last login: Mon Jan 30 06:54:18 2017 from 183.82.117.202
[mdbe@mdbec-max1 ~]$ sudo -i
[root@mdbec-max1 ~]# ps -eaf | grep maxscale
root      47025 46976  0 06:57 pts/0    00:00:00 grep --color=auto maxscale
[root@mdbec-max1 ~]# service maxscale stop
Redirecting to /bin/systemctl stop maxscale.service
[root@mdbec-max1 ~]# netstat -ntalp | grep 4006
[root@mdbec-max1 ~]# netstat -ntalp | grep 4007
[root@mdbec-max1 ~]# netstat -ntalp | grep 4008
[root@mdbec-max1 ~]#

```

```

root@mdbec-max2:~ login as: mdbe
Authenticating with public key "rsa-key-20170110"
Last login: Mon Jan 30 06:44:56 2017 from 183.82.117.202
[mdbe@mdbec-max2 ~]$ sudo -i
[root@mdbec-max2 ~]# ps -eaf | grep maxscale
maxscale 44556      1  0 06:48 ?          00:00:00 /usr/bin/maxscale --user=maxscale
root      46012 45902  0 06:54 pts/0    00:00:00 grep --color=auto maxscale
[root@mdbec-max2 ~]# service maxscale stop
Redirecting to /bin/systemctl stop maxscale.service
[root@mdbec-max2 ~]# netstat -ntalp | grep 4006
[root@mdbec-max2 ~]# netstat -ntalp | grep 4007
[root@mdbec-max2 ~]# netstat -ntalp | grep 4008
[root@mdbec-max2 ~]#

```

2. Check whether we can still access the MariaDB service using

**To Open MariaDB of VM access URLs** from Access information provided in Test Drive which shows High Availability (in this case:

**mysql -h mariasmghc.centralus.cloudapp.azure.com -P 4006 -u myapp pMariadb123 mydb).**

```

root@mdbec-max1:~ login as: mdbe
Authenticating with public key "rsa-key-20170110"
Last login: Mon Jan 30 06:54:18 2017 from 183.82.117.202
[mdbe@mdbec-max1 ~]$ sudo -i
[root@mdbec-max1 ~]# ps -eaf | grep maxscale
root      47025 46976  0 06:57 pts/0    00:00:00 grep --color=auto maxscale
[root@mdbec-max1 ~]# service maxscale stop
Redirecting to /bin/systemctl stop maxscale.service
[root@mdbec-max1 ~]# netstat -ntalp | grep 4006
[root@mdbec-max1 ~]# netstat -ntalp | grep 4007
[root@mdbec-max1 ~]# netstat -ntalp | grep 4008
[root@mdbec-max1 ~]# mysql -h mariasmghc.centralus.cloudapp.azure.com -P 4006 -u myapp -pMariadb123 mydb

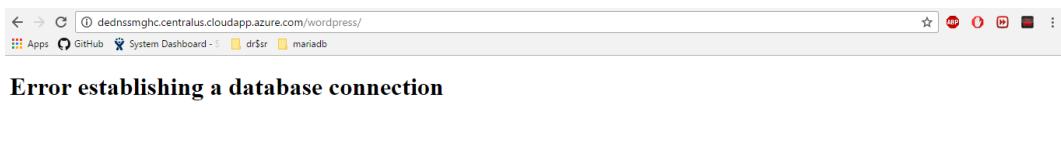
```

```

root@mdbec-max2:~ login as: mdbec
Authenticating with public key "rsa-key-20170110"
Last login: Mon Jan 30 06:44:56 2017 from 183.82.117.202
[mdbec@mdbc... ~]$ sudo -i
[root@mdbc... ~]# ps -eaf | grep maxscale
maxscale 44556      1  0 06:48 ?          00:00:00 /usr/bin/maxscale --user=maxscale
root      46012 45902  0 06:54 pts/0    00:00:00 grep --color=auto maxscale
[root@mdbc... ~]# service maxscale stop
Redirecting to /bin/systemctl stop maxscale.service
[root@mdbc... ~]# netstat -ntalp | grep 4006
[root@mdbc... ~]# netstat -ntalp | grep 4007
[root@mdbc... ~]# netstat -ntalp | grep 4008
[root@mdbc... ~]# mysql -h mariasmghc.centralus.cloudapp.azure.com -P 4006 -u myapp -pMariadb123 mydb

```

3. Alternatively, we can also check the MaxScale Database service by using the WordPress URL provided in Access Information and run it in the Browser.



4. Start the MaxScale service in one of the MaxScale VM (MaxScaleVM1) using **service maxscale start** command and let the other remain same (without starting the MaxScale service) and check for the MariaDB service.

```

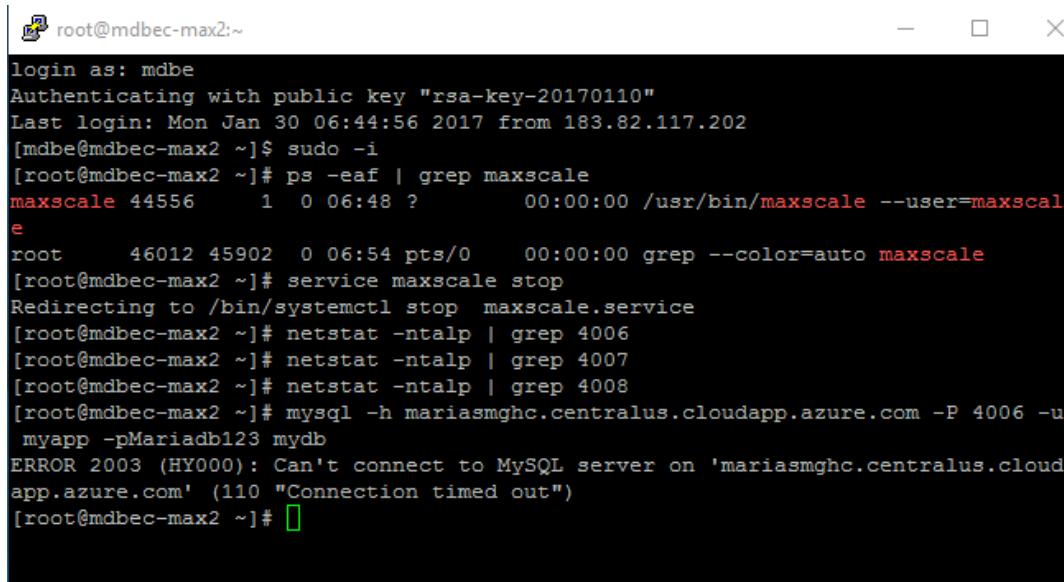
root@mdbc... ~]# service maxscale start
Redirecting to /bin/systemctl start maxscale.service
[root@mdbc... ~]# mysql -h mariasmghc.centralus.cloudapp.azure.com -P 4006 -u myapp -pMariadb123 mydb
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 49070
Server version: 10.1.19-MariaDB Enterprise Cluster MariaDB Enterprise Certified
Binary

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [mydb]>

```

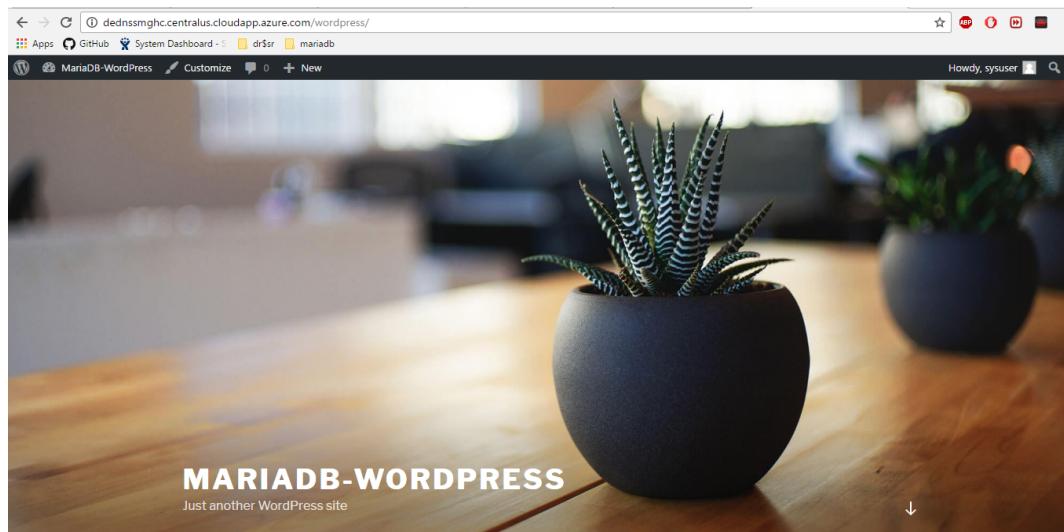


```

root@mdbec-max2:~ login as: mdbe
Authenticating with public key "rsa-key-20170110"
Last login: Mon Jan 30 06:44:56 2017 from 183.82.117.202
[mdbe@mdbec-max2 ~]$ sudo -i
[root@mdbec-max2 ~]# ps -eaf | grep maxscale
maxscale 44556      1  0 06:48 ?          00:00:00 /usr/bin/maxscale --user=maxscale
root      46012 45902  0 06:54 pts/0    00:00:00 grep --color=auto maxscale
[root@mdbec-max2 ~]# service maxscale stop
Redirecting to /bin/systemctl stop maxscale.service
[root@mdbec-max2 ~]# netstat -ntalp | grep 4006
[root@mdbec-max2 ~]# netstat -ntalp | grep 4007
[root@mdbec-max2 ~]# netstat -ntalp | grep 4008
[root@mdbec-max2 ~]# mysql -h mariasmghc.centralus.cloudapp.azure.com -P 4006 -u myapp -pMariadb123 mydb
ERROR 2003 (HY000): Can't connect to MySQL server on 'mariasmghc.centralus.cloudapp.azure.com' (110 "Connection timed out")
[root@mdbec-max2 ~]#

```

5. Alternatively, we can also check the MariaDB Database connectivity by using the WordPress URL provided in Access Information and run it in the Browser.



6. Finally, we observe the High Availability feature of MariaDB Enterprise Cluster using this Test Drive.