MariaDB

Get Started with MariaDB TX 2.0 in Azure

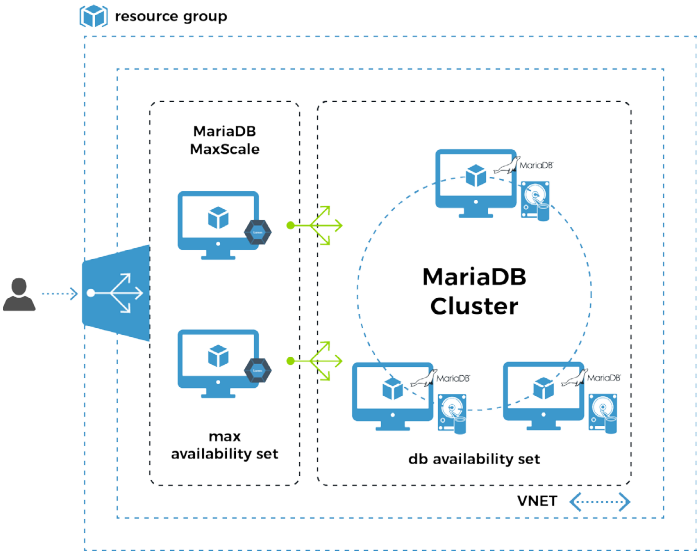
WHITE PAPER

A Step-By-Step Guide for Easy Deployment

Table of Contents

INTRODUCTION

MariaDB – the fastest growing open source database in the world – is a modern, relational database with extensible architecture and enterprise-grade reliability. It has more than 12 million users globally and 500 customers in more than 45 countries.



Want a quick and easy way to experience MariaDB for yourself?

MariaDB TX in Microsoft Azure offers an easy-to-deploy cluster with three data nodes in the back, along with two MariaDB MaxScale nodes to offer redundancy.

Here's a brief overview of the MariaDB TX solution in Azure components:

MariaDB Server is a secure relational database that combines familiar SQL interfaces with open extensibility to support a breadth of use cases. It supports traditional operational use cases as well as new use cases featuring a complete set of JSON functions and an optimized transactional storage engine (MyRocks).

MariaDB Cluster extends MariaDB's widely adopted open source database with Galera clustering technology featuring multiple active masters. It

ABOUT THIS GUIDE

This guide provides the step-by-step instructions you need to set up and deploy MariaDB TX in Microsoft Azure.

For developers and DBAs just getting started with MariaDB, or deploying MariaDB Cluster for the first time, MariaDB TX in Microsoft Azure is a great place to dip your toes in the water. Get familiar with the topology of the offering, see how the components work together and work through some scenarios.

NOTE

MariaDB's Cluster offering in Azure does not demonstrate the full range of capabilities of MariaDB TX and is not intended for production deployment. If you are planning to run MariaDB in a production environment, please contact MariaDB for consultative services and support.

achieves high scalability, scaling out on demand using automatic membership control of new nodes joining the cluster and failed nodes dropping from the cluster, and scaling up with multi-core processors.

MariaDB MaxScale is a database proxy that decouples client applications from the complexities of a database cluster. In this architecture, the two MaxScale nodes are A-series VMs.* They are set up behind an Azure load balancer, configured to automatically failover if one instance of MaxScale becomes unavailable. In addition to load balancing and failover, MariaDB MaxScale provides caching, streaming and sharding mechanisms.

The combined technology of this solution lets you read and write to any cluster nodes without slave lag. Indeed, MariaDB MaxScale is automatically configured to provide the following services across the three back-end nodes:

- **Read/Write Split:** Assesses the content of the query and determines whether to send it to the master node or to one of the slaves
- **Write Connection Router:** If you have an app that already splits traffic, all traffic sent to this port will be sent to the master node
- **Read Connection Router:** Again, if you have an app that already splits traffic, all traffic sent to this port will be dynamically balanced to the slave nodes

NOTE

* A-series VMs have CPU performance and memory configurations best suited for entry-level workloads like development and test. They are economical and provide a low-cost option to get started with Azure.

DEPLOYING MARIADB TO THE AZURE CLOUD

To get started with your deployment, go to the MariaDB TX page in the Azure Marketplace. From there, you'll be prompted to enter information on a sequence of 5 screens. This guide will walk you through each step of the setup and deployment process.

Please take a few minutes to review all sections of this guide *before* you begin your deployment. Doing so will let you know what's in store – and ensure your deployment experience is as quick and easy as possible.

NOTE

The Azure Free Trial does not have enough core quota to deploy this solution. However, MariaDB Test Drive lets you deploy the solution for two hours at no charge.

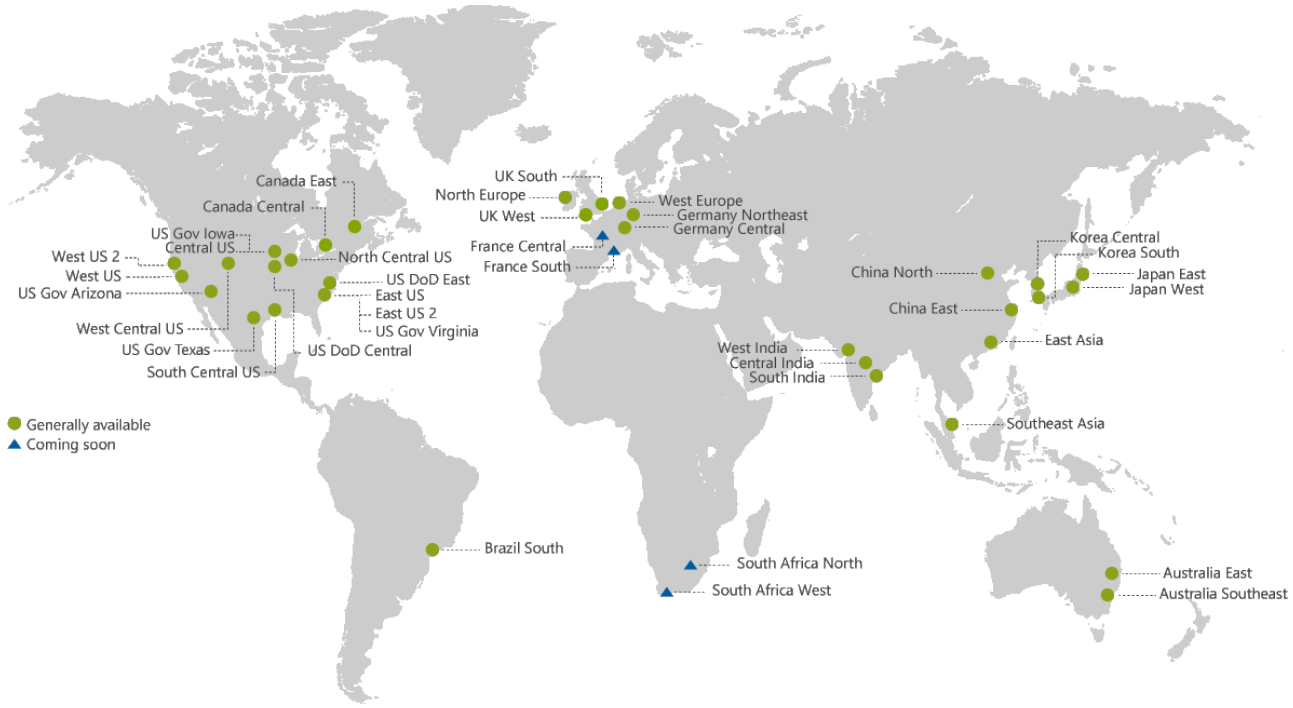
Microsoft Azure Subscription

You must have an Azure Subscription to be able to deploy this solution. If you use minimum VM sizes, each with 2 cores, your deployment should use 10 cores, which you should be able to deploy under the default core quota for a paid Azure subscription. In order to deploy more than 10 cores in a single region, you may need to increase your quota. See Azure quota limitations for more information.

Payment for VMs, storage and other Azure resources deployed as part of this solution are the responsibility of the end user. Pricing will depend on the options chosen during deployment. Your subscription with Microsoft defines the terms of payment.

STEP-BY-STEP INSTRUCTIONS

SCREEN 1: Basics



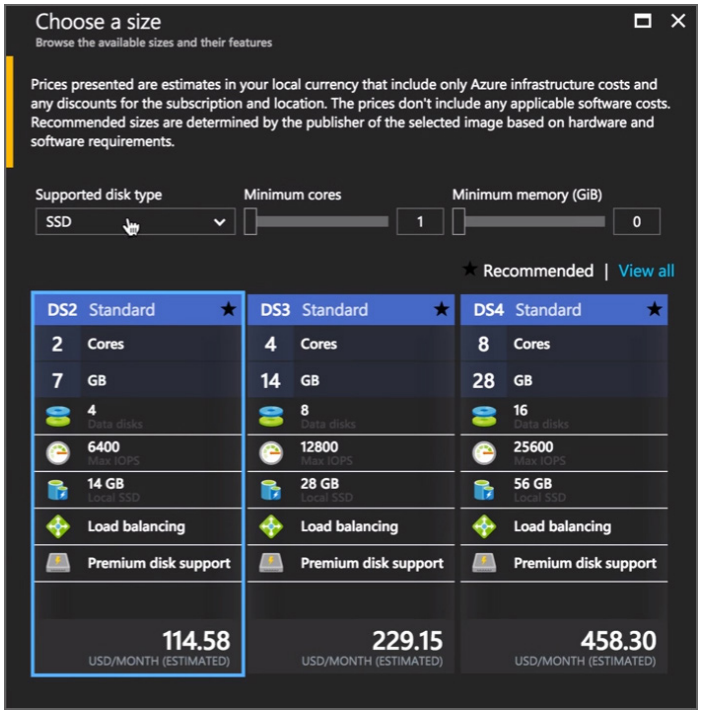
- **Cluster Name:** The name you choose should be globally unique, as it will be used to set up DNS and storage accounts in Azure.
- **Subscription:** From the dropdown list, select the subscription under which you wish to deploy the cluster.
- **Resource Group:** In creating the resource group for your cluster, it makes most sense to give it the same value as your Cluster Name.
- **Location:** From the dropdown list, choose a location where the cluster resources will be deployed. Be sure the location you choose supports the types of VMs and storage you want to use.

    › The "DS" series of VMs include support for "Premium" solid-state (SSD) storage, which offers greatly improved performance over the "Standard" spinning-disk storage supported by the "A" and "D" series VMs.

    › The "DS" series of VMs are available in only a limited subset of locations/regions. If you want to be able to use "DS" series VMs and high-performance premium storage, you must create your Resource Group in a location/region that supports Premium storage.

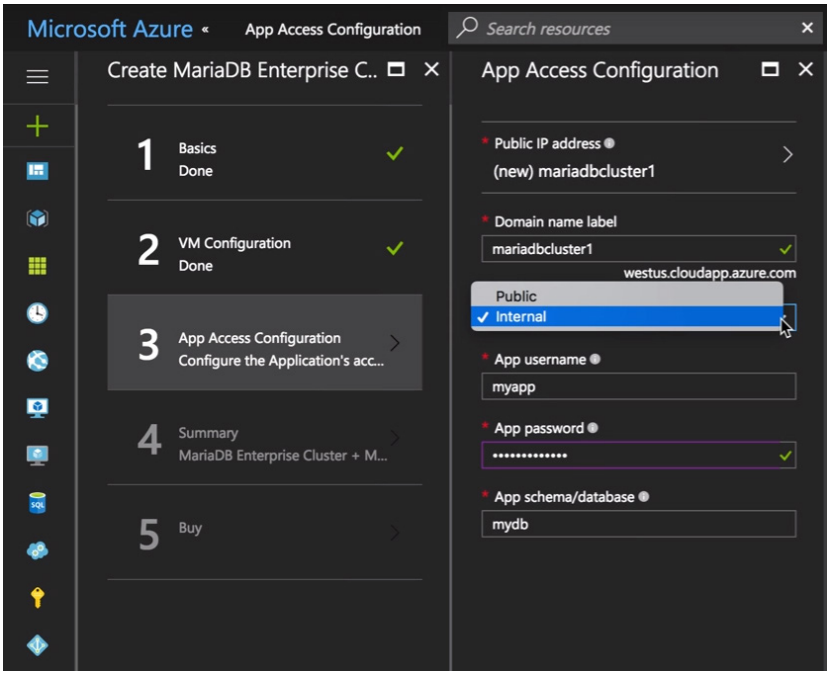
    › *For detailed information about Azure products by region and/or Premium storage, see corresponding links in the Resources section of this guide.*

SCREEN 2: VM Configuration



- **Data node VM size:** From the dropdown list, choose the appropriate size for your budget and needs.
  - › Azure allows you to choose from a variety of different VM types and sizes.
  - › If you want to use SSD (Premium) storage, you must choose a DS-series VM. DS-series VMs are only available in a limited number of regions in Azure.
  - › *For more information about Azure VM types and sizes and/or product regions, see the Resources section of this guide.*
- **Data node storage account name:** This is set to your Cluster Name by default. You can keep that default value if the Azure UI allows you to do so. Otherwise, choose a different name. The name you choose should be globally unique.
- **Data node storage account type:** From the dropdown list, choose the account type you want to use, either Premium-LRS (SSD) or Standard-LRS (HDD). (For more information on Storage Pricing in Azure or Premium Storage, see the appropriate link in the Resources section of this guide).
- **Disk size:** From the dropdown list, choose the size that suits your needs. Note that each data node in the cluster will place its data directory on this disk. When using Premium (SSD) storage, choosing a larger disk will have an effect on the available IOPS (i.e., you get better performance if you're using a bigger device, even if you're not using the whole thing).
- **MaxScale VM size:** Use the default value unless you identify a CPU or memory bottleneck on the MaxScale node.
- **SSH login name:** The name you create will be used to administer the nodes of the cluster using SSH.
- **Public SSH key:** In order to facilitate administration of the cluster, you must provide an SSH public key. The file that holds this key is often called id_rsa.pub and will typically be located in the .ssh subdirectory of your home directory on a Linux or other Unix computer. The private key corresponding to this public key must be used to connect to the nodes in the cluster using SSH. You are responsible for generating and maintaining your own key pair.

SCREEN 3: App Access Configuration

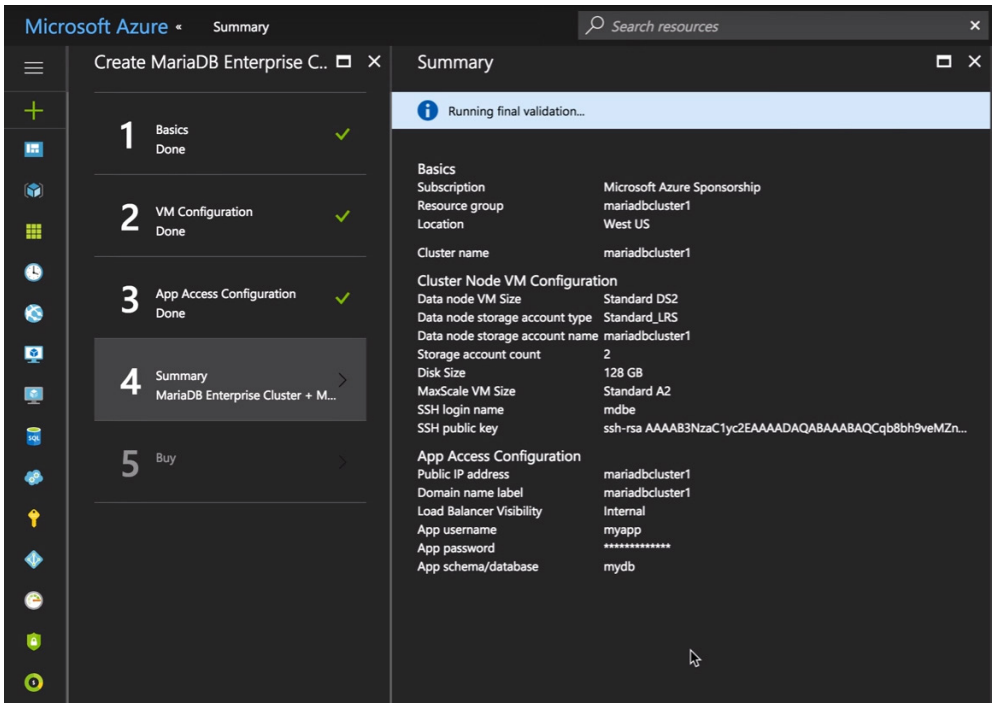


- **Public IP address:** The address configured will be used to connect to your MaxScale node. By default, Azure should do this for you automatically.
- **Domain name label:** This will form the DNS name for your MaxScale node. The UI will show the suffix below the text box.
- **Load balance visibility:** From the dropdown list, select "Public" or "Internal" to determine whether you want to connect to the exposed MariaDB services on the MaxScale nodes from outside Azure (Public) or only from inside Azure (Internal).
  - › Choose "Internal" if you intend to create additional VMs to run your application inside Azure.
  - › Choose "Public" only if you understand the implications of sending unencrypted database traffic across the open internet – this is not a recommended configuration and should only be used for testing or POC purposes.
- **App username:** The name you specify will be used by database clients to connect to the MaxScale service.
- **Password:** The name you specify will be used by database clients to connect to the MaxScale service. Use a high-quality password.
- **App schema/database:** This database will be created for you, and the App username you specify will have full access (ALL PRIVILEGES) to this database.

NOTE

This deployment topology does not support SSL connections from clients to MaxScale at this time (though you can theoretically set this up yourself after deployment is complete). For better security, you should set up a VPN connection to Azure, you should use an SSH tunnel to interact with the MaxScale node, or you should connect to the MaxScale node from another VM in Azure that you place into the same virtual network as your MariaDB Cluster deployment. The implementation details of those approaches are beyond the scope of this guide.

SCREEN 4: Summary



Here, you'll get the chance to review the information you've submitted on the previous screens. If you need to go back and make any changes, you can do so now.

SCREEN 5: Buy

Read and agree to the text – and deploy the solution!

NOTE

When you hit the "Purchase" button, you are not buying a MariaDB subscription. You are simply purchasing the virtual resources from Microsoft Azure.

MONITOR YOUR DEPLOYMENT

It usually takes about 20 minutes for the deployment to complete, but be patient; it can sometimes take much longer. You can monitor its progress using the Azure Portal. Here's how:

1. Load the Resource Groups blade (access it here: https://portal.azure.com/#blade/HubsExtension/BrowseResourceBlade/resourceType/Microsoft.Resources%2Fsubscriptions%2FresourceGroups).
2. Click the Resource Group that you chose in the Basics step of your deployment.
3. You should see a "Last deployment" field with today's date and "Deploying" in parentheses. Click that.
4. You'll get a "Deployment history" blade that shows your current deployment. Click the deployment.
5. Review all the parameters set for this deployment.
6. Scroll to the bottom and you will see the list of resources being deployed.

TROUBLESHOOTING

If any resources have failed to deploy, follow the Troubleshooting instructions in the MariaDB Knowledge Base.

CONNECTING TO MARIADB TX

After your deployment is complete, you'll be able to log in to the MaxScale node using <*clusterName*>.<*location*>.cloudapp.azure.com. Your *clusterName* is the value you gave in the very first field when setting up your deployment, and *location* is the name of the location to which you deployed the solution, with spaces removed (e.g., West US becomes "westus" and East US 2 becomes "eastus2").

> EXAMPLE: If your *clusterName* was "mytestcluster" and you deployed to the "West US" location, the fully qualified domain name for your MaxScale node would be mytestcluster.westus.cloudapp.azure.com

Direct MariaDB/MySQL client access:

- If you chose "Internal" Load Balancer Visibility, you will only be able to connect applications to MaxScale from a VM running in the same virtual network in Azure. Deploying and configuring those VMs is the responsibility of the customer.
- If you chose "Public" Load Balancer Visibility during deployment, you'll be able to connect directly to your MaxScale node using any MariaDB or MySQL client, like this:

```
mysql -h mytestcluster.westus.cloudapp.azure.com -P 4006 -u myapp -pmyapppassword mydb
```

```
$ mysql -h mytestcluster.westus.cloudapp.azure.com -u myapp -pmyapppassword -P 4006 mydb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2387
Server version: 10.0.20-MariaDB Enterprise Cluster MariaDB Enterprise Certified Binary, wsrep_2

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

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

mysql 10.0.20-MariaDB Enterprise Cluster (myapp) [mydb]> select current_user();
+----------------+
| current_user() |
+----------------+
| myapp@%        |
+----------------+
1 row in set (0.09 sec)
```

ADVANCED USAGE

Detailed information about connecting to, using and administering MariaDB TX in Azure is available on the Usage and Administration page in the MariaDB Knowledge Base.

ADDITIONAL MARIADB SOLUTIONS IN AZURE (including a FREE Test Drive!)

TEST DRIVE: This solution gives you 2 hours of FREE access to MariaDB TX in Azure. "Kick the tires," take a look around and get a feel for its capabilities. With a cluster always running, the test drive can deploy in as little as 30 seconds. What's more, it includes an extra VM (WordPress) so you can see what it looks like to connect an application to the cluster and understand how the solution works on the back-end.

STANDALONE: Single MariaDB Server instance. VM deploys in about 5 minutes.

Do More with Your Deployment

Once you've deployed MariaDB TX in Azure, we encourage you to review use cases and create some scenarios to work through.

Want to see how MaxScale failover works? There are two practice scenarios included on the Test Drive user guide.

RESOURCES

- **Azure Marketplace – MariaDB TX**
  https://azure.microsoft.com/en-us/marketplace/partners/mariadb/cluster-maxscale/
- **Azure Quota Limitations**
  https://mariadb.com/kb/en/mariadb-enterprise/mariadb-enterprise-cluster-in-azure-troubleshooting/
- **MariaDB Test Drive**
  https://mariadb.com/kb/en/mariadb-enterprise/mariadb-enterprise-cluster-in-azure-test-drive/
- **Premium Storage in Azure**
  https://azure.microsoft.com/en-us/documentation/articles/storage-premium-storage-preview-portal/
- **Products by Region in Azure**
  https://azure.microsoft.com/en-us/regions/services/
- **Storage Pricing in Azure**
  https://azure.microsoft.com/en-us/pricing/details/storage/
- **Troubleshooting**
  https://mariadb.com/kb/en/mariadb-enterprise/mariadb-enterprise-cluster-in-azure-troubleshooting/
- **Usage and Administration – MariaDB Cluster in Azure**
  https://mariadb.com/kb/en/mariadb-enterprise-cluster-in-azure-usage-and-administration/
- **VM Pricing in Azure**
  https://azure.microsoft.com/en-us/pricing/details/virtual-machines/

Planning a production deployment? We're here to help!

If you're planning to deploy MariaDB in a production environment, a MariaDB TX Subscription is recommended. MariaDB offers a combination of curated packages and comprehensive support, including break/fix, security alerts, updates, consultative services and rapid response – everything you need to confidently deploy MariaDB as part of your core infrastructure and in the cloud.

At MariaDB, our consultants will help you find the right support for your database deployment. Contact us today.

Americas: sales-AMER@mariadb.com
Europe, Middle East, Africa: sales-EMEA@mariadb.com
Asia Pacific: sales-APAC@mariadb.com