✓ 100 XP

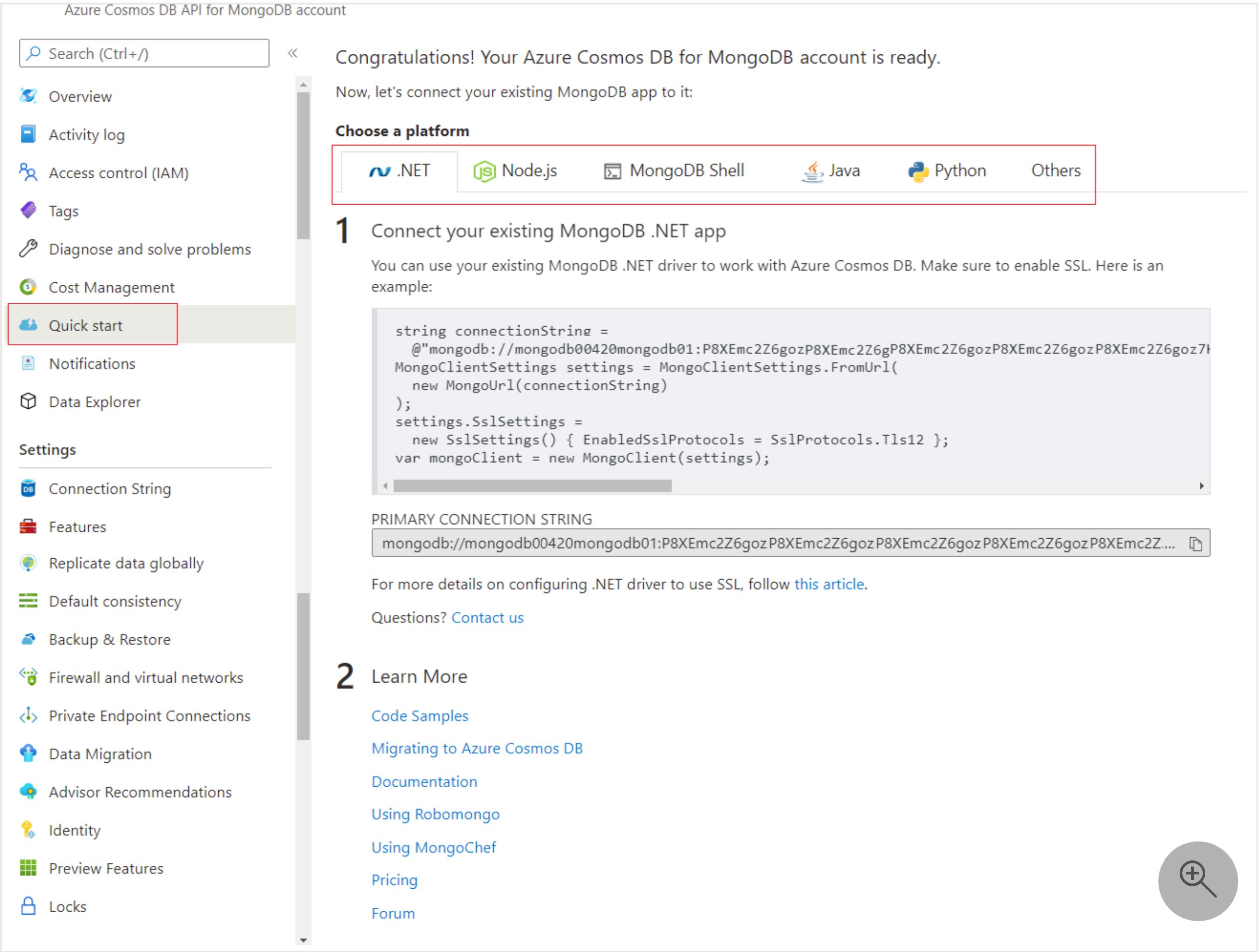
Connect and query with an Azure Cosmos DB for MongoDB

2 minutes

Once you've created an Azure Cosmos DB for MongoDB account, connecting to that account will be easy. Regardless if you're connecting from .NET, Node.js, MongoDB Shell, Java, Python or other sources, you can find easy to follow connection instructions or scripts on the Azure portal itself. This reduces the guesswork of how to create your own connection strings or use the MongoDB Shell to connect to your account. Let's take a quick look on where we can find this information.

1. In the Azure portal, navigate to your Azure Cosmos DB for MongoDB account.

2. On the left-hand Azure Cosmos DB for MongoDB account menu, select **Quick start**.



Congratulations! Your Azure Cosmos DB for MongoDB account is ready.

Now, let's connect your existing MongoDB app to it:

1 Connect your existing MongoDB .NET app

You can use your existing MongoDB .NET driver to work with Azure Cosmos DB. Make sure to enable SSL. Here is an example:

```
string connectionString =
  @"mongodb://mongodb00420mongodb01:P8XEmc2Z6gozP8XEmc2Z6gP8XEmc2Z6gozP8XEmc2Z6gozP8XEmc2Z6goz7k
MongoClientSettings settings = MongoClientSettings.FromUrl(
  new MongoUrl(connectionString)
);
settings.SslSettings =
  new SslSettings() { EnabledSslProtocols = SslProtocols.Tls12 };
var mongoClient = new MongoClient(settings);
```

PRIMARY CONNECTION STRING

mongodb://mongodb00420mongodb01:P8XEmc2Z6gozP8XEmc2Z6gozP8XEmc2Z6gozP8XEmc2Z6gozP8XEmc2Z....

For more details on configuring .NET driver to use SSL, follow this article.

Questions? Contact us

2 Learn More

Code Samples
Migrating to Azure Cosmos DB
Documentation
Using Robomongo
Using MongoChef
Pricing
Forum

There are tabs for different programming languages. Each of these tabs will already have prepared the code snippet you'll need to connect to the Azure Cosmos DB for MongoDB account. All you need to do is copy and paste the code with no changes into your application code or tool. If your programming language or tool isn't listed, the last tab list most common parameters like host, port, username and password so you can create your own connection string on that language or tool.

Let's review and example of the connection string format we would use to connect to an Azure Cosmos DB for MongoDB account.

```JavaScript
    "mongodb://mongodb00420mongodb01:P8XEmc2Z6gozP8XEmc2Z6gozP8XEmc2Z6goz-
P8XEmc2Z6gozP8XEmc2Z6gozP8XEmc2Z6gozP8XEmc2Z6goz==@mongodb00420mongodb01.mo
ngo.cosmos.azure.com:10255/?
ssl=true&retrywrites=false&replicaSet=globaldb&maxIdleTimeMS=120000&appName
=@mongodb00420mongodb01@";
```

This code snippet tells us that the Azure Cosmos DB for MongoDB account name is *mongodb00420mongodb01* and its password is *P8XEmc2Z6gozP8XEmc2Z6gozP8XEmc2Z6gozP8XEmc2Z6gozP8XEmc2Z6gozP8XEmc2Z6gozP8XEmc2Z6goz==*. The port used will be 10255, which is the standard Azure Cosmos DB for MongoDB account port. Finally note that our host address will be *mongodb00420mongodb01.mongo.cosmos.azure.com*.

While it's that simple to connect to an Azure Cosmos DB for MongoDB account, we should still take care of choosing the right partitioning and indexing strategy when designing our Azure Cosmos DB for MongoDB databases. It should be more than just connecting and loading/reading data. We'll discuss these topics in more detail in the *Partitioning and Indexing* Module.

Next unit: Capacity estimation using the Azure Cosmos DB Capacity Calculator

Continue ›