COSC 416: Topics in Databases (DBaaS)

TOPIC 9: MICROSOFT AZURE SQL

SCHEDULE

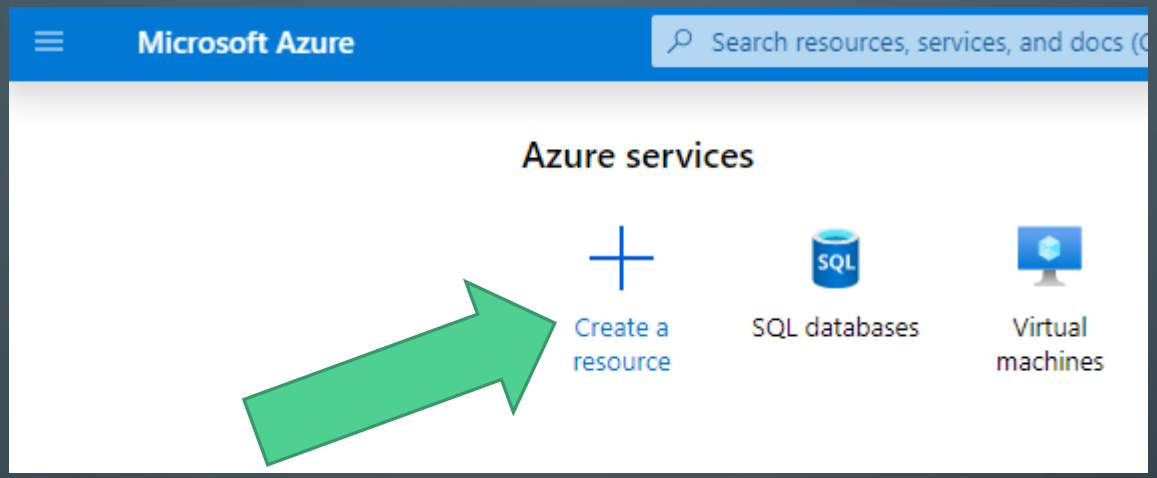
1. *Quiz #1 – Recap*
2. *Azure Instance Creation*
3. *The Azure CLI – installation and basic use*

CREATING AZURE INSTANCES

PREVIOUSLY ON COSC 416

- In our last lecture (before we had the quiz) we did a brief introduction to Microsoft Azure's DBaaS options
- Today, we'll take a somewhat more in-depth dive into using Azure, specifically creating our database instances and managing them through the Azure interface

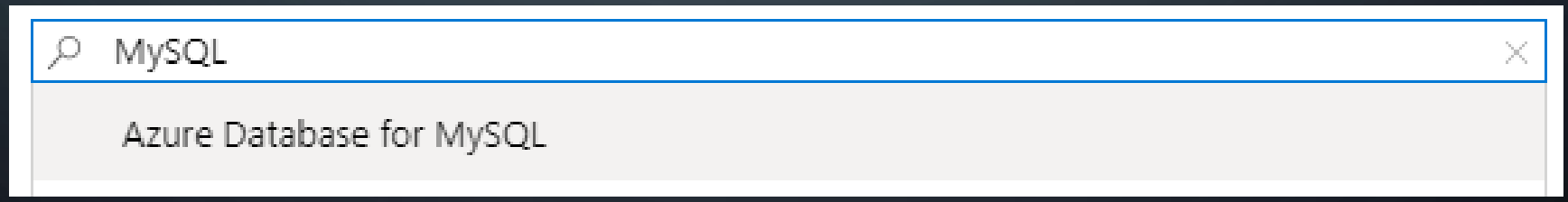
NAVIGATING THE AZURE WEB INTERFACE



- Microsoft Azure has it's own web dashboard, similar to that in AWS
- When you first log in to Azure, you may need to click the "Portal" button in the upper-right hand side to get to Azure dashboard
- Start by clicking "Create a resource"

CREATING A NEW MYSQL INSTANCE

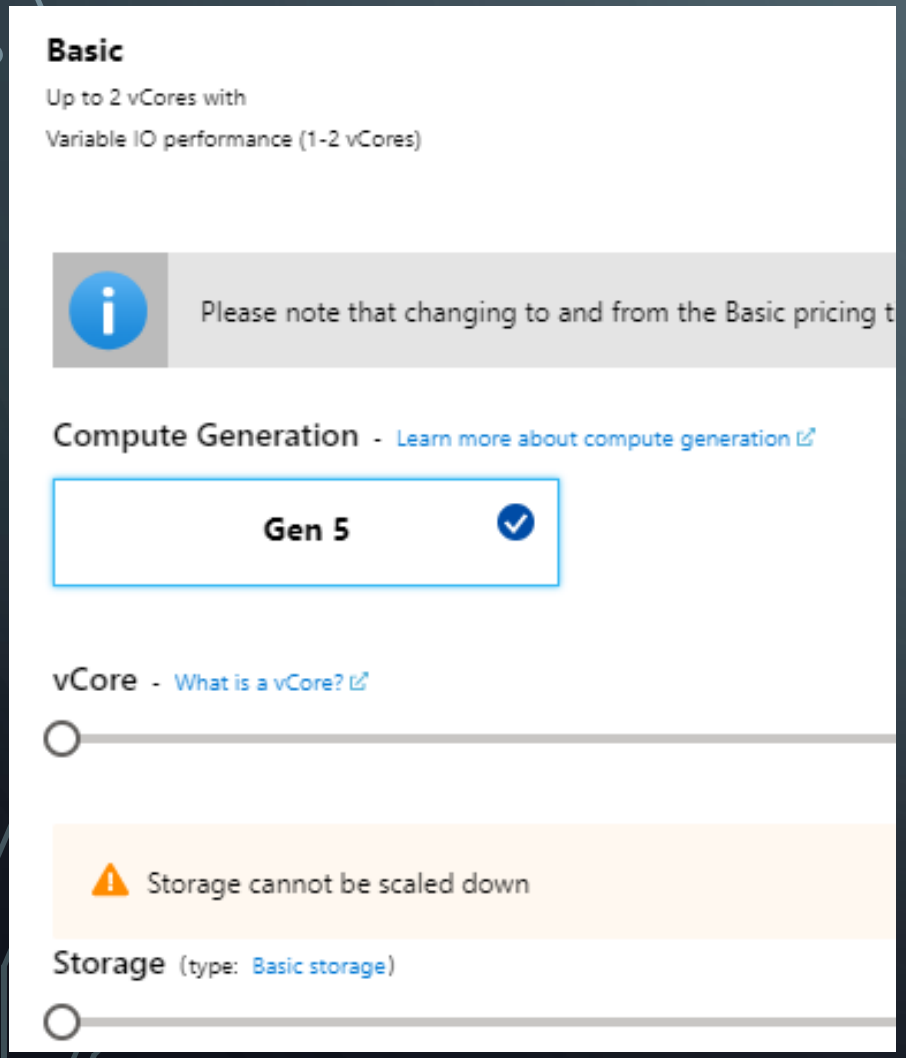
- Azure offers many kinds of MySQL instances, but we need to make sure we pick the correct service
- In the “New” page search bar, you’ll want to enter “Azure Database for MySQL”



CREATION INFORMATION

- Clicking the “Create” button on the Azure Database for MySQL page will take you to a creation form
- Standard data required: Create a resource group (similar to a VPC in AWS), set an instance name, admin username and password
- A reminder to be careful with pricing and machine size!

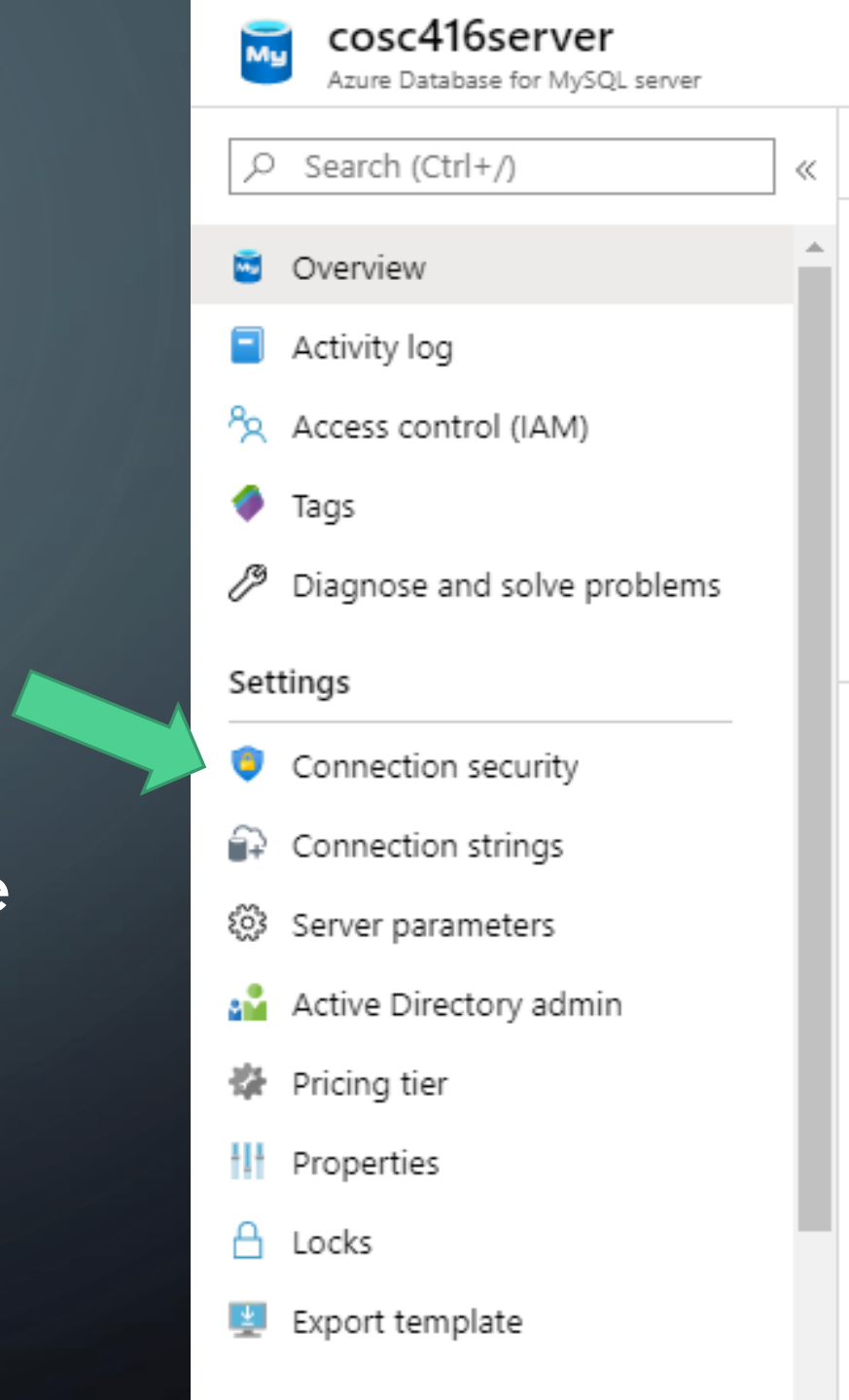
MACHINE SIZE SETTINGS



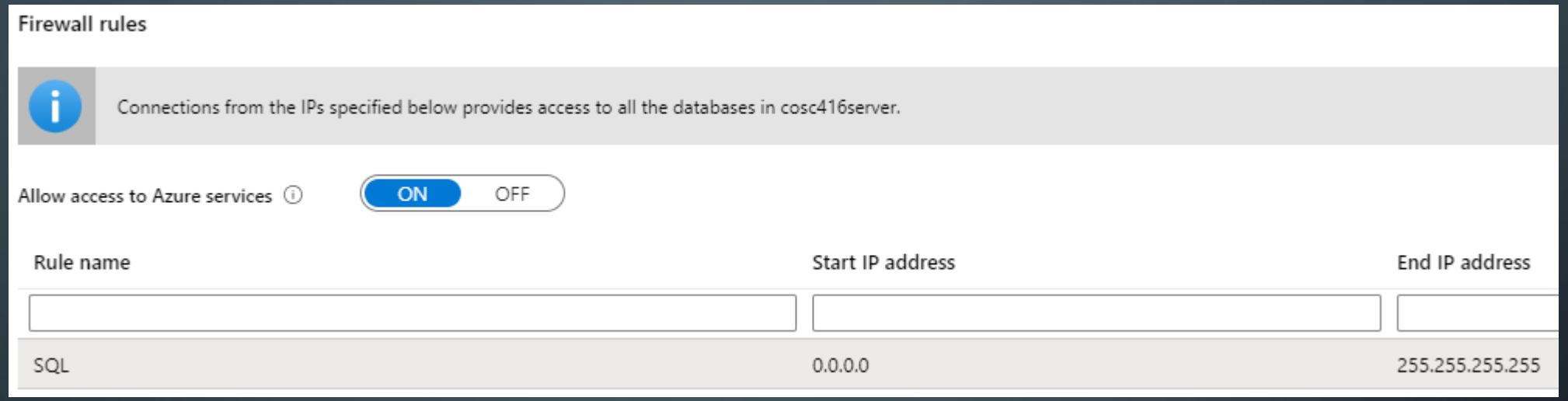
- When configuring your machine size, make sure you do the following:
  - Switch to the “Basic” tab
  - Move the vCore and Storage sliders to their minimum values (1 vCore, 5GB storage)
  - Disable Auto-growth

MANAGING THE INSTANCE

- Once your instance is online, it'll appear under your list of instances and you'll be able to go to it's dashboard
- Once you're there, check the menu on the left and click on "Connection Security"



MANAGING CONNECTIONS

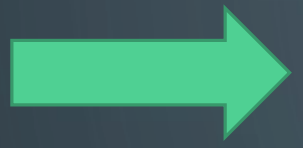


- We'll need to create a new firewall rule allowing traffic for our IP address
- You can use a specific range, or 0.0.0.0 to 255.255.255.255 to cover all IP addresses
- Make sure you save your changes!

AZURE SSL

- By default, Azure will require that all connections to the database instance be done via SSL
- This allows encryption-in-transit of both database commands and query results
- HeidiSQL and similar clients can be set up to use SSL in their connections, but for now we'll just disable it – the option is right below the firewall rules

INSTANCE ENDPOINT URL



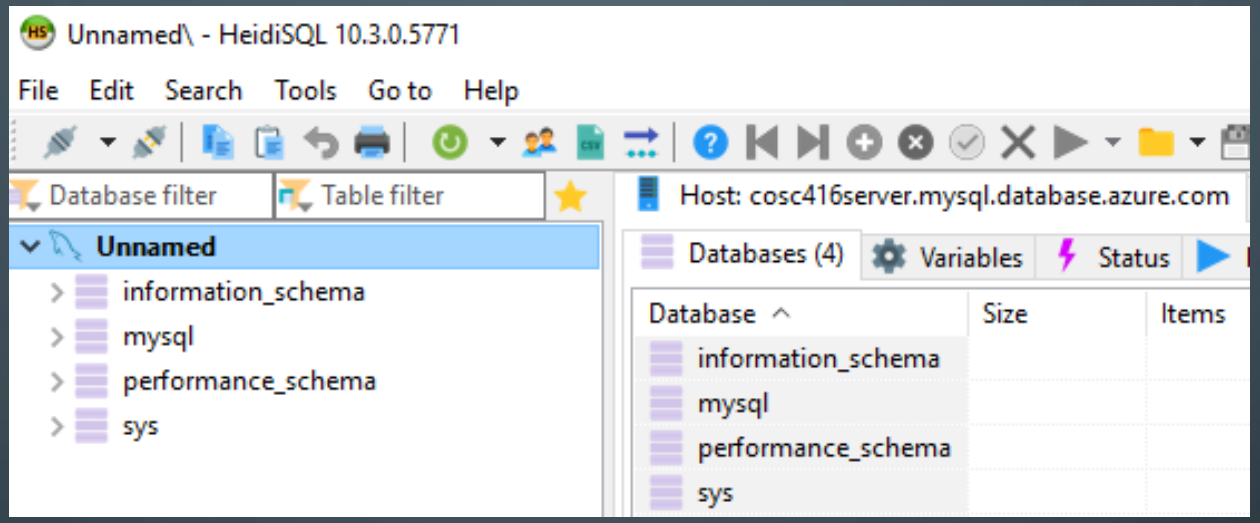
Server name	:	cosc416server.mysql.database.azure.com
Server admin login name	:	fritter@cosc416server
MySQL version	:	5.7
Performance configurati...	:	Basic, 1 vCore(s), 5 GB
SSL enforce status	:	DISABLED

- The endpoint URL we can connect to can be found on the resource overview page, under “Server name”
- Unlike in AWS, all instances have a public-facing URL, even if they don’t allow any traffic through those connections

CONNECTING TO YOUR INSTANCE

- Check the checklist:
  - Enabled access to the Azure application
  - Added firewall rule
  - Disabled SSL
- If you followed all of these steps, you should be able to connect to your Azure instance just as you would any other MySQL database

CONNECTING WITH HEIDISQL



- Simply enter your server name as the database address, and enter your username and password you set when creating the instance – you should be able to connect

CONNECTING TO YOUR AZURE INSTANCE WITH PYTHON

- Since your Azure instance is effectively just a MySQL database, the same as that used in AWS or a traditional database server, we can connect using the same code
- Just change the server name, username, and password in your Python code to point to the new server

A TRICKY NOTE ON USERNAME

- Simply using your admin username in Python as the user when connecting to your MySQL instance won't work
- You'll get a big long error message that doesn't provide much information on how to fix it
- The solution: use "user@instance name", for example I used:

fritter@cosc416server

AN EXAMPLE PYTHON CONNECTION

```
import mysql.connector
mydb = mysql.connector.connect(
  host="cosc416server.mysql.database.azure.com",
  user="fritter@cosc416server",
  passwd="**********"
)
mycursor = mydb.cursor()
mycursor.execute("CREATE DATABASE mynewdatabase")
mydb.commit()
```

THE AZURE CLI – INSTALLATION AND BASIC USE

GETTING STARTED WITH THE AZURE CLI

- Like AWS, Azure offers a command line interface (CLI) for interacting directly with the Azure ecosystem
- Head on over to the following link, and download a copy of Azure CLI suitable for your system:

https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest

LOGGING INTO THE AZURE CLI

- Once the CLI is installed (it will take a little time), we're ready to log in
- The CLI can be run in Windows under the standard Windows Command Prompt, or from Powershell
- Open up a command prompt and enter "az login"
- This should automatically open a browser window with a login page

OUTPUT OF AZ LOGIN

- Once the login is complete, you should see output like this:

```
C:\Users\generic>az login
You have logged in. Now let us find all the subscriptions to which you have access...
[
  {
    "cloudName": "AzureCloud",
    "id": "6c0fdd9a-f749-48ed-af2b-ecfd44f91b62",
    "isDefault": true,
    "name": "Free Trial",
    "state": "Enabled",
    "tenantId": "299ff174-53a3-44d4-b707-f12f93d85934",
    "user": {
      "name": "mfritter@telus.net",
      "type": "user"
    }
  }
]
```

CREATING A NEW INSTANCE WITH THE CLI

- We can create a new Azure Database for MySQL instance from the command line, replacing arguments as required:

```
az mysql server create --resource-group myresourcegroup --name mydemoserver --location westus --admin-user myadmin --admin-password <server_admin_password> --sku-name GP_Gen5_2 --version 5.7
```

- Note that you'll need a resource group before you can create your instance – those can be made from the CLI as well

A NOTE ON SKUS

- In Azure, the CLI uses the --sku-name field to determine the size and configuration of the instance that you want
- If we want to stick to the low-cost (free trial) database, we'll need to use a --sku-name of B_Gen5_1
- This stands for Basic, Gen 5, 1 vCore – the smallest instance that Azure offers (and also the cheapest)

DELETING INSTANCES FROM THE CLI

- As one would expect, we can also use the CLI to delete existing instances:

```
az mysql server delete --resource-group myresourcegroup --name mydemoserver
```

- Obviously, you should double-check that you've specified the correct instance and resource-group (!)

LEARN MORE ABOUT CLI COMMANDS

- Microsoft offers fairly good documentation of the CLI tool online, including what commands are available for the many different types of instances that Azure offers
- If you're in a hurry and want to know a command, you can also try:

az find <search term>

SCHEDULE

1. *Quiz #1 – Recap*
2. *Azure Instance Creation*
3. *The Azure CLI – installation and basic use*

SO LONG, FOLKS!