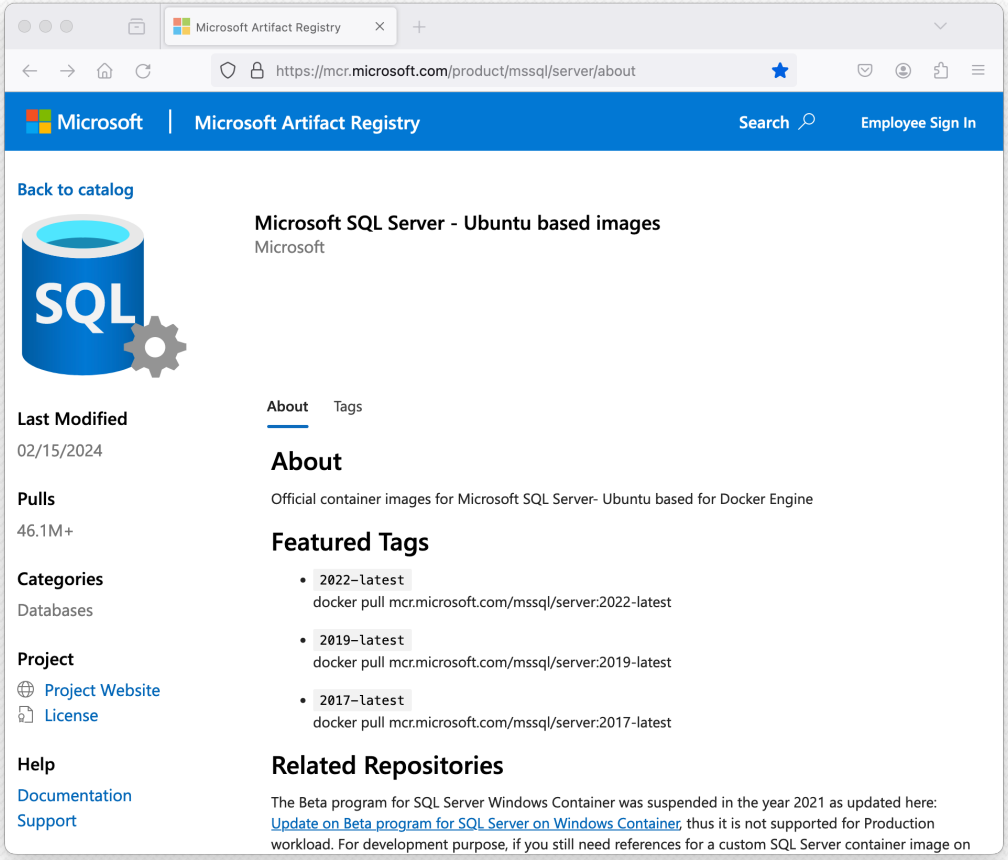
Computer Science

Ken Gamradt

Docker SQL Server

02-22-24

SQL Server – Microsoft Artifact Registry



Database Software

SQL Server

- SQL Server is a database management system (DBMS) that is used to store and manage data
- The Developer edition is a full-featured free edition

Azure Data Studio

- Azure Data Studio is a free, open-source graphical tool for managing and administering SQL Server and Azure databases
- It allows you to easily create and maintain databases, run SQL queries, and view and edit data

Required Software

docker

- Docker is a tool designed to make it easier to create, deploy, and run applications
- It does this by using containers, which are essentially self-contained environments that contain all of the necessary parts for an application to run, such as libraries, dependencies, and runtime
- Docker helps to simplify the process of building and deploying applications

Install docker

- First, make sure you have Docker installed on your computer
- If you don't have it, you can download it from the Docker website

Installation

Run the docker compose file

- A docker compose file is a way to define and run multi-container Docker applications
- It's like a blueprint for your application, telling Docker what images to use and how to connect them together
- This will save time and hassle because you can set everything up in one place and start everything with a single command
- It's written in yaml format and has sections for defining the version and the different services that make up the application
- Each service can be customized with options like images, environment variables, volumes, ports, and links to other services

docker-compose.yaml (.yml) file

```yaml
services:
  mssql:
    container_name: mssql-db
    hostname: mssql-db
    image: mcr.microsoft.com/mssql/server:2022-latest
    environment:
      ACCEPT_EULA: 'Y'                # required
      MSSQL_PID: 'Developer'          # required - default
      # MSSQL_SA_USER: 'sa'
      MSSQL_SA_PASSWORD: 'Pa55w0rd'   # required
      MSSQL_TCP_PORT: 1433            # default
      MSSQL_DATA_DIR: /var/opt/mssql/data
    ports:
      - "1433:1433"
    restart: always
    volumes:
      - ./data:/var/opt/mssql/data
      - ./log:/var/opt/mssql/log
      - ./secrets:/var/opt/mssql/secrets
```

docker-compose.yaml (.yml) file

Version

- Specifies the version of the docker compose file format

Services

- Defines the services (i.e., containers) that make up the application
- In this case, we have one service: mssql

Images

- Specifies the docker image to be used for the container
- For the mssql service, we are using the following image
  - mcr.microsoft.com/mssql/server:2022-latest

docker-compose.yaml (.yml) file

Environments

- Sets environment variables for the container
- In the postgres service, we are setting the following variables used to configure the SQL Server database
  - MSSQL_SA_USER
  - MSSQL_SA_PASSWORD
  - …

Volumes

- Mounts a volume (i.e., persistent storage) for the container
- In the mssql service, we are mounting volumes to store SQL Server data
  - data, log, secrets
- This allows the data to persist even if the container is stopped or removed

docker-compose.yaml (.yml) file

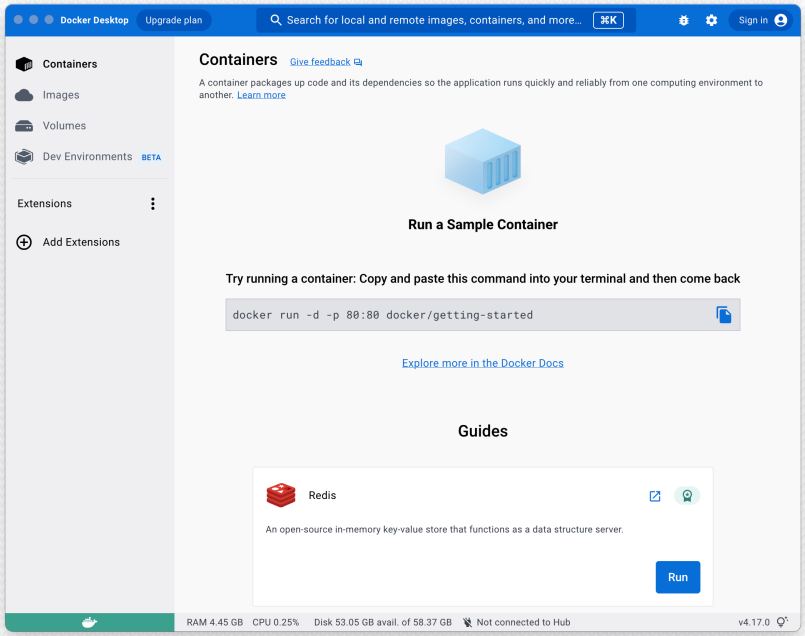
Ports

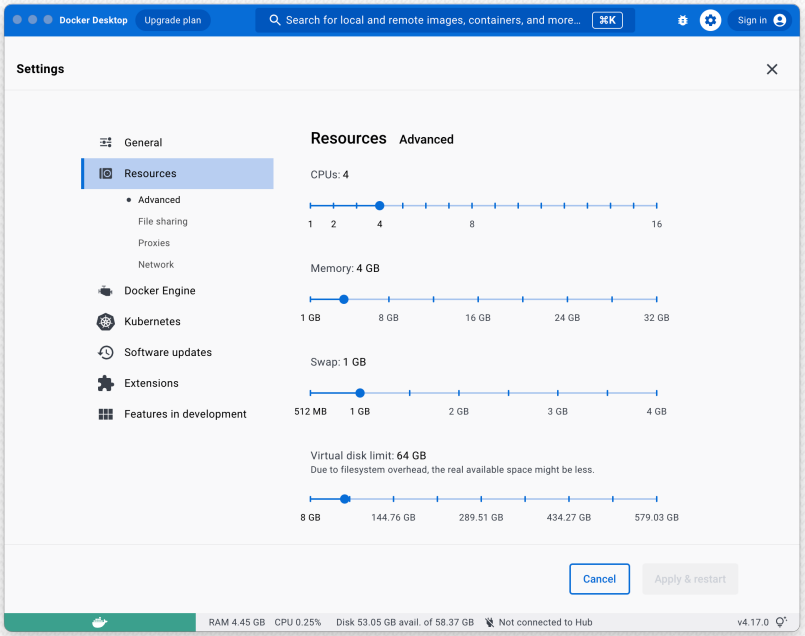
- Exposes a container's port to the host
- In the postgres service, we are exposing the default SQL Server port (1433) on the host

Depends on (not used here)

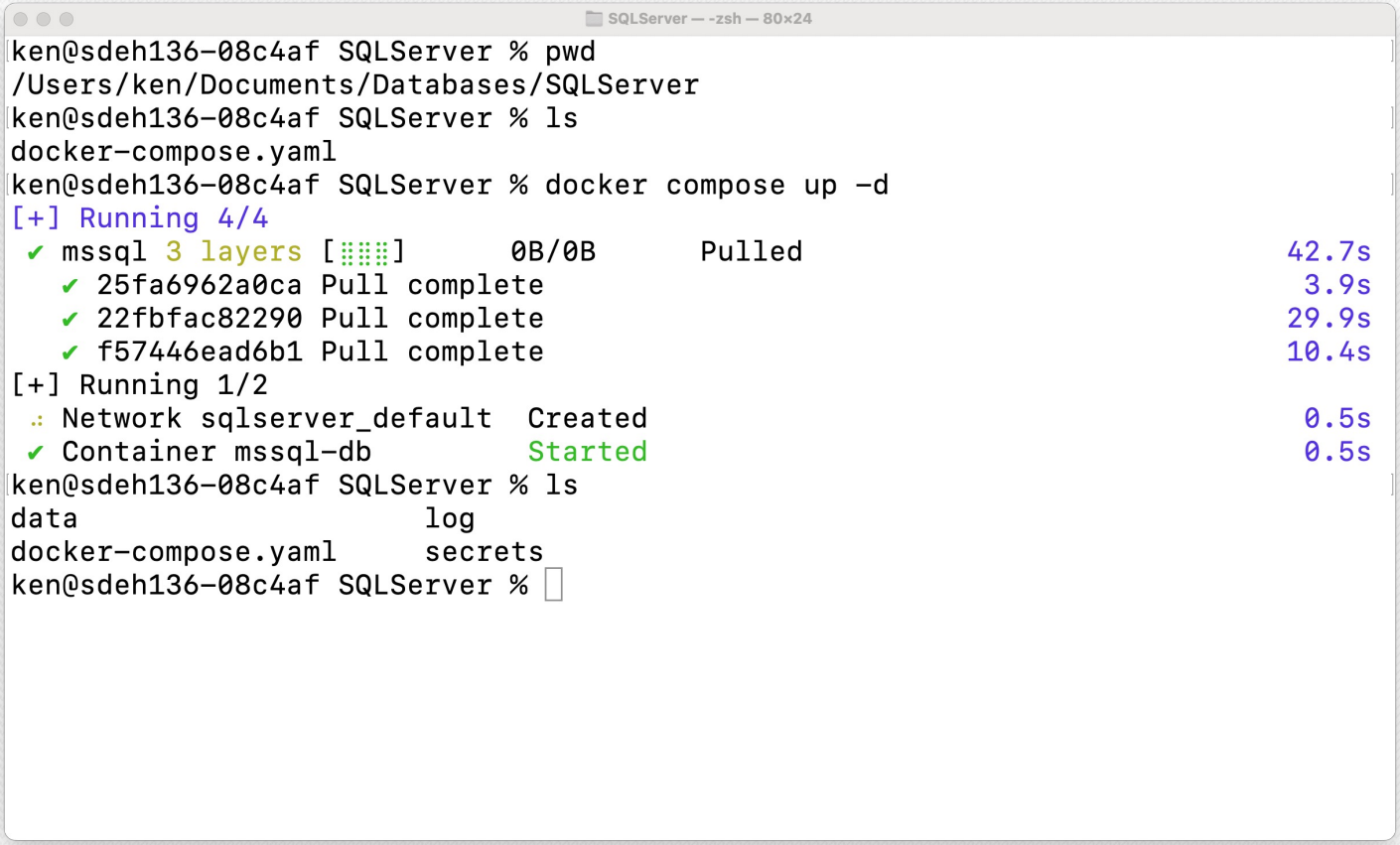
- Specifies that this service requires other services to already be running

Setting up Docker and SQL Server

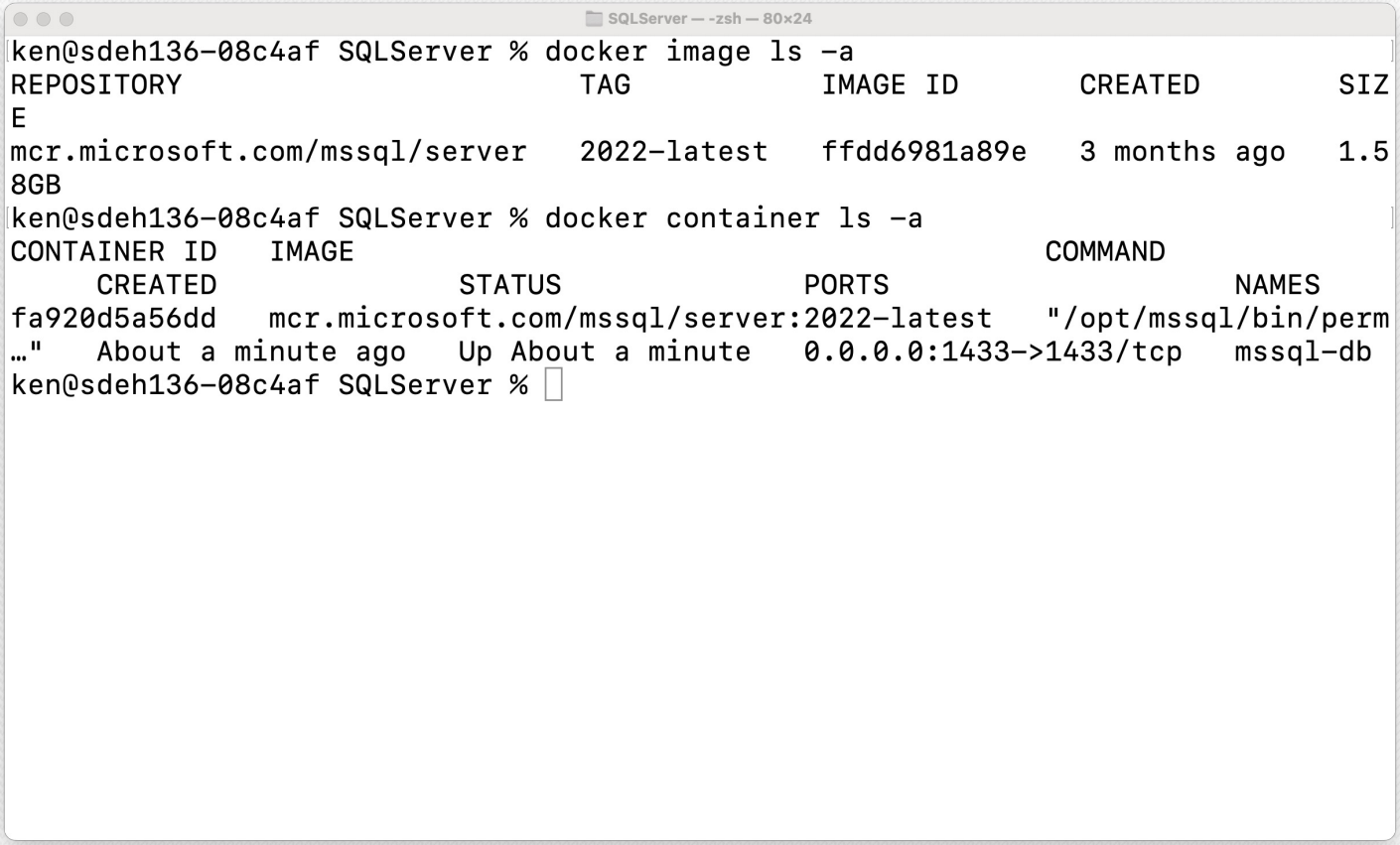




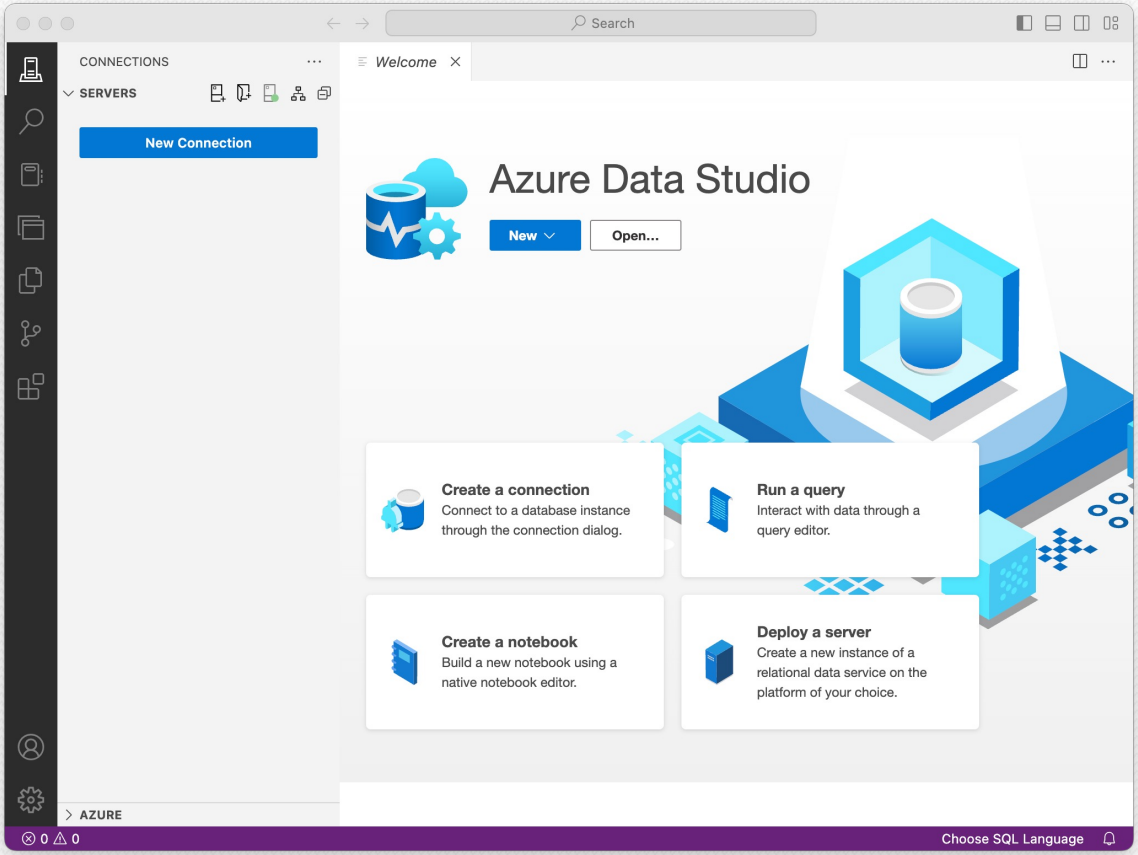
Setting up Docker and SQL Server



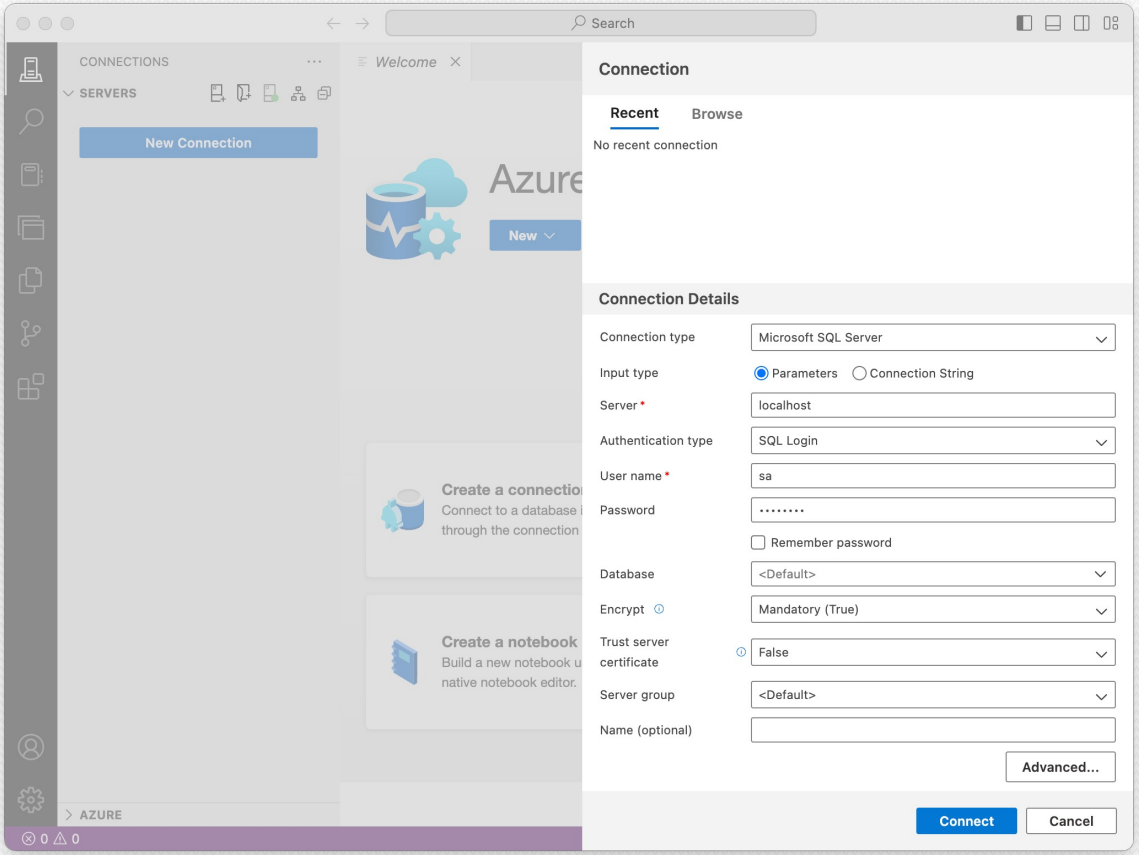
Setting up Docker and SQL Server



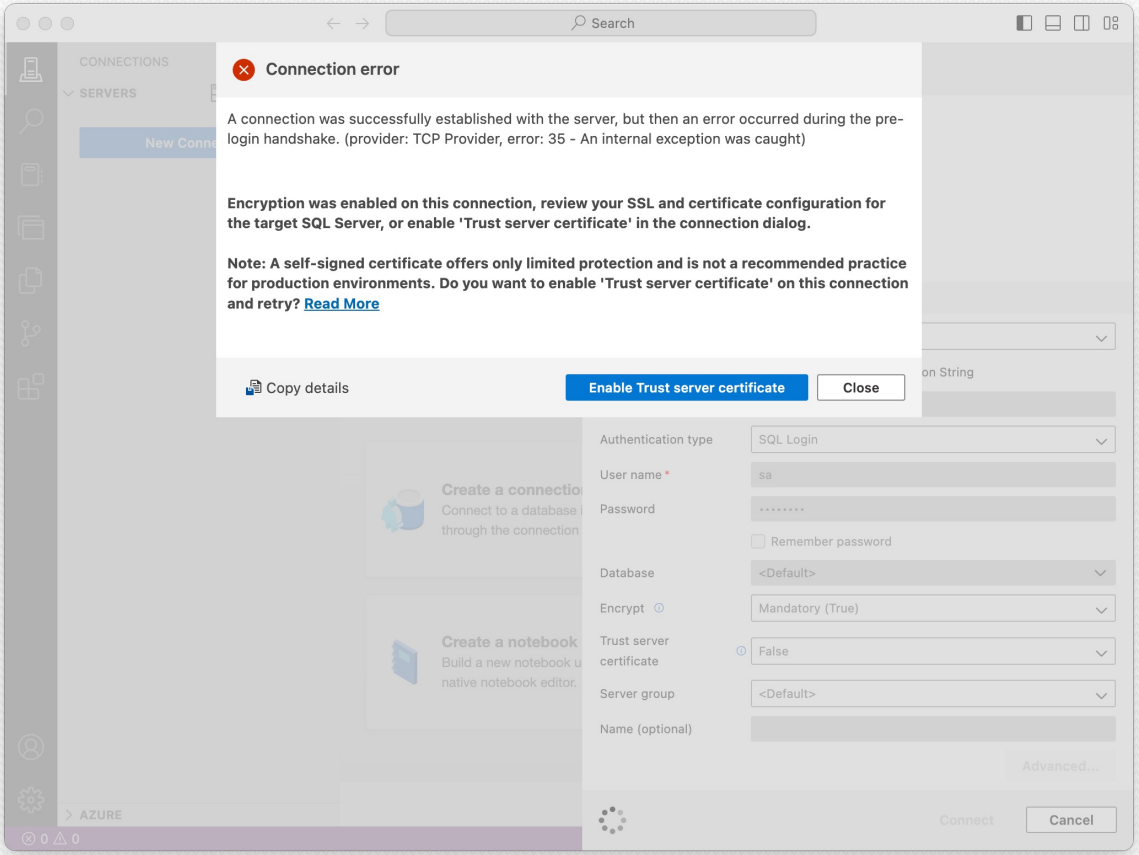
Setting up Azure Data Studio and SQL Server



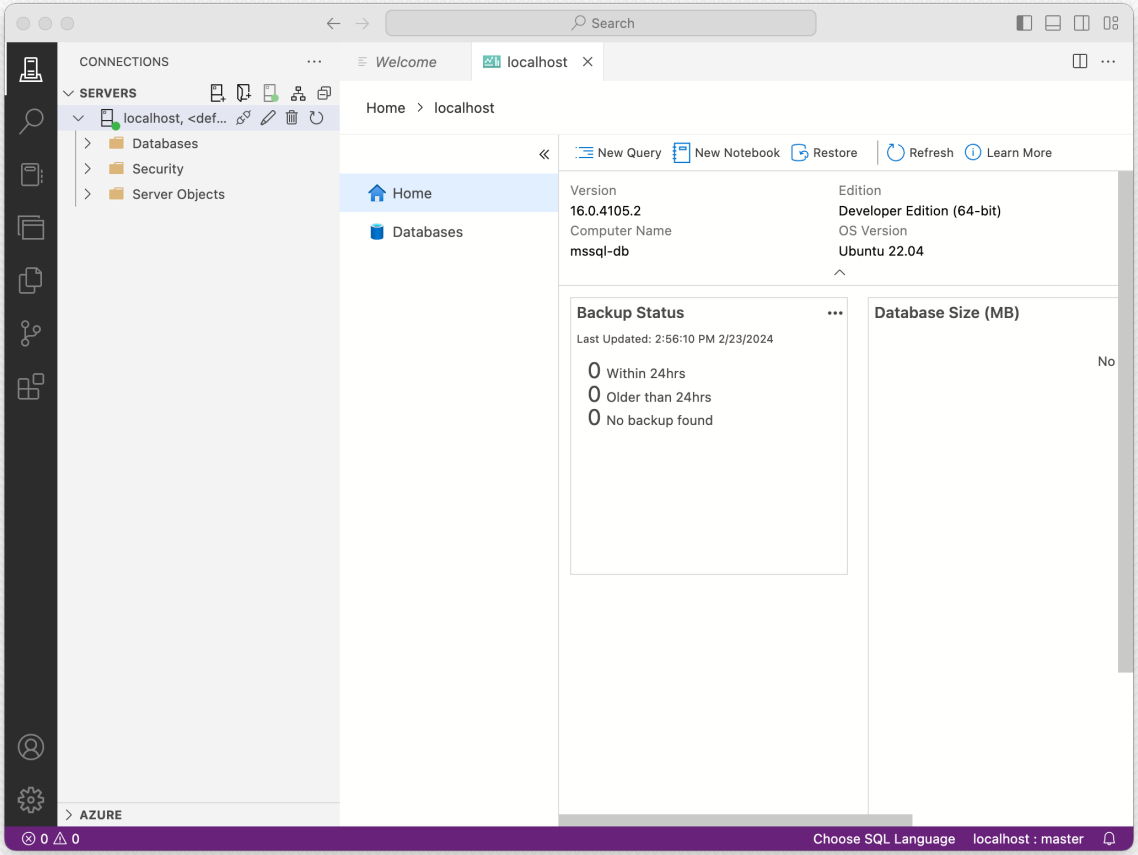
Setting up Azure Data Studio and SQL Server



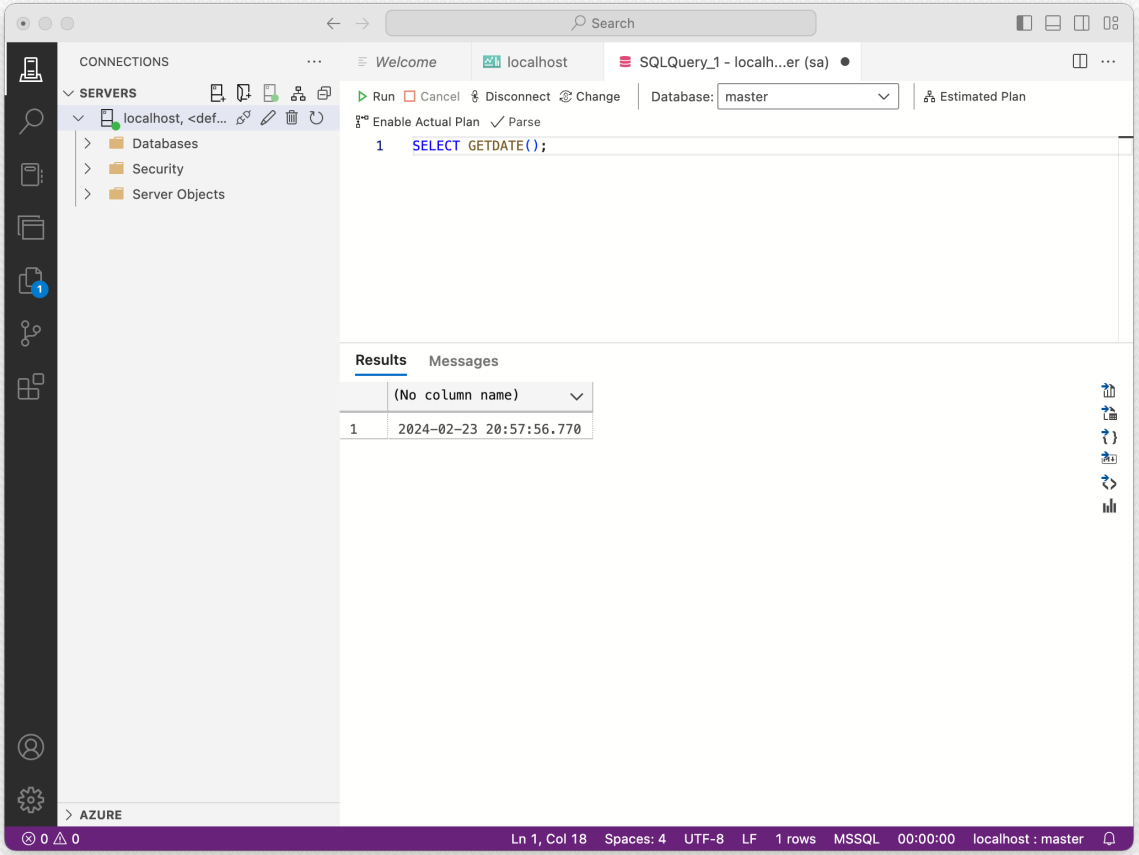
Setting up Azure Data Studio and MS SQL



Setting up Azure Data Studio and MS SQL



Setting up Azure Data Studio and MS SQL



Database Information

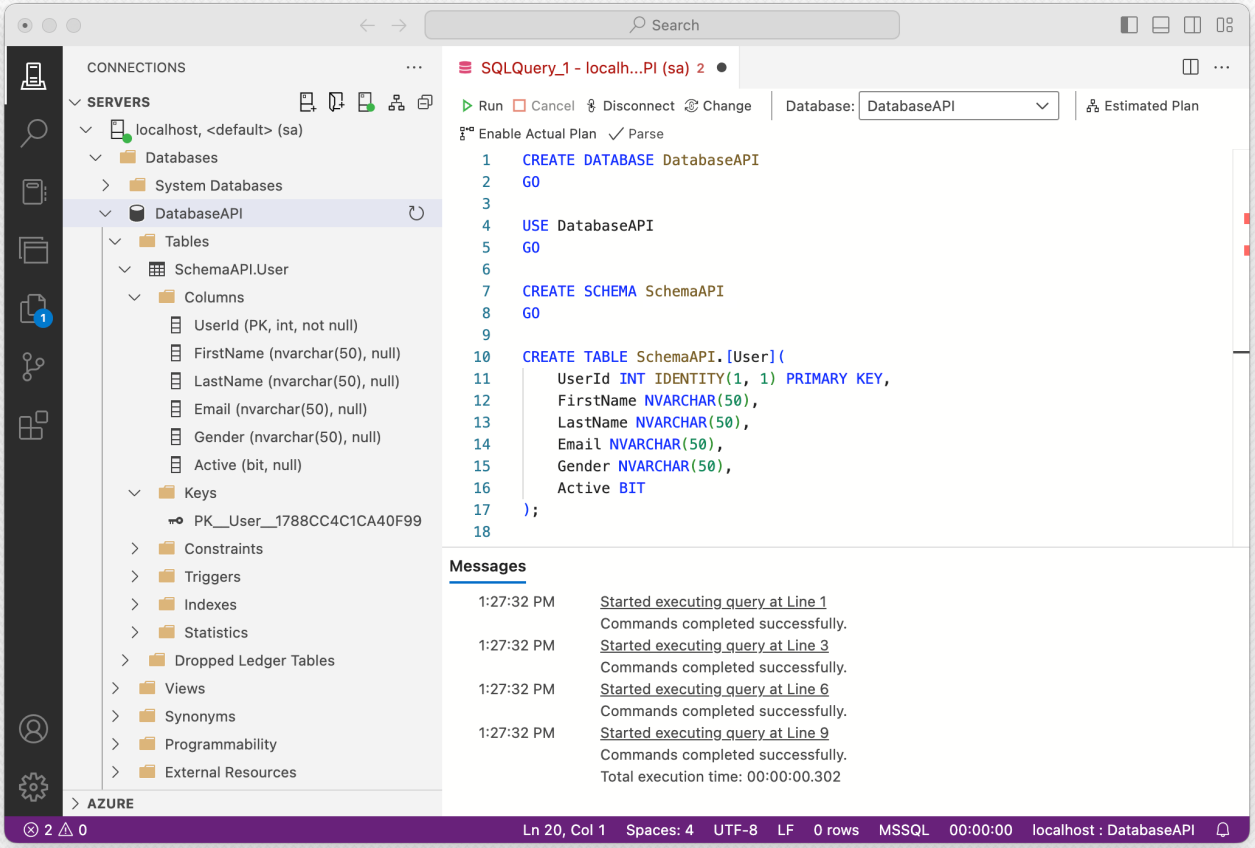
```
CREATE DATABASE DatabaseAPI
GO

USE DatabaseAPI
GO

CREATE SCHEMA SchemaAPI
GO

CREATE TABLE SchemaAPI.[User](
    UserId INT IDENTITY(1, 1) PRIMARY KEY,
    FirstName NVARCHAR(50),
    LastName NVARCHAR(50),
    Email NVARCHAR(50),
    Gender NVARCHAR(50),
    Active BIT
);
```

- Connection String:

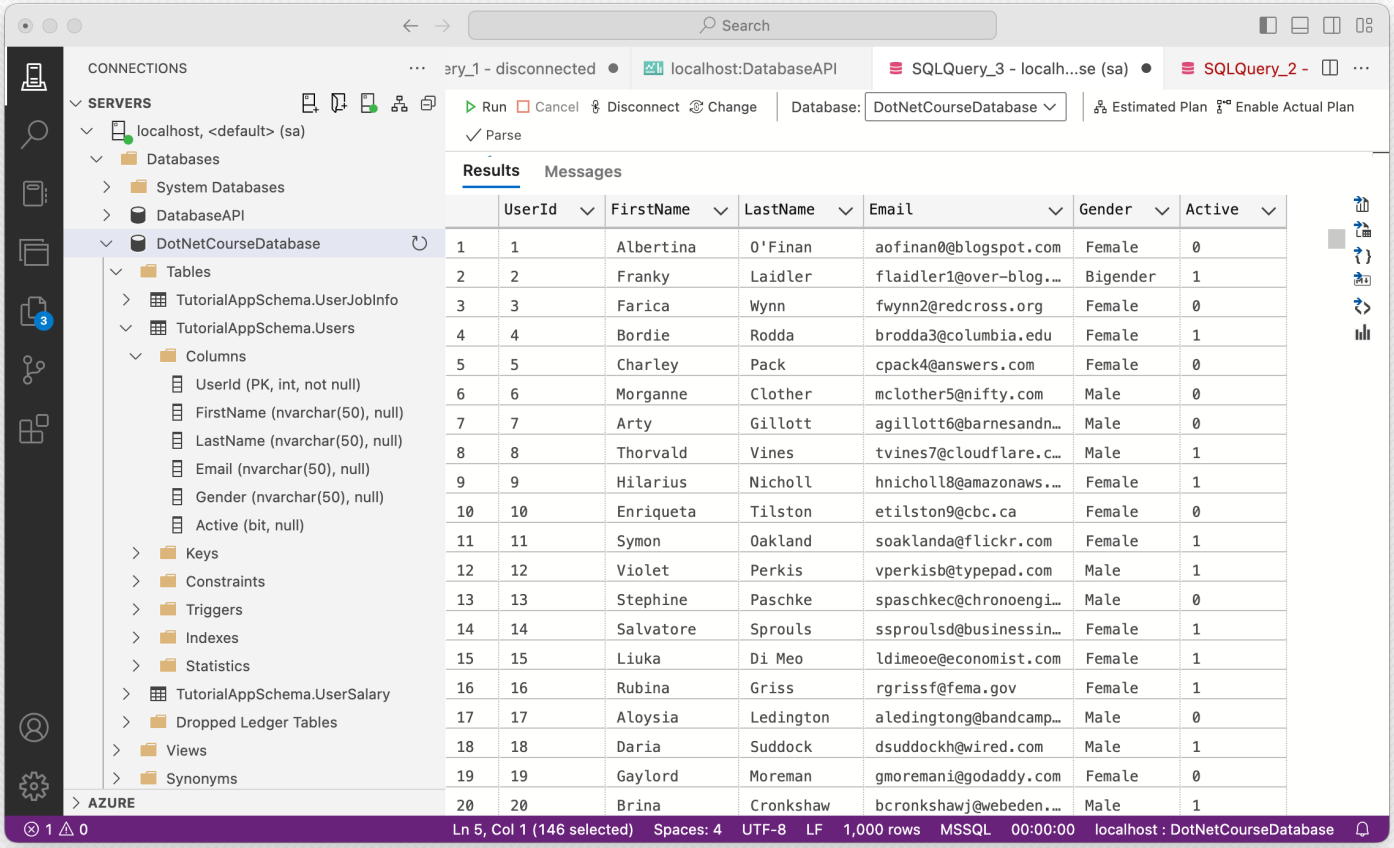
```
"Server=localhost;
Database=DatabaseAPI;
Trusted_Connection=false;
TrustedServerCertificate=true;
User Id=sa;
Password=Pa55w0rd;"
```

- All one string – 1 line

Database Creation



Database Testing – after seeding



	UserId	FirstName	LastName	Email	Gender	Active
1	1	Albertina	O'Finan	aofinan0@blogspot.com	Female	0
2	2	Franky	Laidler	flaidler1@over-blog.…	Bigender	1
3	3	Farica	Wynn	fwynn2@redcross.org	Female	0
4	4	Bordie	Rodda	brodda3@columbia.edu	Female	1
5	5	Charley	Pack	cpack4@answers.com	Female	0
6	6	Morganne	Clother	mclother5@nifty.com	Male	0
7	7	Arty	Gillott	agillott6@barnesandn…	Male	0
8	8	Thorvald	Vines	tvines7@cloudflare.c…	Male	1
9	9	Hilarius	Nicholl	hnicholl8@amazonaws.…	Female	1
10	10	Enriqueta	Tilston	etilston9@cbc.ca	Female	0
11	11	Symon	Oakland	soaklanda@flickr.com	Female	1
12	12	Violet	Perkis	vperkisb@typepad.com	Male	1
13	13	Stephine	Paschke	spaschkec@chronoengi…	Male	0
14	14	Salvatore	Sprouls	ssproulsd@businessin…	Female	1
15	15	Liuka	Di Meo	ldimeoe@economist.com	Female	1
16	16	Rubina	Griss	rgrissf@fema.gov	Female	1
17	17	Aloysia	Ledington	aledingtong@bandcamp…	Male	0
18	18	Daria	Suddock	dsuddockh@wired.com	Male	1
19	19	Gaylord	Moreman	gmoremani@godaddy.com	Female	0
20	20	Brina	Cronkshaw	bcronkshawj@webeden.…	Male	1

Acknowledgements

- docker – Docker Desktop
  - https://www.docker.com/products/docker-desktop/
- docker docs – Docker Compose
  - https://docs.docker.com/compose/
- dockerhub
  - https://hub.docker.com/
- SQL Server
  - https://www.microsoft.com/en-us/sql-server/sql-server-downloads
- Azure Data Studio
  - https://azure.microsoft.com/en-us/products/data-studio