Microsoft SQL Server On Containers & Kubernetes Explained

Author: Dinesha Akalanka

Microsoft SQL Server On Containers & Kubernetes Explained

Getting Started with MS SQL Server deployments on Docker Containers & Kubernetes

Author: Dinesha Akalanka

Table of Content

Contents

Preface

Over the past few years DevOps has become more popular and practical approach for the software industry. DevOps is a culture which consist of tools, practices, and philosophies. It provides ability to organization's to deliver services and applications in a quick manner with high quality standards.

In the DevOps culture, containers are one of main concept being used. The containers always relate to Docker and Kubernetes Services. Docker and Kubernetes provide a standard approach to deploy stateful application like Microsoft SQL Server.

After following this book, you will be able to install MS SQL Server on Docker containers and Azure Kubernetes services.

Who this book is for

This book is for the beginners who are seeking knowledge on Docker contains and Kubernetes services.

What this book Cover

Section 01: Docker Overview, gives you overview about Docker desktop. In the section, it provides practical examples of deploying MSSQL Server into Docker container.

Section 02: Kubernetes Overview, gives you overview Azure Kubernetes Services. In the section, it provides practical examples of deploying MS SQL Server on Azure Kubernetes Services (AKS).

Section 1
Docker Basics and Microsoft SQL Deployment

In this section, Docker Basics, Docker Desktop, and installation MSSQL on Docker image will be covered.

Section 1

Docker Desktop overview

Docker desktop is community base toll where we can use this software freely in our local environments.

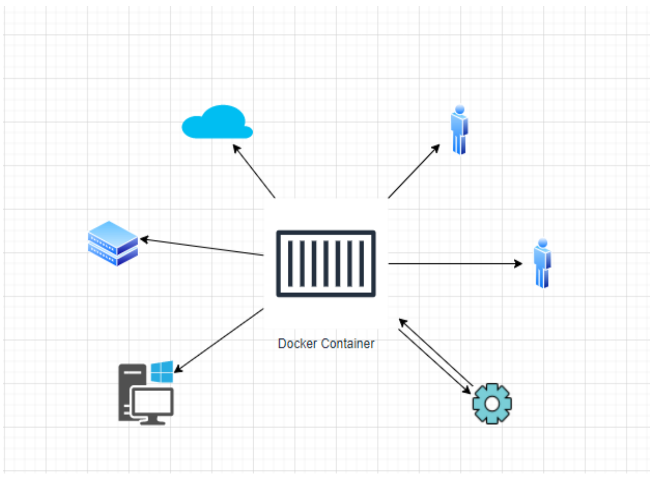
This provides more flexibility explore features in the docker containerize environment.

The following topics will be covered in this section.

- Docker Overview.
- Understanding Docker Architecture
- Installing Docker Desktop
- Installing MS SQL Server image on Docker Container.

What is docker container?

Docker container is a software where we can build, share, and deploy our applications into containerized environment. It’s a method of packaging the all the code and dependencies. This will enable us to run the applications more efficiently and quick manner.



Docker is an open platform for distributed environment.

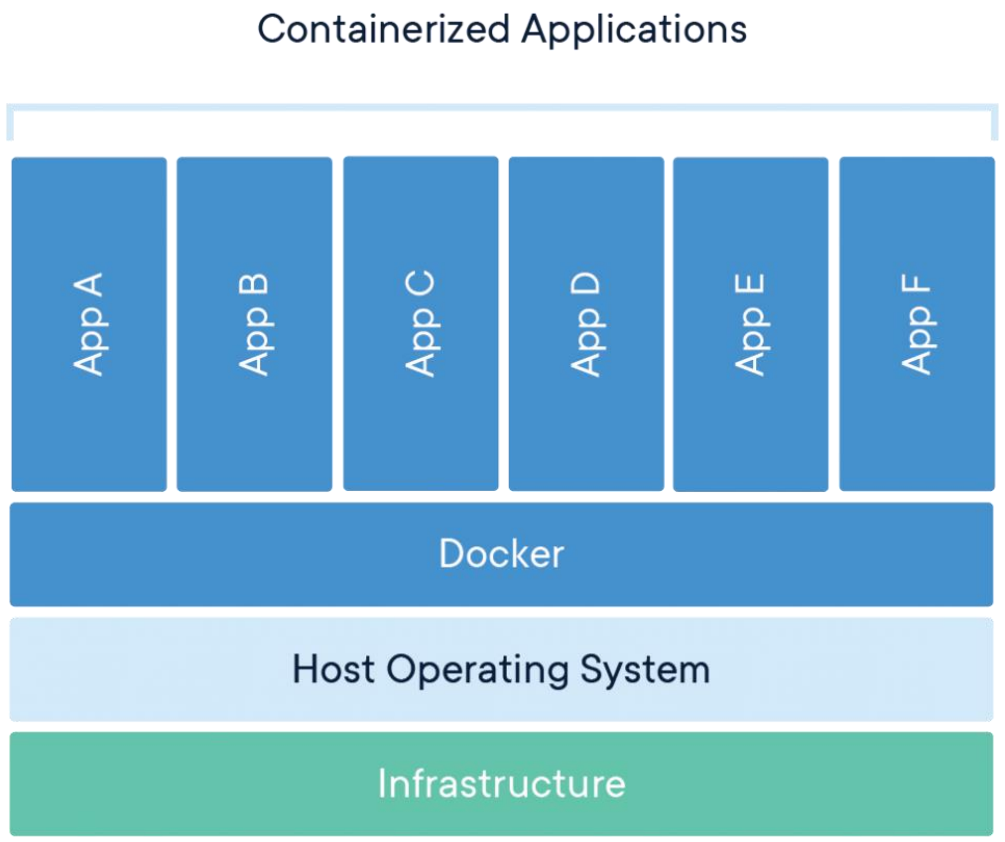
Docker Engine: lightweight application runtime with packaging environment.

Docker provide high portability.

Docker Hub: A cloud base service for sharing applications

Containerized Architecture

In a single host operating system, where we can run multiple application with minimum resource requirements. This containerization done in host operating system level. This will enable run multiple isolated applications without launching virtual machines. These applications run on single host environment using same kernel.



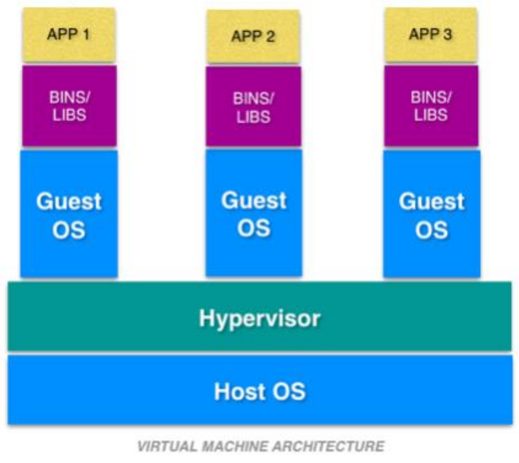
Typical Virtual machines are not required in Docker environment. Docker environment deployed on the host operating system. Docker engine holds the containerized applications.

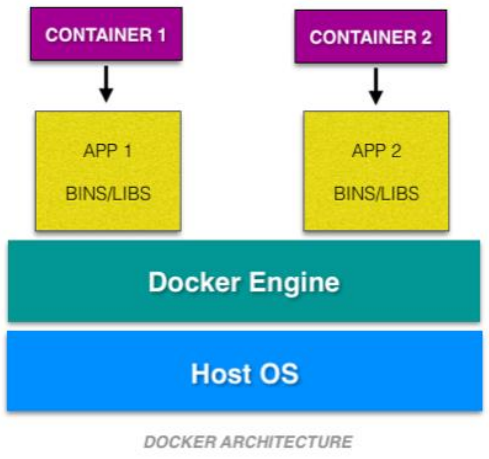
Docker Containers vs Virtual Machines

Docker containers provide more benefits over the typical virtual machines.

Docker Containers benefits.

- Guest operating systems not required.
- Less IT Management on resources
- Less time to setup a Dev or production environment
- Minimum software licenses required
- Reduce the upfront cost
- Easy to manage and reduce IT team workload





Docker containers that run on Docker Engine:

Standard: Docker provide high quality standards. Therefore, these containers provide more compatibility and portability.

Lightweight: All the containers run system kernel level. Therefore, no additional guest operating systems required.

Secure: With the high isolation of applications, Docker provide safer place.

Installing Docker Desktop

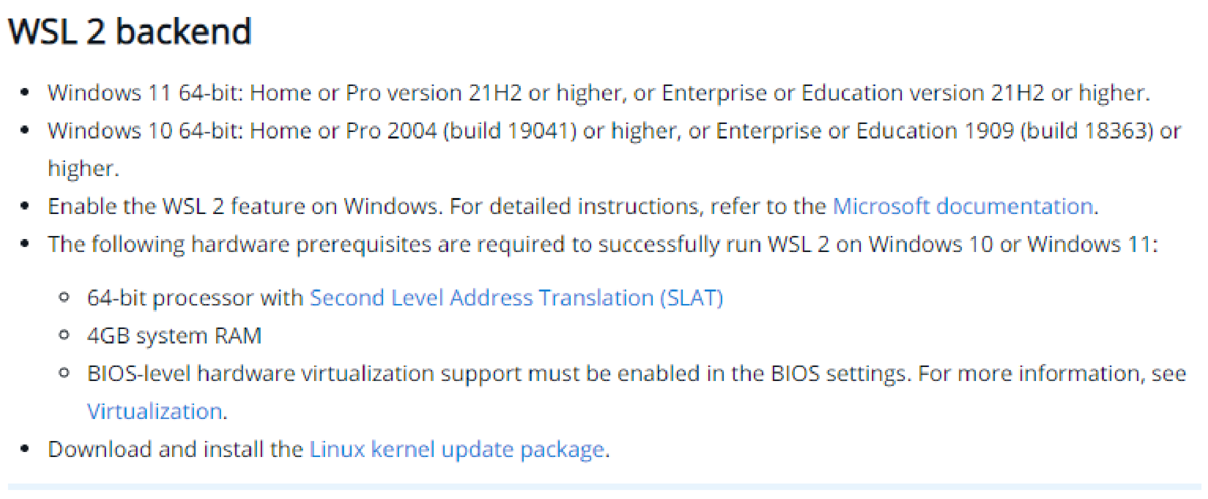
Let's start deploying Docker desktop.

Download and install the latest version of Docker Desktop. In this tutorial I will install docker desktop for windows

- Download for Mac
- Download for Windows

Alternatively, install the Docker Compose CLI for Linux.

Other system Requirements.



WSL 2 backend

- Windows 11 64-bit: Home or Pro version 21H2 or higher, or Enterprise or Education version 21H2 or higher.
- Windows 10 64-bit: Home or Pro 2004 (build 19041) or higher, or Enterprise or Education 1909 (build 18363) or higher.
- Enable the WSL 2 feature on Windows. For detailed instructions, refer to the Microsoft documentation.
- The following hardware prerequisites are required to successfully run WSL 2 on Windows 10 or Windows 11:
  - 64-bit processor with Second Level Address Translation (SLAT)
  - 4GB system RAM
  - BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see Virtualization.
- Download and install the Linux kernel update package.

You can fine more information from below link.

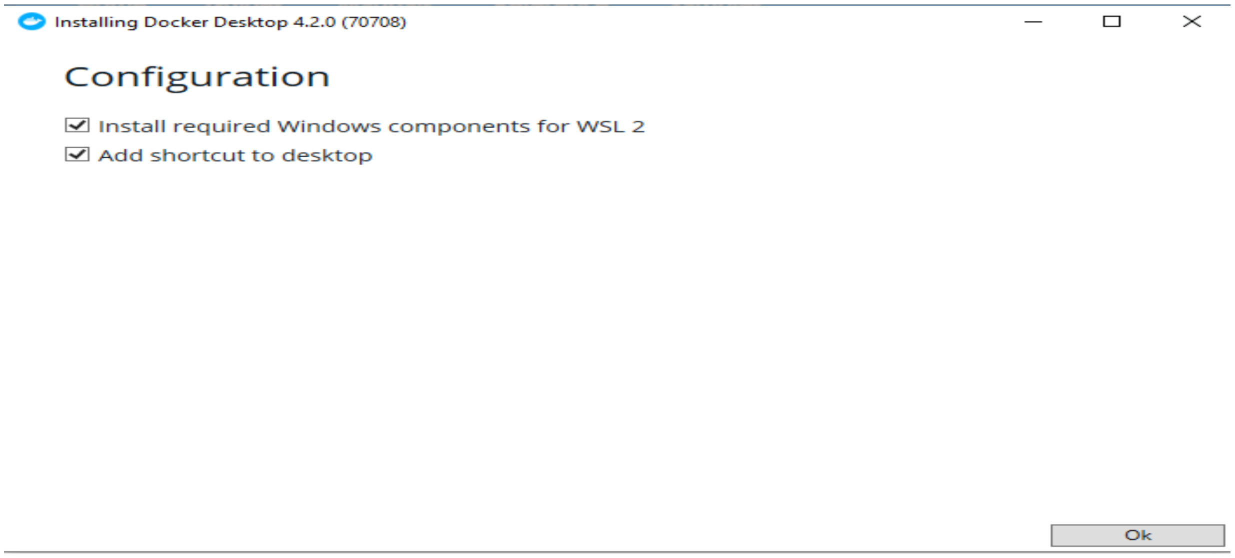
- https://docs.docker.com/desktop/windows/install/

Let’s start the software installation

Step 01: Doble click on the downloaded docker desktop installer.

Step 02: Enable Hyper-V Windows Features or the Install required Windows components for WSL 2

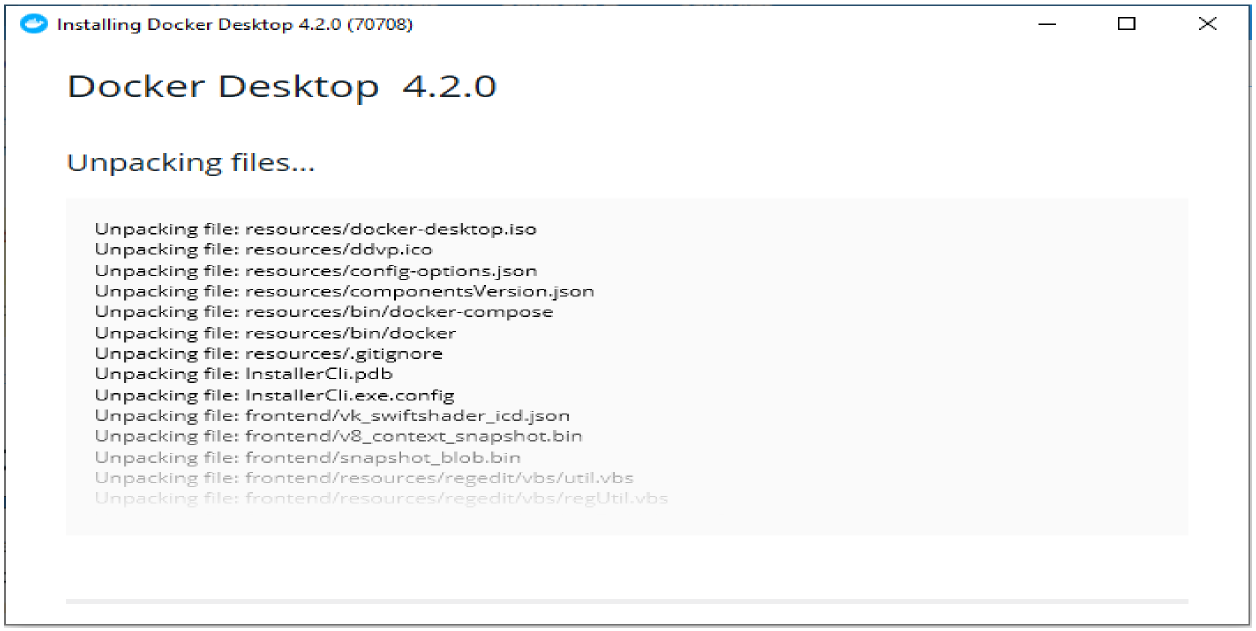
Step 03: Tick the required configurations and start installation.



Installing Docker Desktop 4.2.0 (70708)

Configuration

[✓] Install required Windows components for WSL 2
[✓] Add shortcut to desktop



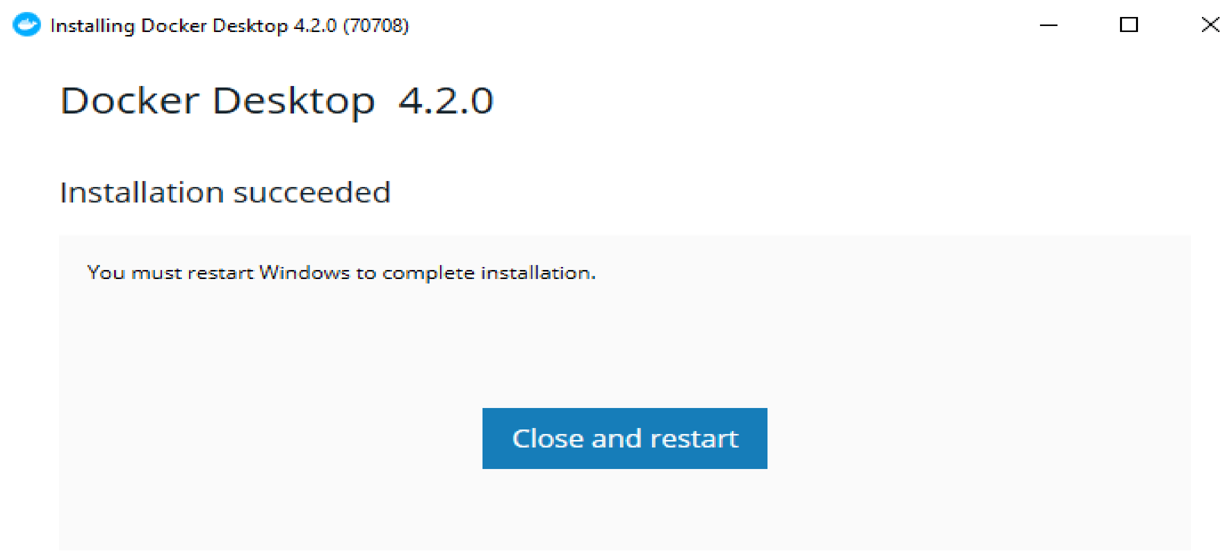
Installing Docker Desktop 4.2.0 (70708)

Docker Desktop 4.2.0

Unpacking files...

```
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/.gitignore
Unpacking file: InstallerCli.pdb
Unpacking file: InstallerCli.exe.config
Unpacking file: frontend/vk_swiftshader_icd.json
Unpacking file: frontend/v8_context_snapshot.bin
Unpacking file: frontend/snapshot_blob.bin
Unpacking file: frontend/resources/regedit/vbs/util.vbs
Unpacking file: frontend/resources/regedit/vbs/regUtil.vbs
```

Step 04: Restart Windows to complete the installation.

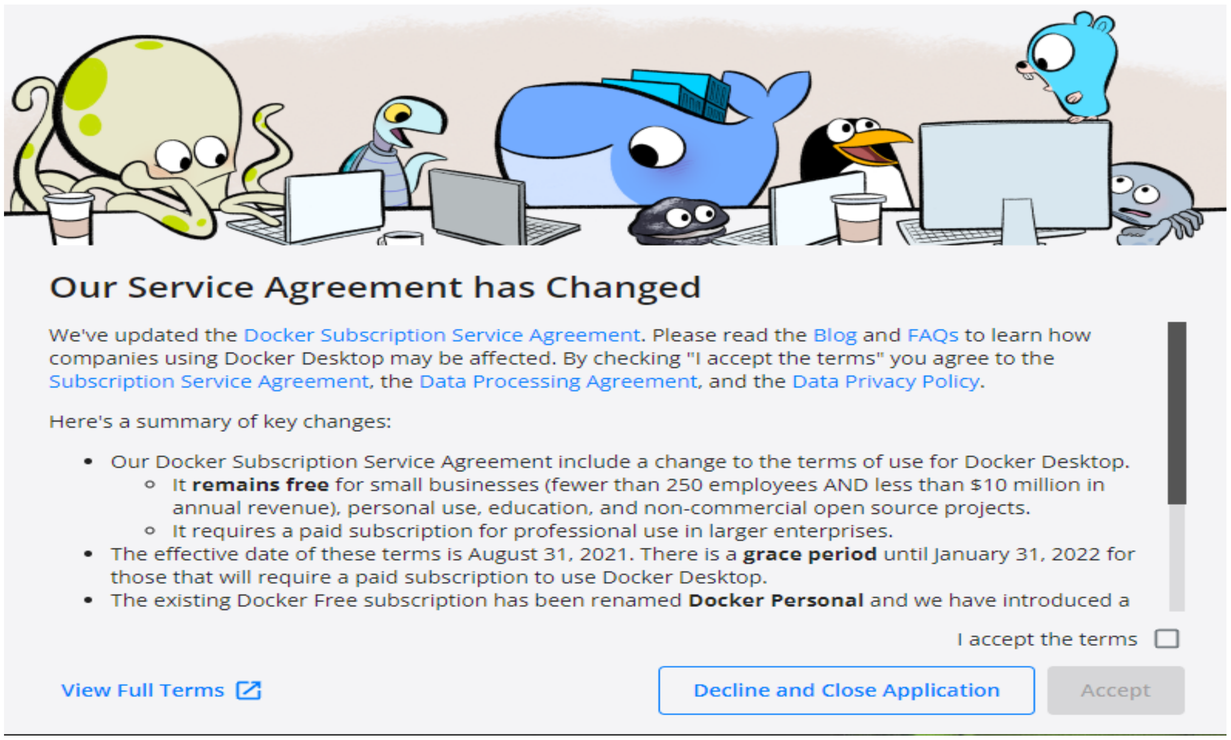

Installing Docker Desktop 4.2.0 (70708)

Docker Desktop 4.2.0

Installation succeeded

You must restart Windows to complete installation.

Step 05: In order to start using the docker engine you need to accept the service agreement

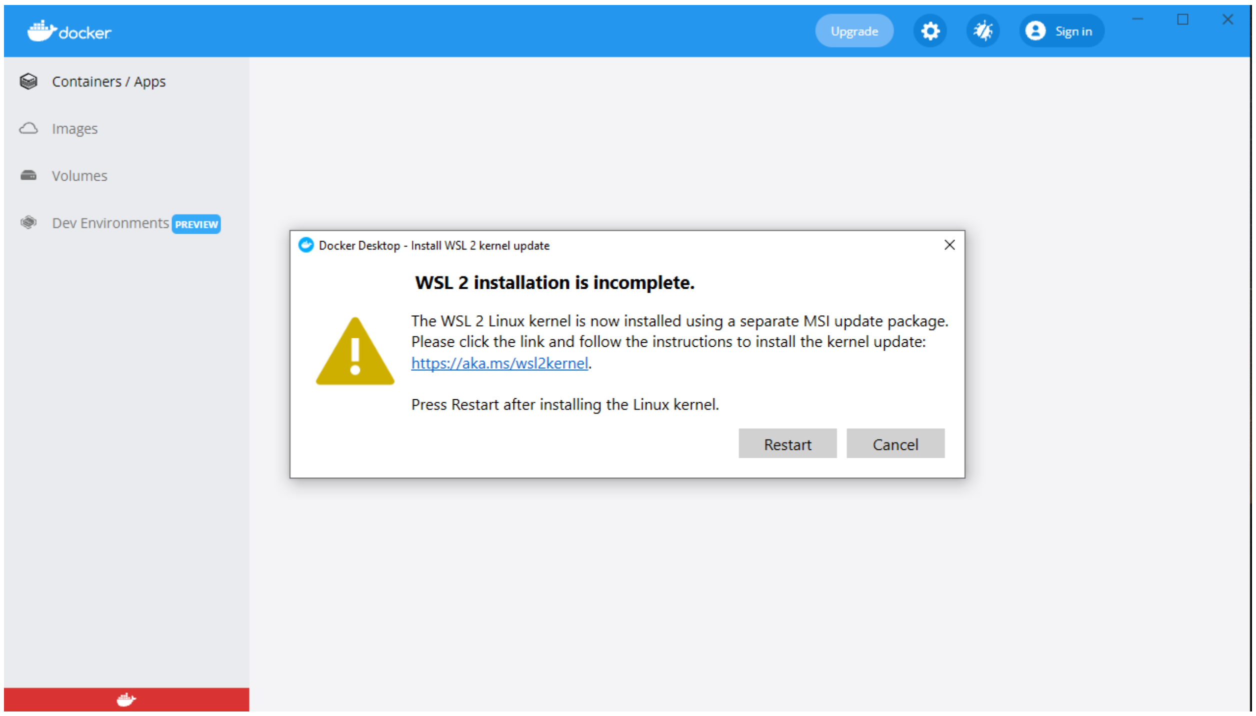

Our Service Agreement has Changed

We've updated the Docker Subscription Service Agreement. Please read the Blog and FAQs to learn how companies using Docker Desktop may be affected. By checking "I accept the terms" you agree to the Subscription Service Agreement, the Data Processing Agreement, and the Data Privacy Policy.

Here's a summary of key changes:

- Our Docker Subscription Service Agreement include a change to the terms of use for Docker Desktop.
  - It **remains free** for small businesses (fewer than 250 employees AND less than $10 million in annual revenue), personal use, education, and non-commercial open source projects.
  - It requires a paid subscription for professional use in larger enterprises.
- The effective date of these terms is August 31, 2021. There is a **grace period** until January 31, 2022 for those that will require a paid subscription to use Docker Desktop.
- The existing Docker Free subscription has been renamed **Docker Personal** and we have introduced a

Step 06: Restart the Docker engine.



WSL 2 installation is incomplete.

The WSL 2 Linux kernel is now installed using a separate MSI update package. Please click the link and follow the instructions to install the kernel update: https://aka.ms/wsl2kernel.

Press Restart after installing the Linux kernel.

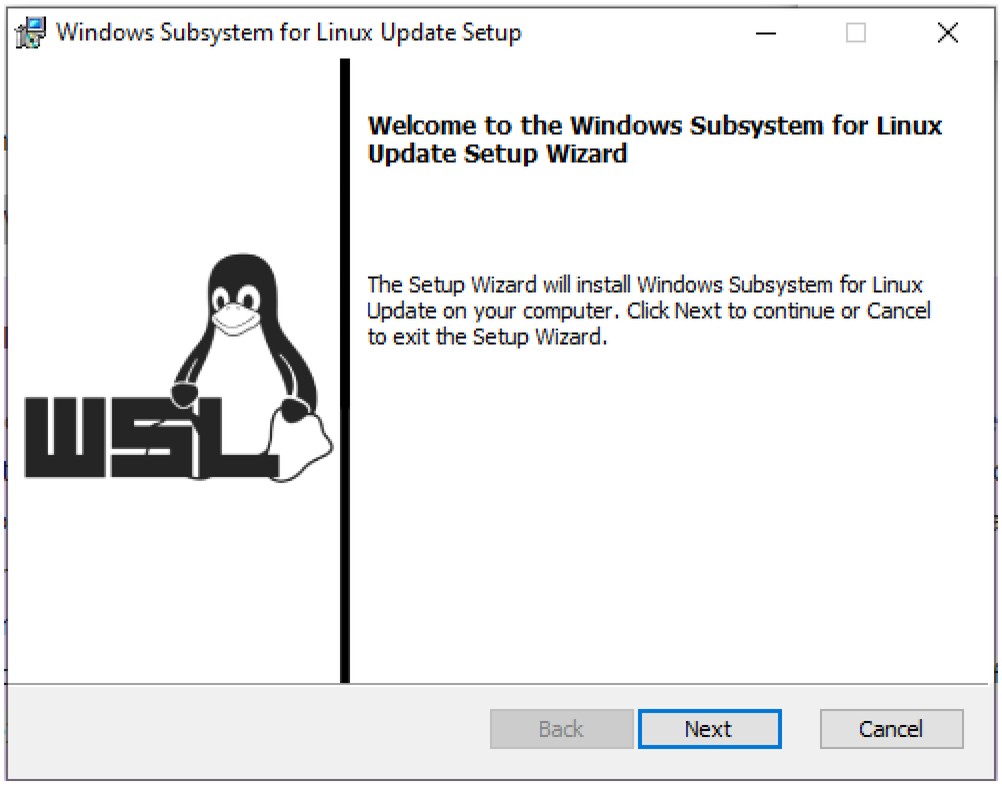
Sometime there can requirement to manual installation of WSL 2. Please find more information from below link to install older version of WSL.

Manual installation steps for older versions of WSL | Microsoft Docs

Use below link to download WSL2 Linux kernel update package for x64 machines

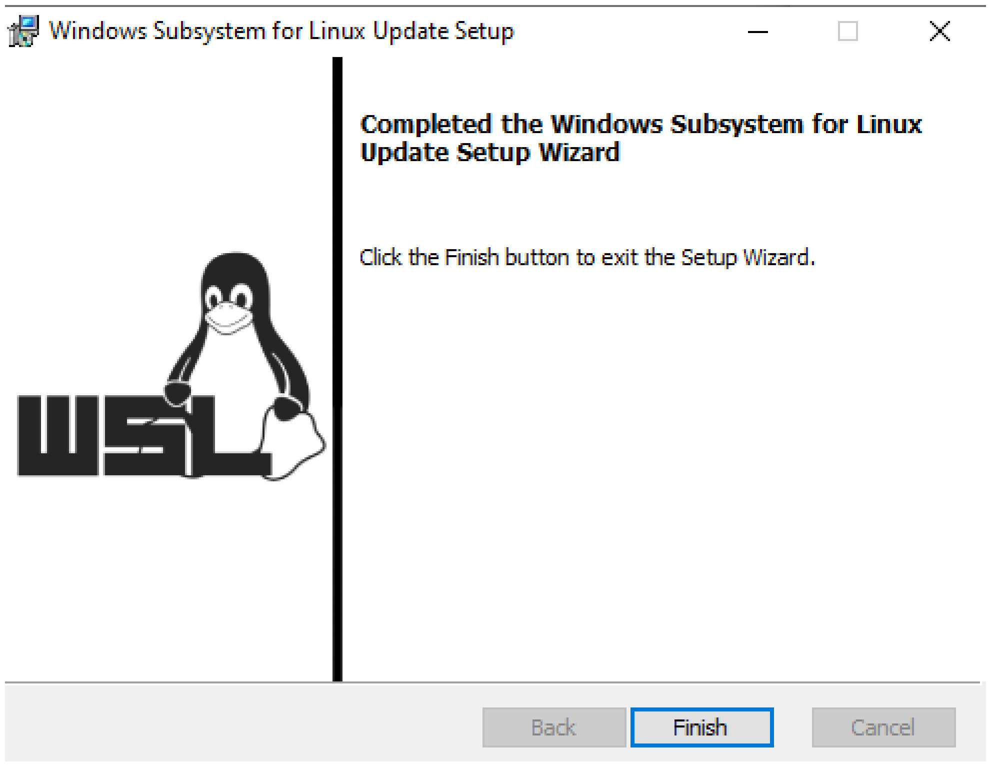
WSL2 Linux kernel update package for x64 machines

Linux Update setup


Windows Subsystem for Linux Update Setup

Welcome to the Windows Subsystem for Linux Update Setup Wizard

The Setup Wizard will install Windows Subsystem for Linux Update on your computer. Click Next to continue or Cancel to exit the Setup Wizard.


Windows Subsystem for Linux Update Setup

Completed the Windows Subsystem for Linux Update Setup Wizard

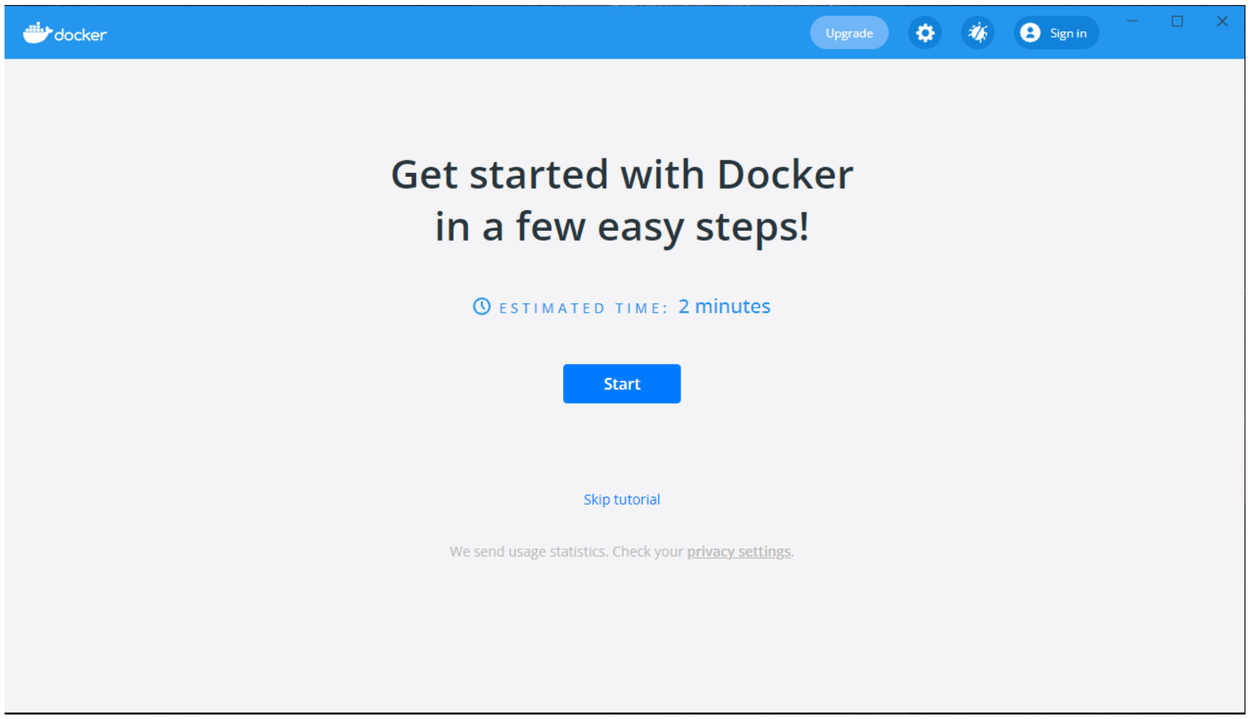
Click the Finish button to exit the Setup Wizard.

Getting Started with Docker

Follow the tutorial to familiar with docker environment. Docker engines provide tutorial for the beginners. This will find very useful if you are new to Docker.

In the tutorial, it will cover

- Clone a repository
- Build Image
- Run Container
- Share Container

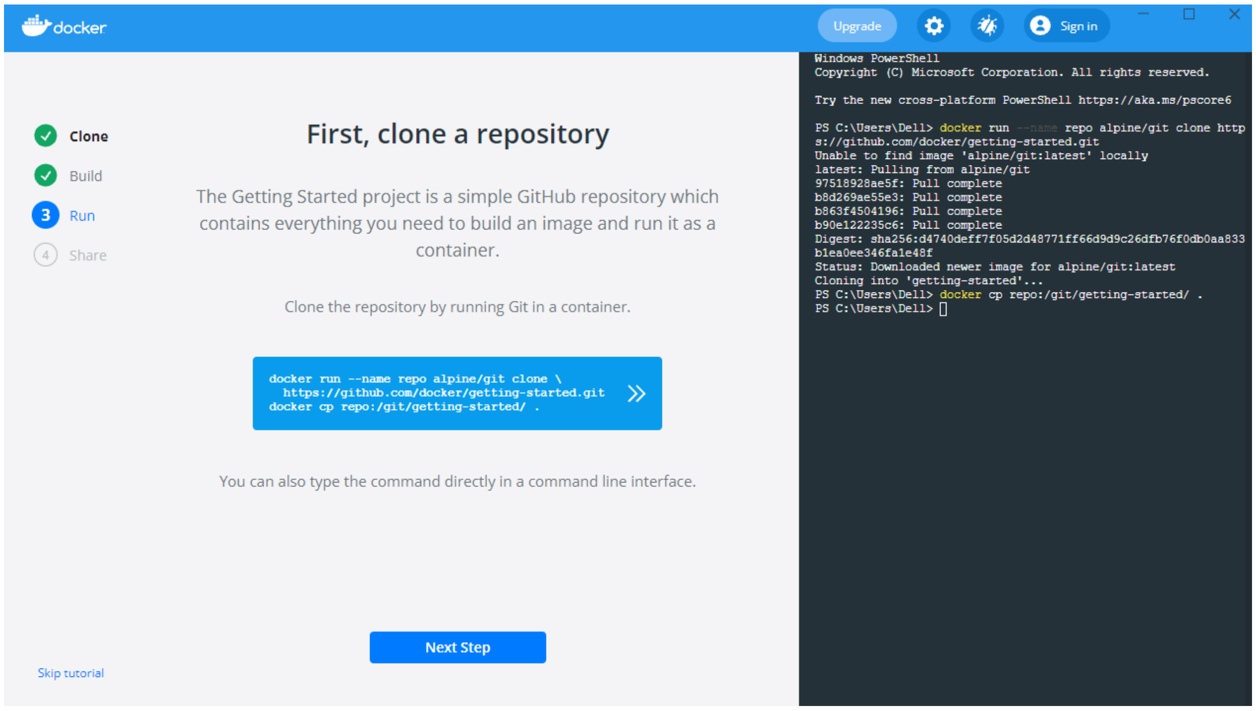


Get started with Docker
in a few easy steps!

ESTIMATED TIME: 2 minutes

We send usage statistics. Check your privacy settings.

Step 01: Clone a repository.



First, clone a repository

The Getting Started project is a simple GitHub repository which contains everything you need to build an image and run it as a container.

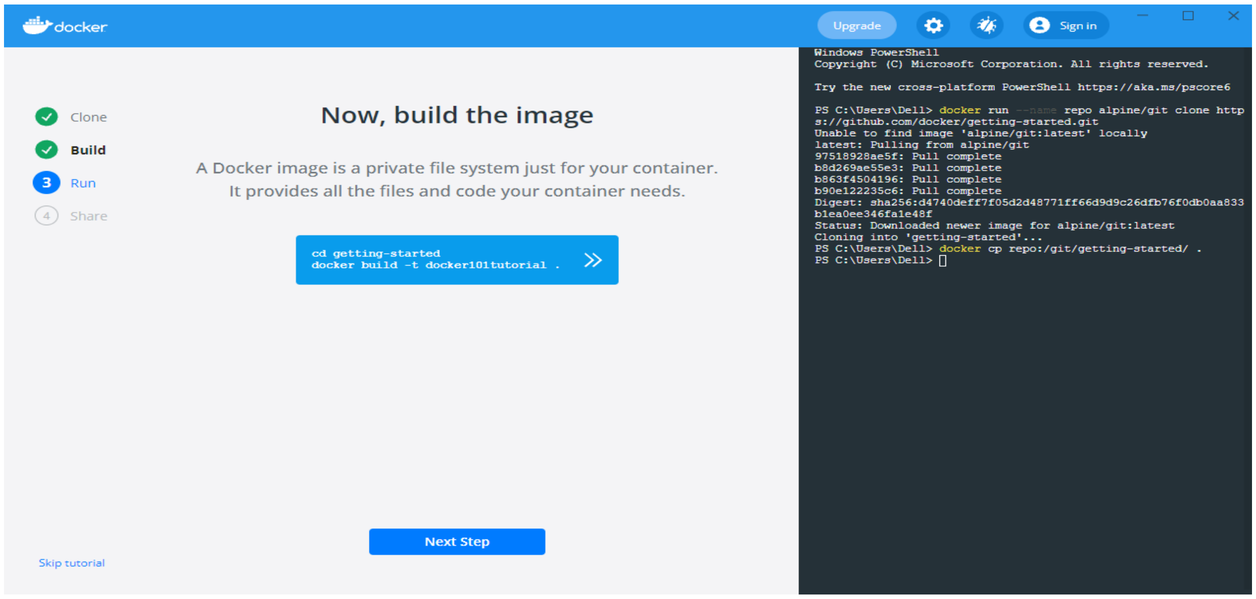
Clone the repository by running Git in a container.

```
docker run --name repo alpine/git clone \
  https://github.com/docker/getting-started.git
docker cp repo:/git/getting-started/ .
```

You can also type the command directly in a command line interface.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Dell> docker run --name repo alpine/git clone http
s://github.com/docker/getting-started.git
Unable to find image 'alpine/git:latest' locally
latest: Pulling from alpine/git
97518928ae5f: Pull complete
b8d269ae55e3: Pull complete
b863f4504196: Pull complete
b90e122235c6: Pull complete
Digest: sha256:d4740deff7f05d2d48771ff66d9d9c26dfb76f0db0aa833
b1ea0ee346fa1e48f
Status: Downloaded newer image for alpine/git:latest
Cloning into 'getting-started'...
PS C:\Users\Dell> docker cp repo:/git/getting-started/ .
PS C:\Users\Dell> 
```

Step 02: Build the image. Sample CLI available to test the command.



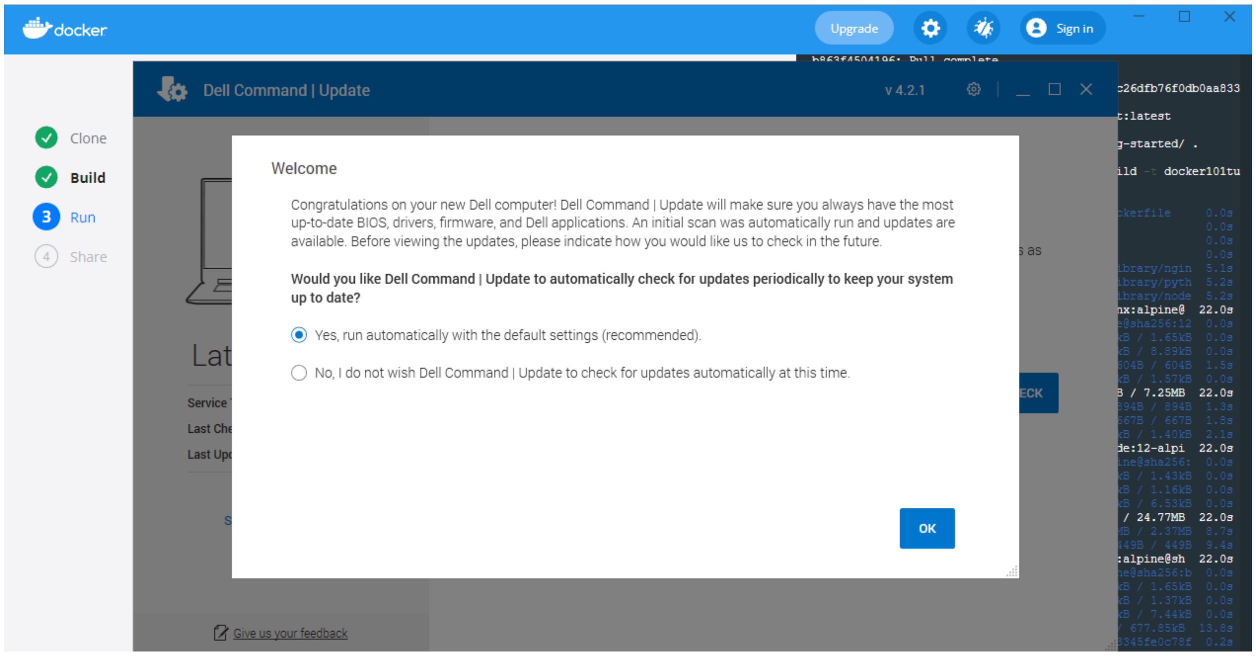
Now, build the image

A Docker image is a private file system just for your container. It provides all the files and code your container needs.

```
cd getting-started
docker build -t docker101tutorial .
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Dell> docker run --name repo alpine/git clone http
s://github.com/docker/getting-started.git
Unable to find image 'alpine/git:latest' locally
latest: Pulling from alpine/git
97518928ae5f: Pull complete
b8d269ae55e3: Pull complete
b863f4504196: Pull complete
b90e122235c6: Pull complete
Digest: sha256:d4740deff7f05d2d48771ff66d9d9c26dfb76f0db0aa833
b1ea0ee346fa1e48f
Status: Downloaded newer image for alpine/git:latest
Cloning into 'getting-started'...
PS C:\Users\Dell> docker cp repo:/git/getting-started/ .
PS C:\Users\Dell> 
```

PC Manufacturer update on the command line.



Welcome

Congratulations on your new Dell computer! Dell Command | Update will make sure you always have the most up-to-date BIOS, drivers, firmware, and Dell applications. An initial scan was automatically run and updates are available. Before viewing the updates, please indicate how you would like us to check in the future.

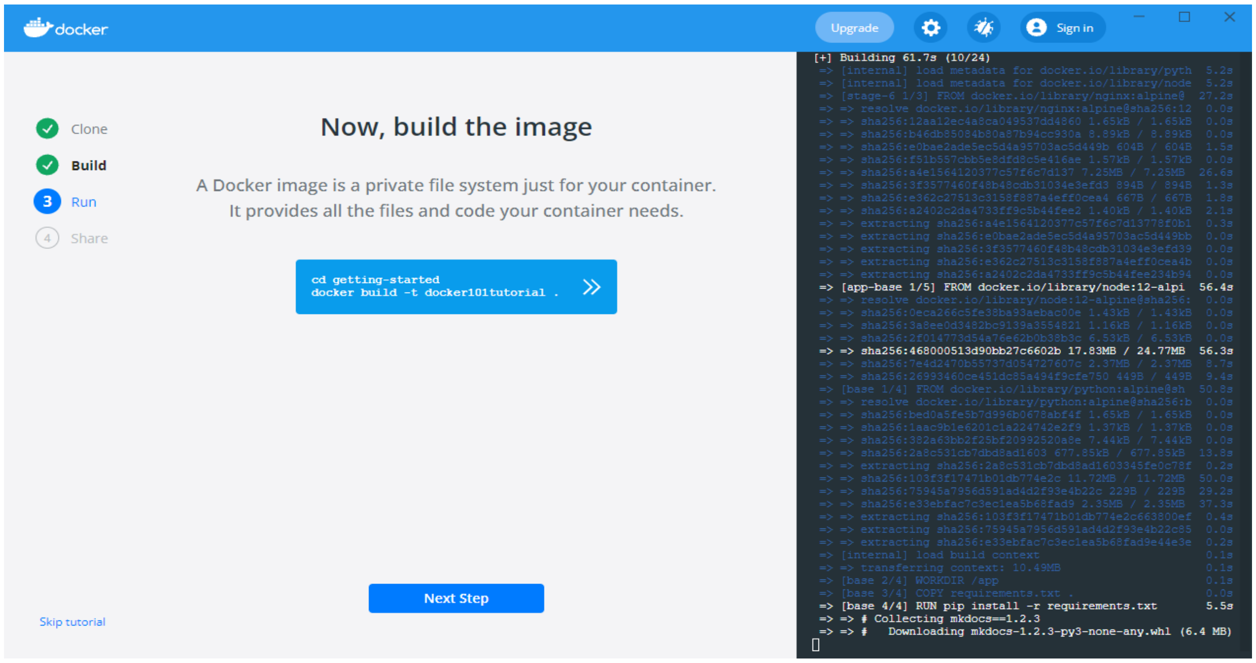
Would you like Dell Command | Update to automatically check for updates periodically to keep your system up to date?

[✓ Yes, run automatically with the default settings (recommended).]

No, I do not wish Dell Command | Update to check for updates automatically at this time.

OK

```
b863f4504196: Pull complete
c26dfb76f0db0aa833
t:latest
g-started/ .
ild -t docker101tu

ckerfile      0.0s
              0.0s
              0.0s
              0.0s
ibrary/ngin   5.1s
ibrary/pyth   5.2s
ibrary/node   5.2s
nx:alpine@   22.0s
e@sha256:12   0.0s
kB / 1.65kB   0.0s
kB / 8.89kB   0.0s
604B / 604B   1.5s
kB / 1.57kB   0.0s
B / 7.25MB   22.0s
894B / 894B   1.3s
667B / 667B   1.8s
kB / 1.40kB   2.1s
de:12-alpi   22.0s
ine@sha256:   0.0s
kB / 1.43kB   0.0s
kB / 1.16kB   0.0s
kB / 6.53kB   0.0s
 / 24.77MB   22.0s
MB / 2.37MB   8.7s
449B / 449B   9.4s
:alpine@sh   22.0s
ne@sha256:b   0.0s
kB / 1.65kB   0.0s
kB / 1.37kB   0.0s
kB / 7.44kB   0.0s
/ 677.85kB   13.8s
3345fe0c78f   0.2s
```



Now, build the image

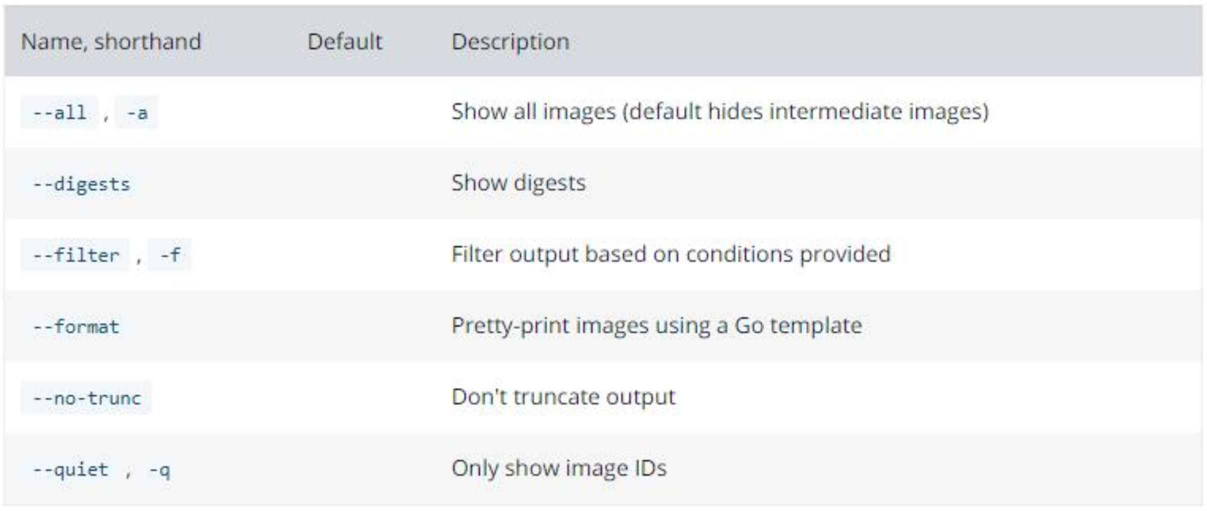
A Docker image is a private file system just for your container. It provides all the files and code your container needs.

```
cd getting-started
docker build -t docker101tutorial .
```

```
[+] Building 61.7s (10/24)
 => [internal] load metadata for docker.io/library/pyth  5.2s
 => [internal] load metadata for docker.io/library/node  5.2s
 => [stage-6 1/3] FROM docker.io/library/nginx:alpine@  27.2s
 => => resolve docker.io/library/nginx:alpine@sha256:12  0.0s
 => => sha256:12aa12ec4a8ca049537dd4860 1.65kB / 1.65kB  0.0s
 => => sha256:b46db85084b80a87b94cc930a 8.89kB / 8.89kB  0.0s
 => => sha256:e0bae2ade5ec5d4a95703ac5d449b 604B / 604B  1.5s
 => => sha256:f51b557cbb5e8dfd8c5e416ae 1.57kB / 1.57kB  0.0s
 => => sha256:a4e1564120377c57f6c7d137 7.25MB / 7.25MB  26.6s
 => => sha256:3f3577460f48b48cdb31034e3efd3 894B / 894B  1.3s
 => => sha256:e362c27513c3158f887a4eff0cea4 667B / 667B  1.8s
 => => sha256:a2402c2da4733ff9c5b44fee2 1.40kB / 1.40kB  2.1s
 => => extracting sha256:a4e1564120377c57f6c7d13778f0b1  0.3s
 => => extracting sha256:e0bae2ade5ec5d4a95703ac5d449bb  0.0s
 => => extracting sha256:3f3577460f48b48cdb31034e3efd39  0.0s
 => => extracting sha256:e362c27513c3158f887a4eff0cea4b  0.0s
 => => extracting sha256:a2402c2da4733ff9c5b44fee234b94  0.0s
 => [app-base 1/5] FROM docker.io/library/node:12-alpi  56.4s
 => => resolve docker.io/library/node:12-alpine@sha256:  0.0s
 => => sha256:0eca266c5fe38ba93aebac00e 1.43kB / 1.43kB  0.0s
 => => sha256:3a8ee0d3482bc9139a3554821 1.16kB / 1.16kB  0.0s
 => => sha256:2f014773d54a76e62b0b38b3c 6.53kB / 6.53kB  0.0s
 => => sha256:468000513d90bb27c6602b 17.83MB / 24.77MB  56.3s
 => => sha256:7e4d2470b55737d054727607c 2.37MB / 2.37MB  8.7s
 => => sha256:26993460ce451dc85a494f9cfe750 449B / 449B  9.4s
 => [base 1/4] FROM docker.io/library/python:alpine@sh  50.8s
 => => resolve docker.io/library/python:alpine@sha256:b  0.0s
 => => sha256:bed0a5fe5b7d996b0678abf4f 1.65kB / 1.65kB  0.0s
 => => sha256:1aac9b1e6201c1a224742e2f9 1.37kB / 1.37kB  0.0s
 => => sha256:382a63bb2f25bf20992520a8e 7.44kB / 7.44kB  0.0s
 => => sha256:2a8c531cb7dbd8ad1603 677.85kB / 677.85kB  13.8s
 => => extracting sha256:2a8c531cb7dbd8ad1603345fe0c78f  0.2s
 => => sha256:103f3f17471b01db774e2c 11.72MB / 11.72MB  50.0s
 => => sha256:75945a7956d591ad4d2f93e4b22c 229B / 229B  29.2s
 => => sha256:e33ebfac7c3ec1ea5b68fad9 2.35MB / 2.35MB  37.3s
 => => extracting sha256:103f3f17471b01db774e2c663800ef  0.4s
 => => extracting sha256:75945a7956d591ad4d2f93e4b22c85  0.0s
 => => extracting sha256:e33ebfac7c3ec1ea5b68fad9e44e3e  0.2s
 => [internal] load build context                        0.1s
 => => transferring context: 10.49MB                     0.1s
 => [base 2/4] WORKDIR /app                              0.1s
 => [base 3/4] COPY requirements.txt .                   0.0s
 => [base 4/4] RUN pip install -r requirements.txt       5.5s
 => => # Collecting mkdocs==1.2.3
 => => #   Downloading mkdocs-1.2.3-py3-none-any.whl (6.4 MB)
```

Docker image command to retrieve information of images.

```
docker images [OPTIONS] [REPOSITORY[:TAG]]
```



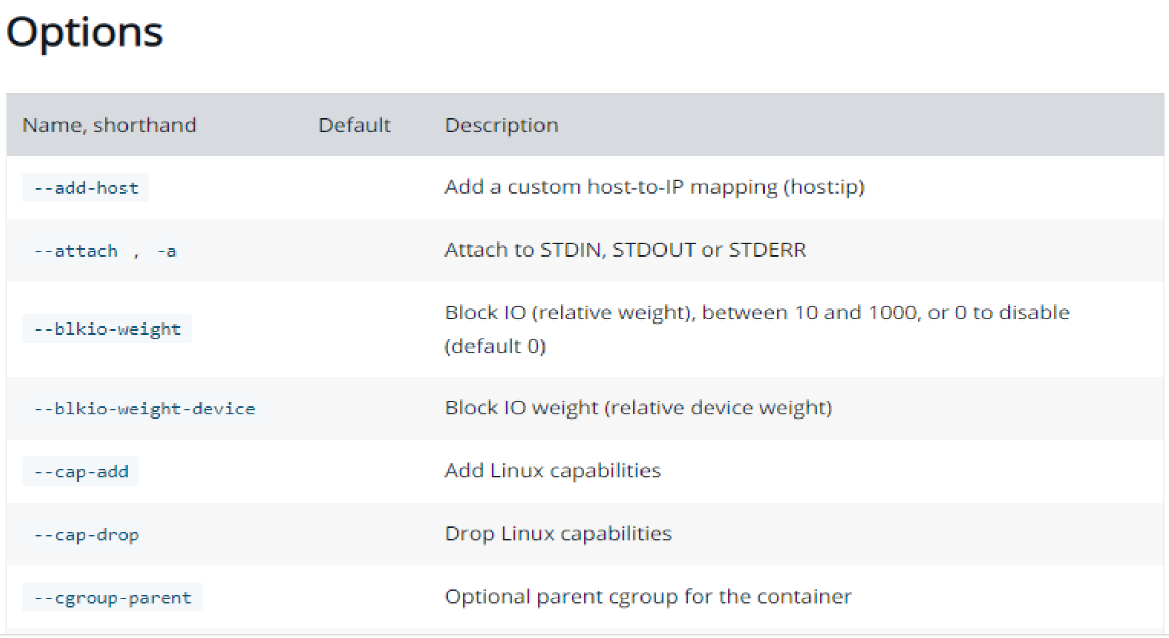
Name, shorthand	Default	Description
`--all` , `-a`		Show all images (default hides intermediate images)
`--digests`		Show digests
`--filter` , `-f`		Filter output based on conditions provided
`--format`		Pretty-print images using a Go template
`--no-trunc`		Don't truncate output
`--quiet` , `-q`		Only show image IDs

You can find more information on these commands using below link.

https://docs.docker.com/engine/reference/commandline/images/

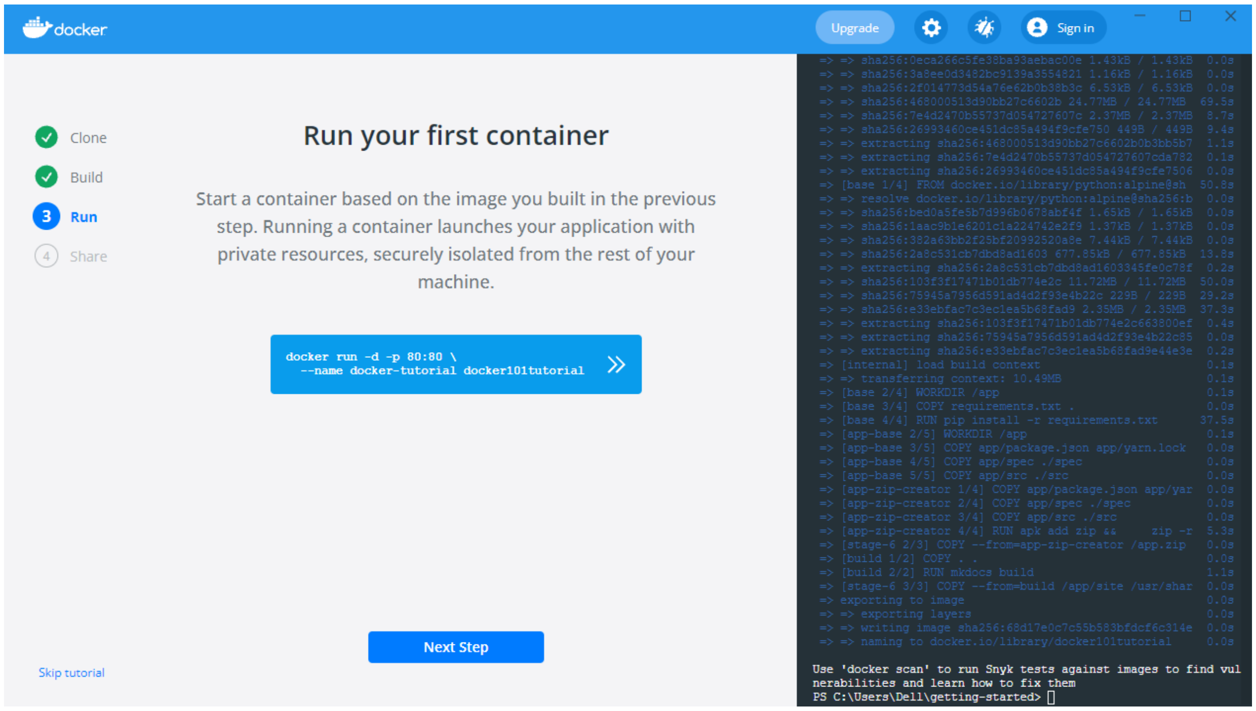
Step 03: Run the container

```
docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
```



Options

Name, shorthand	Default	Description
`--add-host`		Add a custom host-to-IP mapping (host:ip)
`--attach` , `-a`		Attach to STDIN, STDOUT or STDERR
`--blkio-weight`		Block IO (relative weight), between 10 and 1000, or 0 to disable (default 0)
`--blkio-weight-device`		Block IO weight (relative device weight)
`--cap-add`		Add Linux capabilities
`--cap-drop`		Drop Linux capabilities
`--cgroup-parent`		Optional parent cgroup for the container

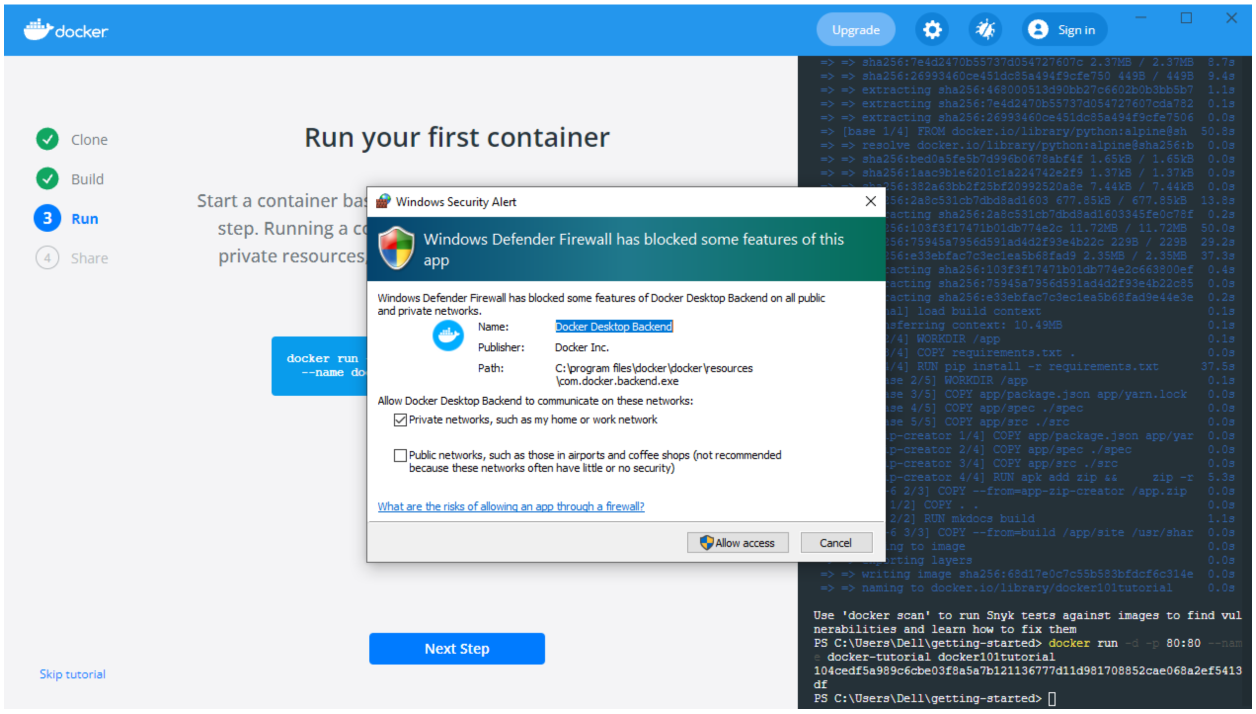


Run your first container

Start a container based on the image you built in the previous step. Running a container launches your application with private resources, securely isolated from the rest of your machine.

```
docker run -d -p 80:80 \
    --name docker-tutorial docker101tutorial
```

```
 => => sha256:0eca266c5fe38ba93aebac00e 1.43kB / 1.43kB  0.0s
 => => sha256:3a8ee0d3482bc9139a3554821 1.16kB / 1.16kB  0.0s
 => => sha256:2f014773d54a76e62b0b38b3c 6.53kB / 6.53kB  0.0s
 => => sha256:468000513d90bb27c6602b 24.77MB / 24.77MB  69.5s
 => => sha256:7e4d2470b55737d054727607c 2.37MB / 2.37MB  8.7s
 => => sha256:26993460ce451dc85a494f9cfe750 449B / 449B  9.4s
 => => extracting sha256:468000513d90bb27c6602b0b3bb5b7  1.1s
 => => extracting sha256:7e4d2470b55737d054727607cda782  0.1s
 => => extracting sha256:26993460ce451dc85a494f9cfe7506  0.0s
 => [base 1/4] FROM docker.io/library/python:alpine@sh  50.8s
 => => resolve docker.io/library/python:alpine@sha256:b  0.0s
 => => sha256:bed0a5fe5b7d996b0678abf4f 1.65kB / 1.65kB  0.0s
 => => sha256:1aac9b1e6201c1a224742e2f9 1.37kB / 1.37kB  0.0s
 => => sha256:382a63bb2f25bf20992520a8e 7.44kB / 7.44kB  0.0s
 => => sha256:2a8c531cb7dbd8ad1603 677.85kB / 677.85kB  13.8s
 => => extracting sha256:2a8c531cb7dbd8ad1603345fe0c78f  0.2s
 => => sha256:103f3f17471b01db774e2c 11.72MB / 11.72MB  50.0s
 => => sha256:75945a7956d591ad4d2f93e4b22c 229B / 229B  29.2s
 => => sha256:e33ebfac7c3ec1ea5b68fad9 2.35MB / 2.35MB  37.3s
 => => extracting sha256:103f3f17471b01db774e2c663800ef  0.4s
 => => extracting sha256:75945a7956d591ad4d2f93e4b22c85  0.0s
 => => extracting sha256:e33ebfac7c3ec1ea5b68fad9e44e3e  0.2s
 => [internal] load build context                        0.1s
 => => transferring context: 10.49MB                     0.1s
 => [base 2/4] WORKDIR /app                              0.1s
 => [base 3/4] COPY requirements.txt .                   0.0s
 => [base 4/4] RUN pip install -r requirements.txt      37.5s
 => [app-base 2/5] WORKDIR /app                          0.1s
 => [app-base 3/5] COPY app/package.json app/yarn.lock   0.0s
 => [app-base 4/5] COPY app/spec ./spec                  0.0s
 => [app-base 5/5] COPY app/src ./src                    0.0s
 => [app-zip-creator 1/4] COPY app/package.json app/yar  0.0s
 => [app-zip-creator 2/4] COPY app/spec ./spec           0.0s
 => [app-zip-creator 3/4] COPY app/src ./src             0.0s
 => [app-zip-creator 4/4] RUN apk add zip &&     zip -r  5.3s
 => [stage-6 2/3] COPY --from=app-zip-creator /app.zip   0.0s
 => [build 1/2] COPY . .                                 0.0s
 => [build 2/2] RUN mkdocs build                         1.1s
 => [stage-6 3/3] COPY --from=build /app/site /usr/shar  0.0s
 => exporting to image                                   0.0s
 => => exporting layers                                  0.0s
 => => writing image sha256:68d17e0c7c55b583bfdcf6c314e  0.0s
 => => naming to docker.io/library/docker101tutorial     0.0s

Use 'docker scan' to run Snyk tests against images to find vul
nerabilities and learn how to fix them
PS C:\Users\Dell\getting-started>
```



Windows Defender Firewall has blocked some features of this app

Windows Defender Firewall has blocked some features of Docker Desktop Backend on all public and private networks.

Name: Docker Desktop Backend
Publisher: Docker Inc.
Path: C:\program files\docker\docker\resources \com.docker.backend.exe

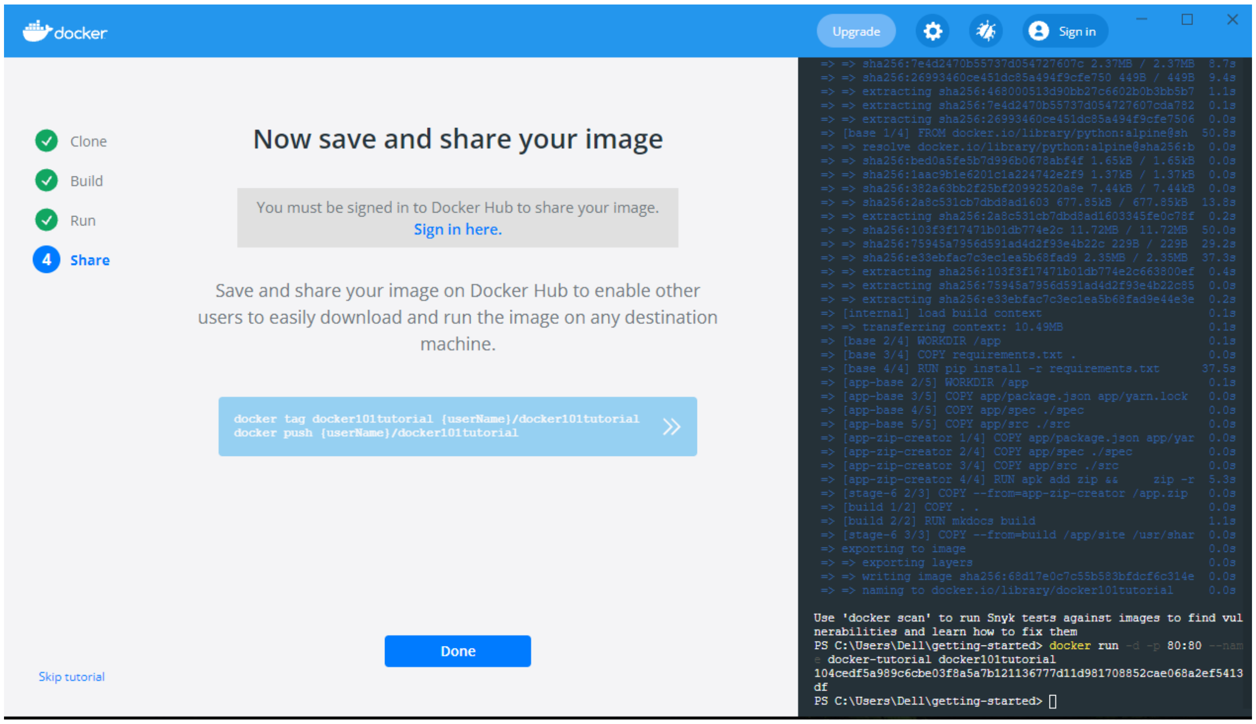
Allow Docker Desktop Backend to communicate on these networks:

- [✓] Private networks, such as my home or work network
- [ ] Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

What are the risks of allowing an app through a firewall?

```
 => => writing image sha256:68d17e0c7c55b583bfdcf6c314e  0.0s
 => => naming to docker.io/library/docker101tutorial     0.0s

Use 'docker scan' to run Snyk tests against images to find vul
nerabilities and learn how to fix them
PS C:\Users\Dell\getting-started> docker run -d -p 80:80 --nam
e docker-tutorial docker101tutorial
104cedf5a989c6cbe03f8a5a7b121136777d11d981708852cae068a2ef5413
df
PS C:\Users\Dell\getting-started>
```

Step 04: Save and share container



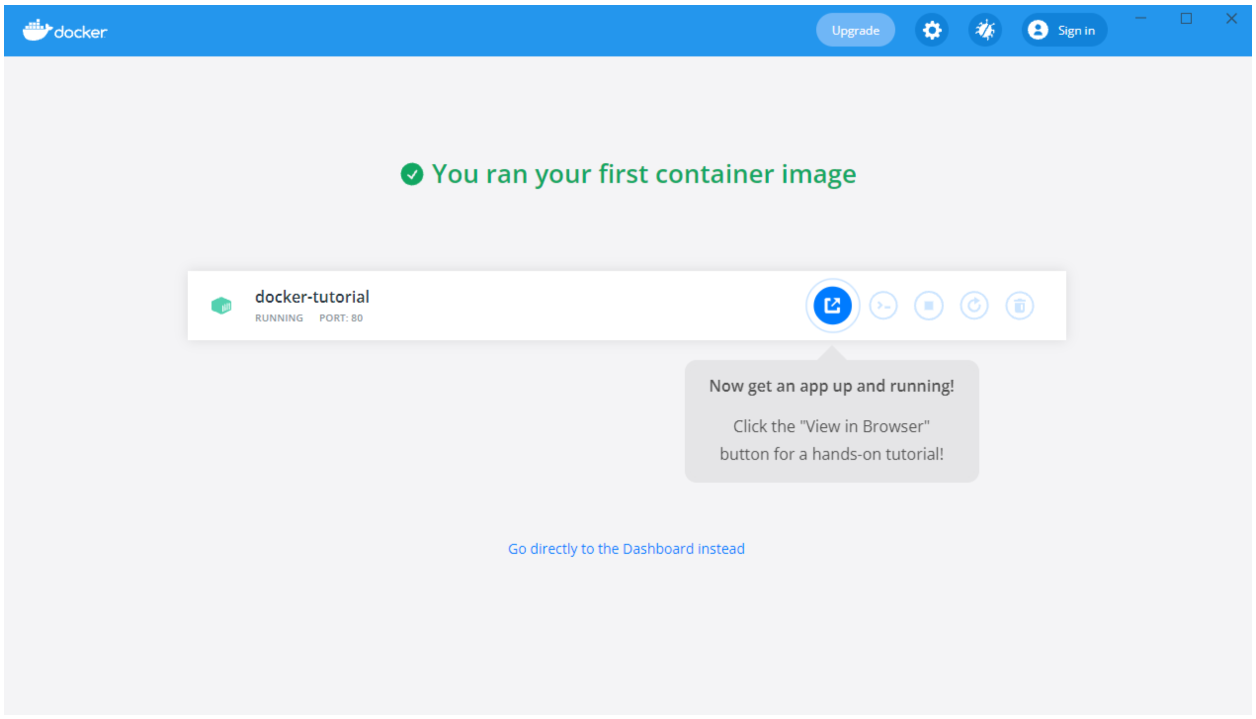
Now save and share your image

You must be signed in to Docker Hub to share your image.
Sign in here.

Save and share your image on Docker Hub to enable other users to easily download and run the image on any destination machine.

```
docker tag docker101tutorial {userName}/docker101tutorial
docker push {userName}/docker101tutorial
```

```
 => => sha256:7e4d2470b55737d054727607c 2.37MB / 2.37MB  8.7s
 => => sha256:26993460ce451dc85a494f9cfe750 449B / 449B  9.4s
 => => extracting sha256:468000513d90bb27c6602b0b3bb5b7  1.1s
 => => extracting sha256:7e4d2470b55737d054727607cda782  0.1s
 => => extracting sha256:26993460ce451dc85a494f9cfe7506  0.0s
 => [base 1/4] FROM docker.io/library/python:alpine@sh  50.8s
 => => resolve docker.io/library/python:alpine@sha256:b  0.0s
 => => sha256:bed0a5fe5b7d996b0678abf4f 1.65kB / 1.65kB  0.0s
 => => sha256:1aac9b1e6201c1a224742e2f9 1.37kB / 1.37kB  0.0s
 => => sha256:382a63bb2f25bf20992520a8e 7.44kB / 7.44kB  0.0s
 => => sha256:2a8c531cb7dbd8ad1603 677.85kB / 677.85kB  13.8s
 => => extracting sha256:2a8c531cb7dbd8ad1603345fe0c78f  0.2s
 => => sha256:103f3f17471b01db774e2c 11.72MB / 11.72MB  50.0s
 => => sha256:75945a7956d591ad4d2f93e4b22c 229B / 229B  29.2s
 => => sha256:e33ebfac7c3ec1ea5b68fad9 2.35MB / 2.35MB  37.3s
 => => extracting sha256:103f3f17471b01db774e2c663800ef  0.4s
 => => extracting sha256:75945a7956d591ad4d2f93e4b22c85  0.0s
 => => extracting sha256:e33ebfac7c3ec1ea5b68fad9e44e3e  0.2s
 => [internal] load build context                        0.1s
 => => transferring context: 10.49MB                     0.1s
 => [base 2/4] WORKDIR /app                              0.1s
 => [base 3/4] COPY requirements.txt .                   0.0s
 => [base 4/4] RUN pip install -r requirements.txt      37.5s
 => [app-base 2/5] WORKDIR /app                          0.1s
 => [app-base 3/5] COPY app/package.json app/yarn.lock   0.0s
 => [app-base 4/5] COPY app/spec ./spec                  0.0s
 => [app-base 5/5] COPY app/src ./src                    0.0s
 => [app-zip-creator 1/4] COPY app/package.json app/yar  0.0s
 => [app-zip-creator 2/4] COPY app/spec ./spec           0.0s
 => [app-zip-creator 3/4] COPY app/src ./src             0.0s
 => [app-zip-creator 4/4] RUN apk add zip &&      zip -r  5.3s
 => [stage-6 2/3] COPY --from=app-zip-creator /app.zip   0.0s
 => [build 1/2] COPY . .                                 0.0s
 => [build 2/2] RUN mkdocs build                         1.1s
 => [stage-6 3/3] COPY --from=build /app/site /usr/shar  0.0s
 => exporting to image                                   0.0s
 => => exporting layers                                  0.0s
 => => writing image sha256:68d17e0c7c55b583bfdcf6c314e  0.0s
 => => naming to docker.io/library/docker101tutorial     0.0s

Use 'docker scan' to run Snyk tests against images to find vul
nerabilities and learn how to fix them
PS C:\Users\Dell\getting-started> docker run -d -p 80:80 --nam
e docker-tutorial docker101tutorial
104cedf5a989c6cbe03f8a5a7b121136777d11d981708852cae068a2ef5413
df
PS C:\Users\Dell\getting-started> 
```



You ran your first container image

Now get an app up and running!

Click the "View in Browser" button for a hands-on tutorial!

Go directly to the Dashboard instead

Let’s familier with more docker commands.

Retrive images with Docker Pull

```
docker pull ubuntu:16.04.
```

Deploy images with Docker run

```
docker run docker/hello-world
```

Grant User access to Docker container

```
sudo usermod -G docker name
```

Retrieve images from repository

```
docker pull ubuntu:latest
```

Stop container instance

```
docker stop (your container ID)
```

Create image using Dokerfile

```
docker build -t testdocker dockerfile
```

List locally stored images

```
docker images
```

Remove container

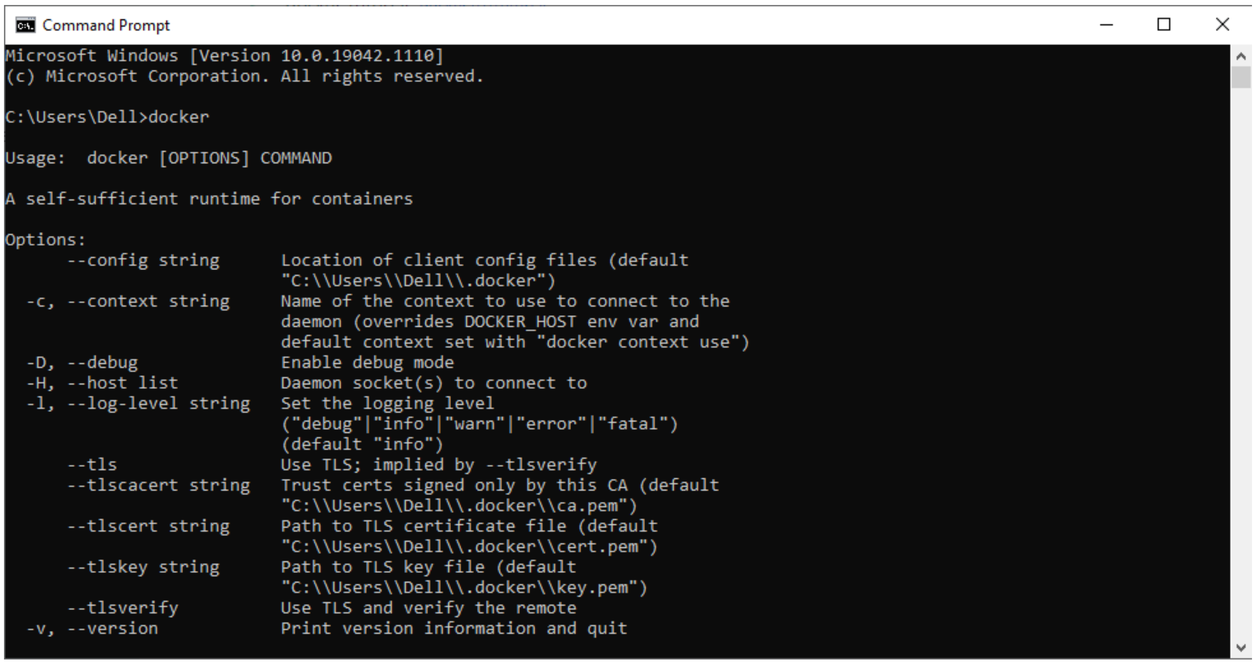
```
docker rm containerID
```

Login to Docker Hub

```
docker login
```

You can find more help form running “docker” in command prompt.

```
Docker
```

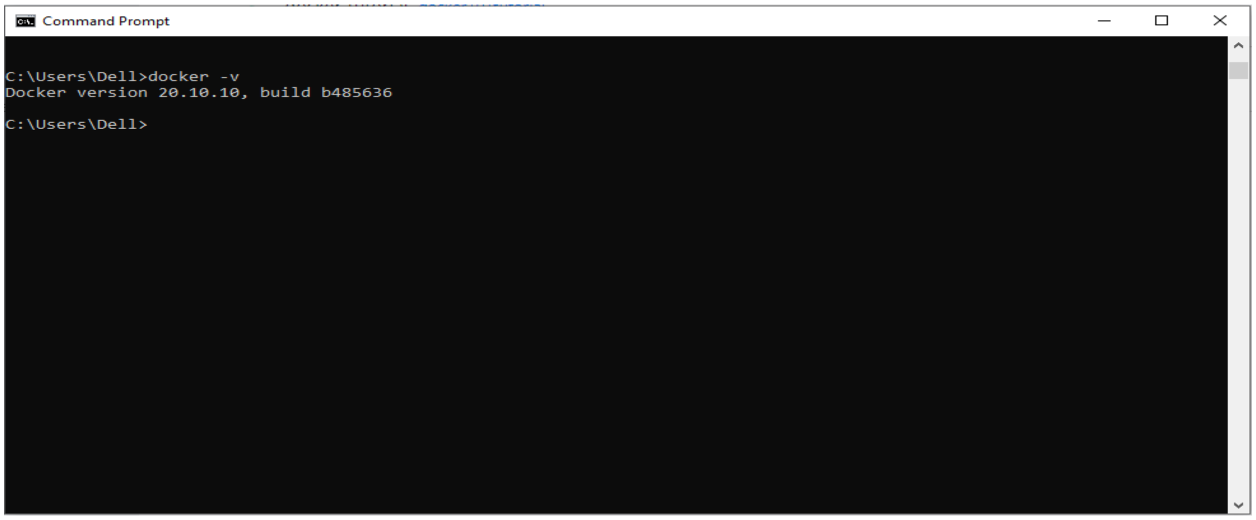


Command Prompt

```
Microsoft Windows [Version 10.0.19042.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dell>docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default
                           "C:\\Users\\Dell\\.docker")
  -c, --context string     Name of the context to use to connect to the
                           daemon (overrides DOCKER_HOST env var and
                           default context set with "docker context use")
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket(s) to connect to
  -l, --log-level string   Set the logging level
                           ("debug"|"info"|"warn"|"error"|"fatal")
                           (default "info")
      --tls                Use TLS; implied by --tlsverify
      --tlscacert string   Trust certs signed only by this CA (default
                           "C:\\Users\\Dell\\.docker\\ca.pem")
      --tlscert string     Path to TLS certificate file (default
                           "C:\\Users\\Dell\\.docker\\cert.pem")
      --tlskey string      Path to TLS key file (default
                           "C:\\Users\\Dell\\.docker\\key.pem")
      --tlsverify          Use TLS and verify the remote
  -v, --version            Print version information and quit
```

Find installed docker version

```
Docker -V
```

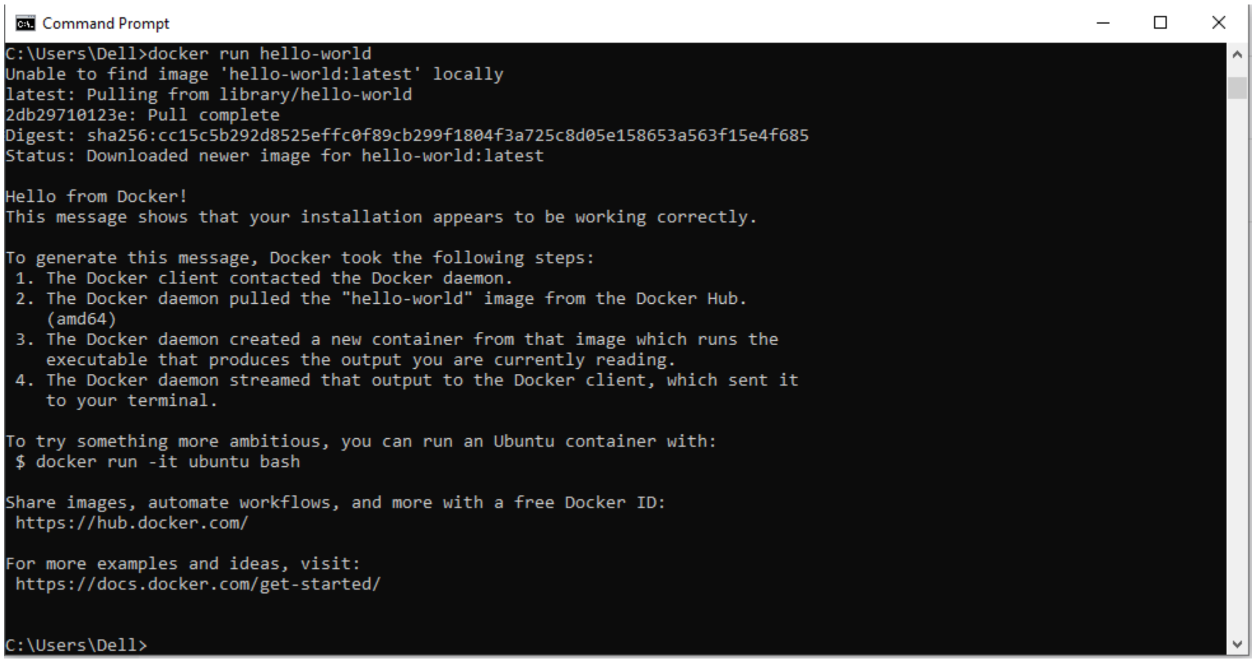


Command Prompt

```
C:\Users\Dell>docker -v
Docker version 20.10.10, build b485636

C:\Users\Dell>
```

We can run docker images using below command. If docker image is not available locally it will pull form the docker image.

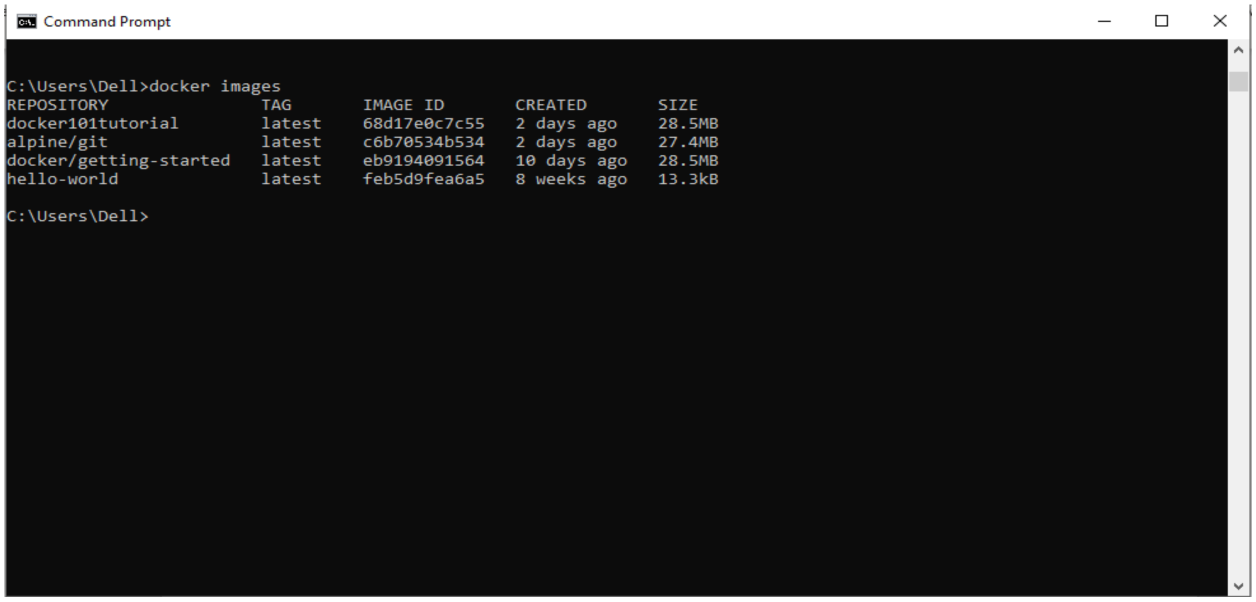
```
Docker run hello-world
```



```
C:\Users\Dell>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:cc15c5b292d8525effc0f89cb299f1804f3a725c8d05e158653a563f15e4f685
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/


C:\Users\Dell>
```

Find docker images available to use.

```
Docker -images
```



```
C:\Users\Dell>docker images
REPOSITORY               TAG       IMAGE ID       CREATED       SIZE
docker101tutorial        latest    68d17e0c7c55   2 days ago    28.5MB
alpine/git               latest    c6b70534b534   2 days ago    27.4MB
docker/getting-started   latest    eb9194091564   10 days ago   28.5MB
hello-world              latest    feb5d9fea6a5   8 weeks ago   13.3kB

C:\Users\Dell>
```

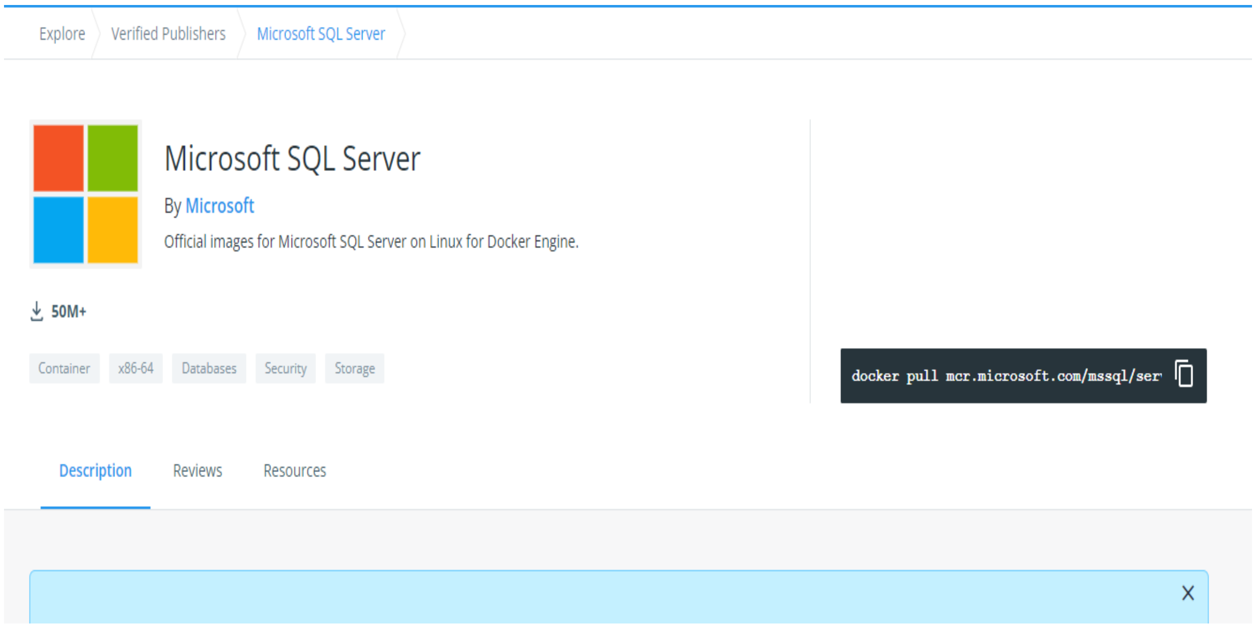
Pulling MS SQL Server image from docker hub

Use below command to download docker image locally

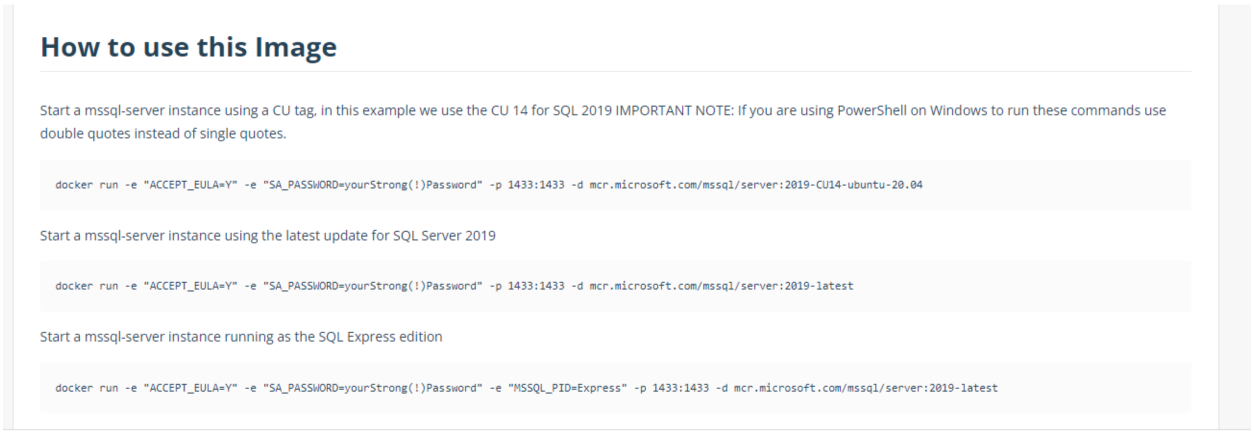
```
docker pull mcr.microsoft.com/mssql/server
```

More Information:

https://hub.docker.com/_/microsoft-mssql-server\



More command available in the same page.



How to use this Image

Start a mssql-server instance using a CU tag, in this example we use the CU 14 for SQL 2019 IMPORTANT NOTE: If you are using PowerShell on Windows to run these commands use double quotes instead of single quotes.

```
docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=yourStrong(!)Password" -p 1433:1433 -d mcr.microsoft.com/mssql/server:2019-CU14-ubuntu-20.04
```

Start a mssql-server instance using the latest update for SQL Server 2019

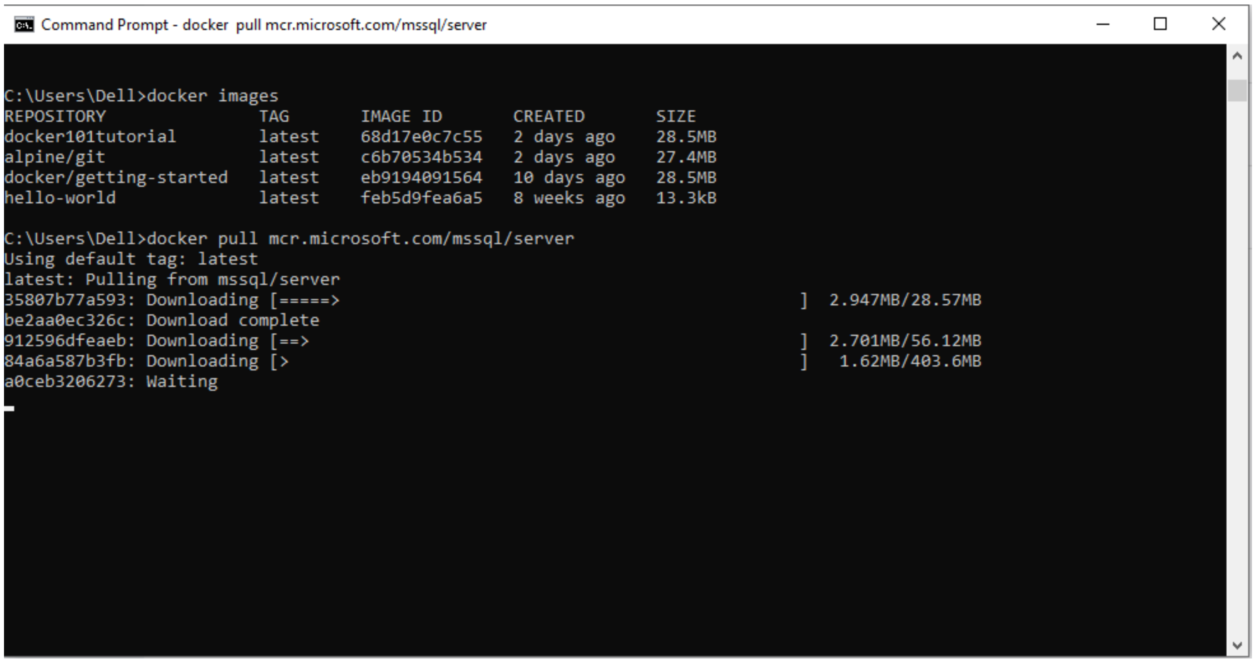
```
docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=yourStrong(!)Password" -p 1433:1433 -d mcr.microsoft.com/mssql/server:2019-latest
```

Start a mssql-server instance running as the SQL Express edition

```
docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=yourStrong(!)Password" -e "MSSQL_PID=Express" -p 1433:1433 -d mcr.microsoft.com/mssql/server:2019-latest
```

Step 01:

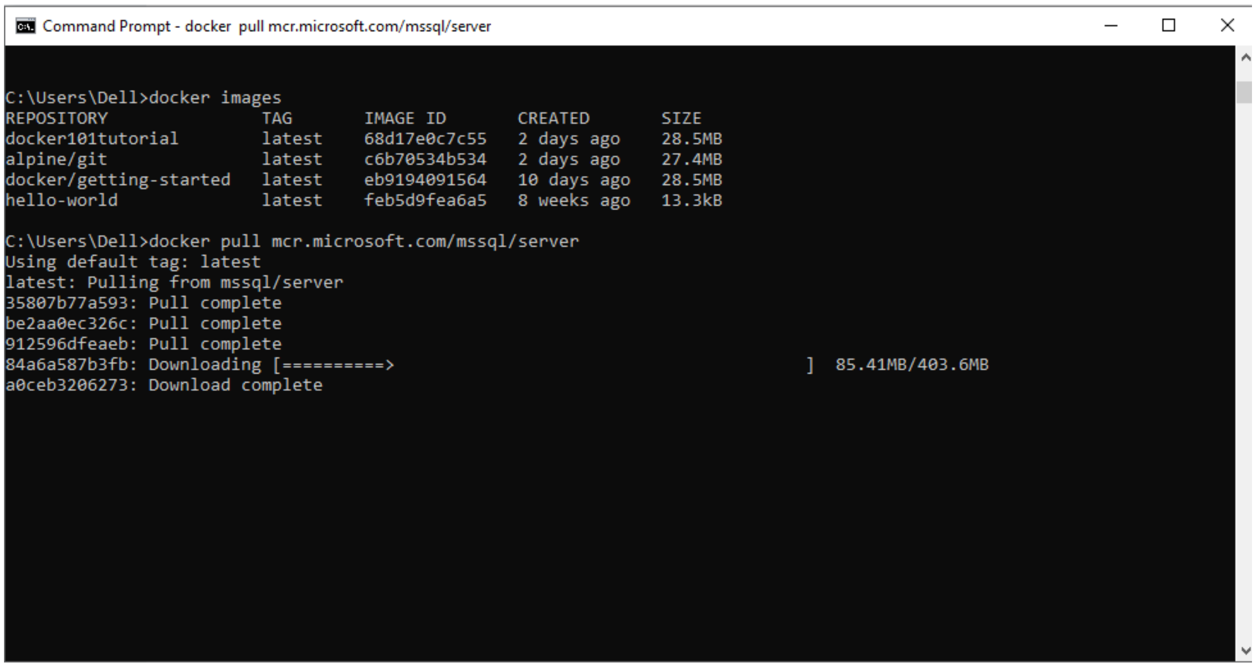
```
docker pull mcr.microsoft.com/mssql/server
```



Command Prompt - docker pull mcr.microsoft.com/mssql/server

```
C:\Users\Dell>docker images
REPOSITORY               TAG       IMAGE ID       CREATED        SIZE
docker101tutorial        latest    68d17e0c7c55   2 days ago     28.5MB
alpine/git               latest    c6b70534b534   2 days ago     27.4MB
docker/getting-started   latest    eb9194091564   10 days ago    28.5MB
hello-world              latest    feb5d9fea6a5   8 weeks ago    13.3kB

C:\Users\Dell>docker pull mcr.microsoft.com/mssql/server
Using default tag: latest
latest: Pulling from mssql/server
35807b77a593: Downloading [=====>                                             ]  2.947MB/28.57MB
be2aa0ec326c: Download complete
912596dfeaeb: Downloading [==>                                                ]  2.701MB/56.12MB
84a6a587b3fb: Downloading [>                                                  ]   1.62MB/403.6MB
a0ceb3206273: Waiting
```

Deployment In progress.



Command Prompt - docker pull mcr.microsoft.com/mssql/server

```
C:\Users\Dell>docker images
REPOSITORY               TAG       IMAGE ID       CREATED        SIZE
docker101tutorial        latest    68d17e0c7c55   2 days ago     28.5MB
alpine/git               latest    c6b70534b534   2 days ago     27.4MB
docker/getting-started   latest    eb9194091564   10 days ago    28.5MB
hello-world              latest    feb5d9fea6a5   8 weeks ago    13.3kB

C:\Users\Dell>docker pull mcr.microsoft.com/mssql/server
Using default tag: latest
latest: Pulling from mssql/server
35807b77a593: Pull complete
be2aa0ec326c: Pull complete
912596dfeaeb: Pull complete
84a6a587b3fb: Downloading [==========>                                        ]  85.41MB/403.6MB
a0ceb3206273: Download complete
```

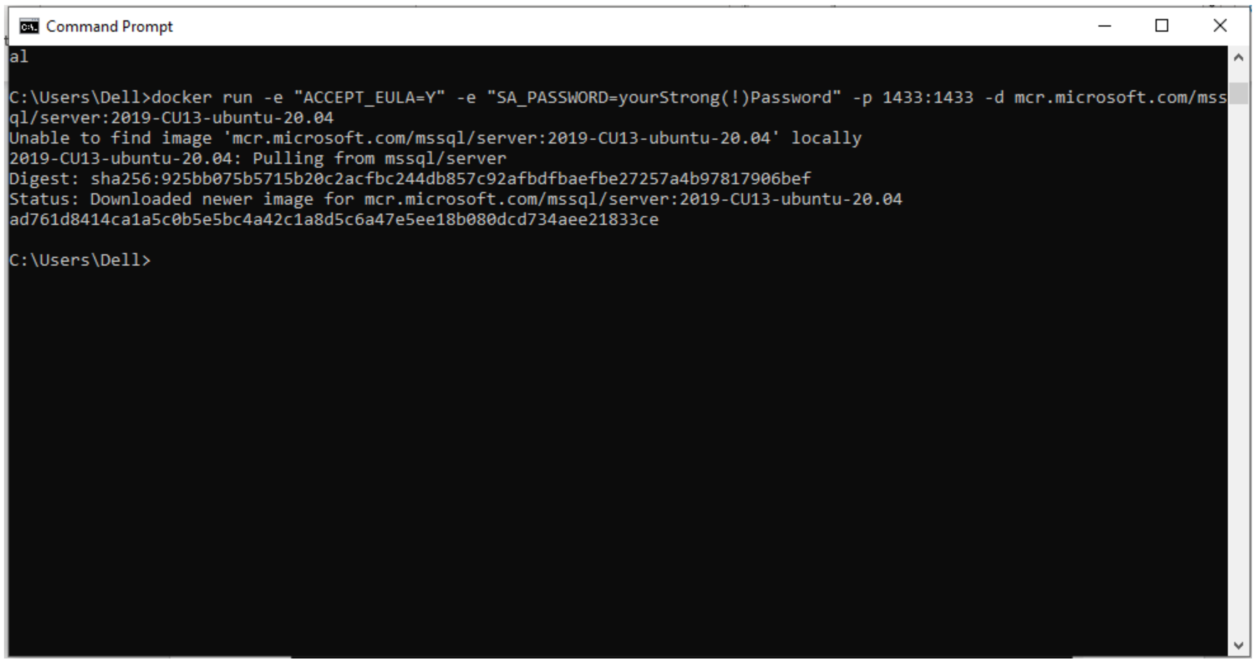
Start MS SQL server instance.

```
docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=yourStrong(!)Password" -p 1433:1433
-d mcr.microsoft.com/mssql/server:2019-CU14-ubuntu-20.04
```

SA_PASSWORD = sa password for MSSQL instance.

-p -port running on MSSQL instance

-d docker image

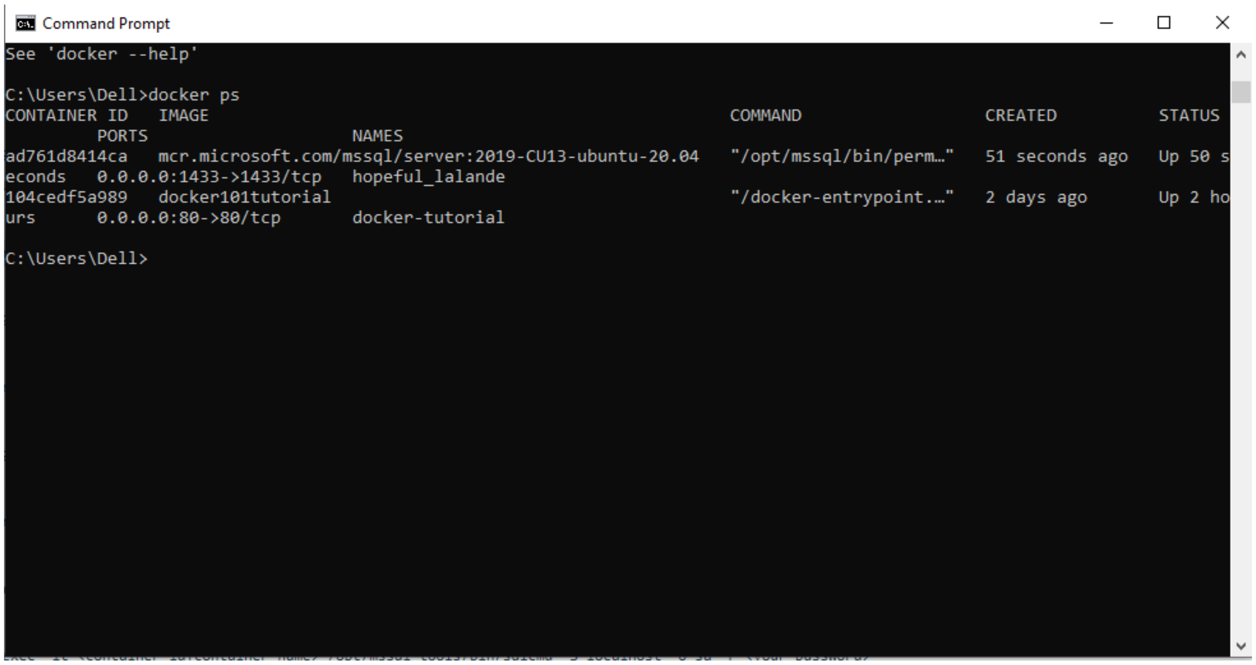


```
al

C:\Users\Dell>docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=yourStrong(!)Password" -p 1433:1433 -d mcr.microsoft.com/mss
ql/server:2019-CU13-ubuntu-20.04
Unable to find image 'mcr.microsoft.com/mssql/server:2019-CU13-ubuntu-20.04' locally
2019-CU13-ubuntu-20.04: Pulling from mssql/server
Digest: sha256:925bb075b5715b20c2acfbc244db857c92afbdfbaefbe27257a4b97817906bef
Status: Downloaded newer image for mcr.microsoft.com/mssql/server:2019-CU13-ubuntu-20.04
ad761d8414ca1a5c0b5e5bc4a42c1a8d5c6a47e5ee18b080dcd734aee21833ce

C:\Users\Dell>
```

Similarly, we can run below command to deploy MSSQLL server express edition.

```
docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=yourStrong(!)Password" -e "MSSQL_PID=Express" -
p 1433:1433 -d mcr.microsoft.com/mssql/server:2019-latest
```

“Docker -PS” provide list of images available to use.



```
See 'docker --help'

C:\Users\Dell>docker ps
CONTAINER ID   IMAGE                                                 COMMAND                  CREATED          STATUS
         PORTS                    NAMES
ad761d8414ca   mcr.microsoft.com/mssql/server:2019-CU13-ubuntu-20.04   "/opt/mssql/bin/perm…"   51 seconds ago   Up 50 s
econds   0.0.0.0:1433->1433/tcp   hopeful_lalande
104cedf5a989   docker101tutorial                                     "/docker-entrypoint.…"   2 days ago       Up 2 ho
urs      0.0.0.0:80->80/tcp       docker-tutorial

C:\Users\Dell>
```

Now we have completed the configuration on the image and MSQL instance. Next let's try to connect to The database management system using Azure data studio.

Fill the below details in new connection Window.

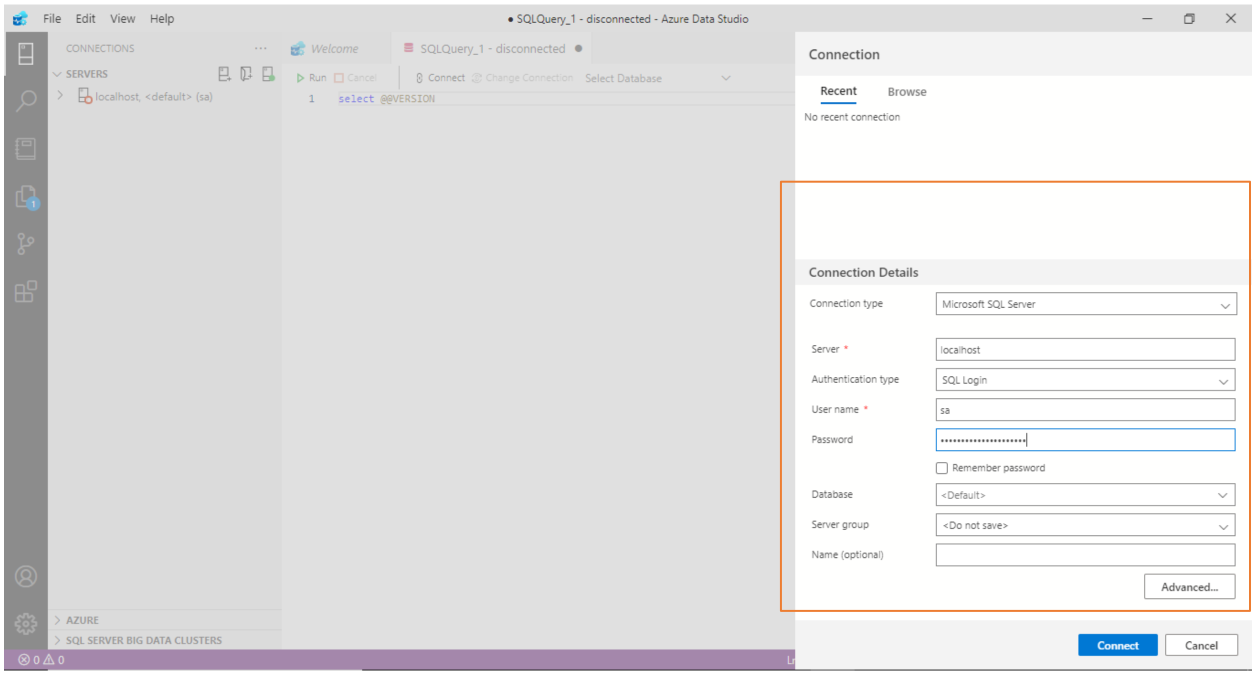
Connection Type: Microsoft SQL Server

Server: localhost

Authentication Type: SQL Login

Username: sa

Password: Your Password


File Edit View Help
• SQLQuery_1 - disconnected - Azure Data Studio
CONNECTIONS
SERVERS
localhost, <default> (sa)
Welcome
SQLQuery_1 - disconnected
Run Cancel Connect Change Connection Select Database
AZURE
SQL SERVER BIG DATA CLUSTERS

```
select @@VERSION
```

Connection

Recent Browse

No recent connection

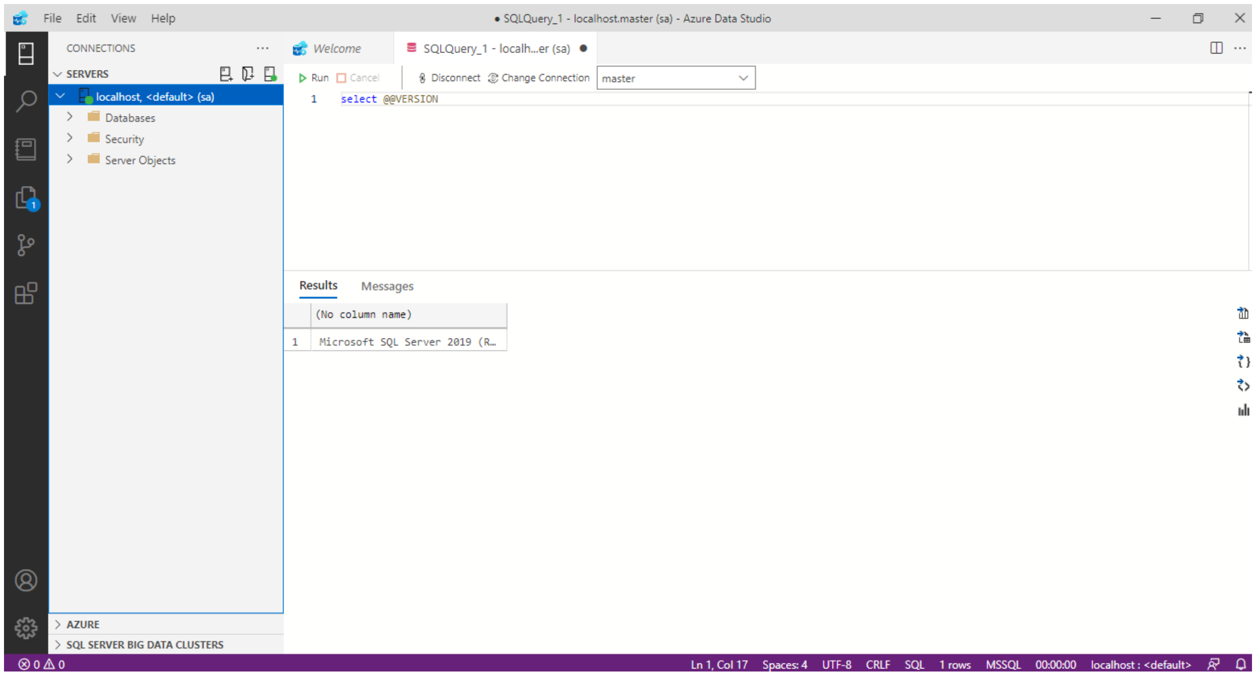
Connection Details

Connection type	Microsoft SQL Server
Server *	localhost
Authentication type	SQL Login
User name *	sa
Password	••••••••••••••••••••••
	☐ Remember password
Database	<Default>
Server group	<Do not save>
Name (optional)	

Advanced...
Connect
Cancel

Run below command to verify deployed MSQL Server Edition.

```
selct @@version
```


File Edit View Help
• SQLQuery_1 - localhost.master (sa) - Azure Data Studio
CONNECTIONS
SERVERS
localhost, <default> (sa)
Databases
Security
Server Objects
Welcome
SQLQuery_1 - localh...er (sa)
Run Cancel Disconnect Change Connection master
Results Messages
AZURE
SQL SERVER BIG DATA CLUSTERS
Ln 1, Col 17 Spaces: 4 UTF-8 CRLF SQL 1 rows MSSQL 00:00:00 localhost : <default>

```
select @@VERSION
```

```
   (No column name)
1  Microsoft SQL Server 2019 (R…
```

Section 2 MSSQL Server deployment on Kubernetes (high availability)

In this section, Azure Kubernetes Basics, and installation MSSQL on Azure Kubernetes Services will be covered.

Section 02

What is Kubernetes?

Kubernetes is an open-source platform where it can provide portability and extensibility for containerized environments. This also known as container orchestration platform. Kubernetes itself is a service which can provide automate process on most of the deployments and managements. There are situation managing multiple containers can be difficult to manage and coordinate. Therefore, Google has developed solution for this issue.

This platform initially developed by Google. It was announced as open-source platform in 2014.Nowdays, Kubernetes maintained by the cloud native computing foundation. Kubernetes mainly working with Docker. However, it is not limited to Docker.

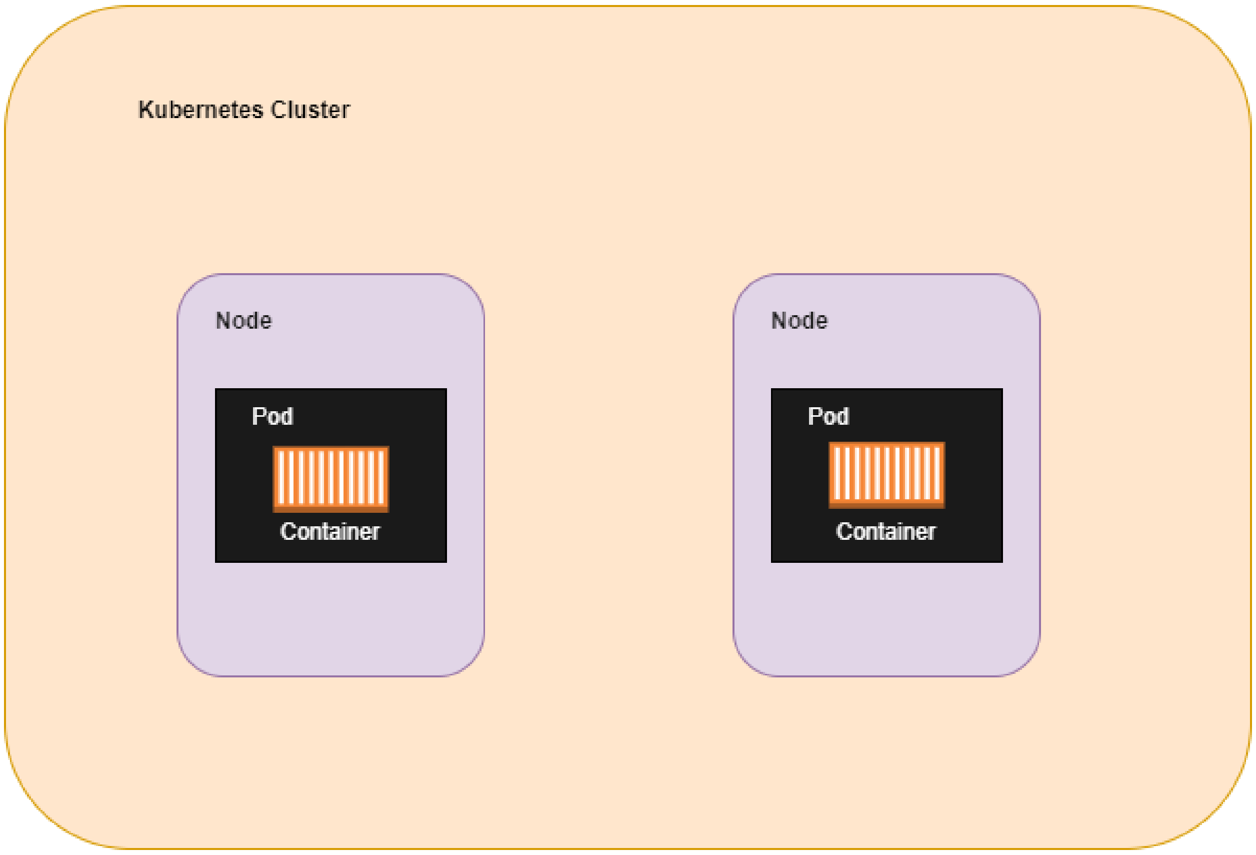
Kubernetes can be deployed on prem servers or workstations. Also, it can be deployed on managed provides sch as Red hat OpenShift and Rancher. Additionally, major cloud provides like Microsoft, Google and Amazon provide Kubernetes as service. In this section, Azure Kubernetes services used for all the examples.

In this section, below sections will be covered.

- Kubernetes basics
- Azure Kubernetes Services,
- Deployment of MSSQL Server on AKS (in 10 steps)
- Test high availability of MSSQL Server on AKS

Components of Kubernetes

Kubernetes Cluster consist of set of worker machines. Every cluster has at least one node. Nodes are more like virtual machines. Nodes are host for the Pods. Each node contains the necessary services to run the Pod. In the Pod, we can have multiple containers. There should be at least one container in the node. Each Pod has unique IP address in the given cluster.



In this section, all the Kubernetes related examples will be on Azure Kubernetes Services. Before start with the MS SQL Server deployment Azure command line need to be configured on you PC.

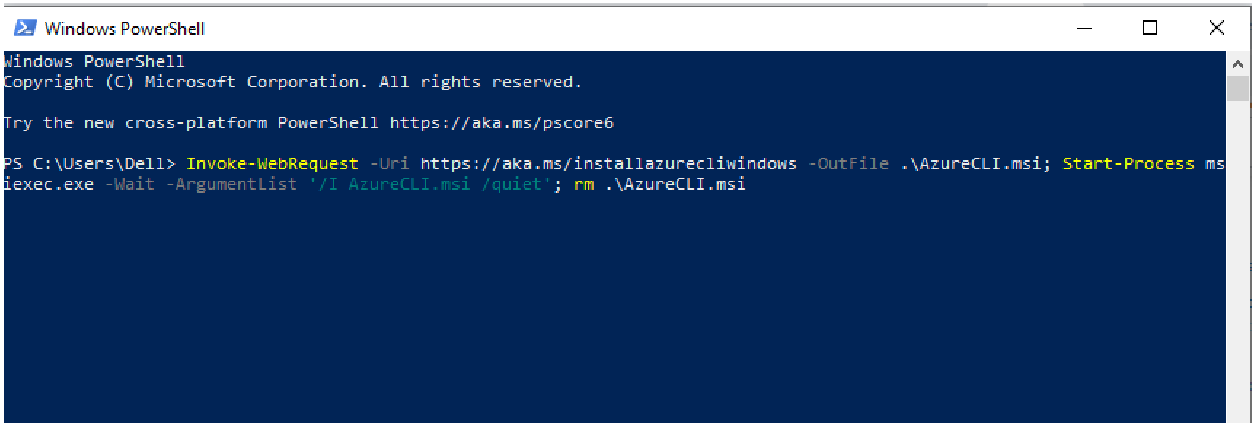
Installing Azure CLI

Azure Command Line Interface is a tool that can be install on local PC. This tool enables us to connect to the azure services.

There are two ways to install Azure CLI.

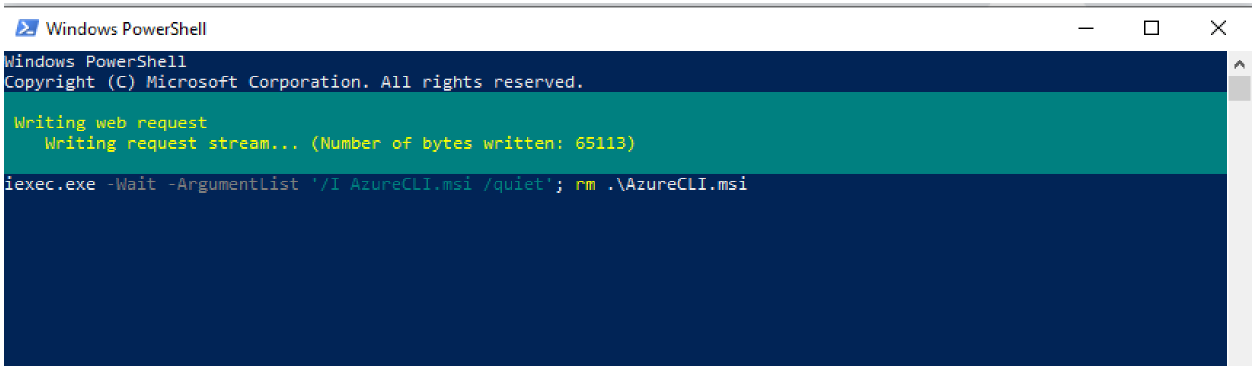
- Using PowerShell.

```
Invoke-WebRequest -Uri https://aka.ms/installazurecliwindows -OutFile
.\AzureCLI.msi; Start-Process msiexec.exe -Wait -ArgumentList '/I AzureCLI.msi
/quiet'; rm .\AzureCLI.msi
```



Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Dell> Invoke-WebRequest -Uri https://aka.ms/installazurecliwindows -OutFile .\AzureCLI.msi; Start-Process ms
iexec.exe -Wait -ArgumentList '/I AzureCLI.msi /quiet'; rm .\AzureCLI.msi
```



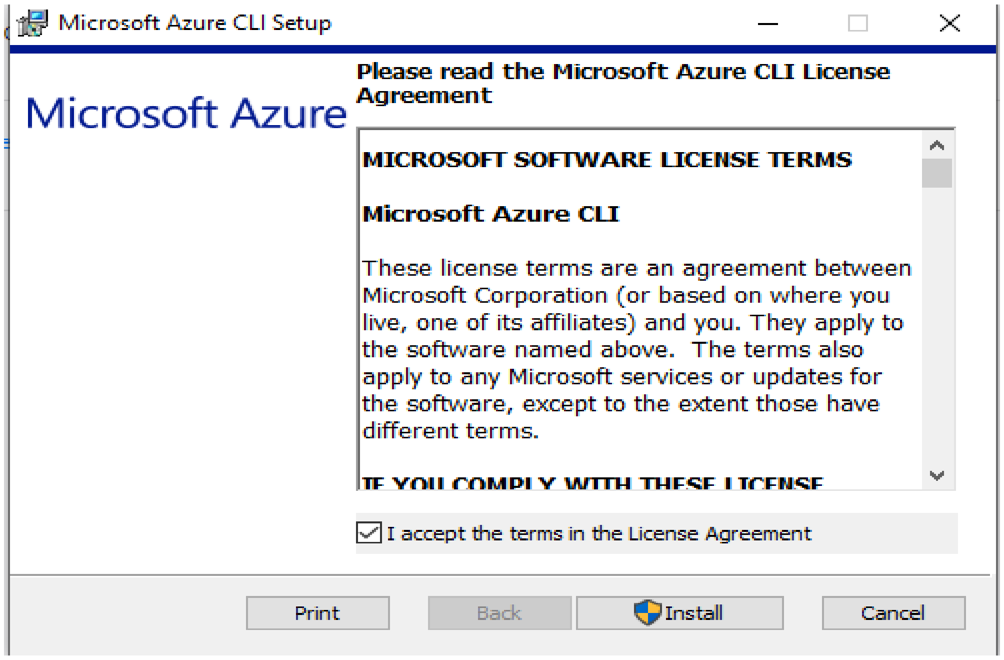
Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

 Writing web request
    Writing request stream... (Number of bytes written: 65113)

iexec.exe -Wait -ArgumentList '/I AzureCLI.msi /quiet'; rm .\AzureCLI.msi
```

- Using MSI Distributable

You can download MSI using below link.

https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows?tabs=azure-cli



Microsoft Azure CLI Setup

Microsoft Azure

Please read the Microsoft Azure CLI License Agreement

MICROSOFT SOFTWARE LICENSE TERMS

Microsoft Azure CLI

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. They apply to the software named above. The terms also apply to any Microsoft services or updates for the software, except to the extent those have different terms.

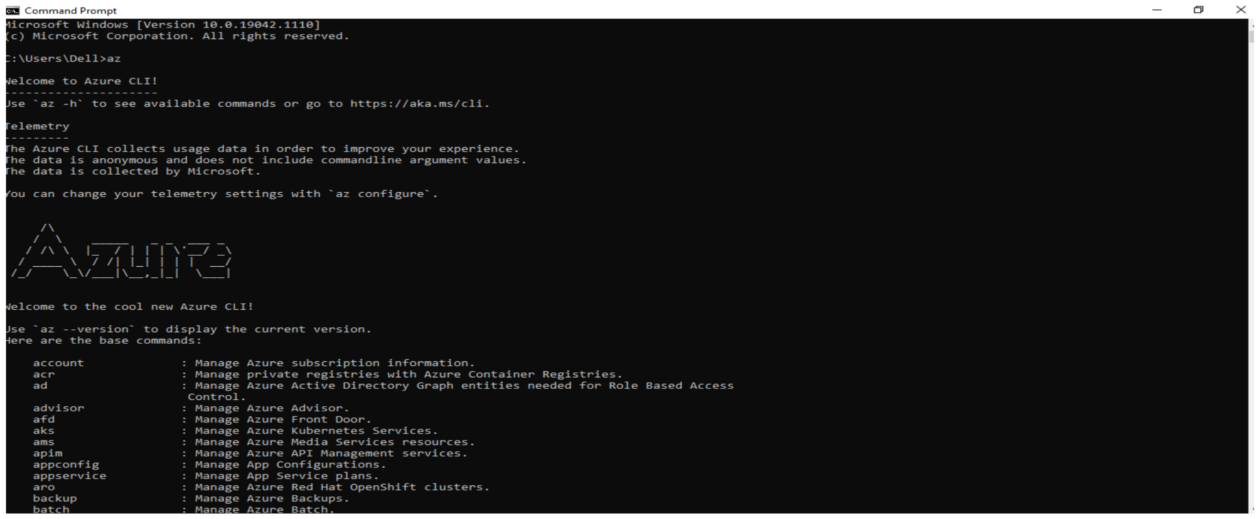
IF YOU COMPLY WITH THESE LICENSE

[✓] I accept the terms in the License Agreement

Print | Back | Install | Cancel

Verify the Azure CLI installation. Run below command in command prompt.

```
AZ
```

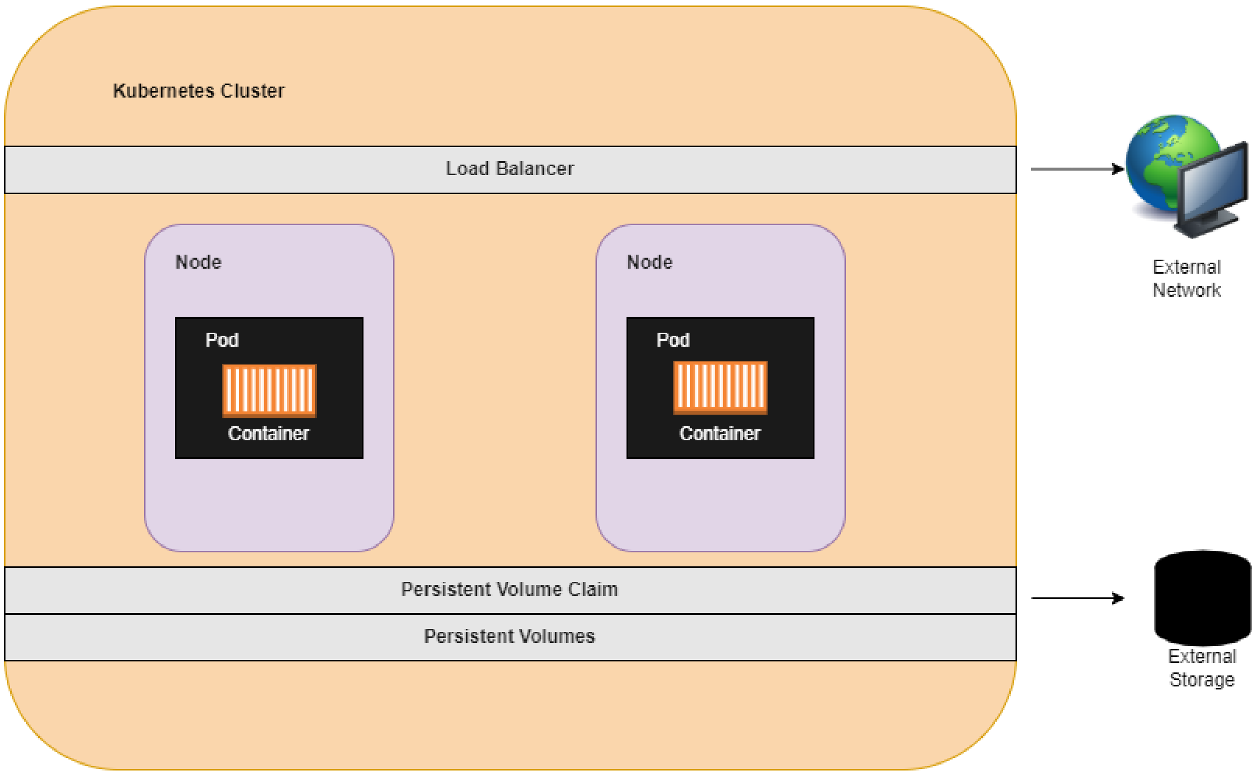


Command Prompt

```
Microsoft Windows [Version 10.0.19042.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dell>az

Welcome to Azure CLI!
---------------------
Use `az -h` to see available commands or go to https://aka.ms/cli.

Telemetry
---------
The Azure CLI collects usage data in order to improve your experience.
The data is anonymous and does not include commandline argument values.
The data is collected by Microsoft.

You can change your telemetry settings with `az configure`.


     /\
    /  \    _____   _ _  ___ _
   / /\ \  |_  / | | | \'__/ _ \
  / ____ \  / /| |_| | | | |  __/
 /_/    \_\/___|\__,_|_|  \___|


Welcome to the cool new Azure CLI!

Use `az --version` to display the current version.
Here are the base commands:

    account                 : Manage Azure subscription information.
    acr                     : Manage private registries with Azure Container Registries.
    ad                      : Manage Azure Active Directory Graph entities needed for Role Based Access
                             Control.
    advisor                 : Manage Azure Advisor.
    afd                     : Manage Azure Front Door.
    aks                     : Manage Azure Kubernetes Services.
    ams                     : Manage Azure Media Services resources.
    apim                    : Manage Azure API Management services.
    appconfig               : Manage App Configurations.
    appservice              : Manage App Service plans.
    aro                     : Manage Azure Red Hat OpenShift clusters.
    backup                  : Manage Azure Backups.
    batch                   : Manage Azure Batch.
```

Installing MS SQL server Azure Kubernetes Service.

Deployment Diagram -proposed example.

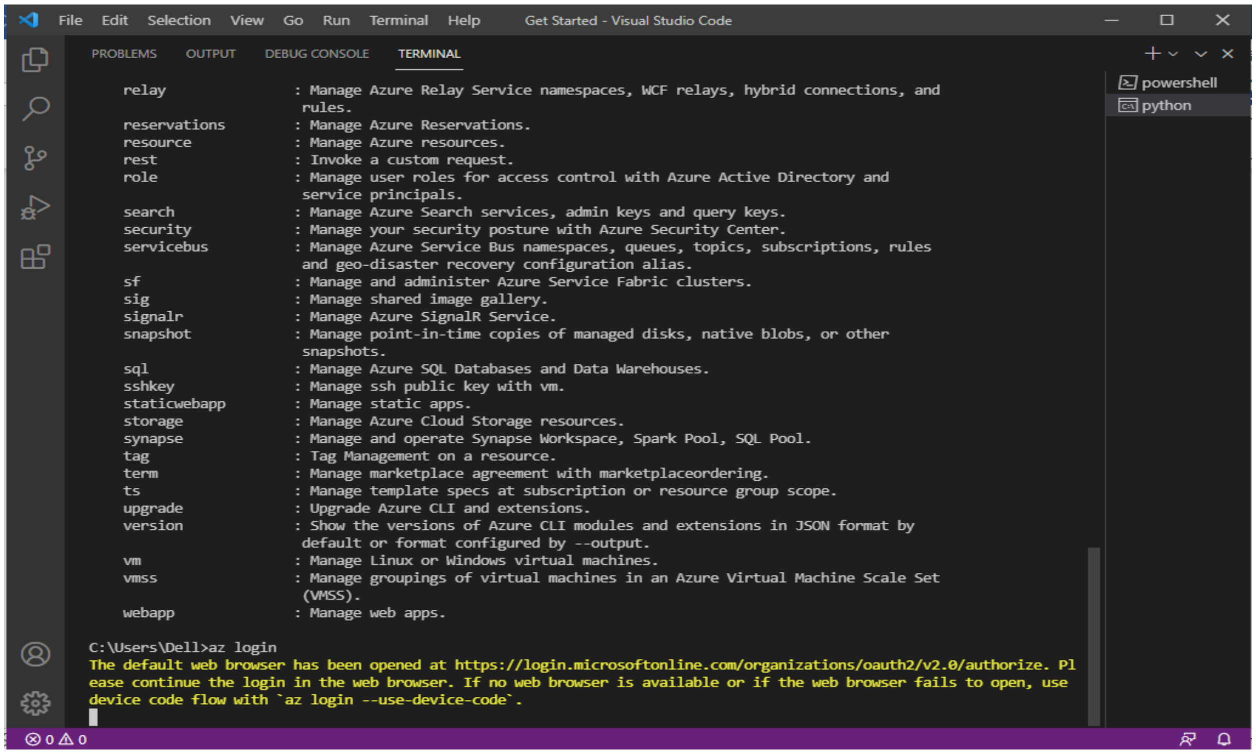


In this deployment, there will be two nodes cluster. Persistent volume claim and persistence volumes are providing high availability to the Pods.

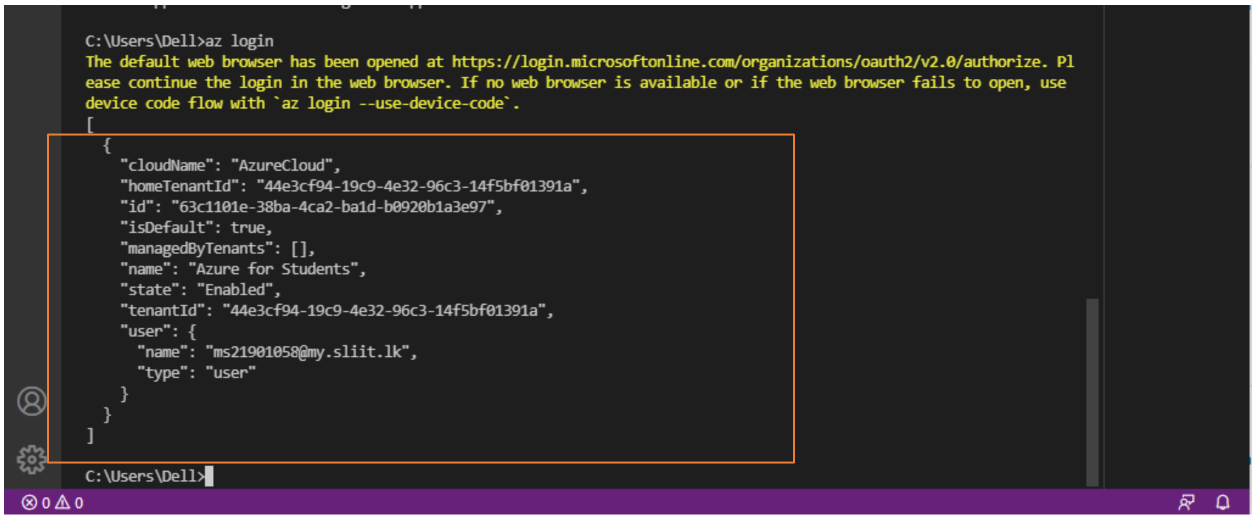
Load Balancer provide outside world access to MSSQL Sever. The external storage provides the database operation of the MSSQL server. Containers are stateless. They are good for the applications. However, external storage required for stateful application like MSSQL Server.

Step 01: Connecting Azure Kubernetes Service using a Microsoft account.

```
AZ Login
```

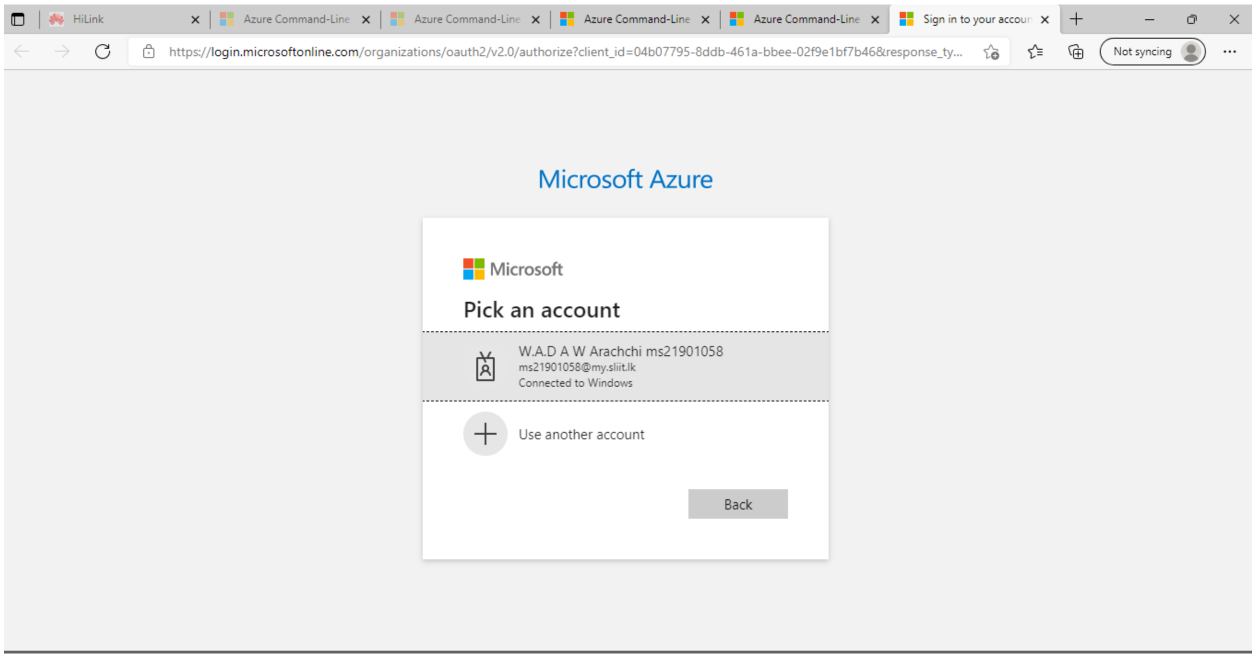


```
    relay                 : Manage Azure Relay Service namespaces, WCF relays, hybrid connections, and
                            rules.
    reservations          : Manage Azure Reservations.
    resource              : Manage Azure resources.
    rest                  : Invoke a custom request.
    role                  : Manage user roles for access control with Azure Active Directory and
                            service principals.
    search                : Manage Azure Search services, admin keys and query keys.
    security              : Manage your security posture with Azure Security Center.
    servicebus            : Manage Azure Service Bus namespaces, queues, topics, subscriptions, rules
                            and geo-disaster recovery configuration alias.
    sf                    : Manage and administer Azure Service Fabric clusters.
    sig                   : Manage shared image gallery.
    signalr               : Manage Azure SignalR Service.
    snapshot              : Manage point-in-time copies of managed disks, native blobs, or other
                            snapshots.
    sql                   : Manage Azure SQL Databases and Data Warehouses.
    sshkey                : Manage ssh public key with vm.
    staticwebapp          : Manage static apps.
    storage               : Manage Azure Cloud Storage resources.
    synapse               : Manage and operate Synapse Workspace, Spark Pool, SQL Pool.
    tag                   : Tag Management on a resource.
    term                  : Manage marketplace agreement with marketplaceordering.
    ts                    : Manage template specs at subscription or resource group scope.
    upgrade               : Upgrade Azure CLI and extensions.
    version               : Show the versions of Azure CLI modules and extensions in JSON format by
                            default or format configured by --output.
    vm                    : Manage Linux or Windows virtual machines.
    vmss                  : Manage groupings of virtual machines in an Azure Virtual Machine Scale Set
                            (VMSS).
    webapp                : Manage web apps.

C:\Users\Dell>az login
The default web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Pl
ease continue the login in the web browser. If no web browser is available or if the web browser fails to open, use
device code flow with `az login --use-device-code`.
```



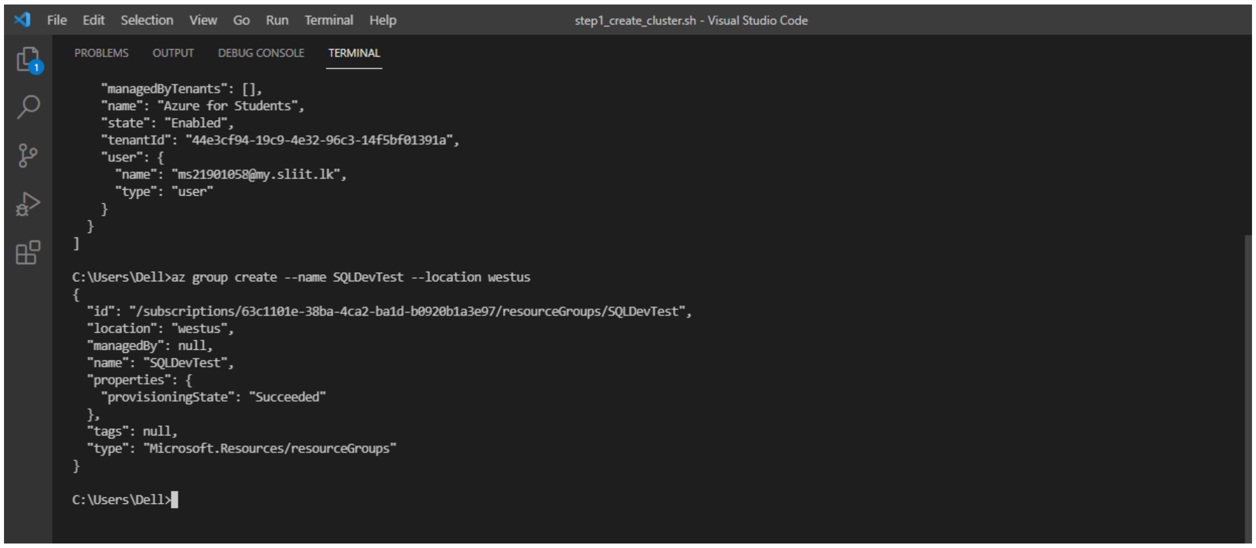
```
C:\Users\Dell>az login
The default web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Pl
ease continue the login in the web browser. If no web browser is available or if the web browser fails to open, use
device code flow with `az login --use-device-code`.
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "44e3cf94-19c9-4e32-96c3-14f5bf01391a",
    "id": "63c1101e-38ba-4ca2-ba1d-b0920b1a3e97",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure for Students",
    "state": "Enabled",
    "tenantId": "44e3cf94-19c9-4e32-96c3-14f5bf01391a",
    "user": {
      "name": "ms21901058@my.sliit.lk",
      "type": "user"
    }
  }
]
C:\Users\Dell>
```

Step 02: Provide Microsoft account portal.


https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize?client_id=04b07795-8ddb-461a-bbee-02f9e1bf7b46&response_ty...
Microsoft Azure
Microsoft
Pick an account
W.A.D A W Arachchi ms21901058
ms21901058@my.sliit.lk
Connected to Windows
Use another account
Back

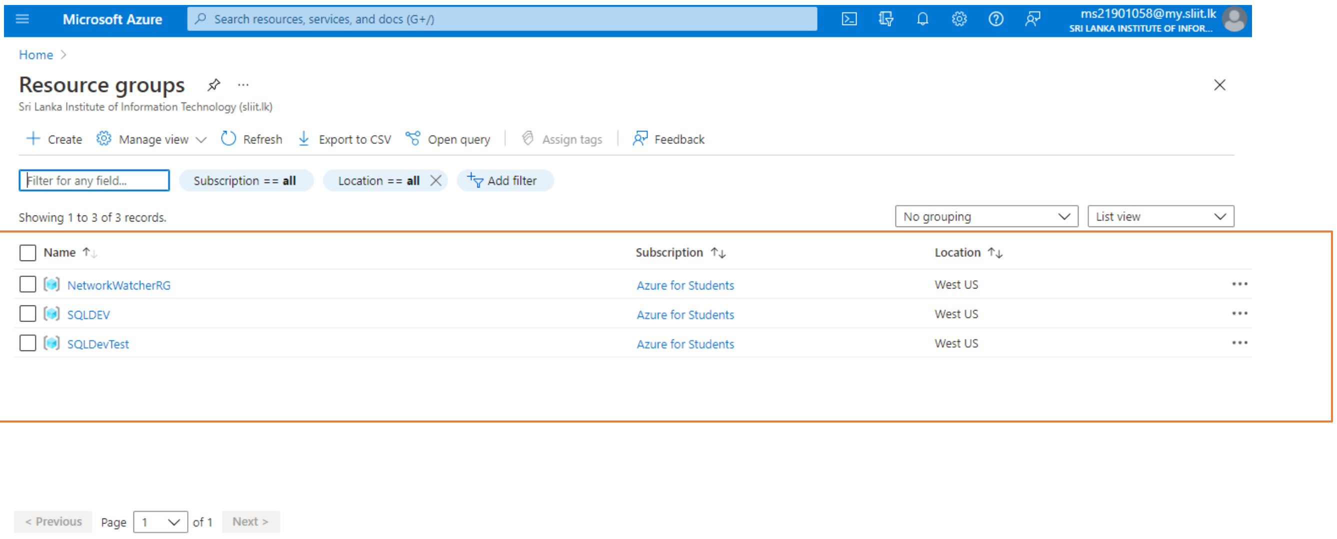
Step 03: Create Resource Group

```
az group create --name SQLDevTest --location westus
```


File Edit Selection View Go Run Terminal Help
step1_create_cluster.sh - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
    "managedByTenants": [],
    "name": "Azure for Students",
    "state": "Enabled",
    "tenantId": "44e3cf94-19c9-4e32-96c3-14f5bf01391a",
    "user": {
      "name": "ms21901058@my.sliit.lk",
      "type": "user"
    }
  }
]

C:\Users\Dell>az group create --name SQLDevTest --location westus
{
  "id": "/subscriptions/63c1101e-38ba-4ca2-ba1d-b0920b1a3e97/resourceGroups/SQLDevTest",
  "location": "westus",
  "managedBy": null,
  "name": "SQLDevTest",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}

C:\Users\Dell>
```

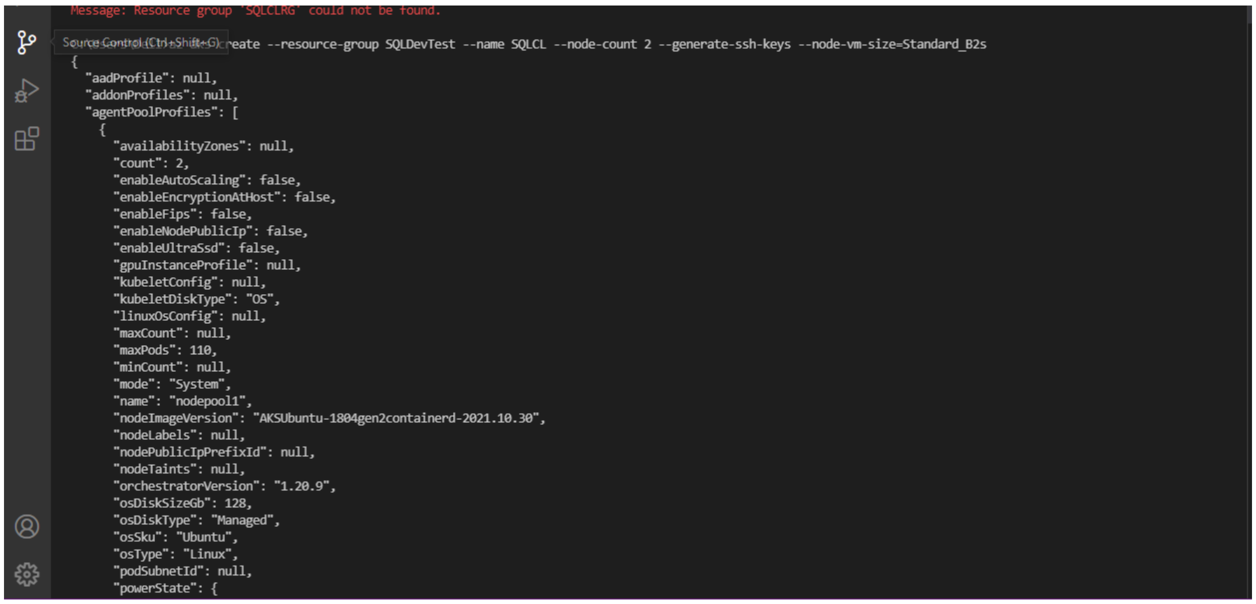
After creating resource group, we can find that in azure portal



Name	Subscription	Location
NetworkWatcherRG	Azure for Students	West US
SQLDEV	Azure for Students	West US
SQLDevTest	Azure for Students	West US

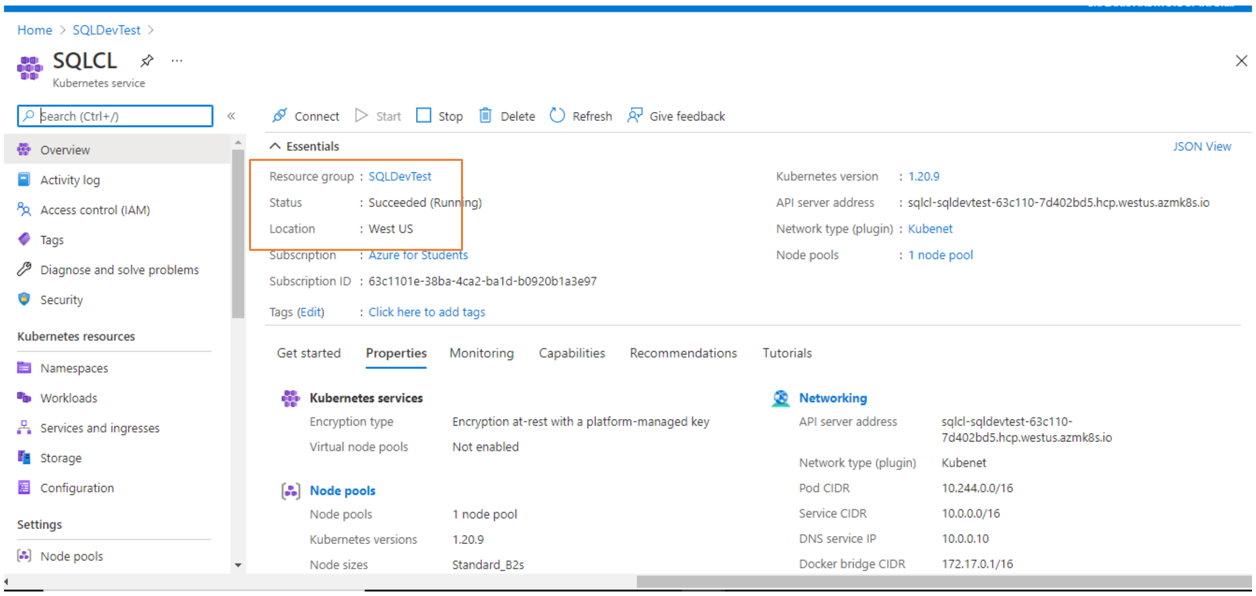
Step 04: Creating Two Node Cluster

```
az aks create --resource-group SQLDevTest --name SQLCL --node-count 2 --generate-
ssh-keys --node-vm-size=Standard_B2s
```



```
Message: Resource group 'SQLCLRG' could not be found.
create --resource-group SQLDevTest --name SQLCL --node-count 2 --generate-ssh-keys --node-vm-size=Standard_B2s
{
  "aadProfile": null,
  "addonProfiles": null,
  "agentPoolProfiles": [
    {
      "availabilityZones": null,
      "count": 2,
      "enableAutoScaling": false,
      "enableEncryptionAtHost": false,
      "enableFips": false,
      "enableNodePublicIp": false,
      "enableUltraSsd": false,
      "gpuInstanceProfile": null,
      "kubeletConfig": null,
      "kubeletDiskType": "OS",
      "linuxOsConfig": null,
      "maxCount": null,
      "maxPods": 110,
      "minCount": null,
      "mode": "System",
      "name": "nodepool1",
      "nodeImageVersion": "AKSUbuntu-1804gen2containerd-2021.10.30",
      "nodeLabels": null,
      "nodePublicIpPrefixId": null,
      "nodeTaints": null,
      "orchestratorVersion": "1.20.9",
      "osDiskSizeGb": 128,
      "osDiskType": "Managed",
      "osSku": "Ubuntu",
      "osType": "Linux",
      "podSubnetId": null,
      "powerState": {
```

You can find Kubernities Services beign created in the Azure Portal.



Resource group : SQLDevTest
Status : Succeeded (Running)
Location : West US
Subscription : Azure for Students
Subscription ID : 63c1101e-38ba-4ca2-ba1d-b0920b1a3e97
Tags (Edit) : Click here to add tags

Kubernetes version : 1.20.9
API server address : sqlcl-sqldevtest-63c110-7d402bd5.hcp.westus.azmk8s.io
Network type (plugin) : Kubenet
Node pools : 1 node pool

Kubernetes services

Encryption type	Encryption at-rest with a platform-managed key
Virtual node pools	Not enabled

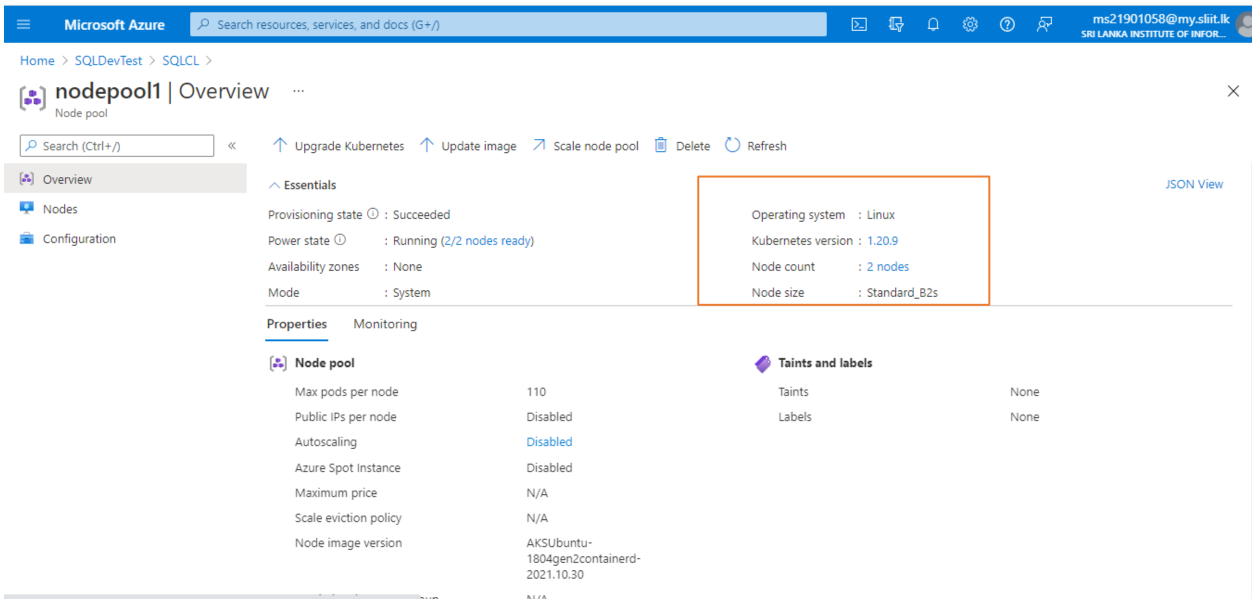
Node pools

Node pools	1 node pool
Kubernetes versions	1.20.9
Node sizes	Standard_B2s

Networking

API server address	sqlcl-sqldevtest-63c110-7d402bd5.hcp.westus.azmk8s.io
Network type (plugin)	Kubenet
Pod CIDR	10.244.0.0/16
Service CIDR	10.0.0.0/16
DNS service IP	10.0.0.10
Docker bridge CIDR	172.17.0.1/16

Here, we have created one node pool.



Provisioning state : Succeeded
Power state : Running (2/2 nodes ready)
Availability zones : None
Mode : System

Operating system : Linux
Kubernetes version : 1.20.9
Node count : 2 nodes
Node size : Standard_B2s

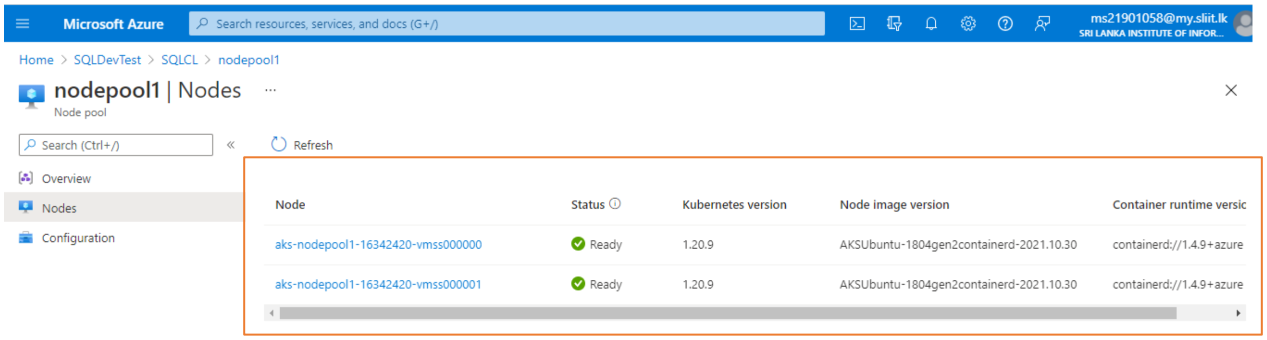
Node pool

Max pods per node	110
Public IPs per node	Disabled
Autoscaling	Disabled
Azure Spot Instance	Disabled
Maximum price	N/A
Scale eviction policy	N/A
Node image version	AKSUbuntu-1804gen2containerd-2021.10.30

Taints and labels

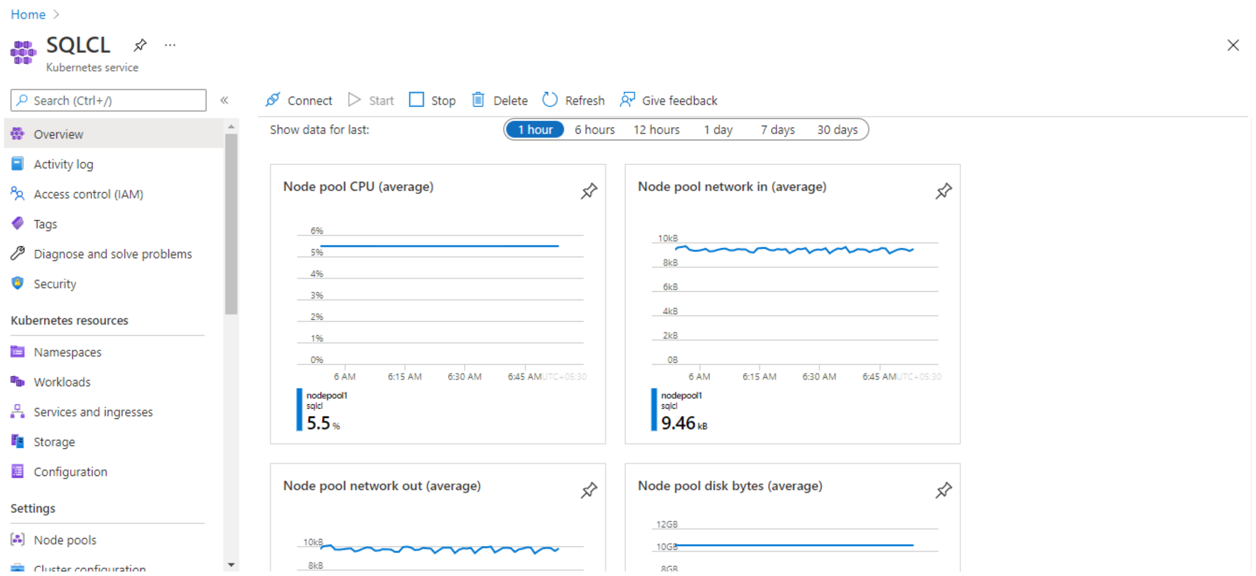
Taints	None
Labels	None

There are two nodes available in the node pool. We have configured two containers.

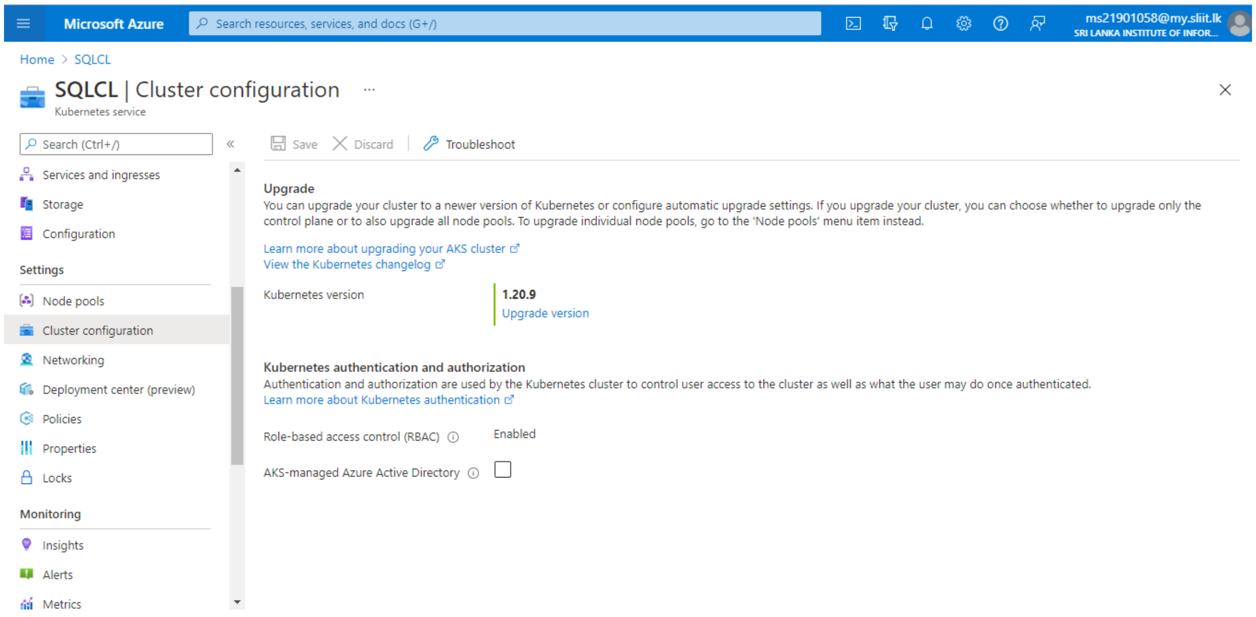


Node	Status	Kubernetes version	Node image version	Container runtime versic
aks-nodepool1-16342420-vmss000000	Ready	1.20.9	AKSUbuntu-1804gen2containerd-2021.10.30	containerd://1.4.9+azure
aks-nodepool1-16342420-vmss000001	Ready	1.20.9	AKSUbuntu-1804gen2containerd-2021.10.30	containerd://1.4.9+azure

In the monitoring window, we can find the resource utilizations of the node pool.



Cluster Details



SQLCL | Cluster configuration

Kubernetes service

Upgrade

You can upgrade your cluster to a newer version of Kubernetes or configure automatic upgrade settings. If you upgrade your cluster, you can choose whether to upgrade only the control plane or to also upgrade all node pools. To upgrade individual node pools, go to the 'Node pools' menu item instead.

Learn more about upgrading your AKS cluster

View the Kubernetes changelog

Kubernetes version	**1.20.9** Upgrade version

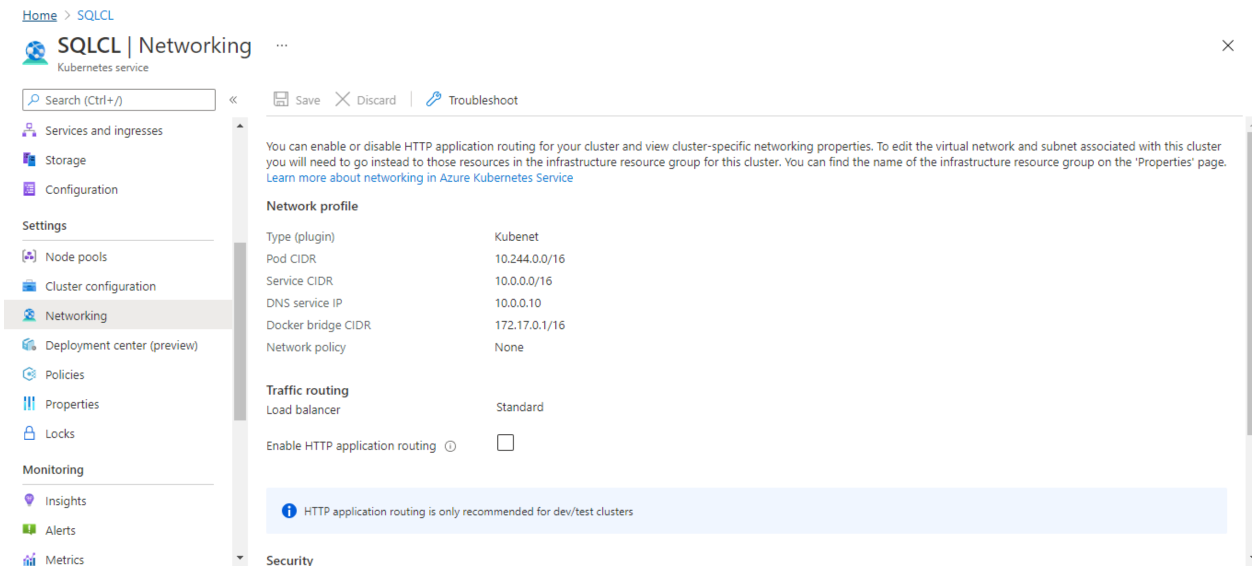
Kubernetes authentication and authorization

Authentication and authorization are used by the Kubernetes cluster to control user access to the cluster as well as what the user may do once authenticated.

Learn more about Kubernetes authentication

Role-based access control (RBAC)	Enabled
AKS-managed Azure Active Directory	☐

Cluster specific networking properties



SQLCL | Networking

Kubernetes service

You can enable or disable HTTP application routing for your cluster and view cluster-specific networking properties. To edit the virtual network and subnet associated with this cluster you will need to go instead to those resources in the infrastructure resource group for this cluster. You can find the name of the infrastructure resource group on the 'Properties' page.

Learn more about networking in Azure Kubernetes Service

Network profile

Type (plugin)	Kubenet
Pod CIDR	10.244.0.0/16
Service CIDR	10.0.0.0/16
DNS service IP	10.0.0.10
Docker bridge CIDR	172.17.0.1/16
Network policy	None

Traffic routing

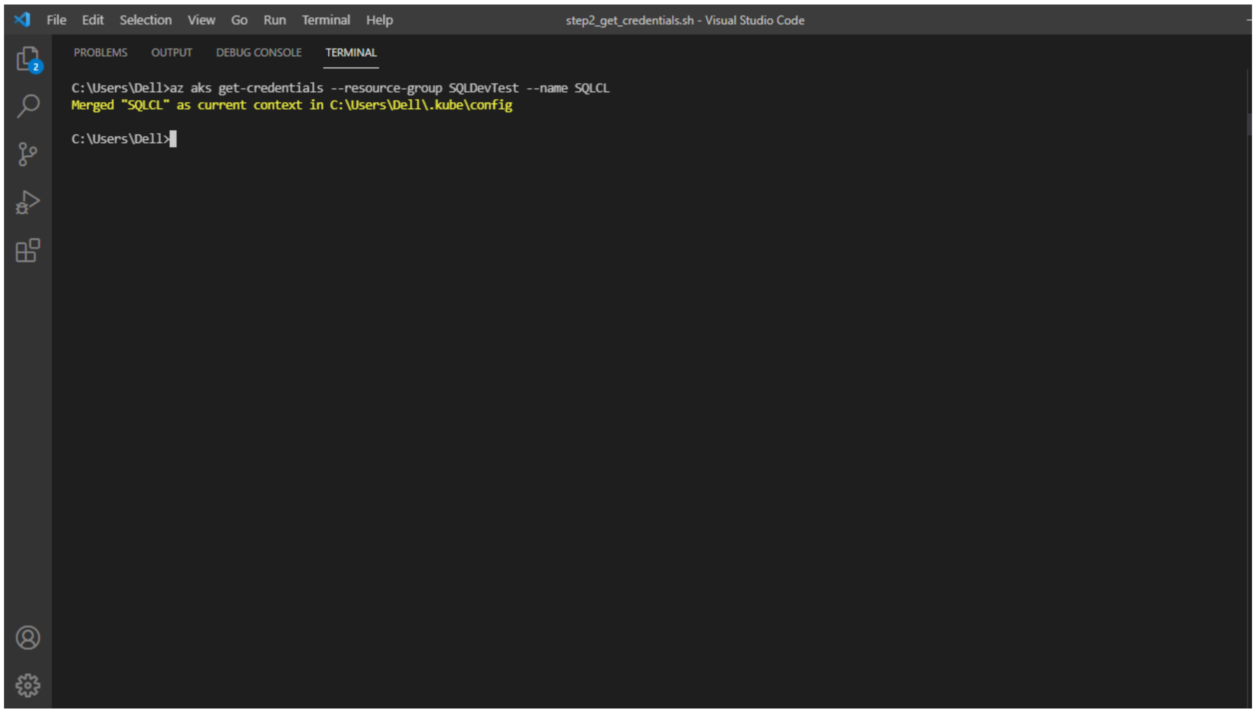
Load balancer	Standard
Enable HTTP application routing	☐

HTTP application routing is only recommended for dev/test clusters

Security

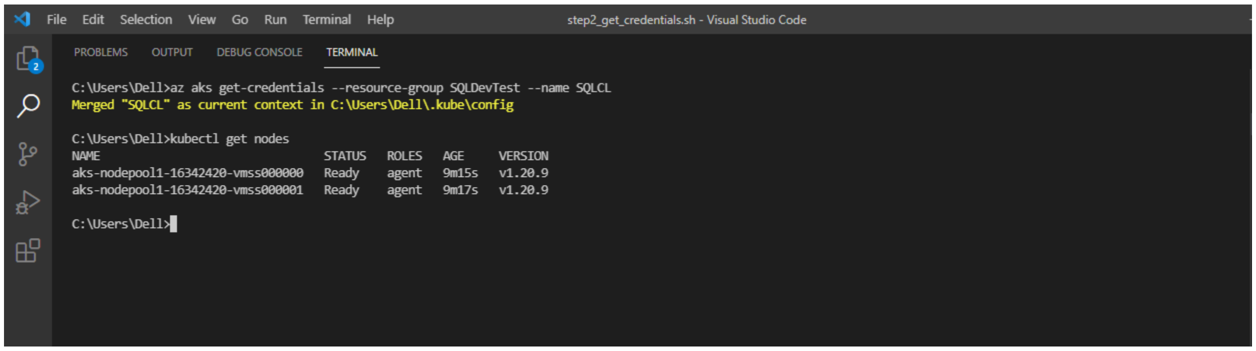
Step 05: Get the credential for this cluster

```
az aks get-credentials --resource-group SQLDevTest --name SQLCL
```


File Edit Selection View Go Run Terminal Help
step2_get_credentials.sh - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Users\Dell>az aks get-credentials --resource-group SQLDevTest --name SQLCL
Merged "SQLCL" as current context in C:\Users\Dell\.kube\config

C:\Users\Dell>
```

Get the list of Nodes available in the cluster.


File Edit Selection View Go Run Terminal Help
step2_get_credentials.sh - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Users\Dell>az aks get-credentials --resource-group SQLDevTest --name SQLCL
Merged "SQLCL" as current context in C:\Users\Dell\.kube\config

C:\Users\Dell>kubectl get nodes
NAME                                STATUS   ROLES   AGE    VERSION
aks-nodepool1-16342420-vmss000000   Ready    agent   9m15s  v1.20.9
aks-nodepool1-16342420-vmss000001   Ready    agent   9m17s  v1.20.9

C:\Users\Dell>
```

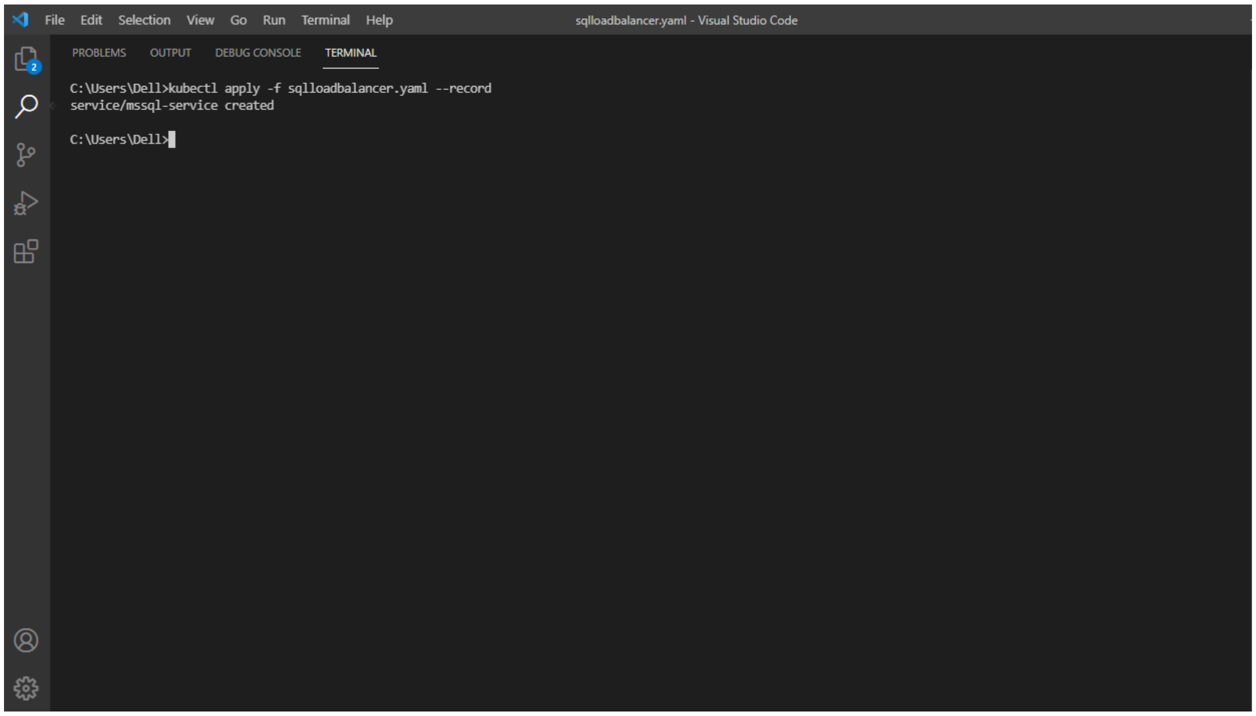
Step 06: Creating the load Balancer service.

Create filename.yaml file using below code (in this example file name is sqlloadbalancer.yaml)

```
# Create the load balancing service
apiVersion: v1
kind: Service
metadata:
  name: mssql-service
spec:
  selector:
    app: mssql
  ports:
    - protocol: TCP
      port: 1433
      targetPort: 1433
  type: LoadBalancer
```

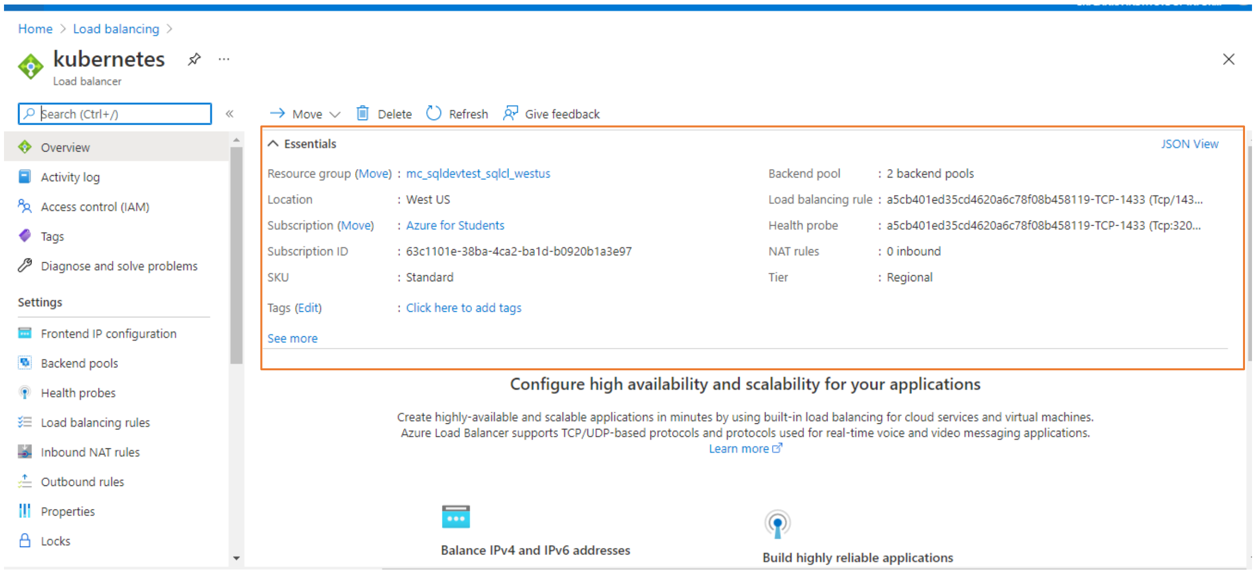
Applying load balancer to the Kubernetes service. The external port **1433** mapped to the internal port **1433**

```
kubectl apply -f sqlloadbalancer.yaml --record
```


File Edit Selection View Go Run Terminal Help
sqlloadbalancer.yaml - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Users\Dell>kubectl apply -f sqlloadbalancer.yaml --record
service/mssql-service created

C:\Users\Dell>
```

Once we create the load balancer from terminal, we can see that in the azure portal.



Home > Load balancing >

kubernetes
Load balancer

Resource group (Move)	: mc_sqldevtest_sqlcl_westus	Backend pool	: 2 backend pools
Location	: West US	Load balancing rule	: a5cb401ed35cd4620a6c78f08b458119-TCP-1433 (Tcp/143...
Subscription (Move)	: Azure for Students	Health probe	: a5cb401ed35cd4620a6c78f08b458119-TCP-1433 (Tcp:320...
Subscription ID	: 63c1101e-38ba-4ca2-ba1d-b0920b1a3e97	NAT rules	: 0 inbound
SKU	: Standard	Tier	: Regional
Tags (Edit)	: Click here to add tags		

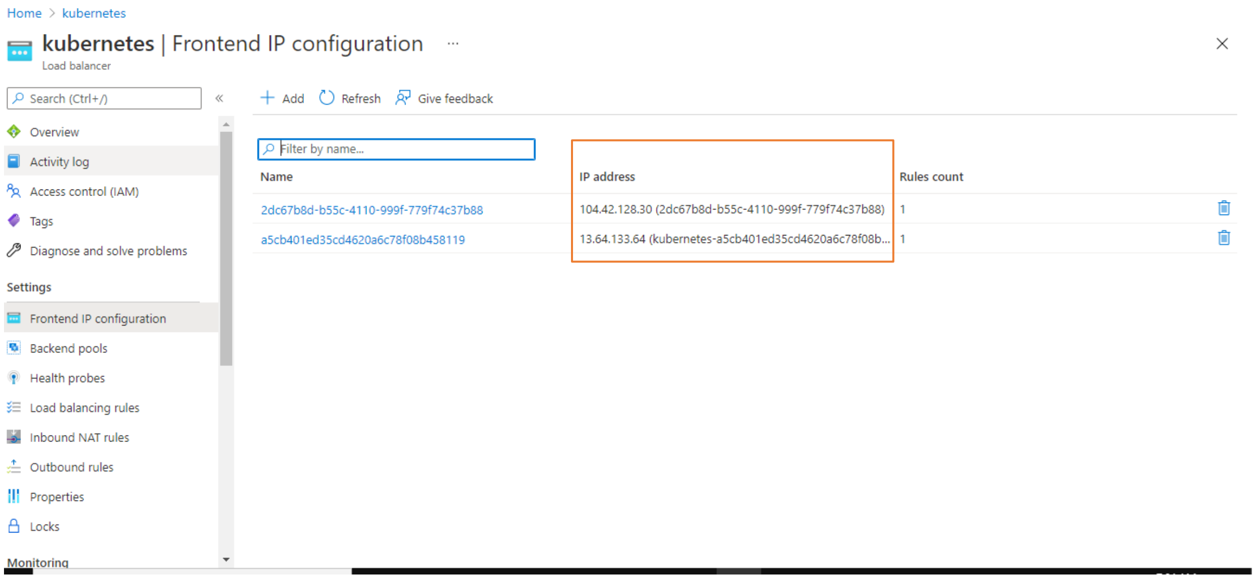
See more

Configure high availability and scalability for your applications

Create highly-available and scalable applications in minutes by using built-in load balancing for cloud services and virtual machines. Azure Load Balancer supports TCP/UDP-based protocols and protocols used for real-time voice and video messaging applications.
Learn more

Balance IPv4 and IPv6 addresses
Build highly reliable applications

Frontend IP Configurations. As you can see two public IP addresses configured to both nodes.

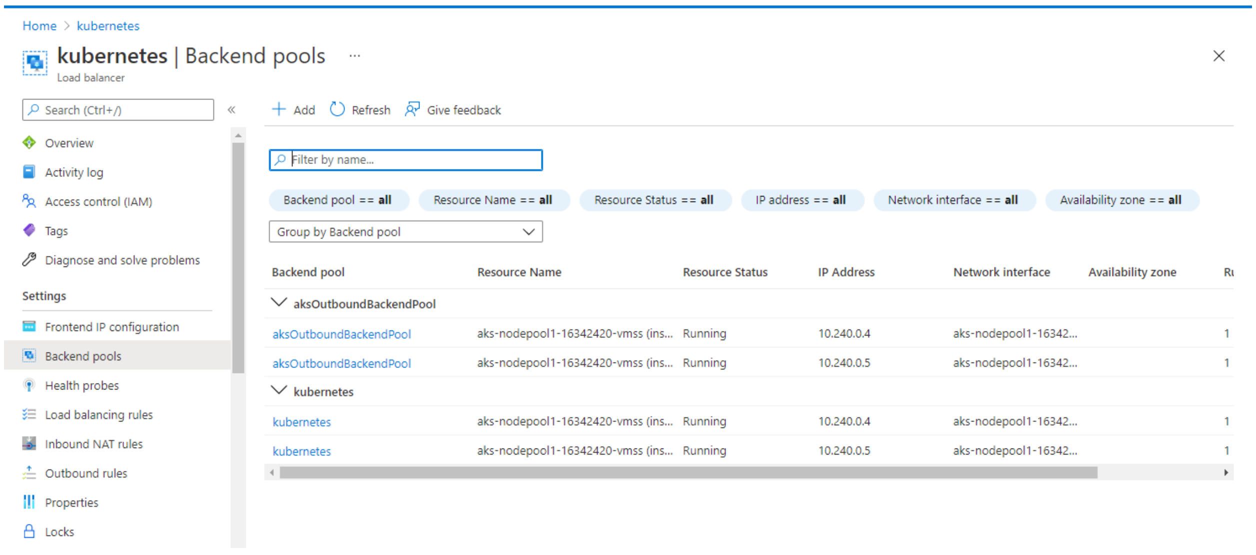


Home > kubernetes

kubernetes | Frontend IP configuration
Load balancer

Name	IP address	Rules count
2dc67b8d-b55c-4110-999f-779f74c37b88	104.42.128.30 (2dc67b8d-b55c-4110-999f-779f74c37b88)	1
a5cb401ed35cd4620a6c78f08b458119	13.64.133.64 (kubernetes-a5cb401ed35cd4620a6c78f08b...	1

Backend Pool configurations should be like this.



Home > kubernetes

kubernetes | Backend pools ···

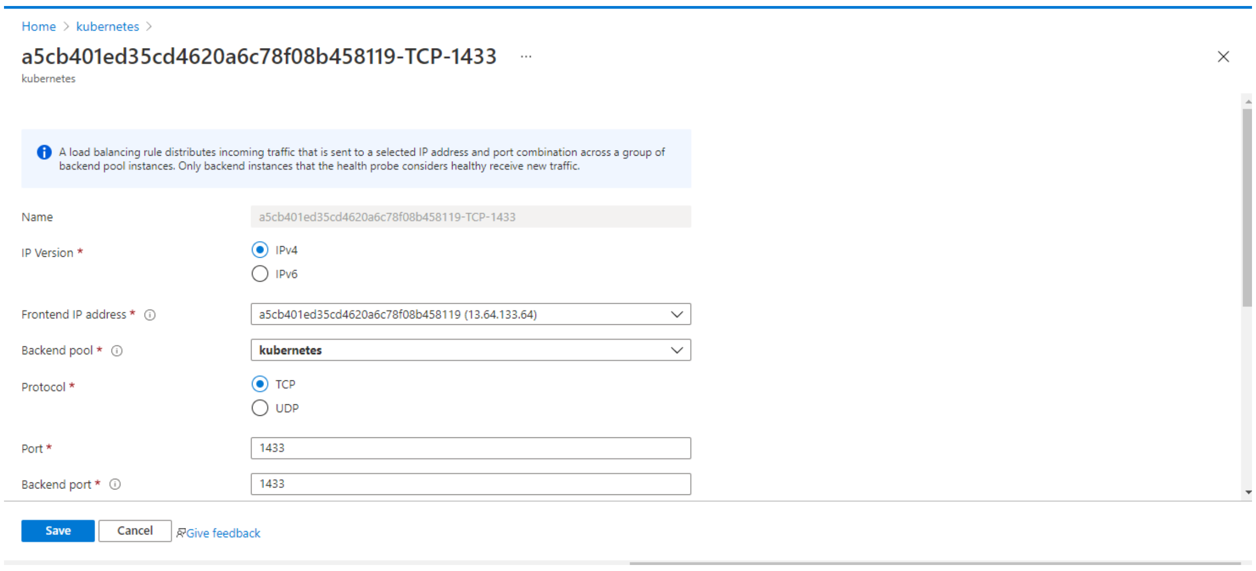
Load balancer

Backend pool == **all** | Resource Name == **all** | Resource Status == **all** | IP address == **all** | Network interface == **all** | Availability zone == **all**

Group by Backend pool

Backend pool	Resource Name	Resource Status	IP Address	Network interface	Availability zone	Ru
aksOutboundBackendPool						
aksOutboundBackendPool	aks-nodepool1-16342420-vmss (ins...	Running	10.240.0.4	aks-nodepool1-16342...		1
aksOutboundBackendPool	aks-nodepool1-16342420-vmss (ins...	Running	10.240.0.5	aks-nodepool1-16342...		1
kubernetes						
kubernetes	aks-nodepool1-16342420-vmss (ins...	Running	10.240.0.4	aks-nodepool1-16342...		1
kubernetes	aks-nodepool1-16342420-vmss (ins...	Running	10.240.0.5	aks-nodepool1-16342...		1

Load Balancing Rules should be configured to expose the application to external network.



Home > kubernetes >

a5cb401ed35cd4620a6c78f08b458119-TCP-1433 ···

kubernetes

A load balancing rule distributes incoming traffic that is sent to a selected IP address and port combination across a group of backend pool instances. Only backend instances that the health probe considers healthy receive new traffic.

Name	a5cb401ed35cd4620a6c78f08b458119-TCP-1433
IP Version *	[✓ IPv4] IPv6
Frontend IP address *	a5cb401ed35cd4620a6c78f08b458119 (13.64.133.64)
Backend pool *	**kubernetes**
Protocol *	[✓ TCP] UDP
Port *	1433
Backend port *	1433

In this example our public IP is 13.64.133.64. The external port is 1433.

So far, we have created Kubernetes service and the load balancer. Next let's create the external storage.

Step 07: Create external Storage (storage class and persistent volume claim)

Use below command to create the storage.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: azure-disk
provisioner: kubernetes.io/azure-disk
parameters:
  storageaccounttype: Standard_LRS
  kind: Managed
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: mssql-data
  annotations:
    volume.beta.kubernetes.io/storage-class: azure-disk
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 8Gi
```

Use below command to apply the storage.

```
kubectl apply -f sqlstorage.yaml
```

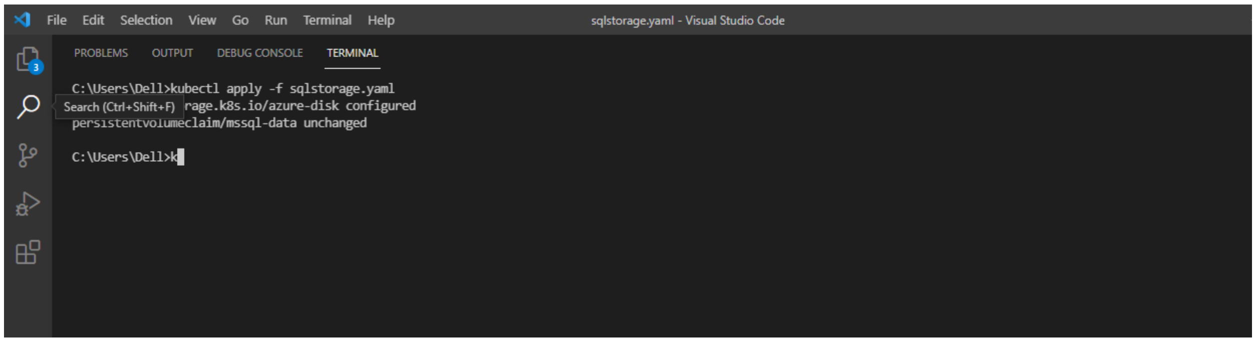
In this example, below azure storage class used.

```
apiVersion: storage.k8s.io/v1
```

You can find many more storage classes available. For more information, please refer azure storage documentations

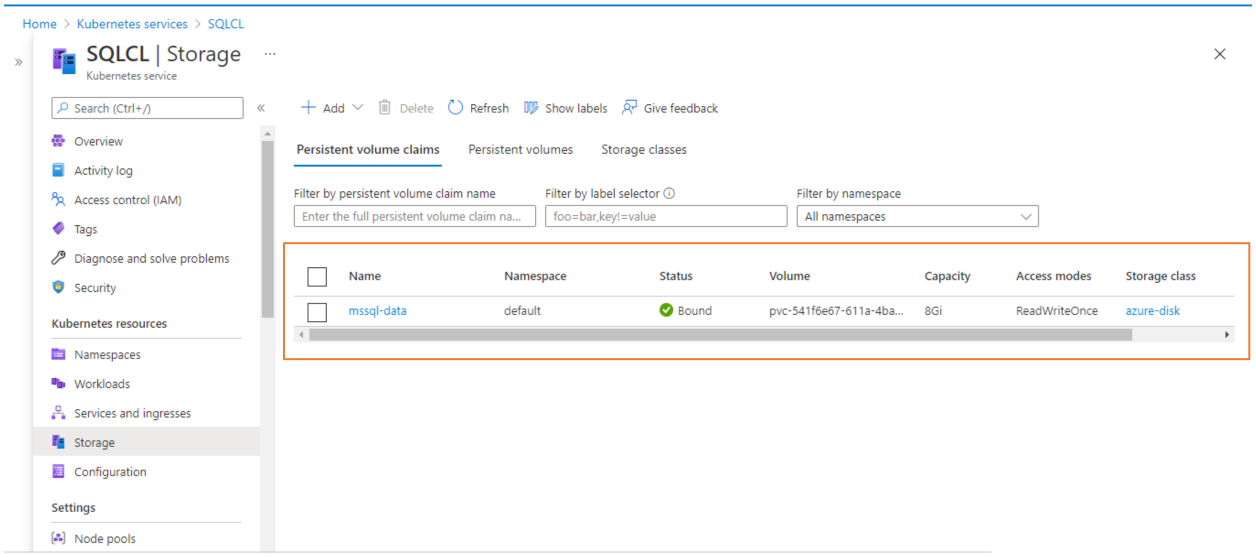
https://docs.microsoft.com/en-us/azure/storage/

Now we have created the external storage.



```
C:\Users\Dell>kubectl apply -f sqlstorage.yaml
Search (Ctrl+Shift+F) rage.k8s.io/azure-disk configured
persistentvolumeclaim/mssql-data unchanged

C:\Users\Dell>k
```

Our persistence volume claim is 8GB.

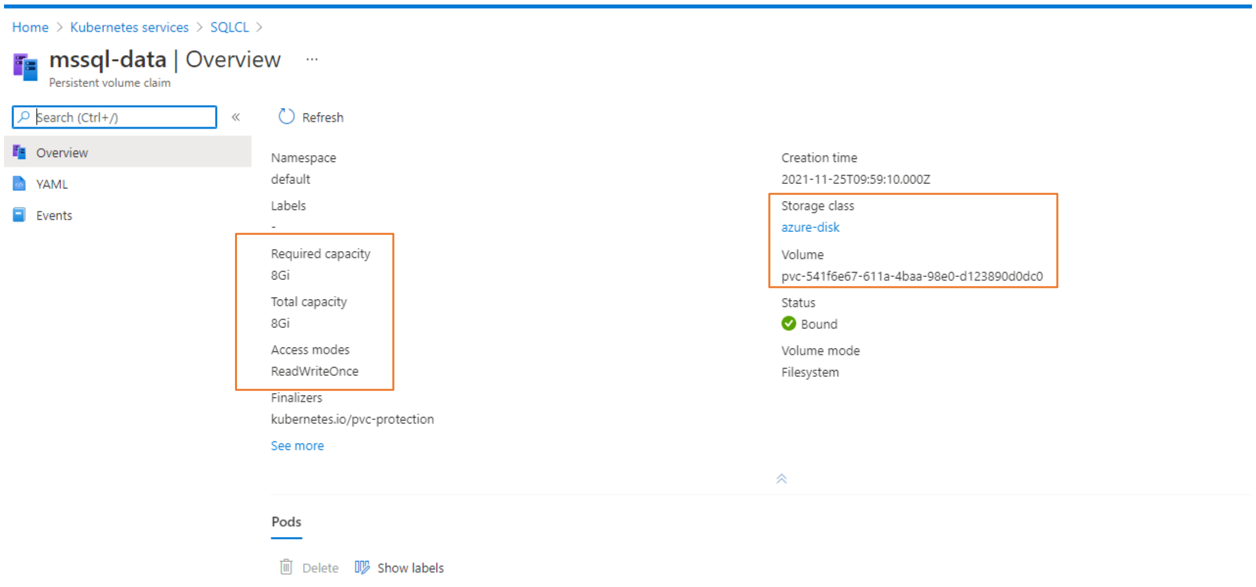


Home > Kubernetes services > SQLCL

SQLCL | Storage
Kubernetes service

Persistent volume claims | Persistent volumes | Storage classes

Name	Namespace	Status	Volume	Capacity	Access modes	Storage class
mssql-data	default	Bound	pvc-541f6e67-611a-4ba...	8Gi	ReadWriteOnce	azure-disk



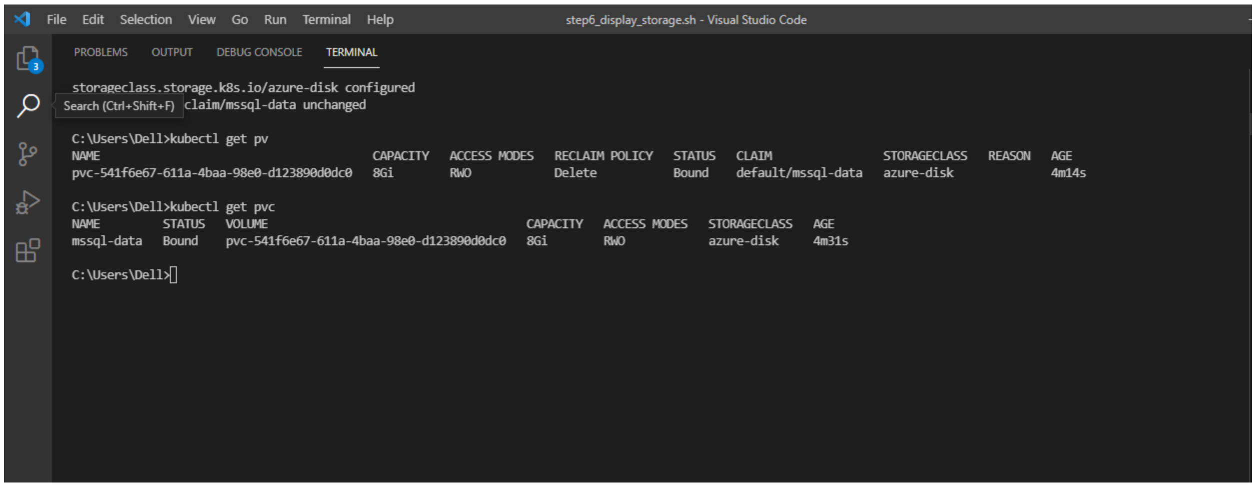
Home > Kubernetes services > SQLCL >

mssql-data | Overview
Persistent volume claim

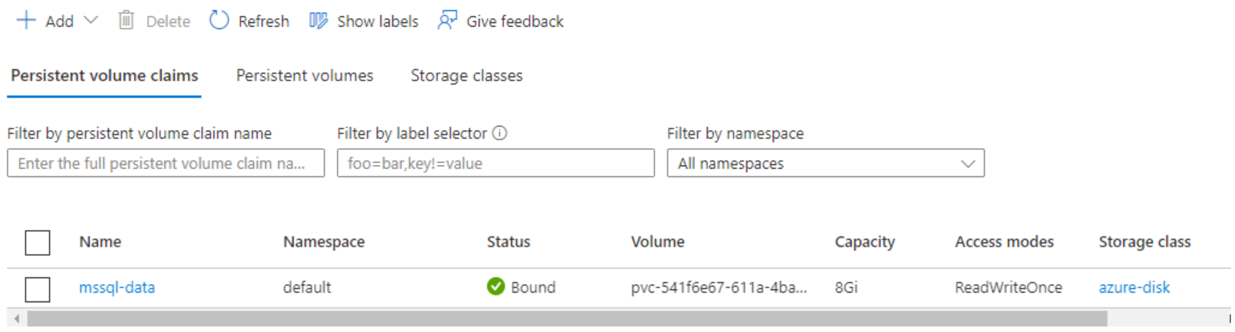
Namespace	default
Labels	-
Required capacity	8Gi
Total capacity	8Gi
Access modes	ReadWriteOnce
Finalizers	kubernetes.io/pvc-protection
Creation time	2021-11-25T09:59:10.000Z
Storage class	azure-disk
Volume	pvc-541f6e67-611a-4baa-98e0-d123890d0dc0
Status	Bound
Volume mode	Filesystem

We can find more information about persistent volume and claim using below commands.

```
# Display the persistent volume and claim
kubectl get pv
kubectl get pvc
```

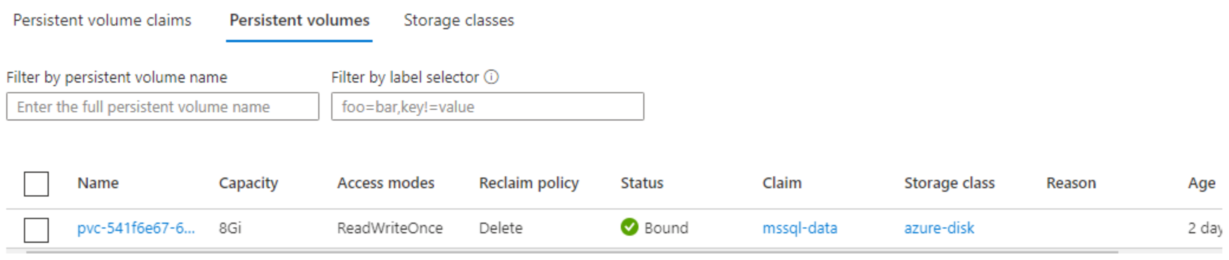


```
storageclass.storage.k8s.io/azure-disk configured
                      claim/mssql-data unchanged

C:\Users\Dell>kubectl get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                STORAGECLASS   REASON   AGE
pvc-541f6e67-611a-4baa-98e0-d123890d0dc0   8Gi        RWO            Delete           Bound    default/mssql-data   azure-disk              4m14s

C:\Users\Dell>kubectl get pvc
NAME         STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mssql-data   Bound    pvc-541f6e67-611a-4baa-98e0-d123890d0dc0   8Gi        RWO            azure-disk     4m31s

C:\Users\Dell>
```



Filter by persistent volume claim name: Enter the full persistent volume claim na...
Filter by label selector: foo=bar,key!=value
Filter by namespace: All namespaces

Name	Namespace	Status	Volume	Capacity	Access modes	Storage class
mssql-data	default	✓ Bound	pvc-541f6e67-611a-4ba...	8Gi	ReadWriteOnce	azure-disk

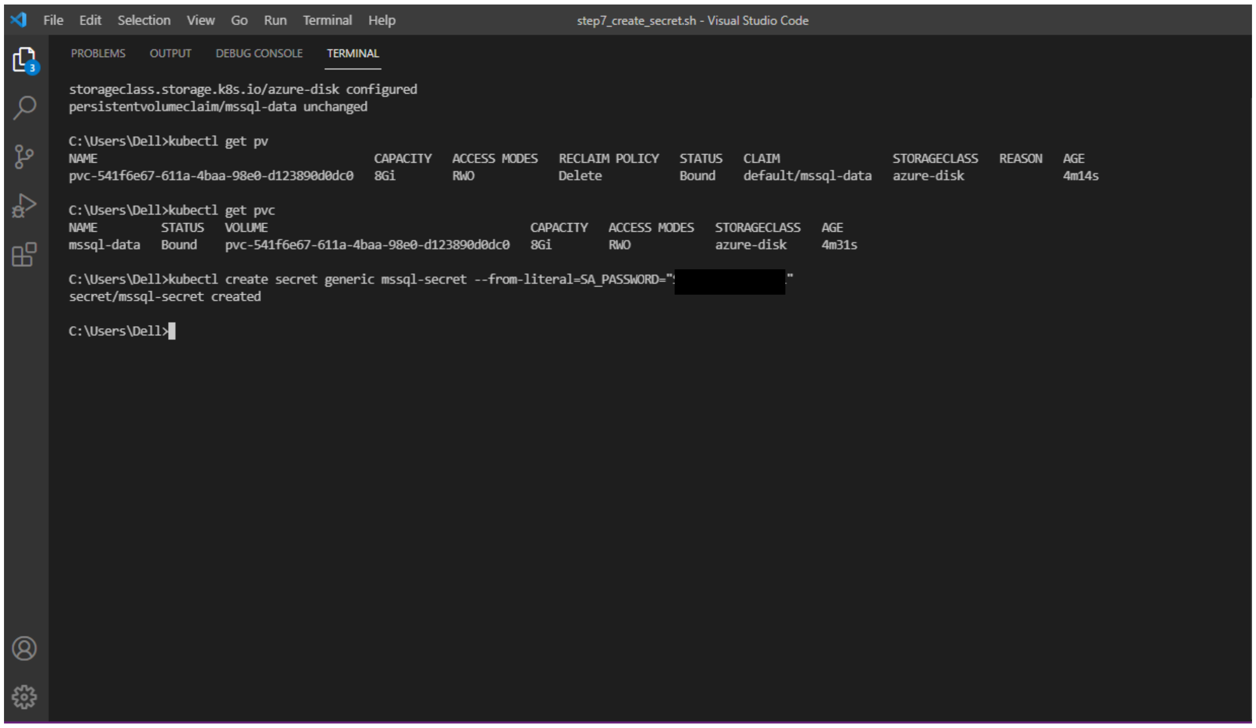


Filter by persistent volume name: Enter the full persistent volume name
Filter by label selector: foo=bar,key!=value

Name	Capacity	Access modes	Reclaim policy	Status	Claim	Storage class	Reason	Age
pvc-541f6e67-6...	8Gi	ReadWriteOnce	Delete	✓ Bound	mssql-data	azure-disk		2 day

Step 08: Create Secret for SQL Server container. It is not required to create a secret. However, it is a best practice to create a secret before SQL server deployment.

```
sa password for SQL Server container
kubectl create secret generic mssql-secret --from-literal=SA_PASSWORD="Your
Password"
```


File Edit Selection View Go Run Terminal Help
step7_create_secret.sh - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
storageclass.storage.k8s.io/azure-disk configured
persistentvolumeclaim/mssql-data unchanged

C:\Users\Dell>kubectl get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                STORAGECLASS   REASON   AGE
pvc-541f6e67-611a-4baa-98e0-d123890d0dc0   8Gi        RWO            Delete           Bound    default/mssql-data   azure-disk              4m14s

C:\Users\Dell>kubectl get pvc
NAME         STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mssql-data   Bound    pvc-541f6e67-611a-4baa-98e0-d123890d0dc0   8Gi        RWO            azure-disk     4m31s

C:\Users\Dell>kubectl create secret generic mssql-secret --from-literal=SA_PASSWORD="[redacted]"
secret/mssql-secret created

C:\Users\Dell>
```

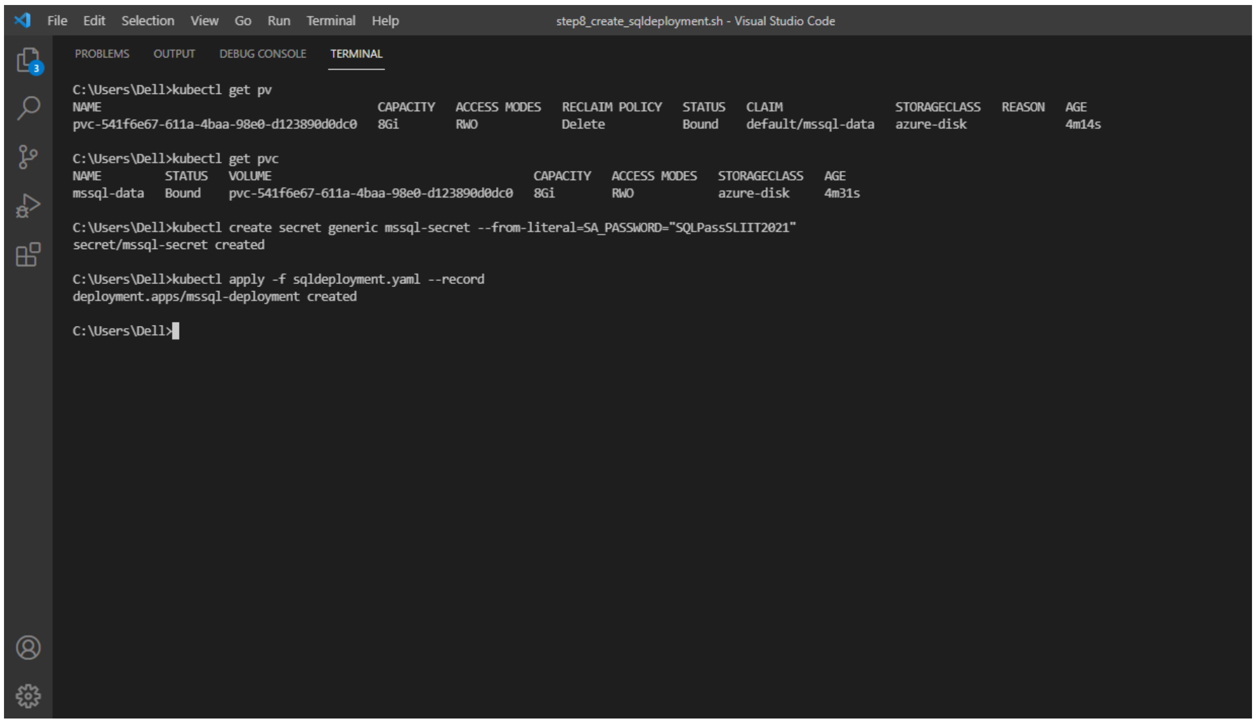
Step 09: Create the SQL server deployment

This will configure MSSQL image on the container. In this example, we are using mcr.microsoft.com/mssql/rhel/server:2019-latest image. In the image, we are using MSSQL Server developer edition.

```
  apiVersion: apps/v1
kind: Deployment
metadata:
  name: mssql-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mssql
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: mssql
    spec:
      terminationGracePeriodSeconds: 10
      securityContext:
        fsGroup: 1000
      containers:
      - name: mssql
        image: mcr.microsoft.com/mssql/rhel/server:2019-latest
        env:
        - name: MSSQL_PID
          value: "Developer"
        - name: ACCEPT_EULA
          value: "Y"
        - name: MSSQL_SA_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mssql-secret
              key: SA_PASSWORD
        volumeMounts:
        - name: mssqldb
          mountPath: /var/opt/mssql
      volumes:
      - name: mssqldb
        persistentVolumeClaim:
          claimName: mssql-data
```

Run below command to create the deployment. It will take few minutes to configure everything.

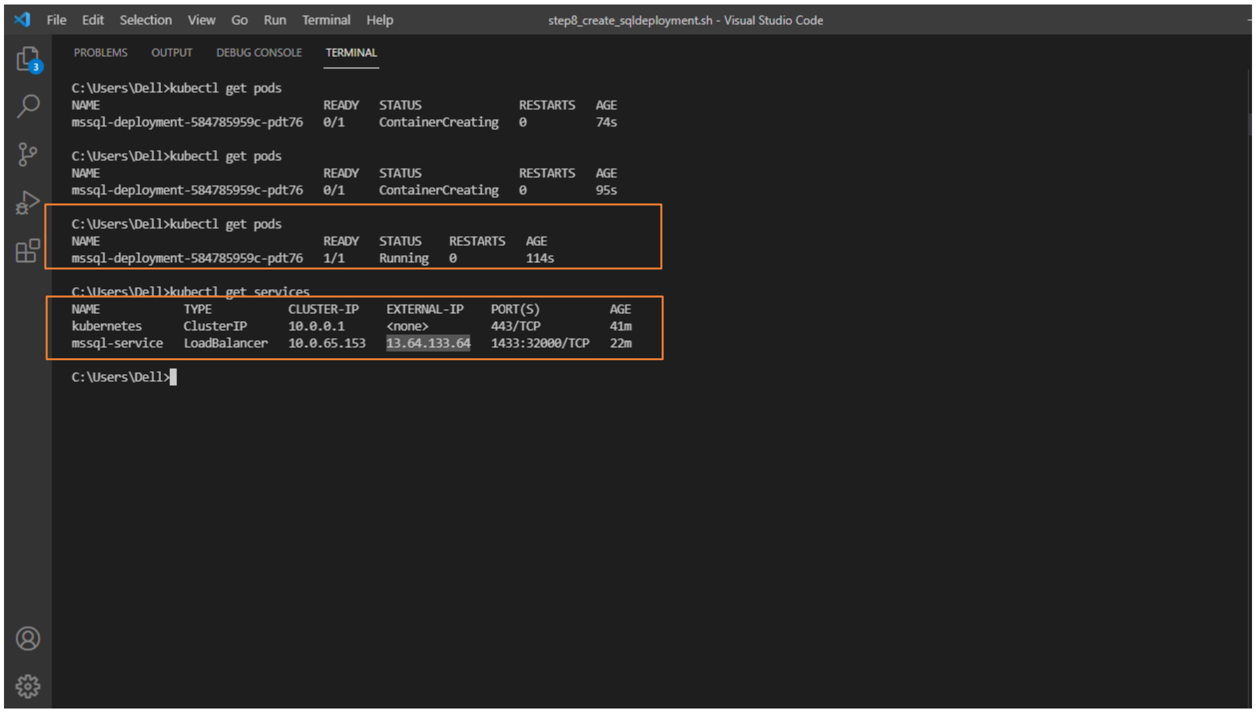
```
kubectl apply -f sqldeployment.yaml --record
```



File Edit Selection View Go Run Terminal Help
step8_create_sqldeployment.sh - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Users\Dell>kubectl get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                STORAGECLASS   REASON   AGE
pvc-541f6e67-611a-4baa-98e0-d123890d0dc0   8Gi        RWO            Delete           Bound    default/mssql-data   azure-disk              4m14s

C:\Users\Dell>kubectl get pvc
NAME         STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mssql-data   Bound    pvc-541f6e67-611a-4baa-98e0-d123890d0dc0   8Gi        RWO            azure-disk     4m31s

C:\Users\Dell>kubectl create secret generic mssql-secret --from-literal=SA_PASSWORD="SQLPassSLIIT2021"
secret/mssql-secret created

C:\Users\Dell>kubectl apply -f sqldeployment.yaml --record
deployment.apps/mssql-deployment created

C:\Users\Dell>
```

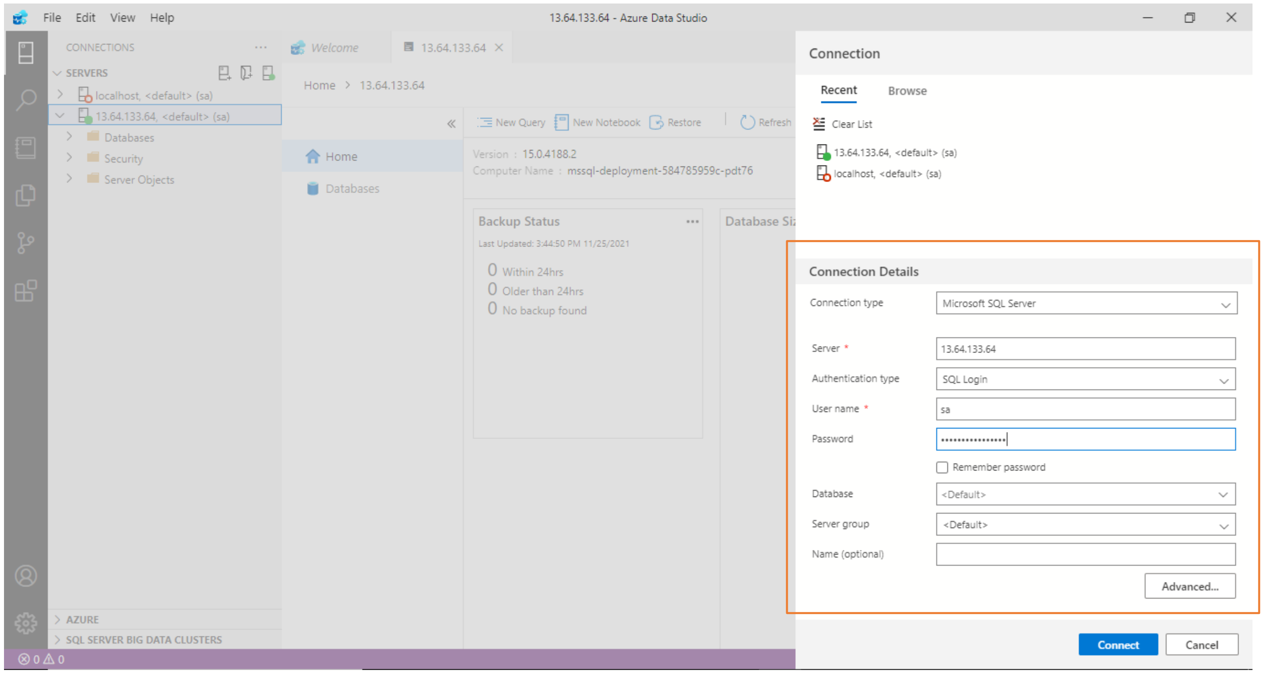
Let’s list down the pods and services configured.



File Edit Selection View Go Run Terminal Help
step8_create_sqldeployment.sh - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Users\Dell>kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
mssql-deployment-584785959c-pdt76   0/1     ContainerCreating   0          74s

C:\Users\Dell>kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
mssql-deployment-584785959c-pdt76   0/1     ContainerCreating   0          95s

C:\Users\Dell>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
mssql-deployment-584785959c-pdt76   1/1     Running   0          114s

C:\Users\Dell>kubectl get services
NAME            TYPE           CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes      ClusterIP      10.0.0.1      <none>         443/TCP          41m
mssql-service   LoadBalancer   10.0.65.153   13.64.133.64   1433:32000/TCP   22m

C:\Users\Dell>
```

Step 10: Test the connections to newly created. You can use azure data studio IDE to connect to the server.



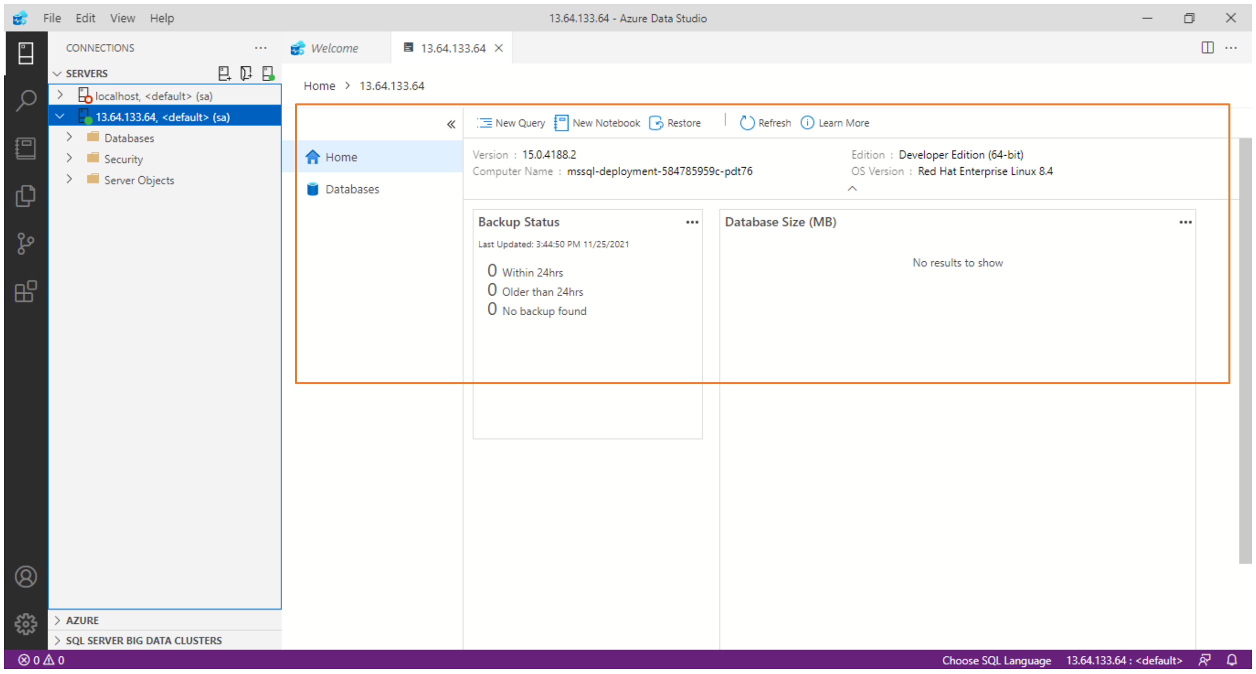
Version : 15.0.4188.2
Computer Name : mssql-deployment-584785959c-pdt76

Last Updated: 3:44:50 PM 11/25/2021
0 Within 24hrs
0 Older than 24hrs
0 No backup found

Connection Details

Connection type	Microsoft SQL Server
Server *	13.64.133.64
Authentication type	SQL Login
User name *	sa
Password	••••••••••••••••
	☐ Remember password
Database	<Default>
Server group	<Default>
Name (optional)	

In the container our operating system is Red Hat Enterprise Linux.

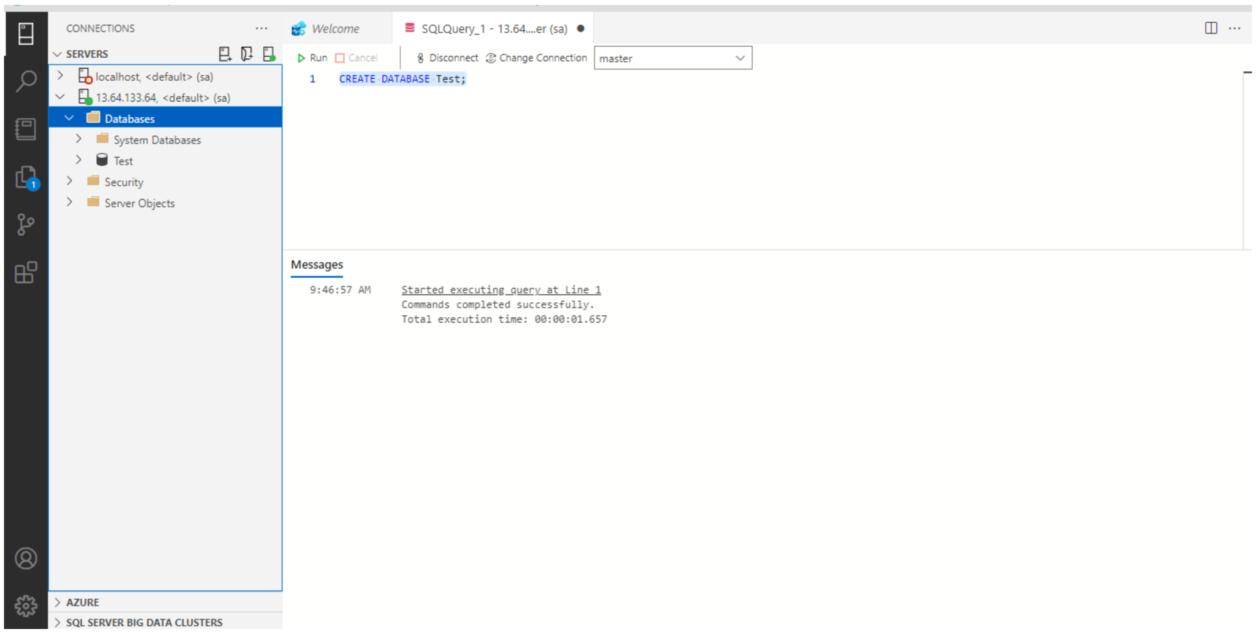


Version : 15.0.4188.2
Computer Name : mssql-deployment-584785959c-pdt76
Edition : Developer Edition (64-bit)
OS Version : Red Hat Enterprise Linux 8.4

Backup Status
Last Updated: 3:44:50 PM 11/25/2021
0 Within 24hrs
0 Older than 24hrs
0 No backup found

Database Size (MB)
No results to show

Let’s ty to create a database in newly created MSSQL Server.



```
CREATE DATABASE Test;
```

```
9:46:57 AM      Started executing query at Line 1
                Commands completed successfully.
                Total execution time: 00:00:01.657
```

Now we have configured MSSQL server with different node. Let's find out high availability of new created Kubernetes service. Let's simulate by killing the pod. You can run below .yaml file to accomplish the simulation.

Assign the pod Name

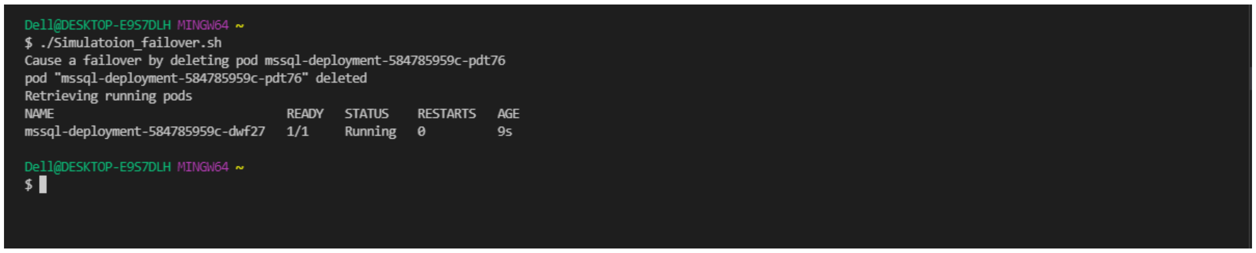
```
podname=$(kubectl get pods | grep mssql | cut -c1-33)
```

Delete the Pod

```
echo Cause a failover by deleting pod $podname
kubectl delete pod $podname
```

Let's check again the running pods.

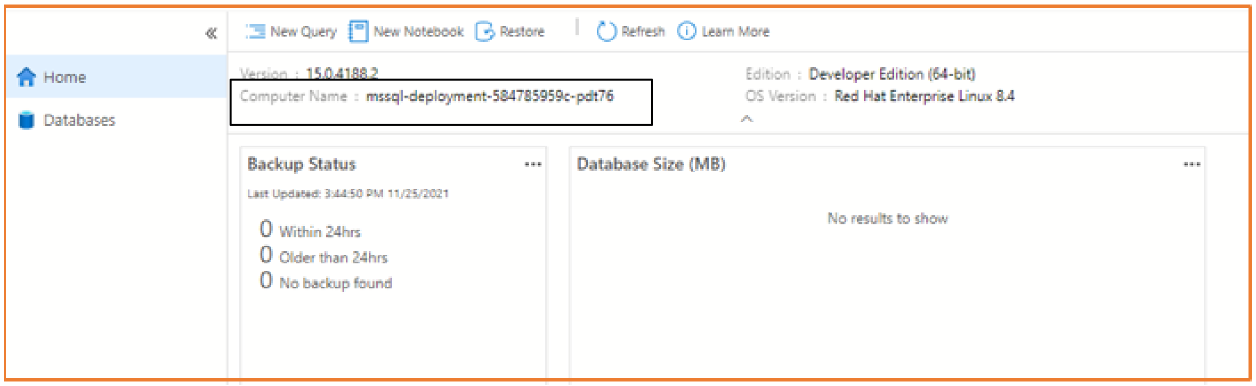
```
echo Retrieving running pods
kubectl get pods
```



```
Dell@DESKTOP-E9S7DLH MINGW64 ~
$ ./Simulatoion_failover.sh
Cause a failover by deleting pod mssql-deployment-584785959c-pdt76
pod "mssql-deployment-584785959c-pdt76" deleted
Retrieving running pods
NAME                                READY   STATUS    RESTARTS   AGE
mssql-deployment-584785959c-dwf27   1/1     Running   0          9s

Dell@DESKTOP-E9S7DLH MINGW64 ~
$ 
```

Now we have deleted the pods. Let's try to connect the SQL server again and compare with previous settings.

Previous settings

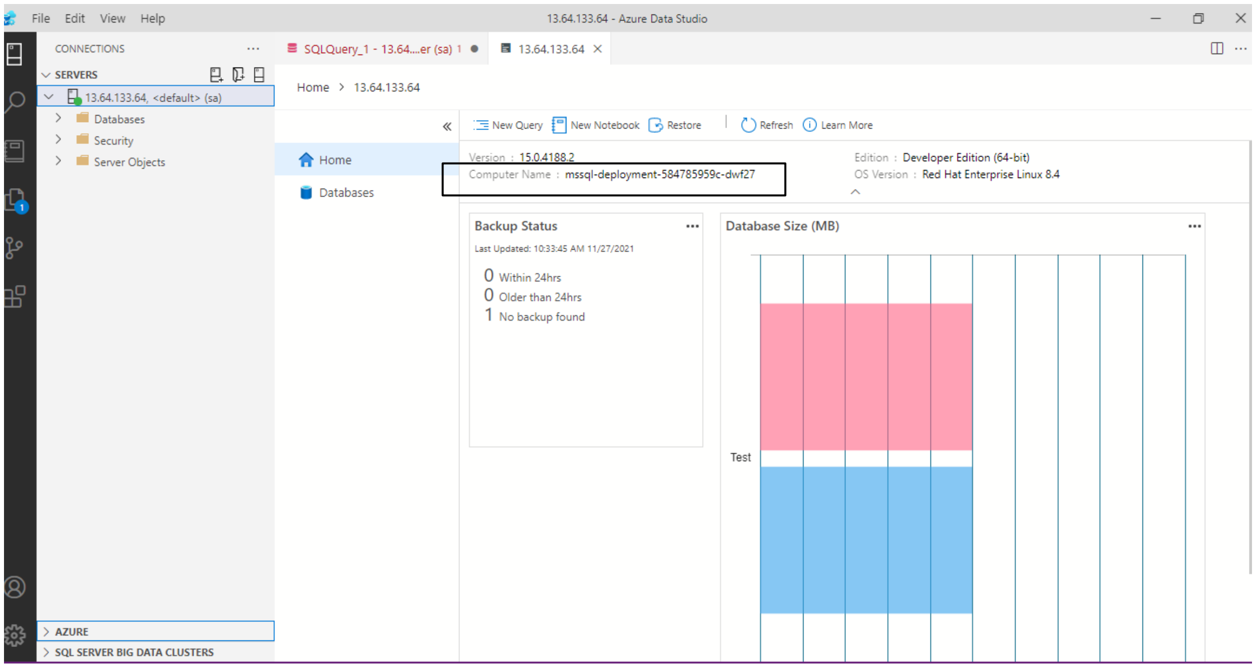


Version : 15.0.4188.2
Computer Name : mssql-deployment-584785959c-pdt76
Edition : Developer Edition (64-bit)
OS Version : Red Hat Enterprise Linux 8.4

Backup Status
Last Updated: 3:44:50 PM 11/25/2021
0 Within 24hrs
0 Older than 24hrs
0 No backup found

Database Size (MB)
No results to show

New configurations



Version : 15.0.4188.2
Computer Name : mssql-deployment-584785959c-dwf27
Edition : Developer Edition (64-bit)
OS Version : Red Hat Enterprise Linux 8.4

Last Updated: 10:33:45 AM 11/27/2021
0 Within 24hrs
0 Older than 24hrs
1 No backup found

As you can see computer name has been changed the after the failover. However, there is no change in the operational database. This way we can ensure that Kubernetes services to handle all the failover and providing the high availability to stateful applications such as MSSQL Server.