

Practical No.1

Aim: Building APT.NET Core MVC Application.

- 1) Install .Net Core Sdk (Link: <https://dotnet.microsoft.com/learn/dotnet/hello-world-tutorial/install>)
- 2) create folder MyMVC folder in C: drive or any other drive
- 3) open command prompt and perform following operations
 Command: to create mvc project
 dotnet new mvc --auth none

Output:



```

C:\windows\system32\cmd.exe

C:\Users>cd..
C:\>cd mymvc
C:\MyMVC>dotnet new mvc --auth none

Welcome to .NET 5.0!
-----
SDK Version: 5.0.301

Telemetry
-----
The .NET tools collect usage data in order to help us improve your experience. It is collected by Microsoft and shared with the community. You can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variable to '1' or 'true' using your favorite shell.












Read more about .NET CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

-----
Installed an ASP.NET Core HTTPS development certificate.
To trust the certificate run 'dotnet dev-certs https --trust' (Windows and macOS only).
Learn about HTTPS: https://aka.ms/dotnet-https
-----
Write your first app: https://aka.ms/dotnet-hello-world
Find out what's new: https://aka.ms/dotnet-whats-new
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cli
-----
Getting ready...
The template "ASP.NET Core Web App (Model-View-Controller)" was created successfully.
This template contains technologies from parties other than Microsoft, see https://aka.ms/aspnetcore/5.0-third-party-notices for details.

Processing post-creation actions...
Running 'dotnet restore' on C:\MyMVC\MyMVC.csproj...
  Determining projects to restore...
  Restored C:\MyMVC\MyMVC.csproj (in 215 ms).
Restore succeeded.

C:\MyMVC>
  
```

- 4) Go to controllers folder and modify HomeController.cs file to match following:

Name	Date modified	Type	Size
 Controllers	08-07-2021 09:46	File folder	
 Models	08-07-2021 09:46	File folder	
 obj	08-07-2021 09:46	File folder	
 Properties	08-07-2021 09:46	File folder	
 Views	08-07-2021 09:46	File folder	
 wwwroot	08-07-2021 09:46	File folder	
 appsettings.Development	08-07-2021 09:46	JSON File	1 KB
 appsettings	08-07-2021 09:46	JSON File	1 KB
 MyMVC	08-07-2021 09:46	CSPROJ File	1 KB
 Program.cs	08-07-2021 09:46	C# Source File	1 KB
 Startup.cs	08-07-2021 09:46	C# Source File	2 KB

HomeController.cs - Notepad

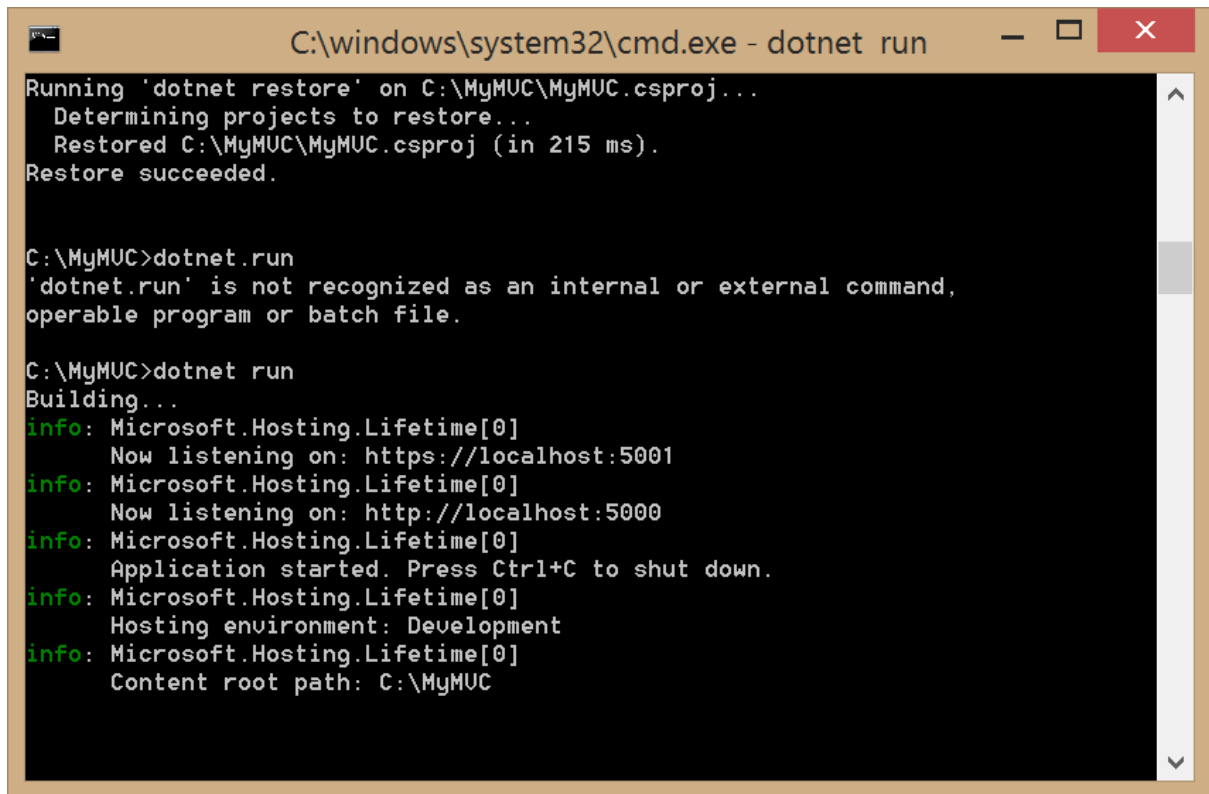
```

File Edit Format View Help
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC.Models;

namespace MyMVC.Controllers
{
    public class HomeController : Controller
    {
        public String Index()
        { return "Hello World"; }
    }
}

```

Run the Project

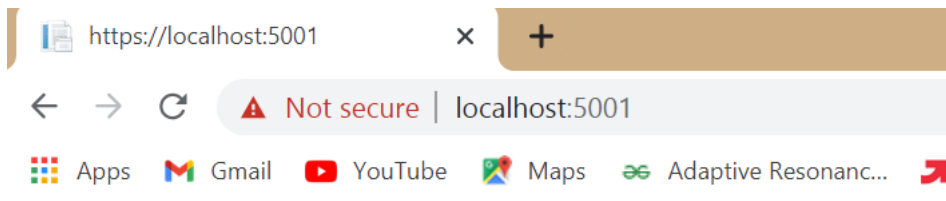


```
C:\windows\system32\cmd.exe - dotnet run
Running 'dotnet restore' on C:\MyMUC\MyMUC.csproj...
Determining projects to restore...
Restored C:\MyMUC\MyMUC.csproj (in 215 ms).
Restore succeeded.

C:\MyMUC>dotnet.run
'dotnet.run' is not recognized as an internal or external command,
operable program or batch file.

C:\MyMUC>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\MyMUC
```

Now open browser and type URL: localhost:5000



Hello World

Now go back to command prompt and stop running project using CTRL+C

```

C:\windows\system32\cmd.exe

Determining projects to restore...
Restored C:\MyMUC\MyMUC.csproj (in 215 ms).
Restore succeeded.

C:\MyMUC>dotnet.run
'dotnet.run' is not recognized as an internal or external command,
operable program or batch file.

C:\MyMUC>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\MyMUC
info: Microsoft.Hosting.Lifetime[0]
      Application is shutting down...

C:\MyMUC>

```

Go to models folder and add new file StockQuote.cs to it with following content

```

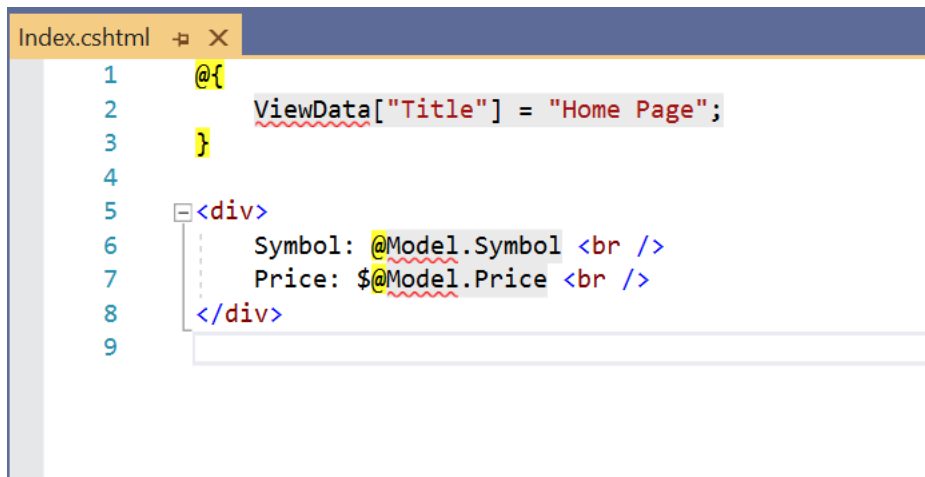
StockQuote.cs - Notepad
File Edit Format View Help
using System;

namespace MyMVC.Models
{
    public class StockQuote
    {
        public string Symbol { get; set; }

        public int Price { get; set; }
    }
}

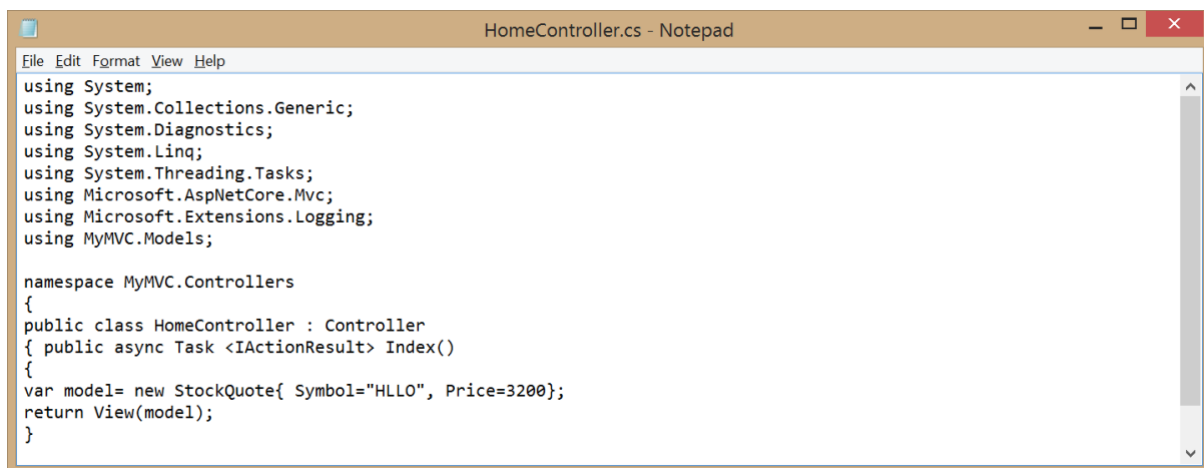
```

Now Add View to folder then home folder in it and modify index.cshtml file to match following



```
1  @{
2      ViewData["Title"] = "Home Page";
3  }
4
5  <div>
6      Symbol: @Model.Symbol <br />
7      Price: $@Model.Price <br />
8  </div>
9
```

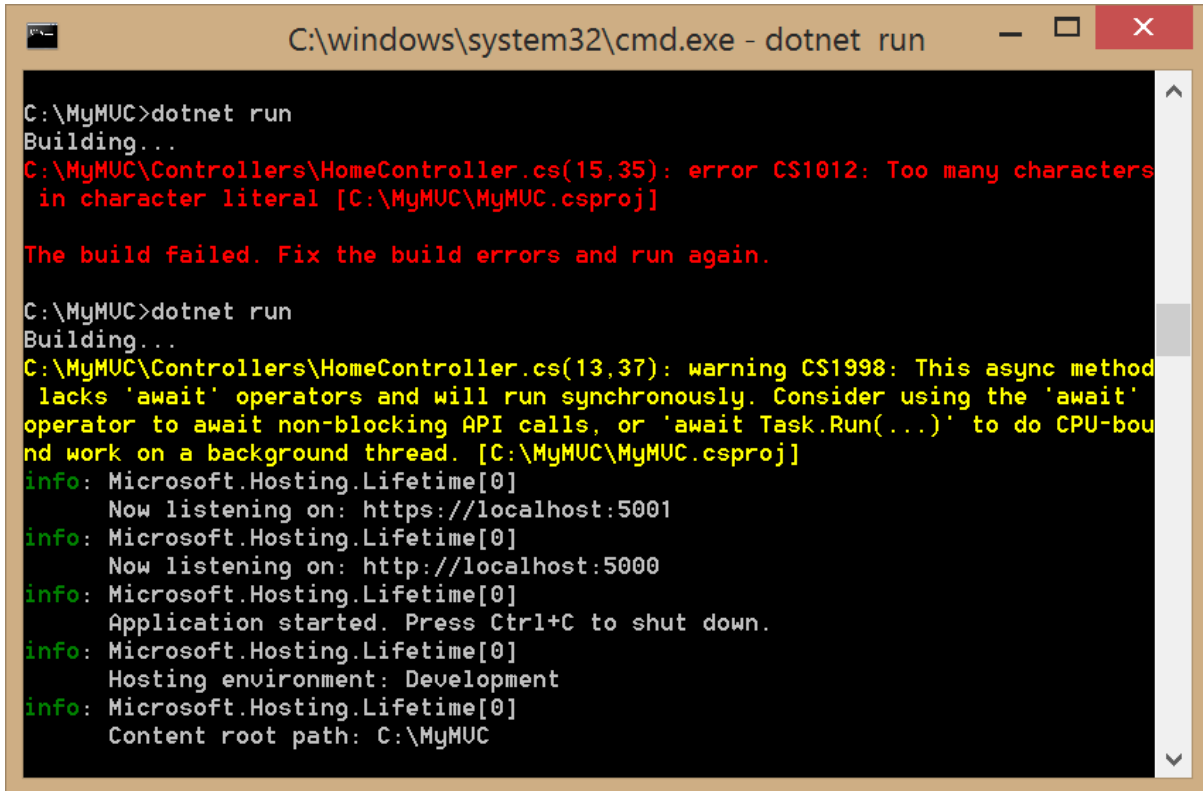
Now modify HomeController.cs file to match following:



```
HomeController.cs - Notepad
File Edit Format View Help
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC.Models;

namespace MyMVC.Controllers
{
    public class HomeController : Controller
    {
        public async Task <ActionResult> Index()
        {
            var model= new StockQuote{ Symbol="HLL0", Price=3200};
            return View(model);
        }
    }
}
```

Now run the project using



```

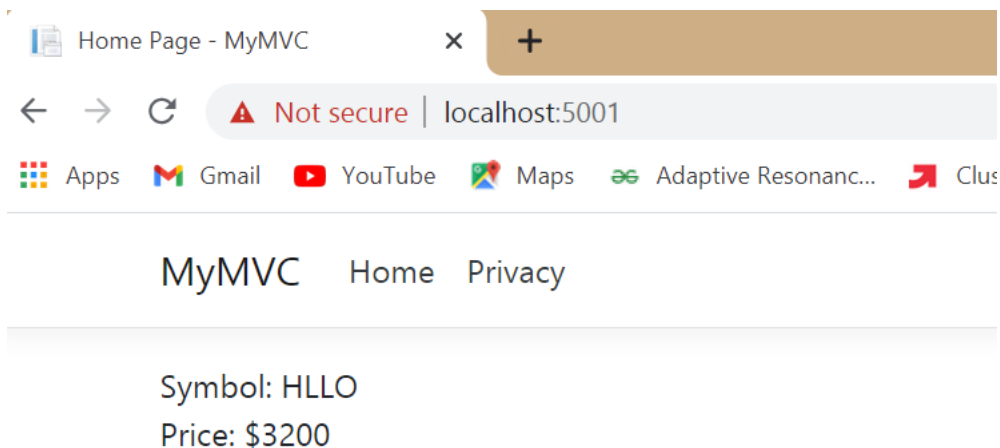
C:\windows\system32\cmd.exe - dotnet run

C:\MyMVC>dotnet run
Building...
C:\MyMVC\Controllers\HomeController.cs(15,35): error CS1012: Too many characters
in character literal [C:\MyMVC\MyMVC.csproj]

The build failed. Fix the build errors and run again.

C:\MyMVC>dotnet run
Building...
C:\MyMVC\Controllers\HomeController.cs(13,37): warning CS1998: This async method
lacks 'await' operators and will run synchronously. Consider using the 'await'
operator to await non-blocking API calls, or 'await Task.Run(...)' to do CPU-bou
nd work on a background thread. [C:\MyMVC\MyMVC.csproj]
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\MyMVC
  
```

Now go back to browser and refresh to get modified view response



Practical NO. 2

Aim: Building ASP.NET Core REST API.

Software requirement:

1. Download and install

To start building .NET apps you just need to download and install the .NET SDK (Software Development Kit version 3.0 above).

Link:

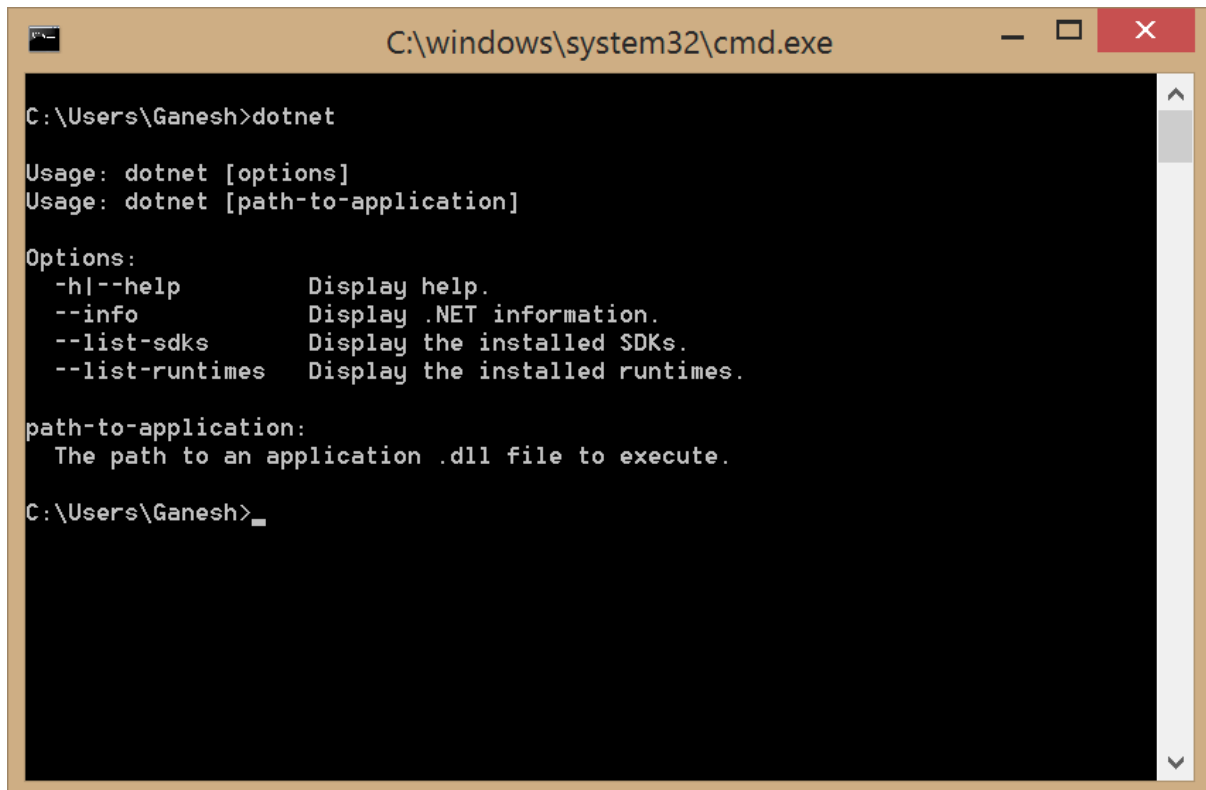
<https://dotnet.microsoft.com/learn/dotnet/hello-world-tutorial/install>

2. Check everything installed correctly

Once you've installed, open a new command prompt and run the following command:

Command prompt

> dotnet



The screenshot shows a Windows Command Prompt window titled "C:\windows\system32\cmd.exe". The prompt is at "C:\Users\Ganesh>". The user has entered the command "dotnet". The output displayed is:

```
C:\Users\Ganesh>dotnet

Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
  -h|--help           Display help.
  --info              Display .NET information.
  --list-sdks         Display the installed SDKs.
  --list-runtimes     Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.

C:\Users\Ganesh>
```

Create your web API

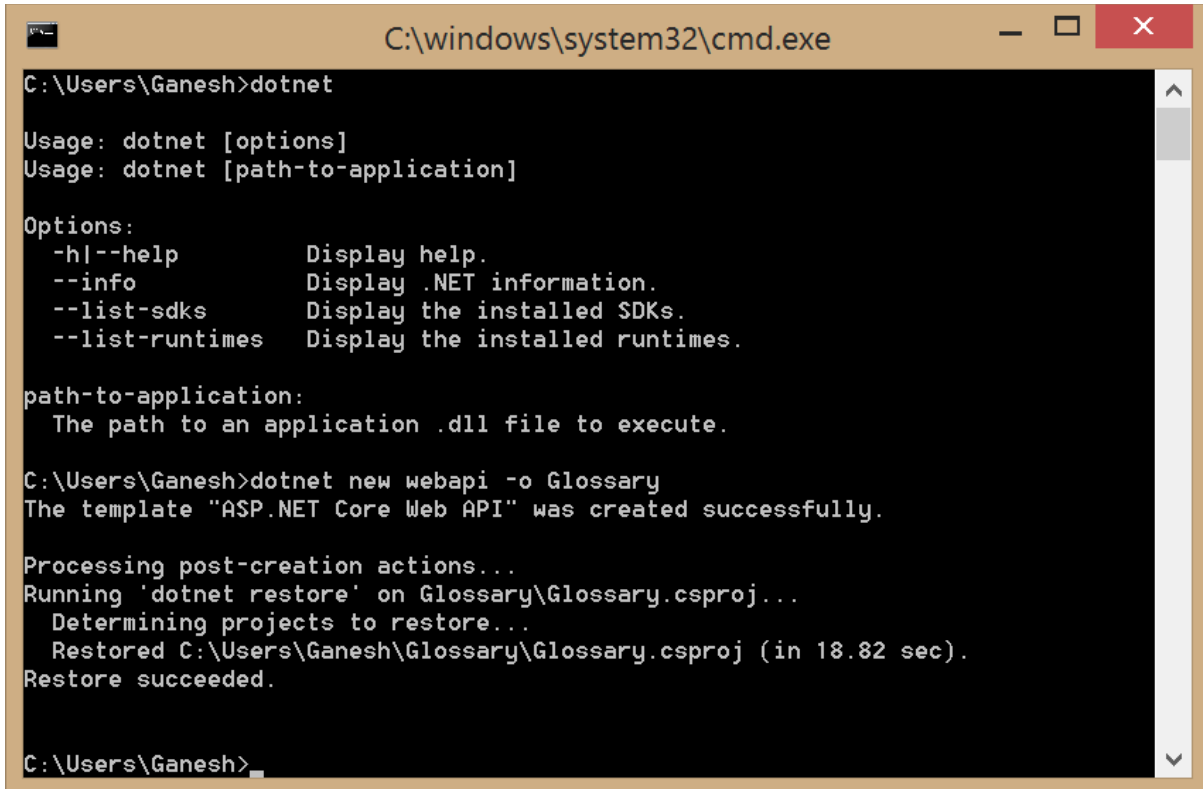
1. Open two command prompts

Command prompt 1:

Command:

dotnet new webapi -o Glossary

output:



```
C:\windows\system32\cmd.exe

C:\Users\Ganesh>dotnet

Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
  -h|--help           Display help.
  --info              Display .NET information.
  --list-sdks         Display the installed SDKs.
  --list-runtimes     Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.

C:\Users\Ganesh>dotnet new webapi -o Glossary
The template "ASP.NET Core Web API" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on Glossary\Glossary.csproj...
  Determining projects to restore...
  Restored C:\Users\Ganesh\Glossary\Glossary.csproj (in 18.82 sec).
Restore succeeded.

C:\Users\Ganesh>
```

Command:

cd Glossary

dotnet run

Output:

The screenshot shows a Windows Command Prompt window titled "C:\windows\system32\cmd.exe - dotnet run". The output shows the process of restoring and running a .NET project named "Glossary".

```

Processing post-creation actions...
Running 'dotnet restore' on Glossary\Glossary.csproj...
  Determining projects to restore...
  Restored C:\Users\Ganesh\Glossary\Glossary.csproj (in 18.82 sec).
Restore succeeded.

C:\Users\Ganesh>cd Glossary

C:\Users\Ganesh\Glossary>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Ganesh\Glossary
  
```

Below the command prompt, a File Explorer window shows the contents of the "Glossary" directory. The files and folders listed are:

Name	Date modified	Type	Size
bin	09-07-2021 07:29	File folder	
Controllers	09-07-2021 07:26	File folder	
obj	09-07-2021 07:29	File folder	
Properties	09-07-2021 07:26	File folder	
appsettings.Development	09-07-2021 07:26	JSON File	1 KB
appsettings	09-07-2021 07:26	JSON File	1 KB
Glossary	09-07-2021 07:26	CSPROJ File	1 KB
Program.cs	09-07-2021 07:26	C# Source File	1 KB
Startup.cs	09-07-2021 07:26	C# Source File	2 KB
WeatherForecast.cs	09-07-2021 07:26	C# Source File	1 KB

Command Prompt 2: (try running readymade weatherforecast class for testing)

Command:

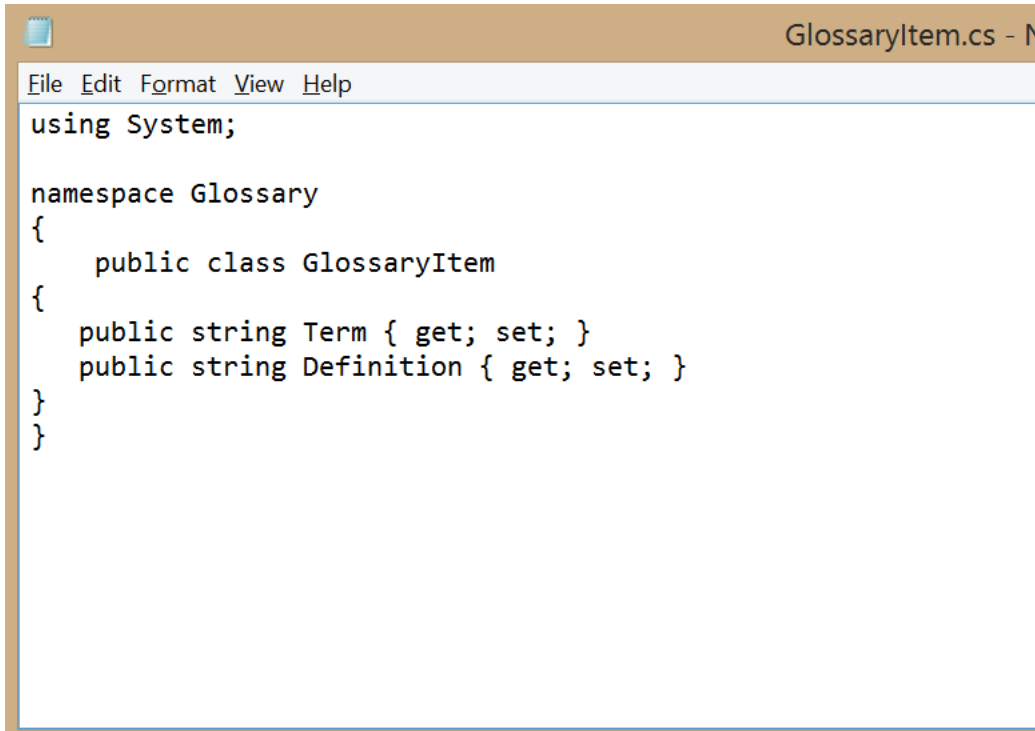
curl --insecure <https://localhost:5001/weatherforecast>

output:

To get started, remove the WeatherForecast.cs file from the root of the project and the WeatherForecastController.cs file from the Controllers folder.

Add Following two files

```
//GlossaryItem.cs
namespace Glossary
{
    public class GlossaryItem
    {
        public string Term { get; set; }
        public string Definition { get; set; }
    }
}
```



```

using System;

namespace Glossary
{
    public class GlossaryItem
    {
        public string Term { get; set; }
        public string Definition { get; set; }
    }
}

```

D:\Glossary\Controllers\ GlossaryController.cs (type it in notepad and save as all files)

//Controllers/GlossaryController.cs

```

using System;
using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc;
using System.IO;
namespace Glossary.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class GlossaryController: ControllerBase
    {
        private static List<GlossaryItem> Glossary = new List<GlossaryItem> {
            new GlossaryItem
            {
                Term= "HTML",
                Definition = "Hypertext Markup Language"
            },
            new GlossaryItem
            {
                Term= "MVC",
                Definition = "Model View Controller"
            },
            new GlossaryItem
            {
                Term= "OpenID",
                Definition = "An open standard for authentication"
            }
        }
    }
}

```

```

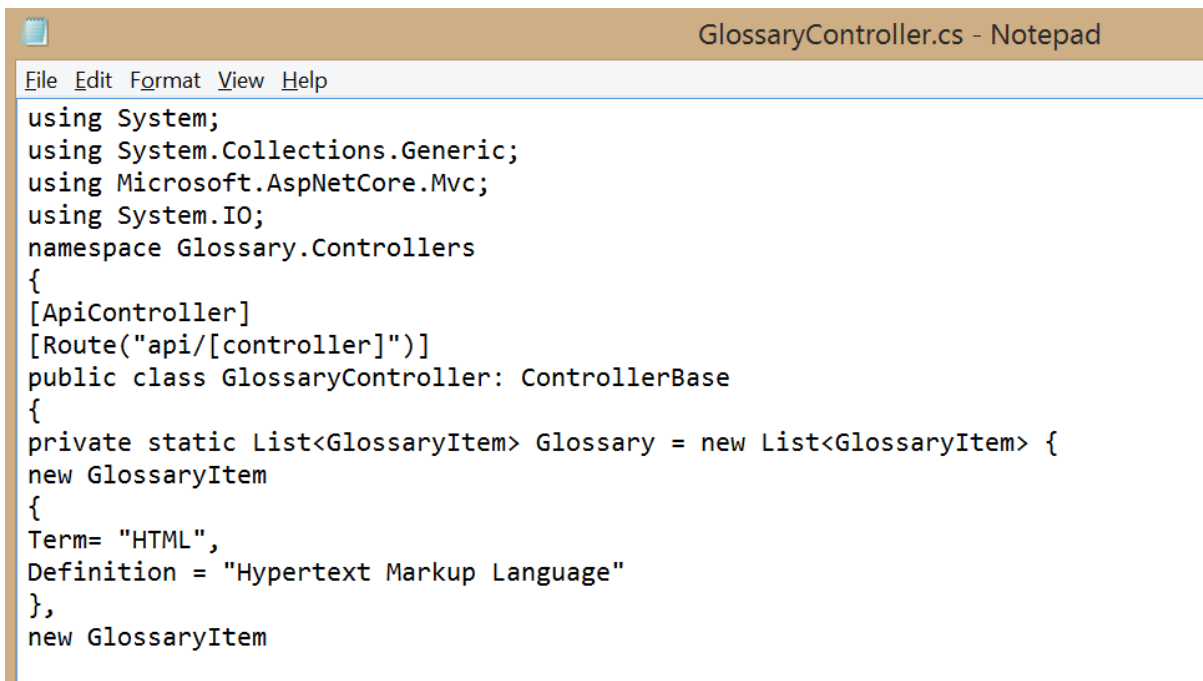
};
[HttpGet]
public ActionResult<List<GlossaryItem>> Get()
{ return Ok(Glossary);
}
[HttpGet]
[Route("{term}")]
public ActionResult<GlossaryItem> Get(string term)
{
    var glossaryItem = Glossary.Find(item =>
        item.Term.Equals(term, StringComparison.InvariantCultureIgnoreCase));
    if (glossaryItem == null)
    { return NotFound();
    } else
    {
        return Ok(glossaryItem);
    }
}
[HttpPost]
public ActionResult Post(GlossaryItem glossaryItem)
{
    var existingGlossaryItem = Glossary.Find(item =>
        item.Term.Equals(glossaryItem.Term, StringComparison.InvariantCultureIgnoreCase));
    if (existingGlossaryItem != null)
    {
        return Conflict("Cannot create the term because it already exists.");
    }
    else
    {
        Glossary.Add(glossaryItem);
        var resourceUrl = Path.Combine(Request.Path.ToString(), Uri.EscapeUriString(glossaryItem.Term));
        return Created(resourceUrl, glossaryItem);
    }
}
[HttpPut]
public ActionResult Put(GlossaryItem glossaryItem)
{
    var existingGlossaryItem = Glossary.Find(item =>
        item.Term.Equals(glossaryItem.Term, StringComparison.InvariantCultureIgnoreCase));
    if (existingGlossaryItem == null)
    {
        return BadRequest("Cannot update a nont existing term.");
    } else
    {
        existingGlossaryItem.Definition = glossaryItem.Definition;
        return Ok();
    }
}
[HttpDelete]
[Route("{term}")]
public ActionResult Delete(string term)

```

```

{
var glossaryItem = Glossary.Find(item =>
item.Term.Equals(term, StringComparison.InvariantCultureIgnoreCase));
if (glossaryItem == null)
{ return NotFound();
}
else
{ Glossary.Remove(glossaryItem);
return NoContent();
}
}
}
}
}

```



```

GlossaryController.cs - Notepad
File Edit Format View Help
using System;
using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc;
using System.IO;
namespace Glossary.Controllers
{
[ApiController]
[Route("api/[controller]")]
public class GlossaryController: ControllerBase
{
private static List<GlossaryItem> Glossary = new List<GlossaryItem> {
new GlossaryItem
{
Term= "HTML",
Definition = "Hypertext Markup Language"
},
new GlossaryItem

```

Output:

Name	Date modified	Type	Size
bin	09-07-2021 07:29	File folder	
Controllers	09-07-2021 09:20	File folder	
obj	09-07-2021 07:29	File folder	
Properties	09-07-2021 07:26	File folder	
appsettings.Development	09-07-2021 07:26	JSON File	1 KB
appsettings	09-07-2021 07:26	JSON File	1 KB
Glossary	09-07-2021 07:26	CSPROJ File	1 KB
GlossaryItem.cs	09-07-2021 09:19	C# Source File	1 KB
Program.cs	09-07-2021 07:26	C# Source File	1 KB
Startup.cs	09-07-2021 07:26	C# Source File	2 KB

Name	Date modified	Type
 GlossaryController.cs	09-07-2021 09:22	C# Source File

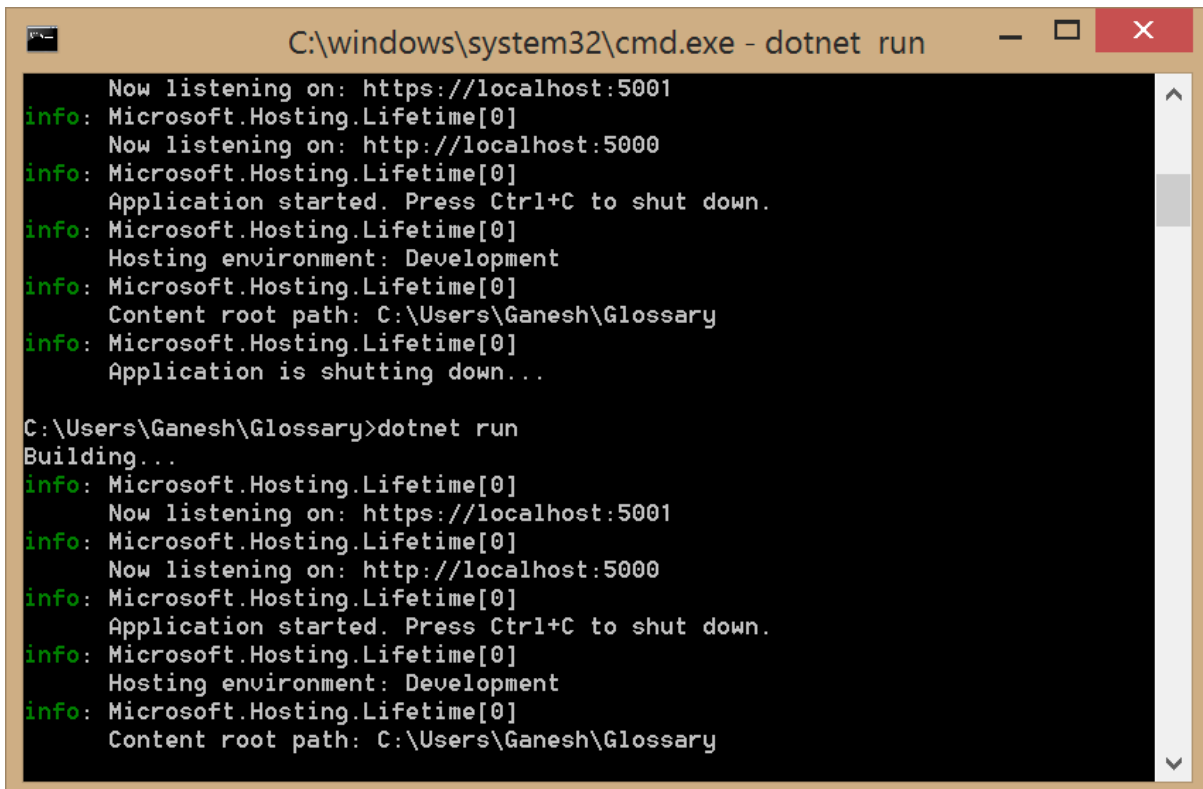
Now stop running previous dotnet run on command prompt 1 using Ctrl+C. and Run it again for new code.

On Command prompt1:

Command:

dotnet run

output:



```

C:\windows\system32\cmd.exe - dotnet run

Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
Content root path: C:\Users\Ganesh\Glossary
info: Microsoft.Hosting.Lifetime[0]
Application is shutting down...

C:\Users\Ganesh\Glossary>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[0]
Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
Content root path: C:\Users\Ganesh\Glossary

```

On Command prompt2:

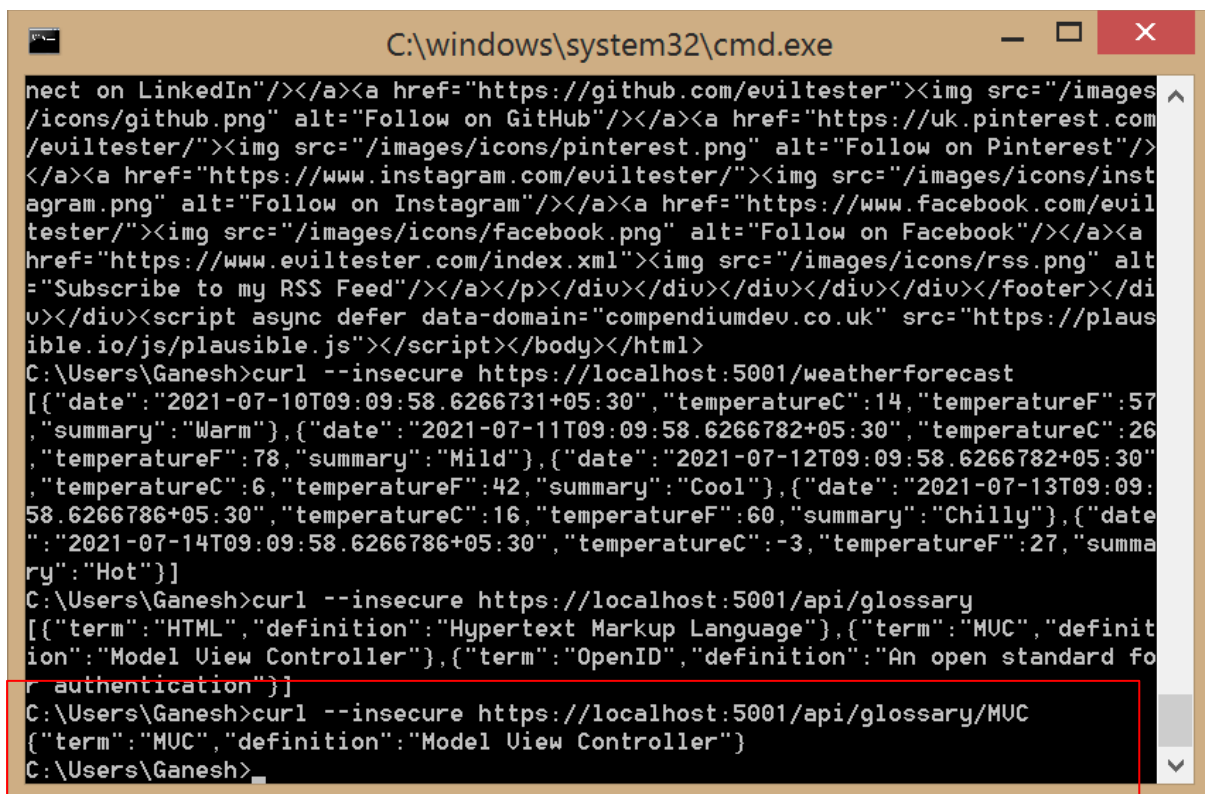
1) Getting a list of items:

Command:

curl --insecure <https://localhost:5001/api/glossary>

Command:

Output:



15

Command:

curl --insecure -X POST -d '{"term": "MFA", "definition": "An authentication process."}' -H "Content-Type:application/json" <https://localhost:5001/api/glossary>

```

C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MUC", "definition": "Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}]
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary/MUC
{"term": "MUC", "definition": "Model View Controller"}
C:\Users\Ganesh>curl --insecure -X POST -d '{"term": "MFA", "definition": "An authentication process."}' -H "Content-Type:application/json" https://localhost:5001/api/glossary
curl: no URL specified!
curl: try 'curl --help' or 'curl --manual' for more information

C:\Users\Ganesh>curl --insecure -X POST -d '{"term": "MFA", "definition": "An authentication process."}' -H "Content-Type:application/json" https://localhost:5001/api/glossary
{"term": "MFA", "definition": "An authentication process."}
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MUC", "definition": "Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}, {"term": "MFA", "definition": "An authentication process."}]
C:\Users\Ganesh>

```

Update Item

Command:

curl --insecure -X PUT -d '{"term": "MVC", "definition": "Modified record of Model View Controller."}' -H "Content-Type:application/json" https://localhost:5001/api/glossary

Output:


```

C:\windows\system32\cmd.exe

curl: try 'curl --help' or 'curl --manual' for more information

C:\Users\Ganesh>curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-Type:application/json" https://localhost:5001/api/glossary
curl: no URL specified!
curl: try 'curl --help' or 'curl --manual' for more information

C:\Users\Ganesh>curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-Type:application/json" https://localhost:5001/api/glossary
{"term": "MFA", "definition": "An authentication process."}
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MUC", "definition": "Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}, {"term": "MFA", "definition": "An authentication process."}]
C:\Users\Ganesh>curl --insecure -X PUT -d "{\"term\": \"MUC\", \"definition\": \"Modified record of Model View Controller.\"}" -H "Content-Type:application/json" https://localhost:5001/api/glossary
{"term": "MUC", "definition": "Modified record of Model View Controller."}
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MUC", "definition": "Modified record of Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}, {"term": "MFA", "definition": "An authentication process."}]
C:\Users\Ganesh>

```

Delete Item

Command:

curl --insecure --request DELETE --url <https://localhost:5001/api/glossary/openid>

Output:

```

C:\windows\system32\cmd.exe

C:\Users\Ganesh>curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-Type:application/json" https://localhost:5001/api/glossary
{"term": "MFA", "definition": "An authentication process."}
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MUC", "definition": "Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}, {"term": "MFA", "definition": "An authentication process."}]
C:\Users\Ganesh>curl --insecure -X PUT -d "{\"term\": \"MUC\", \"definition\": \"Modified record of Model View Controller.\"}" -H "Content-Type:application/json" https://localhost:5001/api/glossary
{"term": "MUC", "definition": "Modified record of Model View Controller."}
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MUC", "definition": "Modified record of Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}, {"term": "MFA", "definition": "An authentication process."}]
C:\Users\Ganesh>curl --insecure --request DELETE --url https://localhost:5001/api/glossary/openid
{"term": "MUC", "definition": "Modified record of Model View Controller."}
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MUC", "definition": "Modified record of Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}, {"term": "MFA", "definition": "An authentication process."}]
C:\Users\Ganesh>

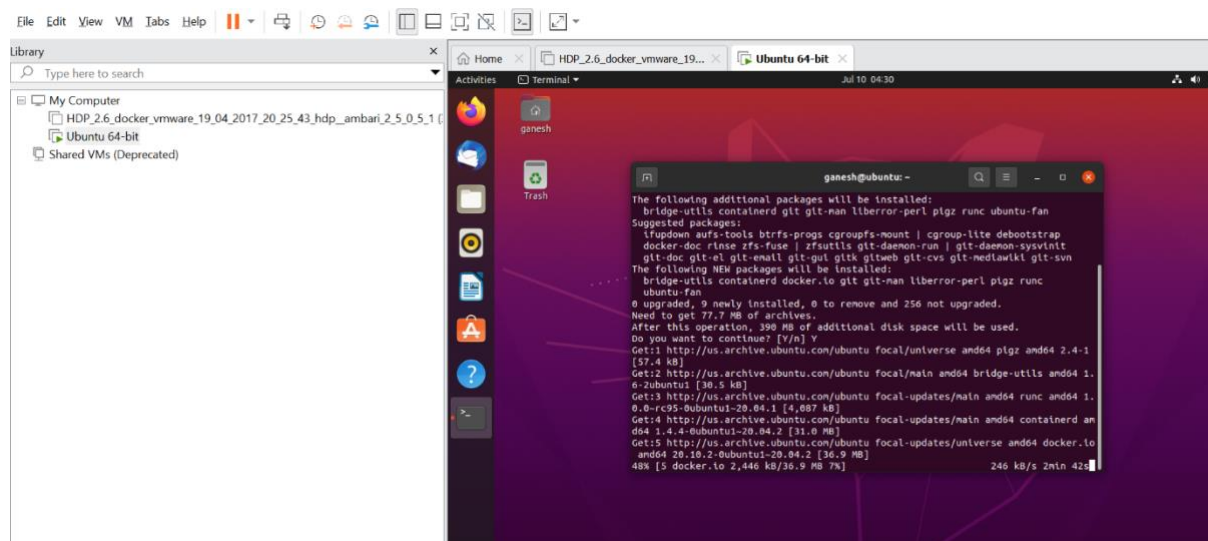
```

Practical No. 3

Aim: Working with Docker, Docker Commands, Docker Images and Containers

After install ubuntu in vmware. Install docker

Command: `sudo apt-get install docker.io`



Install using the repository

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

Set up the repository

Update the apt package index and install packages to allow apt to use a repository over HTTPS:

1. `$ sudo apt-get update`
2. `$ sudo apt-get install \`
`apt-transport-https \`
`ca-certificates \`
`curl \`
`gnupg \`
`lsb-release`

```

ganesh@ubuntu: ~
Experimental: true
Get http://var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version: dial unix /var/run/docker.sock: connect: permission denied
ganesh@ubuntu:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 https://download.docker.com/linux/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 214 kB in 4s (48.8 kB/s)
Reading package lists... Done
ganesh@ubuntu:~$ sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg \
  lsb-release
Reading package lists... Done
Building dependency tree
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu2).
lsb-release set to manually installed.
ca-certificates is already the newest version (20210119~20.04.1)

```

1. Add Docker's official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

Use the following command to set up the **stable** repository

```
$ echo \
  "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```

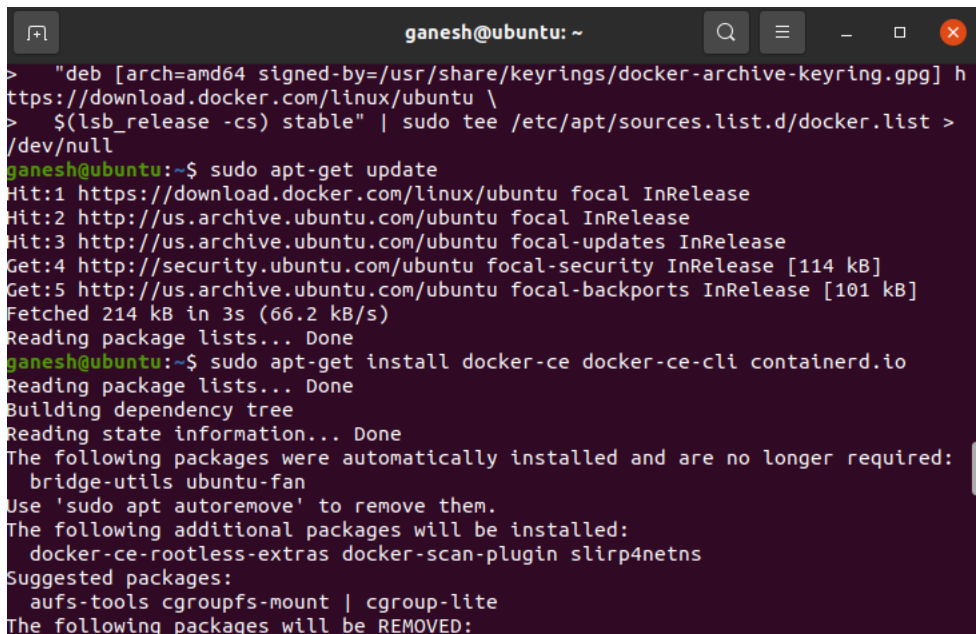
ganesh@ubuntu: ~
Setting up gnupg-utils (2.2.19-3ubuntu2.1) ...
Setting up gpg-agent (2.2.19-3ubuntu2.1) ...
Setting up gpgsm (2.2.19-3ubuntu2.1) ...
Setting up dirmngr (2.2.19-3ubuntu2.1) ...
Setting up gpg-wks-server (2.2.19-3ubuntu2.1) ...
Setting up gpg-wks-client (2.2.19-3ubuntu2.1) ...
Setting up gnupg (2.2.19-3ubuntu2.1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
ganesh@ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
File '/usr/share/keyrings/docker-archive-keyring.gpg' exists. Overwrite? (y/N) Y
ganesh@ubuntu:~$ echo \
> "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
ganesh@ubuntu:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Fetched 214 kB in 3s (66.2 kB/s)

```

Install Docker Engine

Update the apt package index, and install the *latest version* of Docker Engine and containerd, or go to the next step to install a specific version:

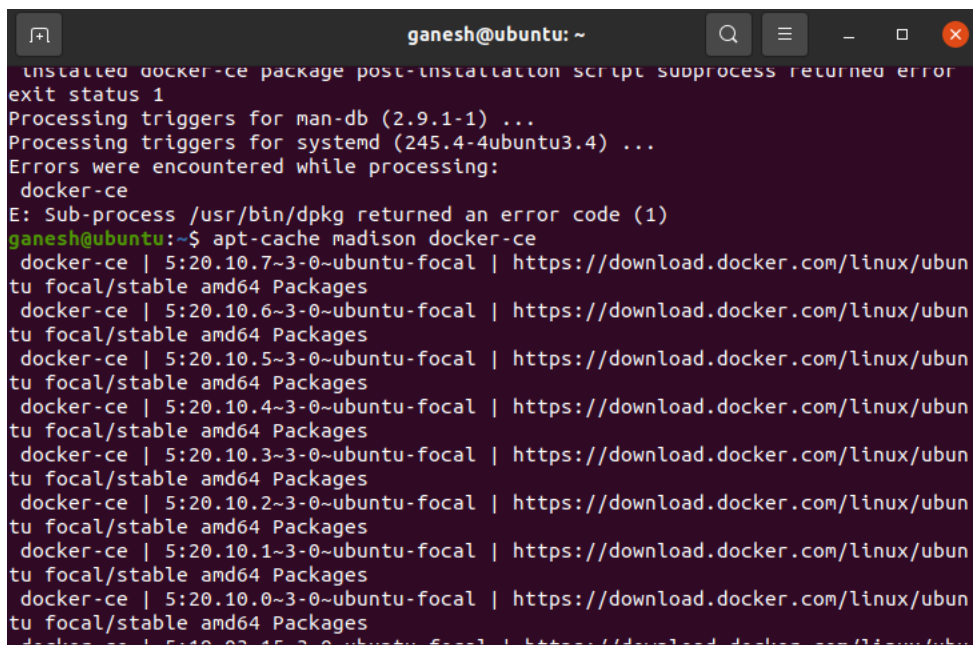
```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```



```
ganesh@ubuntu: ~
> "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
ganesh@ubuntu:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Fetched 214 kB in 3s (66.2 kB/s)
Reading package lists... Done
ganesh@ubuntu:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bridge-utils ubuntu-fan
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following packages will be REMOVED:
```

To install a *specific version* of Docker Engine, list the available versions in the repo, then select and install:

```
apt-cache madison docker-ce
```



```
ganesh@ubuntu: ~
installed docker-ce package post-installation script subprocess returned error
exit status 1
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.4) ...
Errors were encountered while processing:
 docker-ce
E: Sub-process /usr/bin/dpkg returned an error code (1)
ganesh@ubuntu:~$ apt-cache madison docker-ce
docker-ce | 5:20.10.7-3-0~ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
docker-ce | 5:20.10.6-3-0~ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
docker-ce | 5:20.10.5-3-0~ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
docker-ce | 5:20.10.4-3-0~ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
docker-ce | 5:20.10.3-3-0~ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
docker-ce | 5:20.10.2-3-0~ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
docker-ce | 5:20.10.1-3-0~ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
docker-ce | 5:20.10.0-3-0~ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
```

Docker Commands:

Docker --version

Docker version

```

ganesh@ubuntu: ~
E: Package 'docker-ce' has no installation candidate
E: Unable to locate package docker-ce-cli
E: Unable to locate package containerd.io
E: Couldn't find any package by glob 'containerd.io'
E: Couldn't find any package by regex 'containerd.io'
ganesh@ubuntu:~$ docker --version
Docker version 20.10.2, build 20.10.2-0ubuntu1~20.04.2
ganesh@ubuntu:~$ docker version
Client:
 Version:      20.10.2
 API version:  1.41
 Go version:   go1.13.8
 Git commit:   20.10.2-0ubuntu1~20.04.2
 Built:        Tue Mar 30 21:24:57 2021
 OS/Arch:      linux/amd64
 Context:      default
 Experimental: true
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version: dial unix /var/run/docker.sock: connect: permission denied
ganesh@ubuntu:~$ docker images
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.24/images/json: dial unix /var/run/docker.sock: connect: permission denied

```

Docker pull httpd

Pull an image or a repository from a registry

```

ganesh@ubuntu: ~
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/

ganesh@ubuntu:~$ docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
b4d181a07f80: Pull complete
4b72f5187e6e: Pull complete
12b2c44d04b2: Pull complete
35c238b46d30: Pull complete
1adcec05f52b: Pull complete
Digest: sha256:1fd07d599a519b594b756d2e4e43a72edf7e30542ce646f5eb3328cf3b12341a
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
ganesh@ubuntu:~$

```

Docker images

It lists all the images

```

ganesh@ubuntu: ~
$ 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/

ganesh@ubuntu:~$ docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
b4d181a07f80: Pull complete
4b72f5187e6e: Pull complete
12b2c44d04b2: Pull complete
35c238b46d30: Pull complete
1adcec05f52b: Pull complete
Digest: sha256:1fd07d599a519b594b756d2e4e43a72edf7e30542ce646f5eb3328cf3b12341a
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
ganesh@ubuntu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
httpd          latest    bd29370f84ea   38 hours ago   138MB
hello-world    latest    d1165f221234   4 months ago   13.3kB
ganesh@ubuntu:~$

```

#nano Dockerfile

FROM busybox

CMD echo "Hello world! This is my first Docker image."

//above two line we have to add into dockerfile

to save press ctrl+o(to write) then enter then ctrl+x (to exit)

docker build --tag "hello-world:pract1" .

docker images

```

ganesh@ubuntu: ~
invalid argument "Dockerfile:pract1" for "-t, --tag" flag: invalid reference for
mat: repository name must be lowercase
See 'docker build --help'.
ganesh@ubuntu:~$ docker build --tag "hello-world:pract1" .
Sending build context to Docker daemon  10.36MB
Step 1/2 : FROM busybox
latest: Pulling from library/busybox
b71f96345d44: Pull complete
Digest: sha256:930490f97e5b921535c153e0e7110d251134cc4b72bbb8133c6a5065cc68580d
Status: Downloaded newer image for busybox:latest
--> 69593048aa3a
Step 2/2 : CMD echo "Hello world! This is my first Docker image."
--> Running in f7b326450d64
Removing intermediate container f7b326450d64
--> 77ded695389b
Successfully built 77ded695389b
Successfully tagged hello-world:pract1
ganesh@ubuntu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world    pract1     77ded695389b   3 minutes ago   1.24MB
httpd          latest    bd29370f84ea   38 hours ago   138MB
busybox        latest    69593048aa3a   4 weeks ago    1.24MB
hello-world    latest    d1165f221234   4 months ago   13.3kB
ganesh@ubuntu:~$

```

docker run hello-world:pract1


```

ganesh@ubuntu: ~
See 'docker build --help'.
ganesh@ubuntu:~$ docker build --tag "hello-world:pract1" .
Sending build context to Docker daemon 10.36MB
Step 1/2 : FROM busybox
latest: Pulling from library/busybox
b71f96345d44: Pull complete
Digest: sha256:930490f97e5b921535c153e0e7110d251134cc4b72bbb8133c6a5065cc68580d
Status: Downloaded newer image for busybox:latest
--> 69593048aa3a
Step 2/2 : CMD echo "Hello world! This is my first Docker image."
--> Running in f7b326450d64
Removing intermediate container f7b326450d64
--> 77ded695389b
Successfully built 77ded695389b
Successfully tagged hello-world:pract1
ganesh@ubuntu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   pract1    77ded695389b   3 minutes ago 1.24MB
httpd         latest   bd29370f84ea   38 hours ago  138MB
busybox       latest   69593048aa3a   4 weeks ago   1.24MB
hello-world   latest   d1165f221234   4 months ago  13.3kB
ganesh@ubuntu:~$ docker run hello-world:pract1
Hello world! This is my first Docker image.
ganesh@ubuntu:~$

```

docker run 77ded695389b

```

ganesh@ubuntu: ~
Sending build context to Docker daemon 10.36MB
Step 1/2 : FROM busybox
latest: Pulling from library/busybox
b71f96345d44: Pull complete
Digest: sha256:930490f97e5b921535c153e0e7110d251134cc4b72bbb8133c6a5065cc68580d
Status: Downloaded newer image for busybox:latest
--> 69593048aa3a
Step 2/2 : CMD echo "Hello world! This is my first Docker image."
--> Running in f7b326450d64
Removing intermediate container f7b326450d64
--> 77ded695389b
Successfully built 77ded695389b
Successfully tagged hello-world:pract1
ganesh@ubuntu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   pract1    77ded695389b   3 minutes ago 1.24MB
httpd         latest   bd29370f84ea   38 hours ago  138MB
busybox       latest   69593048aa3a   4 weeks ago   1.24MB
hello-world   latest   d1165f221234   4 months ago  13.3kB
ganesh@ubuntu:~$ docker run hello-world:pract1
Hello world! This is my first Docker image.
ganesh@ubuntu:~$ docker run 77ded695389b
Hello world! This is my first Docker image.
ganesh@ubuntu:~$

```

Docker rmi

Remove one or more images

docker rmi -f images-id

docker rmi -f 77ded695389b

After running docker images we can see that 77ded695389b is deleted.

```

ganesh@ubuntu: ~
Package docker-ce is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'docker-ce' has no installation candidate
E: Unable to locate package docker-ce-cli
E: Unable to locate package containerd.io
E: Couldn't find any package by glob 'containerd.io'
E: Couldn't find any package by regex 'containerd.io'
ganesh@ubuntu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   pract1    77ded695389b   13 minutes ago 1.24MB
httpd         latest    bd29370f84ea   38 hours ago   138MB
busybox       latest    69593048aa3a   4 weeks ago    1.24MB
hello-world   latest    d1165f221234   4 months ago   13.3kB
ganesh@ubuntu:~$ docker rmi -f 77ded695389b
Untagged: hello-world:pract1
Deleted: sha256:77ded695389bb5259ef5cdbc14d8a606904fb213e65506dfcbcd76feda73c417
ganesh@ubuntu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
httpd         latest    bd29370f84ea   38 hours ago   138MB
busybox       latest    69593048aa3a   4 weeks ago    1.24MB
hello-world   latest    d1165f221234   4 months ago   13.3kB
ganesh@ubuntu:~$

```

docker rmi -f Respository-name

docker rmi -f Debian

```

ganesh@ubuntu: ~
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
httpd         latest    bd29370f84ea   38 hours ago   138MB
busybox       latest    69593048aa3a   4 weeks ago    1.24MB
hello-world   latest    d1165f221234   4 months ago   13.3kB
ganesh@ubuntu:~$ docker pull debian
Using default tag: latest
latest: Pulling from library/debian
0bc3020d05f1: Pull complete
Digest: sha256:dc320da8d9d73c9dab5059668852555c171d40cdec297da845da9c929b70e0b1
Status: Downloaded newer image for debian:latest
docker.io/library/debian:latest
ganesh@ubuntu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
httpd         latest    bd29370f84ea   38 hours ago   138MB
debian        latest    7a4951775d15   2 weeks ago    114MB
busybox       latest    69593048aa3a   4 weeks ago    1.24MB
hello-world   latest    d1165f221234   4 months ago   13.3kB
ganesh@ubuntu:~$ docker rmi -f debian
Untagged: debian:latest
Untagged: debian@sha256:dc320da8d9d73c9dab5059668852555c171d40cdec297da845da9c929b70e0b1
Deleted: sha256:7a4951775d157843b47250a2a5cc7b561d2abe0b29ae6f19737a04635302eacf
Deleted: sha256:4e006334a6fdea37622f72b21eb75fe1484fc4f20ce8b8526187d6f7bd90a6fe
ganesh@ubuntu:~$

```

docker rmi -f Respository-name:tag

docker rmi -f debian:latest

After this debain image will be deleted


```
ganesh@ubuntu: ~  
Using default tag: latest  
latest: Pulling from library/debian  
0bc3020d05f1: Pull complete  
Digest: sha256:dcb20da8d9d73c9dab5059668852555c171d40cdec297da845da9c929b70e0b1  
Status: Downloaded newer image for debian:latest  
docker.io/library/debian:latest  
ganesh@ubuntu:~$ docker images  
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE  
httpd          latest    bd29370f84ea   38 hours ago   138MB  
debian         latest    7a4951775d15   2 weeks ago    114MB  
busybox        latest    69593048aa3a   4 weeks ago    1.24MB  
hello-world    latest    d1165f221234   4 months ago   13.3kB  
ganesh@ubuntu:~$ docker rmi -f debian:latest  
Untagged: debian:latest  
Untagged: debian@sha256:dcb20da8d9d73c9dab5059668852555c171d40cdec297da845da9c929b70e0b1  
Deleted: sha256:7a4951775d157843b47250a2a5cc7b561d2abe0b29ae6f19737a04635302eacf  
Deleted: sha256:4e006334a6fdea37622f72b21eb75fe1484fc4f20ce8b8526187d6f7bd90a6fe  
ganesh@ubuntu:~$ docker images  
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE  
httpd          latest    bd29370f84ea   38 hours ago   138MB  
busybox        latest    69593048aa3a   4 weeks ago    1.24MB  
hello-world    latest    d1165f221234   4 months ago   13.3kB  
ganesh@ubuntu:~$
```

Practical No. 4

Aim: Installing software packages on Docker, Working with Docker Volumes and Networks.

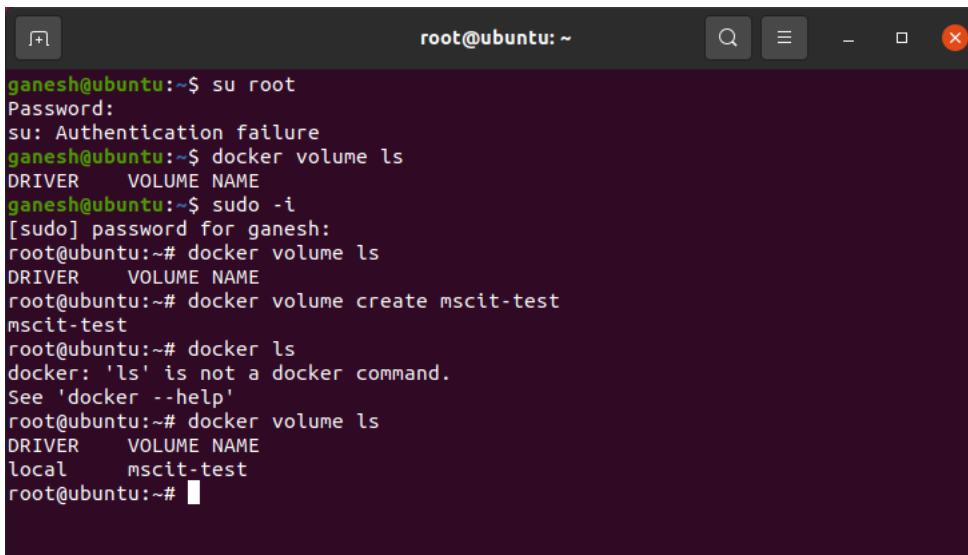
Volumes are the preferred mechanism for persisting data generated by and used by Docker containers. While bind mounts are dependent on the directory structure and OS of the host machine, volumes are completely managed by Docker.

List volumes created

Command: `docker volume ls`

To create volume.

Command: `docker volume create mscit-test`

A terminal window titled 'root@ubuntu: ~' with search, menu, and window control icons. The terminal shows a user 'ganesh' at 'ubuntu' prompt. They attempt to switch to root with 'su root' but fail due to authentication. Then they run 'docker volume ls' which shows no volumes. They then run 'sudo -i' to become root. As root, they run 'docker volume ls' again (no output), then 'docker volume create mscit-test' (output: 'mscit-test'), then 'docker ls' (error: 'ls' is not a docker command), then 'docker volume ls' (output: a table with one row: 'local' driver, 'mscit-test' name).

```
ganesh@ubuntu:~$ su root
Password:
su: Authentication failure
ganesh@ubuntu:~$ docker volume ls
DRIVER      VOLUME NAME
ganesh@ubuntu:~$ sudo -i
[sudo] password for ganesh:
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
root@ubuntu:~# docker volume create mscit-test
mscit-test
root@ubuntu:~# docker ls
docker: 'ls' is not a docker command.
See 'docker --help'
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
local       mscit-test
root@ubuntu:~#
```

Return low-level information on Docker objects

Command: `docker volume inspect mscit-test`

```

root@ubuntu: ~
[sudo] password for ganesh:
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
root@ubuntu:~# docker volume create mscit-test
mscit-test
root@ubuntu:~# docker ls
docker: 'ls' is not a docker command.
See 'docker --help'
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
local       mscit-test
root@ubuntu:~# docker volume inspect mscit-test
[
  {
    "CreatedAt": "2021-07-10T20:58:04-07:00",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/mscit-test/_data",
    "Name": "mscit-test",
    "Options": {},
    "Scope": "local"
  }
]
root@ubuntu:~#

```

Create a directory

`mkdir mscit-volume`

Now, change directory to mscit-volume

`cd mscit-volume/`

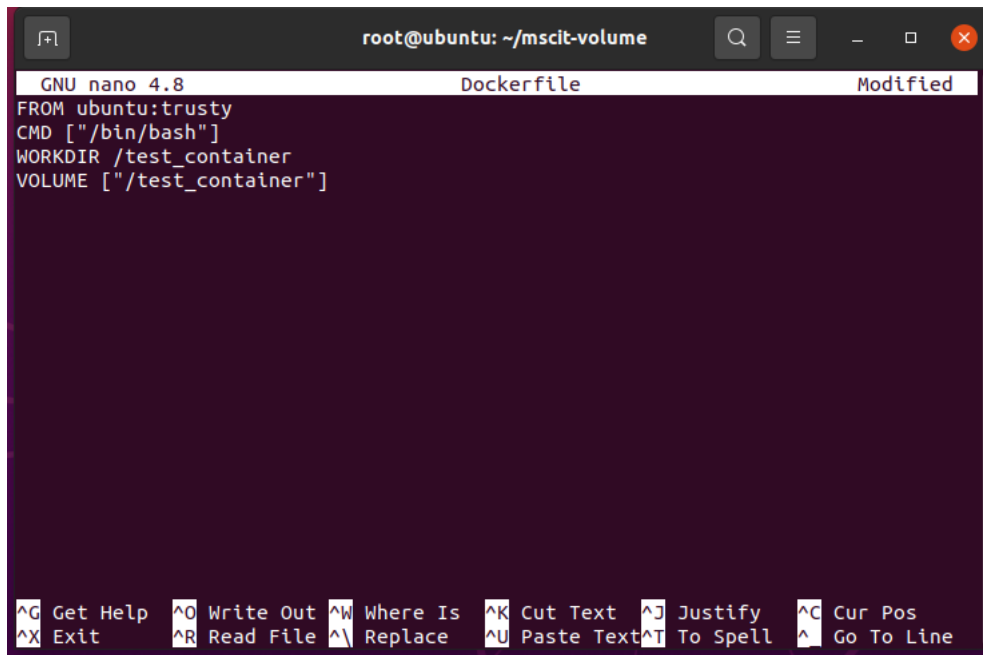
```

root@ubuntu: ~/mscit-volume
root@ubuntu:~# docker ls
docker: 'ls' is not a docker command.
See 'docker --help'
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
local       mscit-test
root@ubuntu:~# docker volume inspect mscit-test
[
  {
    "CreatedAt": "2021-07-10T20:58:04-07:00",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/mscit-test/_data",
    "Name": "mscit-test",
    "Options": {},
    "Scope": "local"
  }
]
root@ubuntu:~# ls
snap
root@ubuntu:~# mkdir mscit-volume
root@ubuntu:~# cd mscit-volume/
root@ubuntu:~/mscit-volume# nano Dockerfile

```

Create a file

Nano Dockerfile



```

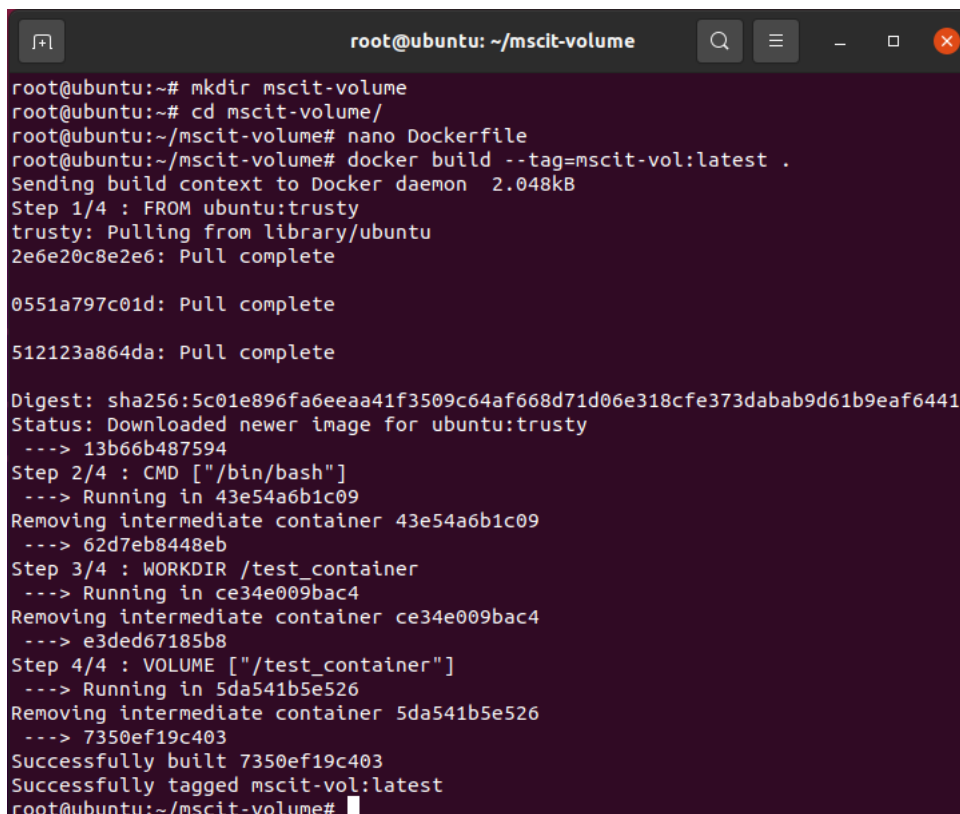
root@ubuntu: ~/mscit-volume
GNU nano 4.8 Dockerfile Modified
FROM ubuntu:trusty
CMD ["/bin/bash"]
WORKDIR /test_container
VOLUME ["/test_container"]

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line

```

To create an image file

`docker build --tag=mscit-vol:latest .`



```

root@ubuntu: ~/mscit-volume
root@ubuntu:~# mkdir mscit-volume
root@ubuntu:~# cd mscit-volume/
root@ubuntu:~/mscit-volume# nano Dockerfile
root@ubuntu:~/mscit-volume# docker build --tag=mscit-vol:latest .
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM ubuntu:trusty
trusty: Pulling from library/ubuntu
2e6e20c8e2e6: Pull complete

0551a797c01d: Pull complete

512123a864da: Pull complete

Digest: sha256:5c01e896fa6eaa41f3509c64af668d71d06e318cfe373dabab9d61b9eaf6441
Status: Downloaded newer image for ubuntu:trusty
--> 13b66b487594
Step 2/4 : CMD ["/bin/bash"]
--> Running in 43e54a6b1c09
Removing intermediate container 43e54a6b1c09
--> 62d7eb8448eb
Step 3/4 : WORKDIR /test_container
--> Running in ce34e009bac4
Removing intermediate container ce34e009bac4
--> e3ded67185b8
Step 4/4 : VOLUME ["/test_container"]
--> Running in 5da541b5e526
Removing intermediate container 5da541b5e526
--> 7350ef19c403
Successfully built 7350ef19c403
Successfully tagged mscit-vol:latest
root@ubuntu:~/mscit-volume#

```

Check the image create

Command: `docker images`

```

root@ubuntu: ~/mscit-volume
2e6e20c8e2e6: Pull complete
0551a797c01d: Pull complete
512123a864da: Pull complete
Digest: sha256:5c01e896fa6eaa41f3509c64af668d71d06e318cfe373dabab9d61b9eaf6441
Status: Downloaded newer image for ubuntu:trusty
---> 13b66b487594
Step 2/4 : CMD ["/bin/bash"]
---> Running in 43e54a6b1c09
Removing intermediate container 43e54a6b1c09
---> 62d7eb8448eb
Step 3/4 : WORKDIR /test_container
---> Running in ce34e009bac4
Removing intermediate container ce34e009bac4
---> e3ded67185b8
Step 4/4 : VOLUME ["/test_container"]
---> Running in 5da541b5e526
Removing intermediate container 5da541b5e526
---> 7350ef19c403
Successfully built 7350ef19c403
Successfully tagged mscit-vol:latest
root@ubuntu:~/mscit-volume# docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mscit-vol	latest	7350ef19c403	3 minutes ago	197MB
httpd	latest	bd29370f84ea	2 days ago	138MB
busybox	latest	69593048aa3a	4 weeks ago	1.24MB
ubuntu	trusty	13b66b487594	3 months ago	197MB
hello-world	latest	d1165f221234	4 months ago	13.3kB

```

root@ubuntu:~/mscit-volume#

```

Mounting the container

`docker run -it --mount src=/mscit-shared,target=/test_container,type=bind mscit-vol`

It will change in root and show test_Container

```

root@b43347ed3af6: /test_container
REPOSITORY TAG IMAGE ID CREATED SIZE
mscit-vol latest 7350ef19c403 26 minutes ago 197MB
httpd latest bd29370f84ea 2 days ago 138MB
busybox latest 69593048aa3a 4 weeks ago 1.24MB
ubuntu trusty 13b66b487594 3 months ago 197MB
hello-world latest d1165f221234 4 months ago 13.3kB
root@ubuntu:~/mscit-volume# cd /mscit-shared
root@ubuntu:/mscit-shared# invalid mount config for type "bind": bind source path does not exist: /mscit-share.
invalid: command not found
root@ubuntu:/mscit-shared# docker run -it --mount src=/mscit-share,target=/test_container,type=bind mscit-vol
docker: Error response from daemon: invalid mount config for type "bind": bind source path does not exist: /mscit-share.
See 'docker run --help'.
root@ubuntu:/mscit-shared# cd /mscit-volume
-bash: cd: /mscit-volume: No such file or directory
root@ubuntu:/mscit-shared# cd..
cd..: command not found
root@ubuntu:/mscit-shared# cd
root@ubuntu:~# cd mscit-volume
root@ubuntu:~/mscit-volume# mkdir /mscit-shared
mkdir: cannot create directory '/mscit-shared': File exists
root@ubuntu:~/mscit-volume# docker run -it --mount src=/mscit-share,target=/test_container,type=bind mscit-vol
docker: Error response from daemon: invalid mount config for type "bind": bind source path does not exist: /mscit-share.
See 'docker run --help'.
root@ubuntu:~/mscit-volume# docker run -it --mount src=/mscit-shared,target=/test_container,type=bind mscit-vol
root@b43347ed3af6:/test_container#

```

Now open other terminal and get into mscit-shared directory and create a file called hi

```
[sudo] password for ganesh:
sudo: i: command not found
ganesh@ubuntu:~$ sudo i
sudo: i: command not found
ganesh@ubuntu:~$ su root
Password:
su: Authentication failure
ganesh@ubuntu:~$ su root
Password:
su: Authentication failure
ganesh@ubuntu:~$ su root
Password:
su: Authentication failure
ganesh@ubuntu:~$ su i
su: user i does not exist
ganesh@ubuntu:~$ sudo -i
root@ubuntu:~# 12345
12345: command not found
root@ubuntu:~# ls /mscit-shared/
root@ubuntu:~# pwd
/root
root@ubuntu:~# cd /mscit-shared/
root@ubuntu:/mscit-shared# ls
root@ubuntu:/mscit-shared# pwd
/mscit-shared
root@ubuntu:/mscit-shared# cat >> hi
hello World
root@ubuntu:/mscit-shared# ls
hi
root@ubuntu:/mscit-shared#
```

Now check the file created in root is listed in test_Container and vice-versa.

```
root@ubuntu:/m th does not exist: /mscit-share.
invalid: command not found
root@ubuntu:/mscit-shared# docker run -it --mount src=/mscit-share,target=/test
_container,type=bind mscit-vol
docker: Error response from daemon: invalid mount config for type "bind": bind
source path does not exist: /mscit-share.
See 'docker run --help'.
root@ubuntu:/mscit-shared# cd /mscit-volume
-bash: cd: /mscit-volume: No such file or directory
root@ubuntu:/mscit-shared# cd..
cd..: command not found
root@ubuntu:/mscit-shared# cd
root@ubuntu:~# cd mscit-volume
root@ubuntu:~/mscit-volume# mkdir /mscit-shared
mkdir: cannot create directory '/mscit-shared': File exists
root@ubuntu:~/mscit-volume# docker run -it --mount src=/mscit-share,target=/tes
t_container,type=bind mscit-vol
docker: Error response from daemon: invalid mount config for type "bind": bind
source path does not exist: /mscit-share.
See 'docker run --help'.
root@ubuntu:~/mscit-volume# docker run -it --mount src=/mscit-shared,target=/te
st_container,type=bind mscit-vol
root@b43347ed3af6:/test_container# ls
root@b43347ed3af6:/test_container# ls
root@b43347ed3af6:/test_container# pwd
/test_container
root@b43347ed3af6:/test_container# ls
hi
root@b43347ed3af6:/test_container# cat hi
hello World
root@b43347ed3af6:/test_container#

root@ubuntu:/mscit-shared# ls
hi
root@ubuntu:/mscit-shared# ls
hi test
root@ubuntu:/mscit-shared#
```

We can see that file location are mapped.

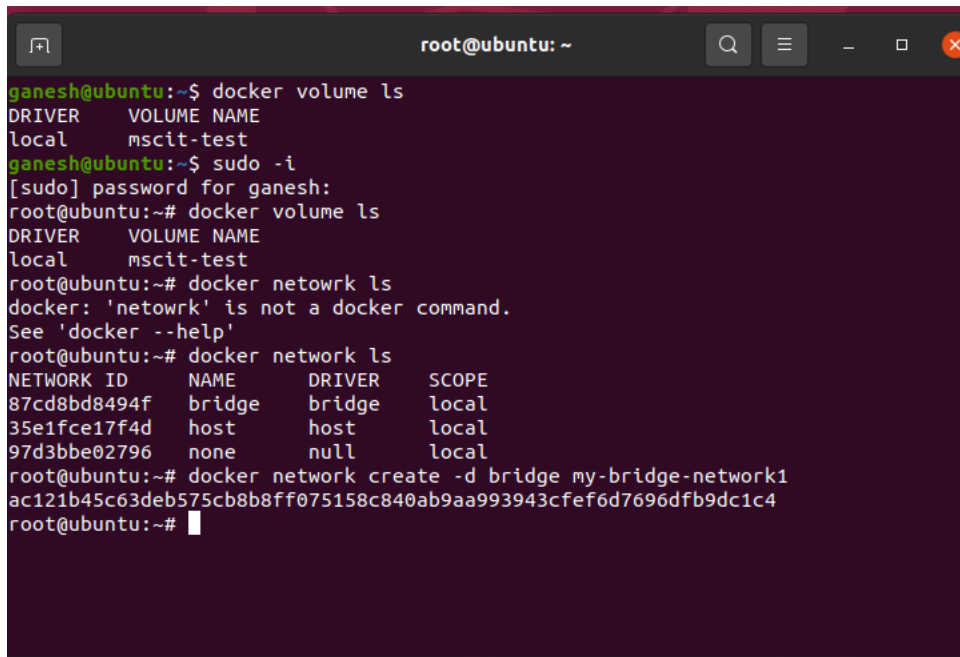
When below command is executed, it will delete the volume.

`docker volume rm mscit-test`

Network:

Create network with following command

`docker network create -d bridge my-bridge-network1`



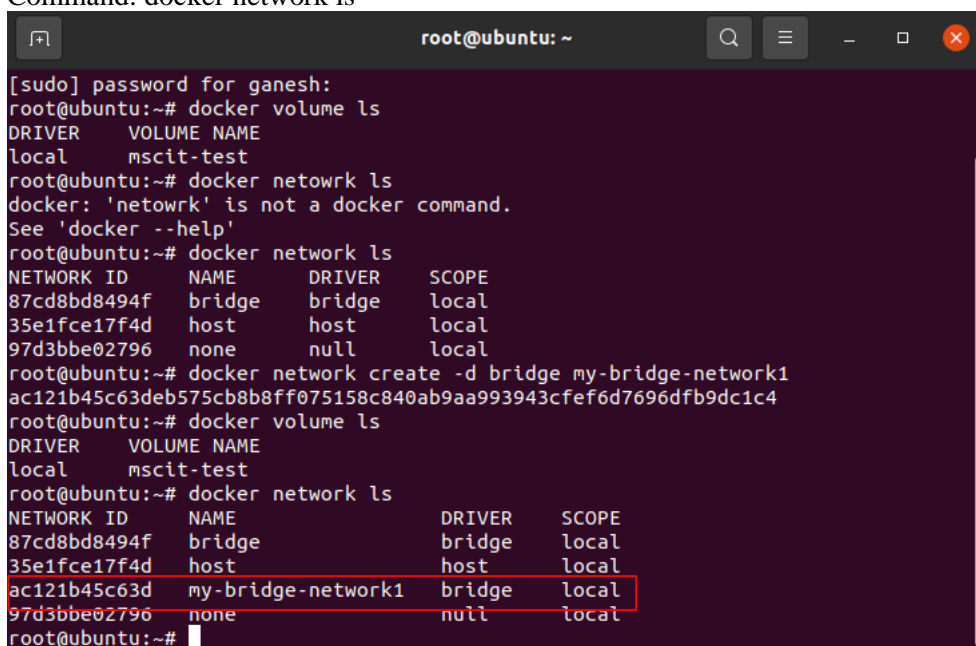
```

ganesh@ubuntu:~$ docker volume ls
DRIVER      VOLUME NAME
local       mscit-test
ganesh@ubuntu:~$ sudo -i
[sudo] password for ganesh:
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
local       mscit-test
root@ubuntu:~# docker netowrk ls
docker: 'netowrk' is not a docker command.
See 'docker --help'
root@ubuntu:~# docker network ls
NETWORK ID    NAME        DRIVER      SCOPE
87cd8bd8494f  bridge     bridge      local
35e1fce17f4d  host       host        local
97d3bbe02796  none       null        local
root@ubuntu:~# docker network create -d bridge my-bridge-network1
ac121b45c63deb575cb8b8ff075158c840ab9aa993943cfef6d7696dfb9dc1c4
root@ubuntu:~#

```

Check network is created with below command

Command: `docker network ls`



```

[sudo] password for ganesh:
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
local       mscit-test
root@ubuntu:~# docker netowrk ls
docker: 'netowrk' is not a docker command.
See 'docker --help'
root@ubuntu:~# docker network ls
NETWORK ID    NAME        DRIVER      SCOPE
87cd8bd8494f  bridge     bridge      local
35e1fce17f4d  host       host        local
97d3bbe02796  none       null        local
root@ubuntu:~# docker network create -d bridge my-bridge-network1
ac121b45c63deb575cb8b8ff075158c840ab9aa993943cfef6d7696dfb9dc1c4
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
local       mscit-test
root@ubuntu:~# docker network ls
NETWORK ID    NAME        DRIVER      SCOPE
87cd8bd8494f  bridge     bridge      local
35e1fce17f4d  host       host        local
ac121b45c63d  my-bridge-network1  bridge      local
97d3bbe02796  none       null        local
root@ubuntu:~#

```

We can inspect the created network with below command
 docker network inspect bridge (network name)

```

root@ubuntu: ~
"EnableIPv6": false,
"IPAM": {
  "Driver": "default",
  "Options": {},
  "Config": [
    {
      "Subnet": "172.18.0.0/16",
      "Gateway": "172.18.0.1"
    }
  ]
},
"Internal": false,
"Attachable": false,
"Ingress": false,
"ConfigFrom": {
  "Network": ""
},
"ConfigOnly": false,
"Containers": {},
"Options": {},
"Labels": {}
}
]
root@ubuntu:~#

```

Now, lets remove the create network using below command.

docker network rm network-name

With docker network ls we can see the my-bridge-network1 is deleted.

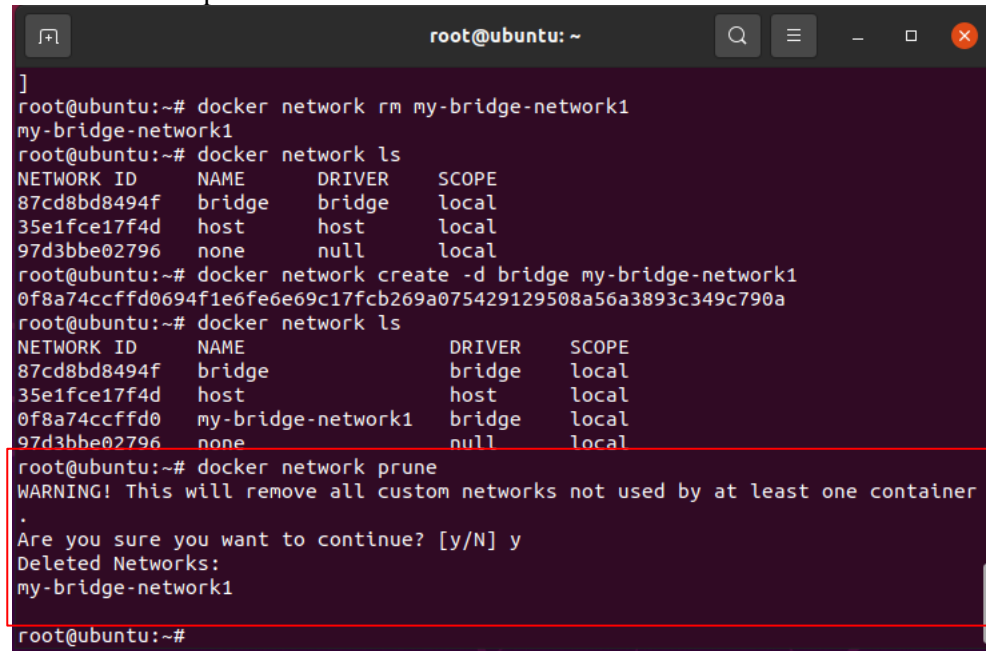
```

root@ubuntu: ~
"ConfigFrom": {
  "Network": ""
},
"ConfigOnly": false,
"Containers": {},
"Options": {
  "com.docker.network.bridge.default_bridge": "true",
  "com.docker.network.bridge.enable_icc": "true",
  "com.docker.network.bridge.enable_ip_masquerade": "true",
  "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
  "com.docker.network.bridge.name": "docker0",
  "com.docker.network.driver.mtu": "1500"
},
"Labels": {}
}
]
root@ubuntu:~# docker network rm my-bridge-network1
my-bridge-network1
root@ubuntu:~# docker network ls
NETWORK ID        NAME                DRIVER              SCOPE
87cd8bd8494f      bridge              bridge              local
35e1fce17f4d      host                host                local
97d3bbe02796      none                null                local
root@ubuntu:~#

```


With below command we can delete unused networks

docker network prune



```
root@ubuntu: ~
]
root@ubuntu:~# docker network rm my-bridge-network1
my-bridge-network1
root@ubuntu:~# docker network ls
NETWORK ID        NAME                DRIVER              SCOPE
87cd8bd8494f      bridge              bridge              local
35e1fce17f4d      host                host                local
97d3bbe02796      none                null                local
root@ubuntu:~# docker network create -d bridge my-bridge-network1
0f8a74ccffd0694f1e6fe6e69c17fcb269a075429129508a56a3893c349c790a
root@ubuntu:~# docker network ls
NETWORK ID        NAME                DRIVER              SCOPE
87cd8bd8494f      bridge              bridge              local
35e1fce17f4d      host                host                local
0f8a74ccffd0      my-bridge-network1  bridge              local
97d3bbe02796      none                null                local
root@ubuntu:~# docker network prune
WARNING! This will remove all custom networks not used by at least one container
.
Are you sure you want to continue? [y/N] y
Deleted Networks:
my-bridge-network1
root@ubuntu:~#
```

Practical No. 5

Aim: Working with Kubernetes.

Kubernetes, or k8s, is an open-source platform that automates Linux container operations. It eliminates many of the manual processes involved in deploying and scaling containerized applications. “In other words, you can cluster together groups of hosts running Linux containers, and Kubernetes helps you easily and efficiently manage those clusters.”

Install MicroK8s on Linux

sudo snap install microk8s --classic

```

root@ubuntu: ~
root@ubuntu:~# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
87cd8bd8494f        bridge              bridge              local
35e1fce17f4d        host                host                local
97d3bbe02796        none                null                local
root@ubuntu:~# docker network create -d bridge my-bridge-network1
0f8a74ccffd0694f1e6fe6e69c17fcb269a075429129508a56a3893c349c790a
root@ubuntu:~# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
87cd8bd8494f        bridge              bridge              local
35e1fce17f4d        host                host                local
0f8a74ccffd0        my-bridge-network1  bridge              local
97d3bbe02796        none                null                local
root@ubuntu:~# docker network prune
WARNING! This will remove all custom networks not used by at least one container
.
Are you sure you want to continue? [y/N] y
Deleted Networks:
my-bridge-network1

root@ubuntu:~# docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
root@ubuntu:~# sudo snap install microk8s --classic
Download snap "microk8s" (2262) from channel "1.21/stable" 72% 250kB/s 3m32s

```

```

root@ubuntu:~# docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
root@ubuntu:~# sudo snap install microk8s --classic
microk8s (1.21/stable) v1.21.1 from Canonical✓ installed
root@ubuntu:~#

```

Add your user to the microk8s admin group

MicroK8s creates a group to enable seamless usage of commands which require admin privilege. Use the following commands to join the group:

sudo usermod -a -G microk8s \$USER

sudo chown -f -R \$USER ~/.kube

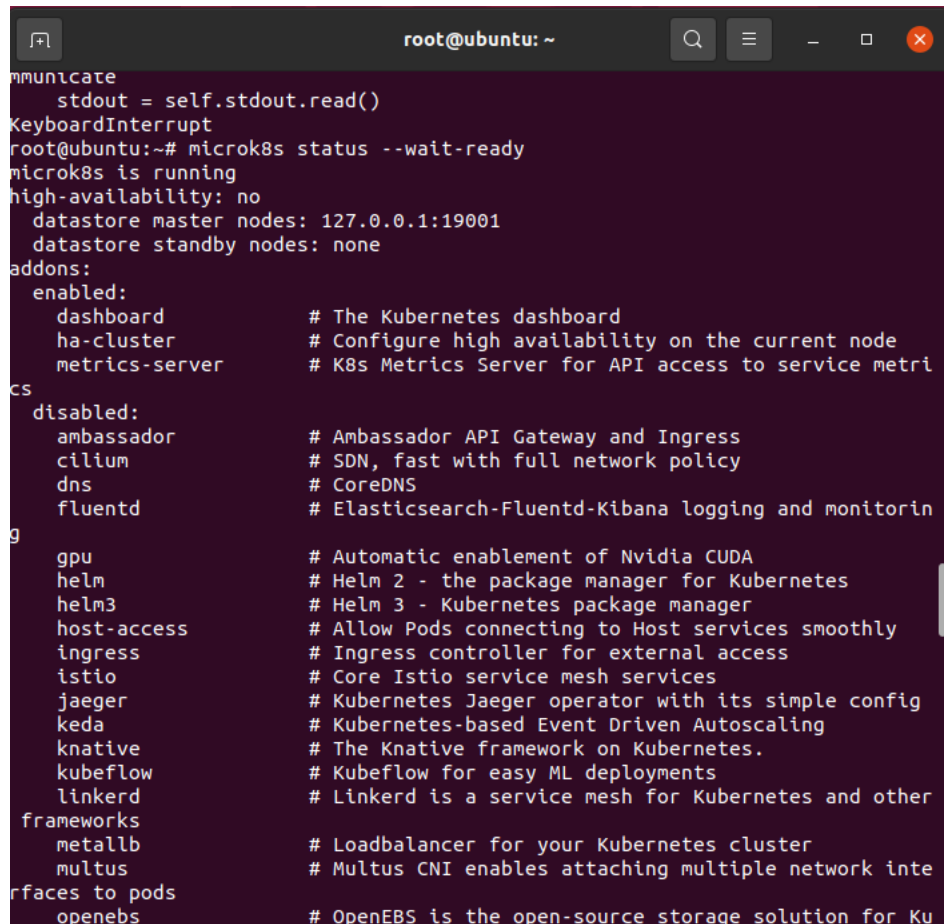
su - \$USER

```

root@ubuntu:~# docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
root@ubuntu:~# sudo snap install microk8s --classic
microk8s (1.21/stable) v1.21.1 from Canonical ✓ installed
root@ubuntu:~# sudo usermod -a -G microk8s $USER
root@ubuntu:~# sudo chown -f -R $USER ~/.kube
root@ubuntu:~# su - $USER

```

Check the status while Kubernetes starts
 microk8s status --wait-ready



```

root@ubuntu: ~
microk8s status --wait-ready
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    dashboard          # The Kubernetes dashboard
    ha-cluster          # Configure high availability on the current node
    metrics-server      # K8s Metrics Server for API access to service metri
  disabled:
    ambassador         # Ambassador API Gateway and Ingress
    cilium              # SDN, fast with full network policy
    dns                 # CoreDNS
    fluentd             # Elasticsearch-Fluentd-Kibana logging and monitorin
  gpu                  # Automatic enablement of Nvidia CUDA
  helm                 # Helm 2 - the package manager for Kubernetes
  helm3                # Helm 3 - Kubernetes package manager
  host-access          # Allow Pods connecting to Host services smoothly
  ingress              # Ingress controller for external access
  istio                # Core Istio service mesh services
  jaeger               # Kubernetes Jaeger operator with its simple config
  keda                 # Kubernetes-based Event Driven Autoscaling
  knative               # The Knative framework on Kubernetes.
  kubeflow              # Kubeflow for easy ML deployments
  linkerd              # Linkerd is a service mesh for Kubernetes and other
  frameworks
  metallb              # Loadbalancer for your Kubernetes cluster
  multus               # Multus CNI enables attaching multiple network inte
  rfaces to pods
  openebs              # OpenEBS is the open-source storage solution for Ku

```

Turn on the services you want

microk8s enable dashboard dns ingress

```

root@ubuntu: ~
root@ubuntu:~# microk8s enable dashboard dns ingress
Addon dashboard is already enabled.
Enabling DNS
Applying manifest
serviceaccount/coredns created
configmap/coredns created
deployment.apps/coredns created
service/kube-dns created
clusterrole.rbac.authorization.k8s.io/coredns created
clusterrolebinding.rbac.authorization.k8s.io/coredns created
Restarting kubelet
DNS is enabled
Enabling Ingress
ingressclass.networking.k8s.io/public created
namespace/ingress created
serviceaccount/nginx-ingress-microk8s-serviceaccount created
clusterrole.rbac.authorization.k8s.io/nginx-ingress-microk8s-clusterrole created
role.rbac.authorization.k8s.io/nginx-ingress-microk8s-role created
clusterrolebinding.rbac.authorization.k8s.io/nginx-ingress-microk8s created
rolebinding.rbac.authorization.k8s.io/nginx-ingress-microk8s created
configmap/nginx-load-balancer-microk8s-conf created
configmap/nginx-ingress-tcp-microk8s-conf created
configmap/nginx-ingress-udp-microk8s-conf created
daemonset.apps/nginx-ingress-microk8s-controller created
Ingress is enabled
root@ubuntu:~# microk8s kubectl get all --all-namespaces

```

NAMESPACE	NAME	RESTARTS	AGE	READY	STATUS
kube-system	pod/calico-kube-controllers-f7868dd95-6ffw9	0	17m	0/1	ContainerCreating
kube-system	pod/metrics-server-8bbfb4bdb-74xg8	0	4m20s	0/1	ContainerCreating
kube-system	pod/kubernetes-dashboard-85fd7f45cb-nwbgq	0	2m52s	0/1	ContainerCreating
kube-system	pod/dashboard-metrics-scraper-78d7698477-m2k7v	0		0/1	ContainerCreating

Start using Kubernetes

microk8s kubectl get all --all-namespaces

```

root@ubuntu: ~
configmap/nginx-ingress-udp-microk8s-conf created
daemonset.apps/nginx-ingress-microk8s-controller created
Ingress is enabled
root@ubuntu:~# microk8s kubectl get all --all-namespaces
NAMESPACE          NAME                                                    READY   STATUS
kube-system        pod/calico-kube-controllers-f7868dd95-6ffw9          0/1     Contai
nerCreating        0              17m
kube-system        pod/metrics-server-8bbfb4bdb-74xg8                  0/1     Contai
nerCreating        0              4m20s
kube-system        pod/kubernetes-dashboard-85fd7f45cb-nwbgq           0/1     Contai
nerCreating        0              2m52s
kube-system        pod/dashboard-metrics-scraper-78d7698477-m2k7v       0/1     Contai
nerCreating        0              2m52s
kube-system        pod/coredns-7f9c69c78c-57lv5                        0/1     Contai
nerCreating        0              46s
kube-system        pod/calico-node-4kqs5                                0/1     Init:2
/3              0              17m

NAMESPACE          NAME                                                    TYPE          CLUSTER-IP
EXTERNAL-IP        PORT(S)          AGE
default            service/kubernetes                                ClusterIP      10.152.183.1
<none>             443/TCP          18m
kube-system        service/metrics-server                            ClusterIP      10.152.183.219
<none>             443/TCP          4m20s
kube-system        service/kubernetes-dashboard                      ClusterIP      10.152.183.241
<none>             443/TCP          4m4s
kube-system        service/dashboard-metrics-scraper                 ClusterIP      10.152.183.252
<none>             8000/TCP         4m4s
kube-system        service/kube-dns                                  ClusterIP      10.152.183.10
<none>             53/UDP,53/TCP,9153/TCP 46s

NAMESPACE          NAME                                                    DESIRED   CU
CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
kube-system        daemonset.apps/calico-node                            1          1
0          1          0          0          kubernetes.io/os=linux 18m

```

Access the Kubernetes dashboard
microk8s dashboard-proxy

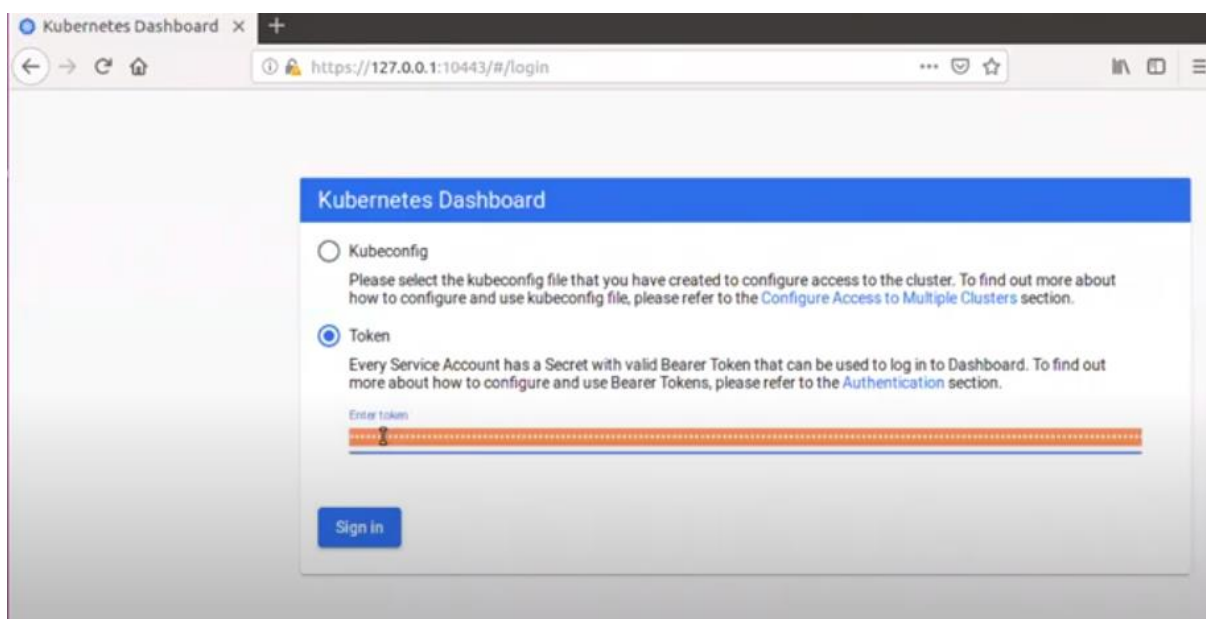
```
root@ubuntu: ~  
kubernetes-dashboard --replicas=1  
kubernetes-dashboard --replicas=1  
root@ubuntu:~# microk8s dashboard-proxy  
Checking if Dashboard is running.  
Dashboard will be available at https://127.0.0.1:10443  
Use the following token to login:  
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50L3NlcjZyVHlY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zeXZJ2AwnLYWtjbnVudC9zeXZJ2AwnLnwFfjY291bnQubmFtZSI6ImRlZmF1bHQiLCJrdjklcm5ldGVzLmlvL3NlcjZyVHlY2NvdW50L3NlcjZyUTYWNjb3VudC51awQiOiJkZTM2ZDZlZS1kNWY1LTlRiMmEtYTZMeY0YVWFJNGMxYjFmMzk1LCJzdWIioiJzeXN0ZW06c2VydmllZWJZFjY291bnQ6a3ViZS1zeXN0ZW06ZGMvYVVsdcJ9.VrRWQF-Lj9KP2BbbFCykmbH000RqvwvxHkavI4E23Au4nR1NCo3vEdwaHf_XVo8CpiwwXONmjdvXk0jcroqq61LR04-7firdj_uwnQMgFFFTX8i150NX900uzSDO799VLG2UGz-CW01a0vRFREYdt-LIQ2U56lsYgtDt5nKppau-Y07tFLvhF3KI_N6BlLKFOvr_CtB6zJ1Gdo29-V2_Mmy2DCTCE85VH3F6GZiph3uLP7Qv8SAOhzvL8e49VVrbfnL-KwemBzhXdjd1X4L2xEh1ZJoY_nhmrcN4MivZE5vlqWiVUTY-nxdIW85tYG0YAyz0oCXBEpSSavVFhxOKW
```

Token for login:

```
eyJhbGciOiJIUzI1NiIsImtpZCI6IkpncUpFTmRfUkJOdko1RVVSUVBsdzVBZUpHYm9z
UUxxb1phSDE4aGYwQncifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50I
iwia3ViZXJlcy5pb3p5ZXJ2aWVhY2NvdW50IjBhY2U0IjRldWJlLXN5c3RlbnIsImt1
YmVybWV0ZXMuaw8vc2VydmljZWZjY291bnQvc2VjcmV0Lm5hbWUiOiJkZ
```

WZhdWx0LXRva2VuLWtnaDR4Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWVjb3VudC9zZ
 XJ2aWNlLWFjY291bnQubmFtZSI6ImRlZmF1bHQiLCJrdWJlcm5ldGVzLm1vL3NlcnZpY
 2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiJkZTM2ZDZlZS1kNWY1LTRiMmEtY
 TgzMy04YWFjNGMxYjFmMzkiLCJzdWIiOiJzeXN0ZW06c2Vydm1jZWJjY291bnQ6a3ViZ
 S1zeXN0ZW06ZGVmYXVsdCJ9.VrRWQF-
 Lj9KP2BbbFCykbMh00RqvWvxHkavI4E23Au4nR1NCo3vEdwAhf_XVo8CpiwvXONmjdV
 Xk0jcroqq61LR04-7firdj_uWnQMgFFFtTX8i150NX900uzSD0799VLG2UgZ-
 cW01a0vRFREYdt-LIQ2U56lsYgTdt5nKppau-
 Y07tFLvhF3KI_N6B1lKf0Vr_CtB6zJ1GDo29-
 V2_MmY2DCTCE85VH3F6GZiph3uLp7Qv8SA0hzV18e49VVrbfNL-
 KwemBZhXdjd1X4L2xEh1ZJqY_nhmrN4MivTZE5yLqWIyUTY-
 nxdiwB5tYG0YAYZ0oCXBEpSsgyVfhx0Kw

Sign in with token:



The screenshot shows the Kubernetes Dashboard interface. The browser address bar displays the URL `https://127.0.0.1:10443/#/clusterrole?namespace=default`. The dashboard header includes the Kubernetes logo, a search bar, and navigation icons. The left sidebar contains a menu with the following sections:

- Cluster**
 - Cluster Roles (selected)
 - Namespaces
 - Nodes
 - Persistent Volumes
 - Storage Classes
- Namespace**
 - default (selected)
- Overview**
- Workloads**
 - Cron Jobs
 - Daemon Sets
 - Deployments
 - Jobs
 - Pods

The main content area is titled "Cluster Roles" and displays a table with the following data:

Name	Age	
nginx-ingress-microk8s-clusterrole	2 hours	⋮
coredns	2 hours	⋮
kubernetes-dashboard	2 hours	⋮

At the bottom of the table, there is a pagination indicator showing "1 - 3 of 3" and navigation arrows.