MICROSERVICE ARCHITECTURE

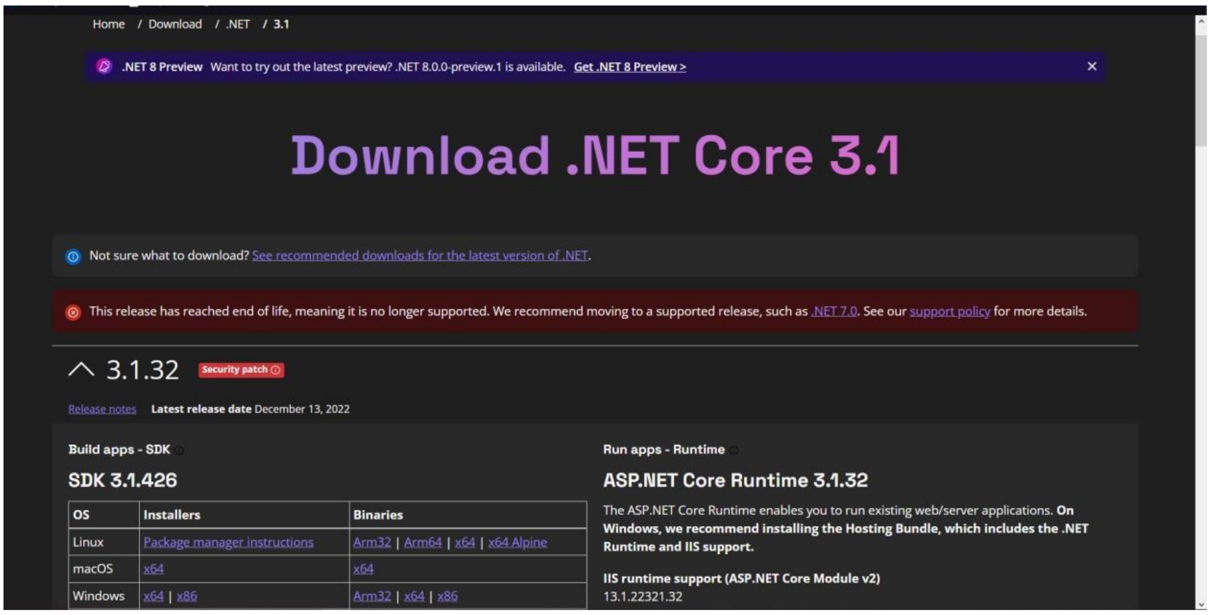
INDEX

Practical 1

Aim: Building ASP.NET Core MVC Application.

Writeup:

Step 1:
Install .Net Core SDK.



Home / Download / .NET / 3.1

.NET 8 Preview Want to try out the latest preview? .NET 8.0.0-preview.1 is available. Get .NET 8 Preview >

Download .NET Core 3.1

Not sure what to download? See recommended downloads for the latest version of .NET.

This release has reached end of life, meaning it is no longer supported. We recommend moving to a supported release, such as .NET 7.0. See our support policy for more details.

3.1.32 Security patch

Release notes Latest release date December 13, 2022

Build apps - SDK

SDK 3.1.426

OS	Installers	Binaries
Linux	Package manager instructions	Arm32 | Arm64 | x64 | x64 Alpine
macOS	x64	x64
Windows	x64 | x86	Arm32 | x64 | x86

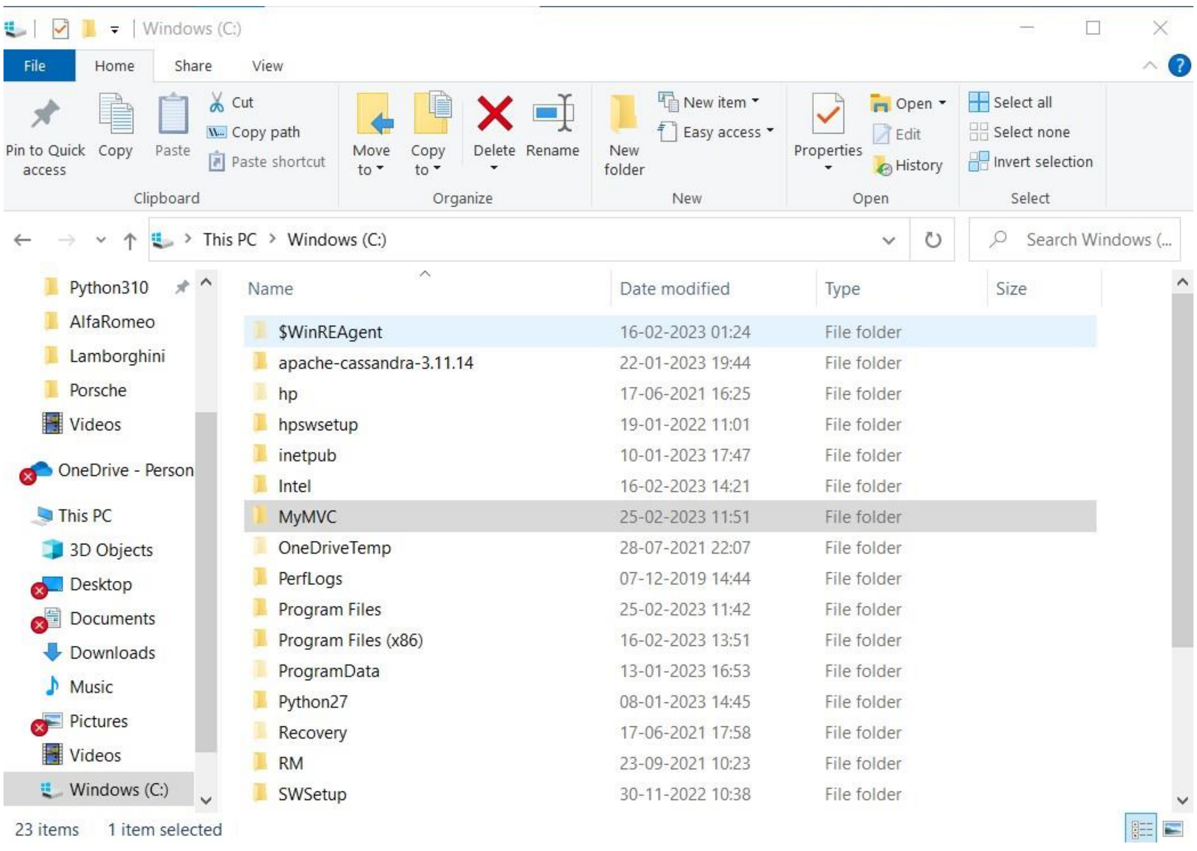
Run apps - Runtime

ASP.NET Core Runtime 3.1.32

The ASP.NET Core Runtime enables you to run existing web/server applications. **On Windows, we recommend installing the Hosting Bundle, which includes the .NET Runtime and IIS support.**

IIS runtime support (ASP.NET Core Module v2)
13.1.22321.32

Step 2:
Create a Folder MyMVC in C: drive.



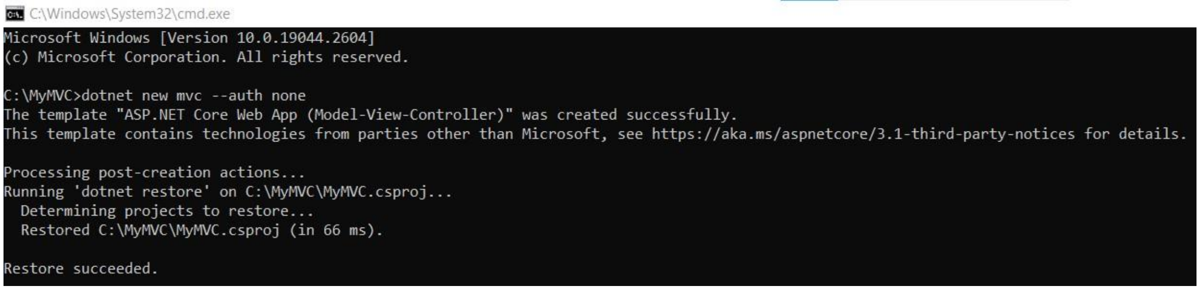
Name	Date modified	Type	Size
$WinREAgent	16-02-2023 01:24	File folder	
apache-cassandra-3.11.14	22-01-2023 19:44	File folder	
hp	17-06-2021 16:25	File folder	
hpswsetup	19-01-2022 11:01	File folder	
inetpub	10-01-2023 17:47	File folder	
Intel	16-02-2023 14:21	File folder	
MyMVC	25-02-2023 11:51	File folder	
OneDriveTemp	28-07-2021 22:07	File folder	
PerfLogs	07-12-2019 14:44	File folder	
Program Files	25-02-2023 11:42	File folder	
Program Files (x86)	16-02-2023 13:51	File folder	
ProgramData	13-01-2023 16:53	File folder	
Python27	08-01-2023 14:45	File folder	
Recovery	17-06-2021 17:58	File folder	
RM	23-09-2021 10:23	File folder	
SWSetup	30-11-2022 10:38	File folder	

23 items 1 item selected

Step 3:

Open Command Prompt and type the following commands:

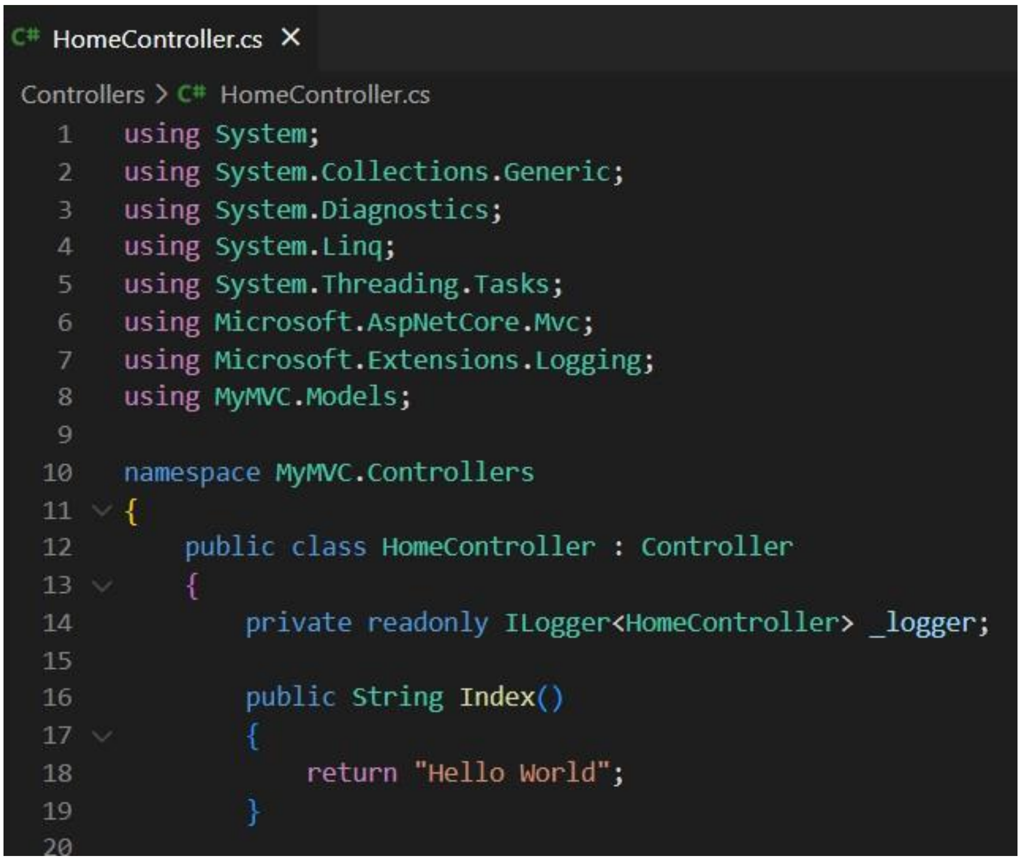
dotnet new mvc --auth none



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2604]
(c) Microsoft Corporation. All rights reserved.

C:\MyMVC>dotnet new mvc --auth none
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/3.1-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 66 ms).

Restore succeeded.
```

Step 4:

Go to the controllers folder and modify HomeController.cs file to match the following code:



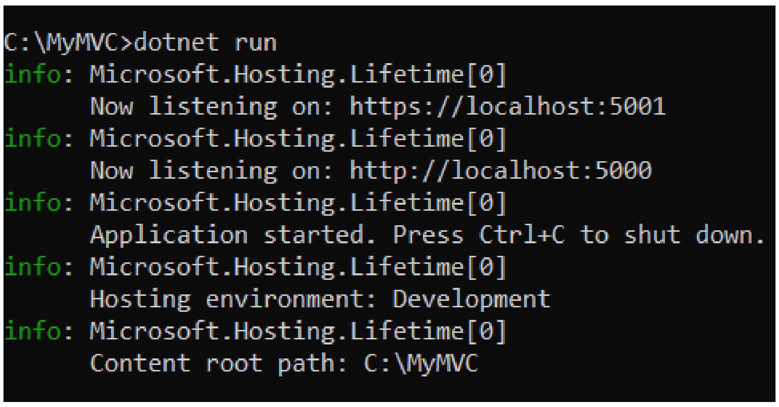
HomeController.cs

Controllers > HomeController.cs

```csharp
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
    {
        private readonly ILogger<HomeController> _logger;

        public String Index()
        {
            return "Hello World";
        }

```

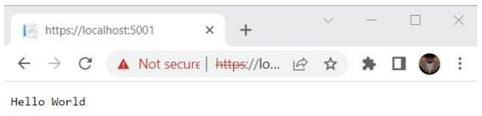
Step 5:

- Run the code



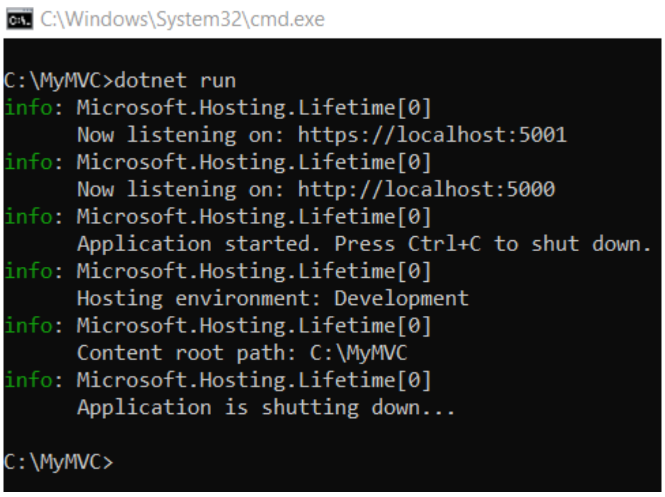
```
C:\MyMVC>dotnet run
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]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

- Paster the localhost:5001 link in your browser (In the case we are using Chrome).



Hello World

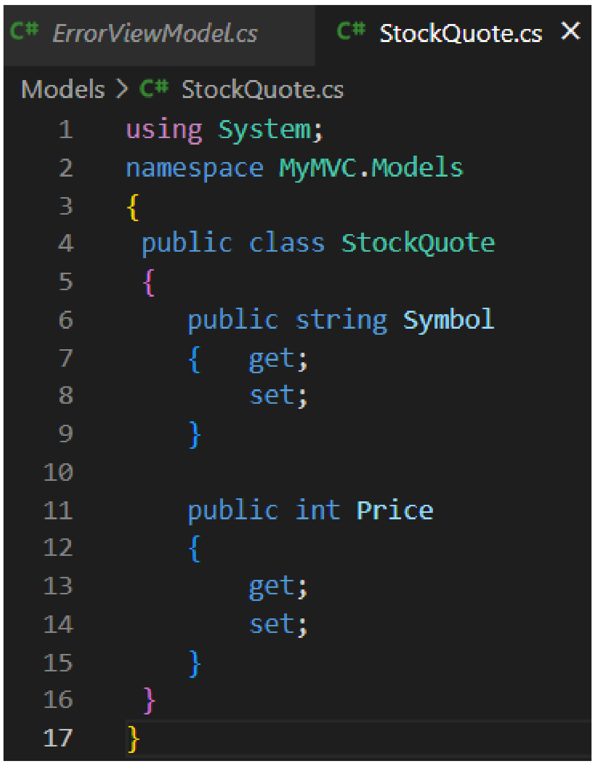
Step 6: Now go back to command prompt and stop running project using CTRL+C.



C:\Windows\System32\cmd.exe

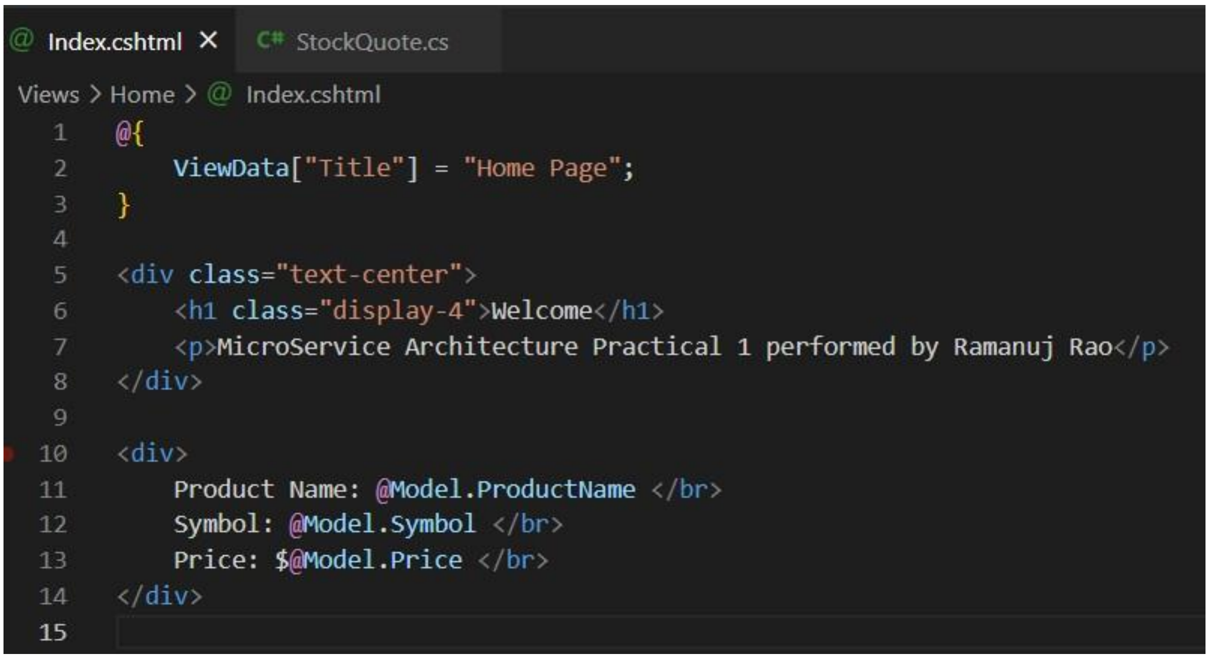
```
C:\MyMVC>dotnet run
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]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

C:\MyMVC>
```

Step 7: Go to models folder and add new file StockQuote.cs to it with following content:



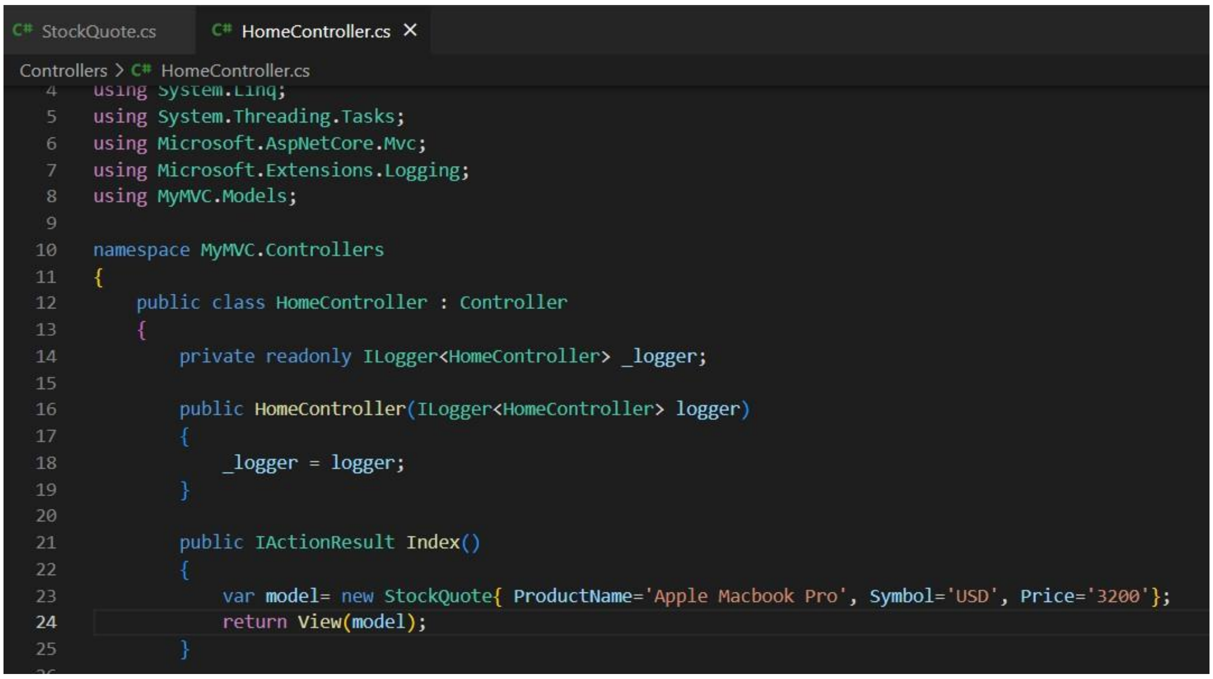
```
using System;
namespace MyMVC.Models
{
 public class StockQuote
 {
     public string Symbol
     {    get;
          set;
     }

     public int Price
     {
          get;
          set;
     }
 }
}
```

Step 8: Now Add View to folder then home folder in it and modify index.cshtml file to match following



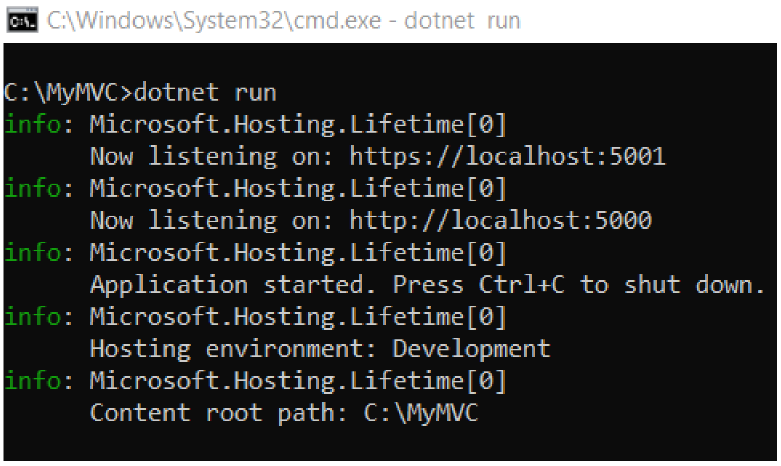
```
@{
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <h1 class="display-4">Welcome</h1>
    <p>MicroService Architecture Practical 1 performed by Ramanuj Rao</p>
</div>

<div>
    Product Name: @Model.ProductName </br>
    Symbol: @Model.Symbol </br>
    Price: $@Model.Price </br>
</div>
```

Step 9: Now modify HomeController.cs file to match following:



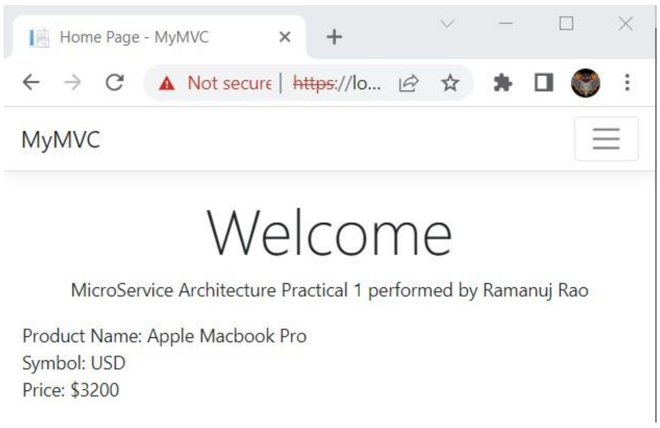
```
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC.Models;

namespace MyMVC.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;

        public HomeController(ILogger<HomeController> logger)
        {
            _logger = logger;
        }

        public IActionResult Index()
        {
            var model= new StockQuote{ ProductName='Apple Macbook Pro', Symbol='USD', Price='3200'};
            return View(model);
        }
```

Step 10: Now run the project using **dotnet run**



```
C:\MyMVC>dotnet run
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
```

Step 11: Now go back to browser and refresh to get modified view response.



Welcome

MicroService Architecture Practical 1 performed by Ramanuj Rao

Product Name: Apple Macbook Pro
Symbol: USD
Price: $3200

Practical 2

Aim: Building an ASP.NET CORE REST API.

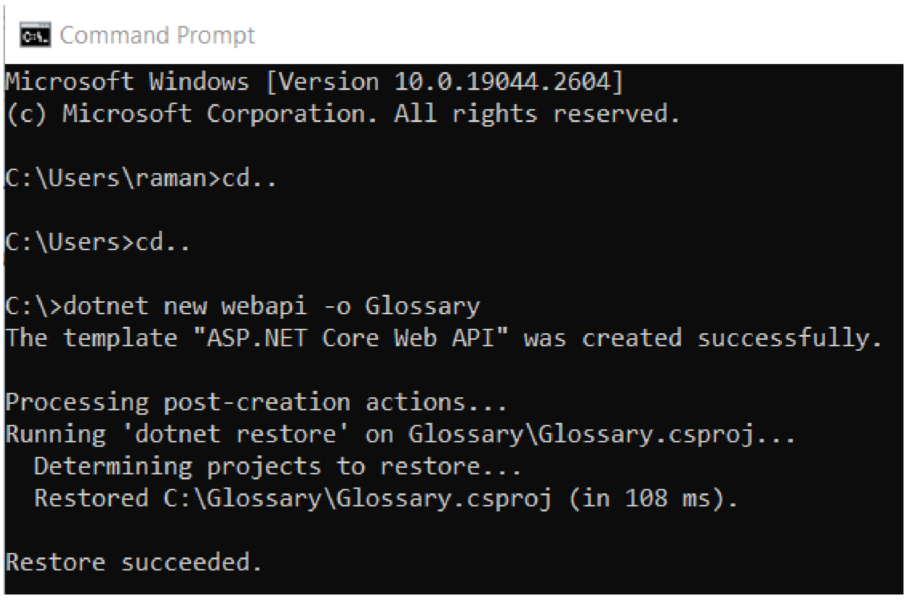
Writeup:

Step 1: Create your Web API.

- Open two command prompts.

CMD-1:

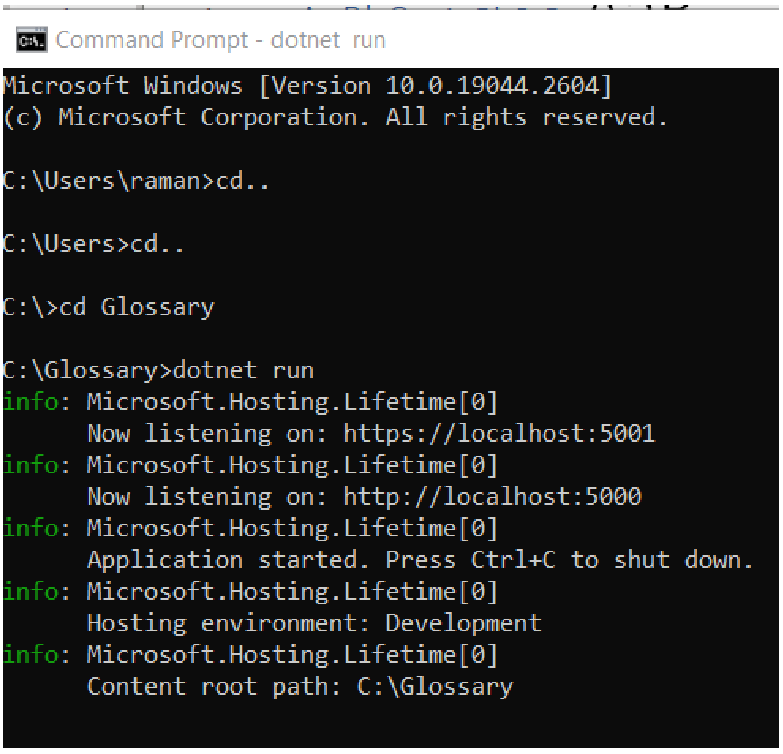
dotnet new webapi -o Glossary



```
Command Prompt
Microsoft Windows [Version 10.0.19044.2604]
(c) Microsoft Corporation. All rights reserved.

C:\Users\raman>cd..

C:\Users>cd..

C:\>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:\Glossary\Glossary.csproj (in 108 ms).

Restore succeeded.
```

CMD-2:

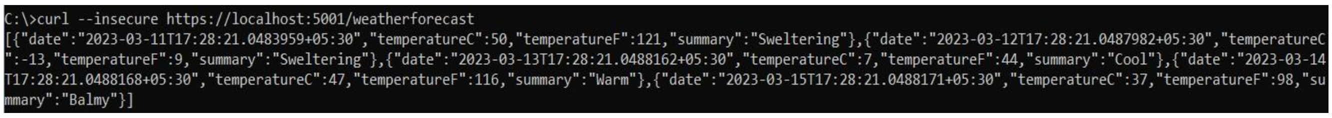
cd Glossary
dotnet run



```
Command Prompt - dotnet  run
Microsoft Windows [Version 10.0.19044.2604]
(c) Microsoft Corporation. All rights reserved.

C:\Users\raman>cd..

C:\Users>cd..

C:\>cd Glossary

C:\Glossary>dotnet run
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:\Glossary
```

Step 2: In Command Prompt 2: (try running ready made weatherforecast class for testing)

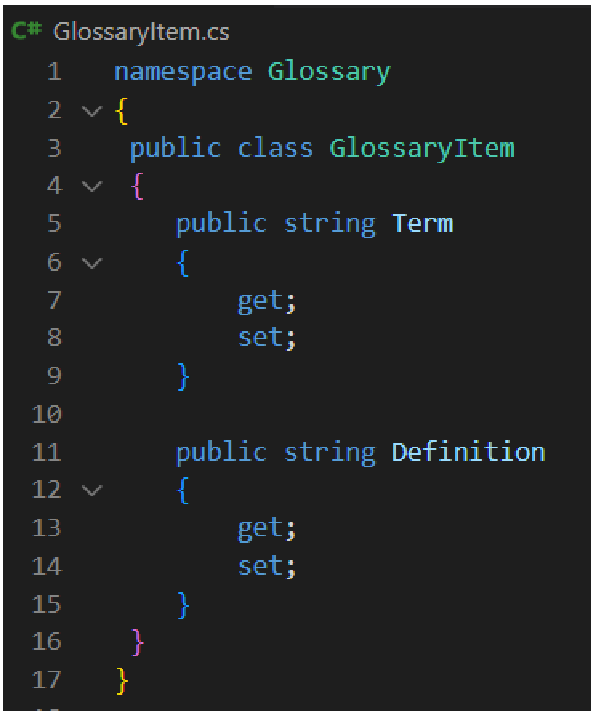
curl --insecure https://localhost:5001/weatherforecast



```
C:\>curl --insecure https://localhost:5001/weatherforecast
[{"date":"2023-03-11T17:28:21.0483959+05:30","temperatureC":50,"temperatureF":121,"summary":"Sweltering"},{"date":"2023-03-12T17:28:21.0487982+05:30","temperatureC
":-13,"temperatureF":9,"summary":"Sweltering"},{"date":"2023-03-13T17:28:21.0488162+05:30","temperatureC":7,"temperatureF":44,"summary":"Cool"},{"date":"2023-03-14
T17:28:21.0488168+05:30","temperatureC":47,"temperatureF":116,"summary":"Warm"},{"date":"2023-03-15T17:28:21.0488171+05:30","temperatureC":37,"temperatureF":98,"su
mmary":"Balmy"}]
```

Step 3: Now Change the content:-
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

[i] : GlossaryItem.cs

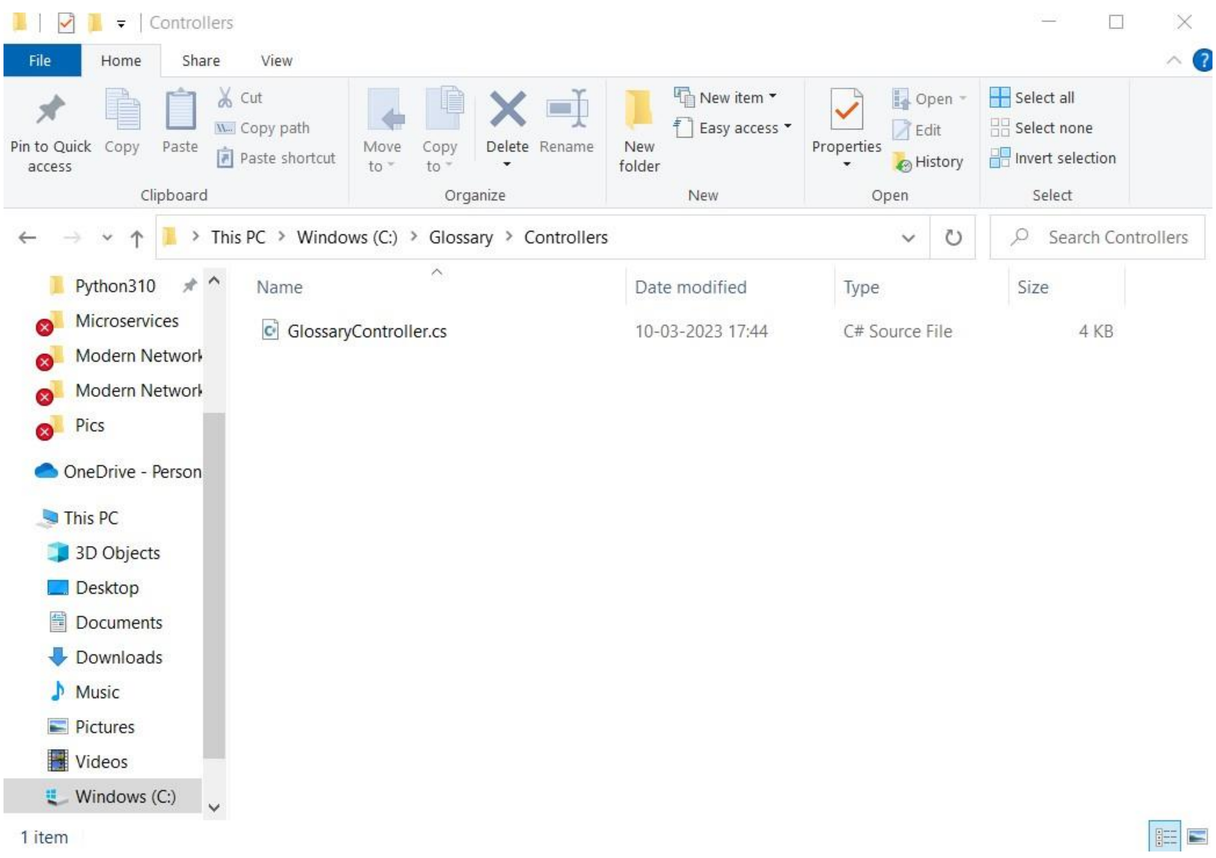


```
namespace Glossary
{
 public class GlossaryItem
 {
    public string Term
    {
        get;
        set;
    }

    public string Definition
    {
        get;
        set;
    }
 }
}
```

[ii] : GlossaryController.cs

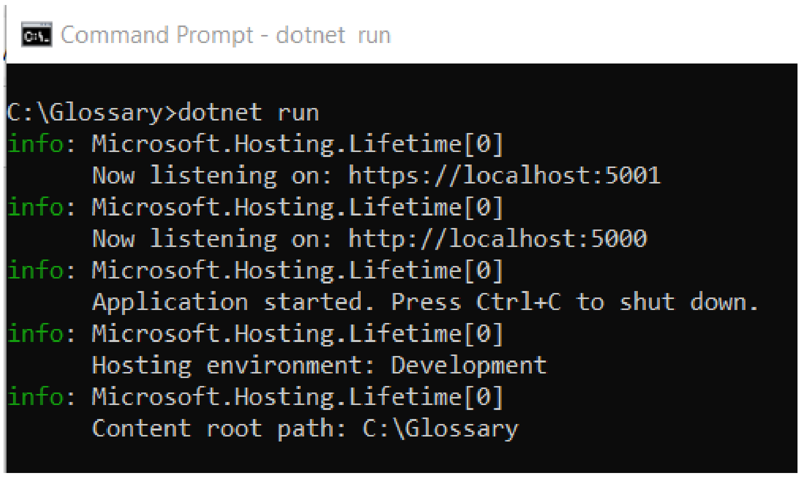
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;

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();
            }
        }
    }
}
```

Output:



Name	Date modified	Type	Size
GlossaryController.cs	10-03-2023 17:44	C# Source File	4 KB

Step 4: Now stop running previous **dotnet run** on command prompt 1 using Ctrl+C. and Run it again for new code.

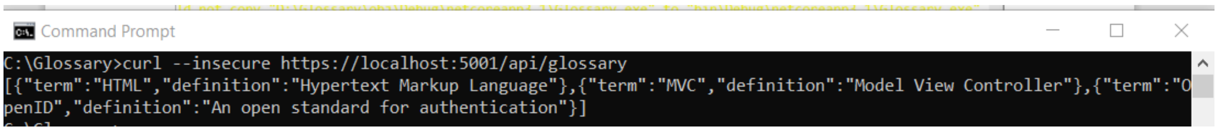


```
C:\Glossary>dotnet run
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]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

Step 5: Run commands to perform certain operations on the GlossaryItem dataset.

[i] : Getting a list of items.

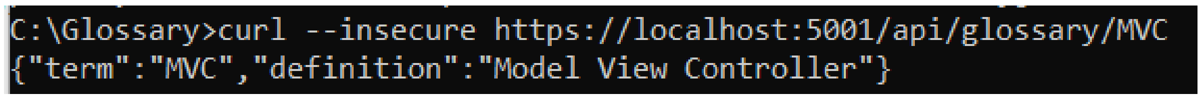
curl --insecure https://localhost:5001/api/glossary



```
C:\Glossary>curl --insecure https://localhost:5001/api/glossary
[{"term":"HTML","definition":"Hypertext Markup Language"},{"term":"MVC","definition":"Model View Controller"},{"term":"O
penID","definition":"An open standard for authentication"}]
```

[ii] : Getting a single item.

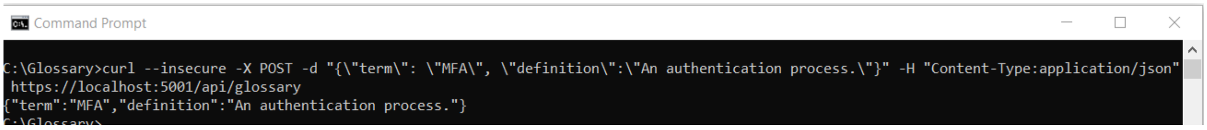
curl --insecure https://localhost:5001/api/glossary/MVC



```
C:\Glossary>curl --insecure https://localhost:5001/api/glossary/MVC
{"term":"MVC","definition":"Model View Controller"}
```

[iii] : Creating an item.

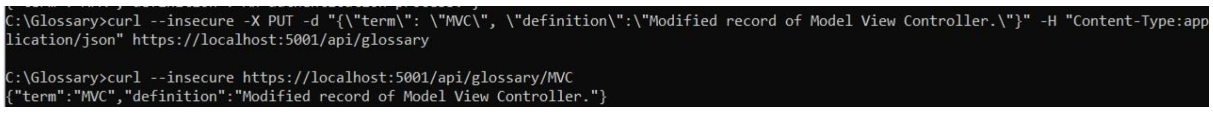
curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\":\"An authentication process.\"}" -H "ContentType:application/json" https://localhost:5001/api/glossary



```
C:\Glossary>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."}
```

[iv] : Update an item.

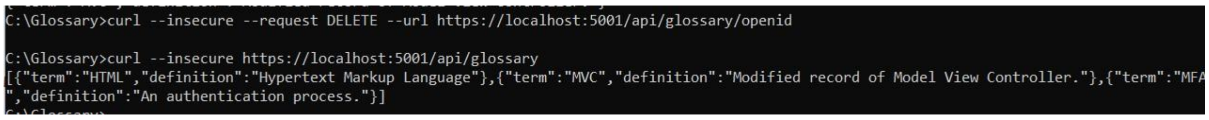
curl --insecure -X PUT -d "{\"term\": \"MVC\", \"definition\":\"Modified record of Model View Controller.\"}" -H "Content-Type:application/json" https://localhost:5001/api/glossary



```
C:\Glossary>curl --insecure -X PUT -d "{\"term\": \"MVC\", \"definition\":\"Modified record of Model View Controller.\"}" -H "Content-Type:app
lication/json" https://localhost:5001/api/glossary

C:\Glossary>curl --insecure https://localhost:5001/api/glossary/MVC
{"term":"MVC","definition":"Modified record of Model View Controller."}
```

[v] : Delete an item.

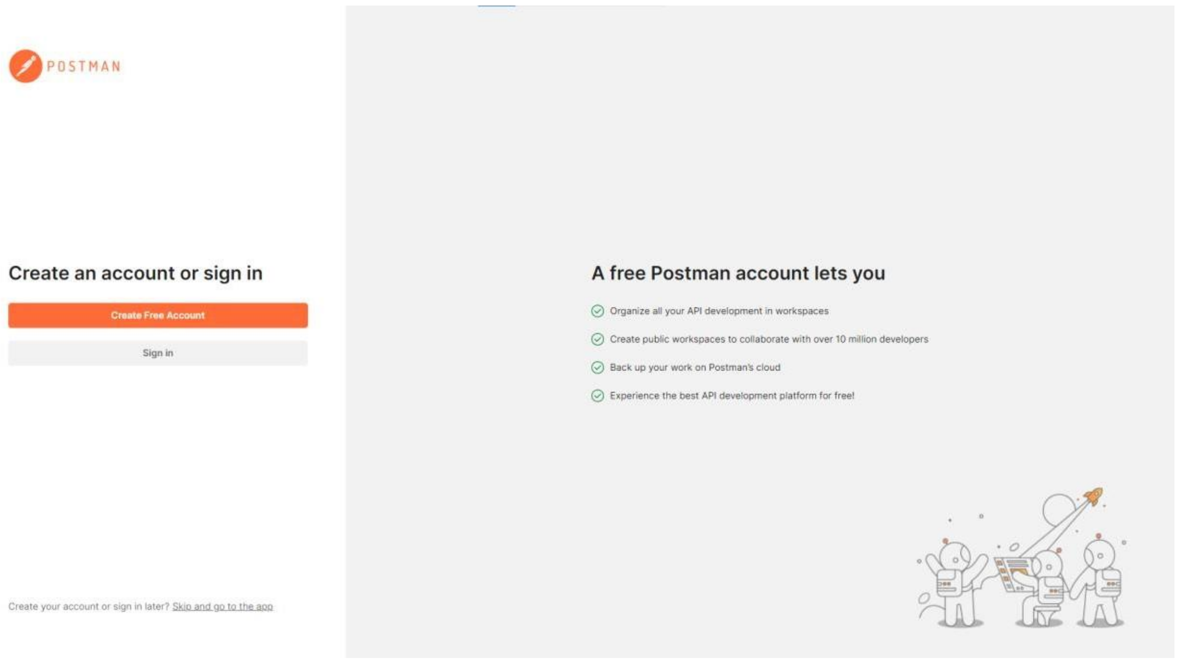
curl --insecure --request DELETE --url https://localhost:5001/api/glossary/openid



```
C:\Glossary>curl --insecure --request DELETE --url https://localhost:5001/api/glossary/openid

C:\Glossary>curl --insecure https://localhost:5001/api/glossary
[{"term":"HTML","definition":"Hypertext Markup Language"},{"term":"MVC","definition":"Modified record of Model View Controller."},{"term":"MFA
","definition":"An authentication process."}]
```

Extra (Step 6): Running commands using Postman.

- First you have you download and install Postman.
- After installing you will see this screen



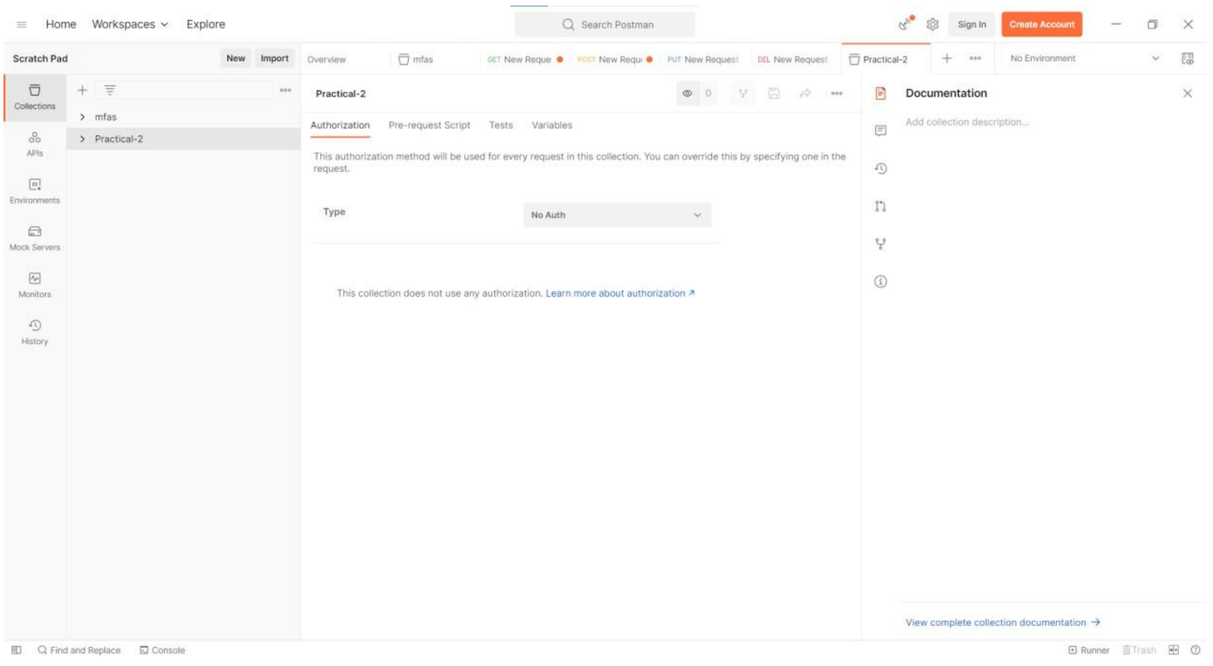
Create an account or sign in

Create your account or sign in later? Skip and go to the app

A free Postman account lets you

- Organize all your API development in workspaces
- Create public workspaces to collaborate with over 10 million developers
- Back up your work on Postman's cloud
- Experience the best API development platform for free!

- Click on "Skip and go to the app"
- Now Click on New Collection and give it a name or your choice



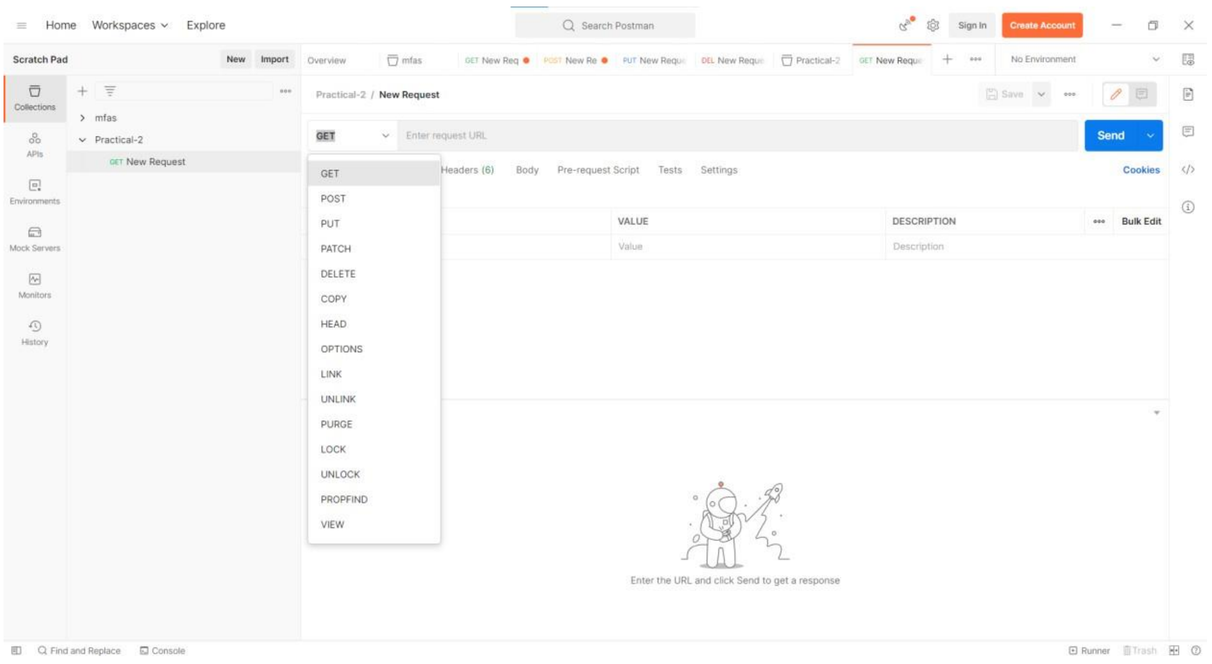
Practical-2

This authorization method will be used for every request in this collection. You can override this by specifying one in the request.

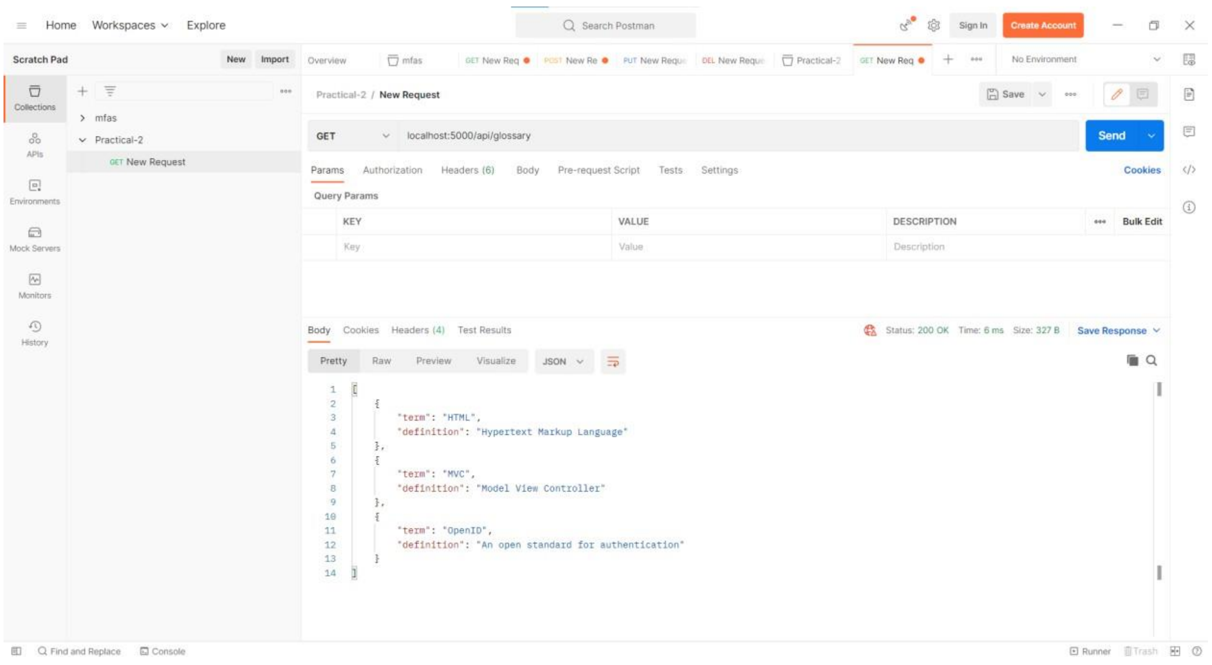
Type: No Auth

This collection does not use any authorization. Learn more about authorization ↗

- Now within the new collection Click on "Add new Request" and within the choices choose the "GET" option.

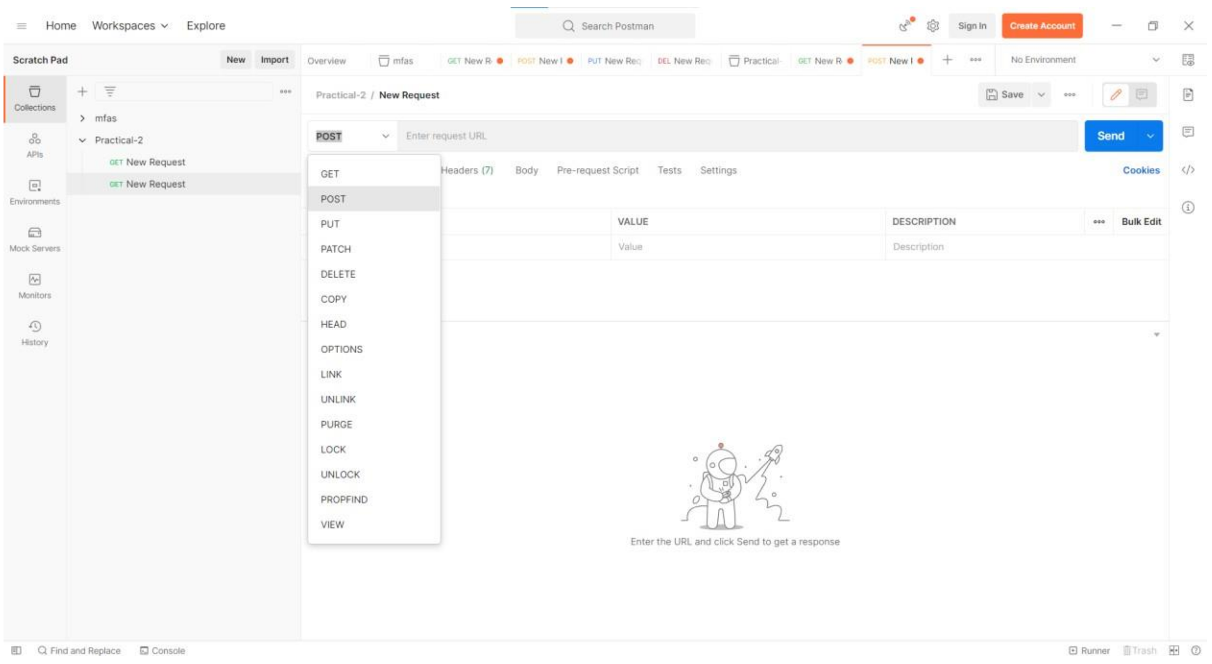


- Now type in **localhost:5000/api/glossary** in the field and Click **Send**.


Practical-2 / New Request
GET
localhost:5000/api/glossary
Send
Status: 200 OK Time: 6 ms Size: 327 B
Pretty
JSON

```
[
    {
        "term": "HTML",
        "definition": "Hypertext Markup Language"
    },
    {
        "term": "MVC",
        "definition": "Model View Controller"
    },
    {
        "term": "OpenID",
        "definition": "An open standard for authentication"
    }
]
```

- Now within the new collection Click on “Add new Request” and within the choices choose the “POST” option.

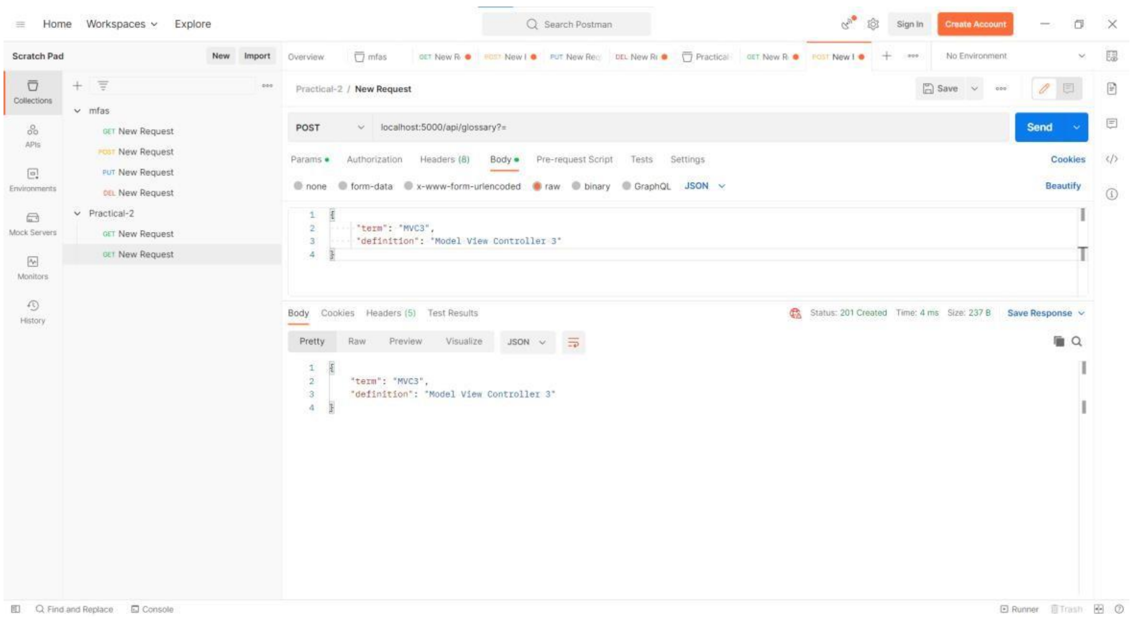


- Now type in **localhost:5000/api/glossary?=** in the field.
- Then choose the “Body” option below and select “Raw” and fill out the following code:

Code:

```
{
       "term": "MVC3",
       "definition": "Model View Controller 3"
}
```

- Now Click **Send**.
- Tip: Make sure to select **JSON** instead of **Text**.



Practical-2 / New Request

POST localhost:5000/api/glossary?=

raw JSON

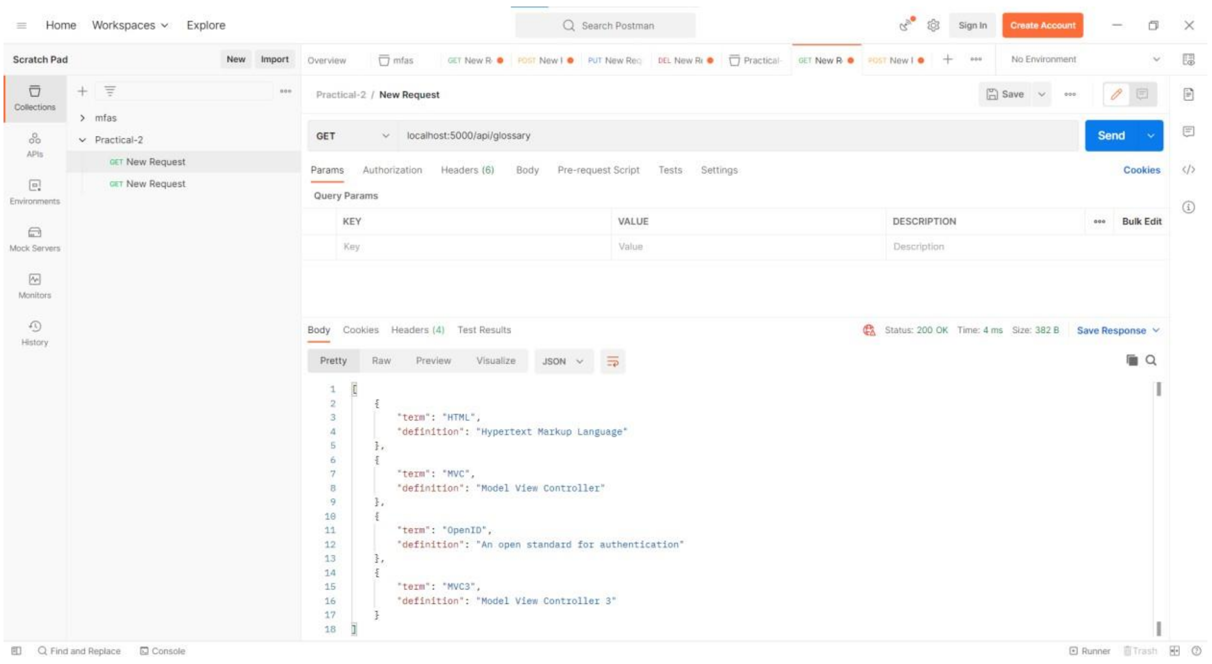
```
{
    "term": "MVC3",
    "definition": "Model View Controller 3"
}
```

Status: 201 Created Time: 4 ms Size: 237 B

Pretty JSON

```
{
    "term": "MVC3",
    "definition": "Model View Controller 3"
}
```

- To confirm that the **POST** Request worked send another **GET** Request and it would have updated with a new value, if not then check you code.



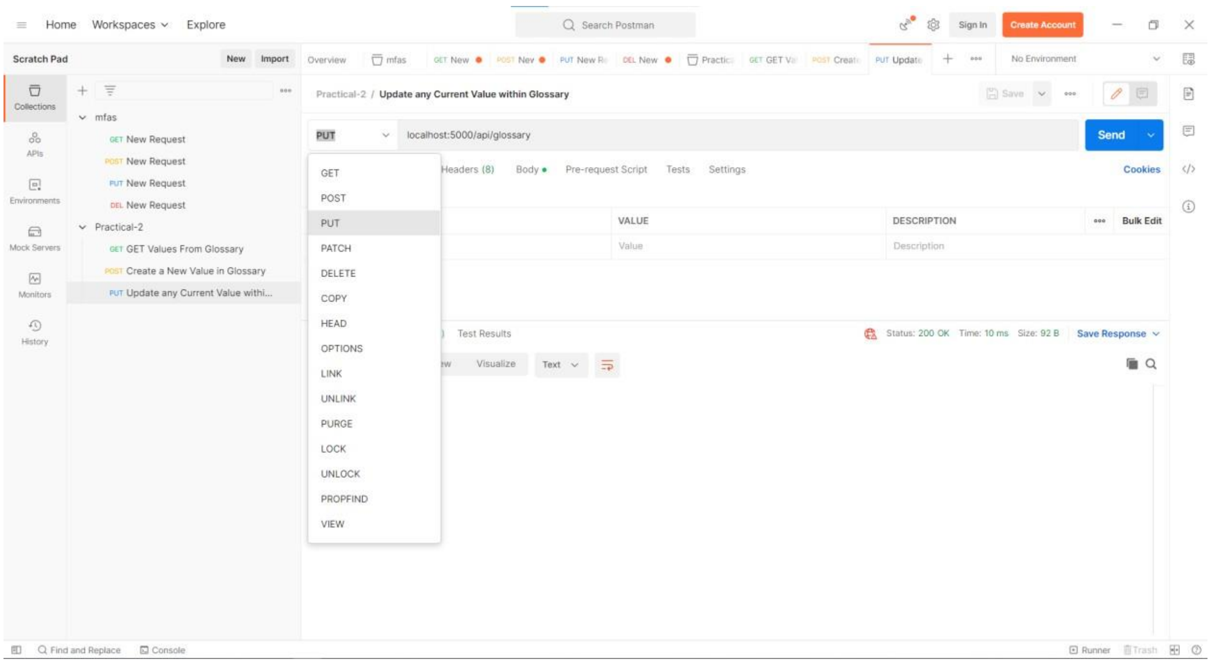
Practical-2 / New Request

GET localhost:5000/api/glossary

Status: 200 OK Time: 4 ms Size: 382 B

```
[
    {
        "term": "HTML",
        "definition": "Hypertext Markup Language"
    },
    {
        "term": "MVC",
        "definition": "Model View Controller"
    },
    {
        "term": "OpenID",
        "definition": "An open standard for authentication"
    },
    {
        "term": "MVC3",
        "definition": "Model View Controller 3"
    }
]
```

- Now within the new collection Click on “Add new Request” and within the choices choose the “PUT” option.



Practical-2 / Update any Current Value within Glossary

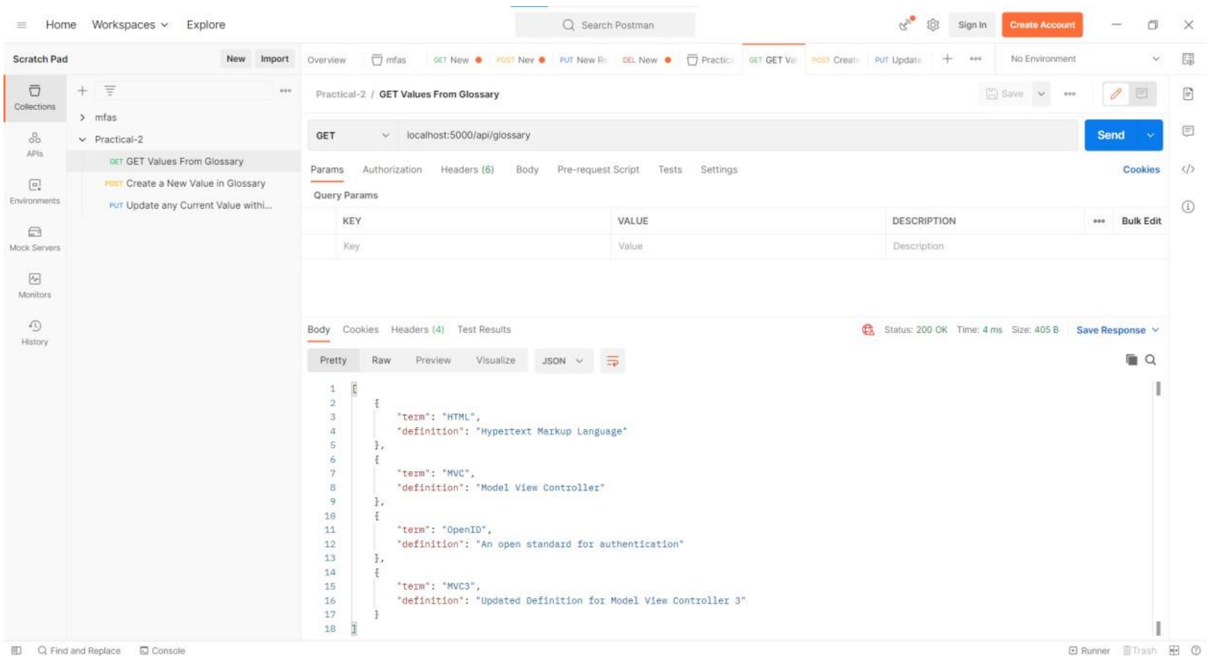
PUT localhost:5000/api/glossary

- Now type in **localhost:5000/api/glossary** in the field.
- Then choose the “Body” option below and select “Raw” and fill out the following code:

Code:

```
{
    "term": "MVC3",
    "definition": "Updated Definition for Model View Controller 3"
}
```

- Click **Send**.
- To confirm the changes made by the PUT Request, send another GET Request to update the values.



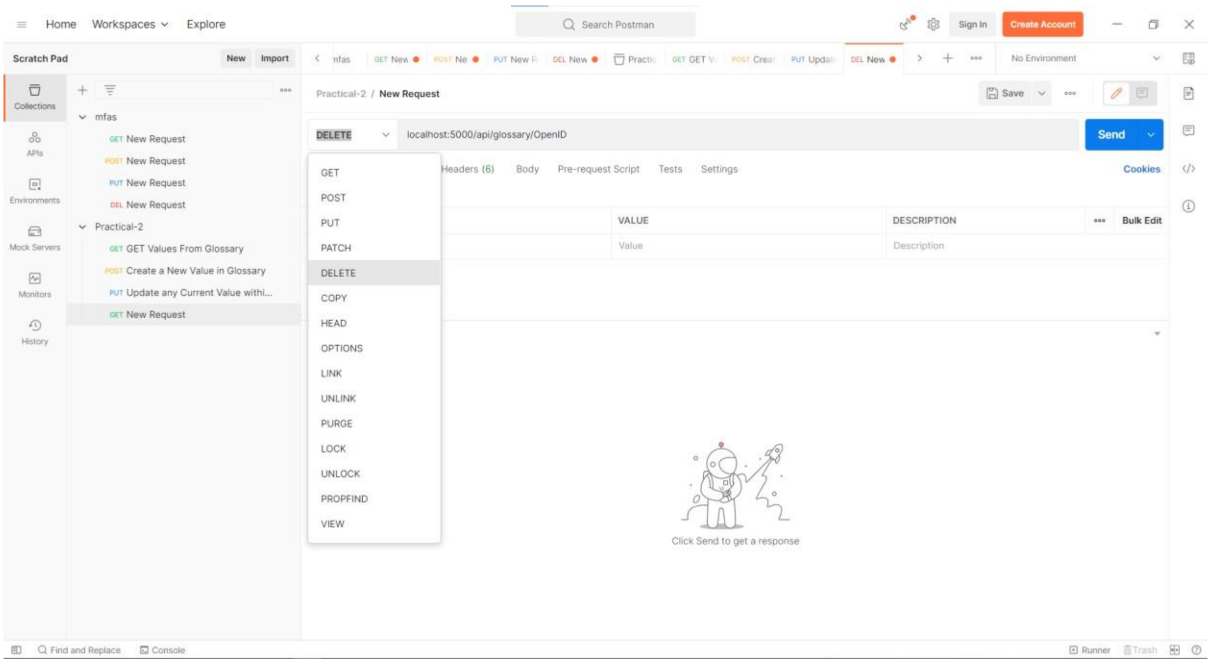
Practical-2 / GET Values From Glossary

GET localhost:5000/api/glossary

Status: 200 OK Time: 4 ms Size: 405 B

```
[
    {
        "term": "HTML",
        "definition": "Hypertext Markup Language"
    },
    {
        "term": "MVC",
        "definition": "Model View Controller"
    },
    {
        "term": "OpenID",
        "definition": "An open standard for authentication"
    },
    {
        "term": "MVC3",
        "definition": "Updated Definition for Model View Controller 3"
    }
]
```

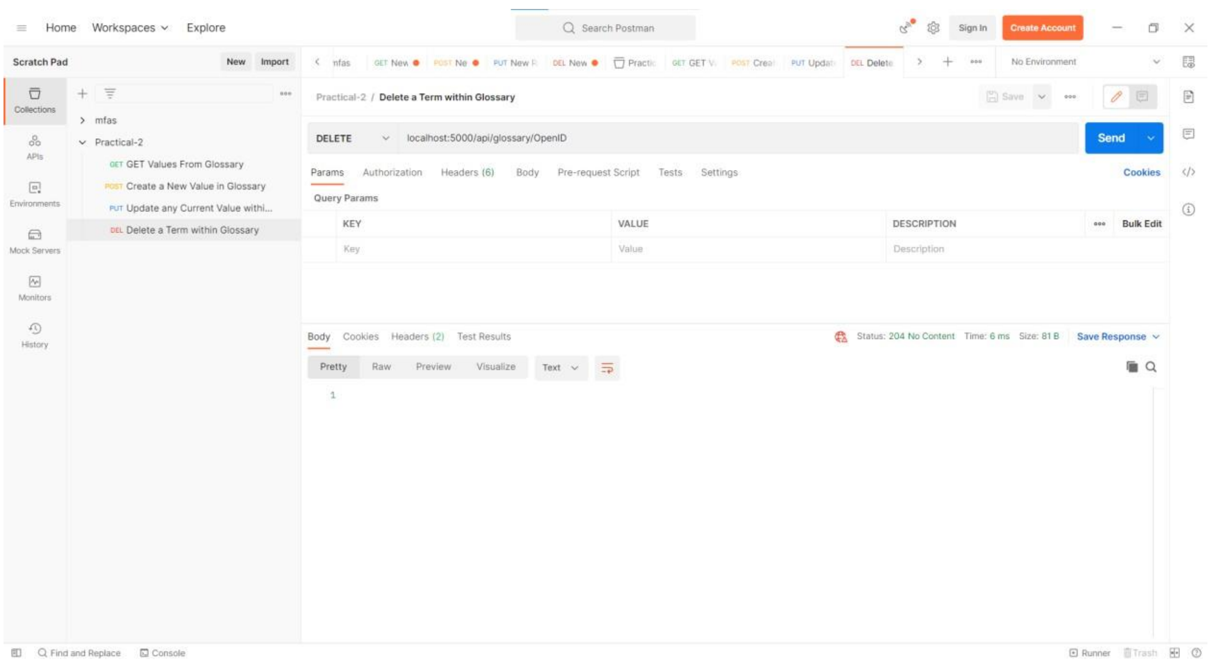
- Now within the new collection Click on "Add new Request" and within the choices choose the "DELETE" option.



Practical-2 / New Request

DELETE localhost:5000/api/glossary/OpenID

- Now type in "**localhost:5000/api/glossary/**" in the field and after it choose the term you would like to remove in this case I have chosen "**OpenID**".
- After this Click **Send**.



Practical-2 / Delete a Term within Glossary

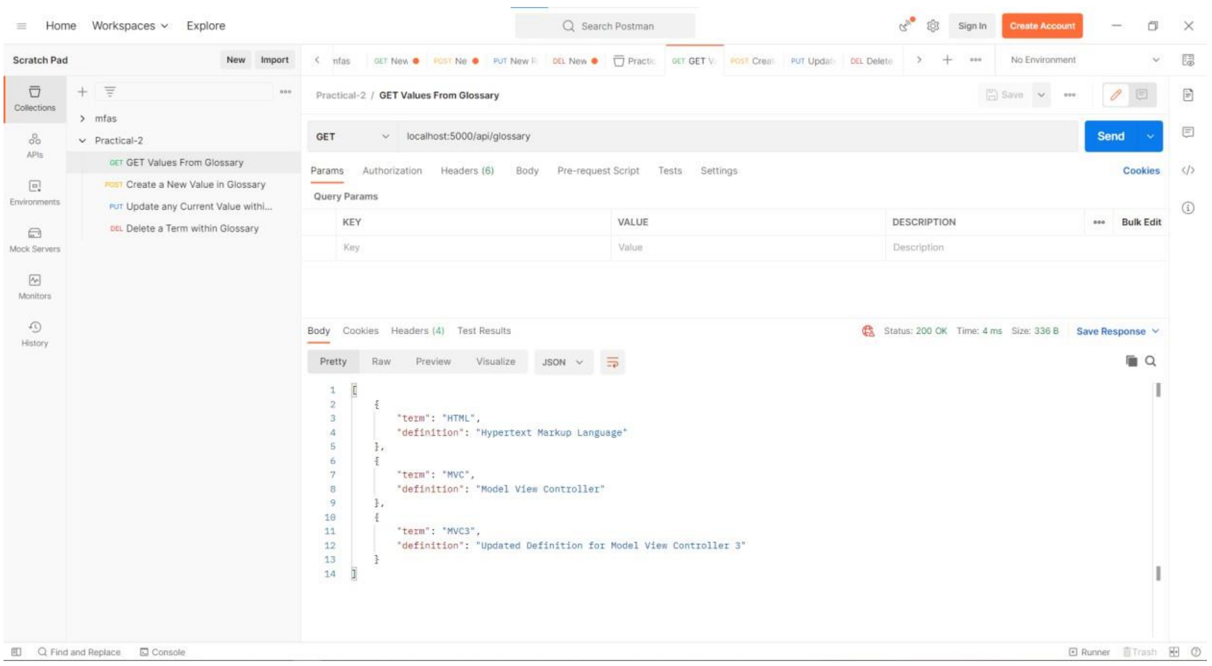
DELETE localhost:5000/api/glossary/OpenID

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (2) Test Results — Status: 204 No Content Time: 6 ms Size: 81 B

- To Confirm the DELETE Request, send another GET Request to update the table.



Practical-2 / GET Values From Glossary

GET localhost:5000/api/glossary

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results — Status: 200 OK Time: 4 ms Size: 336 B

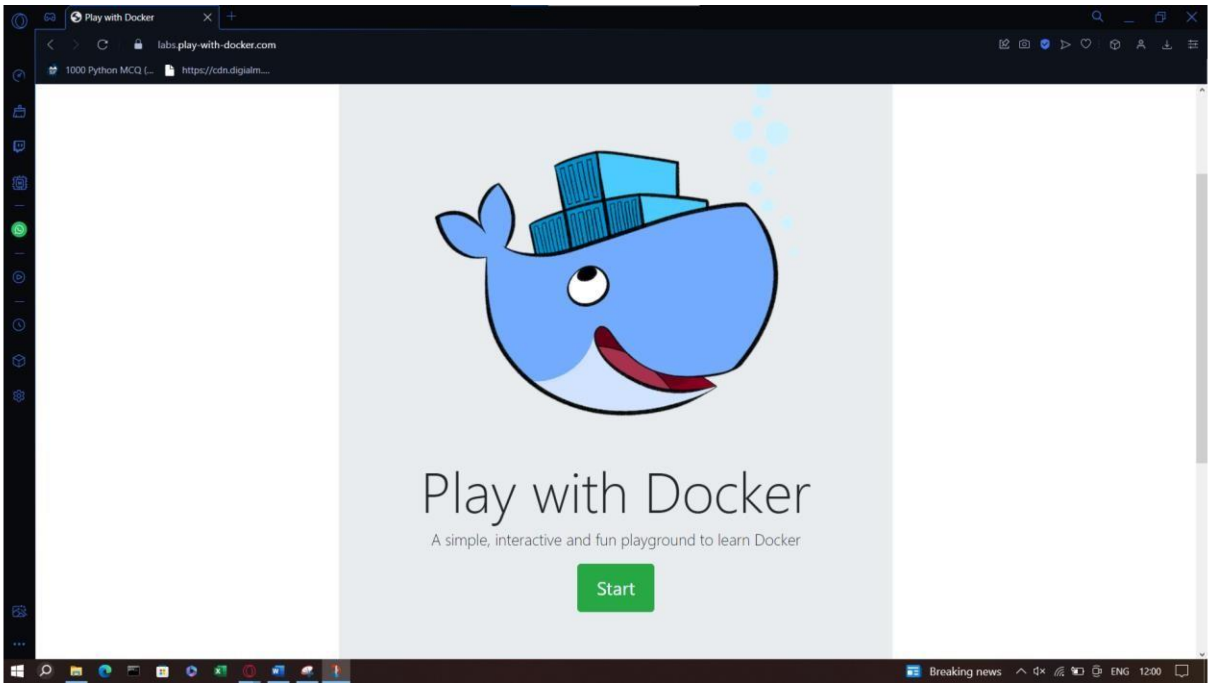
```
[
    {
        "term": "HTML",
        "definition": "Hypertext Markup Language"
    },
    {
        "term": "MVC",
        "definition": "Model View Controller"
    },
    {
        "term": "MVC3",
        "definition": "Updated Definition for Model View Controller 3"
    }
]
```

Practical 3

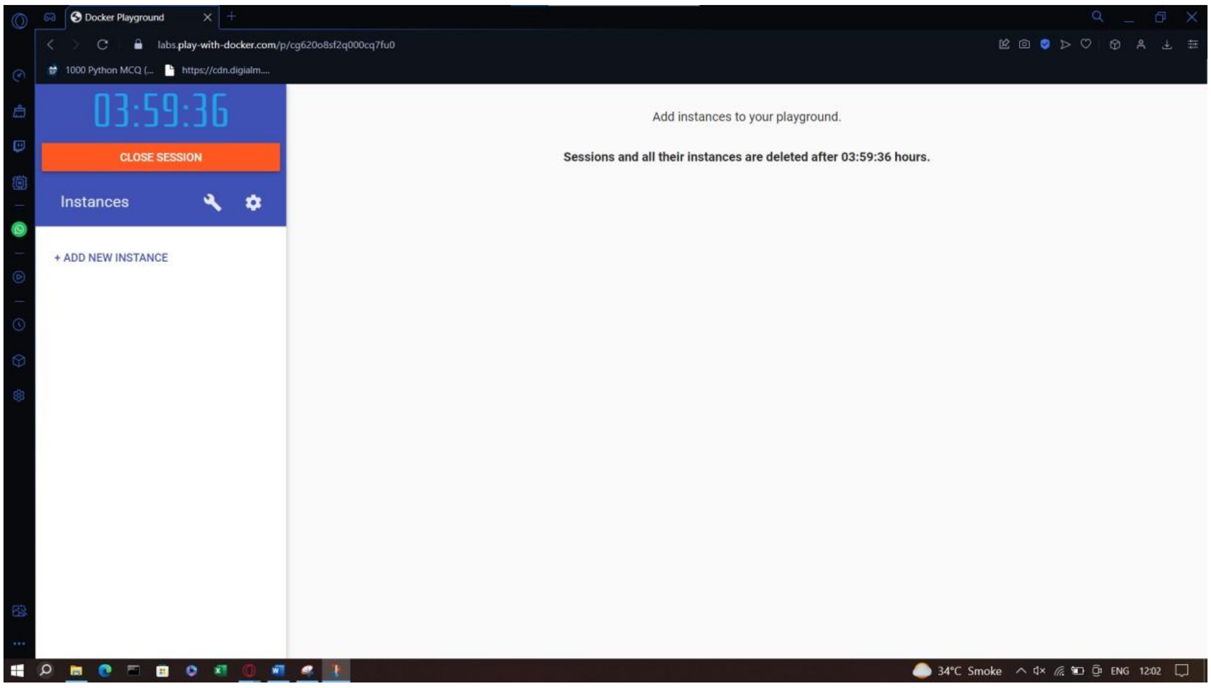
Aim: Working with Docker, Docker Commands, Docker Images and Containers.

Writeup:

Step 1: Create a Docker Hub Account (Sign Up)
Login to this website https://labs.play-with-docker.com/


Play with Docker
labs.play-with-docker.com
Play with Docker
A simple, interactive and fun playground to learn Docker
Start

Step 2: Click on Start and Add a new Instance.



Add instances to your playground.

Sessions and all their instances are deleted after 03:59:36 hours.

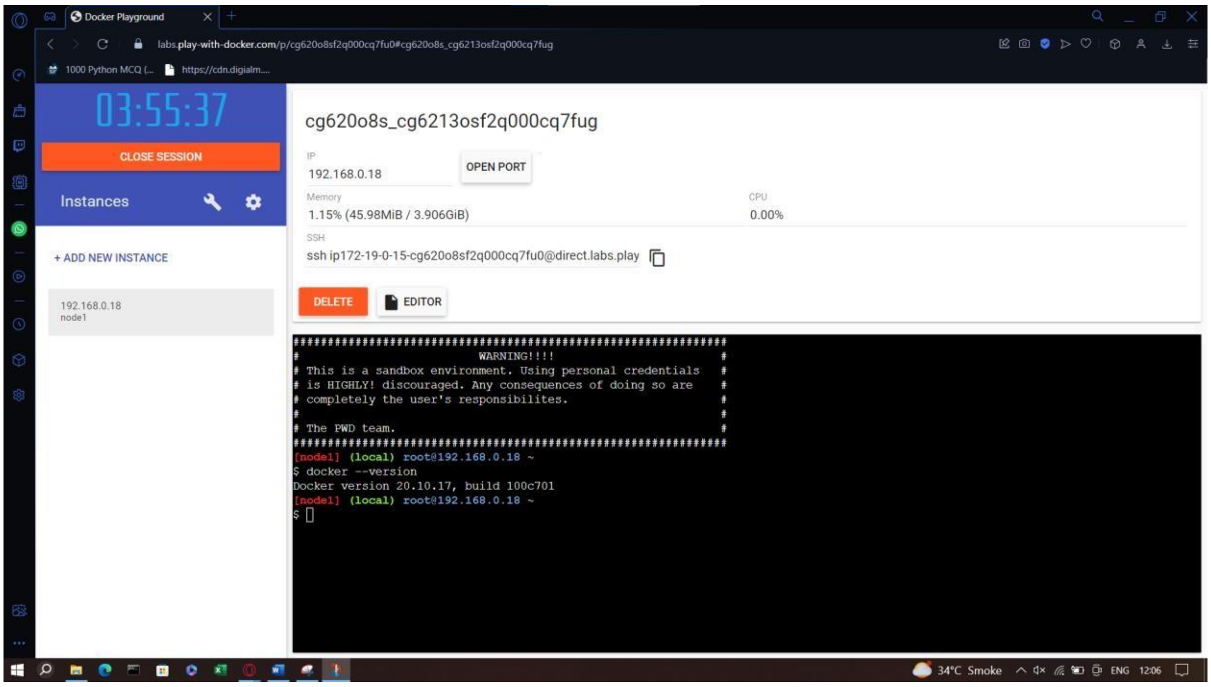
Step 3: Perform the following

Method 1: To pull and push images using docker

Command: To check docker version

docker –version

Output:



cg620o8s_cg6213osf2q000cq7fug

IP 192.168.0.18

Memory 1.15% (45.98MiB / 3.906GiB)

CPU 0.00%

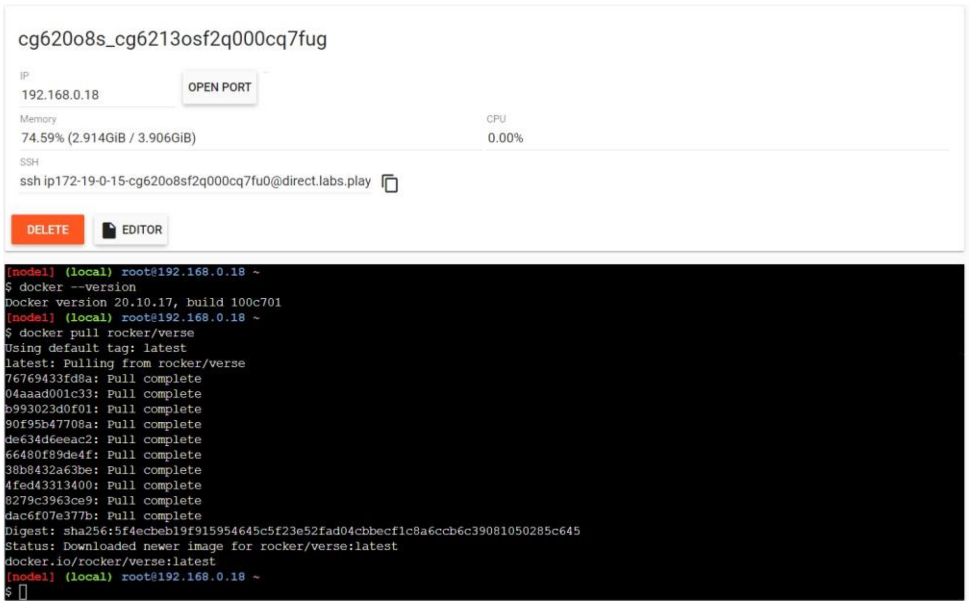
SSH ssh ip172-19-0-15-cg620o8sf2q000cq7fu0@direct.labs.play

```
###################################################################
#                          WARNING!!!!                            #
# This is a sandbox environment. Using personal credentials       #
# is HIGHLY! discouraged. Any consequences of doing so are        #
# completely the user's responsibilites.                          #
#                                                                 #
# The PWD team.                                                   #
###################################################################
[node1] (local) root@192.168.0.18 ~
$ docker --version
Docker version 20.10.17, build 100c701
[node1] (local) root@192.168.0.18 ~
$
```

Command: To pull ready-made images.

docker pull rocker/verse

Output:



cg620o8s_cg6213osf2q000cq7fug

IP 192.168.0.18

Memory 74.59% (2.914GiB / 3.906GiB)

CPU 0.00%

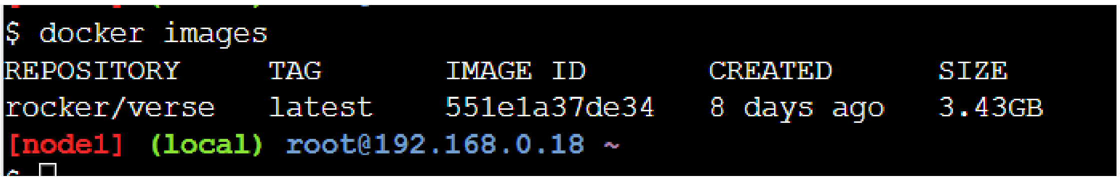
SSH ssh ip172-19-0-15-cg620o8sf2q000cq7fu0@direct.labs.play

```
[node1] (local) root@192.168.0.18 ~
$ docker --version
Docker version 20.10.17, build 100c701
[node1] (local) root@192.168.0.18 ~
$ docker pull rocker/verse
Using default tag: latest
latest: Pulling from rocker/verse
76769433fd8a: Pull complete
04aaad001c33: Pull complete
b993023d0f01: Pull complete
90f95b47708a: Pull complete
de634d6eeac2: Pull complete
66480f89de4f: Pull complete
38b8432a63be: Pull complete
4fed43313400: Pull complete
8279c3963ce9: Pull complete
dac6f07e377b: Pull complete
Digest: sha256:5f4ecbeb19f915954645c5f23e52fad04cbbecf1c8a6ccb6c39081050285c645
Status: Downloaded newer image for rocker/verse:latest
docker.io/rocker/verse:latest
[node1] (local) root@192.168.0.18 ~
$
```

Command: To check images in docker.

docker images

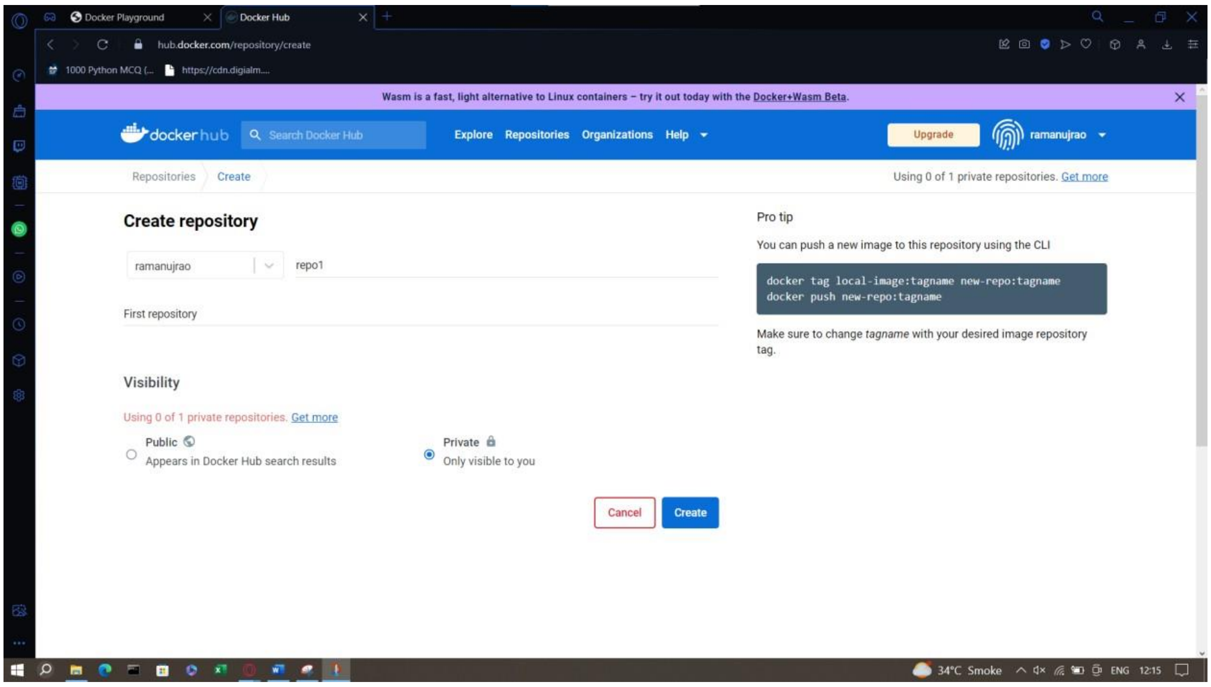
Output:



```
$ docker images
REPOSITORY      TAG        IMAGE ID       CREATED       SIZE
rocker/verse    latest     551e1a37de34   8 days ago    3.43GB
[node1] (local) root@192.168.0.18 ~
```

- Now login to Docker Hub and create a repository.

Output:



hub.docker.com/repository/create

Wasm is a fast, light alternative to Linux containers – try it out today with the Docker+Wasm Beta.

Repositories / Create

Using 0 of 1 private repositories. Get more

Create repository

ramanujrao / repo1

First repository

Visibility

Using 0 of 1 private repositories. Get more

Public — Appears in Docker Hub search results

[✓ Private — Only visible to you]

Cancel Create

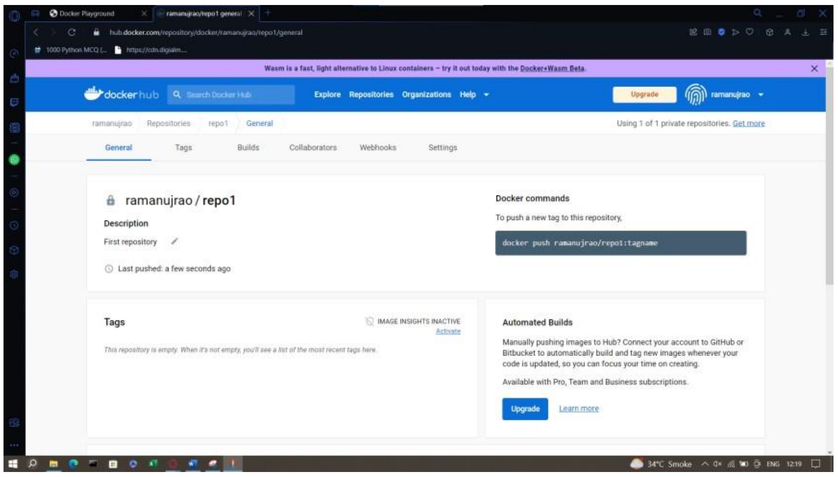
Pro tip

You can push a new image to this repository using the CLI

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to change *tagname* with your desired image repository tag.

- Click on Create Button
- Check to see if the repository has been created



hub.docker.com/repository/docker/ramanujrao/repo1/general

Wasm is a fast, light alternative to Linux containers – try it out today with the Docker+Wasm Beta.

ramanujrao / Repositories / repo1 / General

Using 1 of 1 private repositories. Get more

General | Tags | Builds | Collaborators | Webhooks | Settings

ramanujrao / repo1

Description

First repository

Last pushed: a few seconds ago

Docker commands

To push a new tag to this repository,

```
docker push ramanujrao/repo1:tagname
```

Tags

IMAGE INSIGHTS INACTIVE Activate

This repository is empty. When it's not empty, you'll see a list of the most recent tags here.

Automated Builds

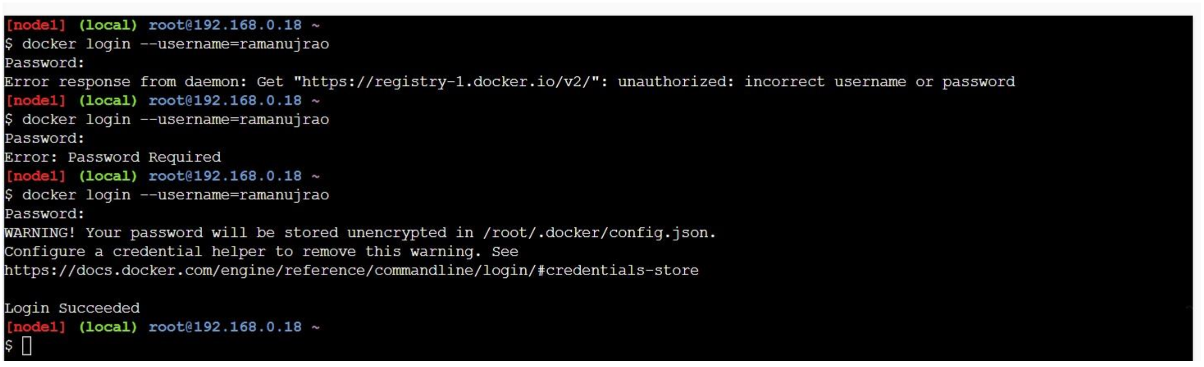
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions.

Upgrade Learn more

Command: to login to your docker account.

docker login --username=ramanujrao
password:

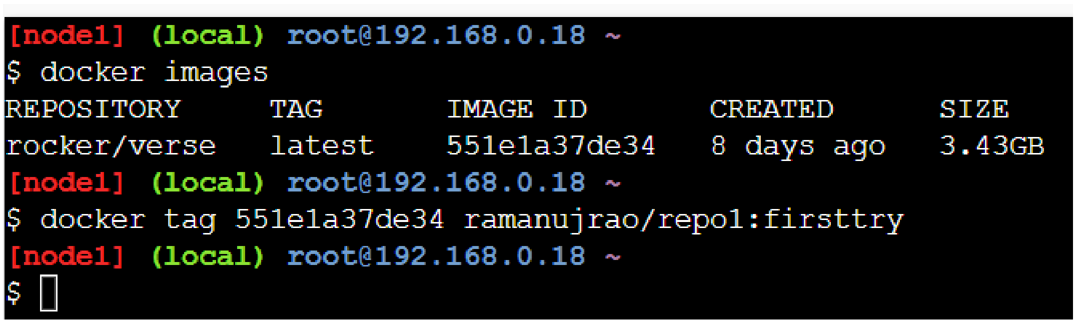


```
[node1] (local) root@192.168.0.18 ~
$ docker login --username=ramanujrao
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
[node1] (local) root@192.168.0.18 ~
$ docker login --username=ramanujrao
Password:
Error: Password Required
[node1] (local) root@192.168.0.18 ~
$ docker login --username=ramanujrao
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[node1] (local) root@192.168.0.18 ~
$ 
```

Command: To tag an image.

docker tag 551e1a37de34 ramanujrao/repo1:firsttry

Output:

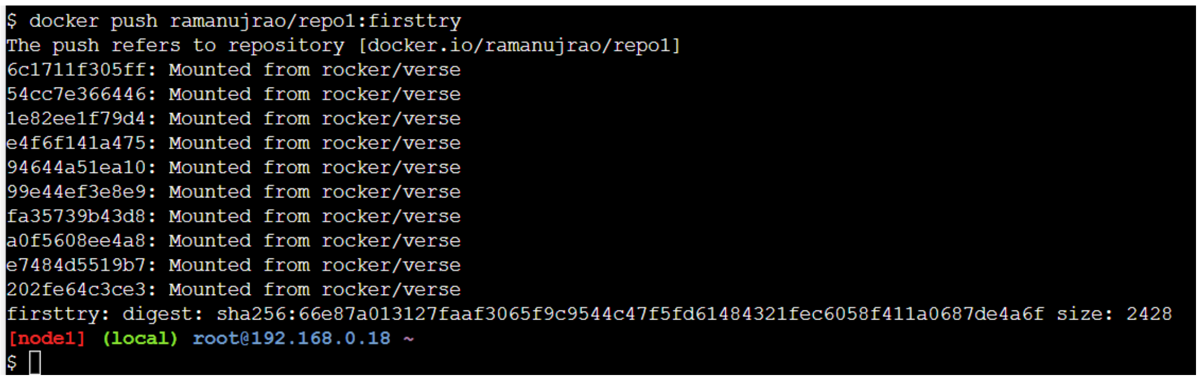


```
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY      TAG       IMAGE ID       CREATED       SIZE
rocker/verse    latest    551e1a37de34   8 days ago    3.43GB
[node1] (local) root@192.168.0.18 ~
$ docker tag 551e1a37de34 ramanujrao/repo1:firsttry
[node1] (local) root@192.168.0.18 ~
$ 
```

Command: To push image to docker hub account.

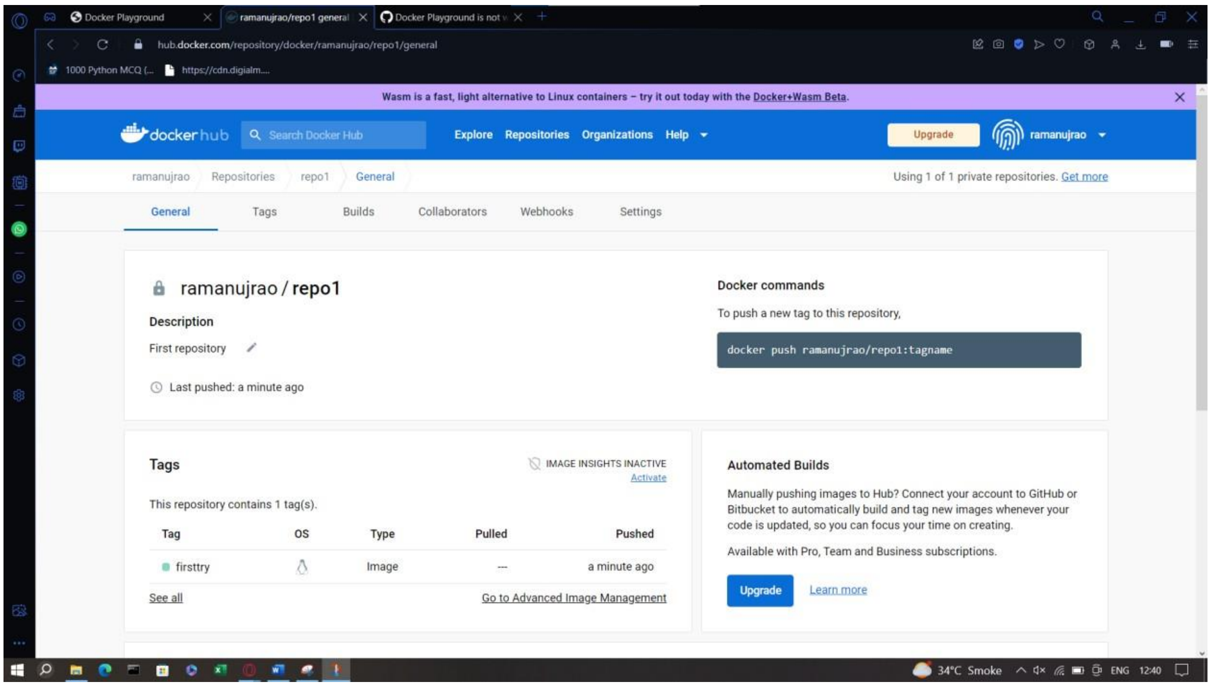
docker push ramanujrao/repo1:firsttry

Output:



```
$ docker push ramanujrao/repo1:firsttry
The push refers to repository [docker.io/ramanujrao/repo1]
6c1711f305ff: Mounted from rocker/verse
54cc7e366446: Mounted from rocker/verse
1e82ee1f79d4: Mounted from rocker/verse
e4f6f141a475: Mounted from rocker/verse
94644a51ea10: Mounted from rocker/verse
99e44ef3e8e9: Mounted from rocker/verse
fa35739b43d8: Mounted from rocker/verse
a0f5608ee4a8: Mounted from rocker/verse
e7484d5519b7: Mounted from rocker/verse
202fe64c3ce3: Mounted from rocker/verse
firsttry: digest: sha256:66e87a013127faaf3065f9c9544c47f5fd61484321fec6058f411a0687de4a6f size: 2428
[node1] (local) root@192.168.0.18 ~
$ 
```

- Check it in Docker Hub now.



Wasm is a fast, light alternative to Linux containers – try it out today with the Docker+Wasm Beta.

ramanujrao / Repositories / repo1 / General — Using 1 of 1 private repositories. Get more

General | Tags | Builds | Collaborators | Webhooks | Settings

ramanujrao / repo1

Description

First repository

Last pushed: a minute ago

Docker commands

To push a new tag to this repository,

```
docker push ramanujrao/repo1:tagname
```

Tags

IMAGE INSIGHTS INACTIVE Activate

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
firsttry		Image	---	a minute ago

See all Go to Advanced Image Management

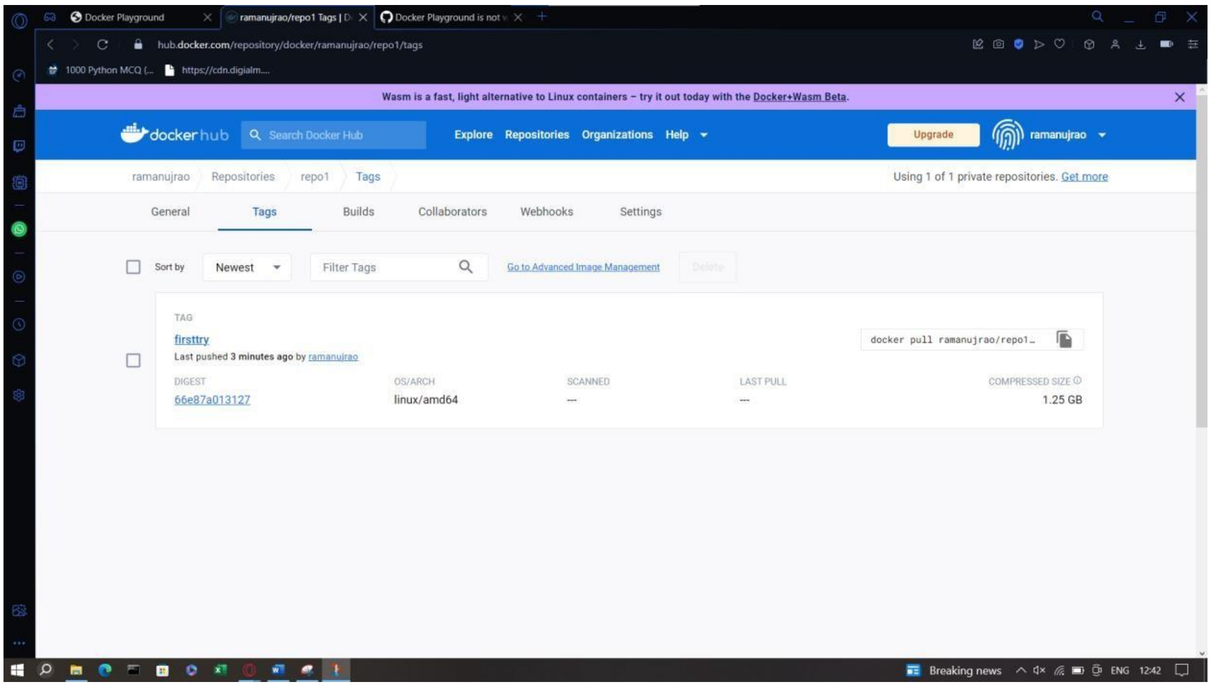
Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions.

Upgrade Learn more

- Click on tags and check



Wasm is a fast, light alternative to Linux containers – try it out today with the Docker+Wasm Beta.

ramanujrao / Repositories / repo1 / Tags — Using 1 of 1 private repositories. Get more

General | Tags | Builds | Collaborators | Webhooks | Settings

Sort by Newest Filter Tags Go to Advanced Image Management Delete

TAG

firsttry

Last pushed 3 minutes ago by ramanujrao

```
docker pull ramanujrao/repo1…
```

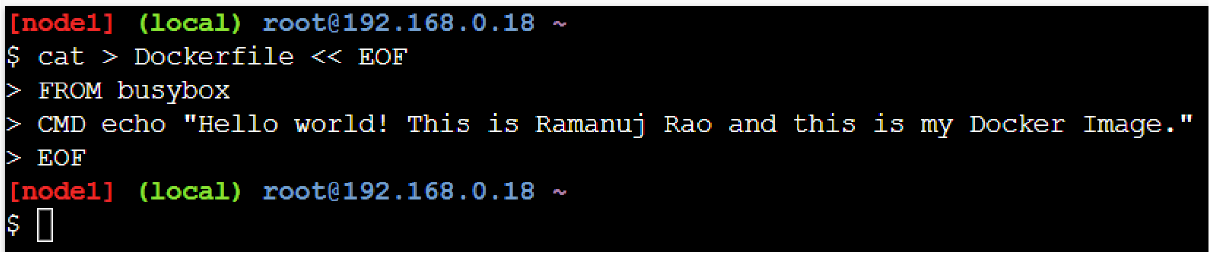
DIGEST	OS/ARCH	SCANNED	LAST PULL	COMPRESSED SIZE
66e87a013127	linux/amd64	---	---	1.25 GB

Method 2: Build an image then push it to docker and run it.

Command: To create a docker file.

cat > Dockerfile <<EOF
FROM busybox
CMD echo "Hello world! This is my first Docker image."
EOF

Output:

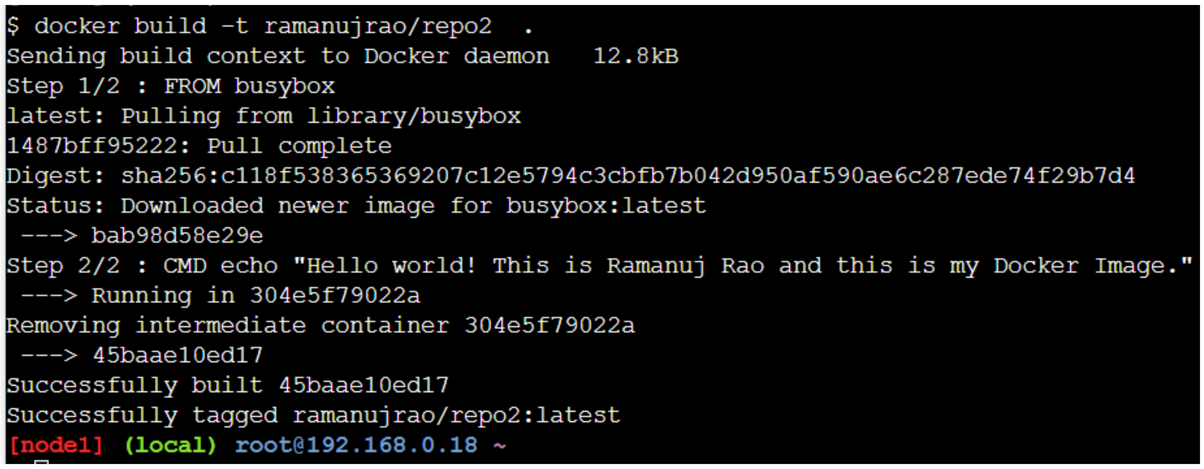


```
[node1] (local) root@192.168.0.18 ~
$ cat > Dockerfile << EOF
> FROM busybox
> CMD echo "Hello world! This is Ramanuj Rao and this is my Docker Image."
> EOF
[node1] (local) root@192.168.0.18 ~
$
```

Command: to build image for a docker file.

docker build –t ramanujrao/repo2

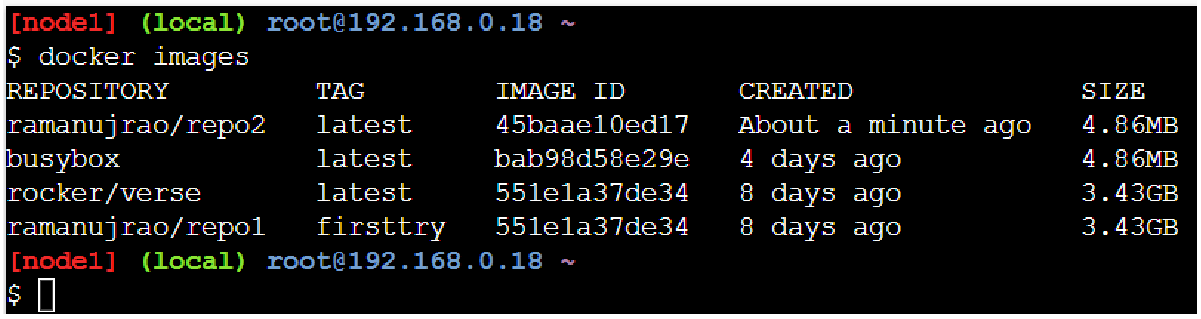
Output:



```
$ docker build -t ramanujrao/repo2  .
Sending build context to Docker daemon   12.8kB
Step 1/2 : FROM busybox
latest: Pulling from library/busybox
1487bff95222: Pull complete
Digest: sha256:c118f538365369207c12e5794c3cbfb7b042d950af590ae6c287ede74f29b7d4
Status: Downloaded newer image for busybox:latest
 ---> bab98d58e29e
Step 2/2 : CMD echo "Hello world! This is Ramanuj Rao and this is my Docker Image."
 ---> Running in 304e5f79022a
Removing intermediate container 304e5f79022a
 ---> 45baae10ed17
Successfully built 45baae10ed17
Successfully tagged ramanujrao/repo2:latest
[node1] (local) root@192.168.0.18 ~
```

Command: To check docker images.

docker images

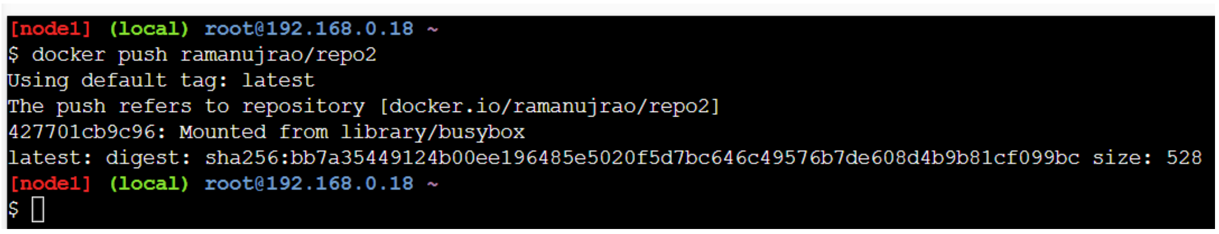


```
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY          TAG         IMAGE ID       CREATED              SIZE
ramanujrao/repo2    latest      45baae10ed17   About a minute ago   4.86MB
busybox             latest      bab98d58e29e   4 days ago           4.86MB
rocker/verse        latest      551e1a37de34   8 days ago           3.43GB
ramanujrao/repo1    firsttry    551e1a37de34   8 days ago           3.43GB
[node1] (local) root@192.168.0.18 ~
$
```

Command: to push image to docker hub.

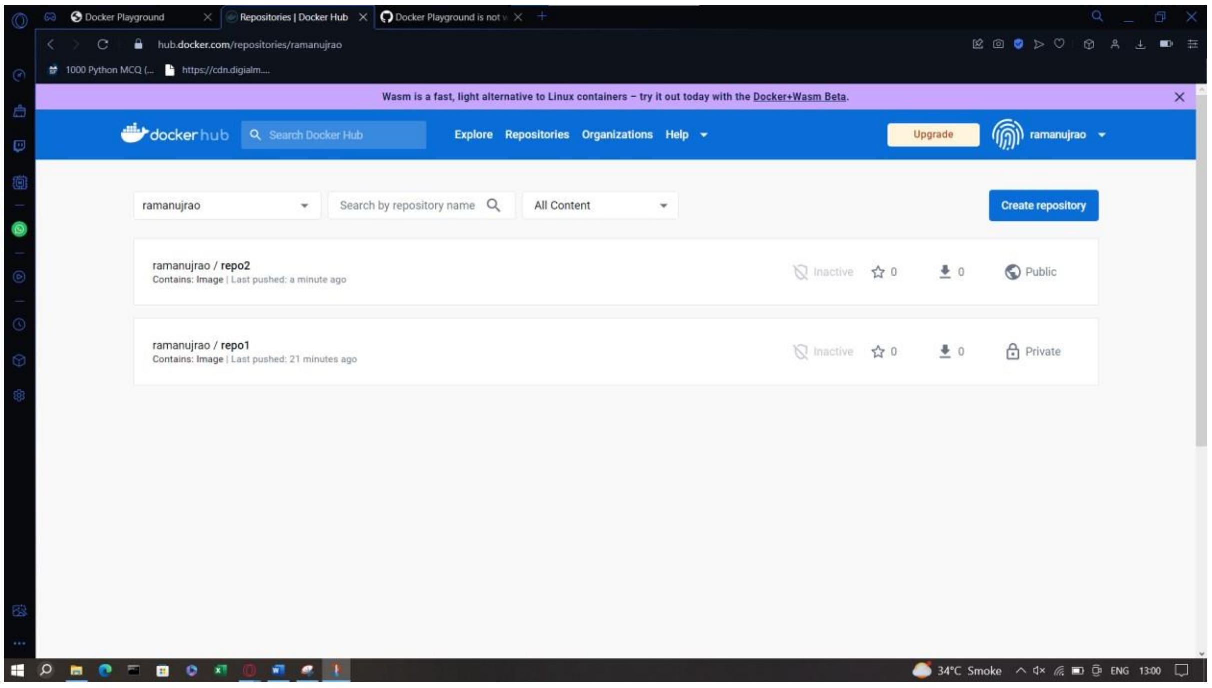
docker push ramanujrao/repo2 .

Output:



```
[node1] (local) root@192.168.0.18 ~
$ docker push ramanujrao/repo2
Using default tag: latest
The push refers to repository [docker.io/ramanujrao/repo2]
427701cb9c96: Mounted from library/busybox
latest: digest: sha256:bb7a35449124b00ee196485e5020f5d7bc646c49576b7de608d4b9b81cf099bc size: 528
[node1] (local) root@192.168.0.18 ~
$ 
```

- Now check in Docker Hub.



Wasm is a fast, light alternative to Linux containers – try it out today with the Docker+Wasm Beta.

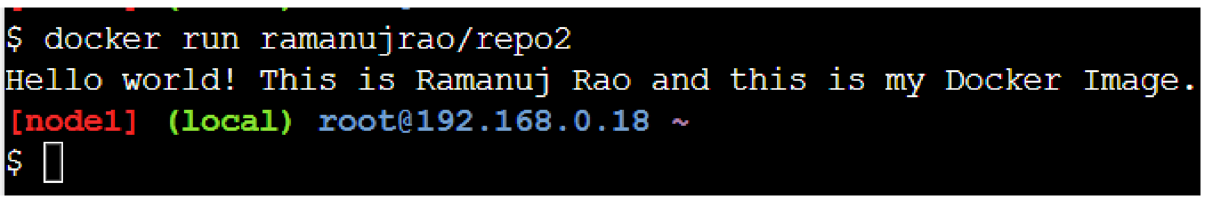
ramanujrao / repo2
Contains: Image | Last pushed: a minute ago   Inactive   0   0   Public

ramanujrao / repo1
Contains: Image | Last pushed: 21 minutes ago   Inactive   0   0   Private

Command: to run docker image:

docker run ramanujrao/repo2

Output:



```
$ docker run ramanujrao/repo2
Hello world! This is Ramanuj Rao and this is my Docker Image.
[node1] (local) root@192.168.0.18 ~
$ 
```

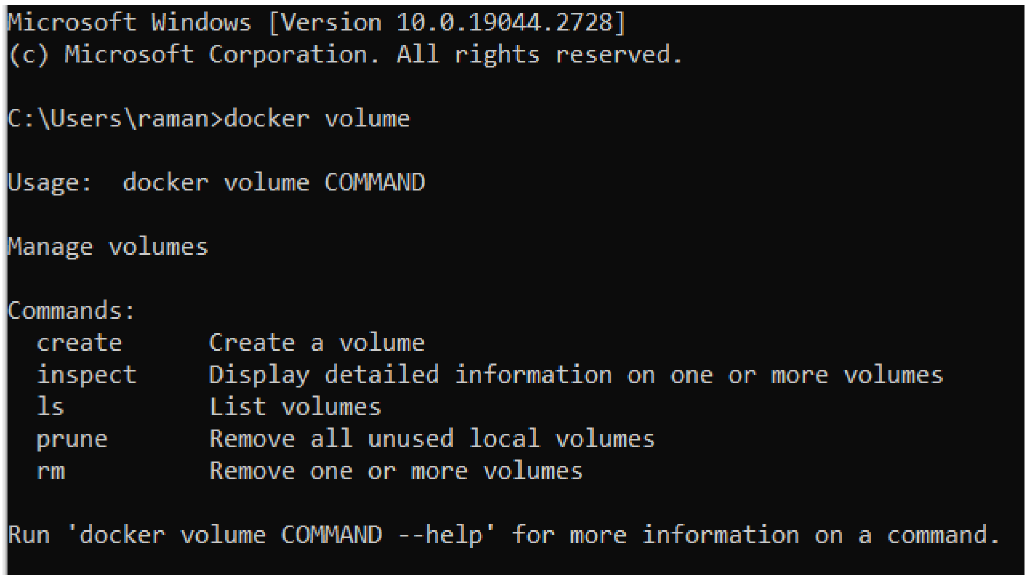
Practical 4

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

Writeup:

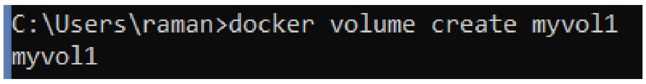
Step 1: Working with Basic Functionalities Docker

[A] : Creating a volume using the **docker volume** command.



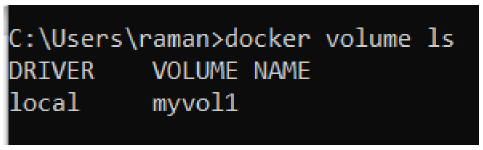
```
Microsoft Windows [Version 10.0.19044.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\raman>docker volume

Usage:  docker volume COMMAND

Manage volumes

Commands:
  create      Create a volume
  inspect     Display detailed information on one or more volumes
  ls          List volumes
  prune       Remove all unused local volumes
  rm          Remove one or more volumes

Run 'docker volume COMMAND --help' for more information on a command.
```

[B] : Creating the Actual Volume using command **docker volume create myvol1**



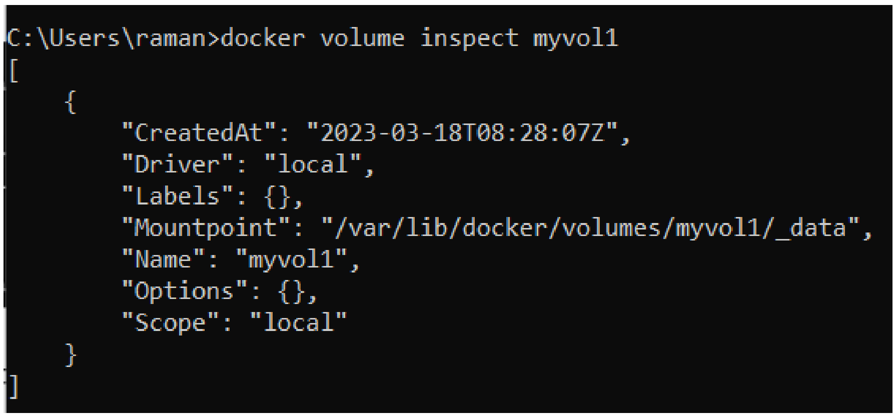
```
C:\Users\raman>docker volume create myvol1
myvol1
```

[C] : To list the volume we will write the command **docker volume ls**



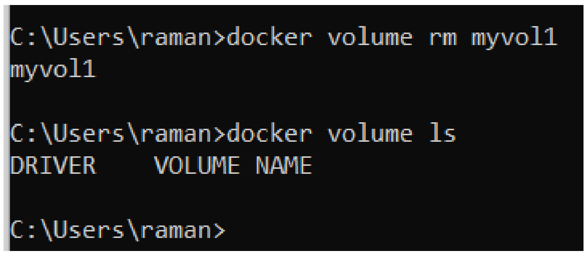
```
C:\Users\raman>docker volume ls
DRIVER    VOLUME NAME
local     myvol1
```

[D] : To get the details of our volume we have to write the command **docker volume inspect myvol1**



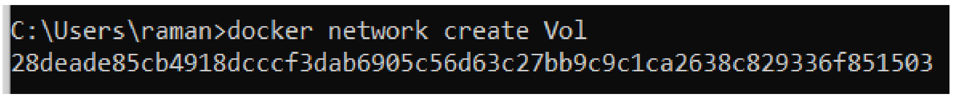
```
C:\Users\raman>docker volume inspect myvol1
[
    {
        "CreatedAt": "2023-03-18T08:28:07Z",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/myvol1/_data",
        "Name": "myvol1",
        "Options": {},
        "Scope": "local"
    }
]
```

[E] **:** To remove your volume you can use the command **docker volume rm myvol1**
Also using **docker volume ls** to confirm that the volume has been removed.

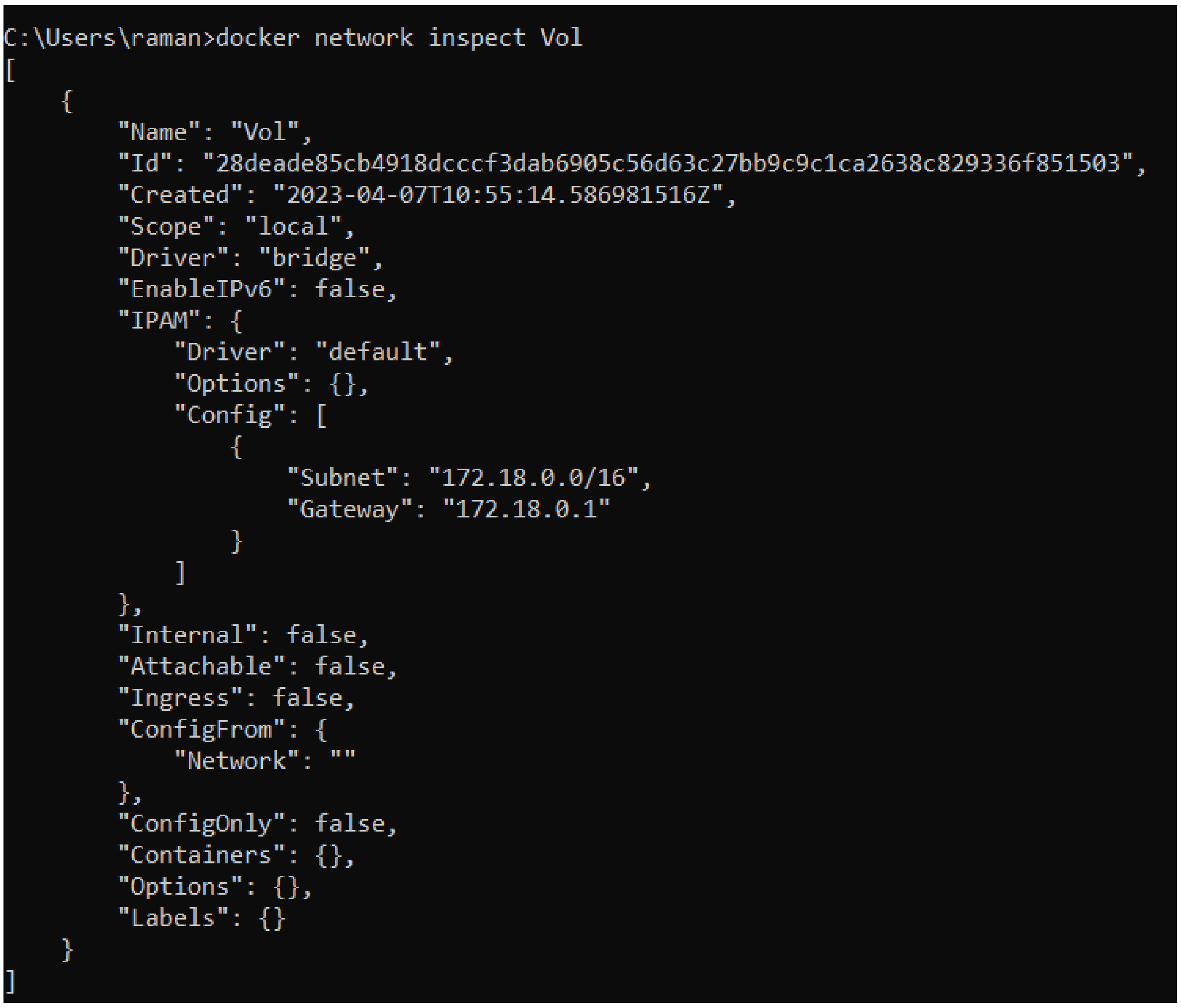

```
C:\Users\raman>docker volume rm myvol1
myvol1

C:\Users\raman>docker volume ls
DRIVER    VOLUME NAME

C:\Users\raman>
```

Step 2: Working with Docker Network

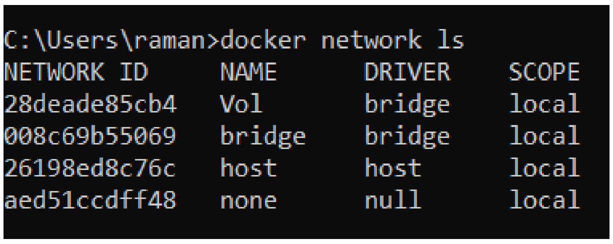
[A] : To Connect a container to a network using command **docker network create Vol**


```
C:\Users\raman>docker network create Vol
28deade85cb4918dcccf3dab6905c56d63c27bb9c9c1ca2638c829336f851503
```

[B] : To get details of a container from a network using command **docker network inspect Vol**

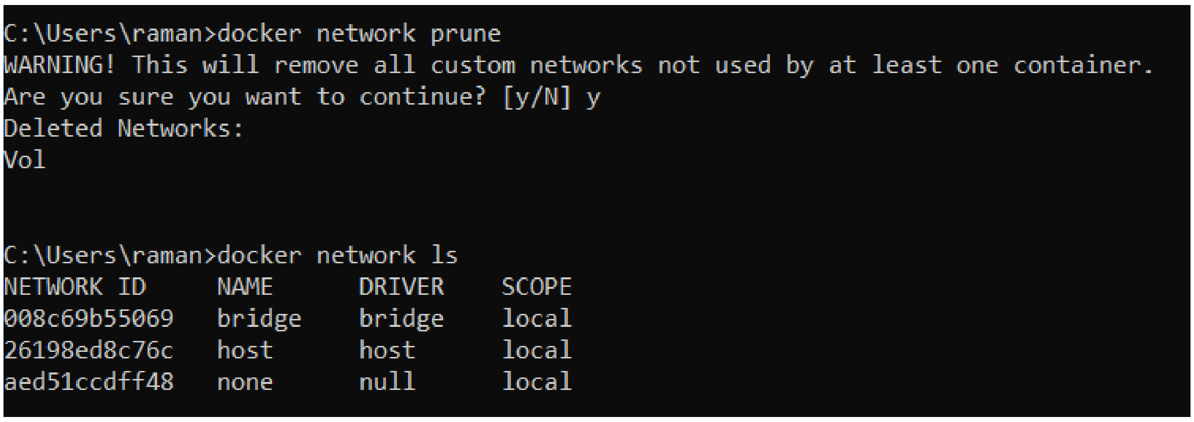

```
C:\Users\raman>docker network inspect Vol
[
    {
        "Name": "Vol",
        "Id": "28deade85cb4918dcccf3dab6905c56d63c27bb9c9c1ca2638c829336f851503",
        "Created": "2023-04-07T10:55:14.586981516Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "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": {}
    }
]
```

[C] : To see the list of networks use command **docker network ls**



```
C:\Users\raman>docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
28deade85cb4   Vol       bridge    local
008c69b55069   bridge    bridge    local
26198ed8c76c   host      host      local
aed51ccdff48   none      null      local
```

[D] : To remove all unused networks using the command **docker network prune**
Also using **docker network ls** to confirm the removal of the network.



```
C:\Users\raman>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:
Vol


C:\Users\raman>docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
008c69b55069   bridge    bridge    local
26198ed8c76c   host      host      local
aed51ccdff48   none      null      local
```

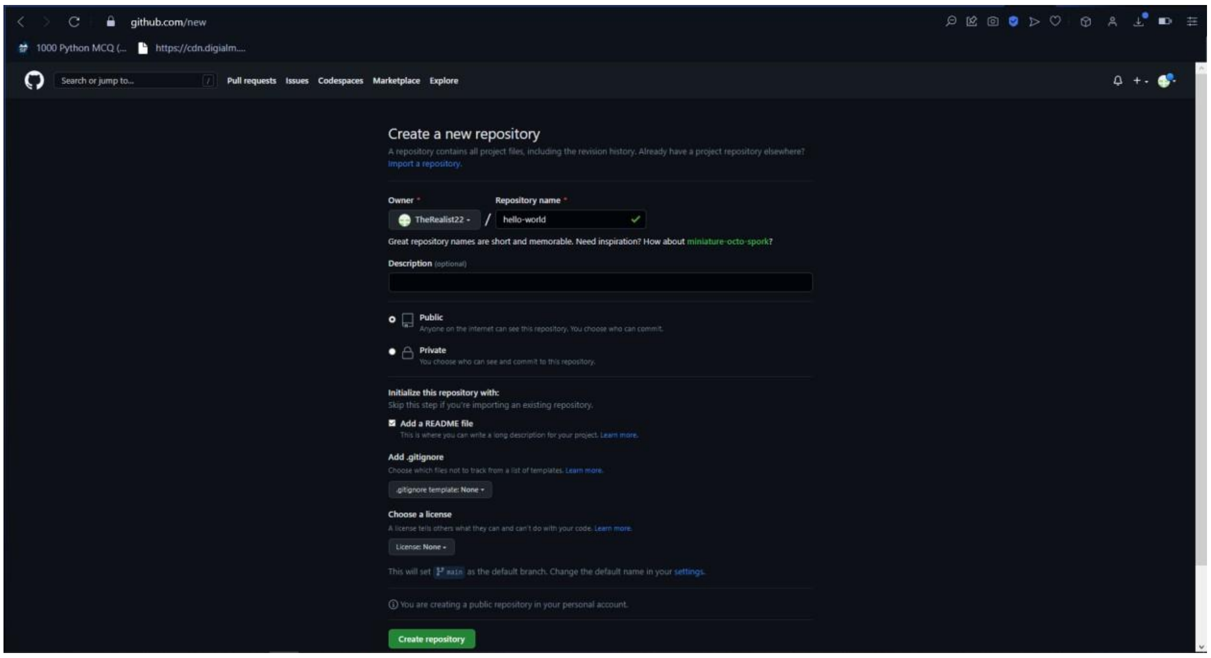
Practical 5

Aim: Working with Circle CI for continuous integration.

Writeup:

Step 1: Create a repository

- Log in to GitHub and begin the process to create a new repository.
- Enter a name for your repository (for example, hello-world).
- Select the option to initialize the repository with a README file.
- Finally, click Create repository.
- There is no need to add any source code for now.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * TheRealist22 / Repository name * hello-world

Great repository names are short and memorable. Need inspiration? How about miniature-octo-spork?

Description (optional)

Public — Anyone on the internet can see this repository. You choose who can commit.

Private — You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

[✓] Add a README file — This is where you can write a long description for your project. Learn more.

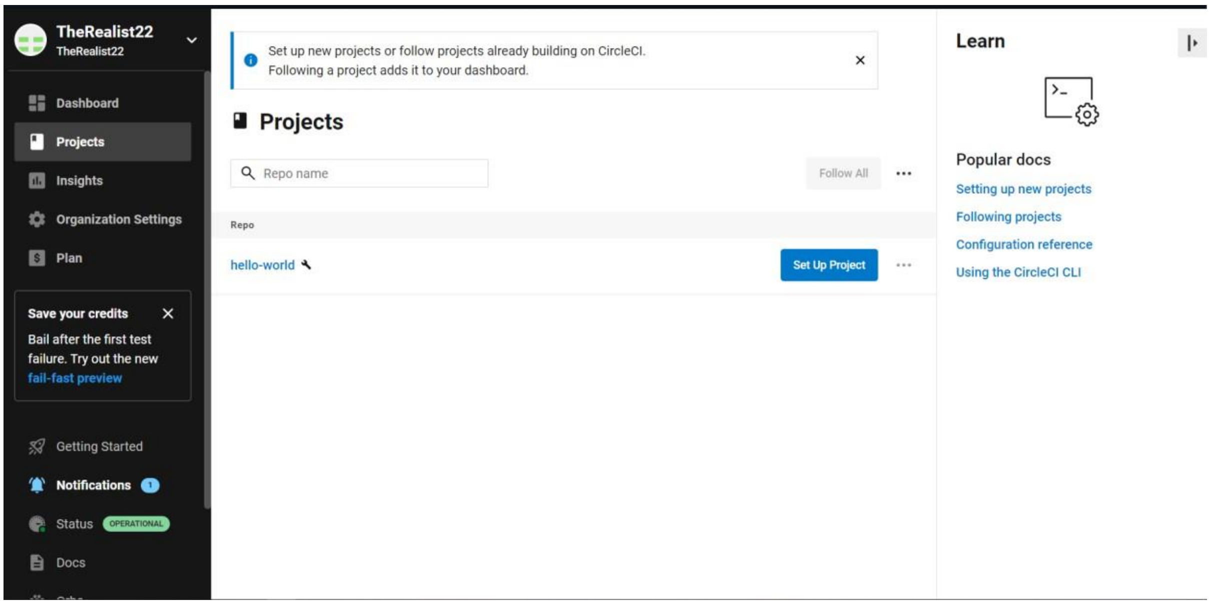
Add .gitignore — Choose which files not to track from a list of templates. Learn more. .gitignore template: None

Choose a license — A license tells others what they can and can't do with your code. Learn more. License: None

This will set main as the default branch. Change the default name in your settings.

You are creating a public repository in your personal account.

- Login to Circle CI https://app.circleci.com/ Using GitHub Login, Once logged in navigate to Projects.



Save your credits — Bail after the first test failure. Try out the new fail-fast preview

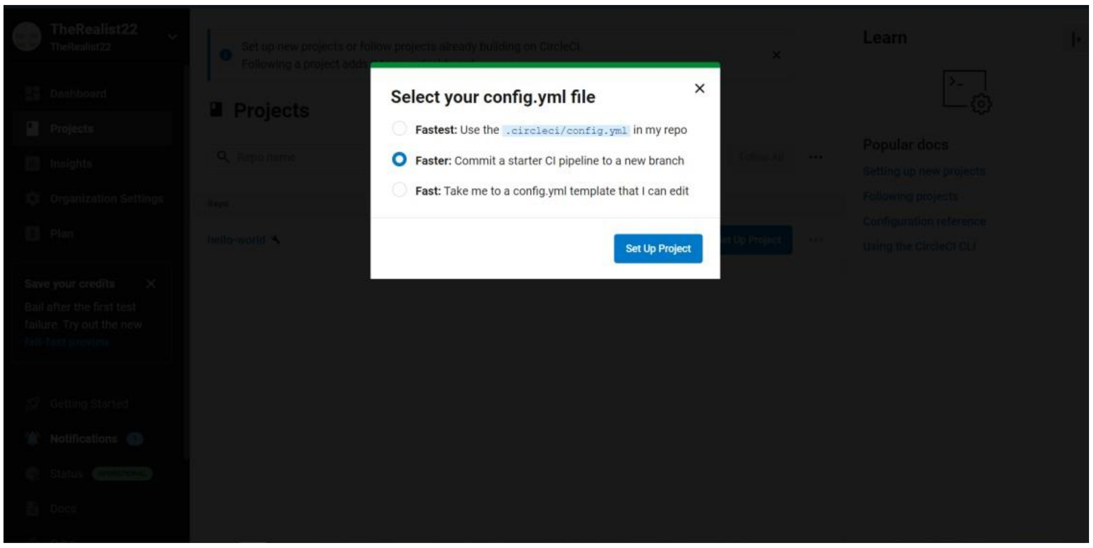
Set up new projects or follow projects already building on CircleCI. Following a project adds it to your dashboard.

Projects

Repo
hello-world

Step 2: Setup CircleCI

- Navigate to the CircleCI Projects page. If you created your new repository under an organization, you will need to select the organization name.
- You will be taken to the Projects dashboard. On the dashboard, select the project you want to set up (hello-world).
- Select the option to commit a starter CI pipeline to a new branch, and click Set Up Project. This will create a file.circleci/config.yml at the root of your repository on a new branch called circleci-project-setup.


Select your config.yml file

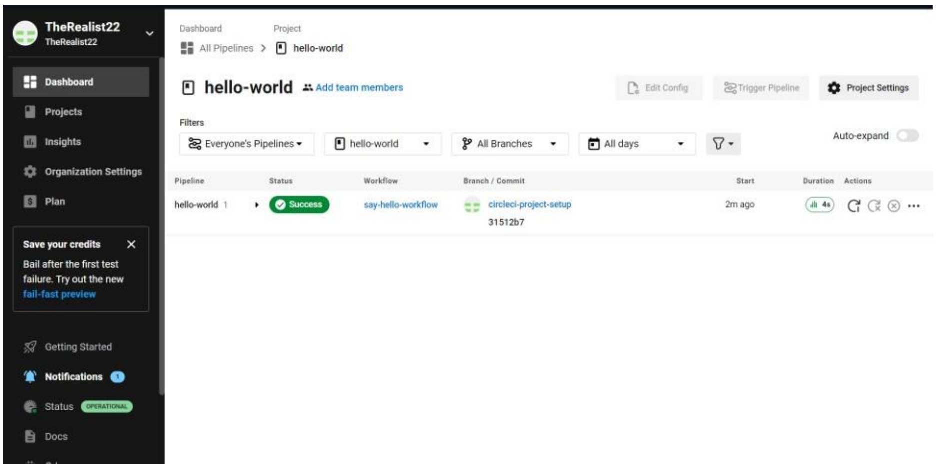
- **Fastest:** Use the `.circleci/config.yml` in my repo
- [✓ **Faster:** Commit a starter CI pipeline to a new branch]
- **Fast:** Take me to a config.yml template that I can edit

Step 3: First Pipeline

- On your project's pipeline page, click the green Success button, which brings you to the workflow that ran (say-hello-workflow).

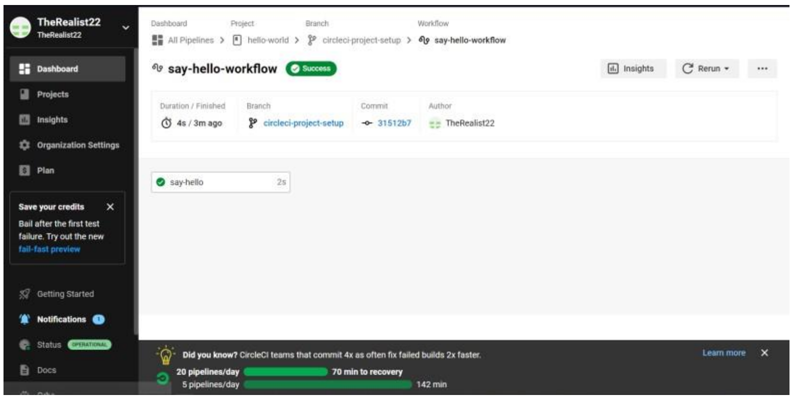
- Within this workflow, the pipeline ran one job, called say-hello. Click say-hello to see the steps in this job:
  a. Spin up environment
  b. Preparing environment variables
  c. Checkout code
  d. Say hello

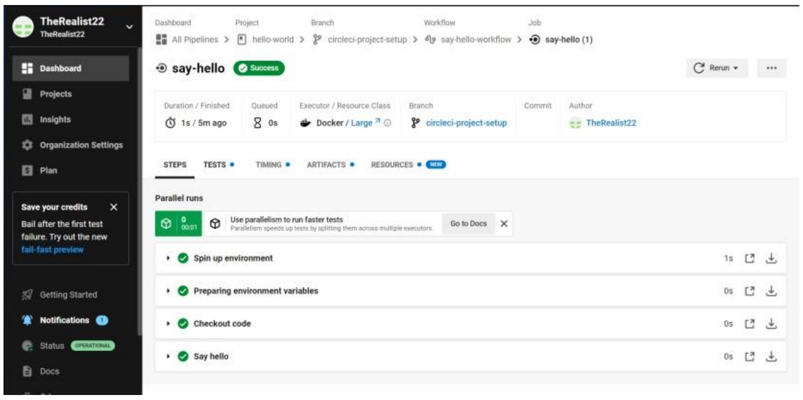
- Now select the "say-hello-workflow" to the right of Success status column.



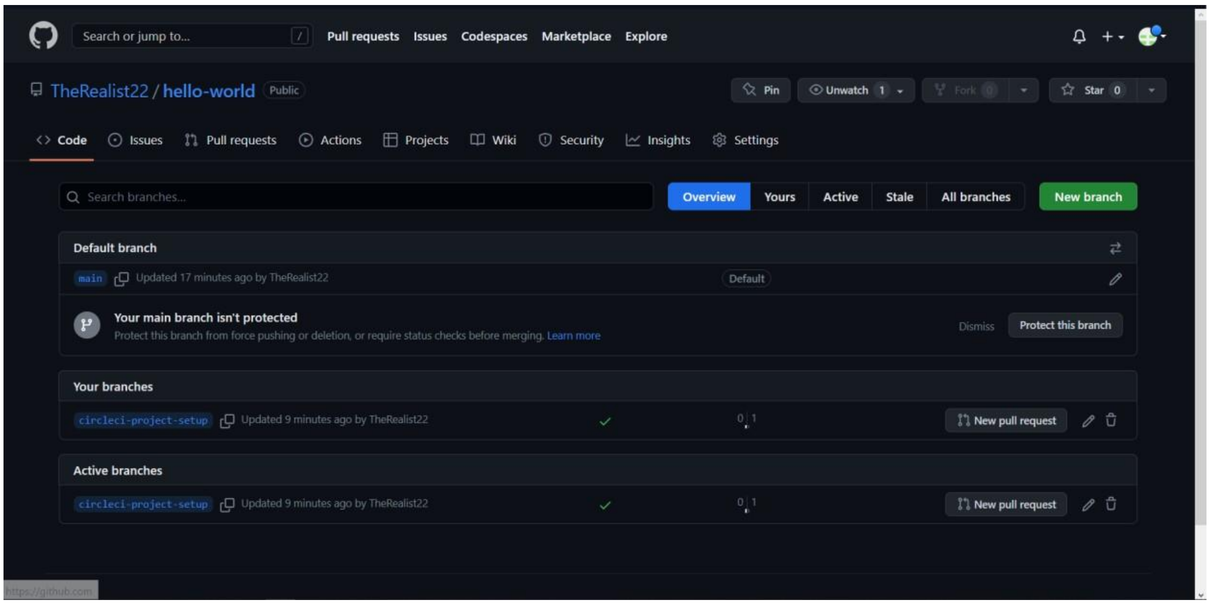
Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
hello-world 1	Success	say-hello-workflow	circleci-project-setup 31512b7	2m ago	4s	

- Select “say-hello” Job with a green tick.


All Pipelines > hello-world > circleci-project-setup > say-hello-workflow
say-hello-workflow Success
Insights
Rerun
Duration / Finished 4s / 3m ago
Branch circleci-project-setup
Commit 31512b7
Author TheRealist22
say-hello 2s
Did you know? CircleCI teams that commit 4x as often fix failed builds 2x faster.
20 pipelines/day
70 min to recovery
5 pipelines/day
142 min
Learn more


All Pipelines > hello-world > circleci-project-setup > say-hello-workflow > say-hello (1)
say-hello Success
Rerun
Duration / Finished 1s / 5m ago
Queued 0s
Executor / Resource Class Docker / Large
Branch circleci-project-setup
Author TheRealist22
STEPS
TESTS
TIMING
ARTIFACTS
RESOURCES NEW
Parallel runs
Use parallelism to run faster tests
Go to Docs
Spin up environment 1s
Preparing environment variables 0s
Checkout code 0s
Say hello 0s

- Select Branch and option circleci-project-setup



Default branch

main Updated 17 minutes ago by TheRealist22

Your main branch isn't protected

Protect this branch from force pushing or deletion, or require status checks before merging. Learn more

Your branches

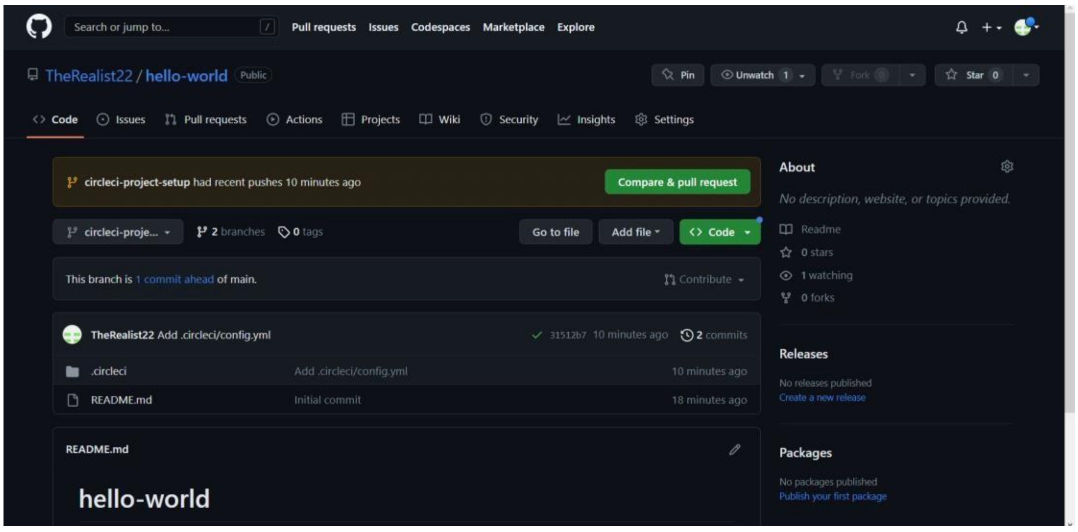
circleci-project-setup Updated 9 minutes ago by TheRealist22 0 | 1

Active branches

circleci-project-setup Updated 9 minutes ago by TheRealist22 0 | 1

Step 4: Break your build.

- In this section, you will edit the .circleci/config.yml file and see what happens if a build does not complete successfully.
- It is possible to edit files directly on GitHub.



circleci-project-setup had recent pushes 10 minutes ago

This branch is 1 commit ahead of main.

TheRealist22 Add .circleci/config.yml ✓ 31512b7 10 minutes ago 2 commits

.circleci	Add .circleci/config.yml	10 minutes ago
README.md	Initial commit	18 minutes ago

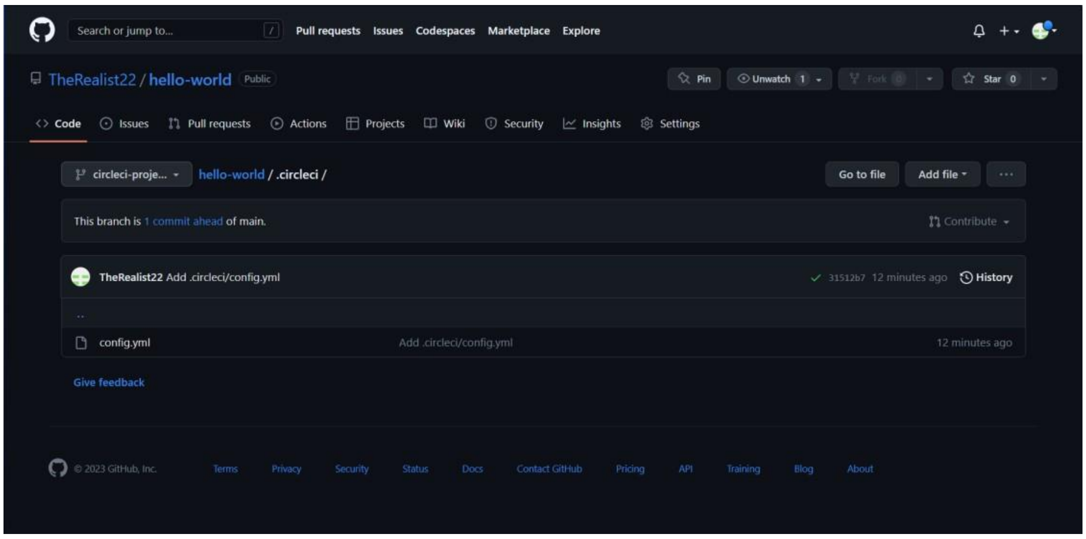
README.md

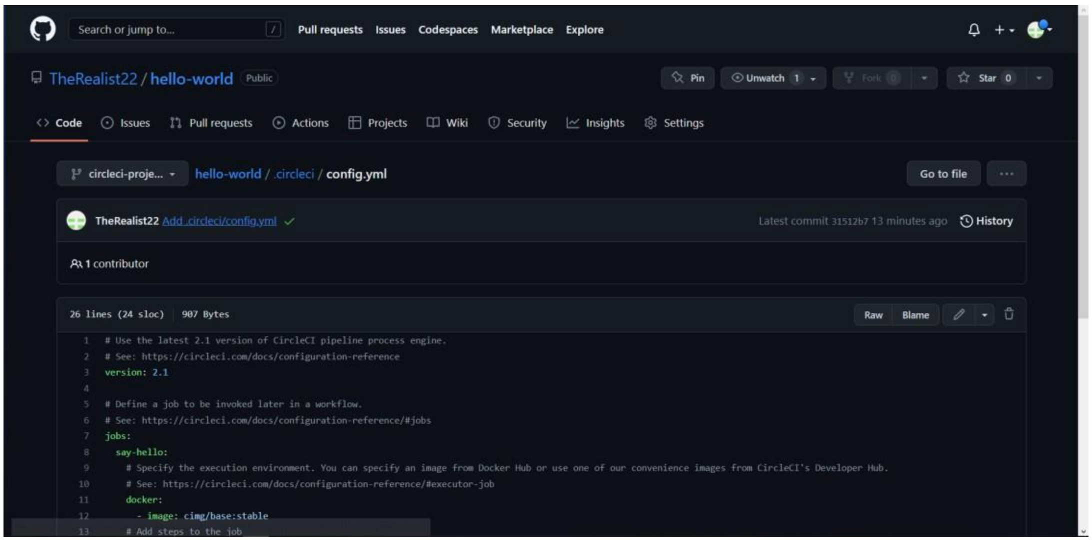
hello-world

No description, website, or topics provided.

No releases published
Create a new release

No packages published
Publish your first package


Search or jump to...
Pull requests Issues Codespaces Marketplace Explore
TheRealist22 / hello-world Public
Pin Unwatch 1 Fork 0 Star 0
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
circleci-proje... hello-world / .circleci /
Go to file Add file
This branch is 1 commit ahead of main.
Contribute
TheRealist22 Add .circleci/config.yml
31512b7 12 minutes ago History
config.yml Add .circleci/config.yml 12 minutes ago
Give feedback
© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

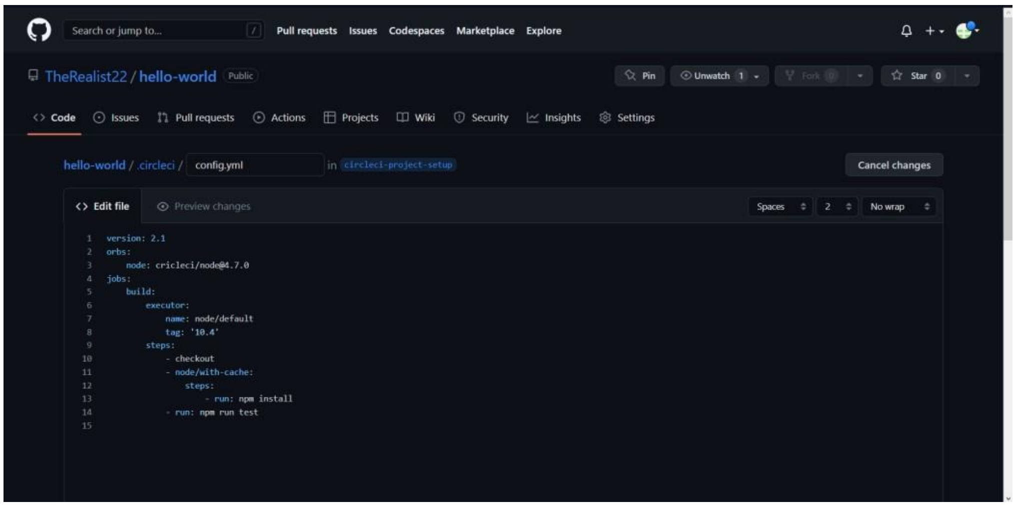

Search or jump to...
Pull requests Issues Codespaces Marketplace Explore
TheRealist22 / hello-world Public
Pin Unwatch 1 Fork 0 Star 0
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
circleci-proje... hello-world / .circleci / config.yml
Go to file
TheRealist22 Add .circleci/config.yml
Latest commit 31512b7 13 minutes ago History
1 contributor
26 lines (24 sloc) 907 Bytes
Raw Blame

```
# Use the latest 2.1 version of CircleCI pipeline process engine.
# See: https://circleci.com/docs/configuration-reference
version: 2.1

# Define a job to be invoked later in a workflow.
# See: https://circleci.com/docs/configuration-reference/#jobs
jobs:
  say-hello:
    # Specify the execution environment. You can specify an image from Docker Hub or use one of our convenience images from CircleCI's Developer Hub.
    # See: https://circleci.com/docs/configuration-reference/#executor-job
    docker:
      - image: cimg/base:stable
    # Add steps to the job
```

- Replace the existing code with new code:

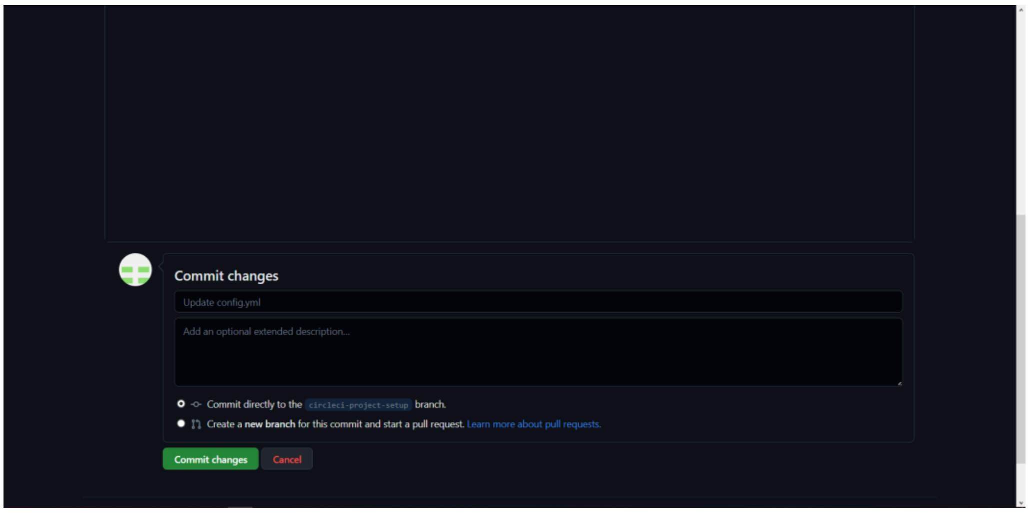
```
version: 2.1
orbs:
    node: cricleci/node@4.7.0
jobs:
    build:
        executor:
            name: node/default
            tag: '10.4'
        steps:
            - checkout
            - node/with-cache:
                steps:
                    - run: npm install
            - run: npm run test
```

- The Github File Editor should look like this


TheRealist22 / hello-world
hello-world / .circleci / config.yml in circleci-project-setup
Cancel changes
Edit file
Preview changes

```yaml
version: 2.1
orbs:
  node: cricleci/node@4.7.0
jobs:
  build:
    executor:
      name: node/default
      tag: '10.4'
    steps:
      - checkout
      - node/with-cache:
          steps:
            - run: npm install
      - run: npm run test
```

- Scroll down and Commit your changes on GitHub


Commit changes
Update config.yml
Add an optional extended description...
Commit directly to the circleci-project-setup branch.
Create a new branch for this commit and start a pull request. Learn more about pull requests.
Commit changes
Cancel

- After committing your changes, then return to the Projects page in CircleCI. You should see a new pipeline running... and it will fail! What's going on? The Node orb runs some common Node tasks. Because you are working with an empty repository, running npm run test, a Node script, causes the configuration to fail. To fix this, you need to set up a Node project in your repository.