



PRACTICAL JOURNAL

in

MICRO SERVICES ARCHITECTURE

Submitted by

KFMSCIT005 HITESH VERSHI BHANUSHALI

for the award of the Degree of

MASTERS OF SCIENCE (INFORMATION TECHNOLOGY)

PART – I

DEPARTMENT OF INFORMATION TECHNOLOGY

KISHINCHAND CHELLARAM COLLEGE

(Affiliated to University of HSNCU)

MUMBAI, 400020

MAHARASHTRA

2023-24

SUBJECT CODE - MS-FIT - 201

MICRO SERVICES ARCHITECTURE



KISHINCHAND CHELLARAM COLLEGE

CHURCHGATE, MUMBAI – 400 020.

DEPARTMENT OF INFORMATION TECHNOLOGY

M.SC.I.T PART- I

CERTIFICATE

This is to certify that the Practical conducted by

Mr. **HITESH VERSHI BHANUSHALI** for M.Sc. (IT) Part- I Semester- II, Seat No: **KFMSCIT005** at Kishinchand Chellaram College in partial fulfillment for the MASTER OF SCIENCE (INFORMATION TECHNOLOGY). Degree Examination for Semester II has been periodically examined and signed, and the course of term work has been satisfactorily carried out for the year 2023 - 2024. This Practical journal had not been submitted for any other examination and does not form part of any other course undergone by the candidate.

Signature

Lecturer-In-Charge

Guided By

Signature

External Examiner

Examined By

Signature

Course Coordination

Certified By

College Stamp

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20
M.Sc (I.T.) Part-1 Semester II

INDEX

Sr. No.	TITLE	Date	Page No.	Signature
	PRACTICALS			
1	Building ASP.NET Core MVC Application	27/1/24	01	
2	Building ASP. NET Core REST API	10/2/24	04	
3	Working with Docker, Docker commands, Docker images & Containers.	10/2/24	11	
4	Installing s/w packages on Docker.	17/2/24	16	
5	Working with Docker volumes & networks.	17/2/24	17	
6	Working with Circle CI for continuous integration.	9/3/24	21	
7	Creating Backing service with ASP.NET 2.0 core.	6/4/24	23	
8	Working with Kubernetes.	6/4/24	26	

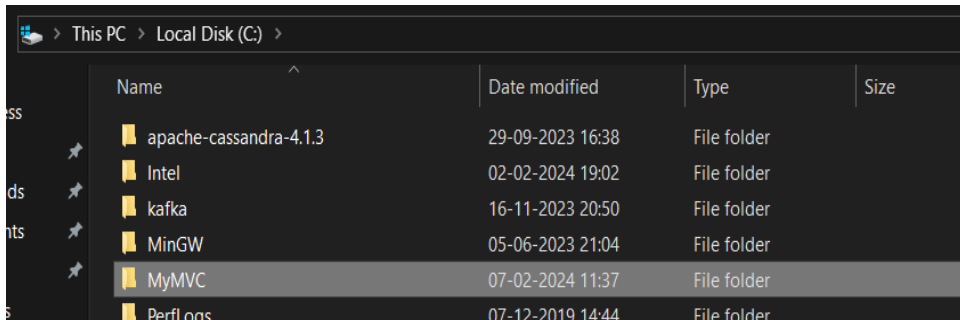
KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

PRACTICAL - 1: ASP.NET CORE MVC

Aim: Building a MVC (Model – View - Control).

Step 1: create a folder.



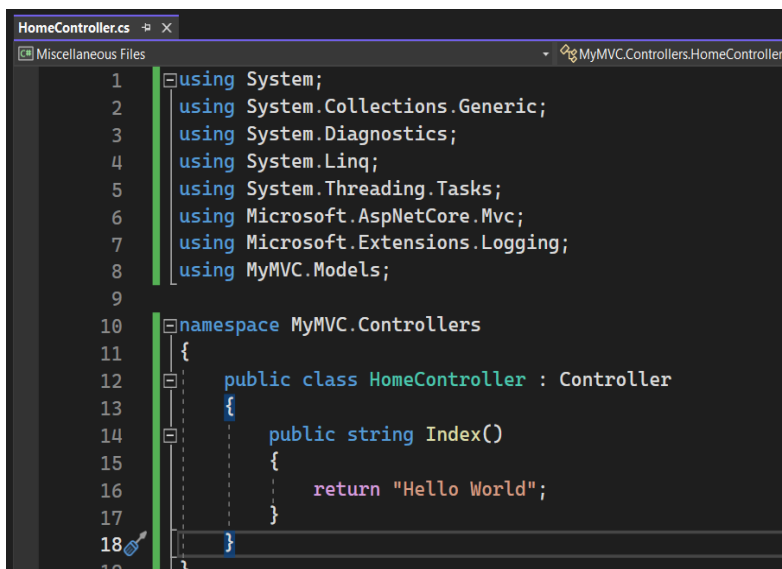
Step 2: create MVC project using “dotnet new mvc --auth none”.

```
C:\Windows\System32\cmd.exe - dotnet run
Microsoft Windows [Version 10.0.19045.3930]
(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-faq for details.

Processing post-creation actions...
Restoring C:\MyMVC\MyMVC.csproj:
  Determining projects to restore...
  Restored C:\MyMVC\MyMVC.csproj (in 88 ms).
Restore succeeded.
```

Step 3: Go to controllers folder and modify HomeController.cs file.

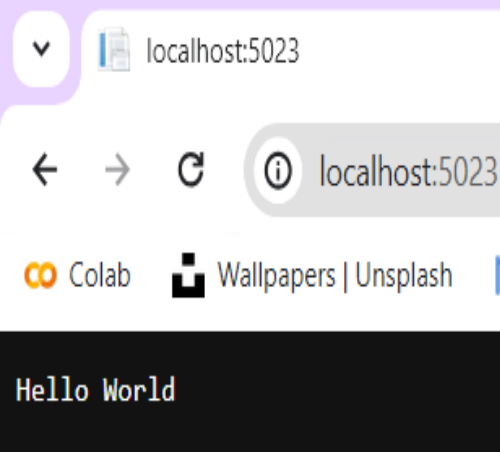


KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

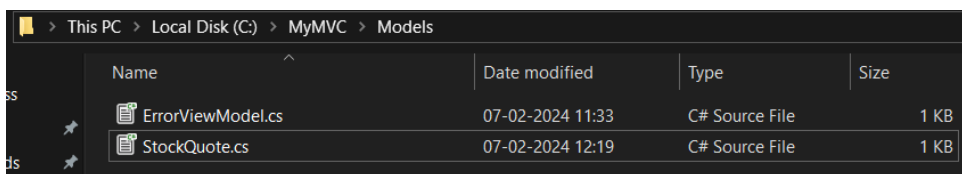
Step 4: Run the project using “dotnet run”.

```
C:\MyMVC>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
       Now listening on: http://localhost:5023
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
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
       Failed to determine the https port for redirect.
```

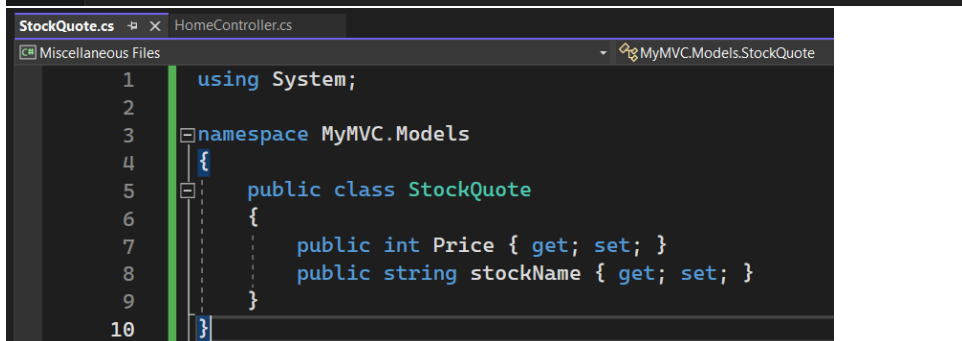


The screenshot shows a web browser window with the address bar set to 'localhost:5023'. The page content displays 'Hello World' in a large, bold, black font. The browser's address bar includes navigation buttons (back, forward, refresh) and a search icon.

Step 5: Go to Models folder and add new file StockQuote.cs to it with following content.

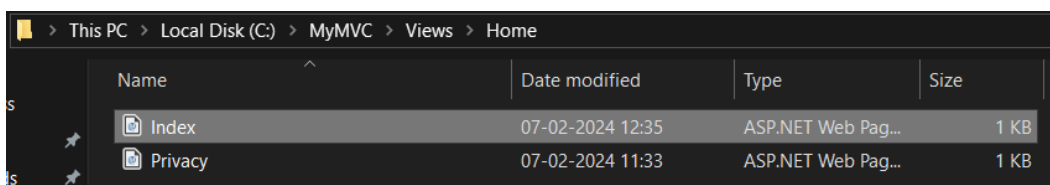


The screenshot shows a File Explorer window with the path 'This PC > Local Disk (C:) > MyMVC > Models'. It lists two files: 'ErrorViewModel.cs' and 'StockQuote.cs', both of which are C# Source Files and 1 KB in size.

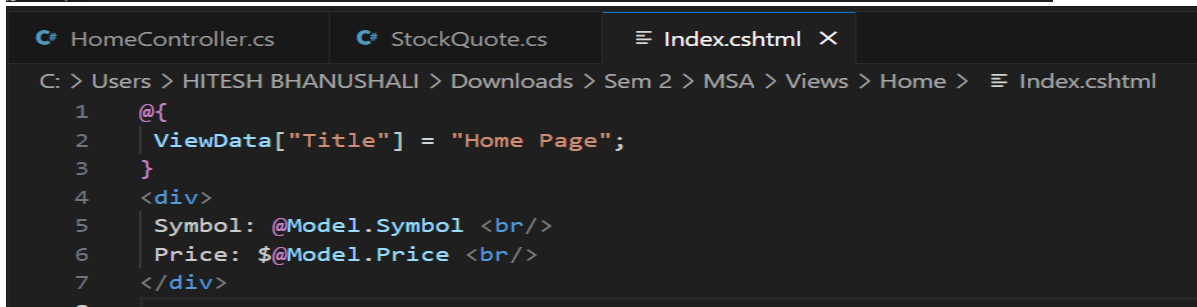


```
1 using System;
2
3 namespace MyMVC.Models
4 {
5     public class StockQuote
6     {
7         public int Price { get; set; }
8         public string stockName { get; set; }
9     }
10 }
```

Step 6: Inside the Views folder edit Index.cshtml file.



The screenshot shows a File Explorer window with the path 'This PC > Local Disk (C:) > MyMVC > Views > Home'. It lists two files: 'Index' and 'Privacy', both of which are ASP.NET Web Pages and 1 KB in size.



```
C:\> Users > HITESH BHANUSHALI > Downloads > Sem 2 > MSA > Views > Home > Index.cshtml
1 @{
2     ViewData["Title"] = "Home Page";
3 }
4 <div>
5     Symbol: @Model.Symbol <br/>
6     Price: $@Model.Price <br/>
7 </div>
8
```

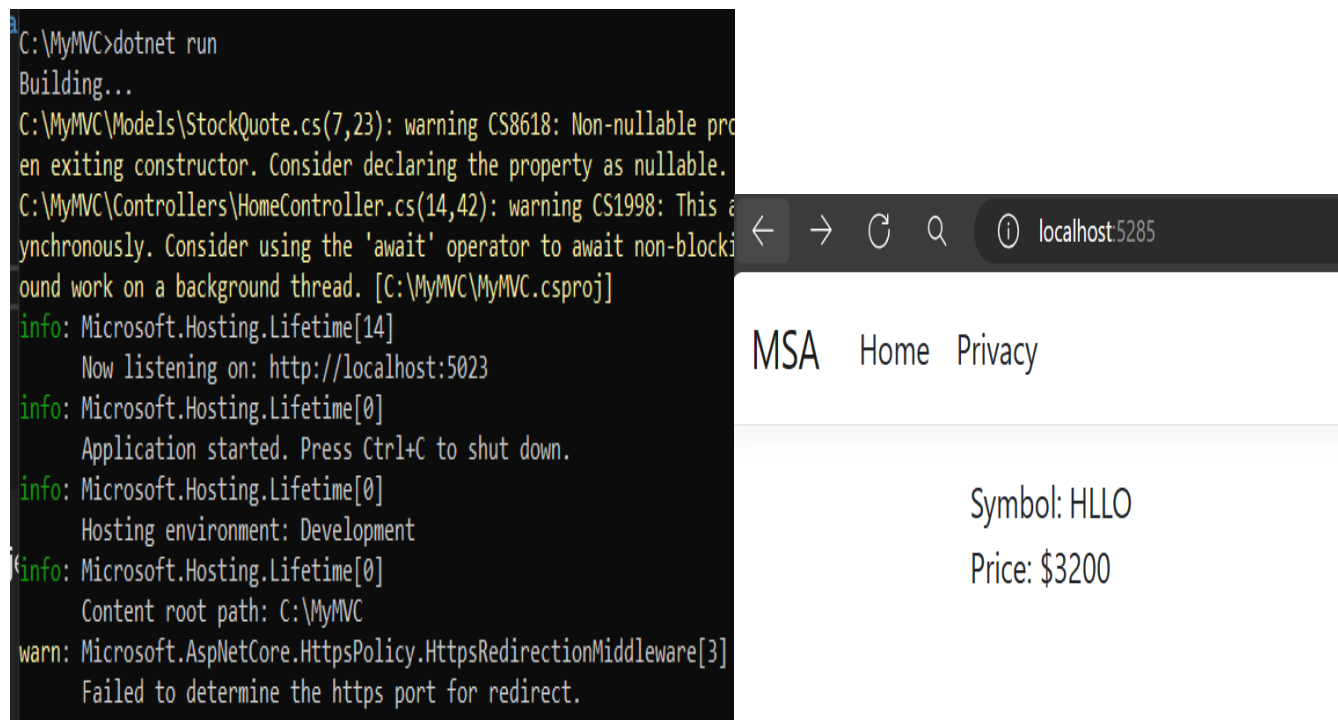
KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

Step 7: Now modify HomeController.cs file to match following:

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

Step 8: Run the project.



The screenshot shows a terminal window on the left and a web browser on the right. The terminal displays the command 'dotnet run' and the output, including warnings and information about the application starting on http://localhost:5023. The web browser shows the application running at localhost:5285, displaying 'MSA Home Privacy' and 'Symbol: HLLO Price: \$3200'.

```
C:\MyMVC>dotnet run
Building...
C:\MyMVC\Models\StockQuote.cs(7,23): warning CS8618: Non-nullable property 'Symbol' is uninitialized when exiting constructor. Consider declaring the property as nullable.
C:\MyMVC\Controllers\HomeController.cs(14,42): warning CS1998: This async method lacks 'await' operators and will run synchronously. Consider using the 'await' operator to await non-blocking asynchronous work on a background thread. [C:\MyMVC\MyMVC.csproj]
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5023
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
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
```

localhost:5285

MSA Home Privacy

Symbol: HLLO
Price: \$3200

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

PRACTICAL - 2: BUILDING ASP.NET CORE REST API

Aim: Building Web API.

Step1: create a Web API

Open two command prompts

Command prompt 1:

Command: dotnet new webapi -o Glossary

Output:

```
C:\Users\HITESH BHANUSHALI\Downloads\Sem 2\MSA>dotnet new webapi -o Glossary
The template "ASP.NET Core Web API" was created successfully.

Processing post-creation actions...
Restoring C:\Users\HITESH BHANUSHALI\Downloads\Sem 2\MSA\Glossary\Glossary.csproj:
  Determining projects to restore...
  Restored C:\Users\HITESH BHANUSHALI\Downloads\Sem 2\MSA\Glossary\Glossary.csproj (in 8.31 sec).
Restore succeeded.
```

Command: cd Glossary dotnet run

Output:

```
C:\Users\HITESH BHANUSHALI\Downloads\Sem 2\MSA\Glossary>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5024
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\HITESH BHANUSHALI\Downloads\Sem 2\MSA\Glossary
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
```

Command prompt 2: (try running ready made weatherforecast class for testing)

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

Output:

```
C:\Users\HITESH BHANUSHALI\Downloads\Sem 2\MSA>curl --insecure http://localhost:5024/weatherforecast
[{"date": "2024-03-16", "temperatureC": -1, "summary": "Scorching", "temperatureF": 31}, {"date": "2024-03-17", "temperatureC": 13, "summary": "Freezing", "temperatureF": 55}, {"date": "2024-03-18", "temperatureC": 47, "summary": "Sweltering", "temperatureF": 116}, {"date": "2024-03-19", "temperatureC": 28, "summary": "Hot", "temperatureF": 82}, {"date": "2024-03-20", "temperatureC": -5, "summary": "Balmy", "temperatureF": 24}]
C:\Users\HITESH BHANUSHALI\Downloads\Sem 2\MSA>
```

Step 2: 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.

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

Add Following two files :

1) [individual system Path]\Glossary\GlossaryItem.cs (type it in notepad and save as all files)

```
//GlossaryItem.cs

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

2) [Individual System Path]\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"
            },
        }
```

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

```
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]
```

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

```
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);
    }
}

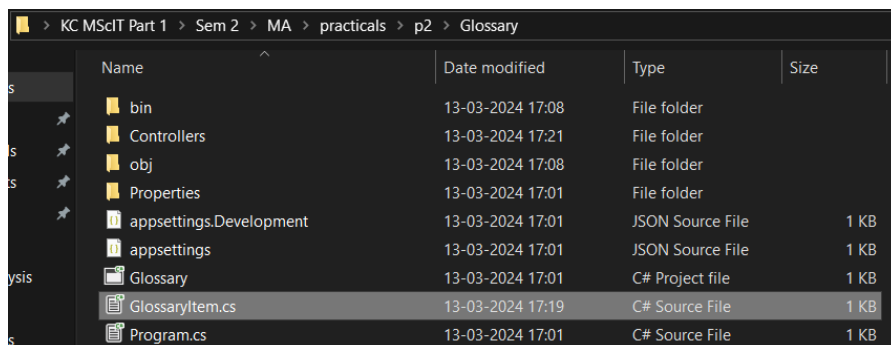
[HttpPut]
public ActionResult Put(GlossaryItem glossaryItem)
{
    var existingGlossaryItem = Glossary.Find(item =>
        item.Term.Equals(glossaryItem.Term, StringComparison.InvariantCultureIgnoreCase));
    if (existingGlossaryItem == null)
    {
        return BadRequest("Cannot update a nont existing term.");
    } else
    {
        existingGlossaryItem.Definition = glossaryItem.Definition;
        return Ok();
    }
}
```

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

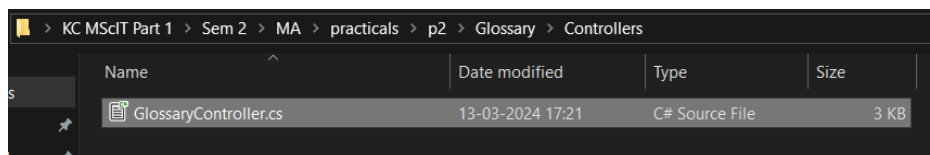
M.Sc (I.T.) Part-1 Semester II

```
[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
bin	13-03-2024 17:08	File folder	
Controllers	13-03-2024 17:21	File folder	
obj	13-03-2024 17:08	File folder	
Properties	13-03-2024 17:01	File folder	
appsettings.Development	13-03-2024 17:01	JSON Source File	1 KB
appsettings	13-03-2024 17:01	JSON Source File	1 KB
Glossary	13-03-2024 17:01	C# Project file	1 KB
GlossaryItem.cs	13-03-2024 17:19	C# Source File	1 KB
Program.cs	13-03-2024 17:01	C# Source File	1 KB



Name	Date modified	Type	Size
GlossaryController.cs	13-03-2024 17:21	C# Source File	3 KB

Step 3: Now stop running previous dotnet run on command prompt 1 using Ctrl+C. and Run it again for new code. On

Command prompt 1:

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

Command: dotnet run

Output:

```
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary>dotnet run
Building...
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary\GlossaryItem.cs(5,16): warn
property 'Term' must contain a non-null value when exiting constructor. Consider declaring the
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary\Glossary.csproj]
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary\GlossaryItem.cs(6,16): warn
property 'Definition' must contain a non-null value when exiting constructor. Consider declarin
ble. [C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary\Glossary.csproj]
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary\Controllers\GlossaryControl
YSLIB0013: 'Uri.EscapeUriString(string)' is obsolete: 'Uri.EscapeUriString can corrupt the Uri
nsider using Uri.EscapeDataString for query string components instead.' (https://aka.ms/dotnet-
\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary\Glossary.csproj]
info: Microsoft.Hosting.Lifetime[14]
    Now listening on: http://localhost:5227
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\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary
```

On Command prompt 2:

1. Getting a list of items:

Command: curl --insecure <https://localhost:5227/api/glossary>

Output:

```
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary>
curl --insecure http://localhost:5227/api/glossary
[{"term":"HTML","definition":"Hypertext Markup Language"}, {"term":"M
VC","definition":"Model View Controller"}, {"term":"OpenID","definiti
on":"An open standard for authentication"}]
```

2) Getting a single item:

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

Output:

```
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary>
curl --insecure http://localhost:5227/api/glossary/MVC
{"term":"MVC","definition":"Model View Controller"}
```

3) Creating an item:

Command: curl --insecure -X POST -d '{"term": "MFA", "definition": "An authentication process."}' -H
"Content-Type: application/json" <http://localhost:5227/api/glossary>

Output:

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

```
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary>
curl --insecure -X POST -d '{"term": "MFA", "definition": "An authentication process."}' -H "Content-Type:application/json" http://localhost:5227/api/glossary
```

4) Update Item:

Command: curl --insecure -X PUT -d '{"term": "MVC", "definition": "Modified record of Model View Controller."}' -H "Content-Type:application/json" http://localhost:5227/api/glossary

Output:

```
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary>
curl --insecure -X PUT -d '{"term": "MVC", "definition": "Modified record of Model View Controller."}' -H "Content-Type:application/json" http://localhost:5227/api/glossary
```

```
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary>
curl --insecure http://localhost:5227/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MVC", "definition": "Modified record of Model View Controller."}, {"term": "OpenID", "definition": "An open standard for authentication"}, {"term": "MFA", "definition": "An authentication process."}]
```

5) Delete Item:

Command: curl --insecure --request DELETE --url <http://localhost:5227/api/glossary/openid>

Output:

```
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary>
curl --insecure --request DELETE --url http://localhost:5227/api/glossary/openid
```

```
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary>
curl --insecure http://localhost:5227/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MVC", "definition": "Modified record of Model View Controller."}, {"term": "MFA", "definition": "An authentication process."}]
C:\Users\hp\Desktop\KC MScIT Part 1\Sem 2\MA\practicals\p2\Glossary>
```

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

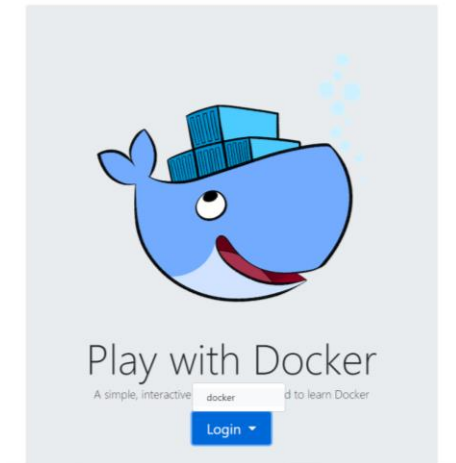
M.Sc (I.T.) Part-1 Semester II

PRACTICAL – 3: Working with Docker

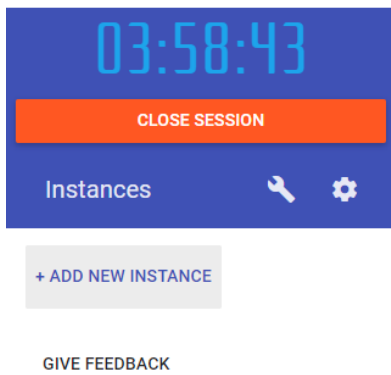
Aim: Working with Docker.

Step 1: Create a Docker account and login into docker using the below mentioned link.

Link: <https://labs.play-with-docker.com/>



Step 2: Add new instance.



Step 3: perform the following:

Method1: To pull and push images using docker.

Command To check docker version: `docker --version`

Output:

```
[node1] (local) root@192.168.0.13 ~
$ docker --version
Docker version 24.0.7, build afdd53b
[node1] (local) root@192.168.0.13 ~
```

Command To pull readymade image: `docker pull rocker/verse`

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

Output:

```
$ docker pull rocker/verse
Using default tag: latest
latest: Pulling from rocker/verse
bccd10f490ab: Pull complete
8e07a7884b74: Pull complete
630cfc57d20b: Extracting [=====>] 88.01MB/270.7MB
96f06a0292f9: Download complete
22ed396a08ac: Download complete
250ad923785a: Download complete
5468ac2d5b43: Download complete
ec72ce13388f: Download complete
7079f735b1b7: Download complete
d90bff875213: Download complete
```

Command To check images in docker: docker images

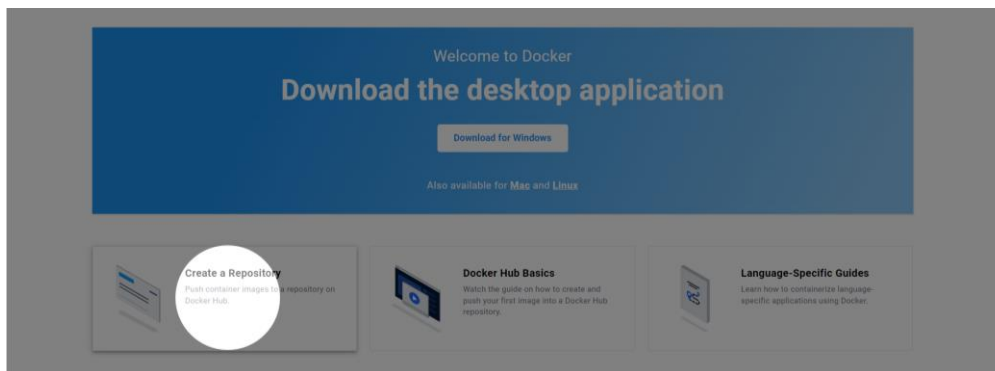
Output:

```
$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
rocker/verse        latest       fd553ed661ad     11 hours ago    3.57GB
[nod1] (local) root@192.168.0.13 ~
$
```

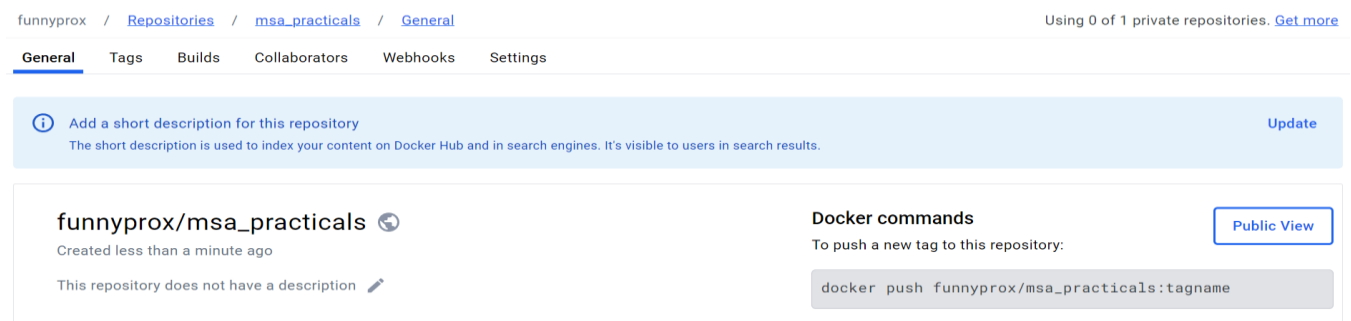
Now Login to docker hub and create repository using below mentioned link.

Link: <https://hub.docker.com/>

Output:



Now Check the created Repository.



KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

Command To login to your docker account: `docker login -u <username>`

password: (docker hub > ID > My Account > Security > Create an Access Token)

Output:

Copy Access Token

When logging in from your Docker CLI client, use this token as a password. [Learn more](#)

ACCESS TOKEN DESCRIPTION

practical

ACCESS PERMISSIONS

Read, Write, Delete

To use the access token from your Docker CLI client:

1. Run `docker login -u funnyprox`
2. At the password prompt, enter the personal access token.

dckr_pat_VkHb729JDafz8uMBo7yk-uSk_04

Copy



WARNING: This access token will only be displayed once. It will not be stored and cannot be retrieved. Please be sure to save it now.

Copy and Close

```
docker:10/rocker/verse:latest
[node1] (local) root@192.168.0.28 ~
$ docker login -u funnyprox
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
```

Command to tag image: `docker tag <image name> <username>/<repository name>:<Tag name>`

Output:

```
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
rocker/verse  latest   fd553ed661ad   12 hours ago   3.57GB
[node1] (local) root@192.168.0.18 ~
```

Command to push image to docker hub account: `docker push <username>/<repository name>:<Tag name>`

Output:

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

```
$ docker push aminirfankhan/repo1:firsttry
The push refers to repository [docker.io/aminirfankhan/repo1]
1dd3605ff7d3: Mounted from rocker/verse
ba9120e4c148: Mounted from rocker/verse
16d63f9d3f63: Mounted from rocker/verse
6500fbfb7b1c: Mounted from rocker/verse
c86449ee7105: Mounted from rocker/verse
07117bdb5a5c: Mounted from rocker/verse
321837ce5aec: Mounted from rocker/verse
8f5a87f20131: Mounted from rocker/verse
a7925f4ea9e1: Mounted from rocker/verse
5498e8c22f69: Mounted from rocker/verse
firsttry: digest: sha256:6fd2bb7544bcbe006372d3eee2e82d4d70562
```

Check it in docker hub now.

aminirfankhan/repo1

Updated 2 minutes ago

MA practical 3 (Working with Docker).

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
firsttry		Image	---	2 minutes ago

[See all](#)

Explore Repositories Organizations

Explore / aminirfankhan/repo1 / firsttry

aminirfankhan/repo1:firsttry Delete Tag

DIGEST: sha256:6fd2bb7544bcbe006372d3eee2e82d4d70562c6c38a43e66d1dee8265776d778

OS/ARCH	COMPRESSED SIZE	LAST PUSHED	TYPE
linux/amd64	1.2 GB	3 minutes ago by aminirfankhan	Image

Image Layers Vulnerabilities

IMAGE LAYERS

1	ARG RELEASE	0 B	Command ARG RELEASE
2	ARG LAUNCHPAD_BUILD_ARCH	0 B	
3	LABEL org.opencontainers.image.ref.name=ubuntu	0 B	
4	LABEL org.opencontainers.image.version=22.04	0 B	
5	ADD file ... in /	28.17 MB	

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

Command to create docker file:

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

1. cat > Dockerfile <<EOF 2. FROM busybox 3. CMD echo "Hello world! This is my first Docker image." 4. EOF

Command to build image from docker file: docker build -t <username>/<repository name> .

Output:

```
[node1] (local) root@192.168.0.28 ~
$ cat > Dockerfile <<EOF
> FROM busybox
> CMD echo "Hello world! This is my first Docker image."
> EOF
```

```
[node1] (local) root@192.168.0.28 ~
$ docker build -t aminirfankhan/repo2 .
[+] Building 1.0s (6/6) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 106B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/busybox:latest
=> [auth] library/busybox:pull token for registry-1.docker.io
=> [1/1] FROM docker.io/library/busybox@sha256:650fd573e056b679a5110a70aabeeb01e26b76e545ec4b9c70a9523f2dfaf18c6
=> => resolve docker.io/library/busybox@sha256:650fd573e056b679a5110a70aabeeb01e26b76e545ec4b9c70a9523f2dfaf18c6
=> => extracting sha256:7b2699543f22d5b8dc8d66a5873eb246767bca37232dee1e7a3b8c9956bceb0c
=> => sha256:ba5dc23f65d4cc4a4535bce55cf9e63b068eb02946e3422d3587e8ce803b6aab 372B / 372B
=> => sha256:7b2699543f22d5b8dc8d66a5873eb246767bca37232dee1e7a3b8c9956bceb0c 2.15MB / 2.15MB
=> => sha256:650fd573e056b679a5110a70aabeeb01e26b76e545ec4b9c70a9523f2dfaf18c6 6.76kB / 6.76kB
=> => sha256:4be429a5fbb2e71ae7958bfa558bc637cf3a61baf40a708cb8fff532b39e52d0 610B / 610B
=> exporting to image
=> => exporting layers
=> => writing image sha256:534572efe9fa5fa06ada6bed3ff3746c43c589d8f8fec2a87863f0e392190094
=> => naming to docker.io/aminirfankhan/repo2
[node1] (local) root@192.168.0.28 ~
```

```
[node1] (local) root@192.168.0.28 ~
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
aminirfankhan/repo2	latest	534572efe9fa	10 months ago	4.26MB

Command to push image to docker hub: docker push <username>/<repository name>

Output:

```
[node1] (local) root@192.168.0.28 ~
$ docker push aminirfankhan/repo2
Using default tag: latest
The push refers to repository [docker.io/aminirfankhan/repo2]
95c4a60383f7: Mounted from library/busybox
latest: digest: sha256:547efefc9ee5ef34c77c22c9cee81a3901987e0268a5531cc175d60ba80db374 size: 527
```

Command to run docker image: docker run <username>/<repository name>

```
[node1] (local) root@192.168.0.28 ~
$ docker run aminirfankhan/repo2
Hello world! This is my first Docker image.
```

Now Close the session.

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

PRACTICAL-4: INSTALLING SOFTWARE PACKAGES ON DOCKER

Step-1: Creating Container

Command: docker pull ubuntu

```
C:\Users\Amink>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
bccd10f490ab: Pull complete
Digest: sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

Step-2: Let's initiate an Ubuntu session by accessing the terminal from the command prompt using the following command: "docker run -it ubuntu"

```
C:\Users\Amink>docker run -it ubuntu
root@afb32e1aec5:/# apt-get -y update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1641 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [44.6 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1081 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
```

```
root@afb32e1aec5:/# apt-get -y install firefox
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

- For vim

Command: apt-get install vim

```
root@afb32e1aec5:/# apt-get install vim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

Step-4: Run the installed software.

Command: vim

```
VIM - Vi IMproved

        version 8.2.2121
        by Bram Moolenaar et al.
    Modified by team+vim@tracker.debian.org
    Vim is open source and freely distributable

    Help poor children in Uganda!
type  :help iccf<Enter>      for information

type  :q<Enter>              to exit
type  :help<Enter> or <F1>   for on-line help
type  :help version8<Enter> for version info
```

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

PRACTICAL - 5: WORKING WITH DOCKER VOLUMES & NETWORKS

Step 1: Enter the following Commands.

a) docker pull nginx nginx: nginx is the image which is already available in docker

b) docker run -it-name-webapp -d -p 80:80 nginx: Create a webapp and run it with nginx image on port 80

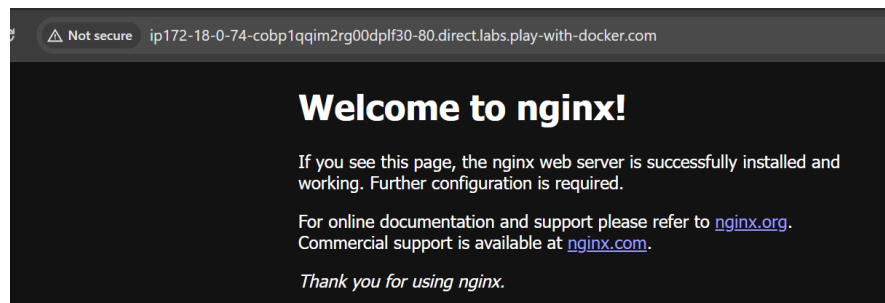
```
$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
e1caac4eb9d2: Pull complete
88f6f236f401: Pull complete
c3ea3344e711: Pull complete
cc1bb4345a3a: Pull complete
da8fa4352481: Pull complete
c7f80e9cdab2: Pull complete
18a869624cb6: Pull complete
Digest: sha256:c26ae7472d624ba1fafd296e73cecc4f93f853088e6a9c13c0d52f6ca5865107
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

```
$ docker run -it --name=webapp -d -p 80:80 nginx
c8d1c8108846f48a263ccb141ab2c7163f3aa017dcdde86d21c8ea019e51d3bb
```

Step 2: Click on Port and enter 80 in the dropdown window and click OK.



Output: The below webpage will be visible



Step 3: Enter the below command to enter bash shell and then open port 80.

```
docker exec -it WebApp bash
```

```
Cd/usr/share/nginx/html
```

```
Echo "Hello Welcome to updated nginx Page."> index.html
```

```
exit
```

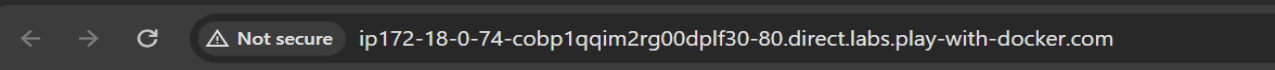
KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

```
$ docker exec -it WebApp bash
Error response from daemon: No such container: WebApp
[node1] (local) root@192.168.0.28 ~
$ Cd/usr/share/nginx/html
bash: Cd/usr/share/nginx/html: No such file or directory
[node1] (local) root@192.168.0.28 ~
$ Cd /usr/share/nginx/html
bash: Cd: command not found
[node1] (local) root@192.168.0.28 ~
$ docker exec -it webapp bash
root@5b96ae718f68:/# cd /usr/share/nginx/html
root@5b96ae718f68:/usr/share/nginx/html# echo "Hey, Hitesh Bhanushali KFMSCIT005, welcome to Nagnix!"
Hey, Hitesh Bhanushali KFMSCIT005, welcome to Nagnix!
root@5b96ae718f68:/usr/share/nginx/html# exit
```

Output: The below webpage will be visible

NOTE: if above step doesn't work, use `cat > index.html` and type the statement, then do `ctrl+c` to close



`echo "HITESH BHANUSHALI trying to update the file" exit`

Step 4: List all the running containers: `docker ps`

```
[node1] (local) root@192.168.0.28 ~
$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
5b96ae718f68   nginx    "/docker-entrypoint...." 6 minutes ago  Up 6 minutes  0.0.0.0:80->80/tcp       webap
p
```

Step 5: Create another container in Docker: `webappl`

Docker run `-it --name=webappl -d -p 80:80 nginx`:

```
[node1] (local) root@192.168.0.28 ~
$ docker run -it --name=webapp2 -d -p 90:90 nginx
e05cf3338a0ef36e07b4d0a0941192ab1c7818fd8dcac71d12004a6ad023782
[node1] (local) root@192.168.0.28 ~
$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
e05cf3338a0e   nginx    "/docker-entrypoint...." 7 seconds ago  Up 5 seconds  80/tcp, 0.0.0.0:90->90/t
cp   webapp2
5b96ae718f68   nginx    "/docker-entrypoint...." 13 minutes ago  Up 13 minutes  0.0.0.0:80->80/tcp       webapp
webapp
[node1] (local) root@192.168.0.28 ~
$
```

Step 6: Click on port and enter 90 in the dropdown and click ok.

Output: the welcome page of nginx should be visible

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

ip172-18-0-43-cnms33012o9000ekkuhg-80.directlabs.play-with-docker.com

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Problem: Updates made in one container are not reflected into another container.

Solution: Volume

Updates made in one container within the volume will be reflected in all the containers of that volume.

Step 7: Creation of Volume(MyVolume)

Command:

a) docker volume create MyVolume

```
$ docker volume create MyVolume
MyVolume
```

b) docker volume inspect MyVolume

```
$ docker volume inspect MyVolume
[
  {
    "CreatedAt": "2024-03-10T14:45:50Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/MyVolume/_data",
    "Name": "MyVolume",
    "Options": null,
    "Scope": "local"
  }
]
```

d) docker stop WebApp2

```
[node1] (local) root@192.168.0.28 ~
$ docker stop webapp2
webapp2
```

Step 8: Create a container (WebApp 21) inside the container MyVolume

docker run -d --name=webapp21 --mount source=MyVolume,destination=/usr/share/nginx/html -p 90:90 nginx

```
[node1] (local) root@192.168.0.28 ~
$ docker run -d --name=webapp21 --mount source=MyVolume,destination=/usr/share/nginx/html -p 90:90 nginx
cde45b96d75edd06bebb97ab2b3899a5d13cd6b69485175a994452adce14cd0c
```

Step 9: Enter the below commands:

a) cd /var/lib/docker

b) ls

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

```
$ cd /var/lib/docker
[node1] (local) root@192.168.0.28 /var/lib/docker
$ ls
buildkit      containers  image      overlay2   runtimes   tmp
containerd    engine-id  network    plugins    swarm      volumes
```

c) cd volumes

```
$ cd volumes
```

d) ls

```
$ ls
MyVolume      backingFsBlockDev  metadata.db
```

Step 10: Edit the index file with the below content to "Display the content on the Webpage."

```
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes
$ echo "Inside MYVolume, Hitesh Bhanushali KFCMSIT005" > index.html
```

Open Port 90

Step 11: Stop the above container (WebApp2) and Create another container within the volume (MyVolume)

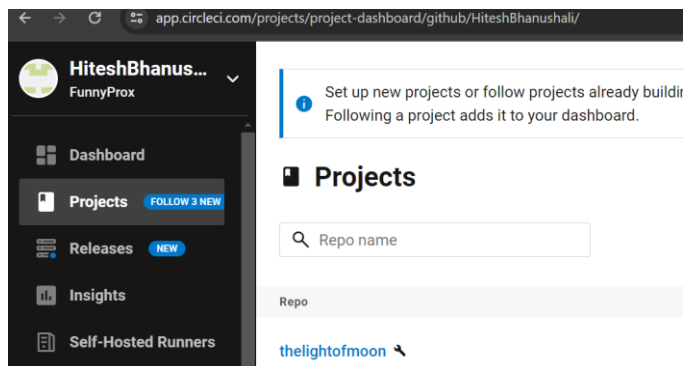
Open port 90

Output: The edits made in one container of the volume will be reflected in all the containers of that volume

PRACTICAL 6:WORKING WITH CIRCLECI FOR CONTINOUS INTEGRATION

Step-1: - Let's initiate the process with these fundamental steps: Log in to GitHub. & Create a public repository named Sabar, ensuring to include a README file.

Step-2: Login to Circle ci with GitHub login & Navigate to Projects tab where you will find the repositories you have created. Once you find the hello-world repository click on setup project.

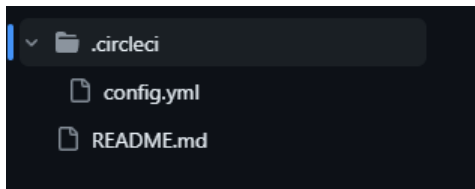


Step-3: Select Faster and click on setup project.

Step-4:"In GitHub, navigate to the 'hello-world' repository, then proceed to the 'circleci-project' branch and open the 'config.yml' file."

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II



```
thelighttofmoo / .circleci / config.yml
HiteshBhanushali Add .circleci/config.yml ✓

Code Blame 31 lines (28 loc) · 1.38 KB Code 55% faster with GitHub Copilot

1 # Use the latest 2.1 version of CircleCI pipeline process engine.
2 # See: https://circleci.com/docs/configuration-reference
3 version: 2.1
4
5 # Define a job to be invoked later in a workflow.
6 # See: https://circleci.com/docs/jobs-steps/#jobs-overview & https://circleci.com/docs/configuration-reference/#jobs
7 jobs:
8   say-hello:
9     # Specify the execution environment. You can specify an image from Docker Hub or use one of our convenience images from CircleCI.
10    # See: https://circleci.com/docs/executor-intro/ & https://circleci.com/docs/configuration-reference/#executor-job
11    docker:
12      # Specify the version you desire here
13      # See: https://circleci.com/developer/images/image/cimg/base
14      - image: cimg/base:current
15
16    # Add steps to the job
17    # See: https://circleci.com/docs/jobs-steps/#steps-overview & https://circleci.com/docs/configuration-reference/#steps
18    steps:
19      # Checkout the code as the first step.
20      - checkout
21      - run:
22        name: "Say hello"
23        command: "echo Hello, World!"
```

Step-5: Edit the config file and commit changes.

```
8 steps:
9   # Checkout the code as the first step.
10  - checkout
11  - run:
12    name: "Hitesh Bhanushali KFMSCT005"
13    command: "echo Hello, MSCIT SEM 2!"
14
15  # Orchestrate jobs using workflows
16  # See: https://circleci.com/docs/workflows/ & https://circleci.com/docs/configuration-reference/#workflows
17  workflow:
18    use Control + Shift + m to toggle the tab key moving focus. Alternatively, use esc then tab to move to the next interactive el
```

Step-6: We can now see that the updates from GitHub are being logged, and the message is being recorded for smooth integration.

Pipeline	Status	Workflow	Trigger Event	Start
thelighttofmoo 5	✓ Success	say-hello-workflow	main 1a501aa Update config.yml	36s ago
Jobs				
✓ say-hello 7				

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

PRACTICAL - 7: CREATING BACKING SERVICE WITH ASP.NET 2.0 CORE

Start typing following commands

Command :

```
docker run -d -p 5000:5000 -e PORT=5000 \
-e LOCATION_URL=http://localhost:5001 \ dotnetcoreservices/teamservice:location
```

Output: (you can observe that it has started port 5000 on top)

co8fhjq_i_co8fm98l2o9000ce4q2g

IP	192.168.0.8	OPEN PORT	5000
Memory	26.00% (1.016GiB / 3.906GiB)		CPU
			1.00%
SSH	ssh ip172-18-0-25-co8fhjqim2rg00dqd31g@direct.labs.play		

DELETE EDITOR

```
[node1] (local) root@192.168.0.8 ~
$ docker run -d -p 5000:5000 -e PORT=5000 \
> -e LOCATION_URL=http://localhost:5001 \
> dotnetcoreservices/teamservice:location
Unable to find image 'dotnetcoreservices/teamservice:location' locally
location: Pulling from dotnetcoreservices/teamservice
693502eb7dfb: Pull complete
081cd4bfd521: Pull complete
5d2dc01312f3: Pull complete
585880aea240: Pull complete
3905d0c644ea: Pull complete
c59037c90022: Pull complete
5a7d450223d5: Pull complete
Digest: sha256:3e9355b72f0ba151d17a2dc9844331a5f590e3afa685d66789458525210346e1
Status: Downloaded newer image for dotnetcoreservices/teamservice:location
43e427a2fbfb9fc877902a5d80eefe16cd46e9c50bf3209281fdb67291ea5e0
[node1] (local) root@192.168.0.8 ~
```

Command: to run location service

```
docker run -d -p 5001:5001 -e PORT=5001 dotnetcoreservices/location-service:nodb
```

Output: (now it has started one more port that is 5001 for location service)

co8fhjq_i_co8fm98l2o9000ce4q2g

IP	192.168.0.8	OPEN PORT	5001 5000
Memory	26.86% (1.049GiB / 3.906GiB)		CPU
			0.97%
SSH	ssh ip172-18-0-25-co8fhjqim2rg00dqd31g@direct.labs.play		

DELETE EDITOR

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

```
[node1] (local) root@192.168.0.8 ~
$ docker run -d -p 5001:5001 -e PORT=5001 \
> dotnetcoreservices/location-service:nodb
Unable to find image 'dotnetcoreservices/location-service:nodb' locally
nodb: Pulling from dotnetcoreservices/location-service
693502eb7dfb: Already exists
081cd4bfd521: Already exists
5d2dc01312f3: Already exists
585880aea240: Already exists
3905d0c644ea: Already exists
c59037c90022: Already exists
dbc03883a4ca: Pull complete
Digest: sha256:5f7aca33c5e2117e04f58a59e0cf96fd20d5cbf2cf66c3cd708118d573255168
Status: Downloaded newer image for dotnetcoreservices/location-service:nodb
7dedc8af691b09efb946687c0ee5b3e12d5fbc12e63a64af1321c960aeddff28b
[node1] (local) root@192.168.0.8 ~
$
```

Command: to check running images in docker \$docker images

Output:

```
[node1] (local) root@192.168.0.8 ~
$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
dotnetcoreservices/teamservice   location     b27d0de8f2de     7 years ago     886MB
dotnetcoreservices/location-service  nodb        03339f0ea9dd     7 years ago     883MB
[node1] (local) root@192.168.0.8 ~
$
```

Command: to create new team

```
curl -H "Content-Type:application/json" -X POST -d \
'{ "id": "e52baa63-d511-417e-9e54-7aab04286281", "name": "KC" }' http://localhost:5000/teams
```

Command: curl <http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281>

Output:

```
[node1] (local) root@192.168.0.8 ~
> '{ "id": "e52baa63-d511-417e-9e54-7aab04286281", "name": "KC" }' http://localhost:500
0/teams
{"name": "KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []} [node1] (local) root@192.168.0.8 ~
$ curl http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281
{"name": "KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []} [node1] (local) root@192.168.0.8 ~
$
```

Command: to add new member to team

```
curl -H "Content-Type:application/json" -X POST -d \
'{ "id": "63e7acf8-8fae-42ce-9349-3c8593ac8292", "firstName": "Manisha", "lastName": "Panigrahy" }'
http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281/members
```

Output:

```
[node1] (local) root@192.168.0.8 ~
> '{ "id": "e52baa63-d511-417e-9e54-7aab04286281", "name": "KC" }' http://localhost:500
0/teams
{"name": "KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []} [node1] (local) root@192.168.0.8 ~
$ curl http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281
{"name": "KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []} [node1] (local) root@192.168.0.8 ~
$ curl -H "Content-Type:application/json" -X POST -d \
> '{ "id": "63e7acf8-8fae-42ce-9349-3c8593ac8292", "firstName": "Manisha", "lastName": "Panigrahy" }' http://localhost:5000/teams/e52baa6
3-d511-417e-9e54-7aab04286281/members
{"teamID": "e52baa63-d511-417e-9e54-7aab04286281", "memberID": "63e7acf8-8fae-42ce-9349-3c8593ac8292"} [node1] (local) root@192.168.0.8 ~
$
```

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

Command: To confirm member added

curl <http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281>

Output:

```
$ curl http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281
{"name":"KC","id":"e52baa63-d511-417e-9e54-7aab04286281","members":[{"id":"63e7acf8-8fae-42ce-9349-3c8593ac8292","firstName":"Manish","lastName":"Panigrahy"}]} [node1] (local) root@192.168.0.8 ~
$
```

Command: To add location for member

curl -H "Content-Type:application/json" -X POST -d \
'{"id":"64c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f", "latitude":12.0,"longitude":12.0,"altitude":10.0,
"timestamp":0,"memberId":"63e7acf8-8fae-42ce-9349-3c8593ac8292"}'
<http://localhost:5001/locations/63e7acf8-8fae-42ce-9349-3c8593ac8292>

output:

```
$ curl -H "Content-Type:application/json" -X POST -d \  
> '{"id":"64c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f", "latitude":12.0,"longitude":12.0,"altitude":10.0, "timestamp":0,"memberId":"63e7acf8-8fae-42ce-9349-3c8593ac8292"}' http://localhost:5001/locations/63e7acf8-8fae-42ce-9349-3c8593ac8292
{"id":"64c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f","latitude":12.0,"longitude":12.0,"altitude":10.0,"timestamp":0,"memberID":"63e7acf8-8fae-42ce-9349-3c8593ac8292"}curl: (6) Could not resolve host: 8fae-42ce-9349-3c8593ac8292
[node1] (local) root@192.168.0.8 ~
$
```

Command : To confirm location is added in member

curl <http://localhost:5001/locations/63e7acf8-8fae-42ce-9349-3c8593ac8292>

Output:

```
[node1] (local) root@192.168.0.8 ~
$ curl http://localhost:5001/locations/63e7acf8-8fae-42ce-9349-3c8593ac8292
[{"id":"64c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f", "latitude":12.0,"longitude":12.0, "altitude":10.0,"timestamp":0,"memberID":"63e7acf8-8fae-42ce-9349-3c8593ac8292"}] [node1] (local)
) root@192.168.0.8 ~
$
```

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

M.Sc (I.T.) Part-1 Semester II

PRACTICAL - 8: WORKING WITH KUBERNETES

Step-1: Add a new instance to your setup, then proceed to create a master node and establish a cluster using the following commands:

Cluster: `kubectl apply -f https://raw.githubusercontent.com/cloudnativelabs/kube-router/master/daemonset/kubeadm-kuberouter.yaml`

```
2. Initialize cluster networking:

kubectl apply -f https://raw.githubusercontent.com/cloudnativelabs/kube-router/master/daemonset/kubeadm-kuberouter.yaml

[node1 ~]$ kubeadm init --apiserver-advertise-address $(hostname -i) --pod-network-cidr 10.5.0.0/16
Initializing machine ID from random generator.
W0406 07:52:38.440277 891 initconfiguration.go:120] Usage of CRI endpoints without URL scheme is deprecated and can cause kubelet errors in the future. Automatically prepending
use your configuration!
W0406 07:52:38.723863 891 version.go:256] remote version is much newer: v1.29.3; falling back to: stable-1.27
[init] Using Kubernetes version: v1.27.12
[preflight] Running pre-flight checks
[WARNING Swap]: swap is enabled; production deployments should disable swap unless testing the NodeSwap feature gate of the kubelet
[preflight] The system verification failed. Printing the output from the verification.
KUBELET_VERSION: v1.27.12-generic
OS: Linux
CGROUPS_CPU: enabled
CGROUPS_CPUQUOTA: enabled
CGROUPS_CPUSET: enabled
CGROUPS_DEVICES: enabled
CGROUPS_FREEZER: enabled
CGROUPS_MEMORY: enabled
CGROUPS_PIDS: enabled
CGROUPS_HUGETLB: enabled
CGROUPS_BLKIO: enabled
[WARNING SystemVerification]: failed to parse kernel config: unable to load kernel module: "configs", output: "", err: exit status 1
[WARNING FileContent--proc-sys-net-bridge-bridge-mf-call-iptables]: /proc/sys/net/bridge/bridge-mf-call-iptables does not exist
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0406 07:52:39.136398 891 images.go:80] could not find officially supported version of etcd for Kubernetes v1.27.12, falling back to the nearest etcd version (3.5.7-0)
```

You have the option to generate a worker node by executing the command provided by the master node on a separate instance.

```
Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 192.168.0.18:6443 --token e7ngqn.xxhotwasq7z20hut \
--discovery-token-ca-cert-hash sha256:0d197e0a42783ae822ca4642ea52f6ea2b95e4d3fe11dc3f70f575769184f4fd

Waiting for api server to startup
Warning: resource daemonsets/kube-proxy is missing the kubectl.kubernetes.io/last-applied-configuration annotation v
d only be used on resources created declaratively by either kubectl create --save-config or kubectl apply. The missi
daemonset.apps/kube-proxy configured
No resources found
[node1 ~]$ kubectl apply -f https://raw.githubusercontent.com/cloudnativelabs/kube-router/master/daemonset/kubeadm-k
configmap/kube-router-cfg created
daemonset.apps/kube-router created
serviceaccount/kube-router created
clusterrole.rbac.authorization.k8s.io/kube-router created
clusterrolebinding.rbac.authorization.k8s.io/kube-router created
```

Basic Commands:

Execute "kubectl get nodes" to inspect the available nodes in the cluster.

Run "kubectl get pods" to observe the list of running pods.

Utilize "kubectl get services" to examine the available services within the cluster.

```
[node1 ~]$ kubectl get nodes
NAME      STATUS   ROLES    AGE   VERSION
node1     Ready    control-plane 4m17s v1.27.2
node2     Ready    <none>     62s   v1.27.2

[node1 ~]$ kubectl get pods
No resources found in default namespace.

[node1 ~]$ kubectl get services
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes ClusterIP  10.96.0.1     <none>        443/TCP     4m43s

[node1 ~]$ docker -v
Docker version 24.0.2, build cb74dfc

[node1 ~]$ kubectl run nginx --image=nginx
pod/nginx created

[node1 ~]$ kubectl create deployment my-dep --image=nginx --replicas=3
deployment.apps/my-dep created

[node1 ~]$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
my-dep-5688dd958f-4lpz5  1/1     Running   0           18s
my-dep-5688dd958f-kt4jw  1/1     Running   0           18s
my-dep-5688dd958f-nr9s4  1/1     Running   0           18s
nginx     1/1     Running   0           62s

[node1 ~]$
```