

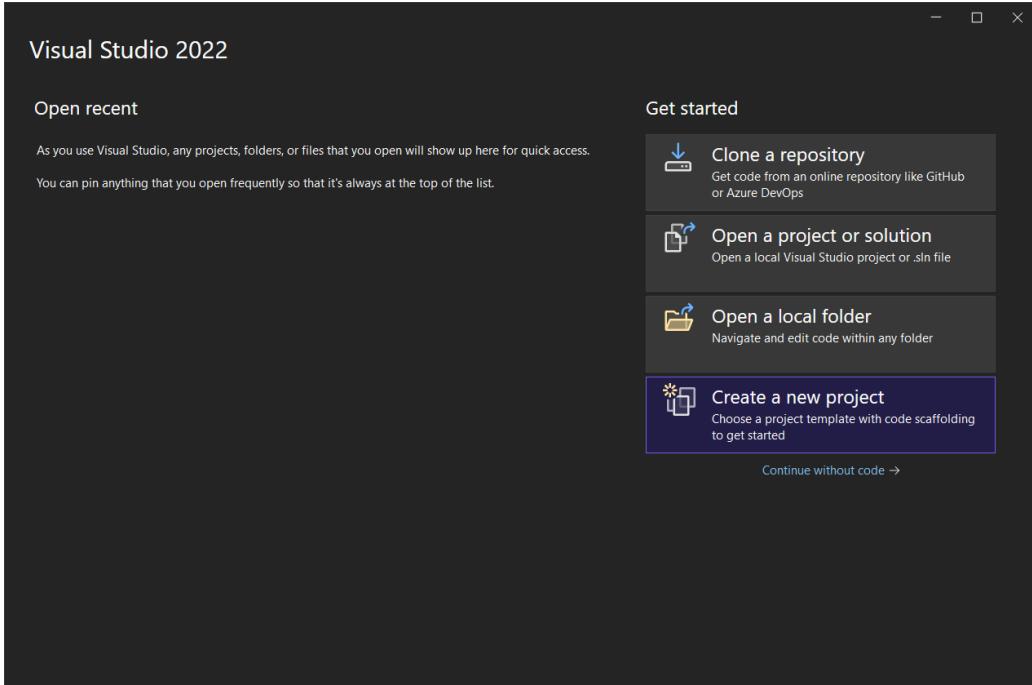
## Practical No. 01

**Aim: Build a Web Application using MVC.**

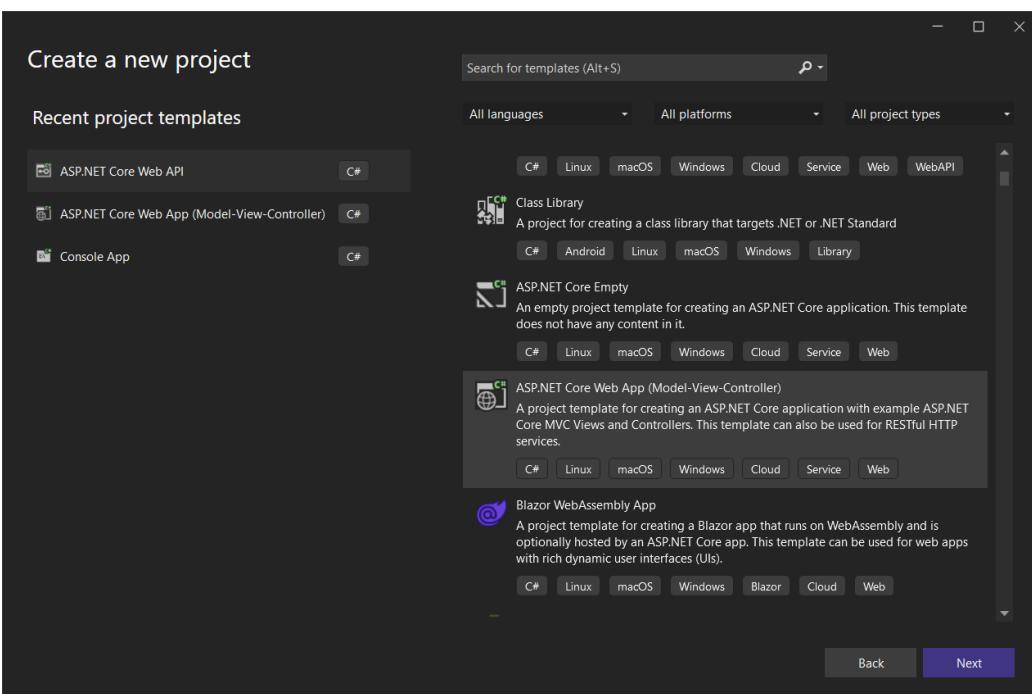
Firstly download, install & open **Microsoft Visual Studio 2022**

(<https://visualstudio.microsoft.com/vs/>)

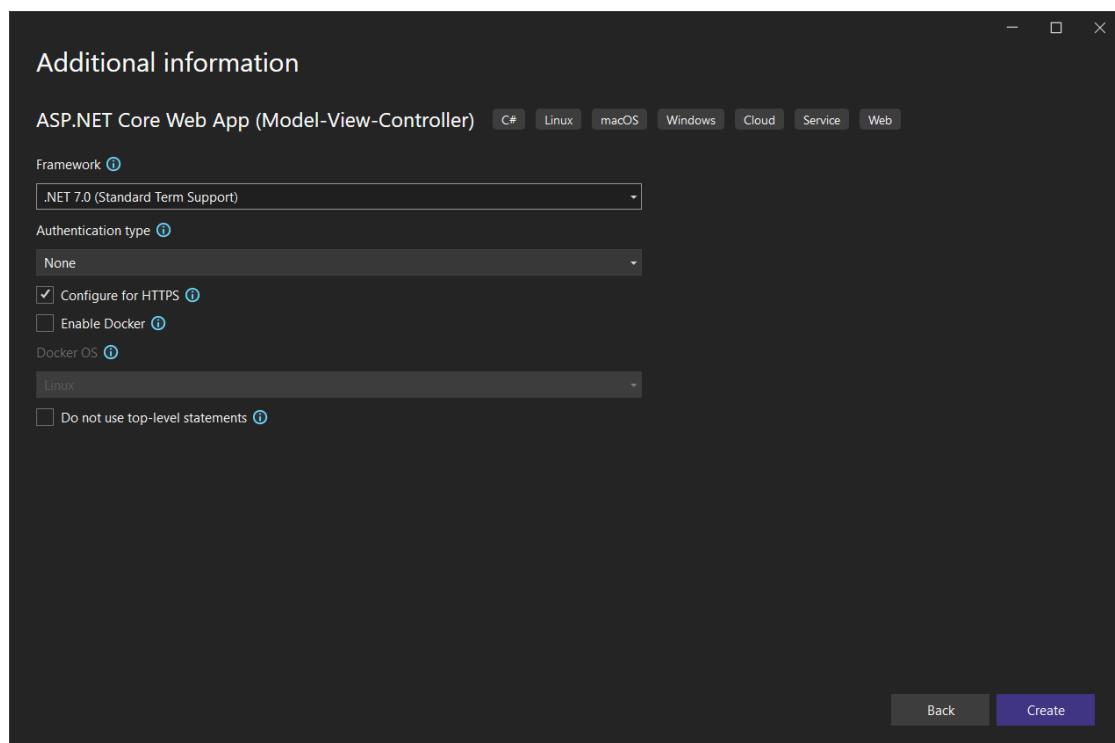
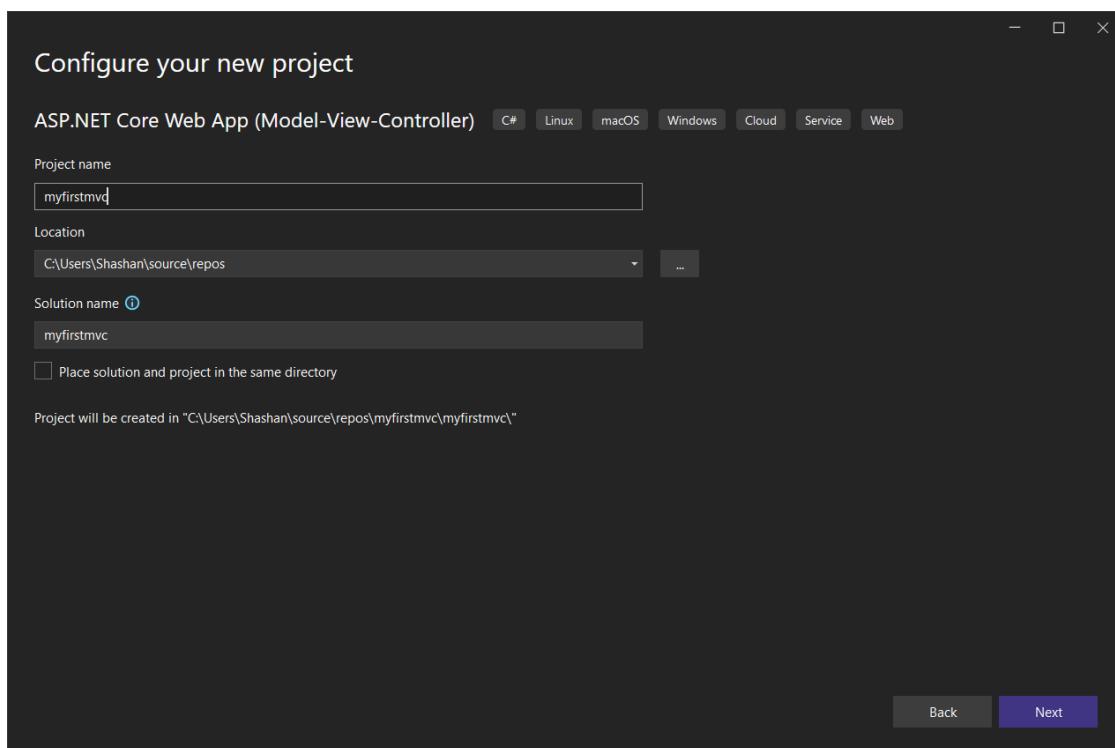
Create a new project:



Select the **ASP.NET Core Web App (Model-View\_Controller)**



Provide a name to the project:



**Code for HomeController.cs under Controller folder:**

```

using Microsoft.AspNetCore.Mvc;
using myfirstmvc.Models;
using System.Diagnostics;
namespace myfirstmvc.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;

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

        public IActionResult Index()
        {
            return View();
        }

        public IActionResult Privacy()
        {
            return View();
        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore =
true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? 
HttpContext.TraceIdentifier });
        }
    }
}

```

**Code for ErrorViewModel.cs under Model folder:**

```

namespace myfirstmvc.Models
{
    public class ErrorViewModel
    {
        public string? RequestId { get; set; }
        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
    }
}

```

**Code for Index.cshtml under View\Home folder:**

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

<div class="text-center">
    <h1 class="display-4">Welcome</h1>
    <p>Learn about <a href="https://docs.microsoft.com/aspnet/core">building Web apps with
    ASP.NET Core</a>.</p>
</div>
```

**Code for Privacy.cshtml under View\Home folder:**

```
@{
    ViewData["Title"] = "Privacy Policy";
}
<h1>@ViewData["Title"]</h1>
<p>Use this page to detail your site's privacy policy.</p>
```

**Code for Program.cs:**

```
var builder = WebApplication.CreateBuilder(args);
// Add services to the container.
builder.Services.AddControllersWithViews();
var app = builder.Build();
// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production
    scenarios, see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}
app.UseHttpsRedirection();
app.UseStaticFiles();
app.UseRouting();
app.UseAuthorization();
app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
app.Run();
```

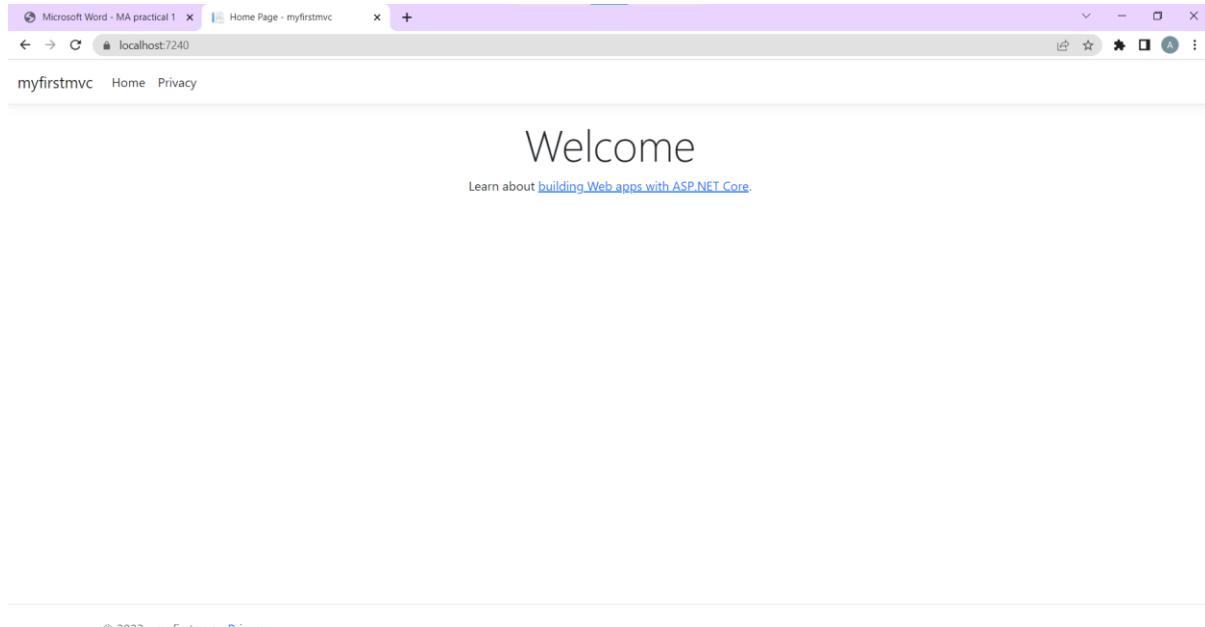
```

1 var builder = WebApplication.CreateBuilder(args);
2
3 // Add services to the container.
4 builder.Services.AddControllersWithViews();
5
6 var app = builder.Build();
7
8 // Configure the HTTP request pipeline.
9 if (app.Environment.IsDevelopment())
10 {
11     app.UseExceptionHandler("/Home/Error");
12     // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
13     app.UseHsts();
14 }
15
16 app.UseHttpsRedirection();
17 app.UseStaticFiles();
18
19 app.UseRouting();
20
21 app.UseAuthorization();
22
23 app.MapControllerRoute(
24     name: "default",
25     pattern: "{controller=Home}/{action=Index}/{id?}");
26
27 app.Run();

```

After that click the run project button under the top ribbon

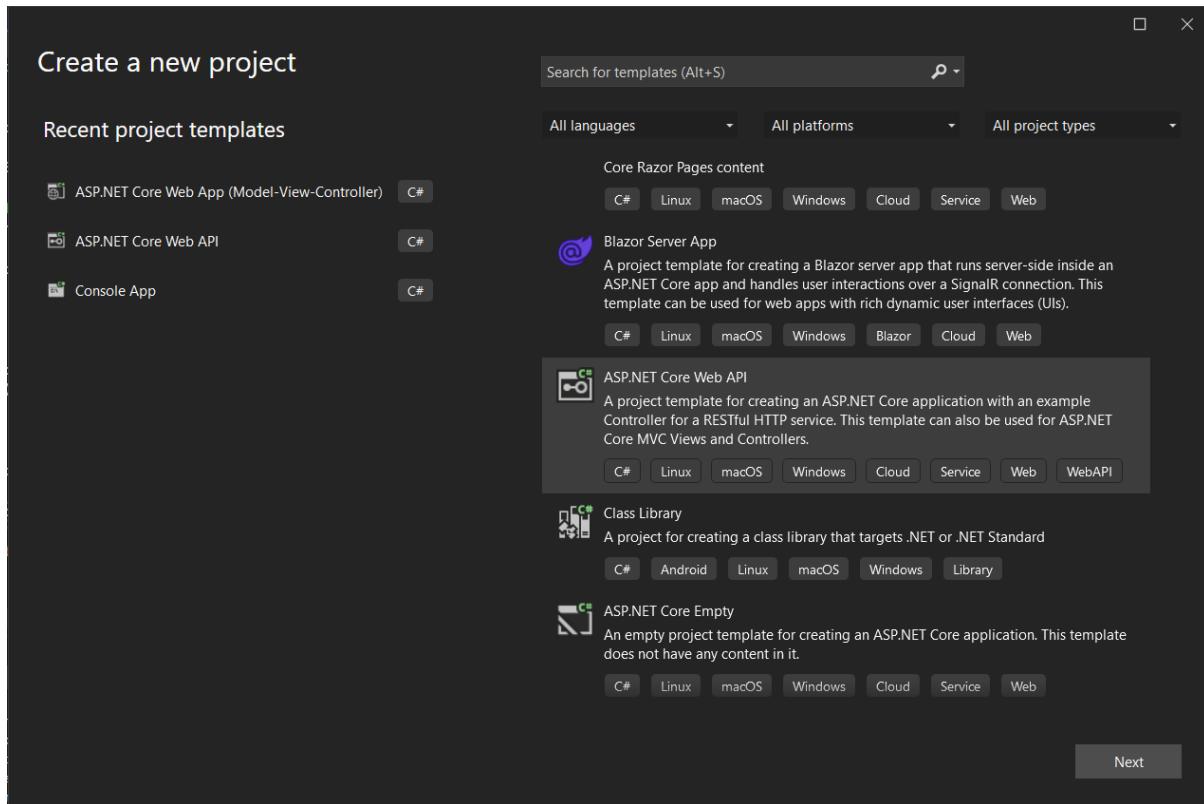
## Output:



## Practical No. 02

**Aim: Build a Web Application using API.**

Create a new project with **ASP.NET Core Web API**



**Code for *Program.cs*:**

```
var builder = WebApplication.CreateBuilder(args);
// Add services to the container.
builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
var app = builder.Build();
// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}
app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();
app.Run();
```

**Code for *WeatherForecastController.cs* under the folder *Controllers*:**

```
using Microsoft.AspNetCore.Mvc;
namespace webappwithAPI.Controllers
{
    [ApiController]
    [Route("[controller]")]
    public class WeatherForecastController : ControllerBase
    {
        private static readonly string[] Summaries = new[]
        {
            "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot",
            "Sweltering", "Scorching"
        };
        private readonly ILogger<WeatherForecastController> _logger;
        public WeatherForecastController(ILogger<WeatherForecastController> logger)
        {
            _logger = logger;
        }
        [HttpGet(Name = "GetWeatherForecast")]
        public IEnumerable<WeatherForecast> Get()
        {
            return Enumerable.Range(1, 5).Select(index => new WeatherForecast
            {
                Date = DateOnly.FromDateTime(DateTime.Now.AddDays(index)),
                TemperatureC = Random.Shared.Next(-20, 55),
                Summary = Summaries[Random.Shared.Next(Summaries.Length)]
            })
            .ToArray();
        }
    }
}
```

**Code for *WeatherForecast.cs*:**

```
namespace webappwithAPI
{
    public class WeatherForecast
    {
        public DateOnly Date { get; set; }
        public int TemperatureC { get; set; }
        public int TemperatureF => 32 + (int)(TemperatureC / 0.5556);
        public string? Summary { get; set; }
    }
}
```

**Output:**

**Replace the *WeatherForecastController.cs* under the folder *Controllers* with *GlossaryController.cs*:**

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

```

```
Term= "OpenID",
Definition = "An open standard for authentication"
}

};

[HttpGet]
public ActionResult<List<GlossaryItem>> Get()
{
    return Ok(Glossary);
}

[HttpGet]
[Route("{term}")]
public ActionResult<GlossaryItem> Get(string term)
{
    var glossaryItem = Glossary.Find(item =>
        item.Term.Equals(term, StringComparison.InvariantCultureIgnoreCase));

    if (glossaryItem == null)
    {
        return NotFound();
    }
    else
    {
        return Ok(glossaryItem);
    }
}

[HttpPost]
public ActionResult Post(GlossaryItem glossaryItem)
{
    var existingGlossaryItem = Glossary.Find(item =>
        item.Term.Equals(glossaryItem.Term,
            StringComparison.InvariantCultureIgnoreCase));

    if (existingGlossaryItem != null)
    {
        return Conflict("Cannot create the term because it already exists.");
    }
    else
    {
        Glossary.Add(glossaryItem);
        var resourceUrl = Path.Combine(Request.Path.ToString(),
            Uri.EscapeUriString(glossaryItem.Term));
        return Created(resourceUrl, glossaryItem);
    }
}

[HttpPut]
public ActionResult Put(GlossaryItem glossaryItem)
{
```

```

        var existingGlossaryItem = Glossary.Find(item =>
item.Term.Equals(glossaryItem.Term, StringComparison.InvariantCultureIgnoreCase));
        if (existingGlossaryItem == null)
        {
            return BadRequest("Cannot update a nont existing term.");
        }
        else
        {
            existingGlossaryItem.Definition = glossaryItem.Definition; return Ok();
        }
    }
    [HttpDelete]
    [Route("{term}")]
    public ActionResult Delete(string term)
    {
        var glossaryItem = Glossary.Find(item =>
item.Term.Equals(term, StringComparison.InvariantCultureIgnoreCase));
        if (glossaryItem == null)
        {
            return NotFound();
        }
        else
        {
            Glossary.Remove(glossaryItem); return NoContent();
        }
    }
}
}

```

**Replace the *WeatherForecast.cs* with *GlossaryItem.cs*:**

```

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

```

**Output:**

The screenshot shows the Swagger UI interface for a service named 'webappwithAPI'. The top navigation bar includes tabs for 'Select a definition' (set to 'webappwithAPI v1') and a dropdown menu. Below the header, the title 'webappwithAPI' is displayed with an OAS3 badge. A link to 'https://localhost:7205/swagger/v1/swagger.json' is provided.

The main content area is titled 'Glossary'. It lists several API endpoints for the '/api/Glossary' resource, each with its method, URL, and a collapse/expand arrow:

- GET /api/Glossary
- POST /api/Glossary
- PUT /api/Glossary
- GET /api/Glossary/{term}
- DELETE /api/Glossary/{term} (highlighted in red)

Below the 'Glossary' section, there is a collapsed 'Schemas' section containing a single entry: 'GlossaryItem'.

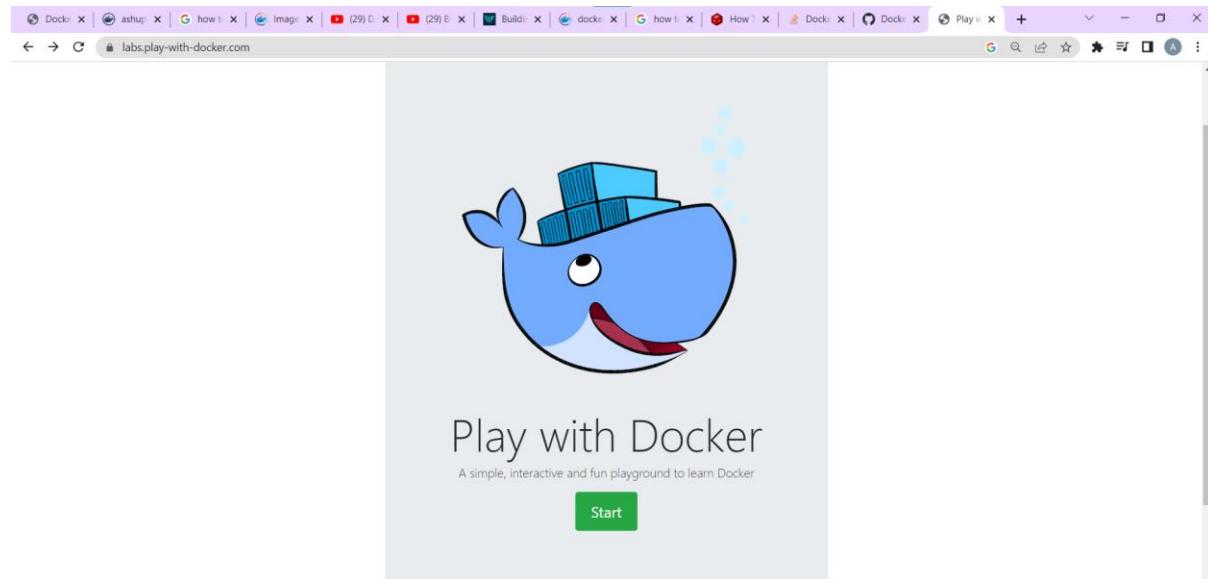
The bottom of the window shows the Windows taskbar with various pinned icons and system status information: 26°C Cloudy, 11:42 PM, 06-07-2023, ENG.

## Practical No. 03

**Aim: Working with Docker Containers and Commands: Pulling and pushing image to docker.**

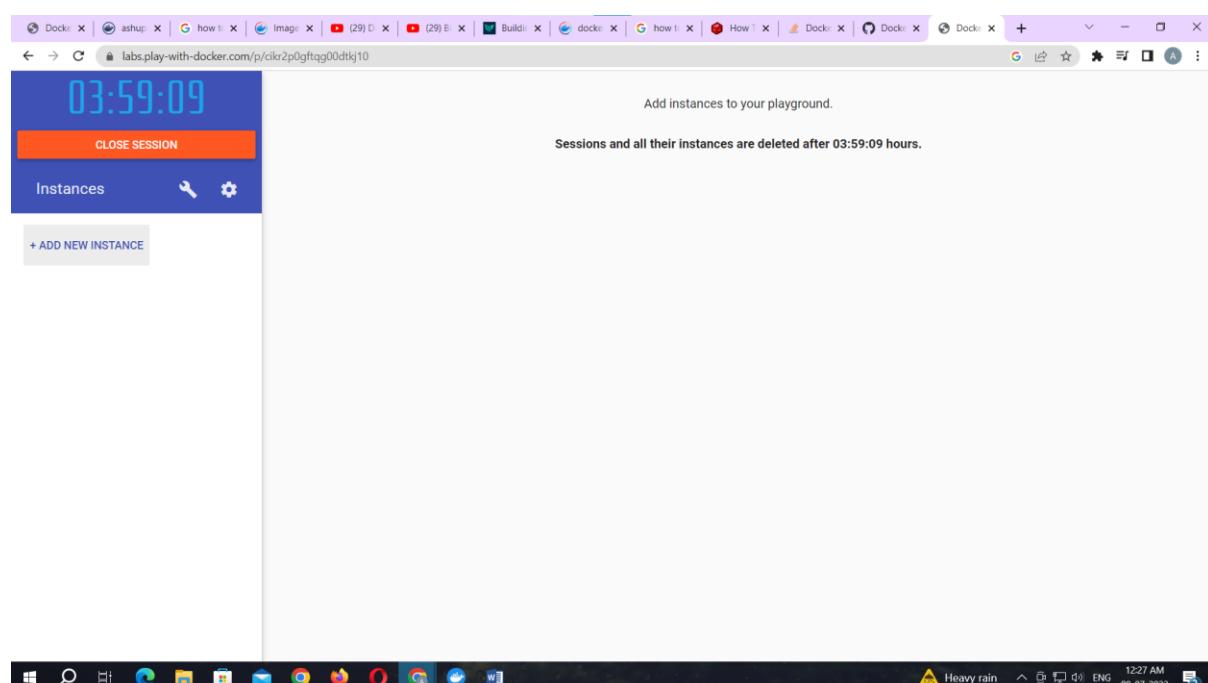
Create Docker Hub account (sign up)

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



Click on start

Add new instance



To pull and push images using docker

Command: to check docker version

docker –version

**Output:**

The screenshot shows a Windows desktop environment with a browser window open to a Docker session. The session interface includes a clock at the top left (01:49:56), a 'CLOSE SESSION' button, and a list of instances on the left. The main area displays the terminal output of the 'docker --version' command.

```

WARNING!!!!
This is a sandbox environment. Using personal credentials
is HIGHLY discouraged. Any consequences of doing so are
completely the user's responsibility.

# The PWD team.

(node1) (local) root@192.168.0.18 ~
$ docker --version
Docker version 24.0.2, build cb74dfc
(node1) (local) root@192.168.0.18 ~
$ 

```

Command: to pull ready made image

docker pull rocker/verse

**Output:**

The screenshot shows a Windows desktop environment with a browser window open to a Docker session. The session interface includes a clock at the top left (01:45:39), a 'CLOSE SESSION' button, and a list of instances on the left. The main area displays the terminal output of the 'docker pull rocker/verse' command.

```

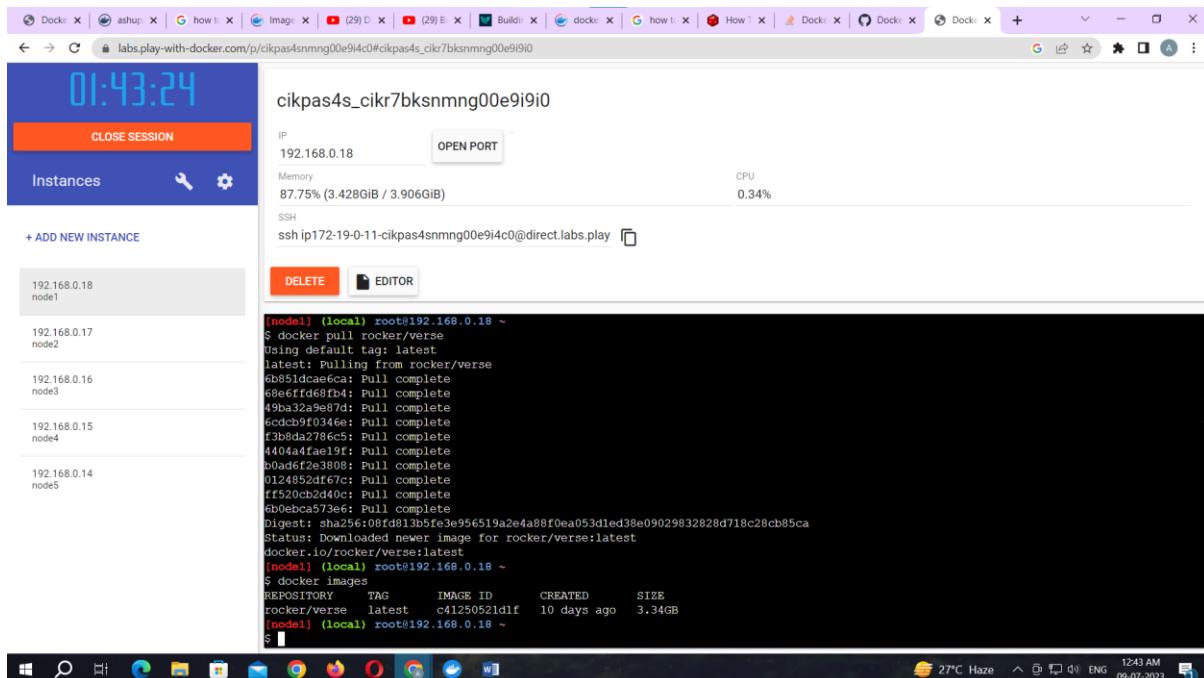
(node1) (local) root@192.168.0.18 ~
$ docker --version
Docker version 24.0.2, build cb74dfc
(node1) (local) root@192.168.0.18 ~
$ docker pull rocker/verse
Using default tag: latest
latest: Pulling from rocker/verse
db851dc4eefca: Pull complete
68eef1fde8f4d4: Pull complete
49ba32a9e0f7d: Pull complete
6ccdb9f0346e: Pull complete
f3b3da278ec5: Pull complete
4404xfae19f: Pull complete
b0ad6f2e3808: Pull complete
0124852df67c: Pull complete
ff520cb2d40c: Pull complete
6b0ebca573ee6: Pull complete
Digest: sha256:08fd13b5fe3e956519a2e4a88f0ea053d1ed38e09029832828d718c28cb85ca
Status: Downloaded newer image for rocker/verse:latest
(node1) (local) root@192.168.0.18 ~
$ 

```

Command: to check images in docker

## docker images

### Output:

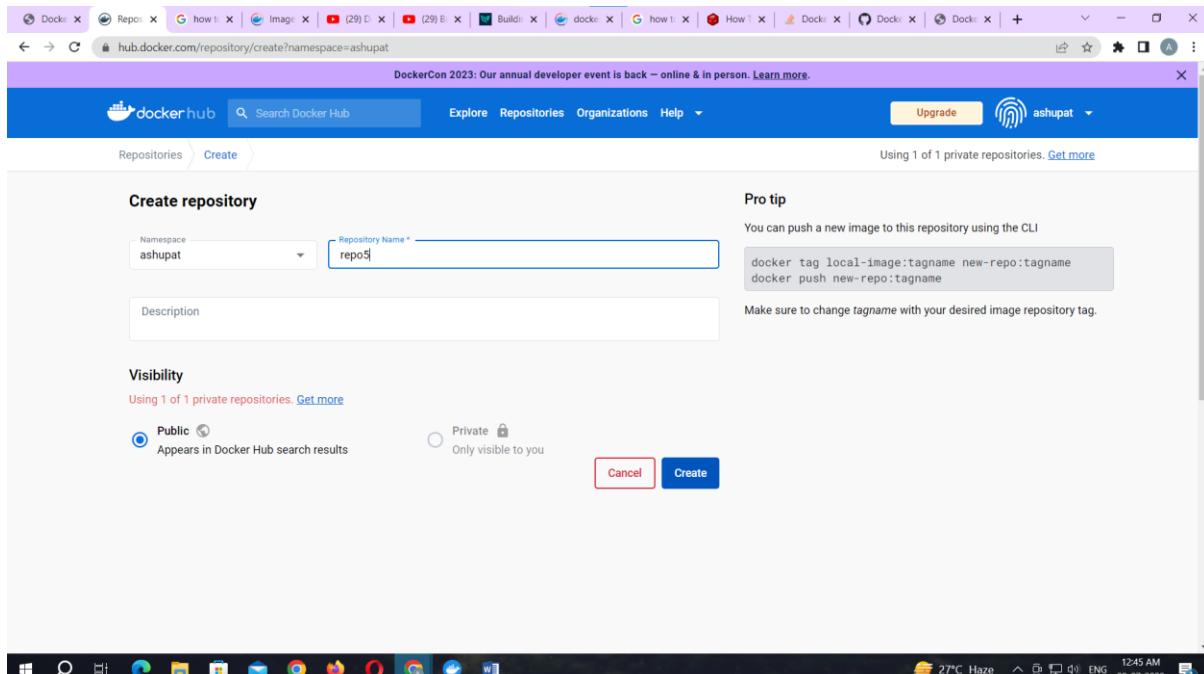


The screenshot shows a Docker session interface. At the top, there's a header bar with multiple tabs. Below it, a sidebar lists nodes: node1 (IP 192.168.0.18), node2 (IP 192.168.0.17), node3 (IP 192.168.0.16), node4 (IP 192.168.0.15), and node5 (IP 192.168.0.14). The main area is a terminal window displaying the following command output:

```
(node1) (local) root@192.168.0.18 ~
$ docker pull rocker/verse
Using default tag: latest
latest: Pulling from rocker/verse
5b851dcacefc... Pull complete
68eef7d69fb4... Pull complete
493a32a5e07d... Pull complete
6cdcbff0346e... Pull complete
f2b8da278ec5... Pull complete
4404afac19f1... Pull complete
b0ad6f2e3808... Pull complete
0124852d4f67c... Pull complete
f520cb2d40c... Pull complete
6b0ebca573e6... Pull complete
Digest: sha256:08fd13b5fe3e956519a2e4a88f0ea053d1ed38e09029832828d718c28cb85ca
Status: Downloaded newer image for rocker/verse:latest
(node1) (local) root@192.168.0.18 ~
$ docker images
REPOSITORY        TAG      IMAGE ID      CREATED       SIZE
rocker/verse      latest   c41250521d1f  10 days ago  3.34GB
(node1) (local) root@192.168.0.18 ~
$
```

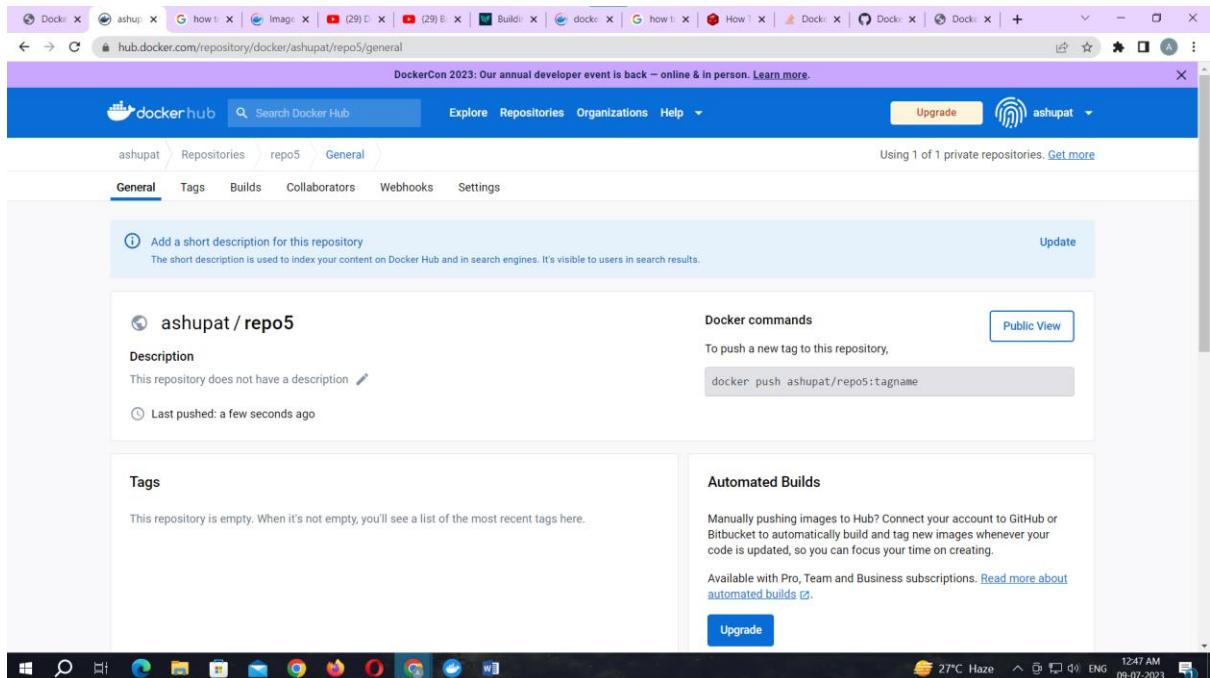
Now Login to docker hub and create repository

### Output:



Click on Create button

Now check repository created

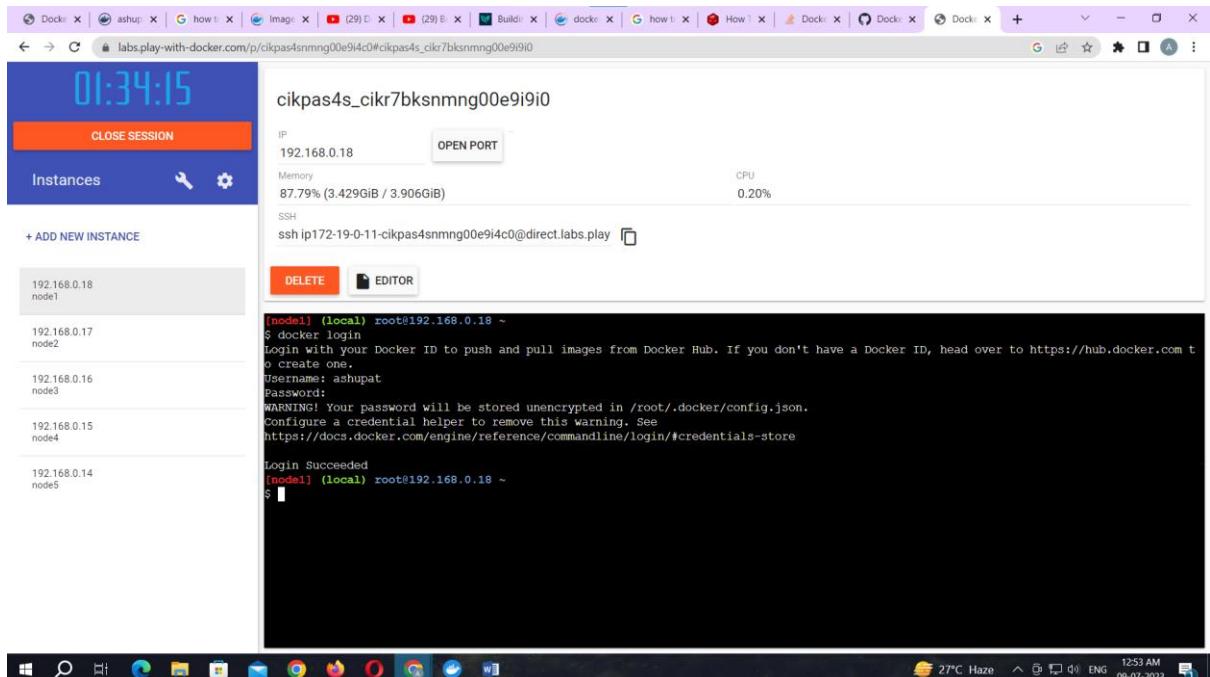


Command: to login to your docker account

docker login –username=ashupat password:

note:ashupat is my docker ID . You will use your docker ID here. And enter your password .

### Output:



Command: to tag image

docker tag c41250521d1f ashupat/repo5:newverse

note: here c41250521d1f this is image id which you can get from docker images command.

**Output:**

The screenshot shows a Docker session interface with a terminal window. The terminal window displays the following command being run:

```
(node) (local) root@192.168.0.18 ~
$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: ashupat
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
(node) (local) root@192.168.0.18 ~
$ docker images
REPOSITORY      TAG          IMAGE ID   CREATED     SIZE
rocker/verse    latest       c41250521dif  10 days ago  3.34GB
(node) (local) root@192.168.0.18 ~
$ docker tag c41250521dif ashupat/repo5:newverse
(node) (local) root@192.168.0.18 ~
$
```

Command: to push image to docker hub account

docker push ashupat/repo5:newverse

note: newverse is tag name created above.

**Output:**

The screenshot shows a Docker session interface with a terminal window. The terminal window displays the following command being run:

```
(node) (local) root@192.168.0.18 ~
$ docker images
REPOSITORY      TAG          IMAGE ID   CREATED     SIZE
rocker/verse    latest       c41250521dif  10 days ago  3.34GB
(node) (local) root@192.168.0.18 ~
$ docker tag c41250521dif ashupat/repo5:newverse
(node) (local) root@192.168.0.18 ~
$ docker push ashupat/repo5:newverse
The push refers to repository [docker.io/ashupat/repo5]
<29e3b74ddcd: Mounted from rocker/verse
edfb61d423ee: Mounted from rocker/verse
01f95d4e8ac3: Mounted from rocker/verse
08dgee8fd293: Mounted from rocker/verse
d9ba628f14d1: Mounted from rocker/verse
cc29ca15609e: Mounted from rocker/verse
6ab696bfc3fe: Mounted from rocker/verse
06757fdf4661: Mounted from rocker/verse
40419120bb3c: Mounted from rocker/verse
cddfc7c392317: Mounted from rocker/verse
newverse: digest: sha256:5cb054653d9ceaf467204391c023edfdb92b251d8ddf11a7f7d80d56c541b0 size: 2428
[node] (local) root@192.168.0.18 ~
```

Check it in docker hub now

**ashupat / repo5**

Description  
This repository does not have a description.

Last pushed: 3 minutes ago

Tag	OS	Type	Pulled	Pushed
newverse		Image	--	3 minutes ago

Tags      Automated Builds

Docker commands  
To push a new tag to this repository.  
`docker push ashupat/repo5:tagname`

Click on tags and check.

**Tags**

TAG	DIGEST	OS/ARCH	LAST PULL	COMPRESSED SIZE
newverse	<a href="#">5c1b054653d9</a>	linux/amd64	---	1.13 GB

Sort by: Newest    Filter Tags    Go to Advanced Image Management    Delete

docker pull ashupat/repo5:newver...

## Practical No. 04

**Aim: Working with Docker Containers and Commands: Build an image then push it to docker and run it**

Command: to create docker file

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

**Output:**

```

01:18:53
CLOSE SESSION

IP: 192.168.0.18
Memory: 87.81% (3.43GB / 3.906GB)
CPU: 0.15%
SSH: ssh ip172-19-0-11-cikpas4snmng00e9i4c0@direct.labs.play

DELETE EDITOR

[node1] (local) root@192.168.0.18 ~
$ cat > Dockerfile <<EOF
> FROM busybox
> CMD echo "Hello World!!!"
> EOF
[node1] (local) root@192.168.0.18 ~
$ 

```

Instances: node1, node2, node3, node4, node5

Command: to build image from docker file  
docker build -t ashupat/repo6 .

**Output:**

```

01:16:52
CLOSE SESSION

IP: 192.168.0.18
Memory: 87.57% (3.421GB / 3.906GB)
CPU: 0.52%
SSH: ssh ip172-19-0-11-cikpas4snmng00e9i4c0@direct.labs.play

DELETE EDITOR

$ EOF
[node1] (local) root@192.168.0.18 ~
$ docker build -t ashupat/repo6 .
[+] Building 1.2s (6/6) FINISHED
--> [internal] load build definition from Dockerfile
--> [internal] transferring dockerfile: 76B
--> [internal] load .dockerignore
--> [internal] load history context: 0B
--> [internal] read history index from docker.io/library/busybox:latest
--> [auth] library/busybox:pull token for registry-1.docker.io
--> [1/1] FROM docker.io/library/busybox@sha256:12759aa1214ba83e771ff252c7b1663216b192fbc5e0fb364a952f85c
--> => resolving docker.io/library/busybox@sha256:12759aa1214ba83e771ff252c7b1663216b192fbc5e0fb364a952f85c
--> => extracting sha256:809dfe20e20232a1e63363187e525d6e05db1bc41560aa63920962700e44fd
--> => sha256:5242710cd55829fc443434ff249913b57ce748989e7e6925285a29f126a79 1.46kB / 1.46kB
--> => sha256:809dfe20e20232a1e63363187e525d6e05db1bc41560aa63920962700e44fd
--> => sha256:12376a0c12759aa1214ba83e771ff252c7b1663216b192fbc5e0fb364a952f85c
--> => sha256:67a8ef88ee2ca4055f00e7cd13aedb9b24148c1451a6832d16fcc997a157eedc 528B / 528B
--> => exporting to image
--> => exporting layers
--> => writing image sha256:431fb3ccaa7118781f02f0cb31162d7027f49bd1772adce42ebc7f187aeae769
--> => naming image as ashupat/repo6
[node1] (local) root@192.168.0.18 ~
$ 

```

Command: to check docker images  
 docker images  
**Output:**

```

<> [internal] load metadata for docker.io/library/busybox:latest
<> [auth] library/busybox token for registry-1.docker.io
>> (1/1) FROM docker.io/library/busybox@sha256:1237ea0a1c1275aa1214ba83e771ff252c7b1663216b192f8e5e0fb364e952f85c
>> => resolve docker.io/library/busybox@sha256:1237ea0a1c1275aa1214ba83e771ff252c7b1663216b192f8e5e0fb364e952f85c
>> => extracting sha256:809d9e0e2032a1ee33651e87c525fd60e5d5b1bc41560aa63920962700c44fd
>> => sha256:1542710cb558396c44b34f249913bfce8748889e7e6925285a29f126aa76 1..4.6KB 0.08
>> => sha256:180949e620e2032a1ee33651fb7c525fd60e5d5b1bc41560aa63920962700c44fd 2..2.2MB 0.18
>> => sha256:1237ea0a1c1275aa1214ba83e771ff252c7b1663216b192f8e5e0fb364e952f85c 2..2.9KB 0.08
>> => sha256:67aef8862ca4055f00e7cd13aedb9b24148c1451a6832d16fc0c997a157eedc 528B 0.08
>> => exporting to image 0.08
>> => exporting layers 0.08
>> => writing image sha256:431fb3ccaa7118781f02f0cb31162d7027f49bd1772adce42ebc7f187aeae769 0.08
>> => naming to docker.io/ashupat/repo6 0.08
(node1) (local) root@192.168.0.18 ~ 0.08
$ ^C 0.08
(node1) (local) root@192.168.0.18 ~ 0.08
$ docker images 0.08
REPOSITORY TAG IMAGE ID CREATED SIZE
ashupat/repo6 latest 431fb3ccaa71 10 days ago 4.26MB
ashupat/repo5 newverse c41250521d1f 10 days ago 3.34GB
rocker/verse latest c41250521d1f 10 days ago 3.34GB
(node1) (local) root@192.168.0.18 ~ 0.08
$ 0.08

```

Command: to tag image

docker tag 431fb3ccaa71 ashupat/repo6:image

Command: to push image to docker hub

docker push ashupat/repo6:image

```

<> => sha256:2376a0c12759aa1214ba83e771ff252c7b1663216b192f8e5e0fb364e952f85c 2..2.9KB 0.05
>> => sha256:67aef8862ca4055f00e7cd13aedb9b24148c1451a6832d16fc0c997a157eedc 528B 0.05
>> => exporting to image 0.05
>> => exporting layers 0.05
>> => writing image sha256:431fb3ccaa7118781f02f0cb31162d7027f49bd1772adce42ebc7f187aeae769 0.05
>> => naming to docker.io/ashupat/repo6 0.05
(node1) (local) root@192.168.0.18 ~ 0.05
$ ^C 0.05
(node1) (local) root@192.168.0.18 ~ 0.05
$ docker images 0.05
REPOSITORY TAG IMAGE ID CREATED SIZE
ashupat/repo6 latest 431fb3ccaa71 10 days ago 4.26MB
ashupat/repo5 newverse c41250521d1f 10 days ago 3.34GB
rocker/verse latest c41250521d1f 10 days ago 3.34GB
(node1) (local) root@192.168.0.18 ~ 0.05
$ docker tag 431fb3ccaa71 ashupat/repo6:image 0.05
(node1) (local) root@192.168.0.18 ~ 0.05
$ docker push ashupat/repo6:image 0.05
The push refers to repository [docker.io/ashupat/repo6]
fb4513d4fb7: Mounted from library/busybox
image: digest: sha256:0af1c0af08af6bec58b5034e7056d1e0794e4c61918848b79131b221b9ef19e size: 527
(node1) (local) root@192.168.0.18 ~ 0.05
$ 0.05

```

Now check it on docker hub.

The screenshot shows the Docker Hub interface. At the top, there are several tabs and a search bar. Below the header, the user's repositories are listed:

- ashupat / repo6**: Contains: Image | Last pushed: 2 minutes ago. Status: Inactive, stars: 0, downloads: 0, public.
- ashupat / repo5**: Contains: Image | Last pushed: 20 minutes ago. Status: Inactive, stars: 0, downloads: 0, public.
- ashupat / repo3**: Contains: Image | Last pushed: an hour ago. Status: Inactive, stars: 0, downloads: 0, public.
- ashupat / newrepo**: Contains: Image | Last pushed: 3 days ago. Status: Inactive, stars: 0, downloads: 0, public.
- ashupat / repo1**: Contains: Image | Last pushed: 3 months ago. Status: Inactive, stars: 0, downloads: 0, private.

A sidebar on the right provides links to "Create an Organization" and "Manage Docker Hub repositories with your team". Below the sidebar is a promotional image for DockerCon 2023.

Command: to run docker image:

docker run kbdocker11/repo2

**Output:**

The screenshot shows a terminal window on a browser. The URL is [https://labs.play-with-docker.com/p/cikpas4snmng00e9i4c0/cikpas4s\\_cikr7bksnmng00e9i9i0](https://labs.play-with-docker.com/p/cikpas4snmng00e9i4c0/cikpas4s_cikr7bksnmng00e9i9i0). The terminal output is as follows:

```

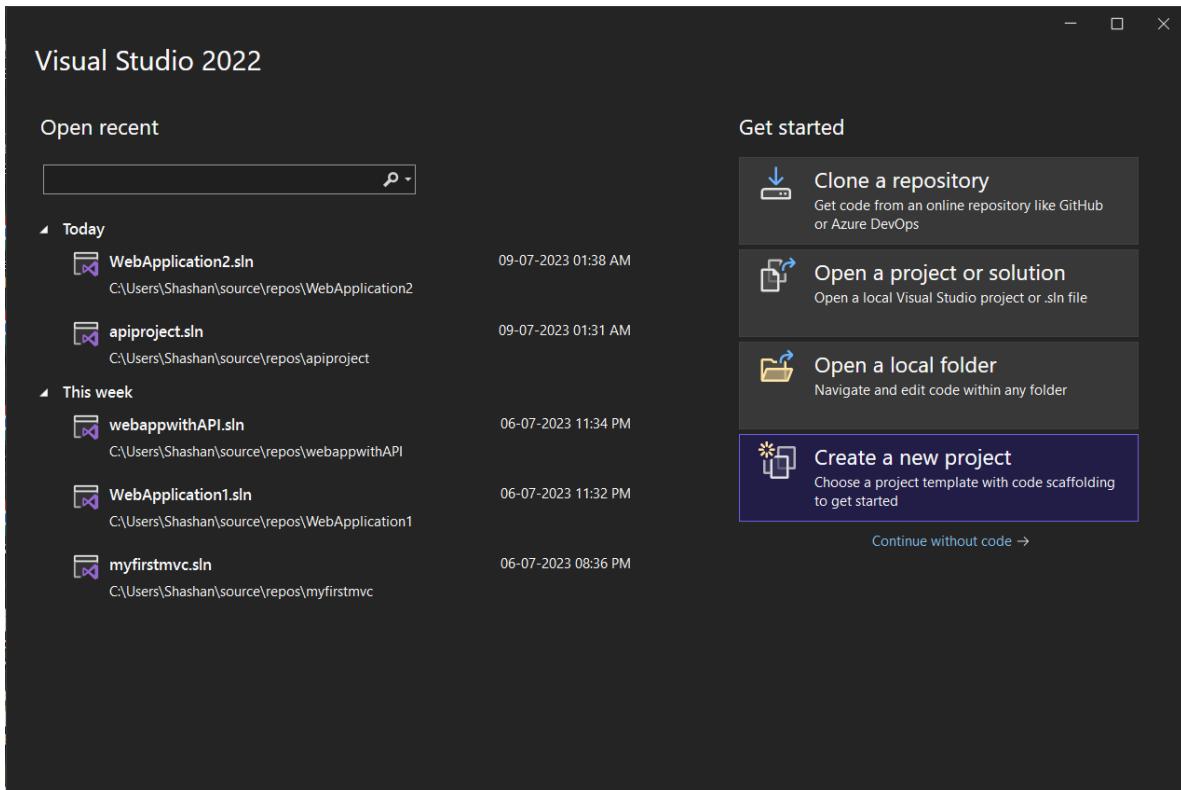
01:05:20
cikpas4s_cikr7bksnmng00e9i9i0
CLOSE SESSION
IP 192.168.0.18 OPEN PORT
Memory 85.53% (3.341GiB / 3.906GiB) CPU 0.14%
SSH ssh ip172-19-0-11-cikpas4snmng00e9i4c0@direct.labs.play
DELETE EDITOR
>> => exporting layers
>> => writing image sha256:431fb3ccaa7118781f02f0cb31162d7027f49bd1772adce42ebc7f187aaeae769
>> => naming to docker.io/ashupat/repo6
[node1] (local) root@192.168.0.18 ~
$ ls
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ashupat/repo6 latest 431fb3ccaa71 10 days ago 4.26MB
ashupat/repo5 newverse c41250521d1f 10 days ago 3.34GB
rocker/verse latest c41250521d1f 10 days ago 3.34GB
[node1] (local) root@192.168.0.18 ~
$ docker tag 431fb3ccaa71 ashupat/repo6:image
[node1] (local) root@192.168.0.18 ~
$ docker push ashupat/repo6:image
The push refers to repository [docker.io/ashupat/repo6]
fb4513d4fb7: Mounted from library/busybox
image: digest: sha256:02af1c0af08af6bec58b5034e7056d1e0794e4c61918848b79131b221be9f19e size: 527
[node1] (local) root@192.168.0.18 ~
$ docker run ashupat/repo6
Hello World!!!
[node1] (local) root@192.168.0.18 ~
$ 

```

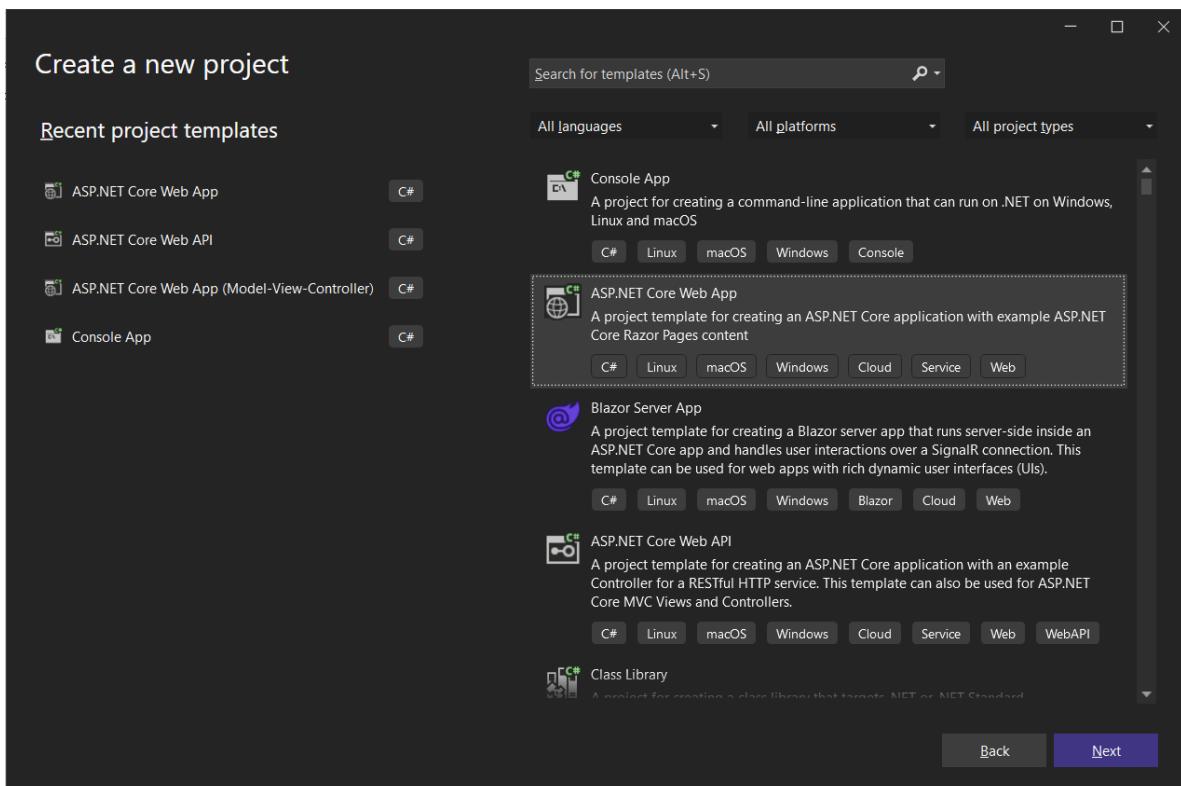
## Practical No. 05

**Aim: Build a Web App and publish it to Docker.**

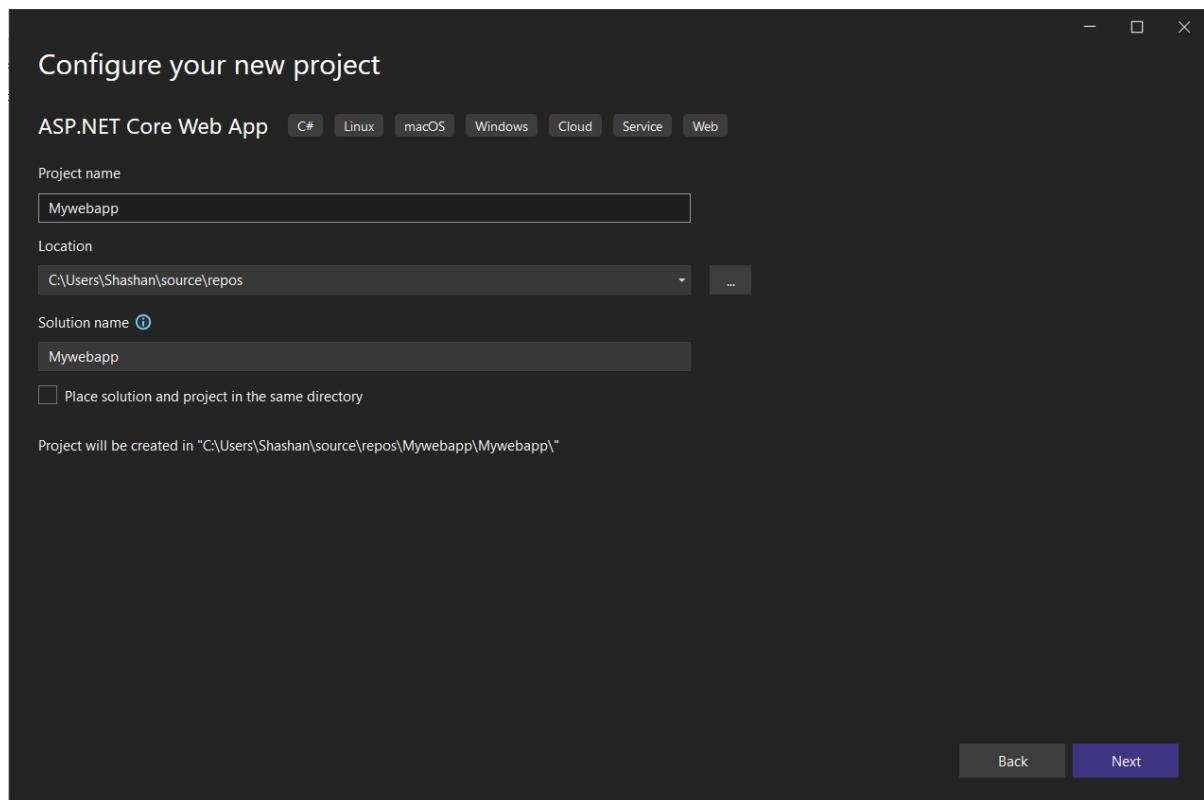
Let's start by creating a new project in the **Microsoft Visual Studio**



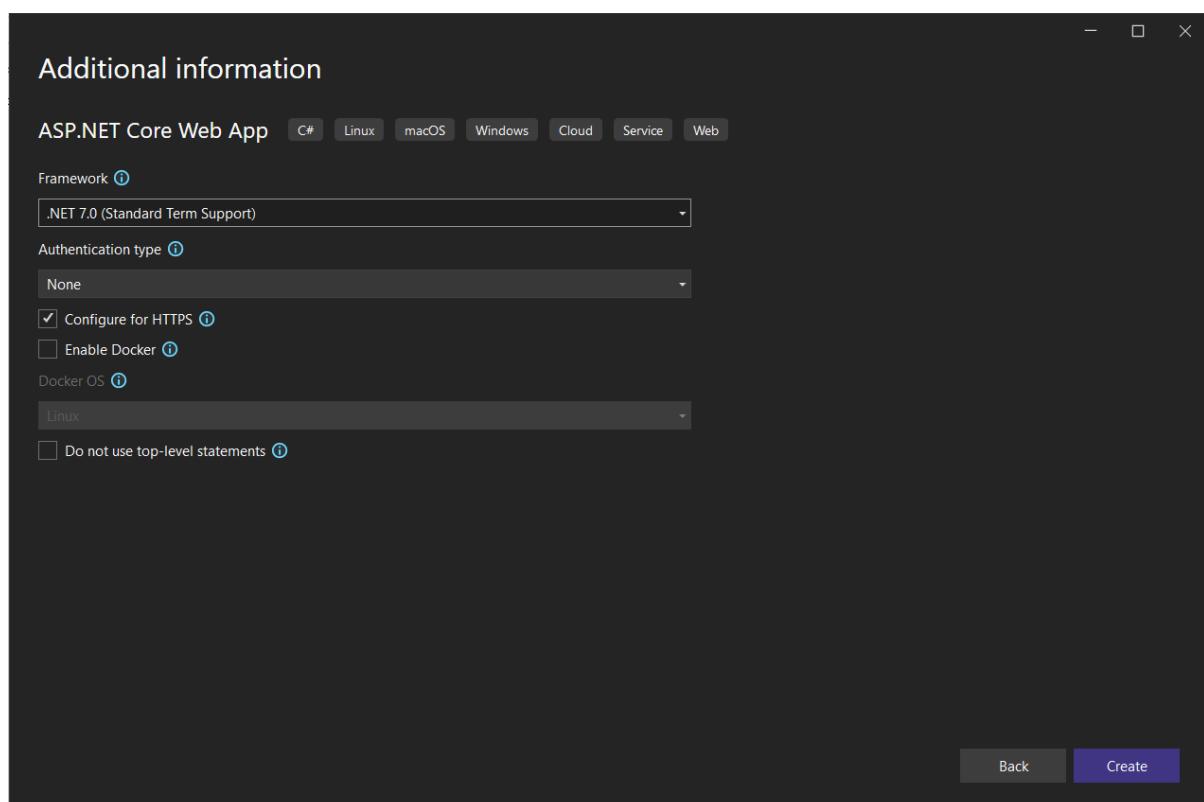
Select the **ASP.NET Core Web App** form the various options.



Give an appropriate name to the project.



Click on the **Create** button.



Once the project is created, Try executing the project to make sure that it works.

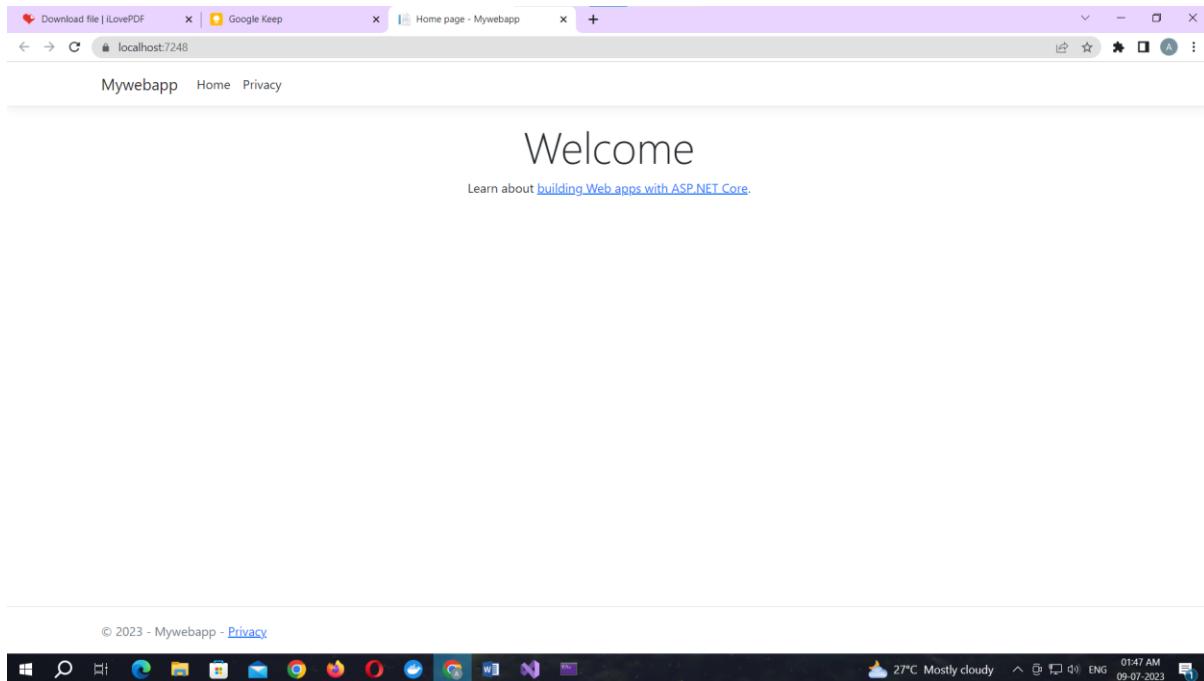
The screenshot shows the Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and Search (Ctrl+Q). The title bar says "Mywebapp". The Solution Explorer on the right shows a single project named "Mywebapp" with files like Connected Services, Dependencies, Properties, wwwroot, Pages, and appsettings.json. The code editor window displays the "Program.cs" file with C# code for setting up an ASP.NET Core application. Below the code editor is the Output window, which is currently empty. The taskbar at the bottom shows various pinned icons and the system tray with weather information (27°C Mostly cloudy) and system status.

```

1 var builder = WebApplication.CreateBuilder(args);
2
3 // Add services to the container.
4 builder.Services.AddRazorPages();
5
6 var app = builder.Build();
7
8 // Configure the HTTP request pipeline.
9 if (app.Environment.IsDevelopment())
10 {
11     app.UseExceptionHandler("/Error");
12     // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
13     app.UseHsts();
14 }
15
16 app.UseHttpsRedirection();
17 app.UseStaticFiles();
18
19 app.UseRouting();
20
21 app.UseAuthorization();
22
23 app.MapRazorPages();
24
25 app.Run();
26

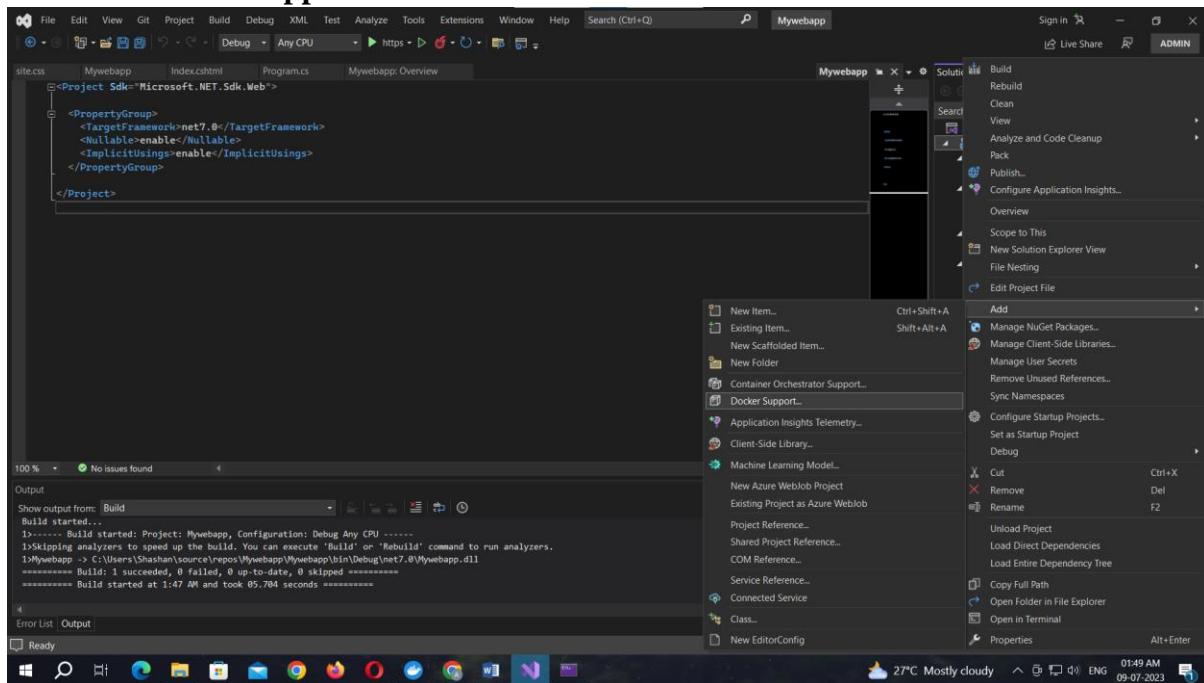
```

## Output:

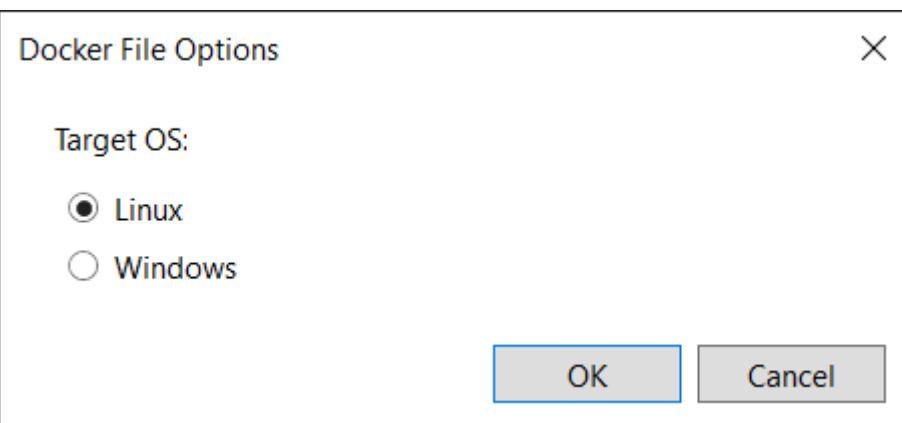


**Right click on the app name then select the Add option form the menu.**

**Select the Docker Support under the Add menu**



Select the Target OS as Linux and click OK button.



The file naming Dockerfile will appear

A screenshot of the Visual Studio IDE interface. The main window shows a Dockerfile with the following content:

```

1 #See https://aka.ms/customizecontainer to learn how to customize your debug container and how Visual Studio uses this Dockerfile to build your images for faster debugging.
2
3 FROM mcr.microsoft.com/dotnet/aspnet:7.0 AS base
4 WORKDIR /app
5 EXPOSE 80
6 EXPOSE 443
7
8 FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build
9 WORKDIR /src
10 COPY ["Mywebapp/Mywebapp.csproj", "Mywebapp/"]
11 RUN dotnet restore "Mywebapp/Mywebapp.csproj"
12 COPY .
13 WORKDIR "/src/Mywebapp"
14 RUN dotnet build "Mywebapp.csproj" -c Release -o /app/build
15
16 FROM build AS publish
17 RUN dotnet publish "Mywebapp.csproj" -c Release -o /app/publish /p:UseAppHost=false
18
19 FROM base AS final
20 WORKDIR /app
21 COPY --from=publish /app/publish .
22 ENTRYPOINT ["dotnet", "Mywebapp.dll"]

```

The Solution Explorer on the right shows a project named 'Mywebapp' with files like Dockerfile, .dockerignore, Program.cs, and launchSettings.json.

Code for Dockerfile:

#See <https://aka.ms/customizecontainer> to learn how to customize your debug container and how Visual Studio uses this Dockerfile to build your images for faster debugging.

FROM mcr.microsoft.com/dotnet/aspnet:7.0 AS base

WORKDIR /app

EXPOSE 80

EXPOSE 443

FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build

WORKDIR /src

COPY ["Mywebapp/Mywebapp.csproj", "Mywebapp/"]

RUN dotnet restore "Mywebapp/Mywebapp.csproj"

COPY ..

WORKDIR "/src/Mywebapp"

RUN dotnet build "Mywebapp.csproj" -c Release -o /app/build

FROM build AS publish

RUN dotnet publish "Mywebapp.csproj" -c Release -o /app/publish /p:UseAppHost=false

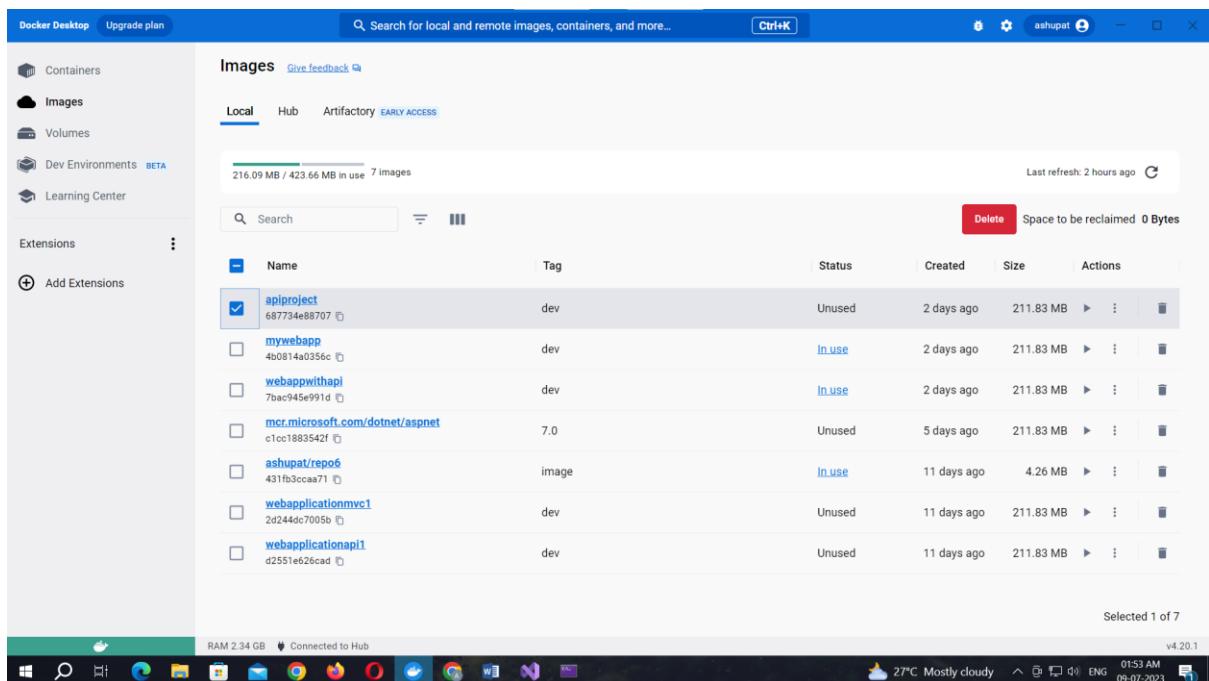
FROM base AS final

WORKDIR /app

COPY --from=publish /app/publish .

ENTRYPOINT ["dotnet", "Mywebapp.dll"]

The project is uploaded to the docker.



Docker Desktop Upgrade plan

Images Give feedback

Local Hub Artifactory EARLY ACCESS

216.09 MB / 423.66 MB in use 7 images Last refresh: 2 hours ago

Search Delete Space to be reclaimed 0 Bytes

Name	Tag	Status	Created	Size	Actions
apiproject	dev	Unused	2 days ago	211.83 MB	⋮
mywebapp	dev	In use	2 days ago	211.83 MB	⋮
webappwithapi	dev	In use	2 days ago	211.83 MB	⋮
mcr.microsoft.com/dotnet/aspnet	7.0	Unused	5 days ago	211.83 MB	⋮
ashupat/repo6	image	In use	11 days ago	4.26 MB	⋮
webapplicationmvc1	dev	Unused	11 days ago	211.83 MB	⋮
webapplicationapi1	dev	Unused	11 days ago	211.83 MB	⋮

Selected 1 of 7

RAM 2.34 GB Connected to Hub v4.20.1

27°C Mostly cloudy 01:53 AM ENG 09-07-2023

## Practical No. 06

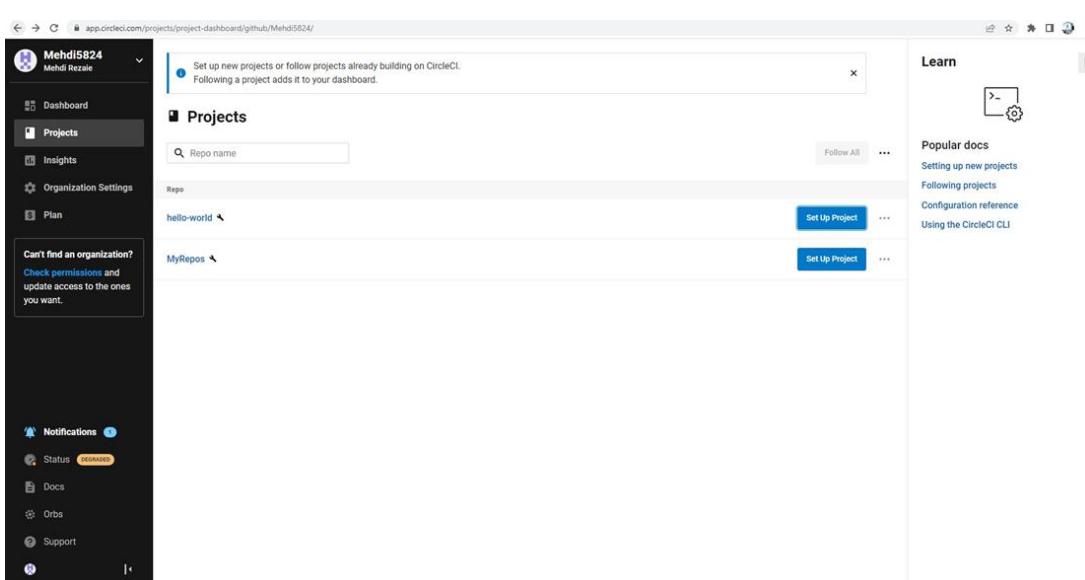
### Aim: Working with the CircleCI

#### Create a Repository

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

The screenshot shows the GitHub 'Create a new repository' interface. The repository name is set to 'hello-world-p'. The 'Public' option is selected. Other settings include 'Add a README file' checked, 'Add .gitignore' with 'None' selected, and 'Choose a license' with 'None' selected. A note at the bottom says 'You are creating a public repository in your personal account.' A 'Create repository' button is at the bottom right.

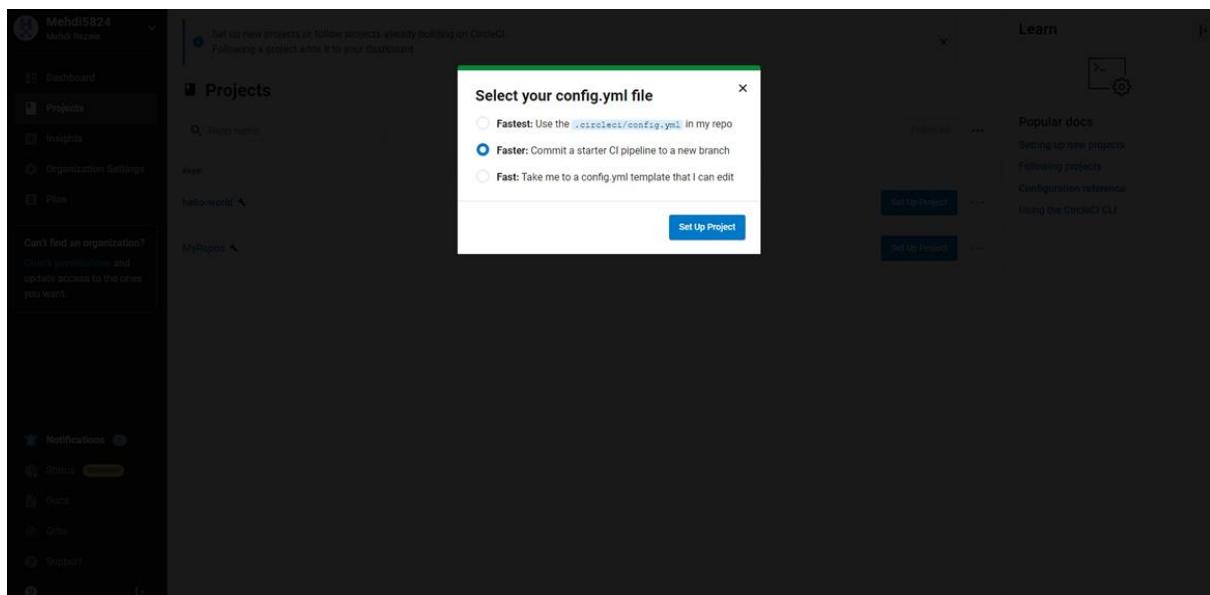
Login to Circle CI <https://app.circleci.com/> Using GitHub Login, Once logged in navigate to Projects.



## Set up CircleCI

1. Navigate to the CircleCI Projects page. If you created your new repository under an organization, you will need to select the organization name.
2. You will be taken to the Projects dashboard. On the dashboard, select the project you want to set up (hello-world).
3. 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.



## Your 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

The screenshot shows the CircleCI Project hello-world dashboard. At the top, there are navigation links for Dashboard, Project, All Pipelines, and hello-world. Below this, a section titled "hello-world" with a green checkmark icon and "Add team members" is displayed. A "Filters" bar includes dropdowns for "Everyone's Pipelines" (set to "hello-world"), "Branches" (set to "All Branches"), and a dropdown menu. On the right, there are buttons for "Edit Config", "Trigger Pipeline", and "Project Settings". A toggle switch for "Auto-expand" is also present. The main table lists the pipeline runs:

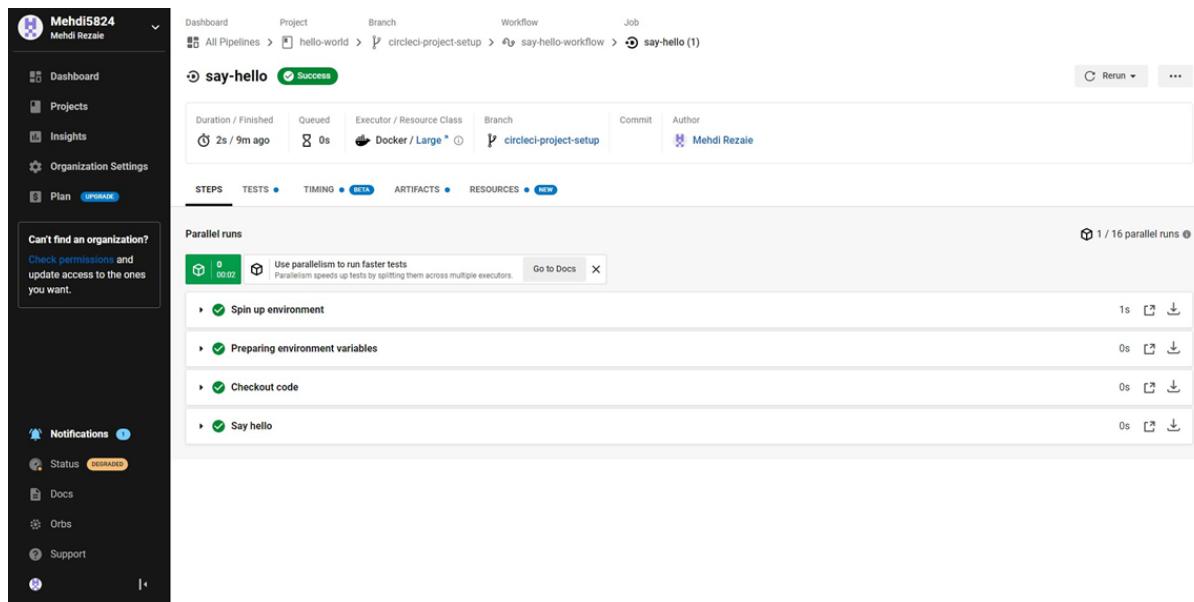
Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
hello-world 1	<span style="color: green;">Success</span>	say-hello-workflow	circleci-project-setup 6b308f0	3m ago	5s	<span style="color: green;">↻</span> <span style="color: green;">↻</span> <span style="color: green;">↻</span> <span style="color: green;">...</span>

Select “say-hello” Job with a green tick

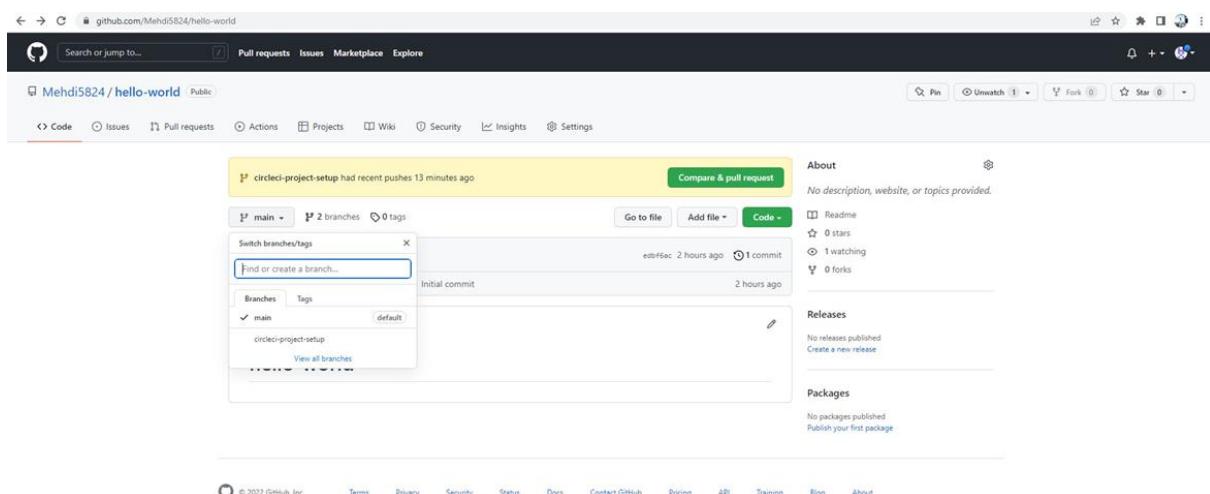
The screenshot shows the details of the "say-hello" job from the previous screenshot. The URL is "https://app.circleci.com/pipelines/hello-world/circleci-project-setup/1/workflows/say-hello-workflow/jobs/say-hello". The job status is "Success". The job details table includes columns for Duration / Finished, Branch, Commit, and Author:

Duration / Finished	Branch	Commit	Author
5s / 8m ago	circleci-project-setup	6b308f0	Mehdi5824

Below the table, a search bar shows "say-hello" and a duration of "3s". At the bottom, a "Did you know?" statistic states: "CircleCI teams that commit 4x as often fix failed builds 2x faster." It compares 20 pipelines/day (70 min to recover) to 5 pipelines/day (142 min). There are "Learn more" and "X" buttons.



Select Branch and option circleci-project-setup



## 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.

The figure consists of three vertically stacked screenshots of a GitHub repository page. All three screenshots show the same basic layout: a header with navigation links (Search or jump to..., Pull requests, Issues, Marketplace, Explore), a main content area with a file list, and a footer with GitHub links (© 2022 GitHub, Inc., Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, About).

**Screenshot 1:** The repository has a single commit named "Add .circleci/config.yml". The commit message is "Add .circleci/config.yml". The commit was made by "Mehdi5824" at 6d388f0 14 minutes ago. The commit message is "Initial commit". The file was added at 6d388f0 14 minutes ago.

```

    diff --git a/.circleci/config.yml b/.circleci/config.yml
    new file mode 100644
    index 0000000..e314471
    --- /dev/null
    +++ b/.circleci/config.yml
    @@ -1 +1 @@
    # This is a sample configuration.
    # See https://circleci.com/docs/2.0/configuration-reference/
    jobs:
      - job: say-hello
        steps:
          - checkout
          - run:
              name: "Say hello"
              command: "echo Hello, World!"

```

**Screenshot 2:** The repository now has two commits. The first commit is the same as in Screenshot 1. The second commit is a merge commit with the message "Merge pull request #1 from Mehdi5824". The merge commit was made by "Mehdi5824" at 6d388f0 15 minutes ago. The merge commit message is "Merge pull request #1 from Mehdi5824". The file was updated at 6d388f0 15 minutes ago.

```

    diff --git a/.circleci/config.yml b/.circleci/config.yml
    index e314471..a11a2c0 100644
    --- a/.circleci/config.yml
    +++ b/.circleci/config.yml
    @@ -1 +1 @@
    # This is a sample configuration.
    # See https://circleci.com/docs/2.0/configuration-reference/
    jobs:
      - job: say-hello
        steps:
          - checkout
          - run:
              name: "Say hello"
              command: "echo Hello, World!"

```

**Screenshot 3:** The repository has a single commit named "Add .circleci/config.yml". The commit message is "Add .circleci/config.yml". The commit was made by "Mehdi5824" at 6d388f0 16 minutes ago. The commit message is "Initial commit". The file was added at 6d388f0 16 minutes ago.

```

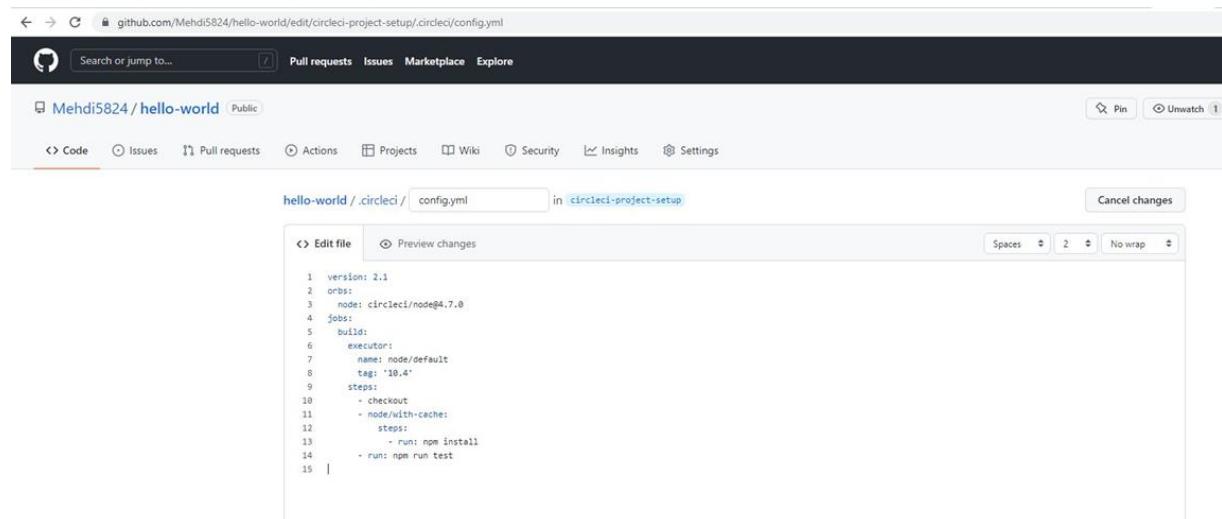
    diff --git a/.circleci/config.yml b/.circleci/config.yml
    new file mode 100644
    index 0000000..e314471
    --- /dev/null
    +++ b/.circleci/config.yml
    @@ -1 +1 @@
    # This is a sample configuration.
    # See https://circleci.com/docs/2.0/configuration-reference/
    jobs:
      - job: say-hello
        steps:
          - checkout
          - run:
              name: "Say hello"
              command: "echo Hello, World!"

```

Let's use the Node orb. Replace the existing config by pasting the following code:

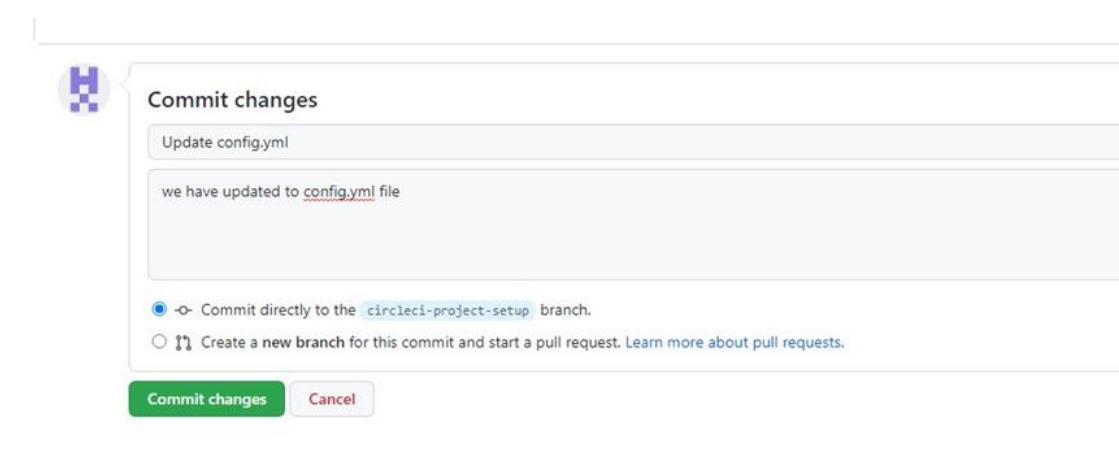
```
version: 2.1
orbs:
  node: cirlceci/node@4.7.0
jobs:
  build:
    executor:
      name: node/default
      tag: '10.4'
    steps:
      - checkout
      - node/with-cache:
          steps:
            - run: npm install
            - run: npm run test
```

The GitHub file editor should look like this

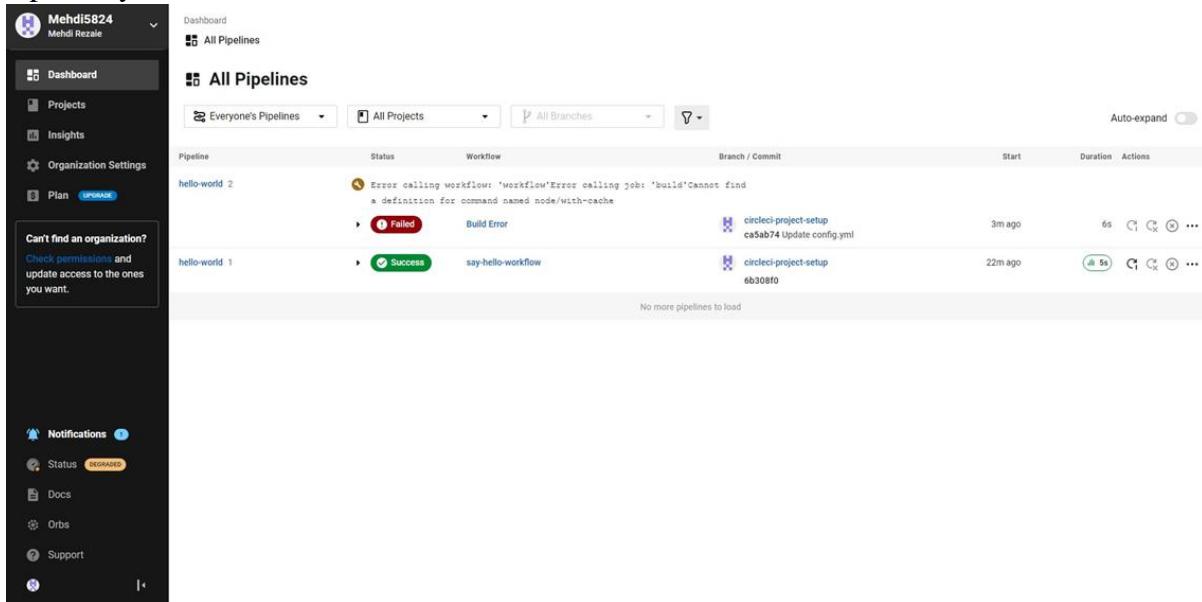


```
version: 2.1
orbs:
  node: cirlceci/node@4.7.0
jobs:
  build:
    executor:
      name: node/default
      tag: '10.4'
    steps:
      - checkout
      - node/with-cache:
          steps:
            - run: npm install
            - run: npm run test
```

Scroll down and Commit your changes on GitHub



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.



The screenshot shows the CircleCI interface. On the left, there's a sidebar with user information (Mehdi5824, Mehdi Rezaie) and navigation links for Dashboard, Projects, Insights, Organization Settings, Plan, Notifications (1), Status (DEGRADED), Docs, Orbs, and Support. The main area is titled "All Pipelines" and displays two entries:

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
hello-world 2	Failed	Error calling workflow: 'workflow' Error calling job: 'build' Cannot find a definition for command named node/with-cache	circleci-project-setup ca5ab74 Update config.yml	3m ago	6s	
hello-world 1	Success	say-hello-workflow	circleci-project-setup 6b308f0	22m ago		

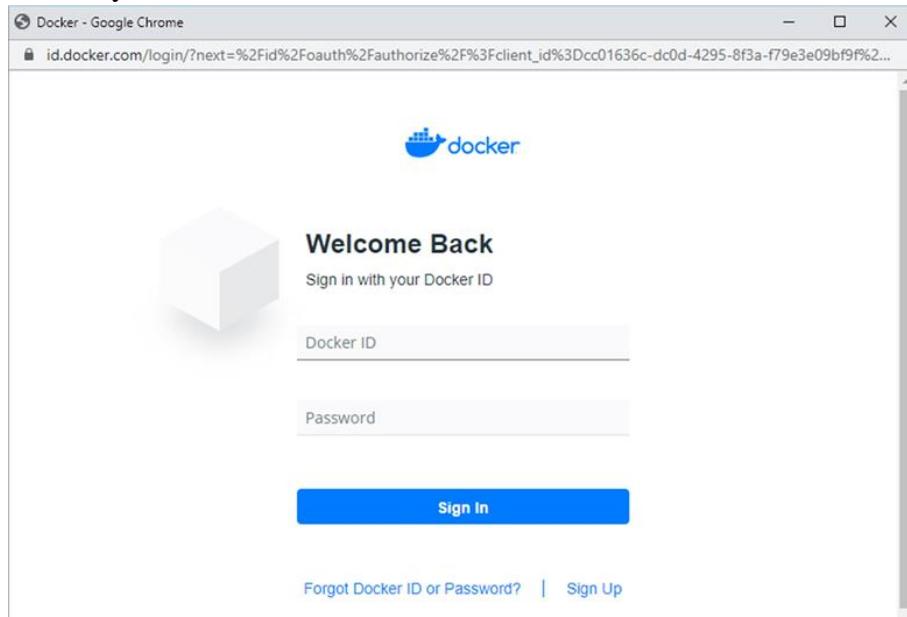
A message at the bottom right says "No more pipelines to load".

## Practical No. 07

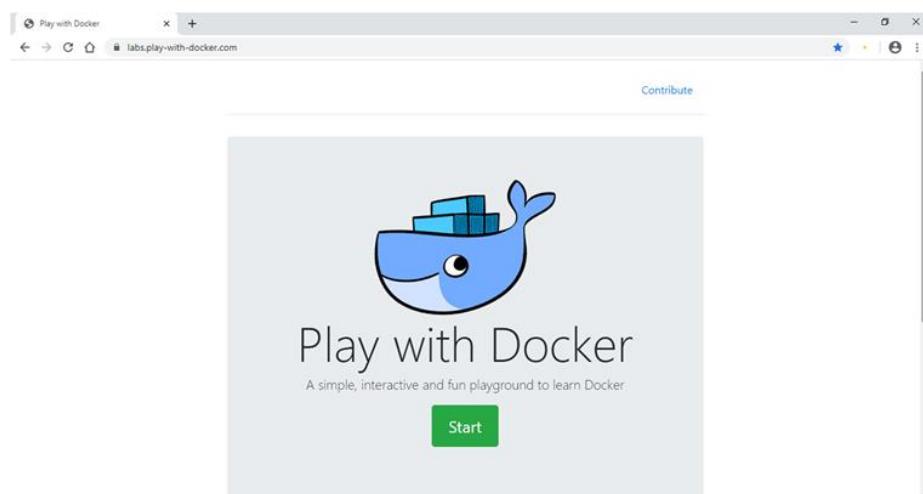
**Aim: Running location service in Docker.**

(Create docker hub login first to use it in play with docker)

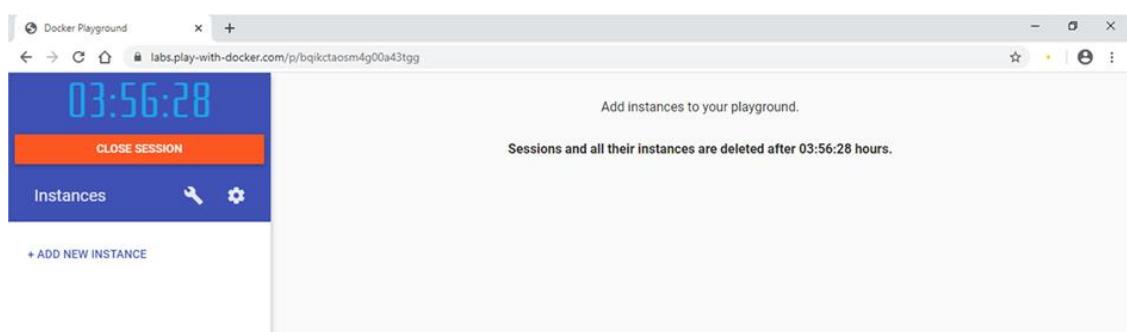
Now login in to Play-With-Docker

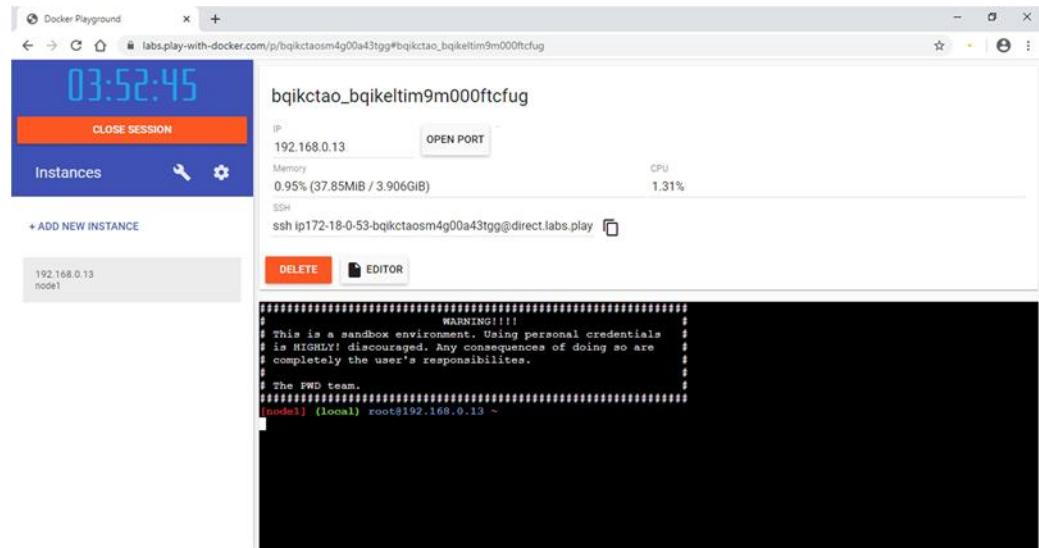


Click on Start.



Click on Add New Instance.



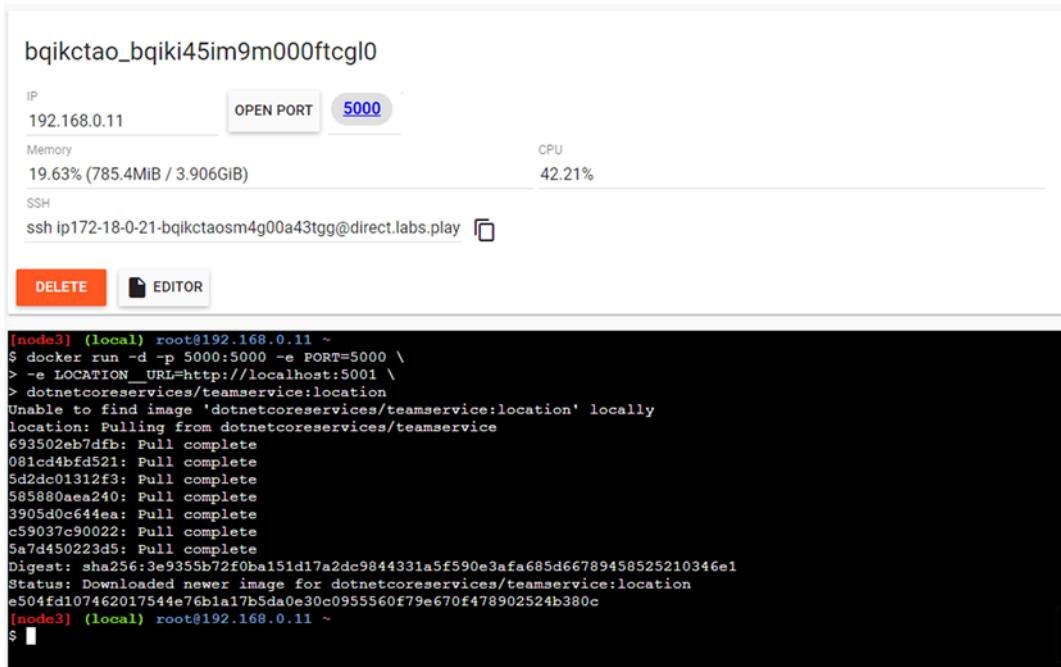


Start typing following commands

Command: To run teamservice

```
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)



Command: to run location service

```
docker run -d -p 5001:5001 -e PORT=5001 \
dotnetcoreservices/locationservice:nodb
```

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

```
[node3] (local) root@192.168.0.11 ~
$ docker run -d -p 5001:5001 -e PORT=5001 \
> dotnetcoreservices/locationservice:nodb
Unable to find image 'dotnetcoreservices/locationservice:nodb' locally
nodb: Pulling from dotnetcoreservices/locationservice
693502eb7dfb: Already exists
081cd4bfd521: Already exists
5d2dc01312f3: Already exists
585880aea240: Already exists
3905d0c64ea1: Already exists
c59037c90022: Already exists
dbc03883a4ca: Full complete
Digest: sha256:5f7aca33c5e2117e04f58a59e0cf96fd20d5cbf2cf66c3cd708118d573255168
Status: Downloaded newer image for dotnetcoreservices/locationservice:nodb
16994c92a644eaf5c0bb7efdf9f230df16cbee84e4c48b9cd434b0e826adb3159
(node3) (local) root@192.168.0.11 ~
$ 
```

Command : to check running images in docker  
docker images

**Output:**

```
[node3] (local) root@192.168.0.11 ~
$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
dotnetcoreservices/teamservice   location  b27d0de8f2de  3 years ago  886MB
dotnetcoreservices/locationservice nodb     03339f0ea9dd  3 years ago  883MB
[node3] (local) root@192.168.0.11 ~
$ 
```

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
```

**Output:**

```
[node3] (local) root@192.168.0.11 ~
$ curl -H "Content-Type:application/json" -X POST -d \
> '{"id":"e52baa63-d511-417e-9e54-7aab04286281", "name":"KC"}' http://localhost:5000/teams
{"name":"KC","id":"e52baa63-d511-417e-9e54-7aab04286281","members":[]}[node3] (local) root@192.168.0.11 ~
$ 
```

Command: To confirm that team is added

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

### Output:

```
[node3] (local) root@192.168.0.11 ~
$ curl -H "Content-Type:application/json" -X POST -d \
> '[{"id":"e52baa63-d511-417e-9e54-7aab04286281", "name":"KC"}]' http://localhost:5000/teams
{"name":"KC","id":"e52baa63-d511-417e-9e54-7aab04286281","members":[]}[node3] (local) root@192.168.0.11 ~
$ curl http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281
{"name":"KC","id":"e52baa63-d511-417e-9e54-7aab04286281","members":[]}[node3] (local) root@192.168.0.11 ~
$ [
```

Command : to add new member to team

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

### Output:

```
[node1] (local) root@192.168.0.23 ~
$ curl -H "Content-Type:application/json" -X POST -d \
> '[{"id":"63e7acf8-8fae-42ce-9349-3c8593ac8292", "firstName":"Kirti", "lastName":"Bhatt"}]' http://localhost:5000/
teams/e52baa63-d511-417e-9e54-7aab04286281/members
{"teamID":"e52baa63-d511-417e-9e54-7aab04286281","memberID":"63e7acf8-8fae-42ce-9349-3c8593ac8292"}[node1] (local)
) root@192.168.0.23 ~
$ [
```

Command: To confirm member added

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

### Output:

```
[node1] (local) root@192.168.0.23 ~
$ curl http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281
{"name":"KC","id":"e52baa63-d511-417e-9e54-7aab04286281","members":null,[{"id":"63e7acf8-8fae-42ce-9349-3c8593ac8
292","firstName":"Kirti","lastName":"Bhatt"}]}[node1] (local) root@192.168.0.23 ~
$ [
```

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:**

```
[node1] (local) root@192.168.0.23 ~
$ 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,"mem
berID":"63e7acf8-8fae-42ce-9349-3c8593ac8292"}[node1] (local) root@192.168.0.23 ~
$
```

Command: To confirm location is added in member

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

**Output:**

```
[node1] (local) root@192.168.0.23 ~
$ 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,"mem
berID":"63e7acf8-8fae-42ce-9349-3c8593ac8292"}][node1] (local) root@192.168.0.23 ~
$
```