



## Microsoft Azure Mini Project

### Group Members:-

Divesh Kailas Kolhe

Shubham D. Mohite

Nikhil Satish Sahu



**Microsoft**  
**Future Ready Talent**  
**Internship**

The graphic features a central purple globe with a pink base, surrounded by four stylized human figures in business attire. One figure stands on the left, another on the right, and two are positioned in front of the globe. They are interacting with various floating digital screens displaying icons like a magnifying glass, a bar chart, and a gear. The background is white with a light gray diagonal band on the right side containing the text.

# **Title: “Efficient CI/CD Pipeline for Azure Web App Deployment.”**

## **Overview:**

The project focuses on solving the problem of manual and error-prone deployment processes for web applications on the Azure platform. It addresses the need for a streamlined, efficient, and reliable deployment workflow.

### **Core Features:**

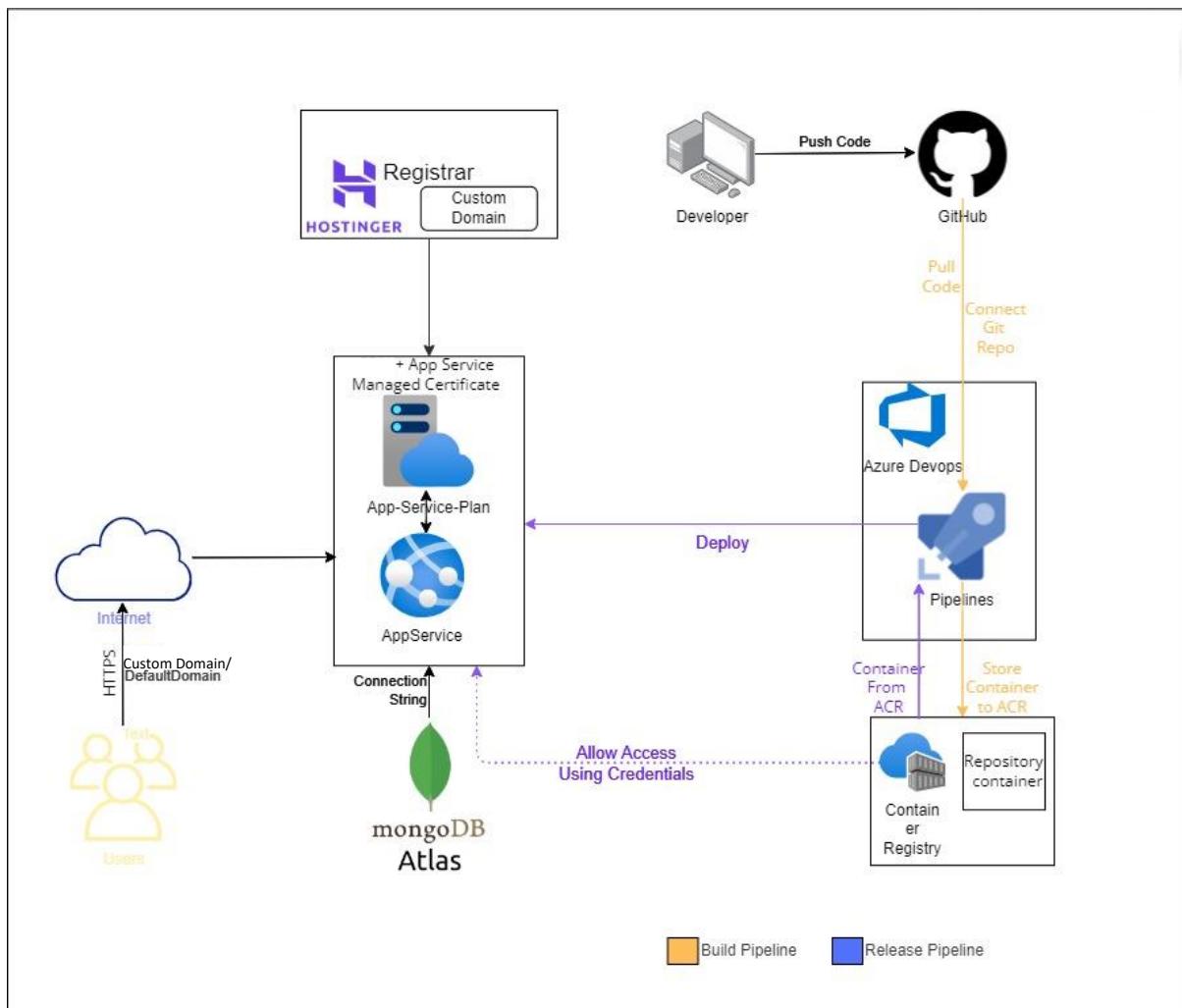
**Continuous Integration and Continuous Deployment (CI/CD):** The project implements a robust CI/CD pipeline that automates the deployment process. Developers push code changes to a GitHub repository, triggering a series of automated actions that build a Docker container of the web application, store it in an Azure Container Registry (ACR), and deploy it to an Azure App Service.

### **Azure Services Utilized:**

1. **Azure Container Registry (ACR):** Used for storing and version-controlling Docker container images.
2. **Azure App Service:** The platform where the web application is hosted, offering auto-scaling and load balancing.
3. **Azure DevOps Integration:** Azure DevOps is used to set up the build and release pipelines, ensuring the seamless automation of the CI/CD process.
4. **Custom Domain:** The project enables access to the web application via a custom domain, leveraging Azure DNS for domain management and resolution.

The project's purpose is to streamline web application deployment on Azure, making it more efficient, reliable, and error-free. It offers developers a solution that minimizes manual deployment efforts, accelerates the release cycle, and ensures consistent and hassle-free updates and new feature deployments. The integration of multiple Azure services and DevOps practices makes this project a powerful and comprehensive solution for Azure-based web application deployment.

# Flow Chart:



## Explanation:

This project involves creating a seamless deployment pipeline for a web application. It leverages Microsoft Azure services, including Azure Container Registry (ACR), Azure App Service, and Azure DevOps. Here's a breakdown of the key components and steps:

1. Azure Container Registry (ACR): An ACR repository is set up to store Docker containers of the web application. It will store the containers created by build pipeline.
2. GitHub Repository: The project code is stored in a GitHub repository, providing version control and collaboration capabilities.
3. Azure DevOps: Azure DevOps is utilized to automate the build and release processes.

4. Build Pipeline: A build pipeline is created in Azure DevOps, which compiles the project code from the GitHub repository, packages it into a Docker container, and stores the container in ACR.
5. Azure App Service: An Azure App Service is configured to host the web application using Docker containers. This service provides a scalable and cost-effective hosting solution, as per the requirement I will not be using auto scaling.
6. Deployment Center: The deployment center is accessed from the Azure portal it is the part of App Service from where I can manually deploy the container stored in ACR on the App Service.
7. Release Pipeline: A release pipeline in Azure DevOps automates the deployment of the web application. Whenever changes are committed to the master branch in GitHub, the build and release processes are triggered automatically.

By automating these processes, the project enhances efficiency, reduces human errors, and allows for easy scaling and continuous integration. The project serves as a model for streamlined web application deployment using Azure services.

## **Problem Statement:**

The project aims to address the challenge of manual and error-prone deployment processes for web applications on the Azure platform. Inefficient deployment practices often result in delays and issues in delivering updates and new features to end-users. This project identifies the problem of manual deployment and seeks to solve it through the implementation of a streamlined Continuous Integration and Continuous Deployment (CI/CD) pipeline. By automating the deployment process, the project mitigates deployment errors, enhances delivery speed, and promotes a more reliable and efficient web application deployment on Azure. This addresses the existing problem of manual and error-prone deployments and offers an opportunity to significantly improve the deployment workflow for web applications hosted on Azure.

## **Project Description:**

The core idea behind this project is to address the problem of manual and error-prone deployment processes for web applications on the Azure platform. We are leveraging Azure's robust feature set to create an automated Continuous Integration and Continuous Deployment (CI/CD) pipeline. This project targets developers and development teams looking to streamline their web application deployment on Azure.

The problem we are solving is the inefficiency of manual deployments, which often lead to delays, errors, and inconsistencies in the release of updates and new features. By implementing an automated CI/CD pipeline, we are mitigating this problem. Our project allows developers to push their code changes to a GitHub repository, and from there, Azure DevOps takes over. It builds a Docker container of the web application, stores it in an Azure Container Registry (ACR), and then deploys it to an Azure App Service. This automated pipeline ensures that the deployment process is error-free and rapid.

Our project addresses the clear need for efficient, reliable, and consistent web application deployment on Azure. By automating the process, it not only eliminates manual errors but also accelerates the release cycle. The purpose and basic functionality of our project are closely aligned with the problem statement, offering a systematic solution that enables developers to focus on coding while the CI/CD pipeline takes care of deployment intricacies.

## **core Azure services:**

1. Azure Container Registry (ACR): ACR is the foundation for storing Docker container images, enabling us to efficiently manage and distribute containerized applications. We use ACR to store and version control our application's container images.
2. Azure App Service: Azure App Service is the platform where we host and run our web application. It provides a managed environment for web app deployment, including auto-scaling, load balancing, and seamless integration with our Docker containers stored in ACR.

## **Additional Azure Services:**

1. Azure DevOps: While not a core Azure service, Azure DevOps plays a crucial role in this project. It is used for setting up the build and release pipelines, automating the CI/CD process, and connecting various components of the project together.
2. DNS (for Custom Domain): The project enables access to the web application via a custom domain, mapping a custom domain name to app service and using App Service Managed Certificate.
3. App Service Managed Certificate: Providing SSL certificates for secure communication with the custom domain..

# Complete Step By Step Process:

The screenshot shows the 'Create container registry' wizard on the Azure portal. The 'Project details' section is filled out with a Subscription of 'Azure for Students' and a Resource group of '(New) FRTProjectCICD'. The 'Instance details' section shows a Registry name of 'ContainerRegistryBlogWeb' and a Location of 'Central India'. A note indicates that availability zones are disabled. The 'Pricing plan' is set to 'Standard'. At the bottom, there are 'Review + create' and 'Next: Networking >' buttons.

## 1. Create Container Repository

The screenshot shows the 'Create container registry' wizard on the Azure portal, with the validation passed. The summary table includes:

Registry name	ContainerRegistryBlogWeb
Subscription	Azure for Students
Resource Group	FRTProjectCICD
Location	Central India
Availability zones	Disabled
Pricing plan	Standard

The 'Networking' section shows 'Public network access' set to 'Yes'. The 'Encryption' section shows 'Customer-Managed Key' set to 'Disabled' and 'Identity' set to 'None'. At the bottom, there is a 'Create' button and a 'Download a template for automation' link.

A Microsoft.Conta x Learning | Future x Projects - Home x Azure Container x G container regist x Registry service x SKstudies/Blogs x +

portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2Fb858338e-efa7-486d-aebd-c1573aab0e40%2Fresource... D : ;

E-CLASS Announcements Getting Started wit... DataCamp's Fast Tr... Best Free Online Co... HTML, CSS, and Java... ESSO-INCOIS-India... Coderbyte | The #1... dsa miss Mahabhum... >

Microsoft Azure Search resources, services, and docs (G+) diveshkkolhe@gmail.com DEFAULT DIRECTORY (DIVESHK)

Home > Microsoft.ContainerRegistry | Overview x ...

Deployment

Search < Delete Cancel Redeploy Download Refresh

Overview Inputs Outputs Template

Your deployment is complete

Deployment name : Microsoft.ContainerRegistry  
Subscription : Azure for Students  
Resource group : FRTProjectCICD

Start time : 10/25/2023, 1:22:23 PM  
Correlation ID : 651176d8-a014-4abc-9ee5-3dbc60018c17

Deployment details

Resource	Type	Status	Operation details
ContainerRegistryBlogWeb	Container registry	OK	<a href="#">Operation details</a>

Next steps

[Go to resource](#)

Give feedback  
[Tell us about your experience with deployment](#)

Cost management  
Get notified to stay within your budget and prevent unexpected charges on your bill.  
[Set up cost alerts >](#)

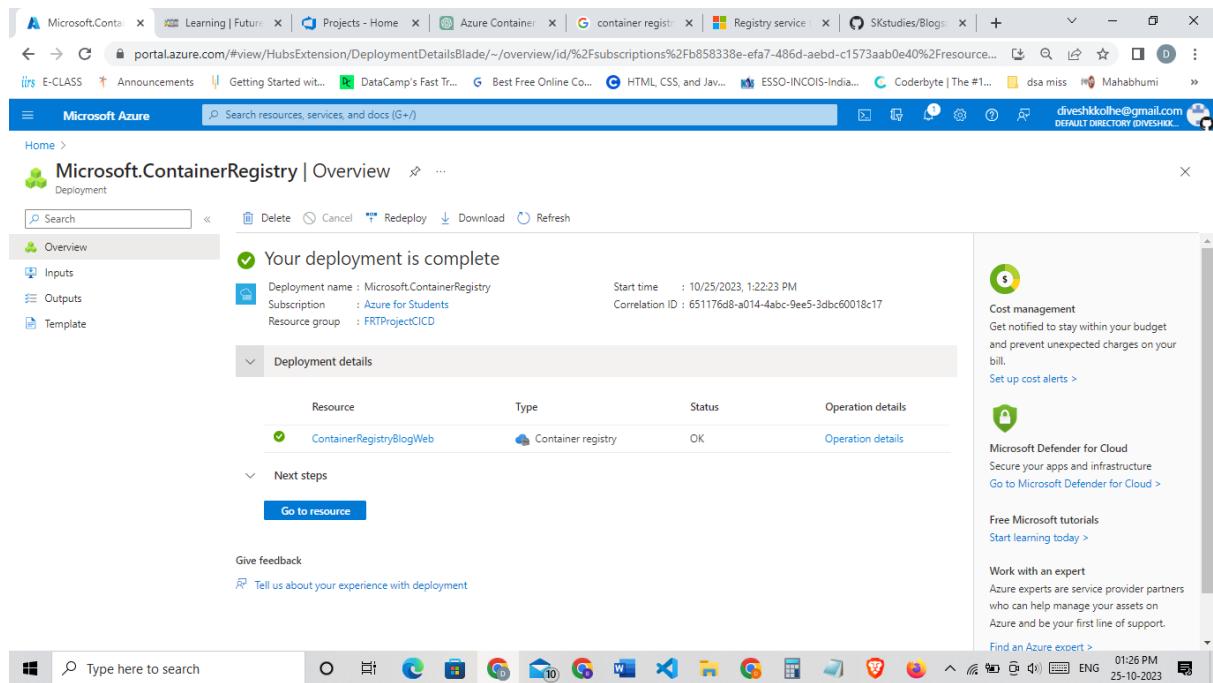
Microsoft Defender for Cloud  
Secure your apps and infrastructure  
[Go to Microsoft Defender for Cloud >](#)

Free Microsoft tutorials  
[Start learning today >](#)

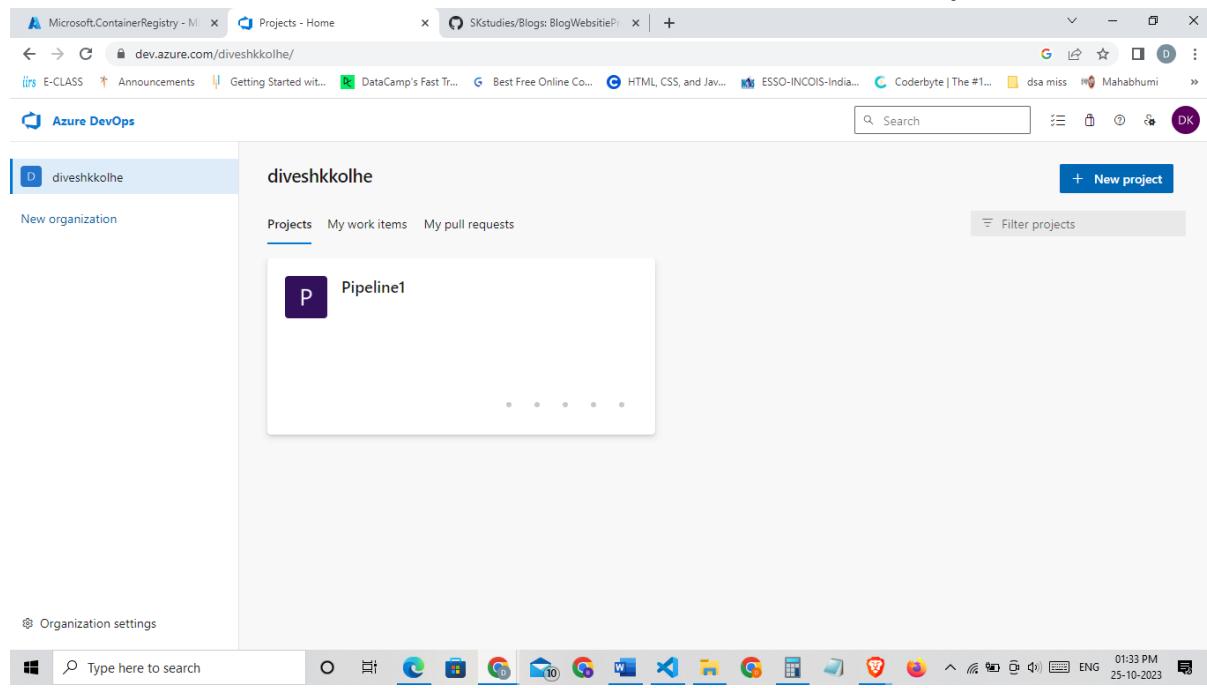
Work with an expert  
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.  
[Find an Azure expert >](#)

Type here to search

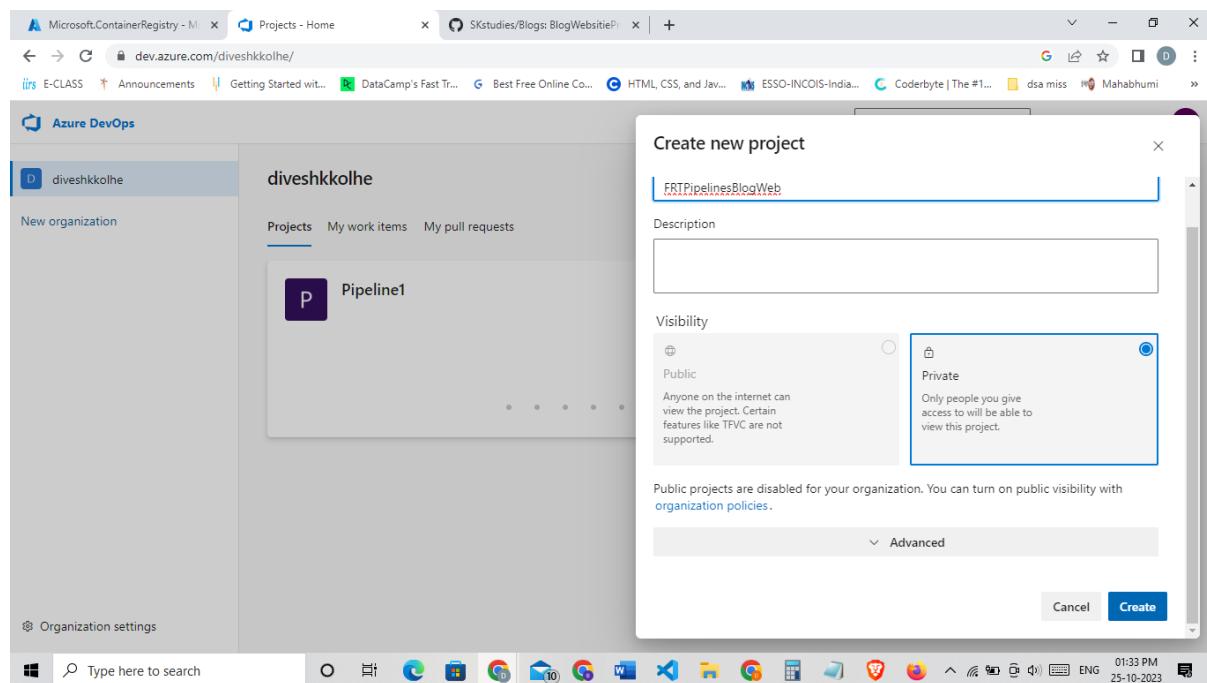
01:26 PM 25-10-2023 ENG

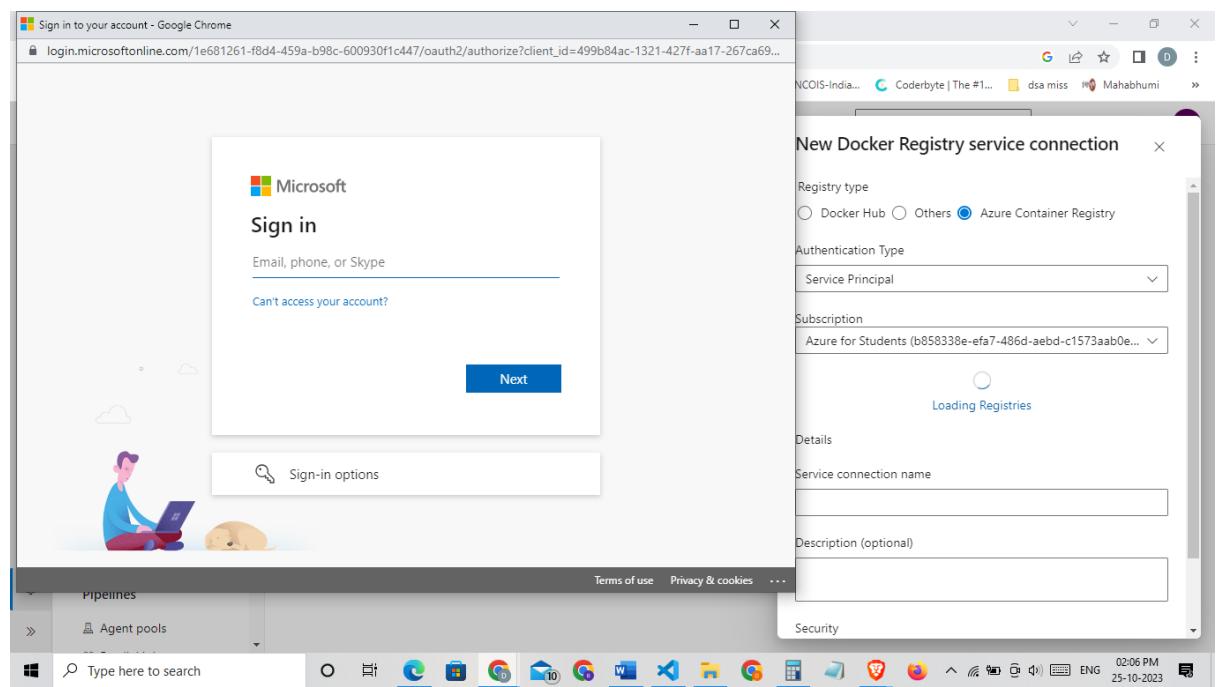
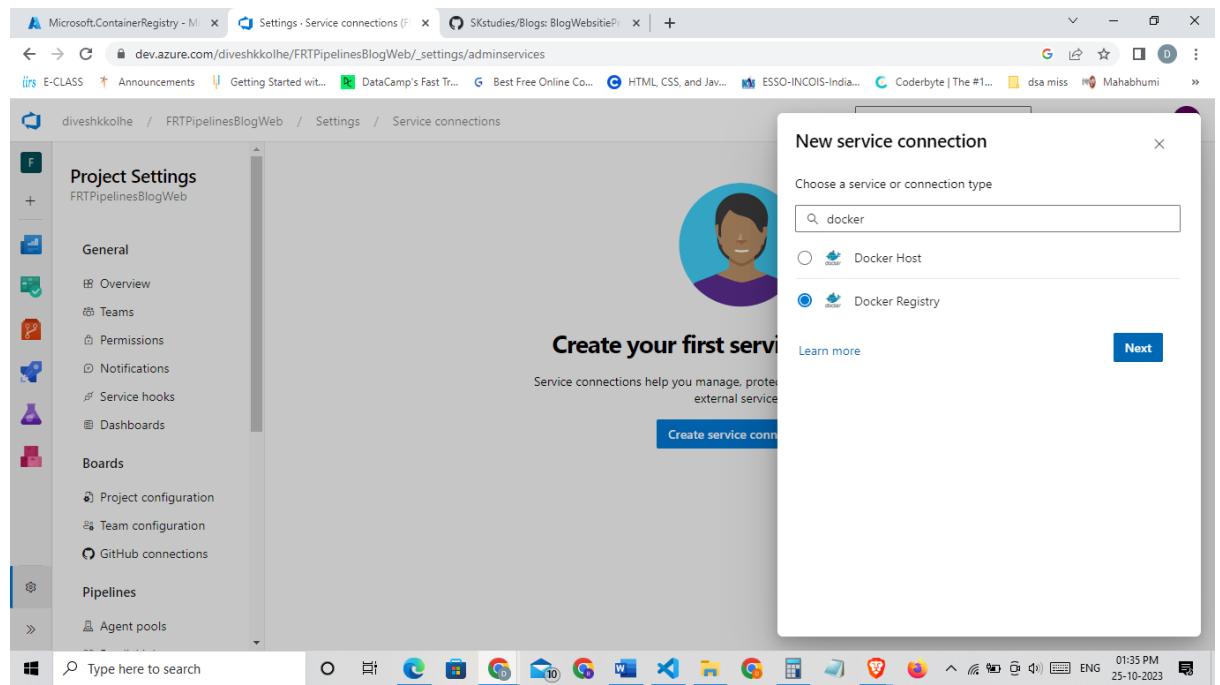


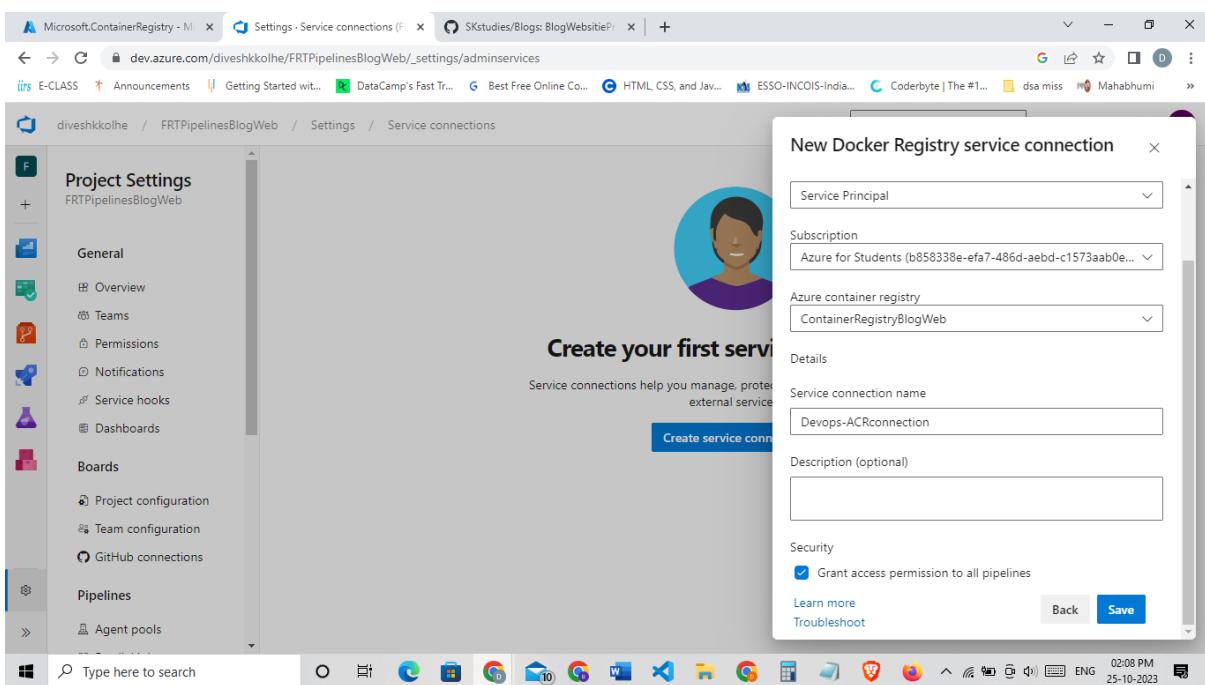
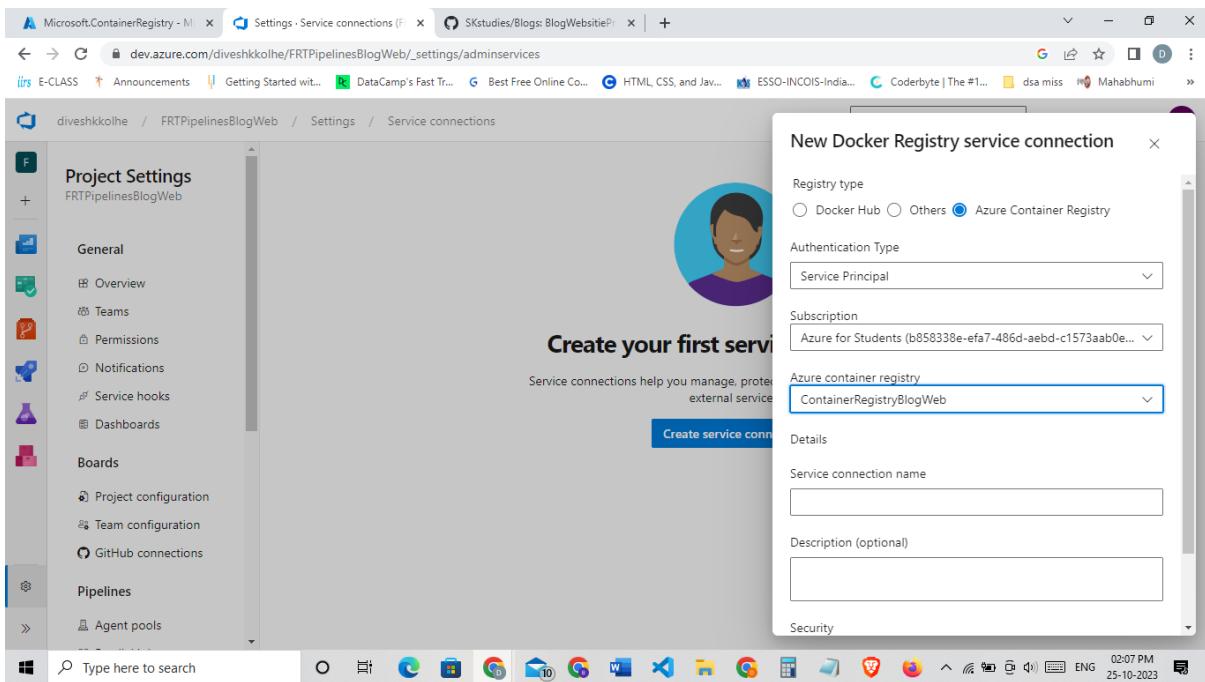
## 2. Create Service Connections From Devops To

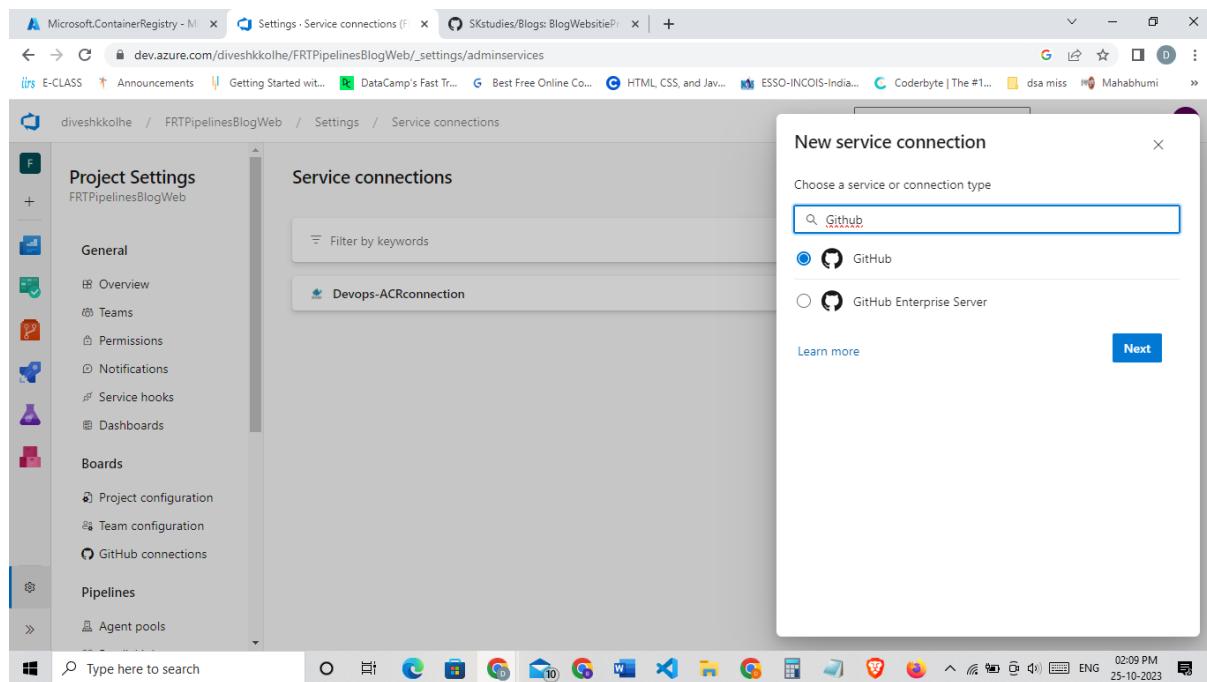
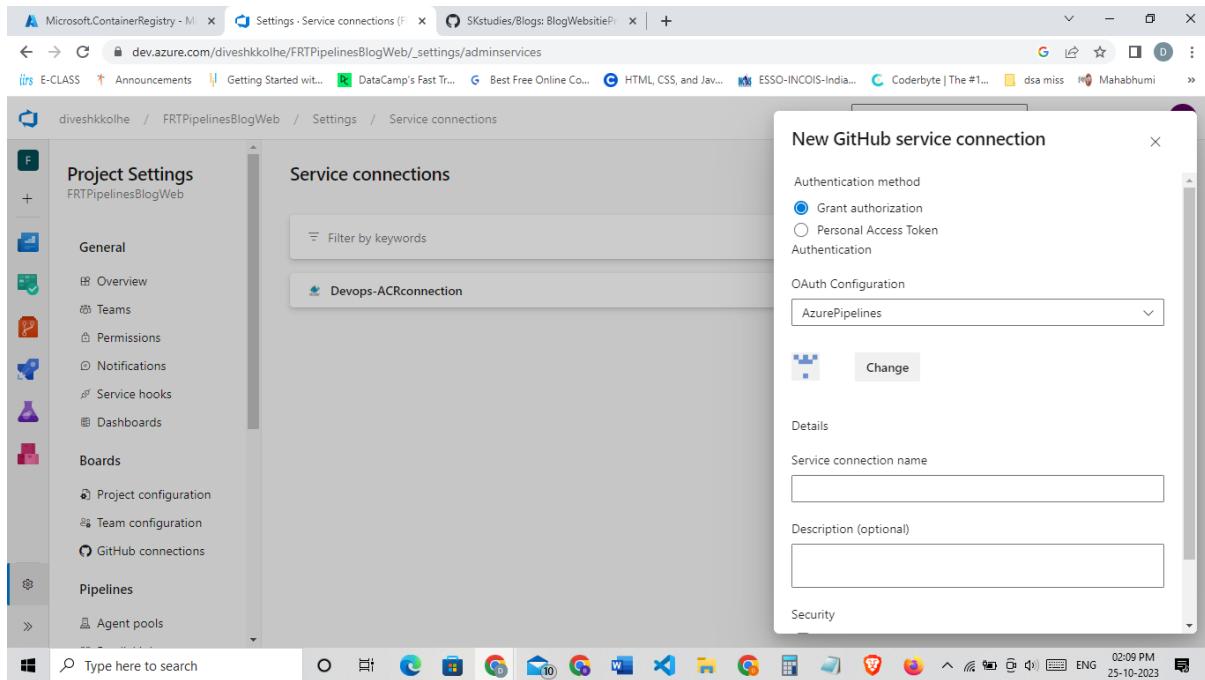


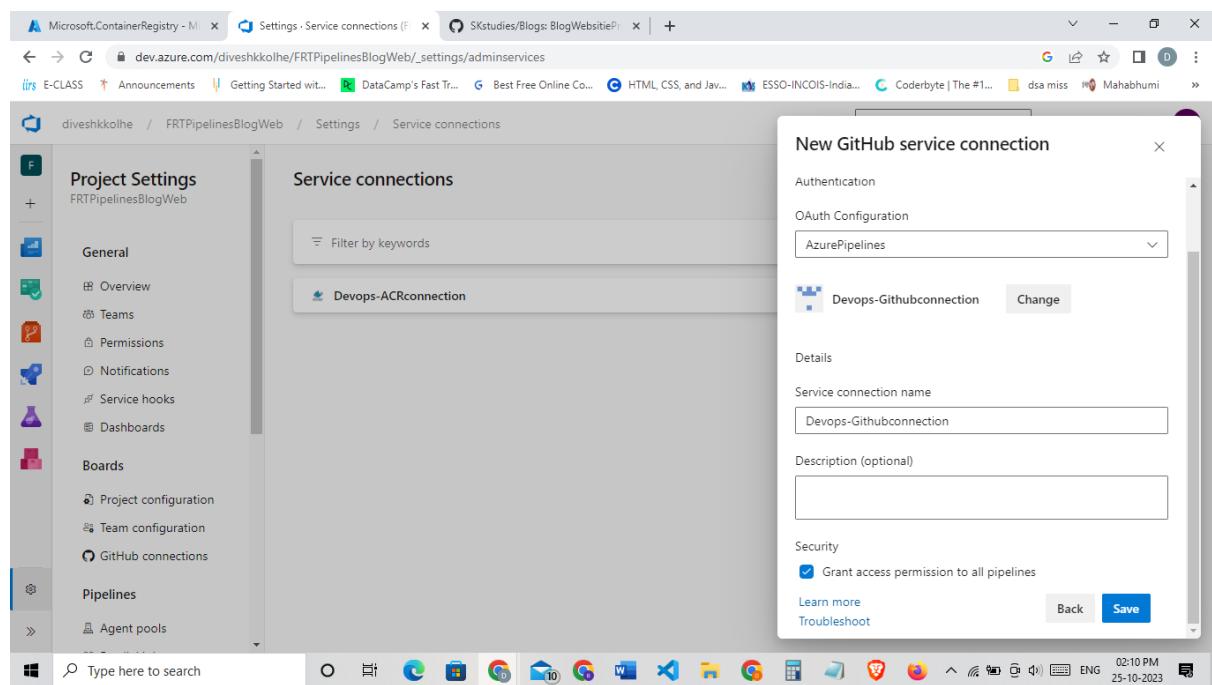
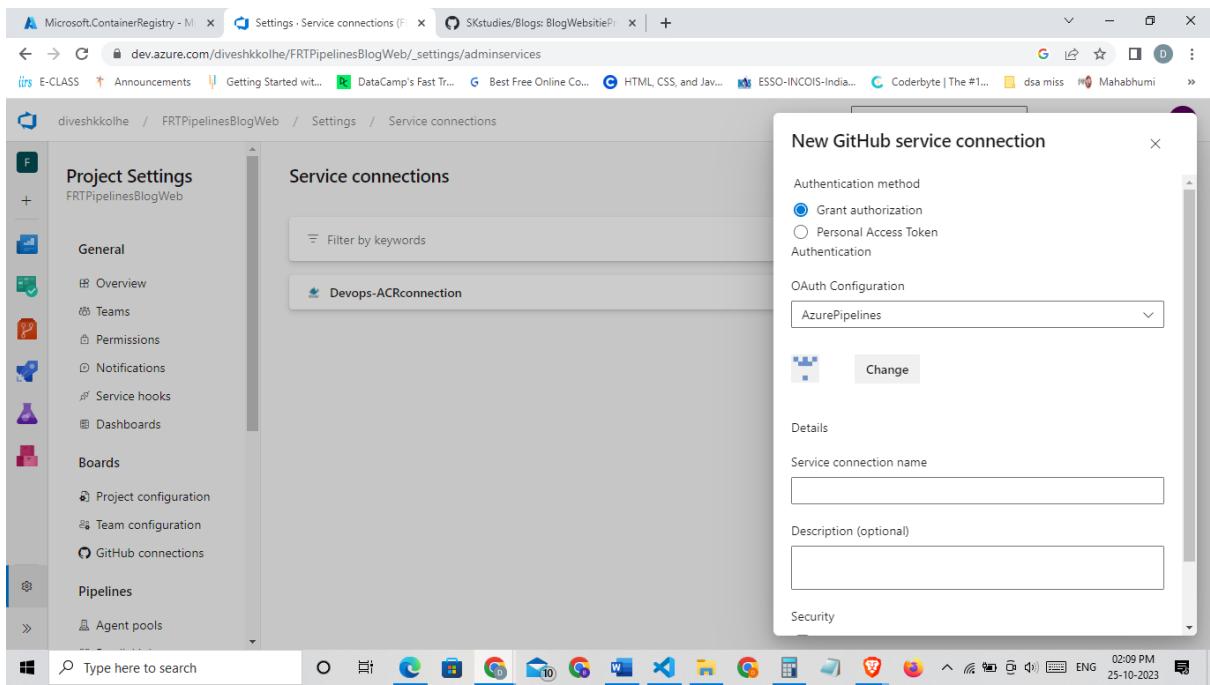
### Github and ACR



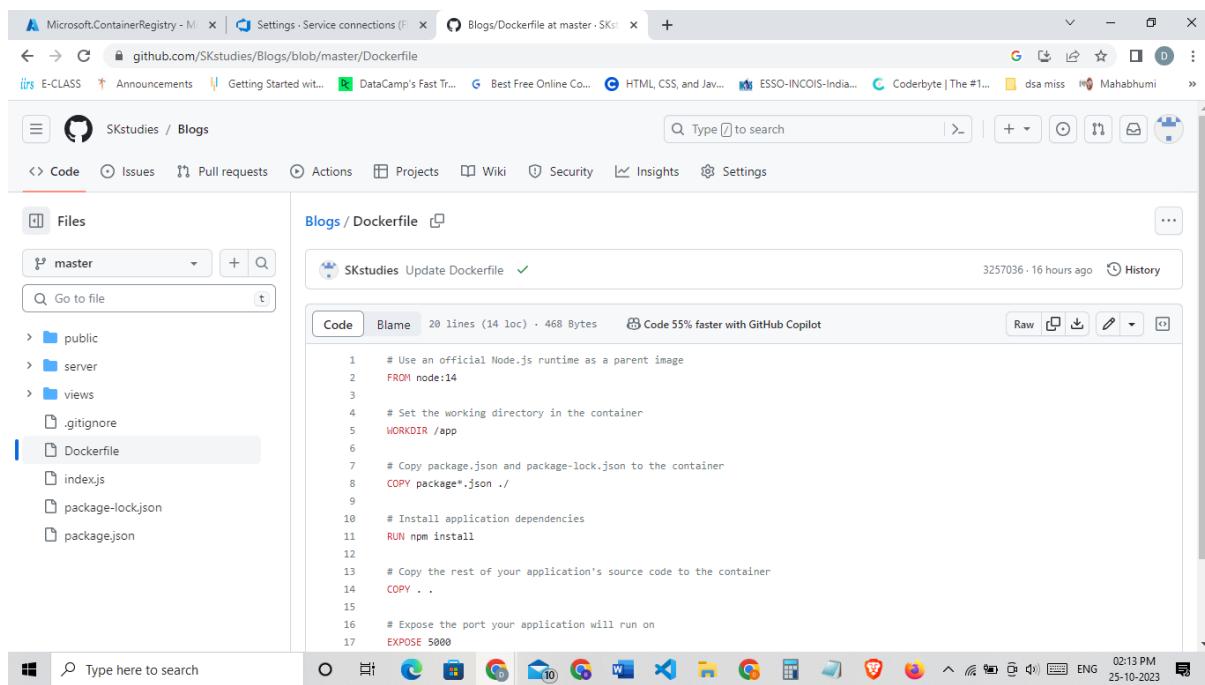
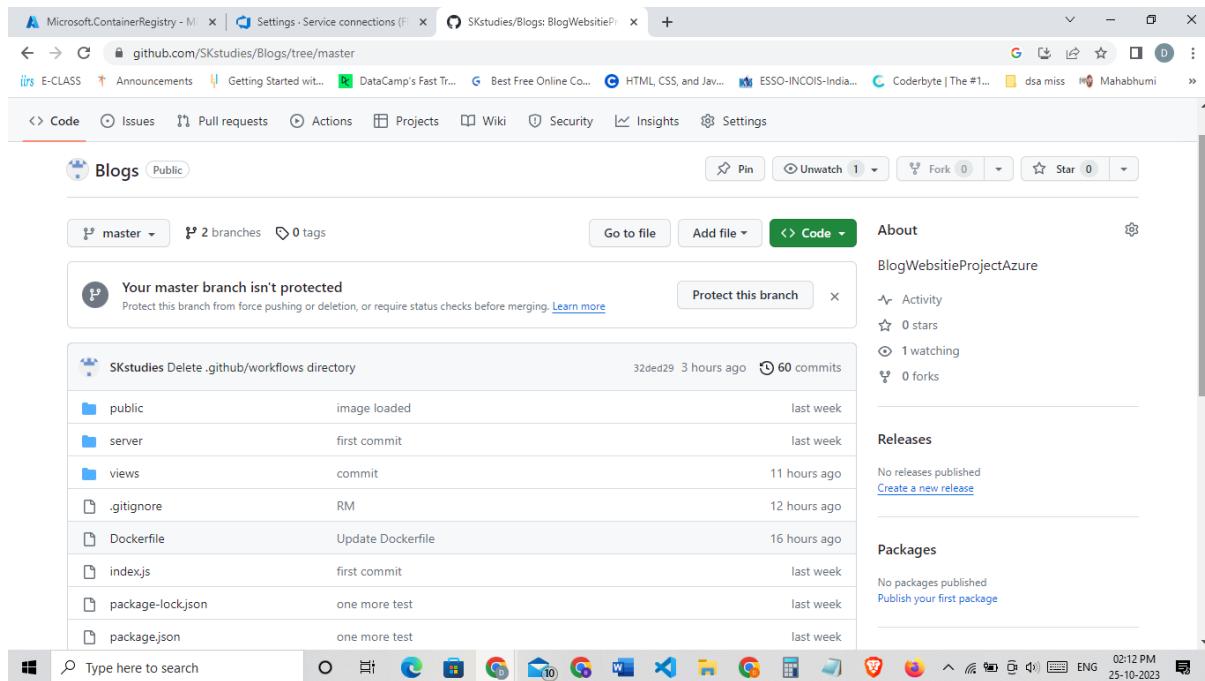








### 3. Push The Code To Github Repo Which Should Contain the Required Docker File.



## 4. Create Build Pipeline.

The screenshot shows the Azure DevOps Pipelines interface. The left sidebar has a 'Pipelines' section with several options: Overview, Boards, Repos, Pipelines (selected), Environments, Releases, Library, Task groups, Deployment groups, and Test Plans. The main area features a cartoon illustration of a robot, a person, and a cat. Below the illustration, the text 'Create your first Pipeline' is displayed, followed by the sub-instruction 'Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes.' A prominent blue 'Create Pipeline' button is centered at the bottom of this section. The browser's address bar shows the URL: dev.azure.com/diveshkkolhe/FRTPIPipelinesBlogWeb/\_build. The taskbar at the bottom includes icons for File Explorer, Task View, Edge, Google Chrome, Mail, Word, Excel, Power BI, and others, along with system status indicators like battery level and date/time (02:14 PM 25-10-2023).

This screenshot is identical to the one above, showing the 'Create your first Pipeline' wizard in the Azure DevOps Pipelines interface. The left sidebar, main content area with the pipeline creation illustration and instructions, and the browser's address bar and taskbar at the bottom are all the same.

Screenshot of the Azure DevOps Pipelines interface showing the 'New pipeline' creation process.

The browser tabs are:

- Microsoft.ContainerRegistry - M
- New pipeline - Pipelines
- SKstudies/Blogs: BlogWebsiteP... (active tab)

The address bar shows: dev.azure.com/diveshkkolhe/FRTPIPelinesBlogWeb/\_apps/hub/ms.vss-build-web.ci-designer-hub

The page title is: Azure DevOps | FRTPIPelinesBlogWeb / Pipelines

The left sidebar menu is:

- Overview
- Boards
- Repos
- Pipelines** (selected)
- Pipelines
- Environments
- Releases
- Library
- Task groups
- Deployment groups
- Test Plans
- Project settings

The main content area displays the 'Where is your code?' section with options for connecting to:

- Azure Repos Git (YAML)
- Bitbucket Cloud (YAML)
- GitHub** (YAML)
- GitHub Enterprise Server (YAML)

The status bar at the bottom shows: 02:14 PM 25-10-2023

Screenshot of the Azure DevOps Pipelines interface showing the 'Select a repository' step of the pipeline creation process.

The browser tabs are:

- Microsoft.ContainerRegistry - M
- New pipeline - Pipelines
- New pipeline - Pipelines

The address bar shows: dev.azure.com/diveshkkolhe/FRTPIPelinesBlogWeb/\_apps/hub/ms.vss-build-web.ci-designer-hub?triggers=ContinuousIntegration%2CPullRequest&telemetrySessionId=...

The page title is: Azure DevOps | FRTPIPelinesBlogWeb / Pipelines

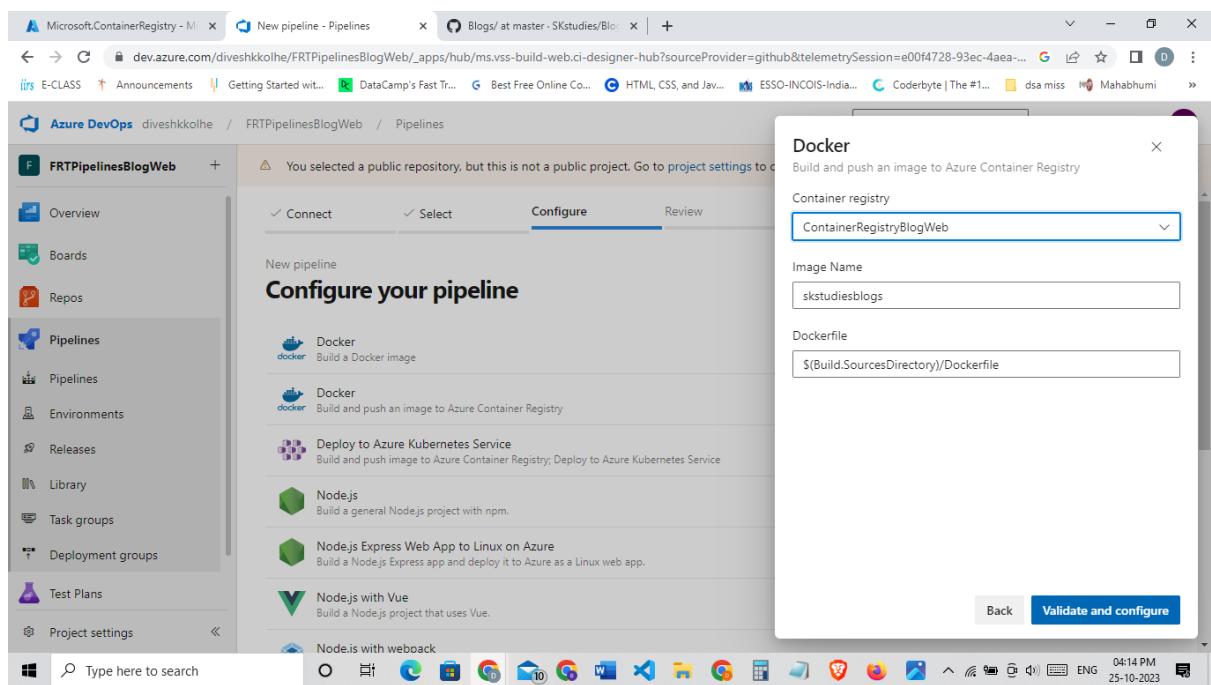
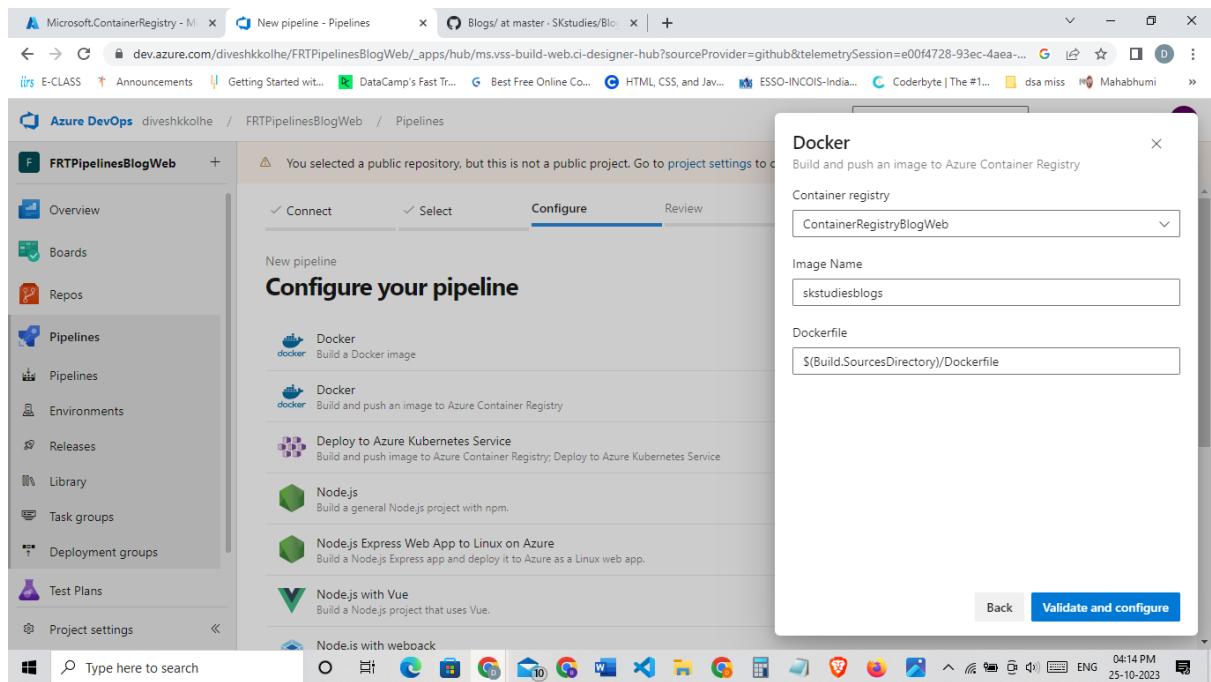
The left sidebar menu is identical to the previous screenshot.

The main content area displays the 'Select a repository' section with a 'Filter by keywords' input field and a list of repositories:

- SKstudies/Blogs (Oct 15)
- SKstudies/ap** (Oct 6) (highlighted)

A note below the list states: Showing the most recently used repositories where you are a collaborator. If you can't find a repository, make sure you provide access. You may also select a specific connection.

The status bar at the bottom shows: 02:28 PM 25-10-2023



The screenshot shows the Azure DevOps Pipelines interface for a project named "FRTpipelinesBlogWeb". The left sidebar is visible with options like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Project settings. The main area is titled "Review your pipeline YAML" and displays the following YAML code:

```
1 # Docker
2 # Build and push an image to Azure Container Registry
3 # https://docs.microsoft.com/azure/devops/pipelines/languages/docker
4
5 trigger:
6 - master
7
8 resources:
9 - repo: self
10
11 variables:
12 # Container registry service connection established during pipeline creation
13 dockerRegistryServiceConnection: 'e84362bd-f59e-4718-a3ab-87aca529528c'
14 imageRepository: 'skstudiesblogs'
15 containerRegistry: 'containerregistryblogweb.azurecr.io'
16 dockerfilePath: '${Build.SourcesDirectory}/Dockerfile'
```

Below the code, there are buttons for "Variables", "Save and run", and a dropdown menu. The status bar at the bottom right shows "04:15 PM 25-10-2023".

The screenshot shows the same Azure DevOps Pipelines interface as above, but with a modal dialog box titled "Save and run" overlaid. The dialog contains the following fields:

- Commit message: "Set up CI with Azure Pipelines"
- Optional extended description: "Add an optional description..." (empty)
- Commit options:
  - Commit directly to the master branch
  - Create a new branch for this commit
- A "Save and run" button at the bottom right of the dialog.

The rest of the interface and status bar are identical to the first screenshot.

The screenshot shows the Azure DevOps Pipelines interface for a project named "FRTpipelinesBlogWeb". On the left, the navigation bar has "Pipelines" selected. The main area displays a "Review your pipeline YAML" section with the following YAML code:

```
1  # Docker
2  # Build and push an image to Azure Container Registry
3  # https://docs.microsoft.com/azure/devops/pipelines/languages/docker
4
5  trigger:
6  - master
7
8  resources:
9  - repo: self
10
11 variables:
12   # Container registry service connection established during pipeline
13   dockerRegistryServiceConnection: 'e84362bd-f59e-4718-a3ab-87aca529'
14   imageRepository: 'skstudieblogs'
15   containerRegistry: 'containerregistryblogweb.azurecr.io'
16   dockerfilePath: '${Build.SourcesDirectory}/Dockerfile'
```

To the right, a "Save and run" dialog box is open, containing fields for "Commit message" (Set up CI with Azure Pipelines), "Optional extended description" (Add an optional description...), and two radio buttons for "Commit directly to the master branch" (selected) and "Create a new branch for this commit". A "Save and run" button is at the bottom right of the dialog.

The screenshot shows the Azure DevOps Pipelines interface for the same project. The left sidebar shows "Pipelines" selected. The main area displays the "Jobs in run #20231025.1" for the "Build" stage. The log output for the "Build" stage is shown below:

```
1 Pool: Azure_Pipelines
2 Image: ubuntu-latest
3 Agent: Hosted Agent
4 Started: Today at 4:15 PM
5 Duration: 2m 19s
6
7 ▶ Job preparation parameters
```

A "View raw log" button is located at the top right of the log viewer. The status of the build job is marked with a green checkmark.

## 5. Docker Container Created In The Repository.

The screenshot shows the Microsoft Azure portal interface. The user is navigating through 'Container registries' and has selected 'ContainerRegistryBlogWeb | Repositories'. On the left sidebar, under 'Services', 'Repositories' is highlighted. The main pane displays a repository named 'skstudieblogs'. The 'Essentials' section shows a single tag ('1'), a manifest count of 1, and a last updated date of 10/25/2023, 4:17 PM GMT+5:30. Below this, a table lists 24 tags, each with a digest and a last modified timestamp of 10/25/2023, 4:17 PM GMT+5:30. The top right corner of the main pane shows the user's email (diveshkkolle@gmail.com) and the date (26-10-2023). The bottom right corner of the screen shows the system date and time (09:05 AM 26-10-2023).

## 6. Allow Access.

The screenshot shows the Microsoft Azure portal interface. The user is navigating through 'Container registries' and has selected 'ContainerRegistryBlogWeb | Access keys'. On the left sidebar, under 'Services', 'Access keys' is highlighted. The main pane displays the access key settings for 'ContainerRegistryBlogWeb'. It shows the 'Admin user activated' status with a green checkmark and the message "'ContainerRegistryBlogWeb' is now an admin". The 'Overview' section shows the registry name as 'ContainerRegistryBlogWeb' and the login server as 'containerregistryblogweb.azurecr.io'. The 'Access control (IAM)' section shows an 'Admin user' checkbox checked, and the 'Username' field contains 'ContainerRegistryBlogWeb'. Two password fields are present: 'password' with value 'I7ROOQGik2oBOG1s1XMvrqjLSKwOARxFYlt2ewC3Qp+AC...' and 'password2' with value 'V3LRGfFBnps3EFt7z4H0thaZqj2198We+Ptlu+ACRC+7...'. The bottom right corner of the main pane shows the user's email (diveshkkolle@gmail.com) and the date (30-10-2023). The bottom right corner of the screen shows the system date and time (05:07 PM 30-10-2023).

## 7. Create Web App.

App Services - Microsoft Azure | Pipelines - Run 20231025.1 logs | Blogs/ at master · SKstudies/Blo... | +

portal.azure.com/#view/HubsExtension/BrowseResource/resourceType/Microsoft.Web%2Fsites

E-CLASS Announcements Getting Started with... DataCamp's Fast Tr... Best Free Online Co... HTML, CSS, and Jav... ESSO-INCOIS-India... Coderbyte | The #1... dsa miss Mahabumi

Microsoft Azure Search resources, services, and docs (G+ /)

Home > App Services

Default Directory

+ Create Manage Deleted Apps Manage view Refresh Export to CSV Open query Assign tags Start Restart Stop Delete

Filter for any field... Subscription equals all Resource group equals all Location equals all Add filter

Showing 0 to 0 of 0 records.

Name ↑ Status ↑ Location ↑ Pricing Tier ↑ App Service Plan ↑ Subscription ↑ App Type ↑

No grouping List view

No app services to display

Create, build, deploy, and manage powerful web, mobile, and API apps for employees or customers using a single back-end. Build standards-based web apps and APIs using .NET, Java, Node.js, PHP, and Python.

Learn more about App Service

Give feedback

Create Web App - Microsoft Azure | Pipelines - Run 20231025.1 logs | Blogs/ at master · SKstudies/Blo... | +

portal.azure.com/#create/Microsoft.WebSite

E-CLASS Announcements Getting Started with... DataCamp's Fast Tr... Best Free Online Co... HTML, CSS, and Jav... ESSO-INCOIS-India... Coderbyte | The #1... dsa miss Mahabumi

Microsoft Azure Search resources, services, and docs (G+ /)

Home > App Services > Create Web App

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*  Resource Group \*  Create new

Instance Details

Need a database? Try the new Web + Database experience.

Name \*  azurewebsites.net

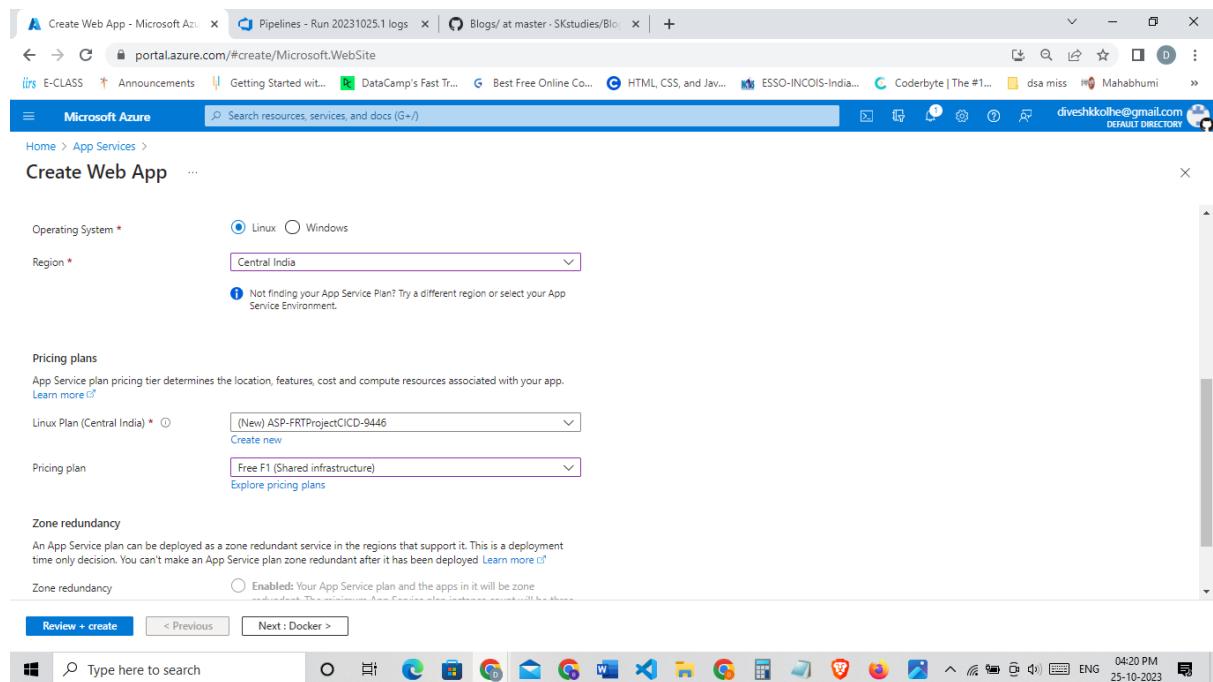
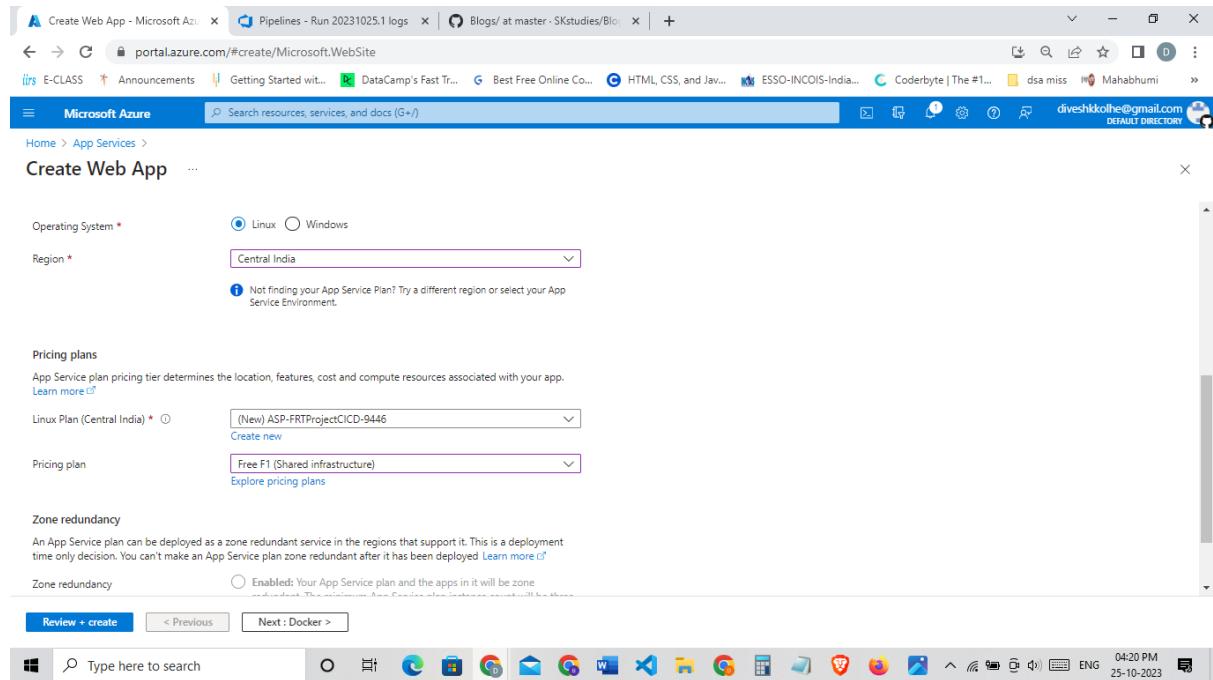
Publish \*  Code  Docker Container  Static Web App

Operating System \*  Linux  Windows

Region \*

Not finding your App Service Plan? Try a different region or select your App Service Environment.

Review + create < Previous Next : Docker >



Microsoft Azure

Search resources, services, and docs (G-)

Home > App Services >

## Create Web App

Name: BlogAppFRT

Publish: Docker Container

Image/Tag: mcr.microsoft.com/appsvc/staticsite:latest

Server URL: https://mcr.microsoft.com

**App Service Plan (New)**

Name	ASP-FRTProjectCI/CD-9446
Operating System	Linux
Region	Central India
SKU	Free
ACU	Shared infrastructure
Memory	1 GB memory

**Monitoring**

Application Insights: Not enabled

**Deployment**

Continuous deployment: Not enabled / Set up after app creation

**Actions**

Create < Previous Next > Download a template for automation

## 8. Modify Application Settings and Deploy.

Add/Edit application setting

Name: MONGODB\_URI

Value: MONGODB\_URI mongodb+srv://braveunknown123:hXtmmJaYOMbiFT60@cluster0.nddw6pl.mongodb.net/blog

Deployment slot setting

ok Cancel

Add/Edit application setting

Name: JWT\_SECRET

Value: MySecretBlog

Deployment slot setting

ok Cancel

Screenshot of the Microsoft Azure portal showing the configuration settings for the "BlogAppFRT" web app. The left sidebar shows the "Configuration" section selected. The main pane displays application settings, including environment variables like DOCKER\_ENABLE\_CI, DOCKER\_REGISTRY\_SERVER\_PASSWORD, DOCKER\_REGISTRY\_SERVER\_URL, DOCKER\_REGISTRY\_SERVER\_USERNAME, JWT\_SECRET, MONGODB\_URI, and WEBSITES\_ENABLE\_APP\_SERVICE\_STORAGE.

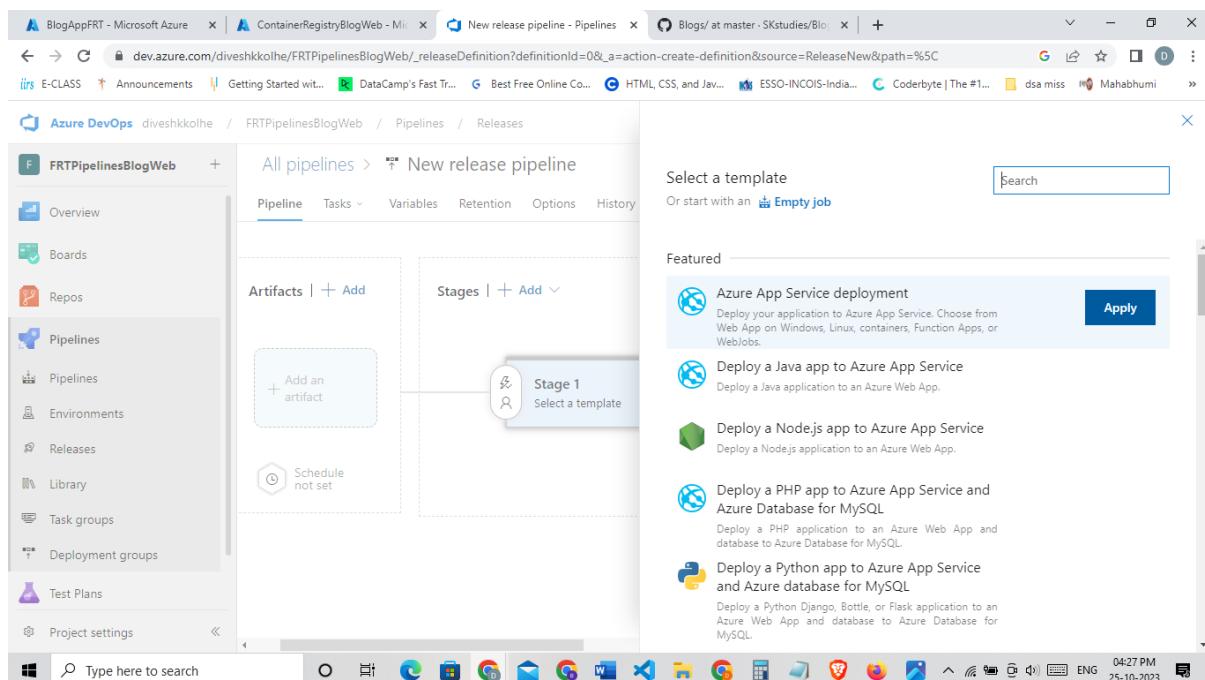
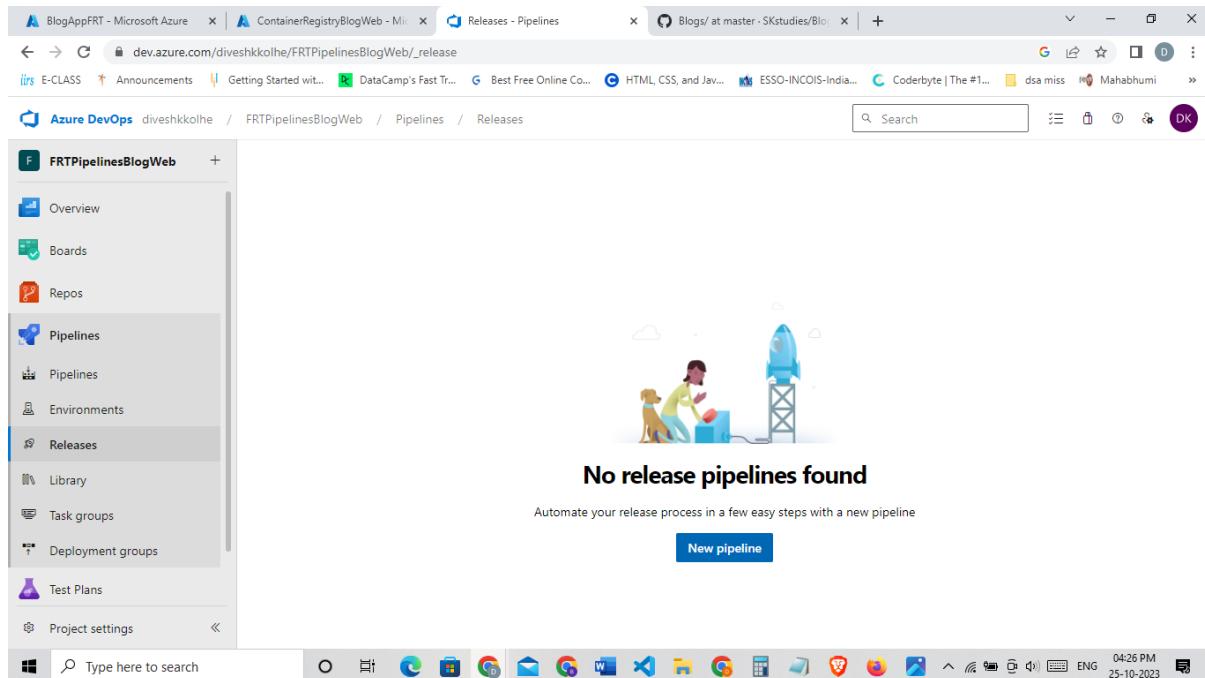
Name	Value	Source	Deployment slot setting	Delete	Edit
DOCKER_ENABLE_CI	Hidden value. Click to show value	App Service			
DOCKER_REGISTRY_SERVER_PASSWORD	Hidden value. Click to show value	App Service			
DOCKER_REGISTRY_SERVER_URL	Hidden value. Click to show value	App Service			
DOCKER_REGISTRY_SERVER_USERNAME	Hidden value. Click to show value	App Service			
JWT_SECRET	Hidden value. Click to show value	App Service			
MONGODB_URI	Hidden value. Click to show value	App Service			
WEBSITES_ENABLE_APP_SERVICE_STORAGE	Hidden value. Click to show value	App Service			

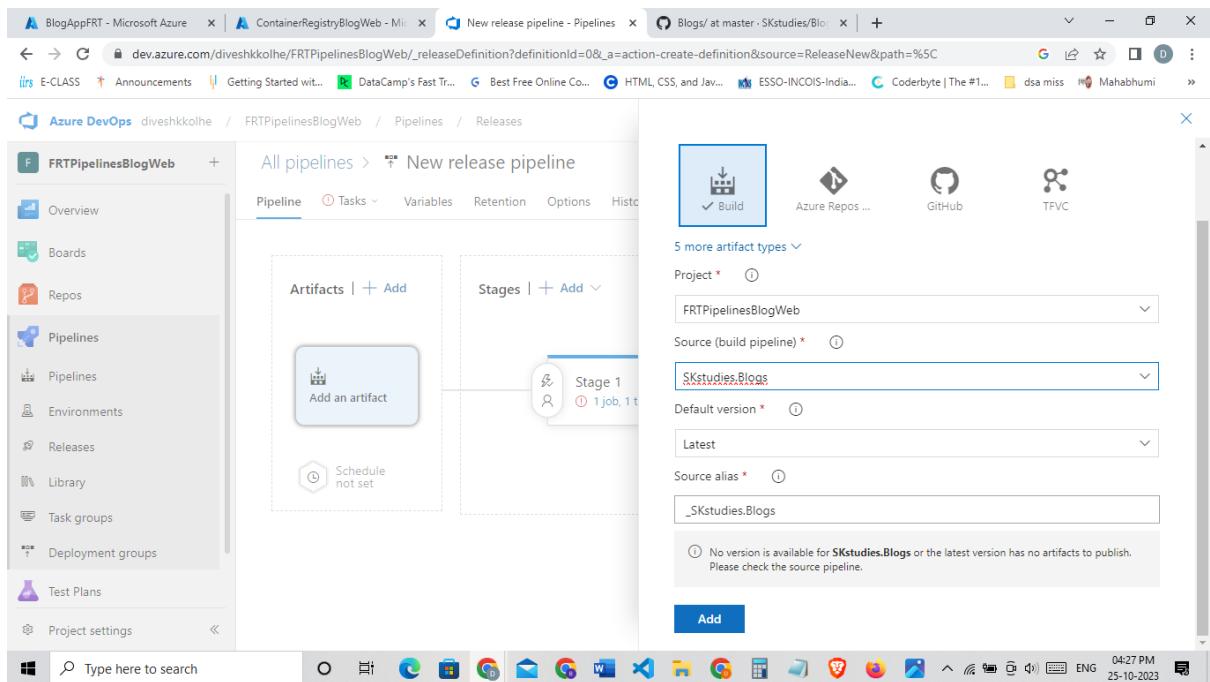
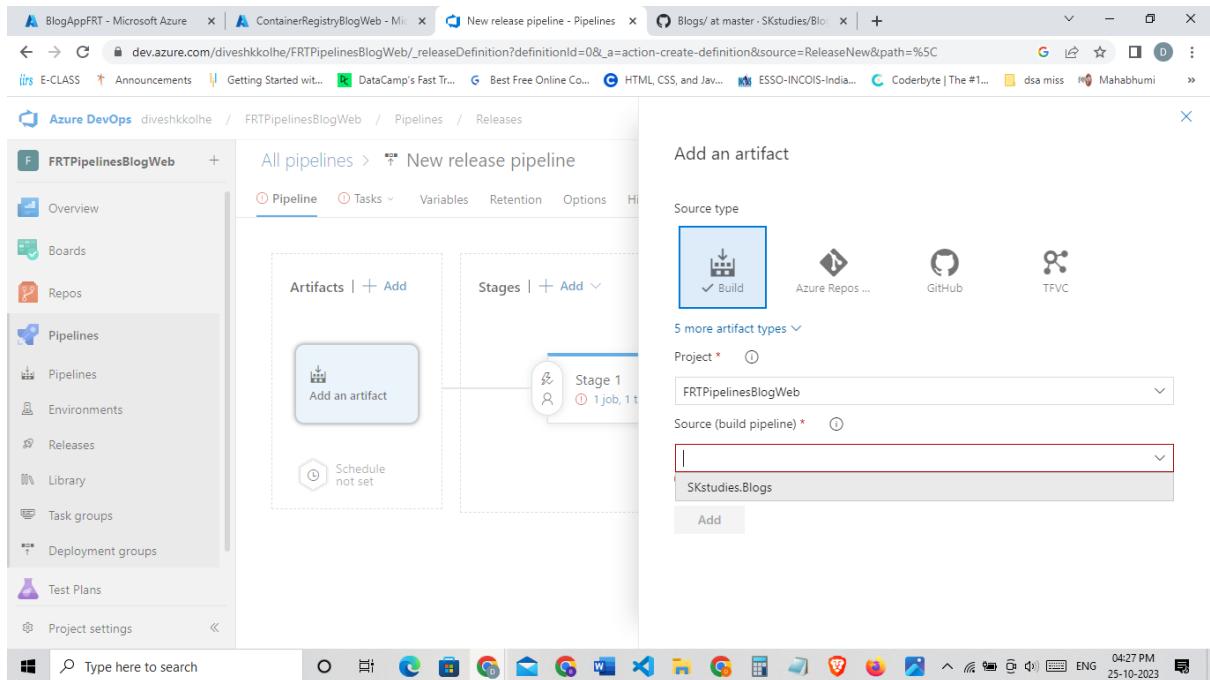
Screenshot of the Microsoft Azure portal showing the deployment center settings for the "BlogAppFRT" web app. The left sidebar shows the "Deployment Center" section selected. The main pane displays deployment settings, including the source (Container Registry), container type (Single Container), registry source (Azure Container Registry), subscription ID (Azure for Students), authentication method (Admin Credentials), and registry name (ContainerRegistryBlogWeb).

The screenshot shows the Azure Deployment Center settings for the 'BlogAppFRT' web app. The left sidebar shows navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Deployment slots, Deployment Center, Configuration, Authentication, Application Insights, Identity, and Backups. The main pane displays the 'Settings' tab under 'Deployment Center'. It includes sections for 'Source\*', 'Registry settings', and 'Deployment'. Under 'Source\*', the 'Container Registry' option is selected. In 'Registry settings', 'Container type' is set to 'Single Container', 'Registry source' is 'Azure Container Registry', 'Subscription ID' is 'Azure for Students', and 'Authentication' is 'Admin Credentials' (selected). The 'Registry\*' dropdown is set to 'ContainerRegistryBlogWeb'. The deployment section shows 'Image\*' as 'skstudieblogs', 'Tag\*' as '24', and 'Continuous deployment' turned 'On'. The status bar at the bottom indicates '04:23 PM 25-10-2023'.

This screenshot shows the same Azure Deployment Center settings page for 'BlogAppFRT', but with different values in the deployment fields. The 'Image\*' dropdown now contains 'skstudiesblogs', and the 'Tag\*' dropdown contains '24'. All other settings remain the same as in the previous screenshot. The status bar at the bottom indicates '04:23 PM 25-10-2023'.

## 9. Create Release Pipeline.





The screenshot shows the Azure DevOps Pipelines interface for a project named "FRTpipelinesBlogWeb". The left sidebar is visible with options like Overview, Boards, Repos, Pipelines (selected), Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Project settings. The main area displays a "New release pipeline" configuration. It includes sections for "Artifacts" (containing an artifact named "\_SKstudies.Blogs") and "Stages" (containing a single stage labeled "Stage 1"). On the right, there are two trigger configurations: "Continuous deployment trigger" (disabled) and "Pull request trigger" (disabled). A tooltip for the Continuous deployment trigger explains that enabling it will create a new release every time a new build is available. The bottom status bar shows the date and time as 25-10-2023 04:28 PM.

This screenshot shows the same Azure DevOps Pipelines interface as the first one, but with changes made to the triggers. The "Continuous deployment trigger" is now enabled, with a tooltip stating "Creates a release every time a new build is available." The "Pull request trigger" remains disabled. The rest of the pipeline configuration (Artifacts, Stages, and other triggers) appears identical to the first screenshot. The bottom status bar shows the date and time as 25-10-2023 04:28 PM.

Screenshot of the Azure DevOps Pipelines interface showing a new release pipeline.

The pipeline structure is as follows:

```
graph LR; Artifacts["Artifacts | + Add"] --> Stage1["Stage 1 | 1 job, 1 task"]; Stage1 --> ViewTasks["View stage tasks"]
```

**Artifacts:** \_SKstudies.Blogs

**Stages:** Stage 1 (1 job, 1 task)

**Trigger:** Continuous deployment trigger

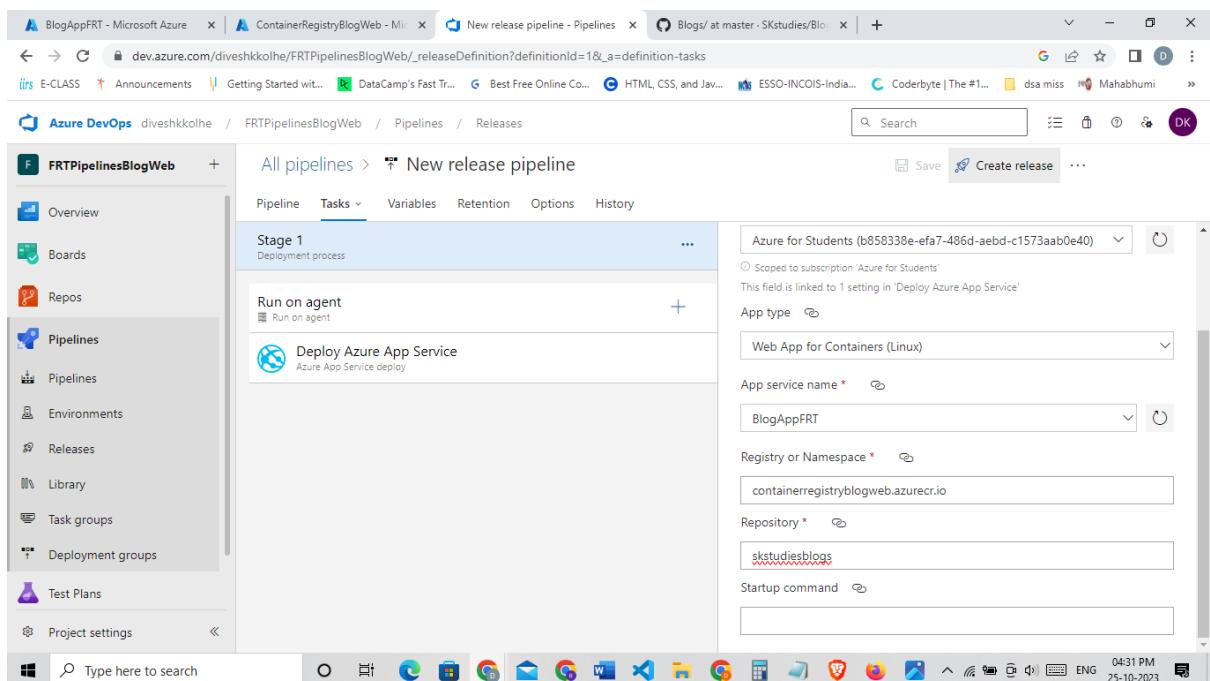
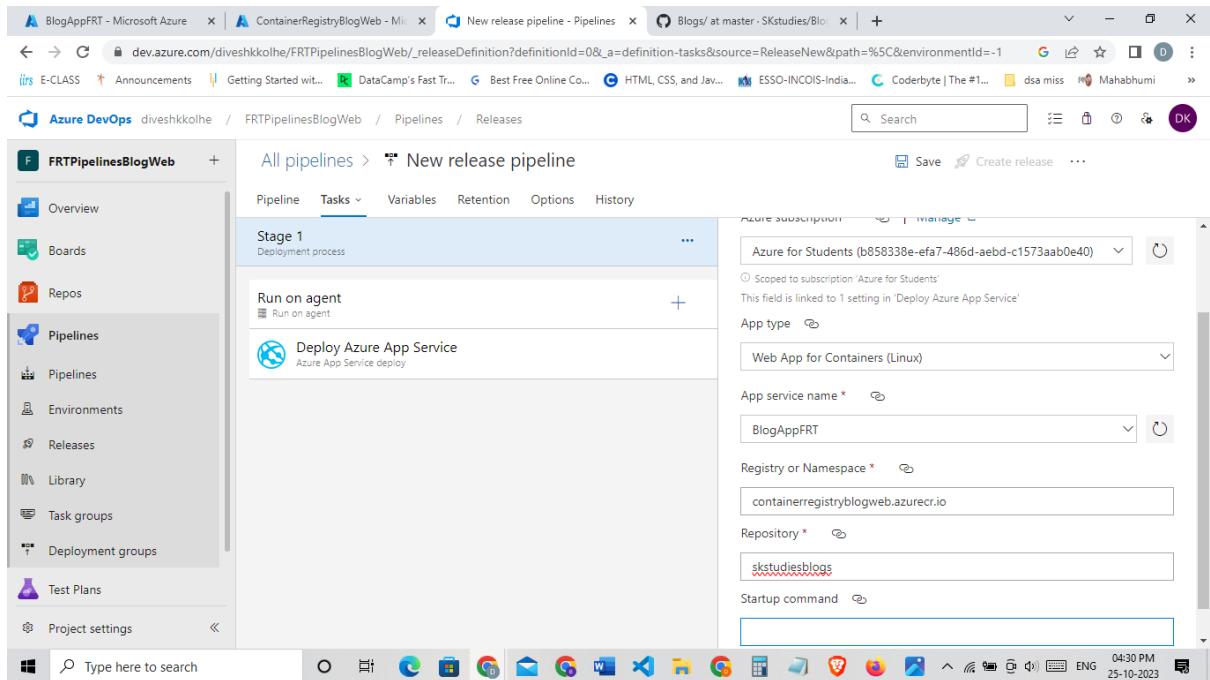
**Artifact Source:** \_SKstudies.Blogs

**Stage 1 Task:** Deploy Azure App Service

Screenshot of the Azure DevOps Pipelines interface showing the configuration of Stage 1.

**Stage 1 Settings:**

- Stage name: Stage 1
- Parameters: Azure subscription (selected: Azure for Students (b858330e-ef87-486d-a))
- Authorization: Authorize (button)
- Description: Click Authorize to configure an Azure service connection. A new Azure service principal will be created and added to the Contributor role, having access to all resources in the selected subscription. To restrict the scope of the service principal to a specific resource group, see connect to Microsoft Azure.
- App type: Web App on Windows
- App service name: (redacted)



Screenshot of the Azure DevOps 'Create a new release' pipeline configuration screen.

The pipeline has one stage named 'Stage 1' with the deployment process set to 'Run on agent'. A single task, 'Deploy Azure App Service', is listed under this stage.

Release settings include:

- Source alias: '\_SKstudies.Blogs'
- Version: '20231025.1'

A 'Release description' field is present but empty.

At the bottom right, there are 'Create' and 'Cancel' buttons.

Screenshot of the Azure DevOps 'Logs' tab for the 'Stage 1' of the 'Release-1' pipeline.

The stage status is 'Succeeded'. The deployment process is also marked as 'Succeeded'.

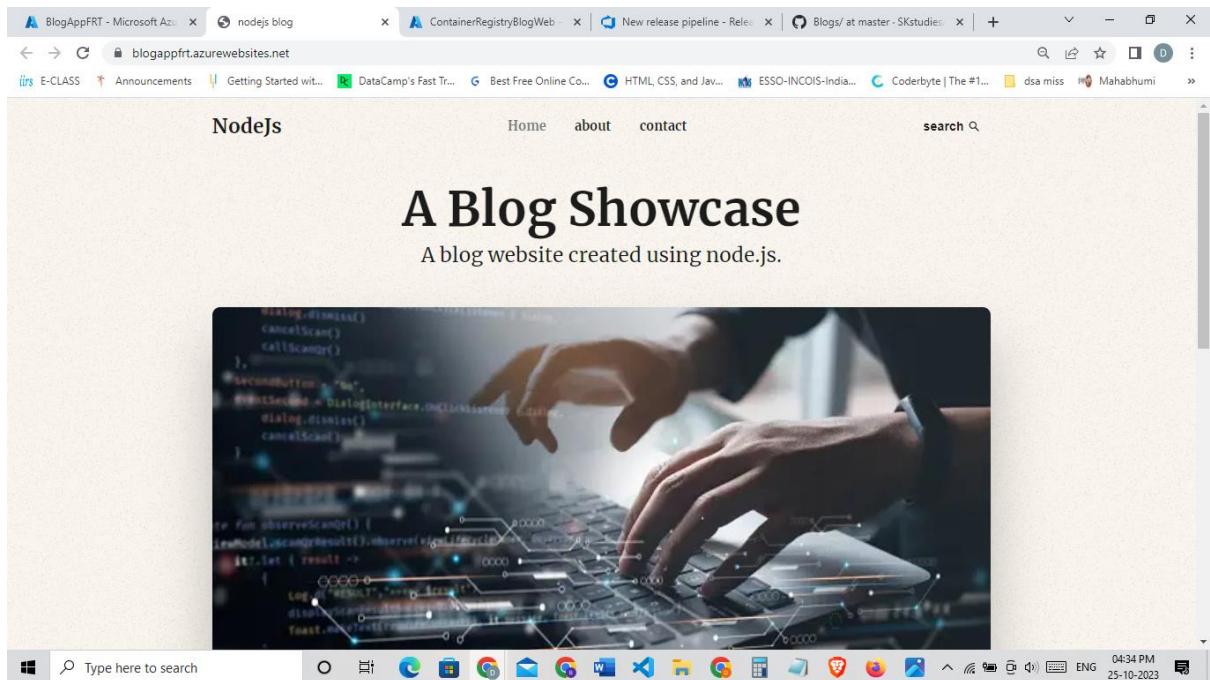
The 'Run on agent' section shows the following tasks:

- Initialize job - succeeded (2s)
- Deploy Azure App Service - succeeded (1m 13s)
- Finalize Job - succeeded (<1s)

Timestamp: Started: 10/25/2023, 4:31:25 PM

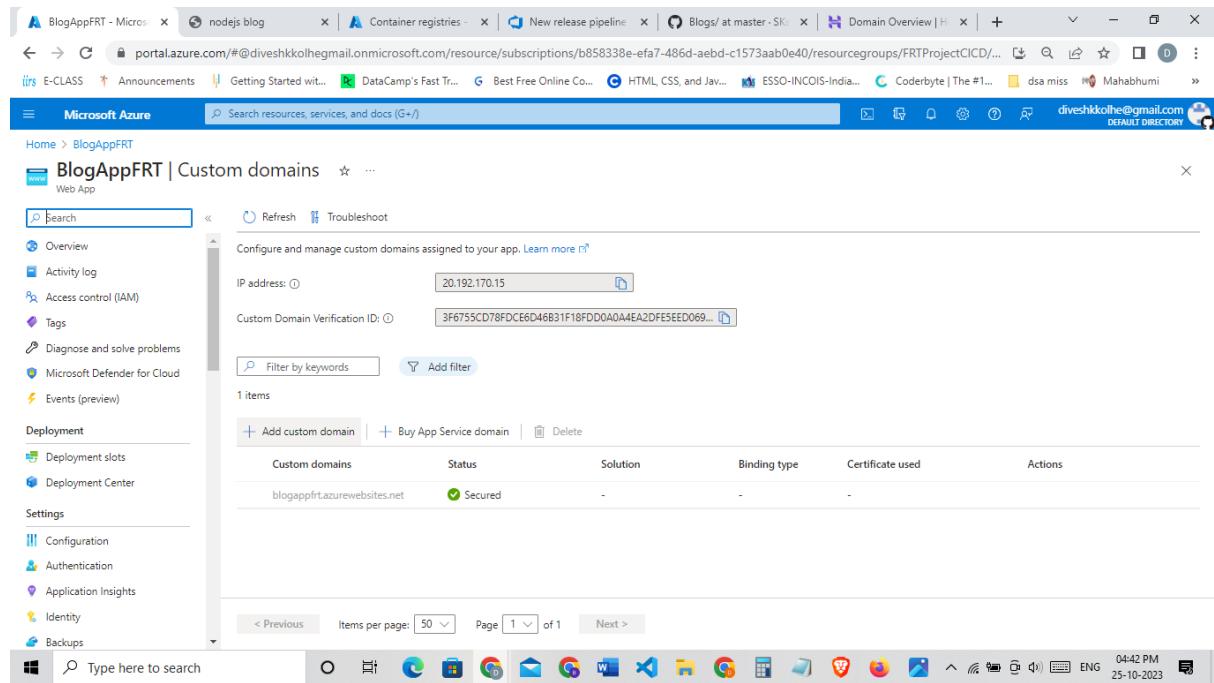
Duration: 1m 16s

## 10. Search Default Domain Webiste Is Running.



11. Create Custom Domain Scale Up The Plan AS Free Plan Does Not Support Custom Domain.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar has several tabs open, including 'BlogAppFRT - Micros...', 'nodejs blog', 'Container registries', 'New release pipeline', 'Blogs/ at master - SK...', 'Domain Overview | H...', and others. The main title is 'BlogAppFRT | Custom domains'. On the left, a sidebar menu is visible with items like 'Deployment slots', 'Deployment Center', 'Settings', 'Configuration', 'Authentication', 'Application Insights', 'Identity', 'Backups', 'Custom domains' (which is selected and highlighted in blue), 'Certificates', 'Networking', 'Scale up (App Service plan)', 'Scale out (App Service plan)', 'Service Connector', and 'Locks'. Below the sidebar is a search bar and a refresh/troubleshoot button. The main content area displays a table for managing custom domains. The table has columns: 'Custom domains', 'Status', 'Solution', 'Binding type', 'Certificate used', and 'Actions'. There is one row listed with the IP address '20.192.170.15' and a verification ID '3F6755CD78FDCE6D46831F18FDD0A0A4EA2DFE5E069...'. Below the table, a section titled 'Upgrade to enable custom domains' with a 'Upgrade now' button is shown. At the bottom, there are navigation links for 'Previous', 'Page', 'Next', and a search bar.



# 12. Add A Custom Domain With App Service Managed Certificate.

The screenshot shows the Microsoft Azure portal interface for managing custom domains. On the left, there's a sidebar with navigation links like Home, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Deployment (Deployment slots, Deployment Center), Settings (Configuration, Authentication, Application Insights, Identity, Backups), and a search bar. The main content area is titled 'BlogAppFRT | Custom domains' and shows a table of custom domains assigned to the app. One row is visible: 'blogappfrt.azurewebsites.net' with status 'Secured'. To the right, a modal window titled 'Add custom domain' is open. It has sections for 'Domain provider' (radio button selected for 'App Service Domain'), 'TLS/SSL certificate' (radio button selected for 'App Service Managed Certificate'), 'TLS/SSL type' (radio button selected for 'SNI SSL'), 'Domain' (text input 'azurereproject.cloud'), and 'Hostname record type' (dropdown 'A record (example.com)'). Below these are sections for 'Domain validation' (with placeholder text) and 'Export to CSV' (button). At the bottom of the modal are 'Validate', 'Add', and 'Cancel' buttons. The status bar at the bottom right shows '05:01 PM 25-10-2023'.

This screenshot is similar to the one above, but the 'Domain provider' section shows 'Add certificate later' selected instead of 'App Service Domain'. The 'TLS/SSL type' section shows both 'SNI SSL' and 'IP based SSL' radio buttons, with 'SNI SSL' selected. The rest of the dialog, including the domain entry, validation section, and table of DNS records, is identical to the first screenshot.

**BlogAppFRT - Microsoft Azure**

portal.azure.com/#/diveshkkolhegmai.onmicrosoft.com/resource/subscriptions/b858338e-efa7-486d-aebd-c1573aab0e40/resourceGroups/FRTProjectCICD/...

E-CLASS Announcements Getting Started with DataCamp's Fast Tr... Best Free Online Co... HTML, CSS, and Java... ESSO-INCOIS-India... Coderbyte | The #1... dsa miss Mahabhum...

**Microsoft Azure** Search resources, services, and docs (G+)

Home > App Services > BlogAppFRT

**App Services** Default Directory (diveshkkolhegmai.onmicrosoft.com)

+ Create ...

Filter for any field...

Name ↗

**BlogAppFRT** ...

Custom domains

Search Application Insights Identity Backups Certificates Networking Scale up (App Service plan) Scale out (App Service plan) Service Connector Locks App Service plan App Service plan Quotas Change App Service plan Development Tools SSH

Configure and manage custom domains assigned to your app. Learn more ⓘ

IP address: 20.192.170.15

Custom Domain Verification ID: 3F6755CD78FDCE6D46B31F18FDD0A0A4EA2DFE5EED069...

Filter by keywords Add filter

2 items

Add custom domain Buy App Service domain Delete

Custom domains	Status	Solution	Binding type	Certificate used
azureproject.cloud	Secured	-	SNI SSL	-
blogappfrt.azurewebsites.net	Secured	-	-	-

< Previous Items per page: 50 Page 1 of 1 Next >

Type here to search

Windows Start button 20 Internet Explorer Edge Google Chrome Microsoft Edge Firefox Microsoft Word Microsoft Excel Microsoft PowerPoint Microsoft Outlook Microsoft OneDrive Microsoft OneNote Microsoft Project Microsoft Visio Microsoft Access Microsoft Publisher Microsoft Word Microsoft Excel Microsoft PowerPoint Microsoft OneDrive Microsoft OneNote Microsoft Project Microsoft Visio Microsoft Access Microsoft Publisher 08:55 PM 30-10-2023 ENG

Add custom domain nodejs blog Container registries New release pipeline Blogs/ at master · SK DNS / Nameservers

E-CLASS Announcements Getting Started with DataCamp's Fast Tr... Best Free Online Co... HTML, CSS, and Java... ESSO-INCOIS-India... Coderbyte | The #1... dsa miss Mahabhum...

**HOSTINGER** Home Hosting Emails Domains VPS Billing Pro Panel BETA

Change Nameservers

Manage DNS records These records define how your domain behaves. Common uses include pointing your domain at web servers or configuring email delivery for your domain.

Type: TXT Name: @asuid.www Name value: 3F6755CD78FDCE6D46B31F18F TTL: 14400 Add Record

Search

Type	Name	Priority	Content	TTL	Action
CNAME	www	0	azureproject.cloud	300	Delete Edit

Give feedback

Type here to search

Windows Start button 20 Internet Explorer Edge Google Chrome Microsoft Edge Firefox Microsoft Word Microsoft Excel Microsoft PowerPoint Microsoft OneDrive Microsoft OneNote Microsoft Project Microsoft Visio Microsoft Access Microsoft Publisher 04:46 PM 25-10-2023 ENG

hpanel.hostinger.com/domain/azureproject.cloud/dns

E-CLASS Announcements Getting Started with DataCamp's Fast Tr... Best Free Online Co... HTML, CSS, and Java... ESSO-INCOIS-India... Coderbyte | The #1... dsa miss Mahabumi

HOSTINGER Home Hosting Emails Domains VPS Billing Pro Panel BETA

Manage DNS records

DNS Record created successfully

Type: A Name: @ Points to: Points to: 14400 Add Record

Search

Type	Name	Priority	Content	TTL	Delete	Edit
TXT	asuid.ww	0	"3F6755CD78FDCE6D46B3F18FDD0AOA4EA2D FE5EED069B2A7F9A2464B57FDC21"	14400	Delete	Edit
CNAME	www	0	blogappfrt.azurewebsites.net	300	Delete	Edit

Give feedback

Type here to search

04:52 PM 25-10-2023

portal.azure.com/#@diveshkkolhegmai.onmicrosoft.com/resource/subscriptions/b858338e-efa7-486d-aebd-c1573aab0e40/resourceGroups/FRTProject...

E-CLASS Announcements Getting Started with DataCamp's Fast Tr... Best Free Online Co... HTML, CSS, and Java... ESSO-INCOIS-India... Coderbyte | The #1... dsa miss Mahabumi

Microsoft Azure Search resources, services, and docs (G+)

Home > App Services > BlogAppFRT

App Services Default Directory (diveshkkolhegmai.onmicrosoft.com)

+ Create ...

Filter for any field...

Name

BlogAppFRT

Certificates

BlogAppFRT | Certificates

Web App

Search Refresh Troubleshoot Send your feedback

Managed certificates Bring your own certificates (.pfx) Public key certificates (.cer)

App Service Managed Certificates are free of cost and fully managed by App Service to maintain the safety and security of your site at the highest level. To understand how to create a managed certificate for your app to consume, click on the learn more link. Learn more

Filter by keywords Add filter

1 items

+ Add certificate Delete

Certificate Status	Domain	Certificate Name	Solution
No action needed	azureproject.cloud	azureproject.cloud-BlogAppFRT	-

Page 1 of 1 Items per page: 50 Page 1 of 1 Next < Previous

https://portal.azure.com/#@diveshkkolhegmai.onmicrosoft.com/resource/subscriptions/b858338e-efa7-486d-aebd-c1573aab0e40/resourceGroups/FRTProjectCICD/providers/Microsoft.Web/sites/BlogAppFRT/certificatesReact

Type here to search

09:33 AM 26-10-2023

# CICD pipeline in action

## 13. Commit Changes To github.

The screenshot shows a Microsoft Windows desktop environment. A browser window is open to the GitHub repository 'SKstudies/Blogs'. The 'Commits' tab is selected. The 'master' branch is chosen. The commit history for October 26, 2023, is displayed, showing the following commits:

- Update README.md (SKstudies committed 8 minutes ago) - Verified, hash 5bbad33
- Add README.md (SKstudies committed 12 minutes ago) - hash ccd3126
- Update README.md (SKstudies committed 43 minutes ago) - Verified, hash 779303e
- Merge branch 'master' of https://github.com/SKstudies/Blogs (SKstudies committed 50 minutes ago) - hash e82c51e
- Add README.md (SKstudies committed 51 minutes ago) - hash 586eedb

Below this, a section for 'Commits on Oct 25, 2023' is partially visible. The taskbar at the bottom shows various application icons, and the system tray indicates the date as 26-10-2023 and the time as 10:36 AM.

## 14. Build Pipeline Triggered

The screenshot shows the Azure DevOps interface for the project 'FRTpipelinesBlogWeb'. The left sidebar has 'Pipelines' selected. The main area shows the 'Runs' tab for the 'SKstudies.Blogs' repository. The table lists the following pipeline runs:

Description	Stages	Time
#20231026.4 • Update README.md Individual CI for master 5bbad335	✓	9m ago 43s
#20231026.3 • Add README.md Individual CI for master ccd31261	✓	12m ago 44s
#20231026.2 • Update README.md Individual CI for master 779303e8	✓	43m ago 54s
#20231026.1 • Merge branch 'master' of https://github.com/SKstudies/Blogs into master Individual CI for master e82c51e7	✓	51m ago 51s
#20231025.1 • Set up CI with Azure Pipelines Individual CI for master 2d013821	✓	Yesterday 2m 25s

The taskbar at the bottom shows various application icons, and the system tray indicates the date as 26-10-2023 and the time as 10:37 AM.

## 15. Containers Created In ACR.

The screenshot shows the Microsoft Azure portal interface. The left sidebar is collapsed. The main content area displays the 'ContainerRegistryBlogWeb | Repositories' page. On the left, there's a navigation menu with 'Repositories' selected under 'Services'. The central area shows a table of tags for the 'skstudiesblogs' repository. The table has columns for 'Tags', 'Digest', and 'Last modified'. The tags listed are 28, 27, 26, 25, and 24, each with a corresponding digest and timestamp. At the top right, there's a 'JSON View' link. The bottom right corner shows the date and time as 26-10-2023 10:37 AM.

## 16. Release Pipeline Triggered.

The screenshot shows the Azure DevOps interface. The left sidebar is collapsed. The main content area displays the 'FRTpipelinesBlogWeb' project's 'Releases' screen. The left sidebar has 'Pipelines' selected. The central area shows a list of releases under the 'New release pipeline' section. There are five releases listed: 'Release-5' (Created 10/26/2023, 10:29:03 AM), 'Release-4' (Created 10/26/2023, 10:26:00 AM), 'Release-3' (Created 10/26/2023, 9:55:03 AM), 'Release-2' (Created 10/26/2023, 9:47:32 AM), and 'Release-1' (Created 10/25/2023, 4:31:12 PM). Each release entry includes a 'Stage 1' button. At the top right, there are 'Edit' and 'Create release' buttons. The bottom right corner shows the date and time as 26-10-2023 10:38 AM.

## 17. Successful Deployment On App Service.

The screenshot shows the Microsoft Azure App Services Deployment Center for the 'BlogAppFRT' web app. The deployment history table lists four successful deployments:

Date	Time	Commit ID	Author	Status	Message
Thursday, October 26, 2023	10:30:56 AM +05:30	5169829	Microsoft.Visu...	Success (Active)	Deployed successfully ↳ Source Version: 5bad33584 ↳ Build: 20231026.4 ↳ Releaser: 5
	10:27:21 AM +05:30	4169829	Microsoft.Visu...	Success	Deployed successfully ↳ Source Version: cd31261ca ↳ Build: 20231026.3 ↳ Releaser: 4
	9:55:57 AM +05:30	3169829	Microsoft.Visu...	Success	Deployed successfully ↳ Source Version: 779303e8c3 ↳ Build: 20231026.2 ↳ Releaser: 3
	9:48:47 AM +05:30	2169829	Microsoft.Visu...	Success	Deployed successfully ↳ Source Version: e82c51e74c ↳ Build: 20231026.1 ↳ Releaser: 2
Wednesday, October 25, 2023	4:32:39 PM +05:30	1169823	Divesh Koihe	Success	Deployed successfully ↳ Source Version: 2d01382169 ↳ Build: 20231025.1 ↳ Releaser: 1

## 18. Site is Running

The screenshot shows a browser window displaying the 'NodeJs' blog website. The page title is 'A Blog Showcase' and the subtitle is 'A blog website created using node.js.' The background features a hand typing on a keyboard with glowing circuit board graphics overlaying it.

# **Future Scope:**

The future scope of your project can be extended to include additional Azure services such as Cosmos DB, Azure Front Door, and Azure Content Delivery Network (CDN) to further enhance and optimize your web application deployment and delivery. Here's how you can integrate these services:

## **1. Cosmos DB Integration:**

**Database Layer Optimization:** Integrate Azure Cosmos DB as a globally distributed, highly available, and scalable database solution. This is particularly useful if your web application requires a database for data storage. Cosmos DB can enhance data scalability, redundancy, and global availability.

**Microservices Architecture:** Implement a microservices architecture and use Cosmos DB to store and manage data for each microservice.

## **2. Azure Front Door:**

**Global Load Balancing and Security:** Azure Front Door can be employed to improve global load balancing, increase application security, and enhance content delivery to users across the world. It offers features such as WAF (Web Application Firewall), DDoS protection, and SSL offloading.

**Application Acceleration:** Use Azure Front Door to accelerate the delivery of your web application by routing user requests to the nearest, most responsive backend service.

## **3. Azure CDN Integration:**

**Content Distribution:** Azure CDN can be used to deliver static content such as images, videos, and large files to users efficiently. It reduces the load on your App Service and improves the user experience by minimizing latency.

**Global Scalability:** Azure CDN caches content in edge locations around the world, ensuring low latency and high availability for users globally.