



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:

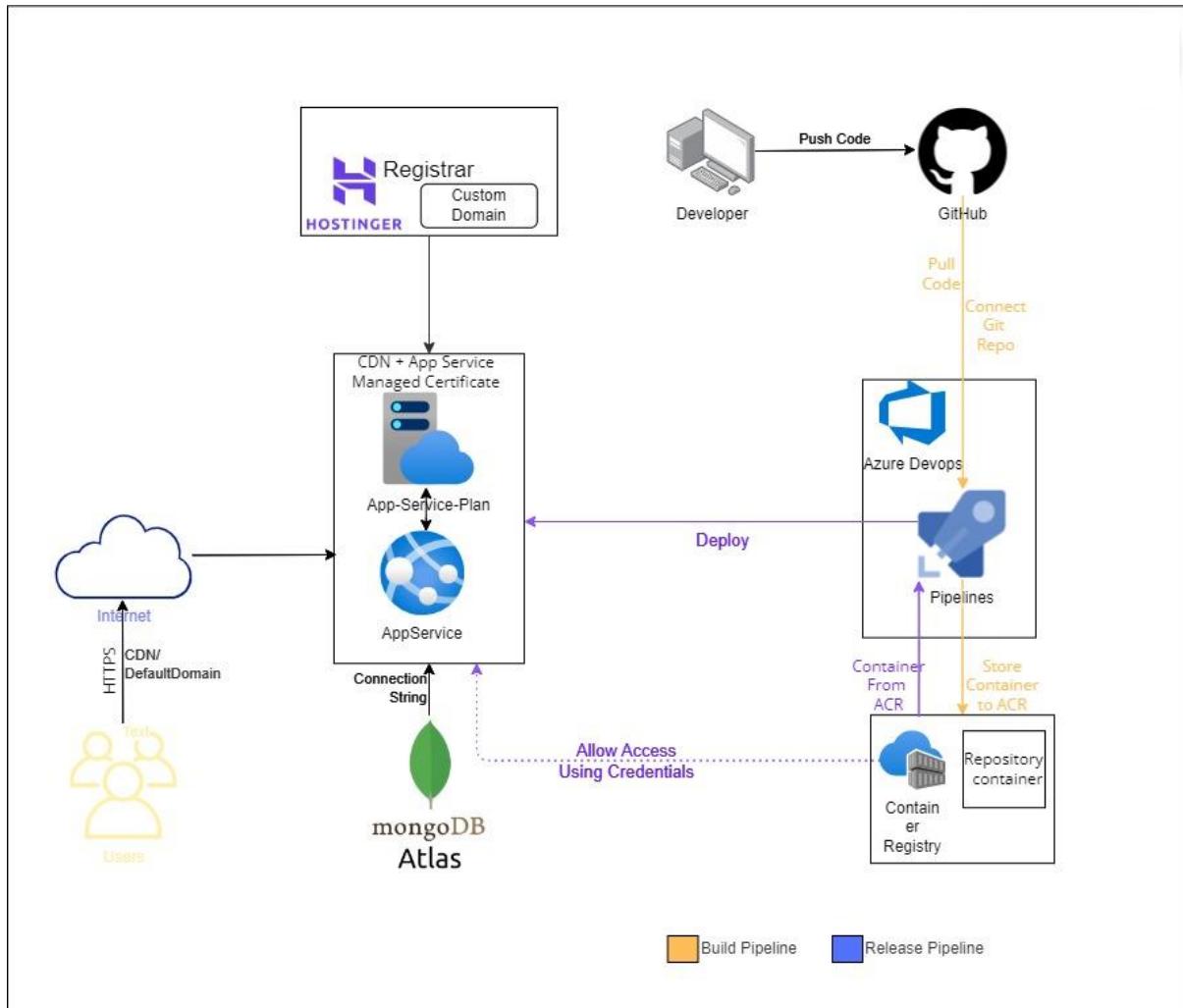
Continuous Integration and Continuous Deployment (CI/CD): The project implements a robust CI/CD pipeline that automates the deployment process. Developers push code changes to a GitHub repository, triggering a series of automated actions that build a Docker container of the web application, store it 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. The admin user is enabled in the access key tab for easy integration.
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.
6. Deployment Center: The deployment center is accessed from the Azure portal, where the Docker container stored in ACR is deployed to the Azure App Service. This ensures the web application is up and running.
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): To enable access to the web application via a custom domain, we utilize Azure DNS for domain name system (DNS) management and resolution. This service ensures proper routing from the custom domain to the Azure App Service hosting our application
3. App Service Managed Certificate: Providing SSL certificates for secure communication with the custom domain..

Complete Step By Step Process:

1. Create Container Repository

Validation passed

Setting	Value
Registry name	ContainerRegistryBlogWeb
Subscription	Azure for Students
Resource Group	FRTProjectCICD
Location	Central India
Availability zones	Disabled
Pricing plan	Standard
Networking	
Public network access	Yes
Encryption	
Customer-Managed Key	Disabled
Identity	None
Key Vault	None
Encryption key	None
Version	None

Create < Previous Next > Download a template for automation

A Create container registry Learning | Future Projects - Home Azure Container Registry service SKStudies/Blog... +

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

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... diveshkkolle@gmail.com DEFAULT DIRECTORY DIVESHOK...

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

Home > Container registries >

Create container registry

pipelines. Use Azure Container Registry Tasks to build container images in Azure on-demand, or automate builds triggered by source code updates, updates to a container's base image, or timers. [Learn more](#)

Project details

Subscription * Azure for Students

Resource group * (New) FRTProjectCICD [Create new](#)

Instance details

Registry name * ContainerRegistryBlogWeb .azurecr.io

Location * Central India

Use availability zones Availability zones are activated on premium registries and in regions that support availability zones. [Learn more](#)

Pricing plan * Standard

[Review + create](#) < Previous Next: Networking >

01:22 PM 25-10-2023

Microsoft.Conta... Learning | Future Projects - Home Azure Container Registry service SKStudies/Blog... +

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... diveshkkolle@gmail.com DEFAULT DIRECTORY DIVESHOK...

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

Home >

Microsoft.ContainerRegistry | Overview

Deployment

Search Delete Cancel Redeploy Download Refresh

Overview Your deployment is complete

Deployment name : Microsoft.ContainerRegistry Start time : 10/25/2023, 1:22:23 PM

Subscription : Azure for Students Correlation ID : 651176d8-a014-4abc-9ee5-3dbc60018c17

Resource group : FRTProjectCICD

Deployment details

Resource	Type	Status	Operation details
ContainerRegistryBlogWeb	Container registry	OK	Operation details

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 >

01:26 PM 25-10-2023

2. Create Service Connections From Devops To Github and ACR

The screenshot shows the Azure DevOps interface for creating a new service connection. The left sidebar lists project settings like General, Boards, Pipelines, and GitHub connections. The main area shows an existing service connection named "Devops-ACRconnection". A modal window titled "New GitHub service connection" is open, prompting for authentication method (Grant authorization selected) and OAuth configuration (AzurePipelines). It also includes fields for service connection name and description.

New GitHub service connection

- Authentication method:
 - Grant authorization
 - Personal Access Token
- OAuth Configuration: AzurePipelines
- Details:
 - Service connection name: [empty]
 - Description (optional): [empty]
- Security

New service connection

Choose a service or connection type

- GitHub
- GitHub Enterprise Server

Learn more **Next**

The screenshot shows the 'New Docker Registry service connection' dialog box overlaid on the Azure DevOps interface. The dialog has the following fields:

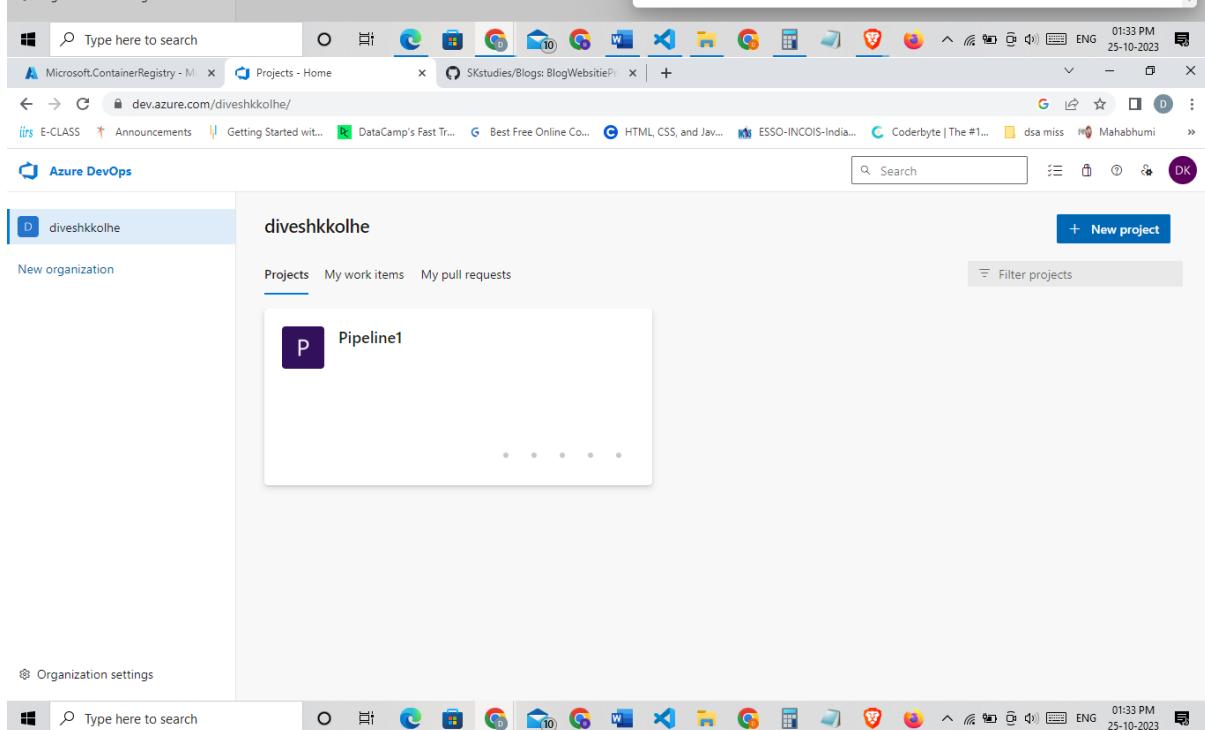
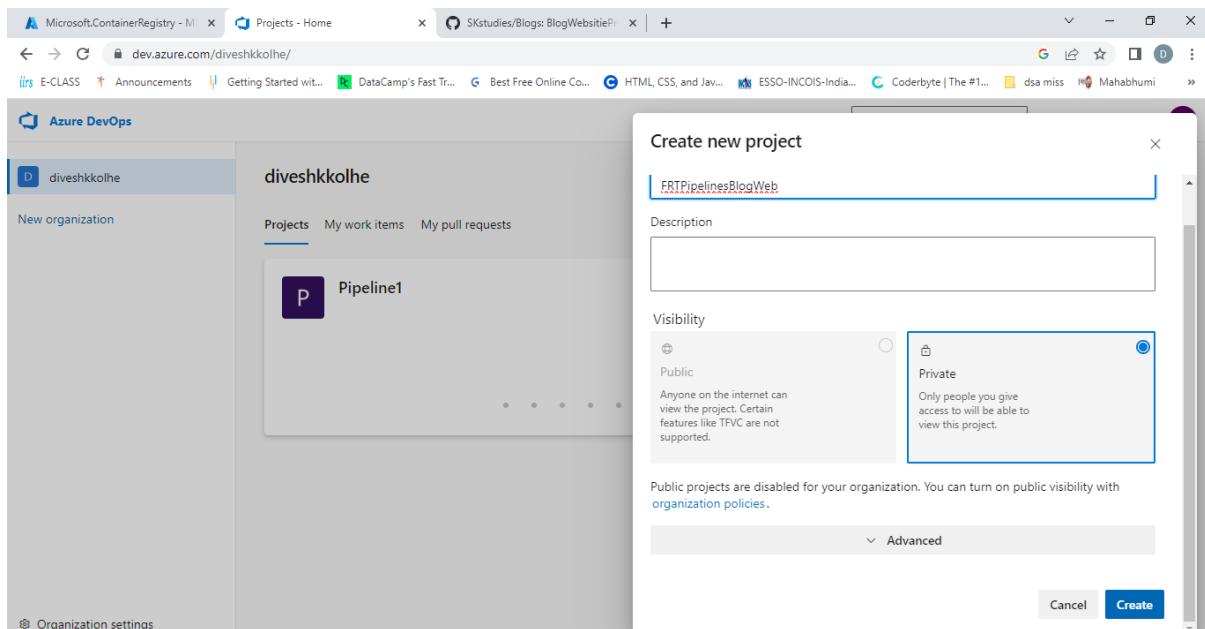
- Service Principal** dropdown (selected)
- Subscription**: Azure for Students (b858338e-efa7-486d-aebd-c1573aab0e...)
- Azure container registry**: ContainerRegistryBlogWeb
- Details** section:
 - Service connection name**: Devops-ACRconnection
 - Description (optional)**: (empty)
- Security** section:
 - Grant access permission to all pipelines
- Buttons**: Back, Save

This screenshot is identical to the one above, except the 'Registry type' section is expanded, showing three options: Docker Hub, Others, and Azure Container Registry. The 'Azure Container Registry' option is selected.

The screenshot shows a Windows desktop environment with three windows open:

- Microsoft Sign in**: A browser window showing a sign-in page for Microsoft. It includes fields for "Email, phone, or Skype", a "Next" button, and a "Sign-in options" link.
- New Docker Registry service connection**: An Azure portal window. Under "Registry type", "Azure Container Registry" is selected. "Authentication Type" is set to "Service Principal". "Subscription" is set to "Azure for Students (b858338e-efa7-486d-aebd-c1573aab0e...)".
- New service connection**: An Azure portal window. A search bar shows "docker". Under "Choose a service or connection type", "Docker Registry" is selected. A "Next" button is visible.

The taskbar at the bottom shows several pinned icons, including Microsoft Edge, File Explorer, Task View, and various Microsoft Office applications. The system tray indicates the date as 25-10-2023 and the time as 02:06 PM.



The screenshot shows the Azure DevOps interface for managing service connections. On the left, the 'Project Settings' sidebar is visible with sections like General, Boards, Pipelines, and Agent pools. The main area is titled 'Service connections' and shows a list of existing connections, including 'Devops-ACRconnection'. A modal window titled 'New GitHub service connection' is open on the right, prompting for connection details. The 'OAuth Configuration' dropdown is set to 'AzurePipelines'. The 'Details' section contains the 'Service connection name' field set to 'Devops-Githubconnection'. Below it is a 'Description (optional)' field which is currently empty. Under the 'Security' section, there is a checked checkbox for 'Grant access permission to all pipelines'. At the bottom of the modal are 'Back' and 'Save' buttons, with the 'Save' button highlighted in blue.

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

The screenshot displays a Windows desktop environment with two open GitHub browser windows. Both windows are set to the 'Code' view of a repository named 'Blogs'.

Top GitHub Window: This window shows the repository's main page. It includes a summary of branches (2 branches), tags (0 tags), and a commit history for the 'master' branch. The commit history lists the following changes:

File	Commit Message	Time Ago
.github/workflows	SKstudies Delete .github/workflows directory	32ded29 3 hours ago
public	image loaded	last week
server	first commit	last week
views	commit	11 hours ago
.gitignore	RM	12 hours ago
Dockerfile	Update Dockerfile	16 hours ago
index.js	first commit	last week
package-lock.json	one more test	last week
package.json	one more test	last week

Bottom GitHub Window: This window shows the content of the 'Dockerfile' file. The code is as follows:

```
1 # Use an official Node.js runtime as a parent image
2 FROM node:14
3
4 # Set the working directory in the container
5 WORKDIR /app
6
7 # Copy package.json and package-lock.json to the container
8 COPY package*.json .
9
10 # Install application dependencies
11 RUN npm install
12
13 # Copy the rest of your application's source code to the container
14 COPY .
15
16 # Expose the port your application will run on
17 EXPOSE 5000
```

4. Create Build Pipeline.

The screenshot shows two windows side-by-side in a Microsoft Edge browser.

Top Window (Completed Pipeline):

- Title Bar:** Pipelines - Run 20231025.1 | Blogs/ at master · SKstudies/Blogs
- Address Bar:** dev.azure.com/diveshkkolhe/FRTpipelinesBlogWeb/_build/results?buildId=24&view=results
- Left Sidebar:** Overview, Boards, Repos, Pipelines (selected), Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, Project settings.
- Summary Tab:** Triggered by SKstudies. Repository and version: SKstudies/Blogs, master. Time started and elapsed: Today at 4:15 PM, 2m 25s. Related: 0 work items. Tests and coverage: Get started.
- Warnings Tab:** 1 warning: No data was written into the file /home/vsts/work/_temp/task_outputs/build_1698230765286.txt. Build and push an image to container registry.
- Jobs Tab:** Build job status: Success, Duration: 2m 19s.

Bottom Window (New Pipeline Creation):

- Title Bar:** Microsoft.ContainerRegistry - M | New pipeline - Pipelines | Blogs/ at master · SKstudies/Blogs
- Address Bar:** dev.azure.com/diveshkkolhe/FRTpipelinesBlogWeb/_apps/hub/ms-vs-build-web.ci-designer-hub?sourceProvider=github&telemetrySession=e00f4728-93ec-4aea...
- Left Sidebar:** Overview, Boards, Repos, Pipelines (selected), Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, Project settings.
- Review Tab:** You selected a public repository, but this is not a public project. Go to project settings to change this.
- Review your pipeline YAML:** A code editor showing the azure-pipelines.yml file for a Docker build and push to Azure Container Registry.
- Save and run Dialog:** Commit message: Set up CI with Azure Pipelines. Optional extended description: Add an optional description... Radio buttons: Commit directly to the master branch (selected) or Create a new branch for this commit. Save and run button.

The screenshot shows two side-by-side views of the Azure DevOps Pipelines interface.

Left View (Pipeline Configuration):

- Header:** Microsoft.ContainerRegistry - M | New pipeline - Pipelines | Blogs/ at master · SKstudies/Blo... | +
- Top Bar:** dev.azure.com/diveshkkolhe/FRTPIPelinesBlogWeb/_apps/hub/ms.vss-build-web.ci-designer-hub?sourceProvider=github&telemetrySession=e00f4728-93ec-4aea-... G 🔒 ⚙ ☆ 📁 D : E-CLASS Announcements Getting Started wit... DataCamp's Fast Tr... Best Free Online Co... HTML, CSS, and Jav... ESSO-INCOIS-India... Coderbyte | The #1... dsa miss Mahabhum... ↗
- Page Title:** Azure DevOps | FRTPIPelinesBlogWeb / Pipelines
- Left Sidebar:** Overview, Boards, Repos, Pipelines (selected), Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, Project settings.
- Main Content:** Review your pipeline YML
- Code Editor:** SKstudies/Blogs / azure-pipelines.yml *
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-f9e-4718-a3ab-87aca529528c'
14 -imageRepository: 'skstudiesblogs'
15 containerRegistry: "containerregistryblogweb.azurecr.io"
16 dockerFilePath: '\$(Build.SourcesDirectory)/Dockerfile'
- Buttons:** Variables, Save and run, Show assistant.

Right View (Task Configuration):

- Header:** Microsoft.ContainerRegistry - M | New pipeline - Pipelines | Blogs/ at master · SKstudies/Blo... | +
- Top Bar:** dev.azure.com/diveshkkolhe/FRTPIPelinesBlogWeb/_apps/hub/ms.vss-build-web.ci-designer-hub?sourceProvider=github&telemetrySession=e00f4728-93ec-4aea-... G 🔒 ⚙ ☆ 📁 D : E-CLASS Announcements Getting Started wit... DataCamp's Fast Tr... Best Free Online Co... HTML, CSS, and Jav... ESSO-INCOIS-India... Coderbyte | The #1... dsa miss Mahabhum... ↗
- Page Title:** Azure DevOps | FRTPIPelinesBlogWeb / Pipelines
- Left Sidebar:** Overview, Boards, Repos, Pipelines (selected), Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, Project settings.
- Main Content:** Configure your pipeline
- Task List:**
 - Docker (Build a Docker image)
 - Docker (Build and push an image to Azure Container Registry)
 - Deploy to Azure Kubernetes Service (Build and push image to Azure Container Registry; Deploy to Azure Kubernetes Service)
 - Node.js (Build a general Node.js project with npm.)
 - Node.js Express Web App to Linux on Azure (Build a Node.js Express app and deploy it to Azure as a Linux web app.)
 - Node.js with Vue (Build a Node.js project that uses Vue.)
 - Node.js with webpack (Build a Node.js project with webpack.)
- Modal Dialog (Docker Task):**
 - Container registry:** ContainerRegistryBlogWeb
 - Image Name:** skstudiesblogs
 - Dockerfile:** \$(Build.SourcesDirectory)/Dockerfile
- Buttons:** Back, Validate and configure.

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 "Configure your pipeline" and shows a "Docker" task selected. The "Configure" tab is active. The "Container registry" dropdown is set to "ContainerRegistryBlogWeb". The "Image Name" field contains "skstudiesblogs". The "Dockerfile" field contains "\${Build.SourcesDirectory}/Dockerfile". At the bottom right of the configuration panel are "Back" and "Validate and configure" buttons.

The screenshot shows the same Azure DevOps Pipelines interface, but the "Select" tab is now active. The main area is titled "Select a repository". A "Filter by keywords" input field is present. Two repositories are listed: "SKstudies/Blogs" (last used on Oct 15) and "SKstudies/ap" (last used on Oct 6). Below the list is a note: "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 indicates the time as 04:14 PM on 25-10-2023.

Screenshot of the Azure DevOps Pipelines interface showing the "Where is your code?" step.

The interface includes:

- A left sidebar with navigation links: Overview, Boards, Repos, Pipelines (selected), Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Project settings.
- A top navigation bar with tabs: Connect, Select, Configure, and Review.
- A main area titled "Where is your code?" listing integration options:
 - Azure Repos Git (YAML)
 - Bitbucket Cloud (YAML)
 - GitHub (YAML)
 - GitHub Enterprise Server (YAML)
- A bottom section titled "Create your first Pipeline" with a sub-section "Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes." and a "Create Pipeline" button.

Screenshot of the Azure DevOps Pipelines interface showing the "Create your first Pipeline" step.

The interface includes:

- A left sidebar with navigation links: Overview, Boards, Repos, Pipelines (selected), Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Project settings.
- A top navigation bar with tabs: Connect, Select, Configure, and Review.
- A main area featuring a cartoon illustration of a robot and a person working on a laptop.
- A section titled "Create your first Pipeline" with the sub-section "Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes." and a "Create Pipeline" button.

The screenshot shows the Azure DevOps Pipelines interface. On the left, a sidebar menu for the project "FRTPIPelinesBlogWeb" is visible, with "Pipelines" selected. The main area displays a "Create your first Pipeline" wizard. It features a cartoon illustration of a robot and a person working together. Below the illustration, the text reads: "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 the wizard area.

Create your first Pipeline

Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes.

Create Pipeline

The screenshot shows the Azure DevOps Pipelines interface again, but this time it displays the results of a completed pipeline run. The sidebar menu is identical. The main area shows a "Build" job under the "Jobs in run #20231025.1" section. The build log details are listed:

```
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 in the top right corner of the build card.

5. Docker Container Created In The Repository.

The screenshot shows the Microsoft Azure portal interface. The URL is https://portal.azure.com/#view/Microsoft_Azure_ContainerRegistries/RepositoryBlade/id/%2Fsubscriptions%2Fb858338e-efa7-486d-aebd-c1573aab0e40%2FresourceGroups%2FRFTProjectCICD%2BContainerRegistry%2BContainerRegistryBlogWeb%2BContainerRegistryBlogWeb%2BRepositories. The page title is "ContainerRegistryBlogWeb | Repositories". The left sidebar shows "Container registry" with "Repositories" selected. The main content area shows the "skstudiesblogs" repository details: Tag count 1, Manifest count 1. The "Tags" table lists 24 entries, with the last entry being sha256:8a5a5ab23fdcad2c40b8eae308c7deac... Last modified 10/25/2023, 4:17 PM GMT+5:30.

6. Allow Access.

The screenshot shows the Microsoft Azure portal interface. The URL is <https://portal.azure.com/#diveshkkolhegmai.onmicrosoft.com/resource/subscriptions/b858338e-efa7-486d-aebd-c1573aab0e40/resourceGroups/RFTProjectCICD%2BContainerRegistry%2BContainerRegistryBlogWeb%2BContainerRegistryBlogWeb%2BAccessKeys>. The page title is "Container RegistryBlogWeb | Access keys". The left sidebar shows "Container registry" with "Access keys" selected. The main content area shows two access keys: "ContainerRegistryBlogWeb" and "containerregistryblogweb.azurecr.io". A "Copy to clipboard" button is visible next to the first key.

7. Create Web App.

The screenshot shows two windows of the Microsoft Azure portal. The top window displays the 'App Services' dashboard with a search bar and filter options. It shows a message: 'No app services to display' and a brief description: '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.' The bottom window shows the 'Create Web App' wizard. It has sections for 'Project Details' (Subscription: Azure for Students, Resource Group: FRTProjectCICD), 'Instance Details' (Name: BlogAppFRT.azurewebsites.net, Publish: Docker Container, Operating System: Linux, Region: Central India), and a note about finding the App Service Plan. At the bottom are 'Review + create' and 'Next: Docker >' buttons. The taskbar at the bottom of both windows shows various pinned icons.

App Services - Microsoft Azure

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

Home > App Services

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

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

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: Azure for Students

Resource Group: FRTProjectCICD

Instance Details

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

Name: BlogAppFRT.azurewebsites.net

Publish: Docker Container

Operating System: Linux

Region: Central India

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

Review + create < Previous Next: Docker >

Screenshot of the Azure portal showing the 'Create Web App' wizard. Step 1: Basic settings.

Operating System: Linux (selected)

Region: Central India

Pricing plans: App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. Learn more.

App Service Plan: (New) ASP-FRTProjectCICD-9446

Pricing plan: Free F1 (Shared infrastructure)

Zone redundancy: Enabled: Your App Service plan and the apps in it will be zone redundant.

Review + create | < Previous | Next : Docker >

Screenshot of the Azure portal showing the 'Create Web App' wizard. Step 2: Docker settings.

Basics | **Docker** (selected) | Networking | Monitoring | Tags | Review + create

Image Source: Quickstart

Quickstart options:

- Sample:** NGINX (selected)
- NGINX web server default site

Image and tag: mcr.microsoft.com/appsvc/staticsite:latest

Review + create | < Previous | Next : Networking >

A Create Web App - Microsoft Az... x A ContainerRegistryBlogWeb - Mi... x Pipelines - Run 20231025.1 logs x Blogs/ at master · SKstudies/Blo... x +

← → C portal.azure.com/#create/Microsoft.WebSite

iirs E-CLASS Announcements Getting Started wit... 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-)                         <img alt="Email icon"

8. Modify Application Settings and Deploy.

The screenshot shows the Azure portal interface for managing application settings. It displays two separate 'Add/Edit application setting' dialog boxes.

Top Dialog (JWT_SECRET):

- Name: JWT_SECRET
- Value: MySecretBlog
- Deployment slot setting

Bottom Dialog (MONGODB_URI):

- Name: MONGODB_URI
- Value: MONGODB_URI mongodb+srv://braveunknown123:hXtmJaYOMbiFT60@cluster0.nddw6pl.mongodb.net/blog
- Deployment slot setting

The left sidebar shows the navigation menu for the Azure portal, including Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Deployment slots, Deployment Center, Configuration (selected), Authentication, Application Insights, Identity, and Backups.

The screenshot shows the Azure Deployment Center for the 'BlogAppFRT' web app. The left sidebar is collapsed, and the main area displays the 'Settings' tab under the 'Deployment Center' section. The 'Container type' is set to 'Single Container', 'Registry source' is 'Azure Container Registry', and 'Subscription ID' is 'Azure for Students'. Under 'Authentication', 'Admin Credentials' is selected. The 'Registry' dropdown is set to 'ContainerRegistryBlogWeb'. The 'Image' dropdown is set to 'skstudieblogs', and the 'Tag' dropdown is set to '24'. The 'Continuous deployment' option is turned 'On'. The top navigation bar shows several tabs: 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.

BlogAppFRT | Deployment Center

Container type: Single Container

Registry source: Azure Container Registry

Subscription ID: Azure for Students

Authentication: Admin Credentials

Registry: ContainerRegistryBlogWeb

Image: skstudieblogs

Tag: 24

Continuous deployment: On

The screenshot shows the Azure Deployment Center 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 'Settings' tab with 'Logs' and 'FTPS credentials' tabs available. Under 'Registry settings', it specifies a Container type as 'Single Container', Registry source as 'Azure Container Registry', Subscription ID as 'Azure for Students', and Authentication as 'Admin Credentials' using the 'ContainerRegistryBlogWeb' registry.

Use these settings to configure your container app deployment model and registry. We recommend GitHub Actions for greater operational efficiency. [Learn more](#)

Source*

Container Registry: Set up your app to pull the container image from a registry.
 GitHub Actions: Build, deploy, and manage your container app automatically with GitHub Actions.
 Azure Pipelines: Configure a robust deployment pipeline for your application using Azure Pipelines, part of Azure DevOps Services (formerly known as VSTS).

Registry settings

Container type: Single Container
Registry source: Azure Container Registry
Subscription ID*: Azure for Students
Authentication: Admin Credentials
Registry*: ContainerRegistryBlogWeb

Type here to search

The screenshot shows the Azure Configuration page for the 'BlogAppFRT' web app. The left sidebar shows the same navigation options as the Deployment Center. The main pane displays application settings. A message at the top says 'Click here to upgrade to a higher SKU and enable additional features. [Learn more](#)'. Below this, it says 'Application settings are encrypted at rest and transmitted over an encrypted channel. You can choose to display them in plain text in your browser by using the controls below. Application Settings are exposed as environment variables for access by your application at runtime. [Learn more](#)'. It includes buttons for '+ New application setting', 'Show values', and 'Advanced edit'. A 'Filter application settings' input field is present. A table lists application settings:

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			

Click here to upgrade to a higher SKU and enable additional features. [Learn more](#)

Application settings are encrypted at rest and transmitted over an encrypted channel. You can choose to display them in plain text in your browser by using the controls below. Application Settings are exposed as environment variables for access by your application at runtime. [Learn more](#)

+ New application setting Show values Advanced edit

Filter application settings

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

Type here to search

9. Create Release Pipeline.

The screenshot shows the Azure DevOps Pipelines interface. On the left, there is a navigation sidebar with options like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Project settings. The Pipelines option is selected. The main area displays a 'New release pipeline' dialog. The pipeline has one stage named 'Stage 1'. Under 'Run on agent', a task named 'Deploy Azure App Service' is listed. To the right, there is a 'Create a new release' dialog with sections for 'Artifacts' (listing '_SKstudies.Blogs' with version 20231025.1) and 'Release description'. At the bottom are 'Create' and 'Cancel' buttons. Below the main interface, there is a taskbar with various icons and system status information.

Screenshot of the Azure DevOps interface showing the creation of a new release pipeline for the FRTpipelinesBlogWeb project. The pipeline consists of a single stage named 'Stage 1' which runs on an agent and performs a 'Deploy Azure App Service' task.

The 'Run on agent' section shows the task configuration:

- App type:** Web App for Containers (Linux)
- App service name:** BlogAppFRT
- Registry or Namespace:** containerregistryblogweb.azurecr.io
- Repository:** skstudieblogs
- Startup command:** (empty)

The status bar at the bottom indicates the date and time: 04:30 PM 25-10-2023.

Screenshot of the Azure DevOps interface showing the creation of a new release pipeline for the FRTpipelinesBlogWeb project. The pipeline consists of a single stage named 'Stage 1' which runs on an agent and performs a 'Deploy Azure App Service' task.

The 'Run on agent' section shows the task configuration:

- App type:** Web App on Windows
- App service name:** (empty)

A note on the right side of the screen says: "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".

The status bar at the bottom indicates the date and time: 04:28 PM 25-10-2023.

The screenshot shows the Azure DevOps interface for creating a new release pipeline. The left sidebar is for the project "FRTpipelinesBlogWeb". The main area displays the "New release pipeline - Pipelines" screen.

Pipeline Overview:

- Artifacts:** Shows an artifact named "_SKstudies.Blogs".
- Stages:** Shows a single stage named "Stage 1" which contains 1 job and 1 task.
- Trigger:** A "Continuous deployment trigger" is selected, with a tooltip explaining it creates a release every time a new build is available.
- Build branch filters:** No filters are added.
- Pull request trigger:** This trigger is disabled.

UI Elements:

- Left Sidebar:** Includes sections for Overview, Boards, Repos, Pipelines (selected), Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Project settings.
- Top Bar:** Shows browser tabs for Microsoft Azure, ContainerRegistryBlogWeb, New release pipeline, and Blogs. It also includes a search bar, save button, and other navigation icons.
- Bottom Bar:** Shows the Windows taskbar with various pinned and running application icons.

The screenshot shows the Azure DevOps Pipelines interface for a project named "FRTpipelinesBlogWeb". The left sidebar navigation includes Overview, Boards, Repos, Pipelines (selected), Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Project settings. The main content area displays a "New release pipeline" configuration. It features two sections: "Artifacts" and "Stages". In the "Artifacts" section, there is a "Continuous deployment trigger" card for a build named "_SKstudies.Blogs" with a status of "Schedule not set". In the "Stages" section, there is a single stage named "Stage 1" which contains one job and one task. To the right of the pipeline diagram, there are two trigger options: "Continuous deployment trigger" (disabled) and "Pull request trigger" (disabled). Below the pipeline diagram, there is a "Build" artifact type selected, with other options like Azure Repos, GitHub, and TFVC available. A note indicates that no version is available for the selected source pipeline. The bottom of the screen shows the Windows taskbar with various pinned icons.

The screenshot shows two side-by-side views of the Azure DevOps Pipelines interface.

Left View: The pipeline configuration screen for a "New release pipeline". It displays the "Artifacts" and "Stages" sections. The "Artifacts" section has a button to "Add an artifact". The "Stages" section shows "Stage 1" with a status of "1 job, 1 t".

Right View: A modal window titled "Add an artifact". It includes a "Source type" section with options like "Build", "Azure Repos ...", "GitHub", and "TFVC". Below this, a "Project" dropdown is set to "FRTpipelinesBlogWeb" and a "Source (build pipeline)" dropdown contains "SKstudies:Blogs". An "Add" button is at the bottom.

Bottom View: The same pipeline configuration screen, but the "Stages" section now shows "Stage 1 Select a template".

Common UI Elements: Both screens include a top navigation bar with tabs like "Pipeline", "Tasks", "Variables", "Retention", "Options", and "History". The left screen also has a sidebar with links for Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Project settings.

The screenshot shows the Azure DevOps Pipelines interface for the project 'FRTpipelinesBlogWeb'. The left sidebar navigation bar is visible, with 'Pipelines' selected. The main content area displays a cartoon illustration of a person launching a rocket from a blue cube. Below the illustration, the text 'No release pipelines found' is centered, followed by the subtext 'Automate your release process in a few easy steps with a new pipeline'. A prominent blue button labeled 'New pipeline' is located at the bottom right of this section.

No release pipelines found

Automate your release process in a few easy steps with a new pipeline

New pipeline

The screenshot shows the Azure DevOps Pipelines interface for the project 'FRTpipelinesBlogWeb'. The left sidebar navigation bar is visible, with 'Pipelines' selected. The main content area displays a deployment log for 'Release-1 > Stage 1'. The log table has two columns: 'Deployment process' and 'Run on agent'. The 'Deployment process' column shows 'Succeeded'. The 'Run on agent' column shows a successful deployment to a 'Hosted Windows 2019' agent. The log details three tasks: 'Initialize job', 'Deploy Azure App Service', and 'Finalize Job', all completed successfully. The log was started at 10/25/2023, 4:31:25 PM and took 1m 16s.

Deployment process	Run on agent
Succeeded	Started: 10/25/2023, 4:31:25 PM Pool: Hosted Windows 2019 with ... · Agent: Hosted Agent ... 1m 16s
	<ul style="list-style-type: none">Initialize job · succeeded 2sDeploy Azure App Service · succeeded 1m 13sFinalize Job · succeeded <1s

10. Search Default Domain Website Is Running.



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

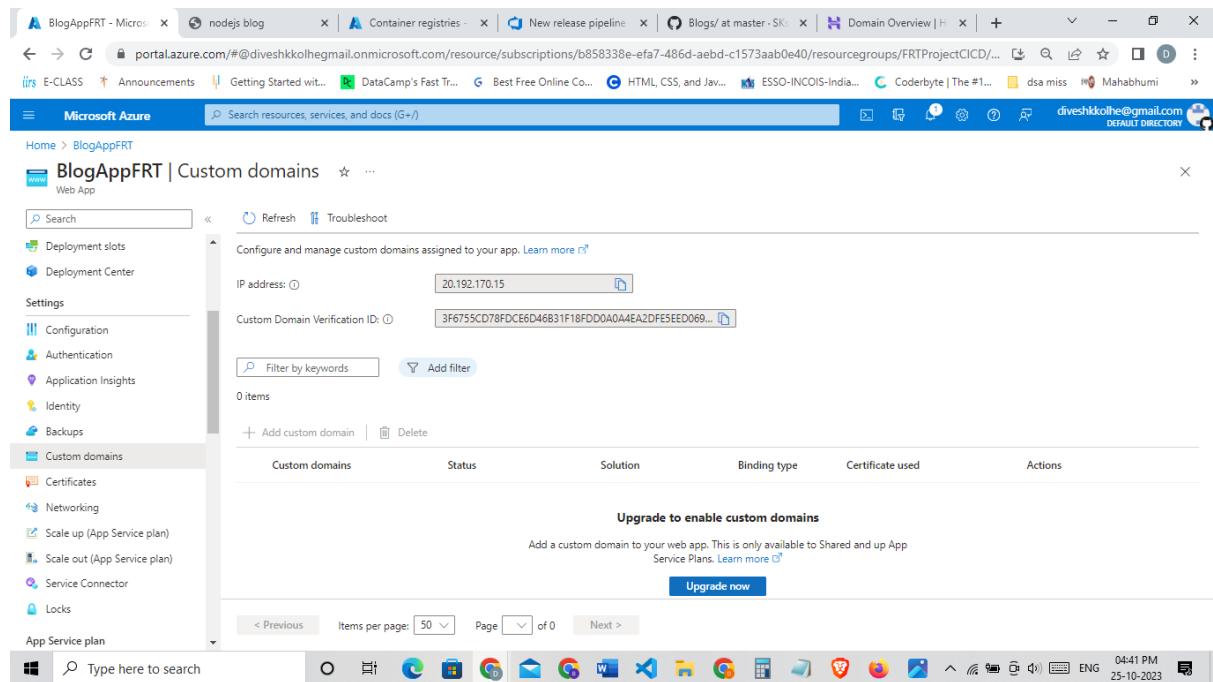
The screenshot shows two windows from the Microsoft Azure portal.

The top window displays the "Select App Service Pricing Plan" page. It lists 18 App Service pricing plans under two categories: "Dev/Test (For less demanding workloads)" and "Production (For most production workloads)".

Name	Custom domain	Auto Scale	Daily backups	Staging slots	Zone Redundant	vNet integration	Single tenant system	Cost per hour (instance)	Cost per month (instance)
Basic B1	✓	Manual	N/A	N/A	-	✓	-		
Basic B2	✓	Manual	N/A	N/A	-	✓	-		
Basic B3	✓	Manual	N/A	N/A	-	✓	-		
Premium v3 P0V3	✓	Rules	50	20	✓	✓	-		
Premium v3 P1V3	✓	Rules, Elastic	50	20	✓	✓	-		
Premium v3 P2V3	✓	Rules, Elastic	50	20	✓	✓	-		
Premium v3 P3V3	✓	Rules, Elastic	50	20	✓	✓	-		
Premium v3 P1mv3	✓	Rules	50	20	✓	✓	-		
Premium v3 P2mv3	✓	Rules	50	20	✓	✓	-		

The bottom window shows the "BlogAppFRT | Custom domains" page for a "Web App". It lists a single custom domain entry:

Custom domains	Status	Solution	Binding type	Certificate used	Actions
blogappfrt.azurewebsites.net	Secured	-	-	-	[Edit]



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

The screenshot shows the Microsoft Azure portal interface for managing custom domains for a web application named "BlogAppFRT".

Left Sidebar: Shows navigation links for 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.

Custom Domains Overview: Displays the IP address (20.192.170.15) and Custom Domain Verification ID (3F6755CD78FDCE6D46B31F18FDD0A0A4EA2DFE5EED069...).

Add custom domain Dialog: This dialog is open on the right.

- Domain provider:** All other domain services (radio button selected).
- TLS/SSL certificate:** App Service Managed Certificate (radio button selected).
- TLS/SSL type:** SNI SSL (radio button selected).
- Domain:** azureproject.cloud
- Hostname record type:** A record (example.com)
- Domain validation:** To validate your domain ownership, copy the hostname records below and enter them with your domain provider. It lists the A record (example.com) and its value (20.192.170.15).
- Table of DNS Records:** Shows two entries:
 - Type: A, Host: @, Value: 20.192.170.15
 - Type: TXT, Host: asuid, Value: 3F6755CD78FDCE6D46B31F18FDD0A0A4EA2DFE5EED069B2A7F9A2464B57FDC21

Buttons at the bottom: Validate, Add, Cancel.

The screenshot shows the Microsoft Azure portal interface. The left sidebar navigation includes Home, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Deployment (with sub-options: Deployment slots, Deployment Center), Settings, Configuration, Authentication, Application Insights, Identity, and Backups. The main content area is titled "BlogAppFRT | Custom domains" under "Web App". It displays the IP address (20.192.170.15) and the Custom Domain Verification ID (3F6755CD78FDCE6D46B31F18FD0A0A4E2DFE5ED069). Below this, there are buttons for "Filter by keywords" and "Add filter". A table lists two items: "Custom domains" (www.azureproject.cloud and blogappfrt.azurewebsites.net), both marked as "Secured" with SNI SSL binding type. At the bottom, there are navigation links for < Previous, Items per page: 50, Page 1 of 1, and Next >. To the right, a "Notifications" panel shows two recent events: "Adding custom domain and SSL binding" for www.azureproject.cloud and "Adding custom domain and SSL binding" for www.azureproject.cloud, both timestamped "a few seconds ago". The top navigation bar has tabs for Container registries, New release pipeline, Blogs, DNS / Nameservers, and more, along with a search bar and user profile information.

The screenshot shows the Hostinger Pro Panel (BETA) with the 'Domains' tab selected. In the 'Manage DNS records' section, a new TXT record is being created. The 'Type' dropdown is set to 'TXT', the 'Name' field contains '@asuid.www', the 'TXT value' field contains '3F6755CD78FDCE6D46B31F18F', and the 'TTL' field is set to '14400'. A purple 'Add Record' button is visible. Below the form, a table lists existing DNS records: a CNAME record for 'www' pointing to 'azureproject.cloud' with a TTL of 300, and a TXT record for '@' with a TTL of 14400. A search bar and filter buttons are also present.

This screenshot shows the same Hostinger Pro Panel interface after a successful A record creation. A green success message 'DNS Record created successfully' is displayed in the top right corner. The 'Manage DNS records' section now includes an A record for '@' with a TTL of 14400. The table below shows the updated list of records, including the newly created A record and the existing TXT and CNAME records. The Windows taskbar at the bottom indicates the screenshot was taken at 04:46 PM on October 25, 2023.

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

Home > App Services > BlogAppFRT

App Services Default Directory (diveshkkolhe@gmail.com)

+ Create ...

Filter for any field... Name ↑

BlogAppFRT ...

Certificates

Search Application Insights Identity Backups Custom domains 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

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

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

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

Type here to search

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

09:33 AM 26-10-2023 ENG

CICD pipeline in action

13. Commit Changes To github.

The screenshot shows a Microsoft Edge browser window with the GitHub URL <https://github.com/SKstudies/Blogs/commits/master>. The page displays a list of commits on the 'master' branch. The commits are as follows:

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

The commits are grouped by date: 'Commits on Oct 26, 2023' and 'Commits on Oct 25, 2023'. The GitHub interface includes standard navigation buttons like back, forward, and search, along with repository settings and insights tabs.

14. Build Pipeline Triggered.

The screenshot shows the Azure DevOps Pipelines interface for the repository 'SKstudies.Blogs'. The left sidebar lists various project management sections: Overview, Boards, Repos, Pipelines (which is selected), Environments, Releases, Library, Task groups, Deployment groups, and Test Plans. The main area displays a table of build runs:

Description	Stages	Time
#20231026.4 • Update README.md	1 stage (green circle)	9m ago (43s)
#20231026.3 • Add README.md	1 stage (green circle)	12m ago (44s)
#20231026.2 • Update README.md	1 stage (green circle)	43m ago (54s)
#20231026.1 • Merge branch 'master' of https://github.com/SKstudies/Blogs into 'master'	1 stage (green circle)	51m ago (51s)
#20231025.1 • Set up CI with Azure Pipelines	1 stage (green circle)	Yesterday (2m 25s)

The table also includes columns for 'Description' and 'Stages'. Each row shows a green circular icon indicating success, the specific commit hash, and the duration of the build. The Azure DevOps interface includes a search bar, a 'Run pipeline' button, and a detailed view for each build run.

15. Containers Created In ACR.

The screenshot shows the Microsoft Azure portal interface. The left sidebar is titled 'ContainerRegistryBlogWeb | Repositories' under 'Container registry'. The main content area displays the 'skstudiesblogs' repository. It shows an 'Essentials' section with a tag count of 5 and a manifest count of 5, both updated on 10/26/2023. Below this is a table of tags, each with a digest and last modified date. The table includes rows for tags 28, 27, 26, 25, and 24. The interface includes a search bar, refresh button, and manage deleted artifacts link.

16. Release Pipeline Triggered.

The screenshot shows the Azure DevOps interface. The left sidebar is titled 'FRTpipelinesBlogWeb' and has sections for Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Project settings. The main content area is titled 'New release pipeline' and shows a list of releases. The table lists five releases: '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 is associated with a Stage 1 status indicator. The interface includes a search bar, edit and create release buttons, and deployment and analytics tabs.

17. Successful Deployment On App Service.

The screenshot shows the Microsoft Azure App Services Deployment Center for the 'BlogAppFRT' web app. The left sidebar lists settings like Configuration, Authentication, Application Insights, Identity, Backups, Custom domains, Certificates, Networking, Scale up (App Service plan), Scale out (App Service plan), Service Connector, and Locks. The main area displays deployment logs for October 26, 2023, and October 25, 2023. Each log entry includes the deployment time, commit ID, author, status, and a detailed message with source version and build information.

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: 5bbad33584 ↳ Build: 20231026.4 ↳ Release: 5
	10:27:21 AM +05:30	4169829	Microsoft.Visu...	Success	Deployed successfully ↳ Source Version: ccd31261ca ↳ Build: 20231026.3 ↳ Release: 4
	9:55:57 AM +05:30	3169829	Microsoft.Visu...	Success	Deployed successfully ↳ Source Version: 799303e8c3 ↳ Build: 20231026.2 ↳ Release: 3
	9:48:47 AM +05:30	2169829	Microsoft.Visu...	Success	Deployed successfully ↳ Source Version: e62c51e74c ↳ Build: 20231026.1 ↳ Release: 2
Wednesday, October 25, 2023	4:32:39 PM +05:30	1169823	Divesh Kolhe	Success	Deployed successfully ↳ Source Version: 2d01382169 ↳ Build: 20231025.1 ↳ Release: 1

18. Site is Running

The screenshot shows a browser window displaying the 'NodeJs' blog website at blogappftrt.azurewebsites.net. The page title is 'A Blog Showcase' and the subtitle is 'A blog website created using node.js.' Below the title is a large image of hands typing on a keyboard with a futuristic, glowing circuit board effect overlaid. The browser taskbar shows other open tabs for Microsoft Azure services.

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.