

Documentation: Azure Web Hosting with Linux and GitHub Deployment

❖ Introduction

Welcome to the documentation on setting up Azure Web Hosting with Linux and deploying projects through your GitHub account. This guide will walk you through the process of creating a web app in Azure, connecting it to GitHub, and deploying your project seamlessly.

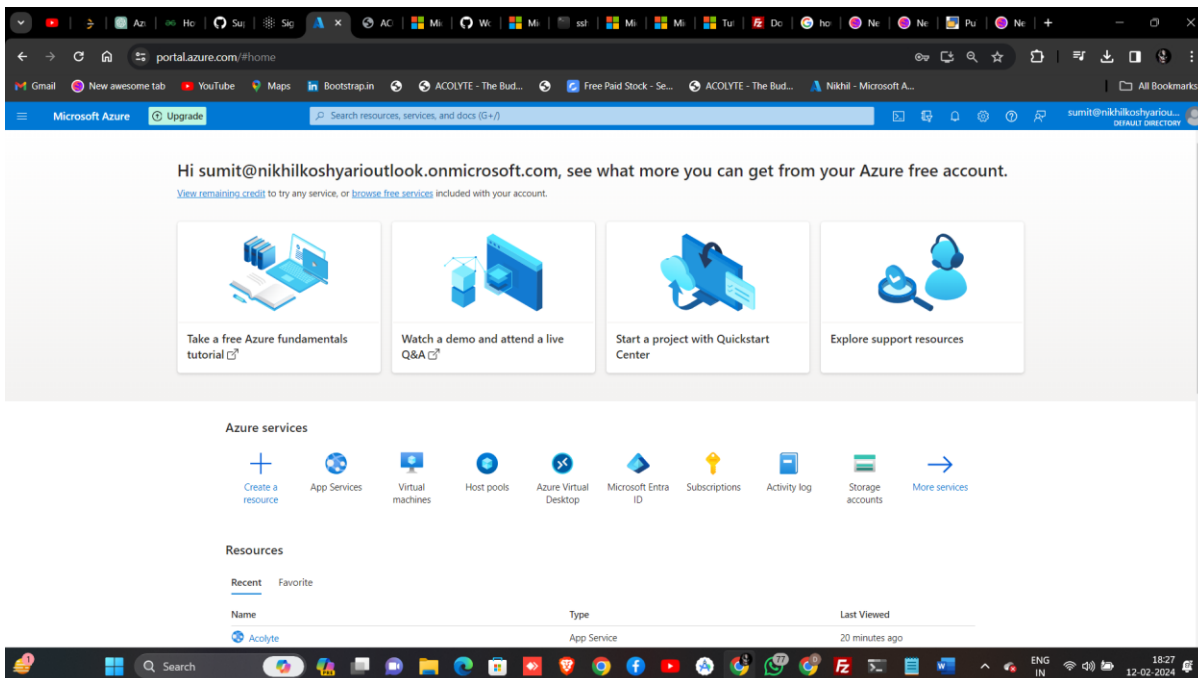
■ Prerequisites

before you begin, ensure you have the following:

- An active Azure account. [Sign up for [Azure](#)] if you don't have one.
- A GitHub account with your project repository ready for deployment.
- Basic familiarity with Git and GitHub.

Step 1: Creating an Azure Web App

1. Sign in to the [[Azure Portal](#)].



Sumit Jadhav

2. Navigate to "Create a resource" > "Web" > "Web App".

Home > App Services > Create Web App

Basics Database Deployment Networking Monitoring Tags Review + create

App Service Web Apps lets you quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. Meet rigorous performance, scalability, security and compliance requirements while using a fully managed platform to perform infrastructure maintenance. [Learn more](#)

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

Name *
azurewebsites.net

Publish * ☒ Code ☐ Docker Container ☐ Static Web App

Runtime stack

Operating System ☒ Linux ☐ Windows

Region

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

3. Fill in the necessary details:

- App name: Unique name for your web app.
- Resource Group: Create a new one or select an existing one.
- Operating System: Choose Linux.
- Runtime stack: Choose your preferred runtime (e.g., Node.js, Python, PHP).

4. Configure other settings as needed and click "Review + create" > "Create".

Home > App Services > Create Web App

Operating System ☒ Linux ☐ Windows

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

Pricing plans
App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. [Learn more](#)

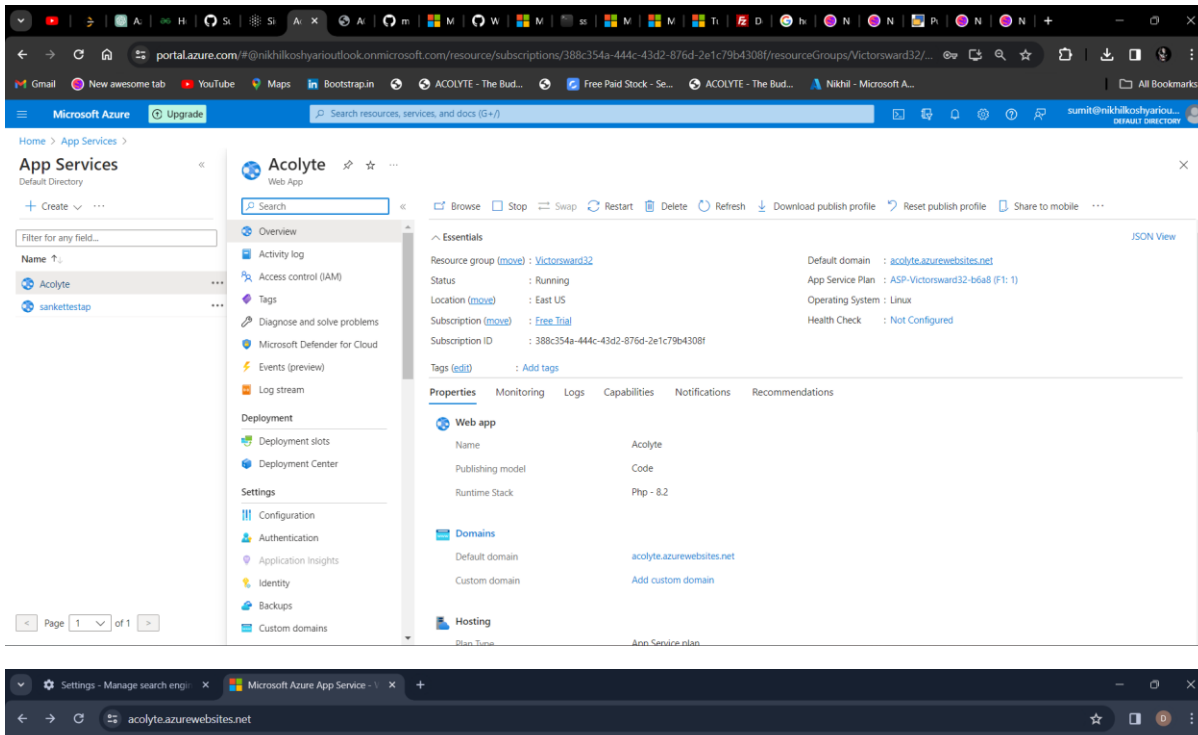
Linux Plan (East US) *
[Create new](#)

Pricing plan **Free F1 (Shared infrastructure)**

Zone redundancy
An App Service plan can be deployed as a zone redundant service in the regions that support it. This is a deployment time only decision. You can't make an App Service plan zone redundant after it has been deployed. [Learn more](#)

Zone redundancy ☐ Enabled: Your App Service plan and the apps in it will be zone redundant. The minimum App Service plan instance count will be three. ☒ Disabled: Your App Service Plan and the apps in it will not be zone redundant. The minimum App Service plan instance count will be one.

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



Your web app is running and waiting for your content

Your web app is live, but we don't have your content yet. If you've already deployed, it could take up to 5 minutes for your content to show up, so come back soon.



Supporting Node.js, Java, .NET and more

Haven't deployed yet?
Use the deployment center to publish code or set up continuous deployment.

Deployment center

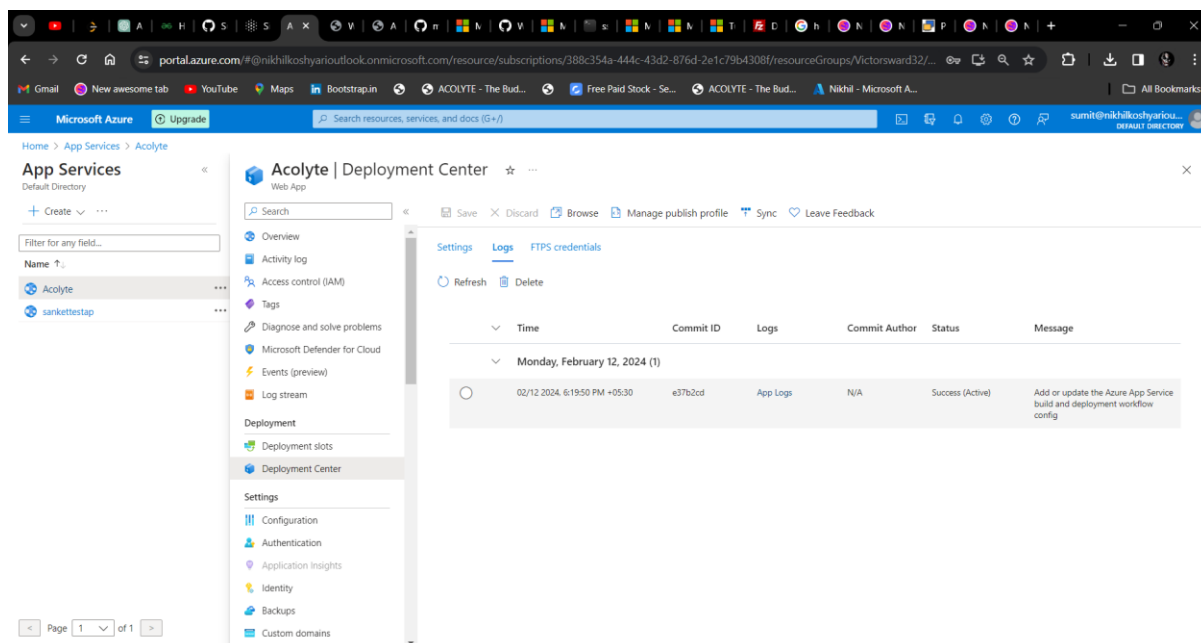
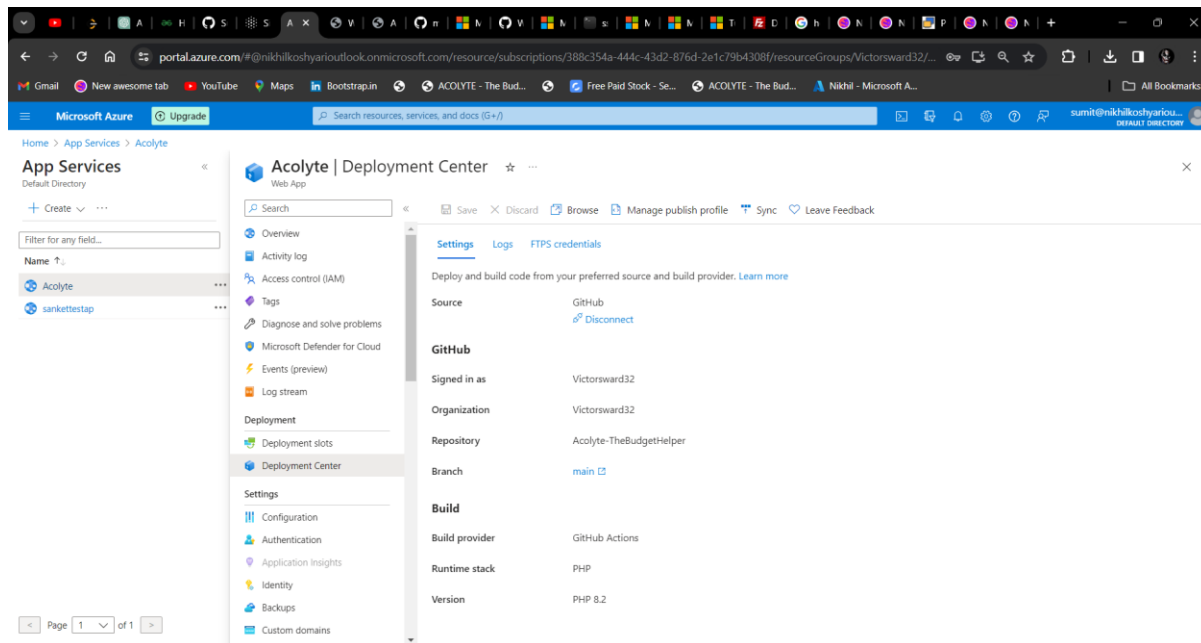
Starting a new web site?
Follow our Quickstart guide to get a web app ready quickly.

Quickstart

Step 2: Connecting Azure Web App to GitHub

- Once your web app is created, navigate to its settings.
- Under "Deployment Center", select GitHub as the source.
- Authorize Azure to access your GitHub account.
- Choose your repository and branch to deploy from.

Sumit Jadhav



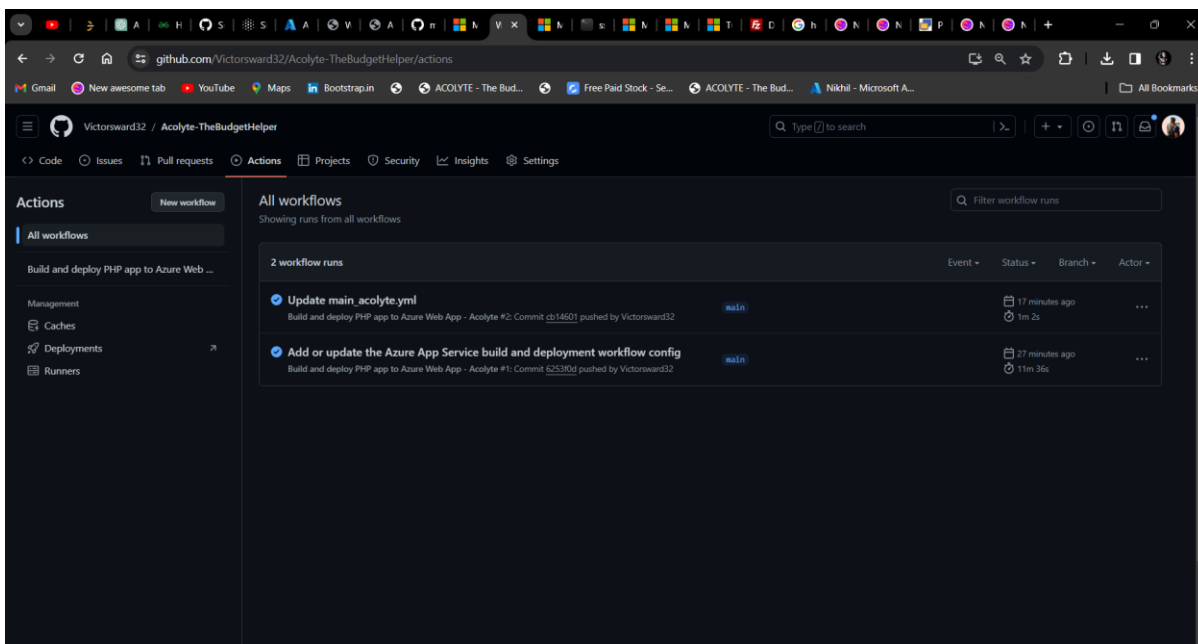
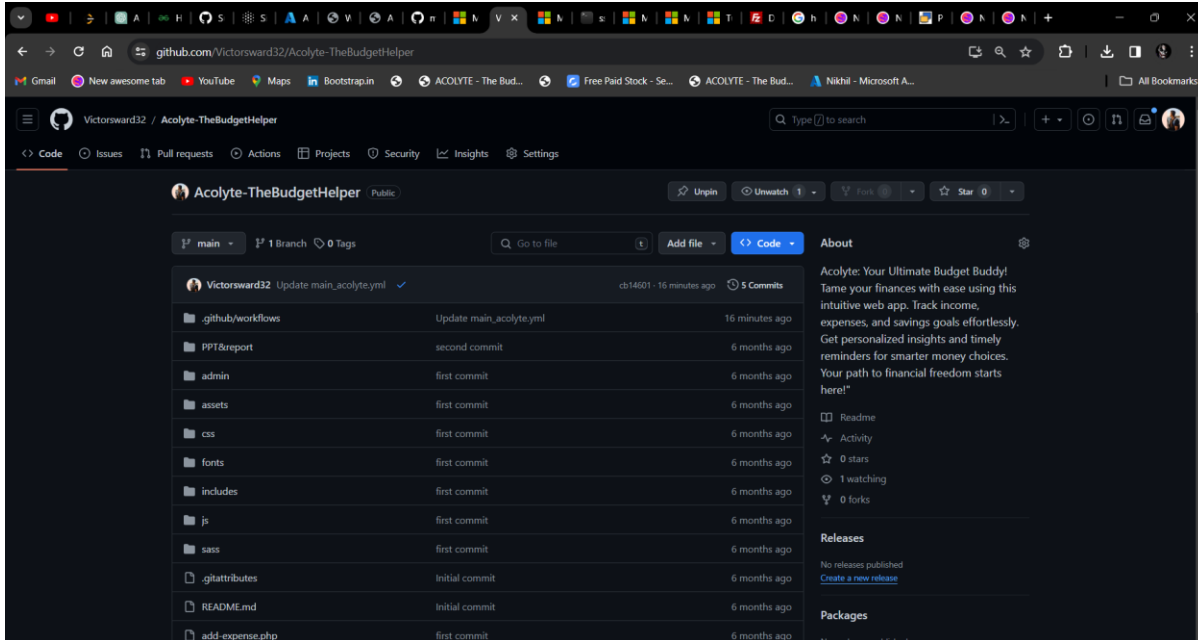
Step 3: Preparing Your Project for Deployment

1. Ensure your project is in a GitHub repository.
2. Make sure your project is properly structured and includes necessary configuration files (e.g., `package.json` for Node.js projects).

Step 4: Deploying Your Project to Azure

Sumit Jadhav

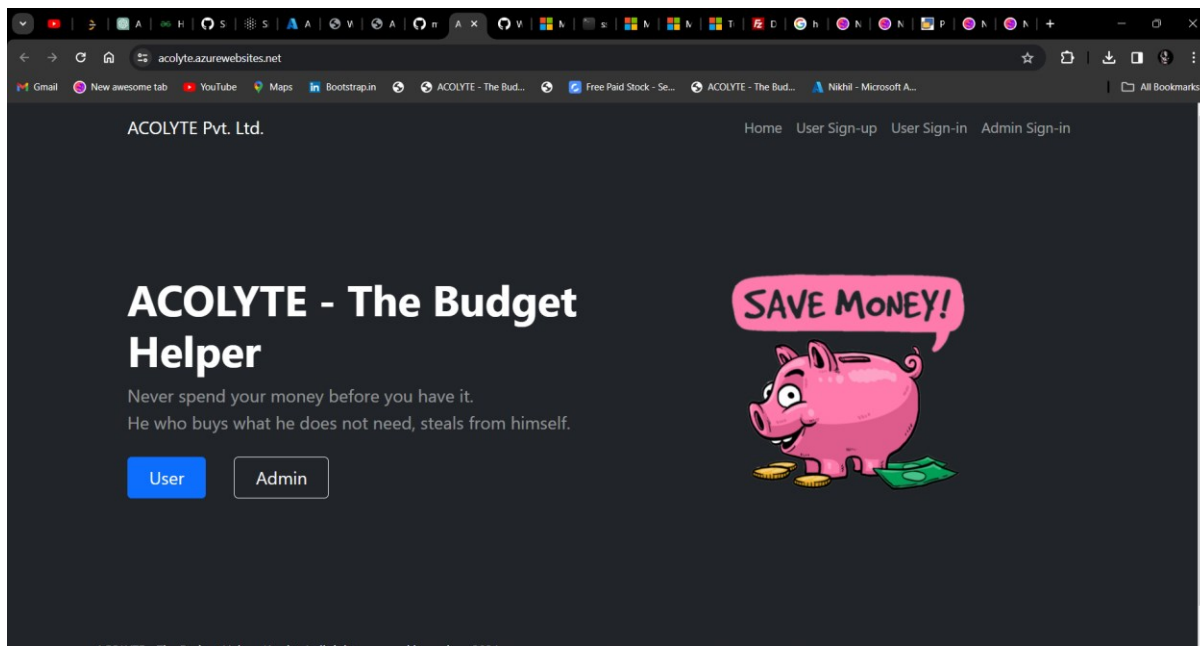
1. Once connected to GitHub, Azure will automatically trigger a deployment when you push changes to the specified branch.
2. Monitor the deployment progress in the Azure Portal.
3. Once deployment is successful, your web app will be updated with the latest changes.



Step 5: Testing Your Deployed Web App

Sumit Jadhav

1. Navigate to the URL of your Azure Web App.
2. Test the functionality of your web app to ensure it's working as expected.
3. If any issues arise, refer to the deployment logs in the Azure Portal for troubleshooting.



Conclusion

Congratulations! You've successfully set up Azure Web Hosting with Linux and deployed your project through GitHub. Remember to explore further resources and documentation for advanced topics.