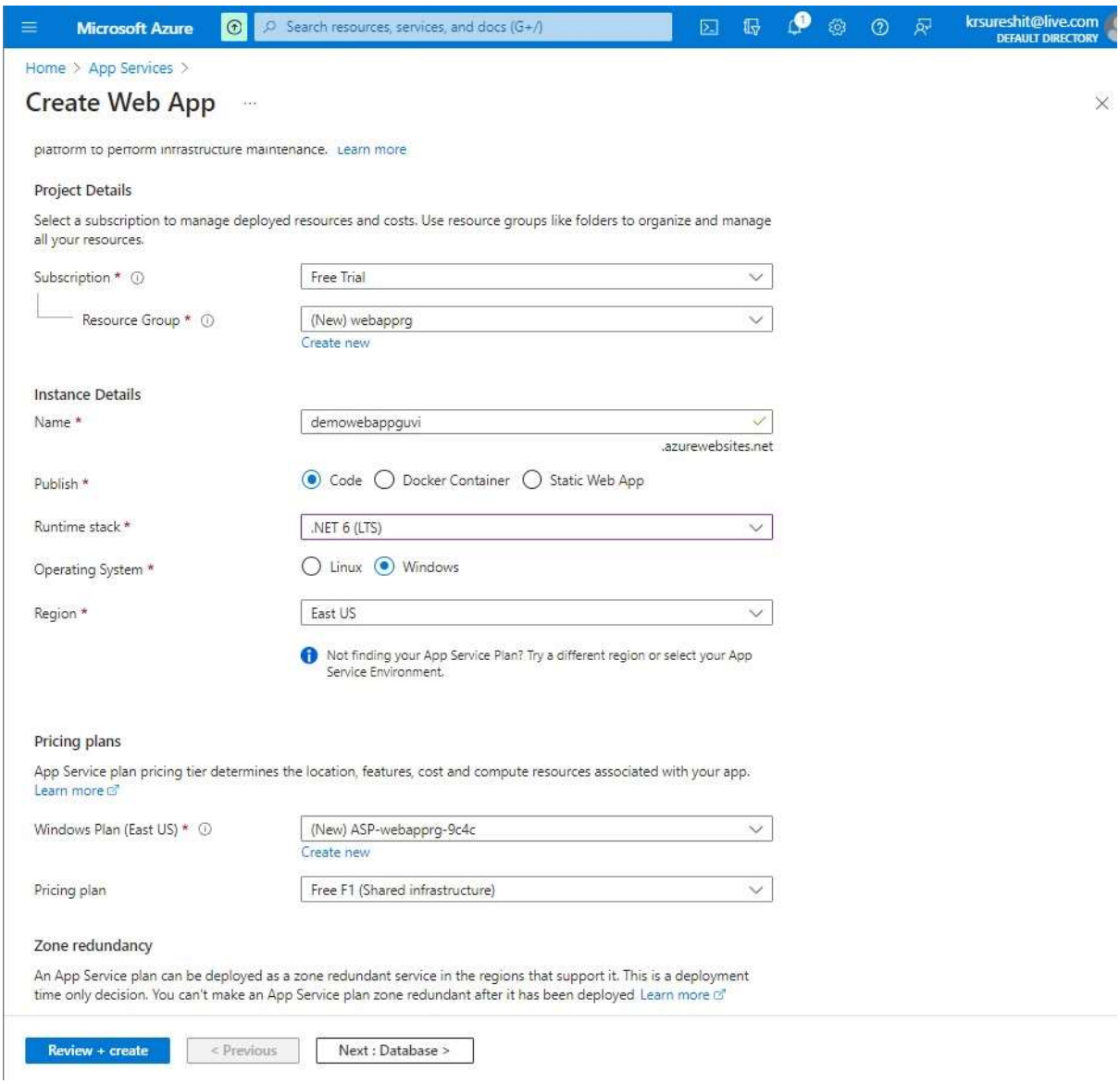
1.Deploy a basic web application using Azure App Service.

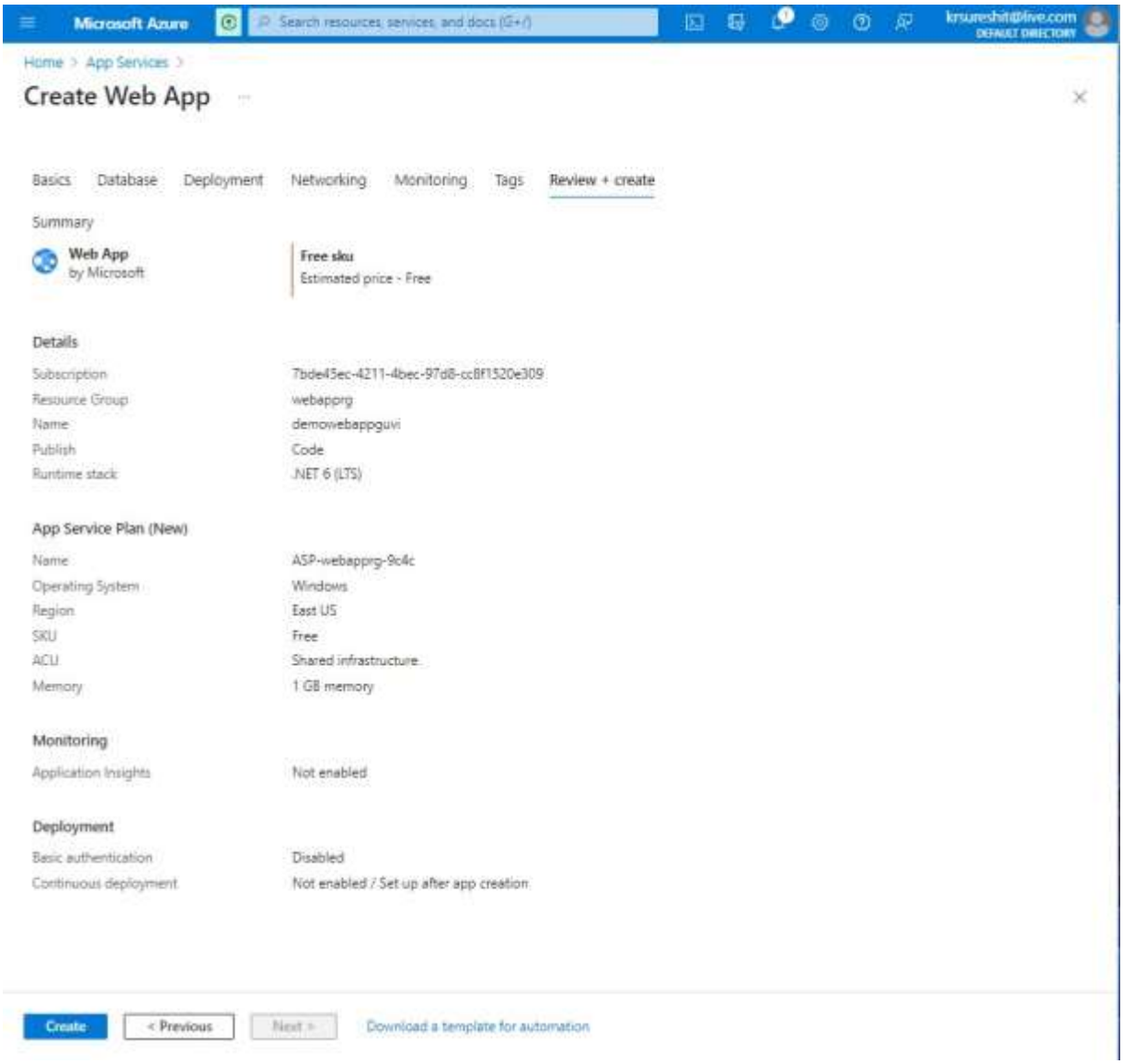
- Choose a unique app name (e.g., "mywebapp").
- Select a runtime stack (e.g., Node.js, .NET).
- Configure other settings (region, operating system, etc.).
- Deploy a simple HTML or sample application.
- Access the web app's URL.

Solution:

1. Create an Azure Web App Using **Azure Portal**
   - Go to the Azure Portal.
   - Click on "Create a resource," then search for " App Services.", then click Create “Web app”
   - Fill in the required details like App name, Subscription, Resource Group, etc.
   - Choose the appropriate OS (Windows/Linux).

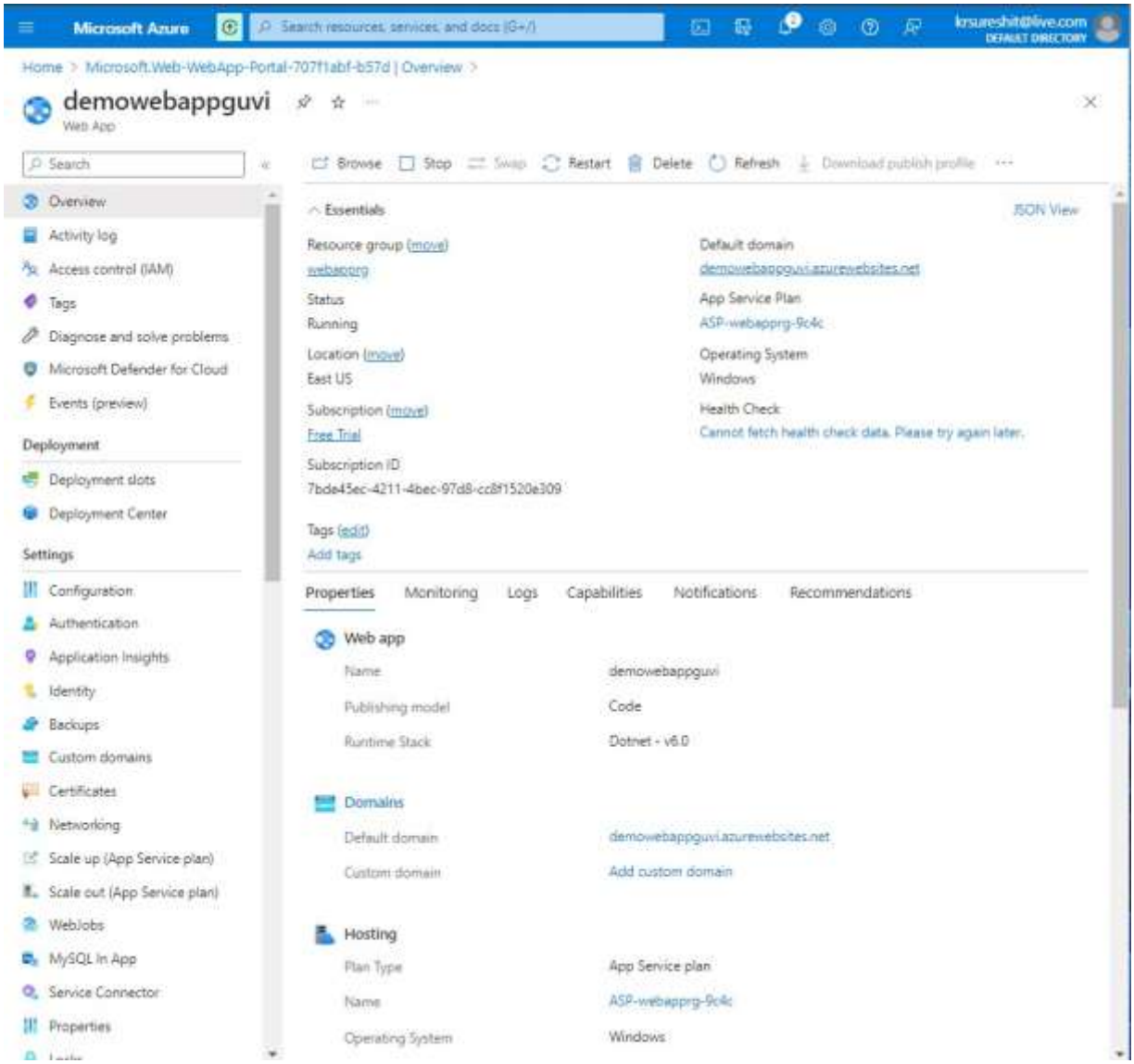


- Configure other settings as needed and click "Review + create" and then "Create."

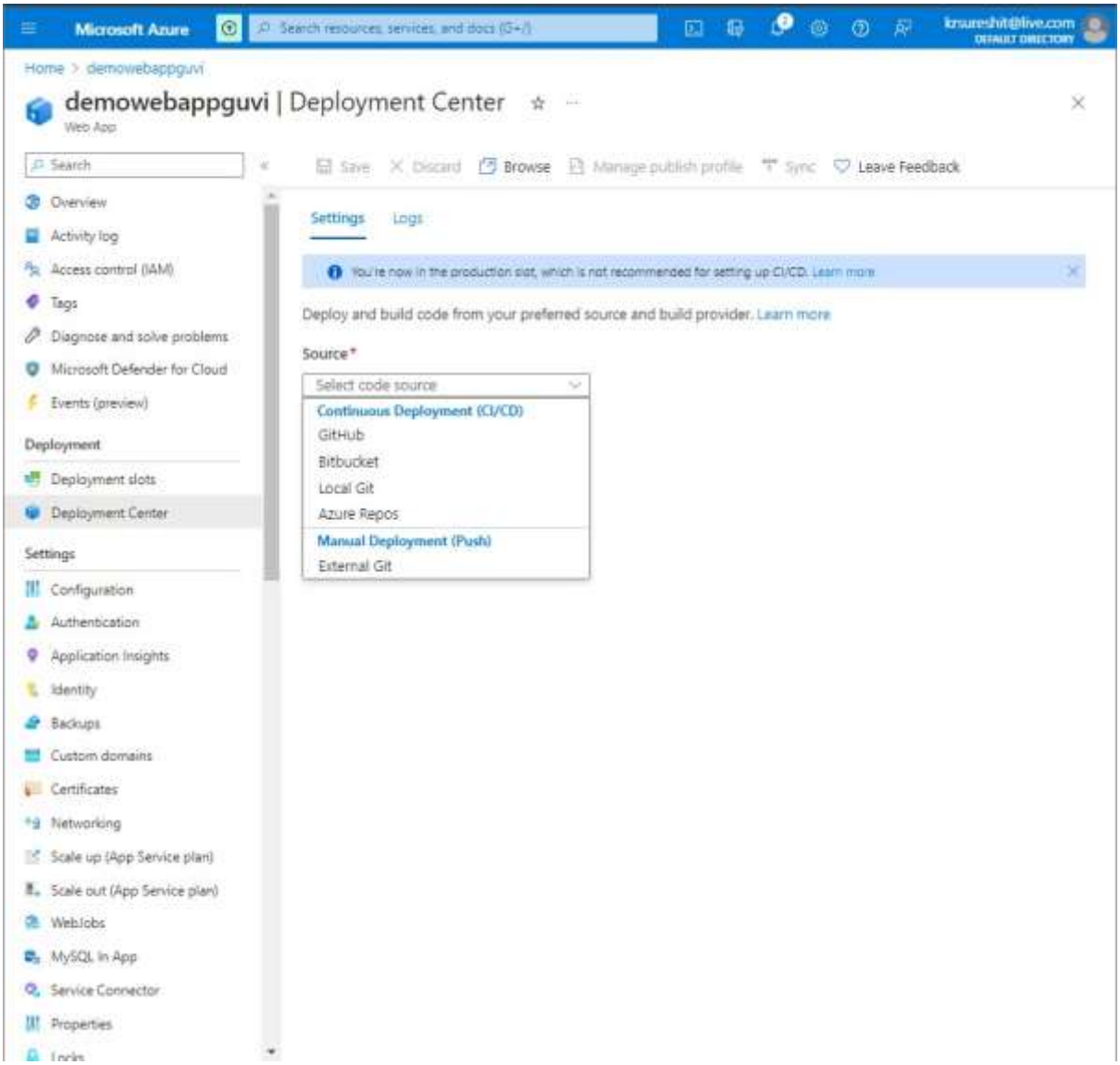


2. Configure Deployment Source using **GitHub Deployment**

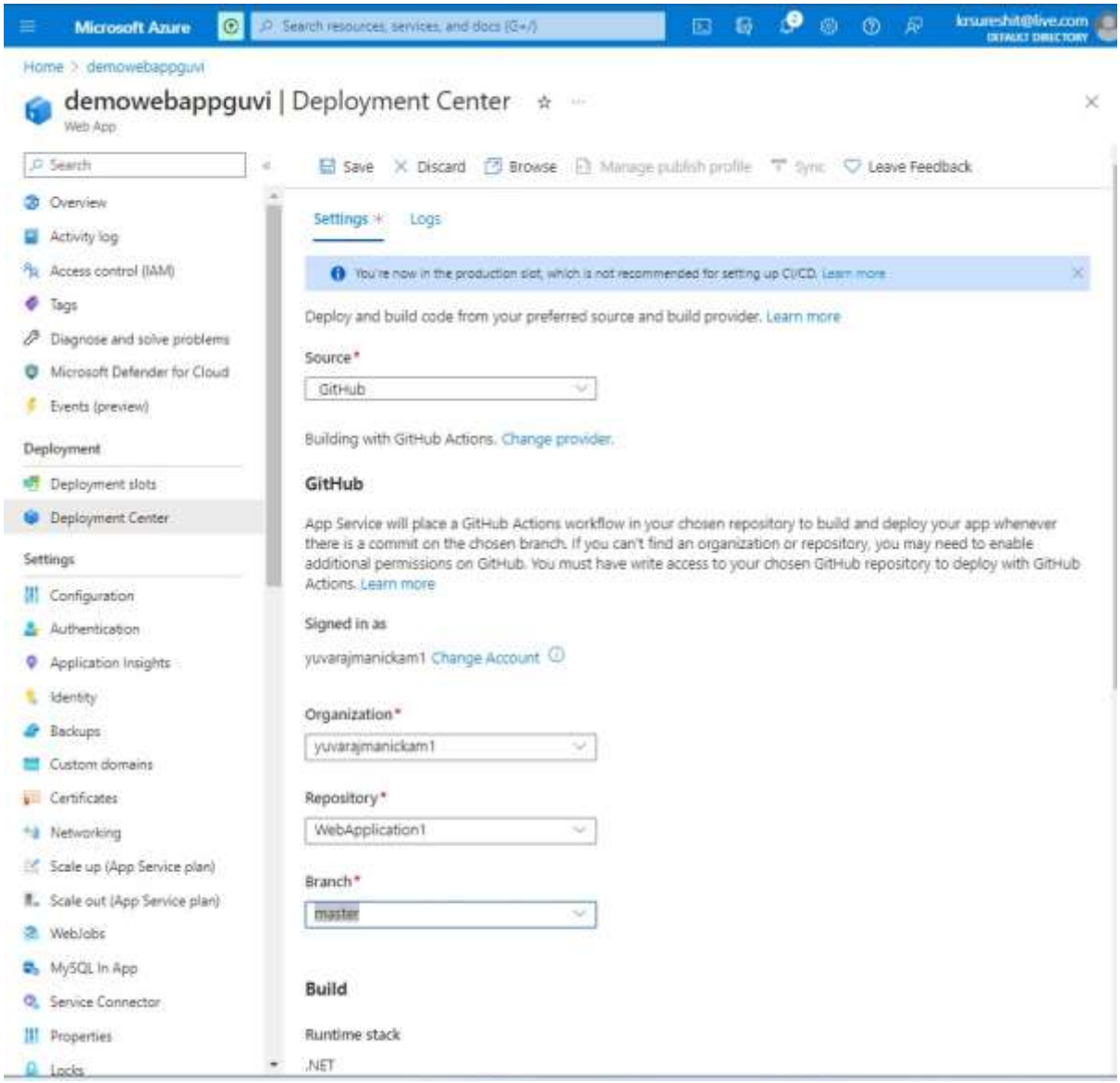
- In the Azure Portal, go to your Web App.



- Under "Deployment Center," choose GitHub as the source.

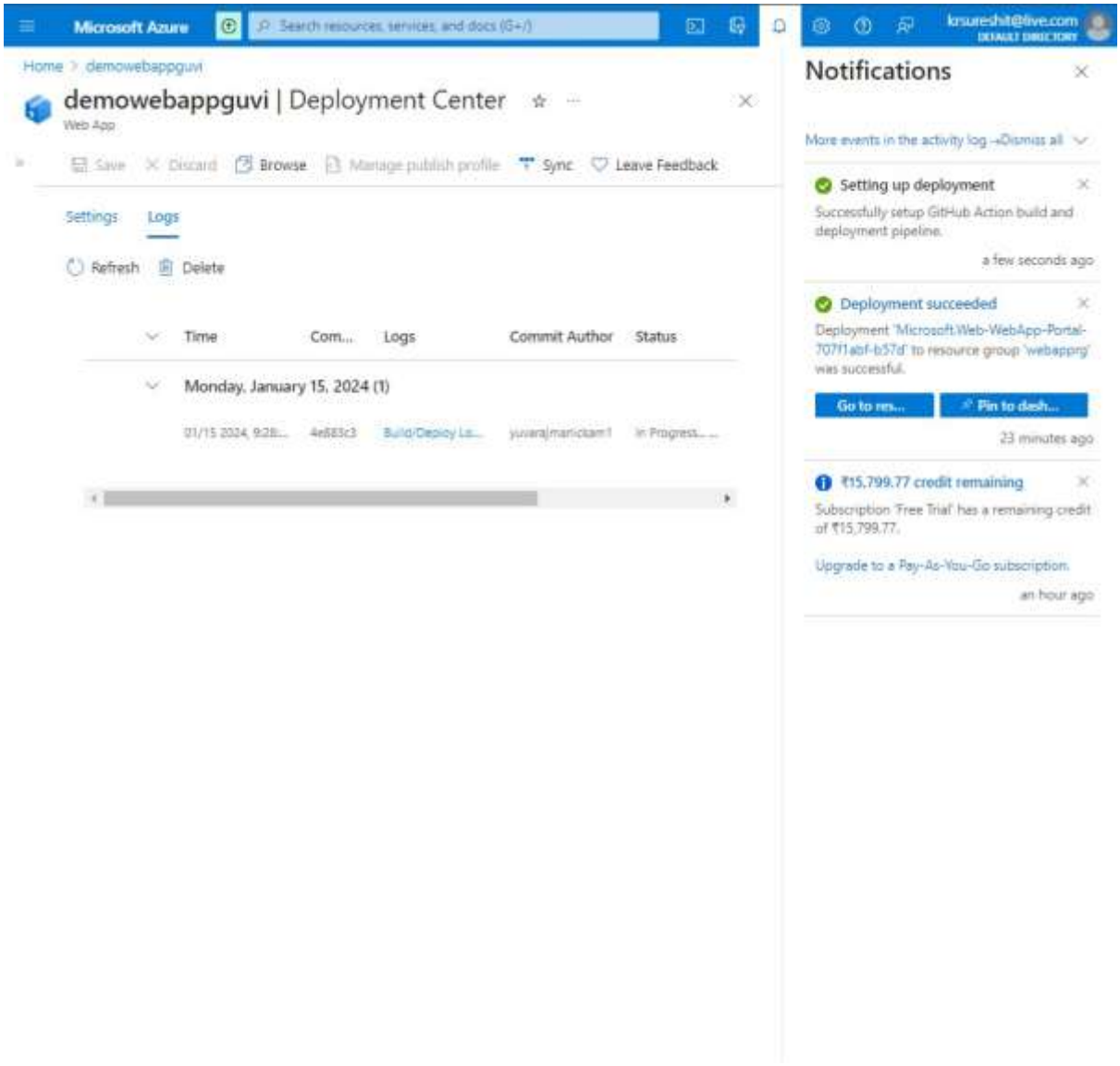


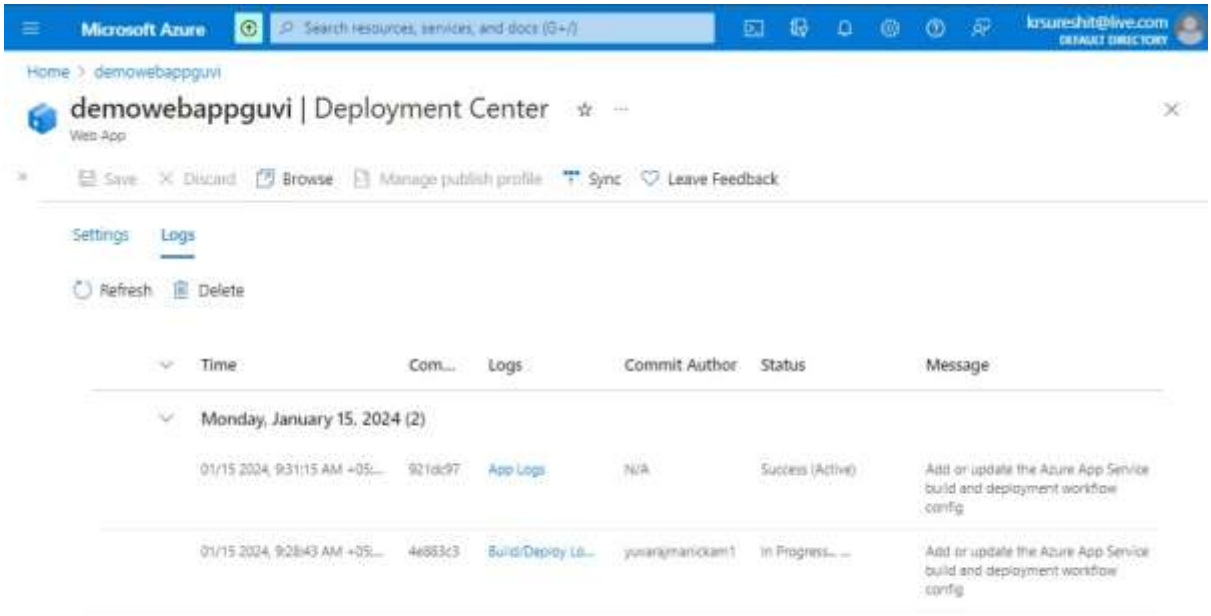
- Authenticate and select your repository



.

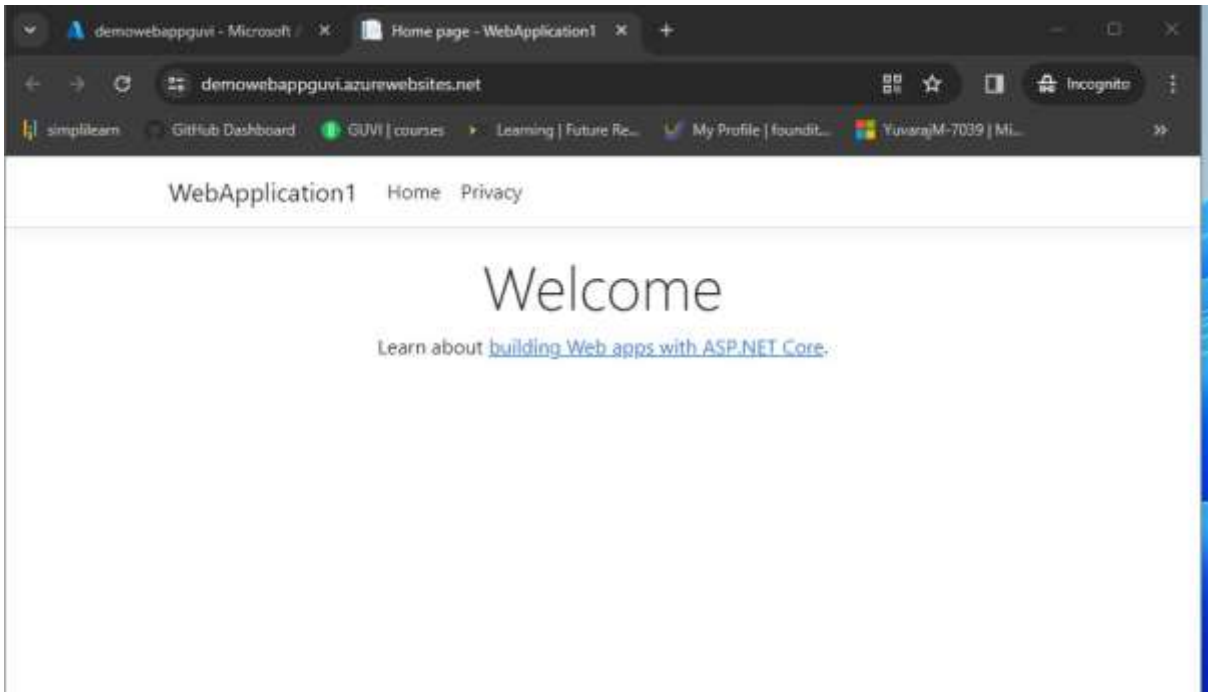
- Configure settings and trigger a deployment.


Microsoft Azure
Search resources, services, and docs (G+/)
krsureshit@live.com
DEFAULT DIRECTORY
Home > demowebappguvi
demowebappguvi | Deployment Center
Web App
Save
Discard
Browse
Manage publish profile
Sync
Leave Feedback
Settings
Logs
Refresh
Delete
Time
Com...
Logs
Commit Author
Status
Monday, January 15, 2024 (1)
01/15 2024, 9:28...
4e883c3
Build/Deploy Lo...
yuvaraj/manickam1
In Progress...
Notifications
More events in the activity log → Dismiss all
Setting up deployment
Successfully setup GitHub Action build and deployment pipeline.
a few seconds ago
Deployment succeeded
Deployment 'Microsoft.Web-WebApp-Portal-707f1abf-b57d' to resource group 'webapprg' was successful.
Go to res...
Pin to dash...
23 minutes ago
₹15,799.77 credit remaining
Subscription 'Free Trial' has a remaining credit of ₹15,799.77.
Upgrade to a Pay-As-You-Go subscription.
an hour ago



3. Verify Deployment:

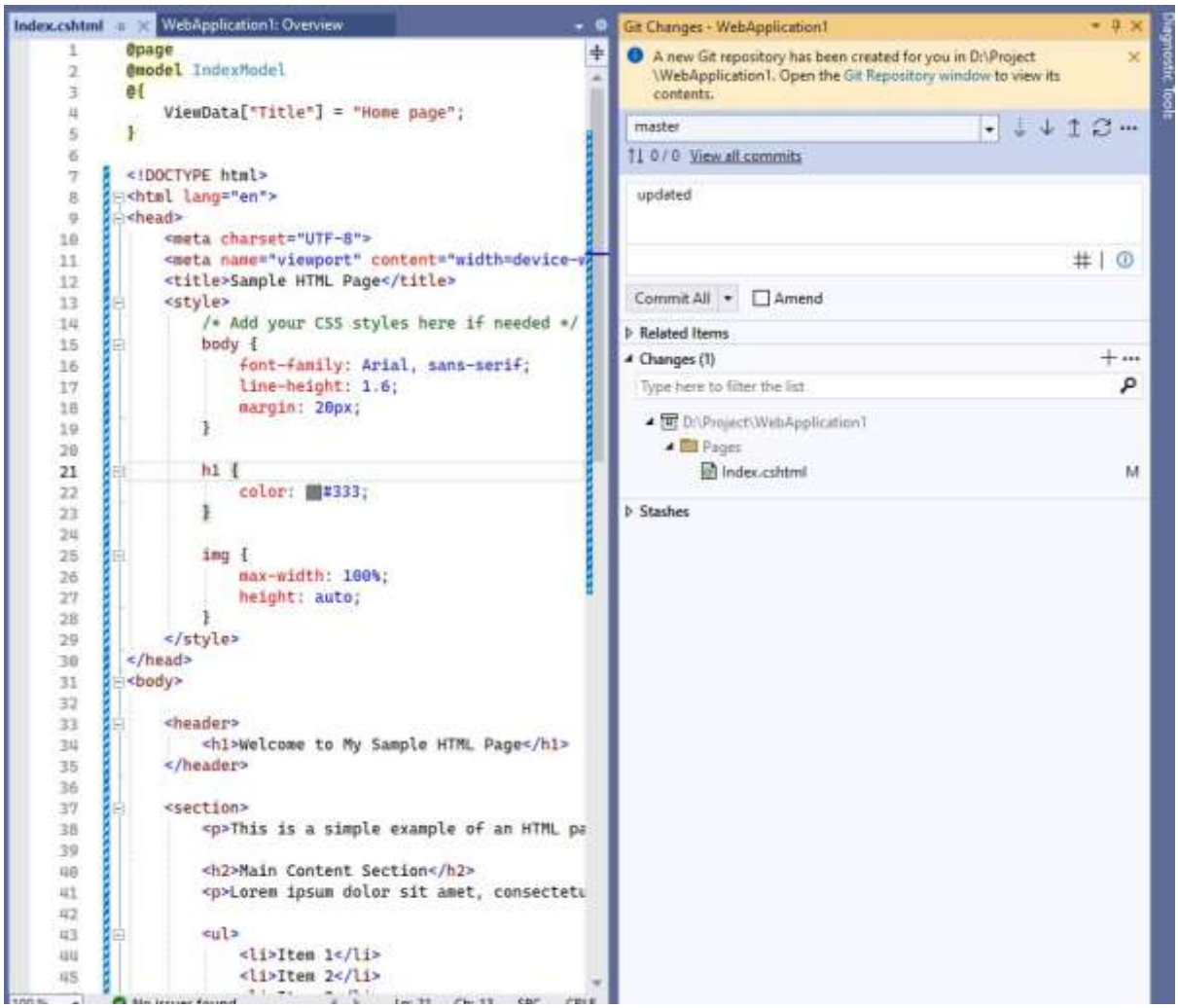
- Once the deployment is complete, you can access your web app using the URL https://demowebappguvi.azurewebsites.net/ provided by Azure.



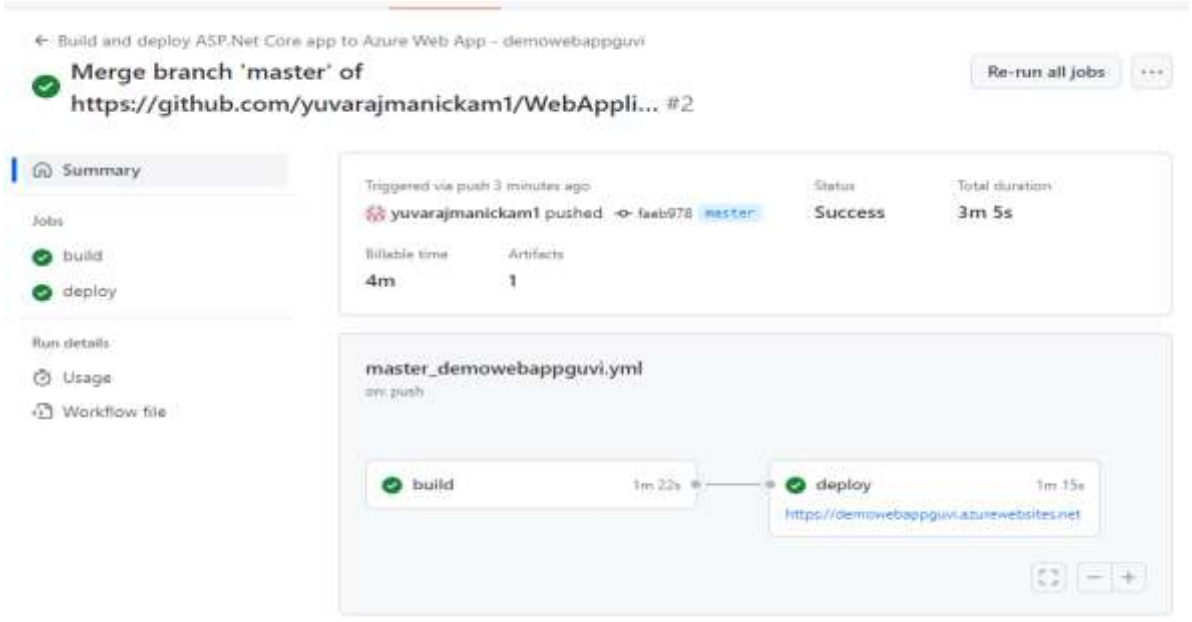
4. Continuous Deployment (Optional):

- Configure continuous deployment if you want automatic deployments when you push changes to your repository.

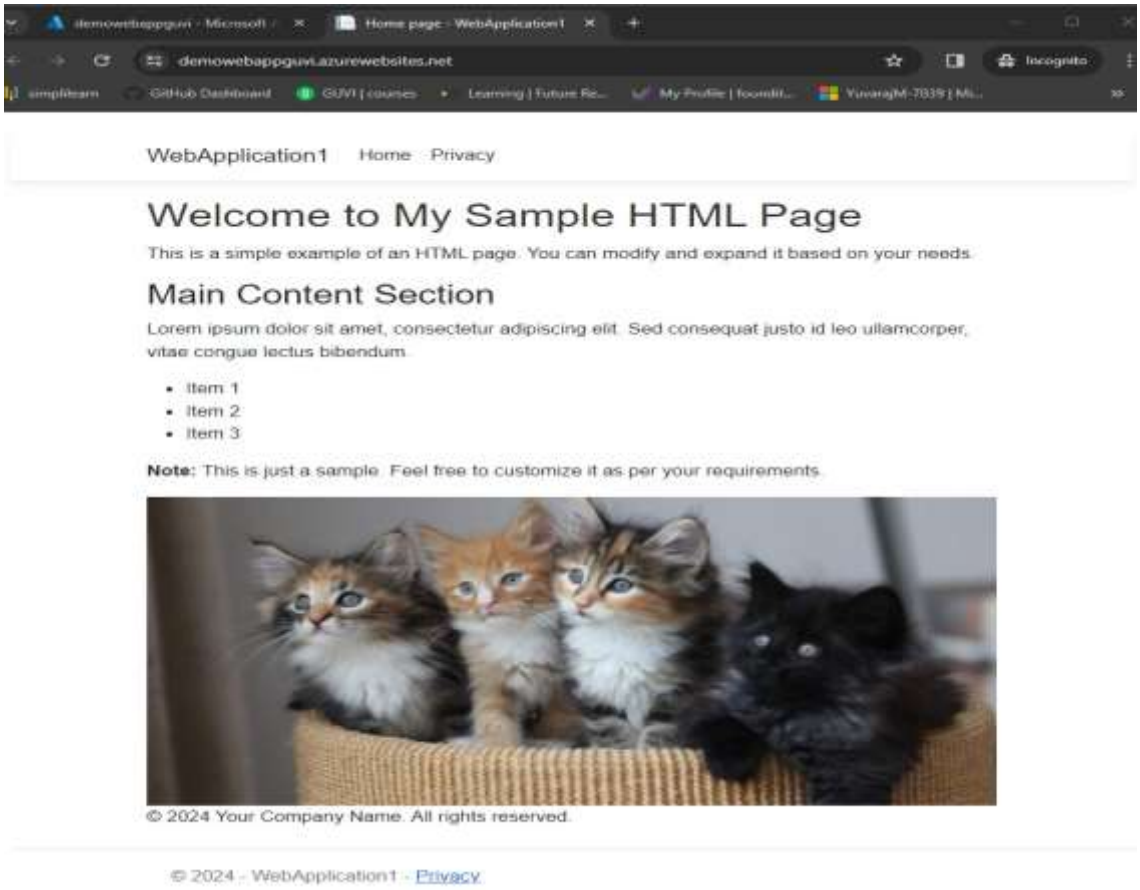
- Make changes to the GitHub repository, commit, and push the changes.



- It will trigger the deployment



Once the deployment is complete, you can access your web app using the URL



What are the benefits of using deployment slots, and how can they be utilized in a real-world scenario?

Azure deployment slots are a feature of Azure App Service that provides a way to host different versions of your web app or API in the same Azure App Service instance. Each deployment slot is like a separate instance of your app with its own host name and settings. Here are some benefits of using deployment slots and how they can be utilized in a real-world scenario:

Benefits of Deployment Slots:

1. **Testing in Production-like Environment:**
   Deployment slots allow you to create a production-like environment for testing without affecting the live production environment. This helps in identifying and addressing issues that may arise only in a production setting.

2. **Staging Environment:**

Deployment slots are commonly used as staging environments where you can deploy a new version of your application and validate it before swapping it into production. This ensures a smoother release process.

3. **Zero Downtime Deployment:**
   By deploying a new version of your app to a staging slot and swapping it with the production slot, you can achieve zero-downtime deployments. Users are seamlessly transitioned to the new version without experiencing downtime.
4. **Rollback Capability:**
   If an issue is identified in the new version after swapping, it's easy to roll back to the previous version by swapping again or promoting the previous version from another slot.
5. **Performance Testing:**
   Deployment slots can be used for performance testing. You can deploy a version of your application to a slot and test its performance under load before promoting it to production.
6. **A/B Testing:**
   Slots can be utilized for A/B testing by deploying different versions of your application to different slots and directing a portion of your traffic to each version. This helps in evaluating the performance and user experience of different features.

Utilizing Deployment Slots in Real-world Scenarios:

1. **Staging and Testing:**
   Use a deployment slot as a staging environment to validate changes and updates before promoting them to the production slot.

2. **Blue-Green Deployments:**
   Implement a blue-green deployment strategy by having two deployment slots - blue and green. Deploy the new version to the green slot, test it thoroughly, and then swap the slots for the new version to go live.

3. **Feature Branch Deployment:**
   If you are working with feature branches in your source control, deploy each feature branch to a separate slot for testing before merging it into the main branch and deploying to production.

4. **Load Testing:**
   Conduct load testing in a deployment slot to ensure that your application can handle increased traffic and load without impacting the production environment.

5. **Data Migration:**
   Use a deployment slot to perform data migration tasks or database schema changes without affecting the live application.

   By leveraging deployment slots, you can enhance the reliability, availability, and efficiency of your deployment processes while minimizing the impact on end-users.