Logo

Description automatically generated 

**Deployment and Operations Guide (Runbook)**

University of Maryland Global Campus

SWEN 670 – Team B

Spring Semester

Version 1.0 

March 25, 2023

|  |  |  |
| --- | --- | --- |
| **Version** | **Issue Date** | **Changes** |
| 0.1 | 3/22/2023 | Initial Version |
| 1.0 | 3/25/2023 | Milestone 3 Submission |

**Contents**

[1. Introduction 4](#_Toc580453984)

[1.1. Purpose 4](#_Toc695489442)

[1.2. Intended Audience 4](#_Toc1477153110)

[1.3. Technical Project Stakeholders 4](#_Toc1713715530)

[1.4. Project Documents 5](#_Toc338150406)

[2. Development Team Workflow 5](#_Toc848546337)

[2.1. Description of Deployment 5](#_Toc1808813890)

[2.2. Points of Contact 6](#_Toc428624148)

[3. Preparing for Use 7](#_Toc2030995937)

[4. Deployment, Testing, and Release 8](#_Toc1741491842)

[4.1 Install the “Azure Tools extension” for Visual Studio Code. 8](#_Toc913085149)

[4.2 Create Azure account (free) from Visual Studio Code 8](#_Toc1895134242)

[4.3 Create Azure Web App Service from Visual Studio Code 12](#_Toc1985061983)

[4.4 Deploy the application code to Azure from Visual Studio Code 13](#_Toc1435729143)

[4.5 Sharing Access to Server 14](#_Toc581425960)

[5. Troubleshooting 17](#_Toc425173597)

[5.1 Deployment zip takes too long and fails 18](#_Toc1504004164)

[5.2 Unable to create tours 21](#_Toc1742123682)

# Introduction

This deployment and operations guide is for the server-side deliverables for the ViroTour application delivered by Team B from the Spring 2023 SWEN 670 Capstone course at the University of Maryland Global Campus. This document is designed to help with the deployment and maintenance of the ViroTour application in a production environment.

## Purpose

The purpose of this guide is to provide a comprehensive resource for engineering professionals who are interested in deploying and operating the ViroTour application. This guide will provide step-by-step instructions and help with understanding the software requirements for deploying the application, as well as providing guidance on how to configure, test, release, and troubleshoot any issues.

## Intended Audience

The intended audience for Team B’s deployment and operations guide are:

* System Administrators, DevOps professionals, other IT professionals who are interested in the application and/or have chosen to continue to develop and maintain the application.
* Technical Project Stakeholders of the Spring 2023 project
* Future Capstone students for reference

## Technical Project Stakeholders

|  |  |
| --- | --- |
| Stakeholder Name | Project Role |
| Dr. Mir Assadullah | Client/Professor |
| Roy Gordon | Project Mentor |
| Robert Wilson | DevSecOps Mentor |
| Ivelin Tchangalov | Project Manager (PM) |
| KC Harden | Product Owner (PO) |
| Alex Armel Wabo Tebu | Business Analyst (BA) |
| Hang Wang | Sr. Software Engineer (Dev Lead) |
| Nancy Lay | Test Manager (Test Lead) |
| Kelvin Huynh | Jr. Software Engineer (Dev) |
| Melika Shahani | Jr. Software Engineer (Dev) |
| Jean Pita Diomi Kazadi | Jr. Software Engineer (Dev) |
| Jah-wilson Teeba | Jr. Software Engineer (Dev) |
| Ronald Milligan | Jr. Software Engineer (Dev) |
| Ian Fischer | Jr. Software Engineer (Dev) |

**Table 1.3 – Stakeholders and Respective Roles.**

## Project Documents

Here is the list of documents in the software documentation package

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Document** | **Version** | **Date** |
| 1 | Project Management Plan (PMP) | 3.0 | 3/25/2023 |
| 2 | Software Requirements Specification (SRS) | 2.0 | 3/25/2023 |
| 3 | Technical Design Document (TDD) | 1.0 | 2/12/2023 |
| 4 | Software Test Plan (STP) | 1.0 | 2/12/2023 |
| 5 | Programmers Guide (PG) | 1.0 | 3/25/2023 |
| 6 | Deployment and Operations (DevOps/Runbook) | 1.0 | 3/25/2023 |
| 7 | User Guide (UG) | 1.0 |  |
| 8 | Test Report (TR) | 1.0 |  |

**Table 1.4 - Project Documents.**

# Development Team Workflow

## Description of Deployment

The ViroTour Server Processing team utilized GitHub to manage all code collaboration and create our final working product. The team created their own GitHub profiles and were given access to the spring2023 repository by Robert Wilson.

|  |  |
| --- | --- |
| Name | GitHub User |
| KC Harden | farkace |
| Ivelin Tchangalov | tchangalov |
| Alex Armel Wabo Tebu | alexarmel |
| Hang Wang | spore42 |
| Nancy Lay | Nhomsombath |
| Kelvin Huynh | khuynh16 |
| Melika Shahani | melikashahani |
| Jean Pita Diomi Kazadi | jeanpitad |
| Jah-wilson Teeba | jwteeba |
| Ronald Milligan | SWIFPAC |
| Ian Fischer | umucipf |

**Table 2.1 - Team Members and GitHub Account.**

Developers start developing by cloning the spring2023 repository from GitHub (https://github.com/umgc/spring2023.git). After the repository is downloaded the spring2023 repository will be on the 'master' branch. The developer will checkout a new branch and create their own features from there. When they are ready to merge, a Pull Request must be created to merge the code to the “development” branch, and finally into the master branch. The final release(s) will be created from the master branch and will be tagged and pushed as follows.

* + Verify the commit hash on master matches the hash on your local branch.

$git log

* + Create the tag

$git tag [tag number]

* + Push the tag

$git push origin [tag number]

## Points of Contact

|  |  |
| --- | --- |
| Name | Role |
| Dr. Mir Assadullah | Professor, Client |
| Roy Gordon | Project Mentor |
| Robert Wilson | Project Mentor |
| Khoa Nguyen | Project Manager, Scrum Master - Frontend |
| Jacob Lynn | Product Owner - Frontend |
| Tilahun Abreha | Business Analyst - Frontend |
| Viet Nguyen | Software Engineer Senior - Frontend |
| Christian Dovel | Software Engineer - Frontend |
| Fedor Menchukov | Software Engineer - Frontend |
| Jude Ibe | Software Engineer - Frontend |
| Nicholas Platt | Software Engineer - Frontend |
| Samson Alemneh | Software Engineer - Frontend |
| Jeffrey Welch | Software Engineer - Frontend |
| Shawn Kagwa | Software Engineer - Frontend |
| KC Harden | Product Owner - Backend |
| Ivelin Tchangalov | Project Manager, Scrum Master - Backend |
| Alex Armel Wabo Tebu | Business Analyst - Backend |
| Hang Wang | Software Engineer Senior - Backend |
| Nancy Lay | Test Manager - Backend |
| Kelvin Huynh | Software Engineer - Backend |
| Melika Shahani | Software Engineer - Backend |
| Jean Pita Diomi Kazadi | Software Engineer - Backend |
| Jah-wilson Teeba | Software Engineer - Backend |
| Ronald Milligan | Software Engineer - Backend |
| Ian Fischer | Software Engineer - Backend |

**Table 2.2 - All Stakeholders and Role.**

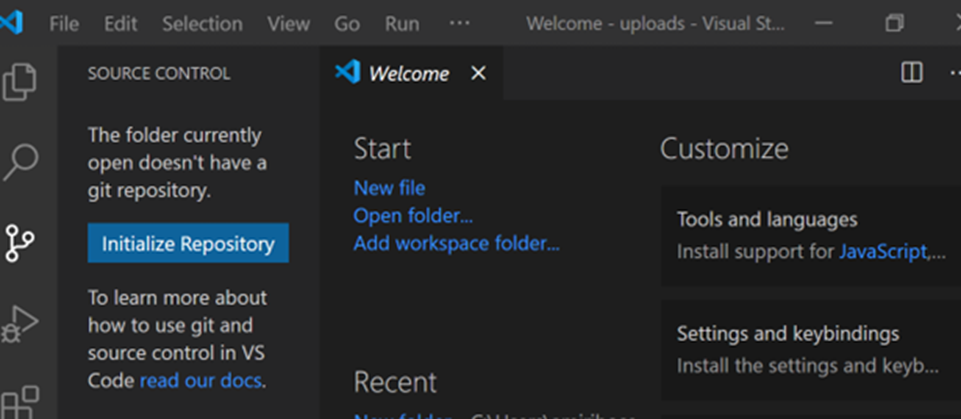
# Preparing for Use

This will cover the software installation required for deploying it to the cloud.

**3.1 Cloning GitHub Repository**

In this section, we must first clone the project from the GitHub repository and put it in our system. Assuming that our repository is private, it should be cloned remotely and opened in VSCode. To do this, we take the following steps:

Open VSCode and “Open folder...” from the GitHub path: “virotour\_local/flask” as the root folder, such that the “requirements.txt” is visible on the root level.



**Figure 3.1 - Visual Studio Code Initial Screen.**

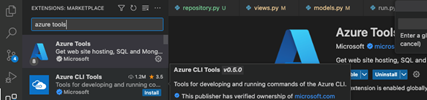
In the next step, it is time to zip the files and transfer them to the server. At this stage, it is enough to prepare our server and transfer the information to it. Our priority is Linux servers that have cPanel.

# Deployment, Testing, and Release

## 4.1 Install the “Azure Tools extension” for Visual Studio Code.

Go to section” Install the Azure Tools Extension Pack” at:

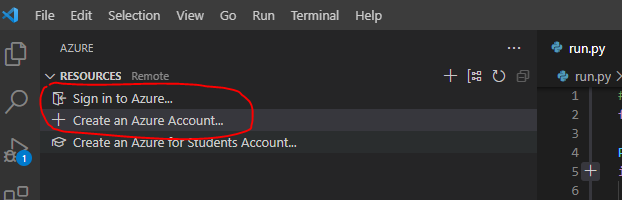
<https://learn.microsoft.com/en-us/dotnet/azure/configure-vs-code>



**Figure 4.1 - Search for Extension.**

## 4.2 Create Azure account (free) from Visual Studio Code

From the Azure Extension Tools, click “Sign in to Azure, or Create an Azure Account.”

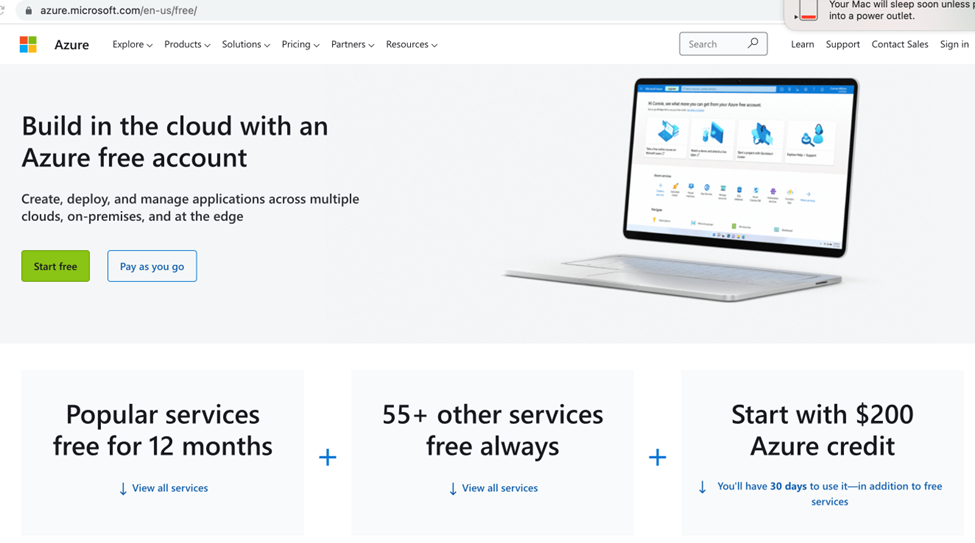


**Figure 4.2.1 - Settings from Azure Extension Tools.**

From the <https://azure.microsoft.com/en-us/free/>

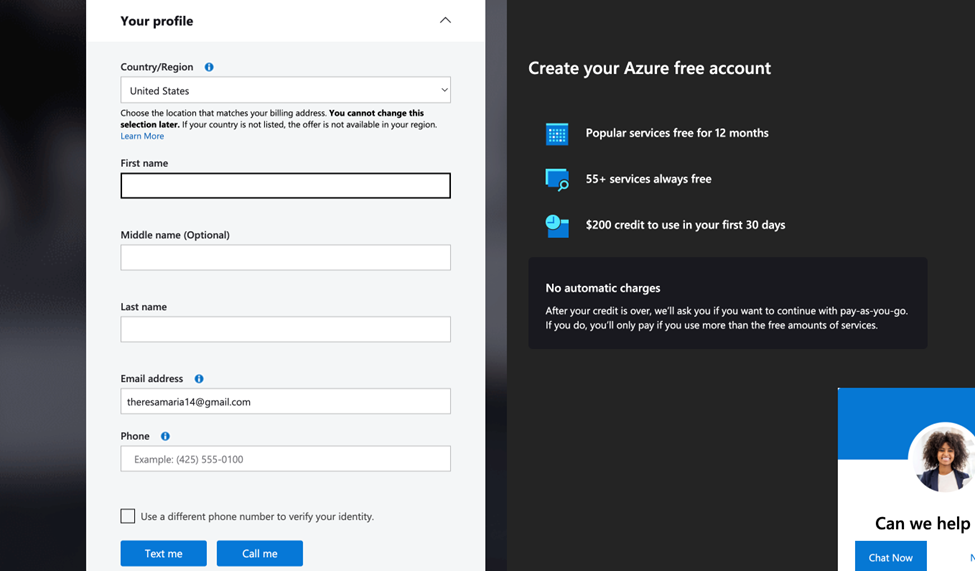
1. Create a free account.

Click the green button “Start free,” and follow the steps.



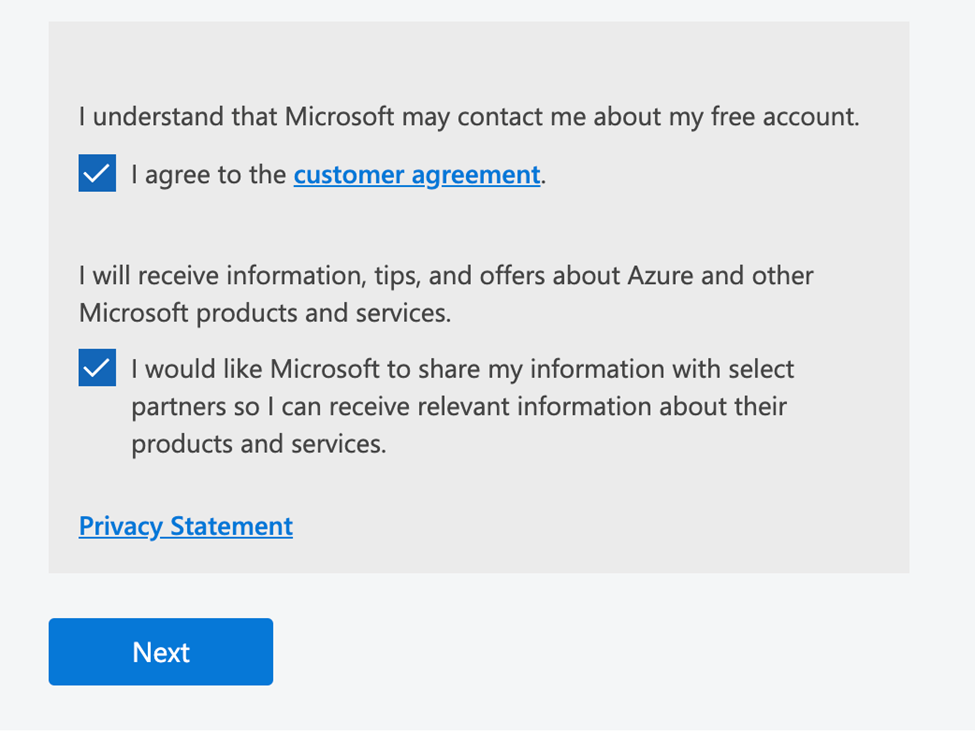
**Figure 4.2.2 - Microsoft Azure Landing Page.**

1. Choose “use another account” to create a new login.
2. Fill the below form.



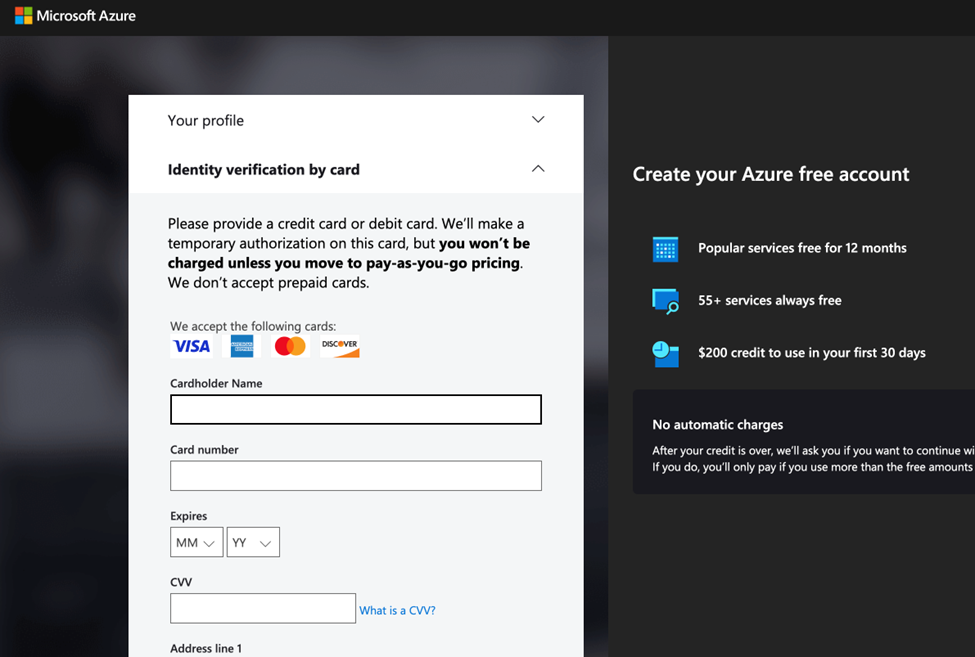
**Figure 4.2.3 - Create Account Form.**

1. Fill in the above form, and make sure you provide a working phone to verify the identity via text!
2. Click “Text me” to receive a verification code.
3. Provide the “Verification Code” and the blue button “Verify code.”
4. The blue “Next” button will be activated at the bottom of the form.



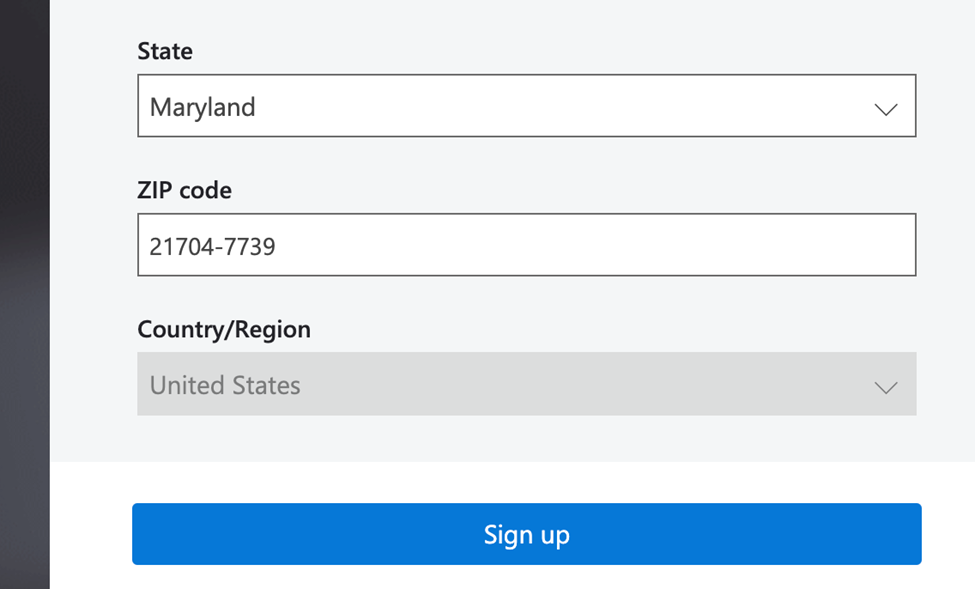
**Figure 4.2.4 - Customer Agreement.**

1. Click “Next” to see the payment screen below.



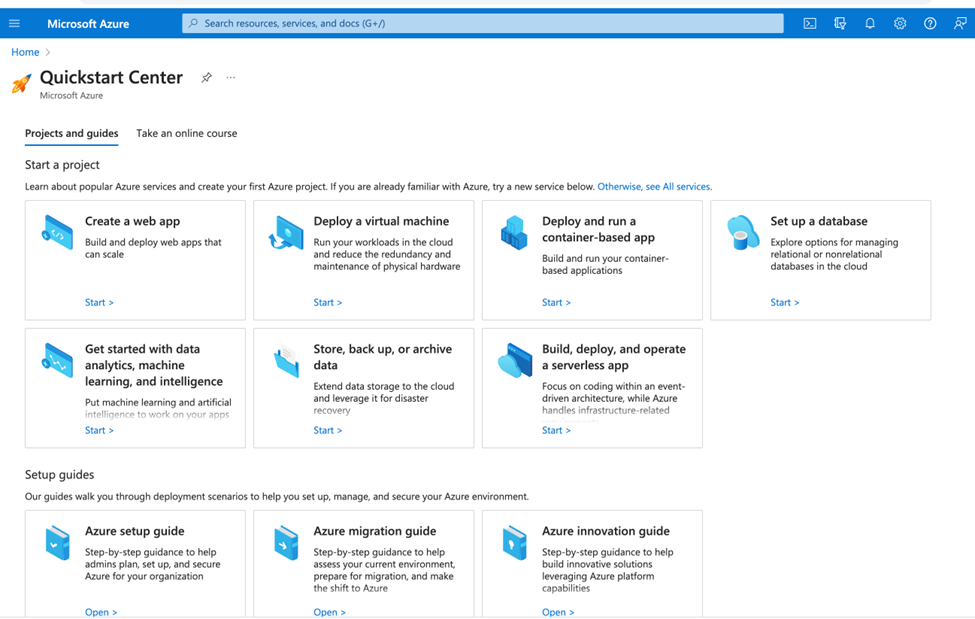
**Figure 4.2.5 - Credit Card Verification.**

1. Fill in the card info. (Note: the card will not be charged unless you move to pay-as-you-go pricing).
2. Click “Sign up.”



**Figure 4.2.6 - Settings from Azure Extension Tools.**

1. Once done, the “Quick Start” page will appear.



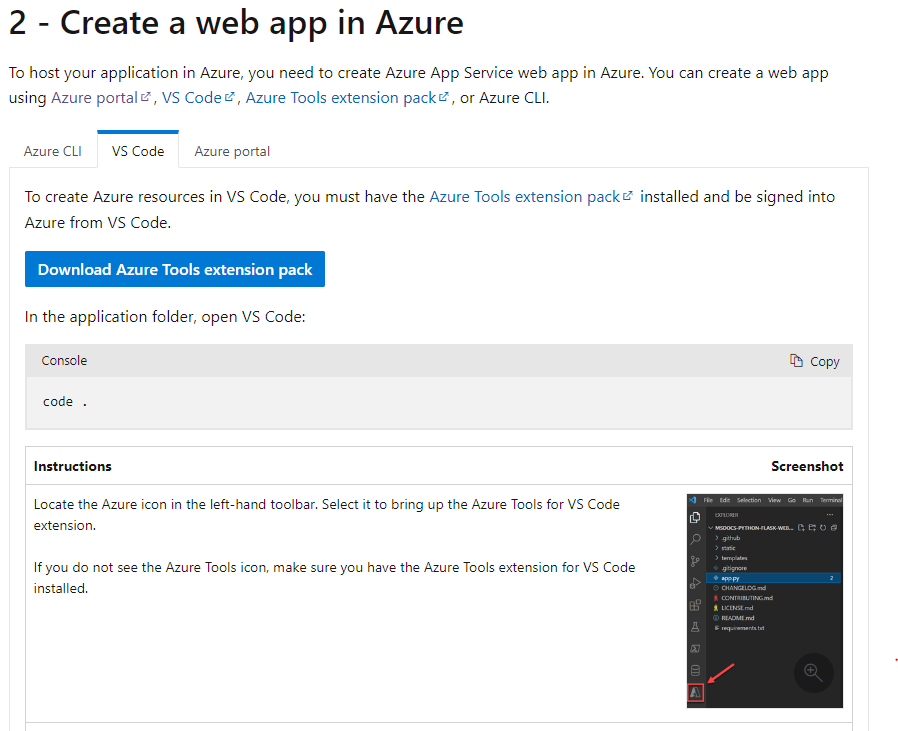
**Figure 4.2.7 - Quickstart Center.**

## 4.3 Create Azure Web App Service from Visual Studio Code

1. For VSCode instructions, please click the below link:

<https://learn.microsoft.com/en-us/azure/app-service/quickstart-python?tabs=flask%2Cwindows%2Cvscode-aztools%2Cvscode-deploy%2Cdeploy-instructions-azportal%2Cterminal-bash%2Cdeploy-instructions-zip-azcli>

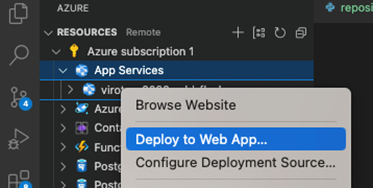
Go to (**2 - Create a web app in Azure** and choose the “**Visual Studio Code**” tab).



**Figure 4.3 - Azure Web App Creation via VSCode.**

## 4.4 Deploy the application code to Azure from Visual Studio Code

1. To deploy into the created “Web App Service,” right click the service as below:



**Figure 4.4 - Application Deployment.**

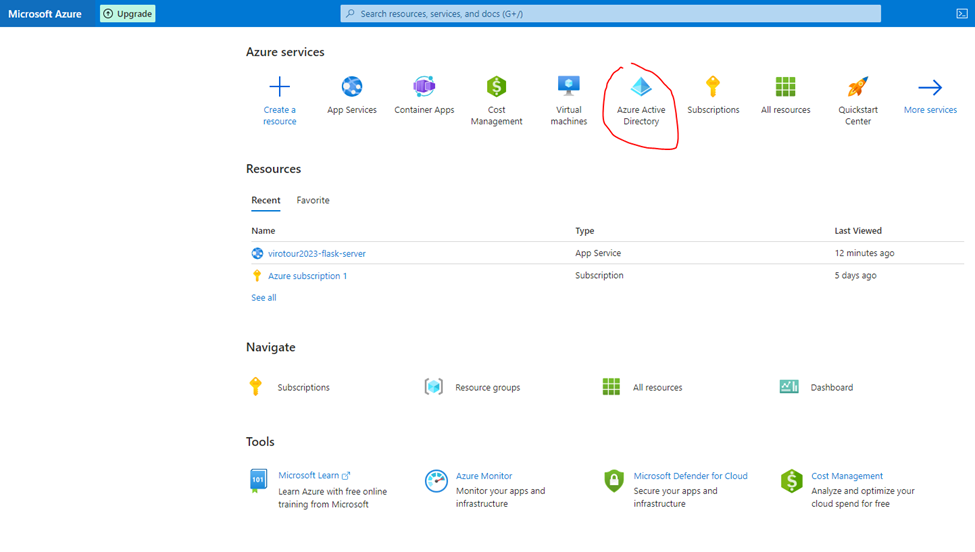
**Note:** Make sure to “Browse” and select the root folder having underneath the “requirement.txt.”

## 4.5 Sharing Access to Server

Steps to share the cloud azure subscription with other team members.

1. Add the user to the Active directory

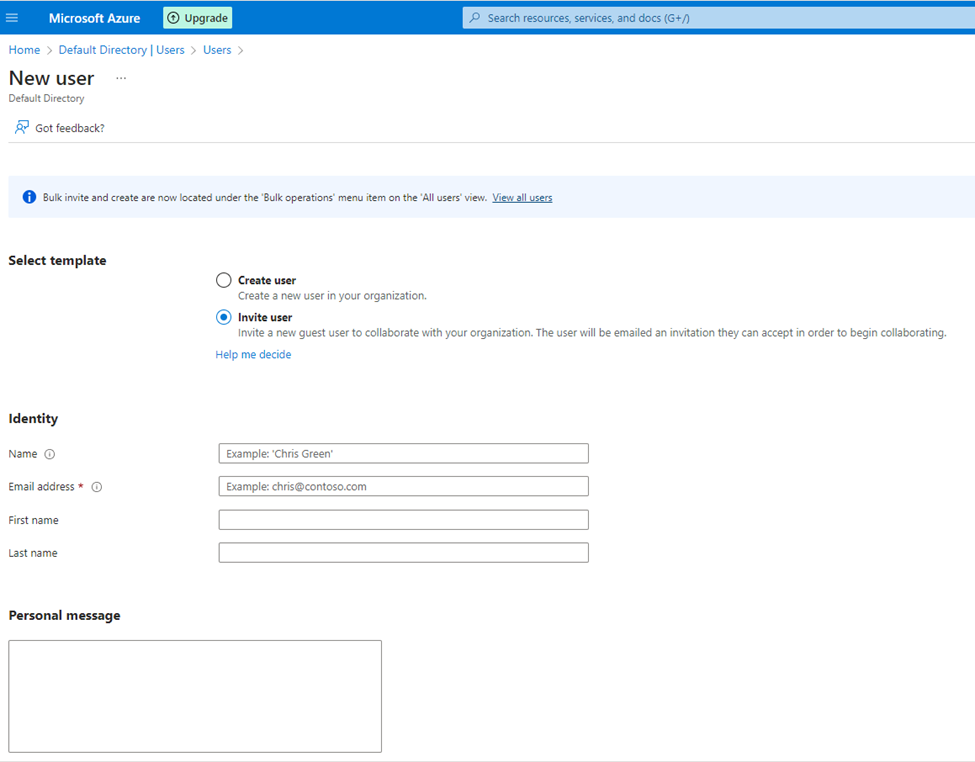
Login to [**Microsoft Azure Portal**](https://azure.microsoft.com/en-us/get-started/azure-portal)→ Azure [Active Directory](https://azure.microsoft.com/en-us/get-started/azure-portal)



**Figure 4.5.1 - Azure Active Directory Setting.**

1. Add a team member.

To add a team member, go to <https://learn.microsoft.com/en-us/azure/active-directory/fundamentals/add-users-azure-active-directory> → “Add New User” section → Invite user.



**Figure 4.5.2 - Adding New User to Directory.**

1. Select “**Invite user**,” fill out the form and click the “**Invite**” button.
2. The system will send an invitation via the above-provided email.

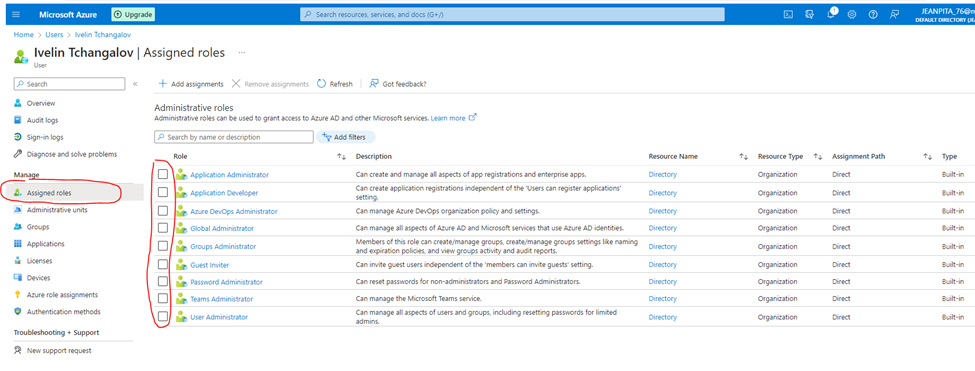
**Note:** The new user will be active only after accepting the invitation.

1. Add a role to the team member.

To Add a role to a team member, go to [https://learn.microsoft.com/en-us/azure/active-](https://learn.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-users-assign-role-azure-portal) directory/fundamentals/active-directory-users-assign-role-azure-portal

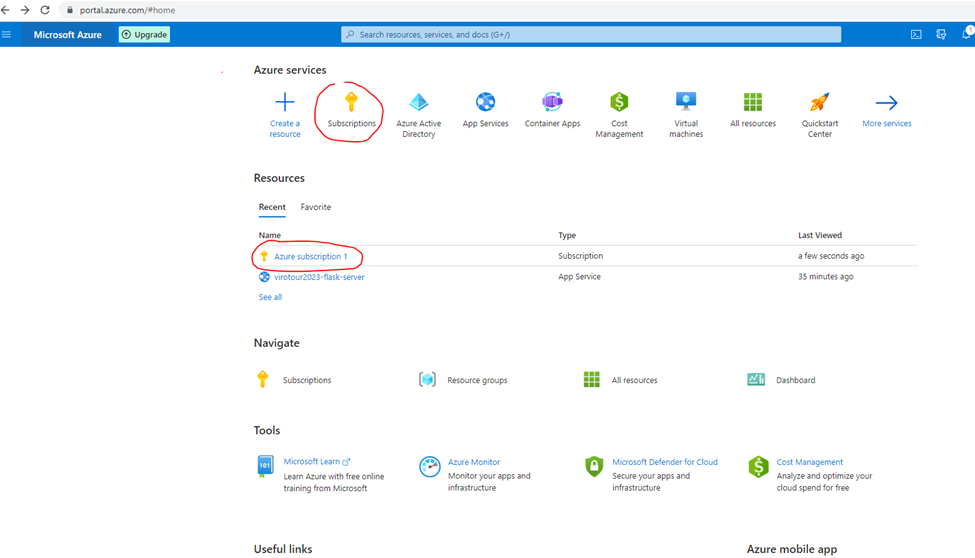
“Select the Role to Assign” → “Assign roles.”

**Note:** The application developer role is sufficient, but you can add any roles as needed.

**Figure 4.5.3 - Role Customization for Team Members.**

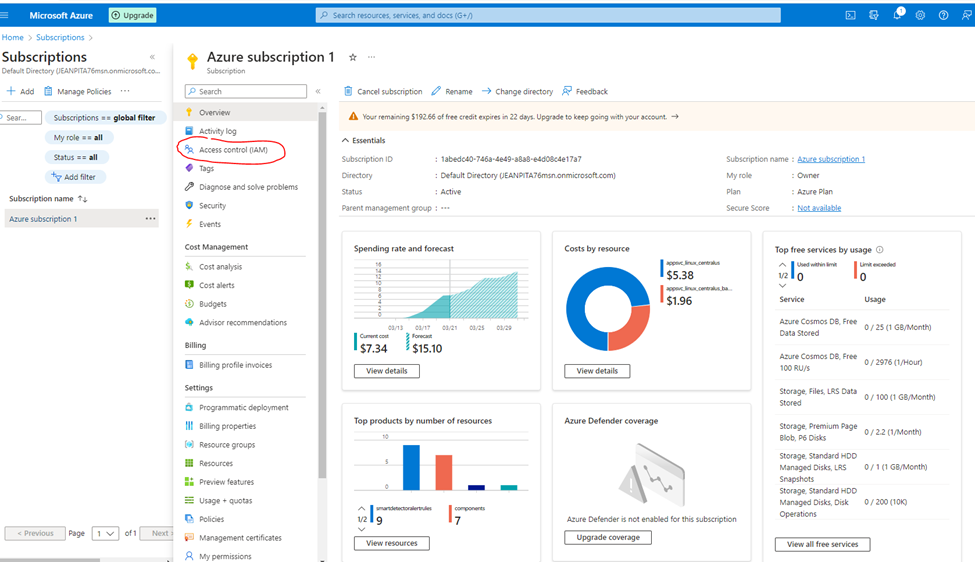
1. Share the subscription with the newly created user.

From the Azure portal, click on the “subscriptions.”



**Figure 4.5.4 - Azure Subscriptions Location.**

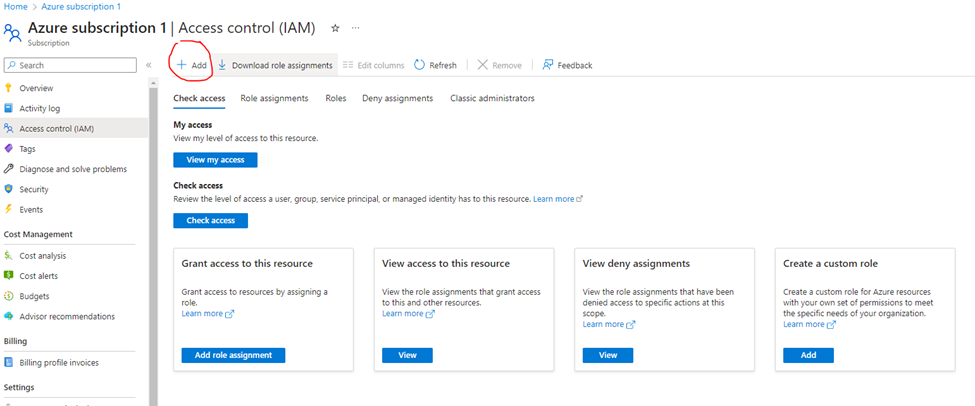
1. Go to “Access Control (AIM).



**Figure 4.5.5 - Access Control Option.**

1. Assign the subscription to a team member.

Click “Add”. Please make sure the user has **prior** accepted the **invitation**.



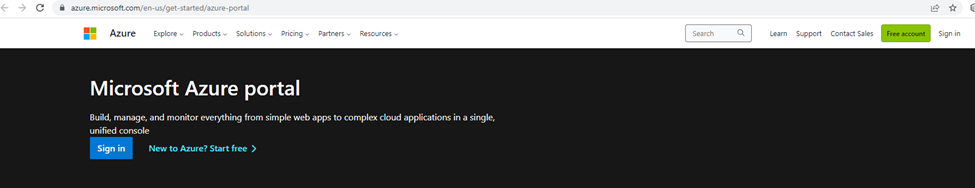
**Figure 4.5.6 - Adding Azure Subscription.**

Follow the instruction at <https://learn.microsoft.com/en-us/partner-center/assign-azure-subscriptions> → “Assign Azure subscriptions to your customers.”

# Troubleshooting

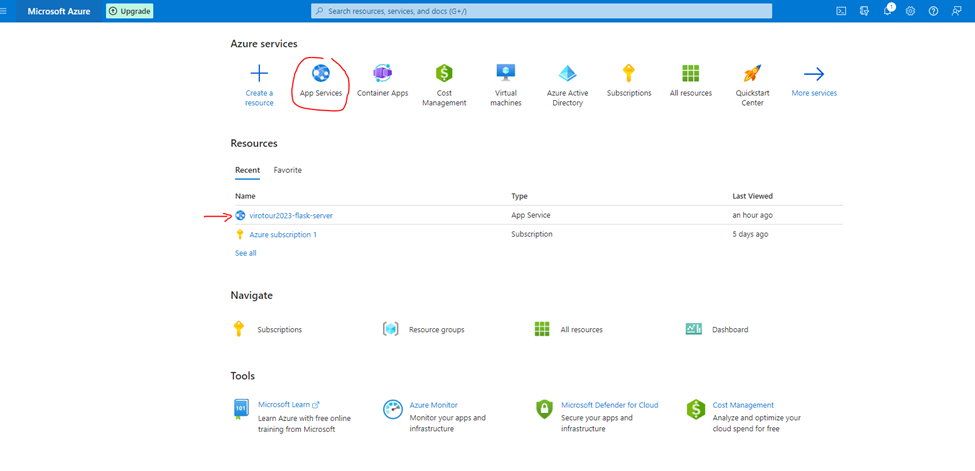
## 5.1 Deployment zip takes too long and fails

1. Login to [**Microsoft Azure Portal .**](https://azure.microsoft.com/en-us/get-started/azure-portal)



**Figure 5.1.1 - Microsoft Azure Portal Landing Page.**

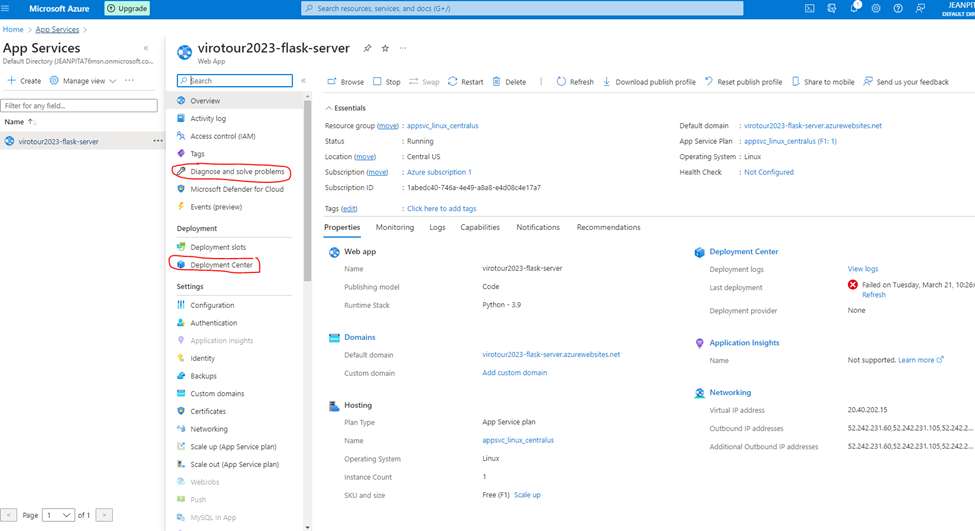
1. Click “App Services” → [virotour2023-flask-server](https://portal.azure.com/#@JEANPITA76msn.onmicrosoft.com/resource/subscriptions/1abedc40-746a-4e49-a8a8-e4d08c4e17a7/resourceGroups/appsvc_linux_centralus/providers/Microsoft.Web/sites/virotour2023-flask-server) (Web App Service created in Visual Studio Code).



**Figure 5.1.2 - App Services Location.**

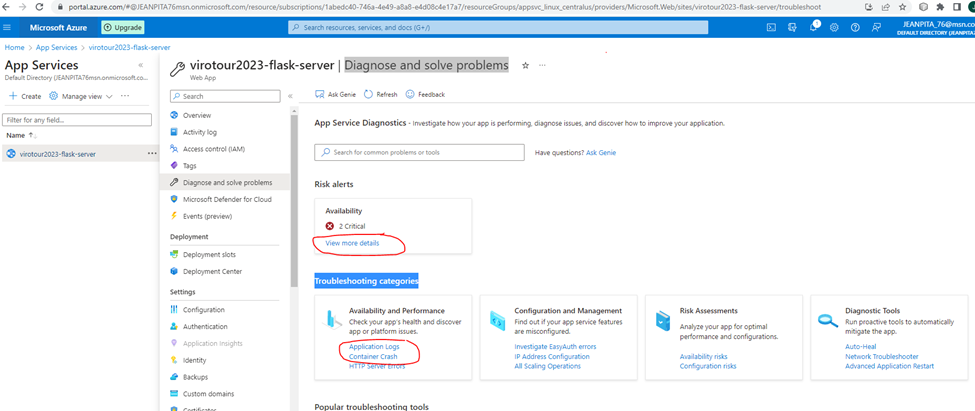
3. Click either “Deployment Center” → Logs tab to download the log files, or “Diagnose and solve problems” → “Troubleshooting categories.”

**App Services view**

`

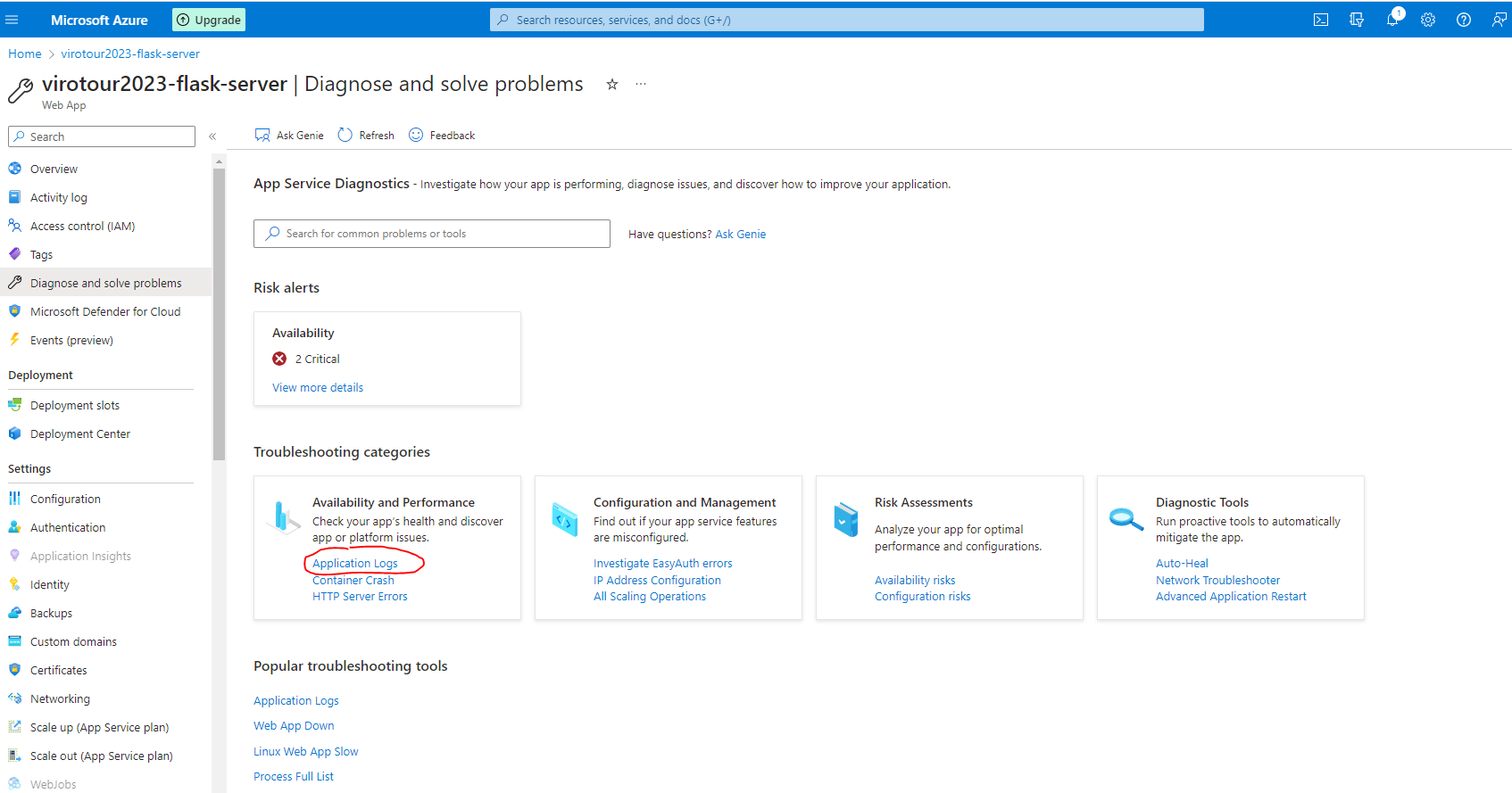
**Figure 5.1.3 - Settings Highlighted in App Services View.**

**Diagnose View**



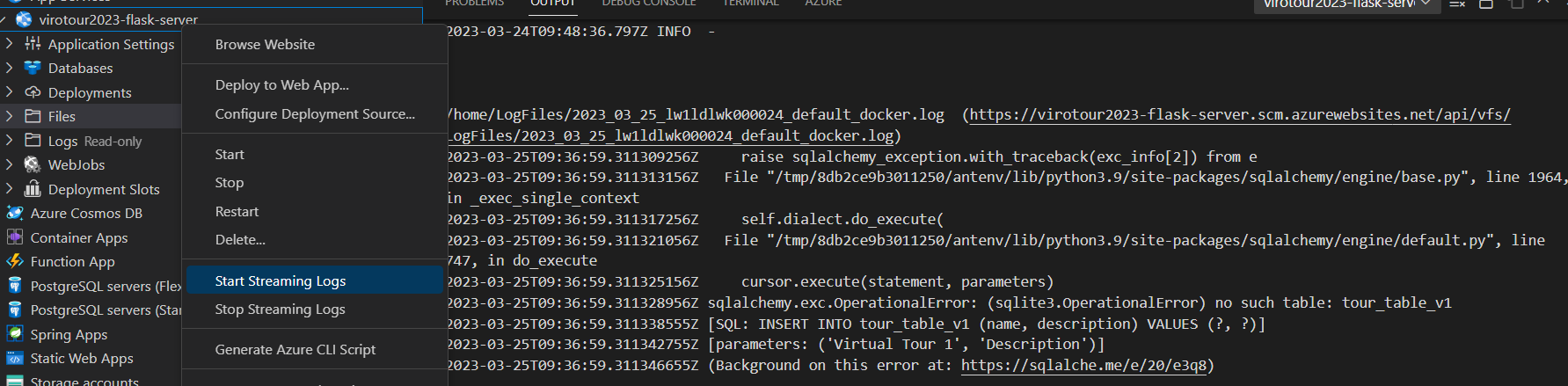
**Figure 5.1.4 - Settings Circled in Diagnose View.**

4. Go to “Application Logs.”



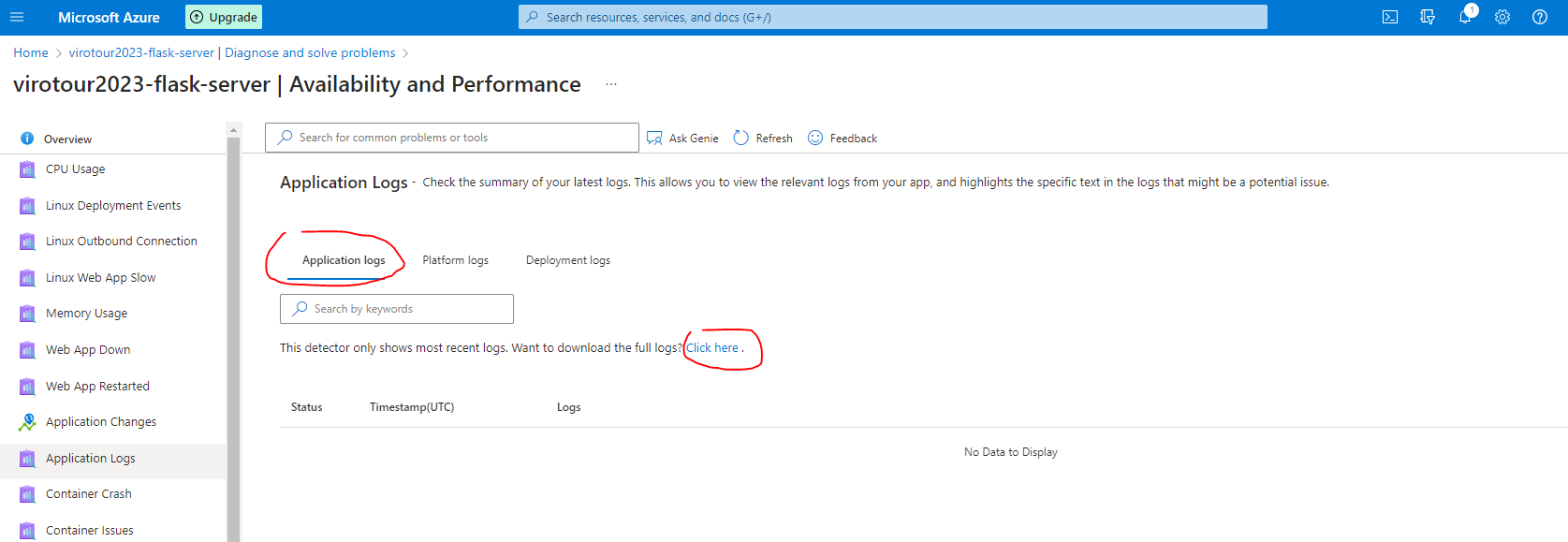
**Figure 5.1.5 - Settings Circled in under Availability and Performance Subsection.**

Another way to see them is from VS Code:



**Figure 5.1.4 - Settings Highlighted in Diagnose View.**

1. Go to “Availability and Performance” Application Logs → Click the link “Click here” to download the ZIP file.

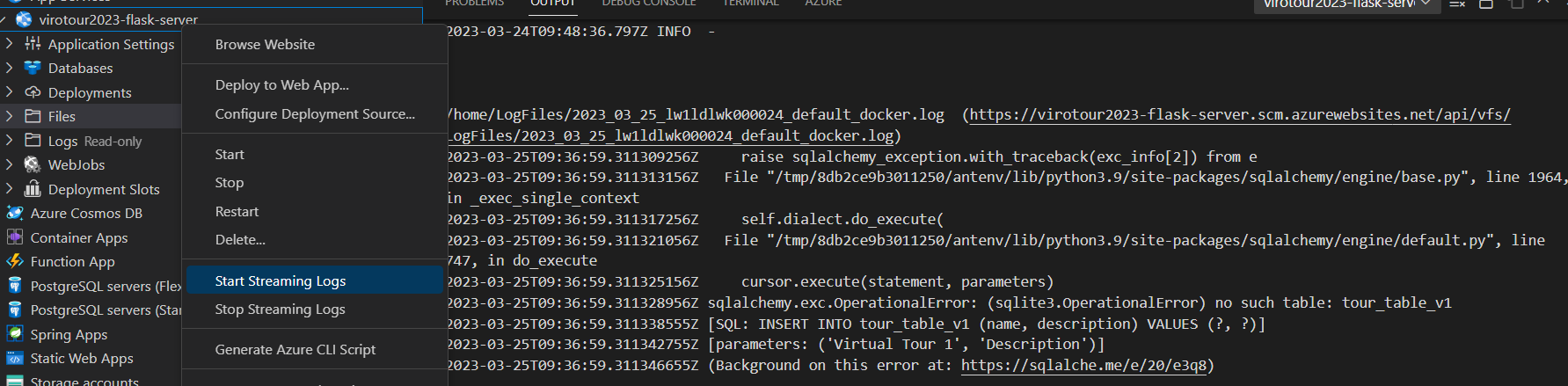


**Figure 5.1.5 - Route to Zip File.**

## 5.2 Unable to create tours

We believe that the database may have some hiccups during provisioning. As a workaround, we have to execute one of the *integration tests locally* in order to create the “app.db” file. This file gets pushed to Azure as part of the deployment, and the rest works fine.

In the future, we should debug how to properly bootstrap the tables. The screenshot below shows the problem you might see in the logs:



**Figure 5.2 - Tour Creation Error Log.**