A

REPORT ON

Project

On

**Azure App Services**

Submitted in partial fulfilment of the requirements

for the award of the Degree of

Bachelor of Computer Applications

Of

Poornima University

Submitted By:

**Chetna Bhurani**

**Arihant Jain**

**Lokesh Sharma**

**Piyush Singh Shekhawat**

II Year, BCA

Submitted To:

**Celebal Technologies**

**ACKNOWLEDGEMENT**

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of my project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank **Ms. Anuradha Raheja**, for providing me an opportunity to do the project work and giving us all support and guidance which made me complete the project duly. I am extremely thankful to her for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs.

We are thankful to **Dr. Manoj Gupta, ProPresident, Poornima University** for providing us a platform to carry out this activity successfully.

We are also very grateful to **Mr. Shamneesh Sharma(HOD, BCA Deaprtment)** for his kind support and guidance.

We are thankful and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs which helped us in successfully completing our project work.

Chetna Bhurani

Arihant Jain

Lokesh Sharma

Piyush Singh Shekhawat

**Abstract**

This project report is basically about how to create and deploy the web app on Microsoft Azure and to create development slots for the same.

Azure App Service is a fully managed "Platform as a Service" (PaaS) that integrates Microsoft Azure Websites, Mobile Services, and BizTalk Services into a single service, adding new capabilities that enable integration with on-premises or cloud systems. Azure App Service gives users several capabilities.

The Azure App Service Environment is an Azure App Service feature that provides a fully isolated and dedicated environment for securely running App Service apps at high scale.

**TABLE OF CONTENT**

Cover Page 1

Acknowledgment 2

Abstract 3

1.1 What is App Service……………………………………………………… 5

1.2 Key Features of App service …………………………………………… 6

1.3 App Service Plans ………………………………………………………. 8

1.4 Virtual Network Support……….……………………………………….. 9

1.5 What are Development Support.……………………………………….. 9

1.6 Steps to Create & deploy an App……………………………………….. 10

1.7 Steps to Create Development slot……………………………………….. 16

1.8 References………………………………………………………….……. 21

### *Project - Web App Service*

### What Is Azure App Service?

Azure App Service is a fully managed "Platform as a Service" (PaaS) that integrates Microsoft Azure Websites, Mobile Services, and BizTalk Services into a single service, adding new capabilities that enable integration with on-premises or cloud systems. Azure App Service gives users several capabilities

* Provision and deploy Web and Mobile Apps in seconds
* Build engaging iOS, Android, and Windows apps
* Automate business processes with a visual design experience
* Integrate with "Software as a Service" (SaaS) applications (Office 365, Salesforce, Dynamics, OneDrive, Box, Dropbox, Twilio, Twitter, Facebook, Marketo, and so on) and on-premises applications

Azure App Service is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. You can develop in your favorite language, be it .NET, .NET Core, Java, Ruby, Node.js, PHP, or Python. Applications run and scale with ease on both Windows and Linux-based environments. App Service not only adds the power of Microsoft Azure to your application, such as security, load balancing, autoscaling, and automated management. You can also take advantage of its DevOps capabilities, such as continuous deployment from Azure DevOps, GitHub, Docker Hub, and other sources, package management, staging environments, custom domain, and TLS/SSL certificates.

The Azure App Service Environment is an Azure App Service feature that provides a fully isolated and dedicated environment for securely running App Service apps at high scale. This capability can host your:

* Windows web apps
* Linux web apps
* Docker containers
* Mobile apps
* Functions

Customers can create multiple ASEs within a single Azure region or across multiple Azure regions. This flexibility makes ASEs ideal for horizontally scaling stateless application tiers in support of high RPS workloads.

* 1. **Key features of App Service:**
* **Multiple languages and frameworks** - App Service has first-class support for ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python. You can also run [PowerShell and other scripts or executables](https://docs.microsoft.com/en-us/azure/app-service/webjobs-create) as background services.
* **Managed production environment** - App Service automatically [patches and maintains the OS and language frameworks](https://docs.microsoft.com/en-us/azure/app-service/overview-patch-os-runtime) for you. Spend time writing great apps and let Azure worry about the platform.
* **DevOps optimization** - Set up [continuous integration and deployment](https://docs.microsoft.com/en-us/azure/app-service/deploy-continuous-deployment) with Azure DevOps, GitHub, BitBucket, Docker Hub, or Azure Container Registry. Promote updates through [test and staging environments](https://docs.microsoft.com/en-us/azure/app-service/deploy-staging-slots). Manage your apps in App Service by using [Azure PowerShell](https://docs.microsoft.com/en-us/powershell/azureps-cmdlets-docs) or the [cross-platform command-line interface (CLI)](https://docs.microsoft.com/en-us/cli/azure/install-azure-cli).
* **Global scale with high availability** - Scale [up](https://docs.microsoft.com/en-us/azure/app-service/manage-scale-up) or [out](https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/insights-how-to-scale) manually or automatically. Host your apps anywhere in Microsoft's global datacenter infrastructure, and the App Service [SLA](https://azure.microsoft.com/support/legal/sla/app-service/) promises high availability.
* **Connections to SaaS platforms and on-premises data** - Choose from more than 50 [connectors](https://docs.microsoft.com/en-us/azure/connectors/apis-list) for enterprise systems (such as SAP), SaaS services (such as Salesforce), and internet services (such as Facebook). Access on-premises data using [Hybrid Connections](https://docs.microsoft.com/en-us/azure/app-service/app-service-hybrid-connections) and [Azure Virtual Networks](https://docs.microsoft.com/en-us/azure/app-service/web-sites-integrate-with-vnet).
* **Security and compliance** - App Service is [ISO, SOC, and PCI compliant](https://www.microsoft.com/en-us/trustcenter). Authenticate users with [Azure Active Directory](https://docs.microsoft.com/en-us/azure/app-service/configure-authentication-provider-aad) or with social login ([Google](https://docs.microsoft.com/en-us/azure/app-service/configure-authentication-provider-google), [Facebook](https://docs.microsoft.com/en-us/azure/app-service/configure-authentication-provider-facebook), [Twitter](https://docs.microsoft.com/en-us/azure/app-service/configure-authentication-provider-twitter), and [Microsoft](https://docs.microsoft.com/en-us/azure/app-service/configure-authentication-provider-microsoft)). Create [IP address restrictions](https://docs.microsoft.com/en-us/azure/app-service/app-service-ip-restrictions) and [manage service identities](https://docs.microsoft.com/en-us/azure/app-service/overview-managed-identity).
* **Application templates** - Choose from an extensive list of application templates in the [Azure Marketplace](https://azure.microsoft.com/marketplace/), such as WordPress, Joomla, and Drupal.
* **Visual Studio integration** - Dedicated tools in Visual Studio streamline the work of creating, deploying, and debugging.
* **API and mobile features** - App Service provides turn-key CORS support for RESTful API scenarios, and simplifies mobile app scenarios by enabling authentication, offline data sync, push notifications, and more.
* **Serverless code** - Run a code snippet or script on-demand without having to explicitly provision or manage infrastructure, and pay only for the compute time your code actually uses (see [Azure Functions](https://docs.microsoft.com/en-us/azure/azure-functions/)).

#### App Service Plans

In App Service, an app runs in an *App Service plan*. An App Service plan defines a set of compute resources for a web app to run. These compute resources are analogous to the server farm in conventional web hosting. One or more apps can be configured to run on the same computing resources (or in the same App Service plan).

When you create an App Service plan in a certain region (for example, West Europe), a set of compute resources is created for that plan in that region. Whatever apps you put into this App Service plan run on these compute resources as defined by your App Service plan. Each App Service plan defines:

1. Region (West US, East US, etc.)
2. Number of VM instances
3. Size of VM instances (Small, Medium, Large)
4. Pricing tier (Free, Shared, Basic, Standard, Premium, PremiumV2, Isolated)

The *pricing tier* of an App Service plan determines what App Service features you get and how much you pay for the plan. There are a few categories of pricing tiers:

* **Shared compute**: **Free** and **Shared**, the two base tiers, runs an app on the same Azure VM as other App Service apps, including apps of other customers. These tiers allocate CPU quotas to each app that runs on the shared resources, and the resources cannot scale out.
* **Dedicated compute**:The **Basic**, **Standard**, **Premium**, and **PremiumV2** tiers run apps on dedicated Azure VMs. Only apps in the same App Service plan share the same compute resources. The higher the tier, the more VM instances are available to you for scale-out.
* **Isolated**: This tier runs dedicated Azure VMs on dedicated Azure Virtual Networks. It provides network isolation on top of compute isolation to your apps. It provides the maximum scale-out capabilities.

## *Virtual network support*

The ASE feature is a deployment of the Azure App Service directly into a customer's Azure Resource Manager virtual network.  An ASE always exists in a virtual network, and more precisely, within a subnet of a virtual network. You can use the security features of virtual networks to control inbound and outbound network communications for your apps.

#### What are Deployment Slots?

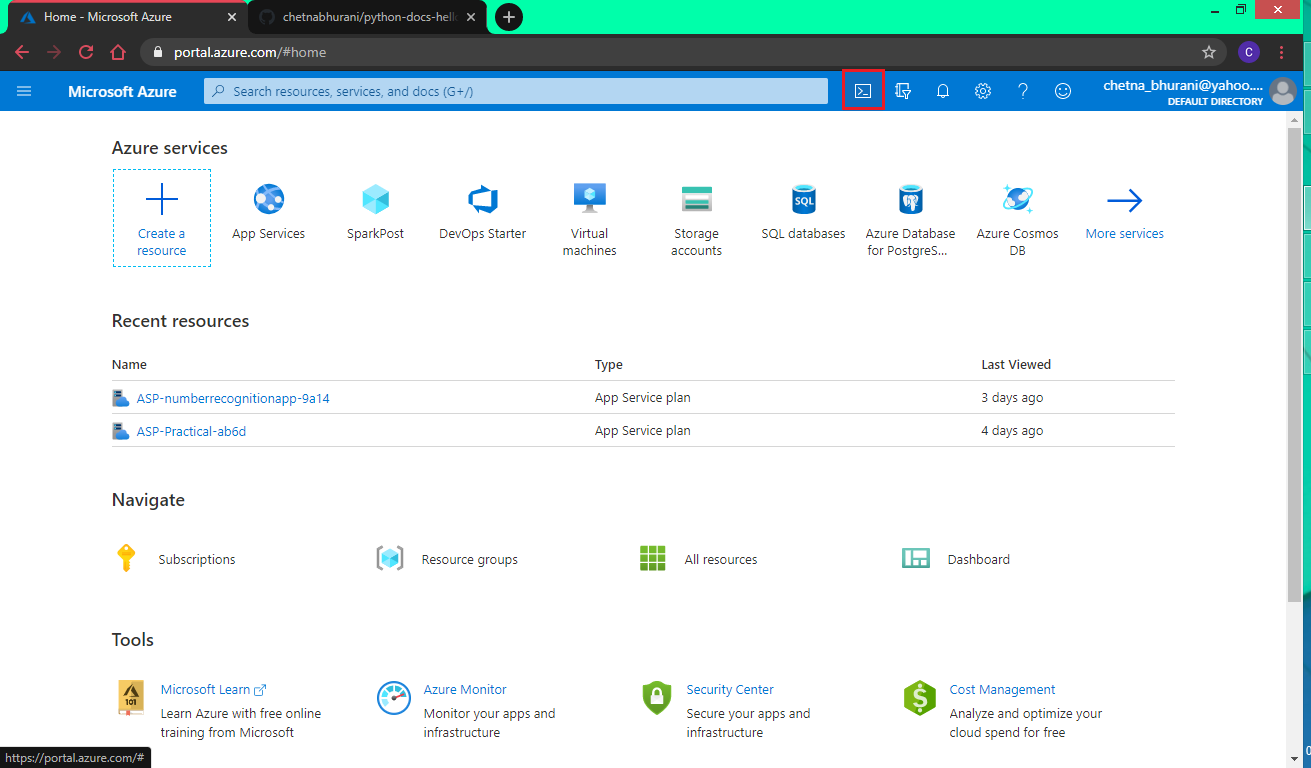
Deployment Slots are a feature of Azure App Service. They actually are live apps with their own hostnames. You can create different slots for your application (for e.g. Dev, Test or Stage). The Production slot is the slot where your live app resides. With deployment slots, you can validate app changes in staging before swapping it with your production slot.

When you deploy your web app, web app on Linux, mobile back end, or API app to Azure App Service, you can use a separate deployment slot instead of the default production slot when you're running in the **Standard**, **Premium**, or **Isolated** App Service plan tier. Deployment slots are live apps with their own host names. App content and configurations elements can be swapped between two deployment slots, including the production slot.

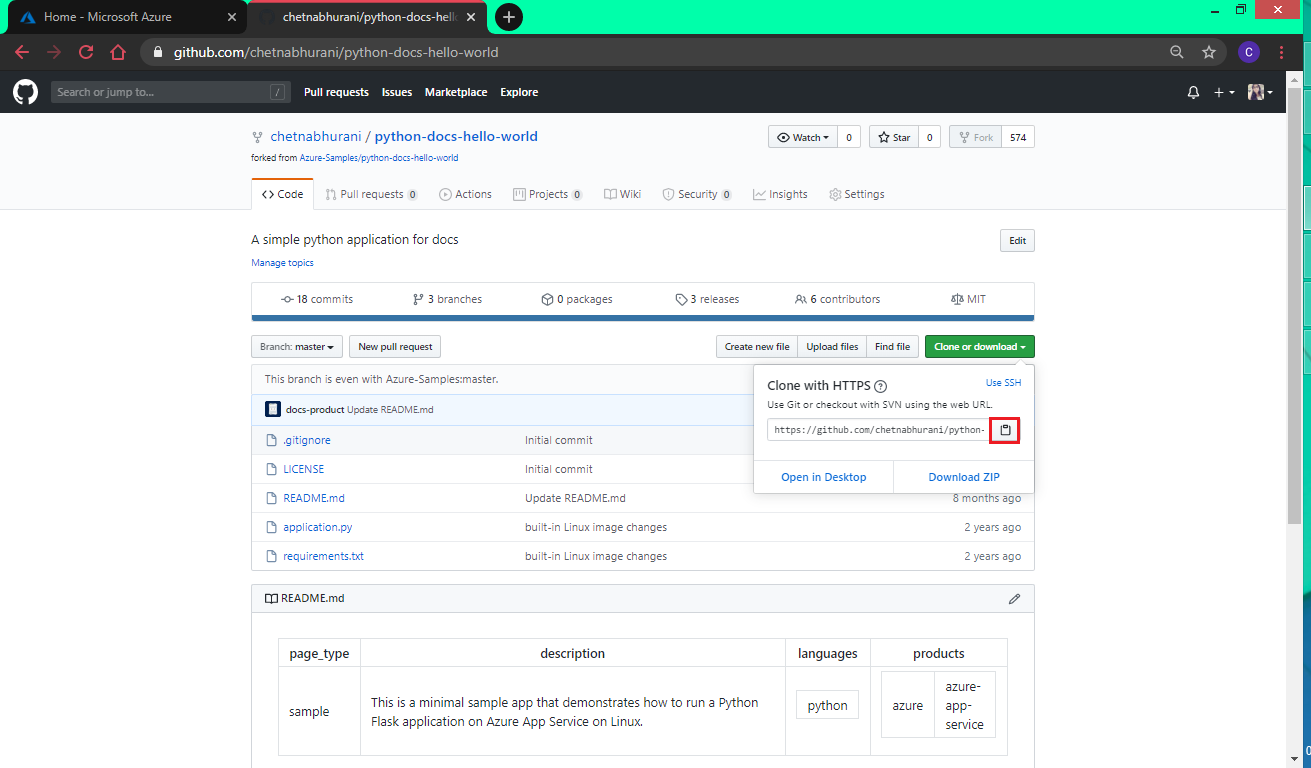
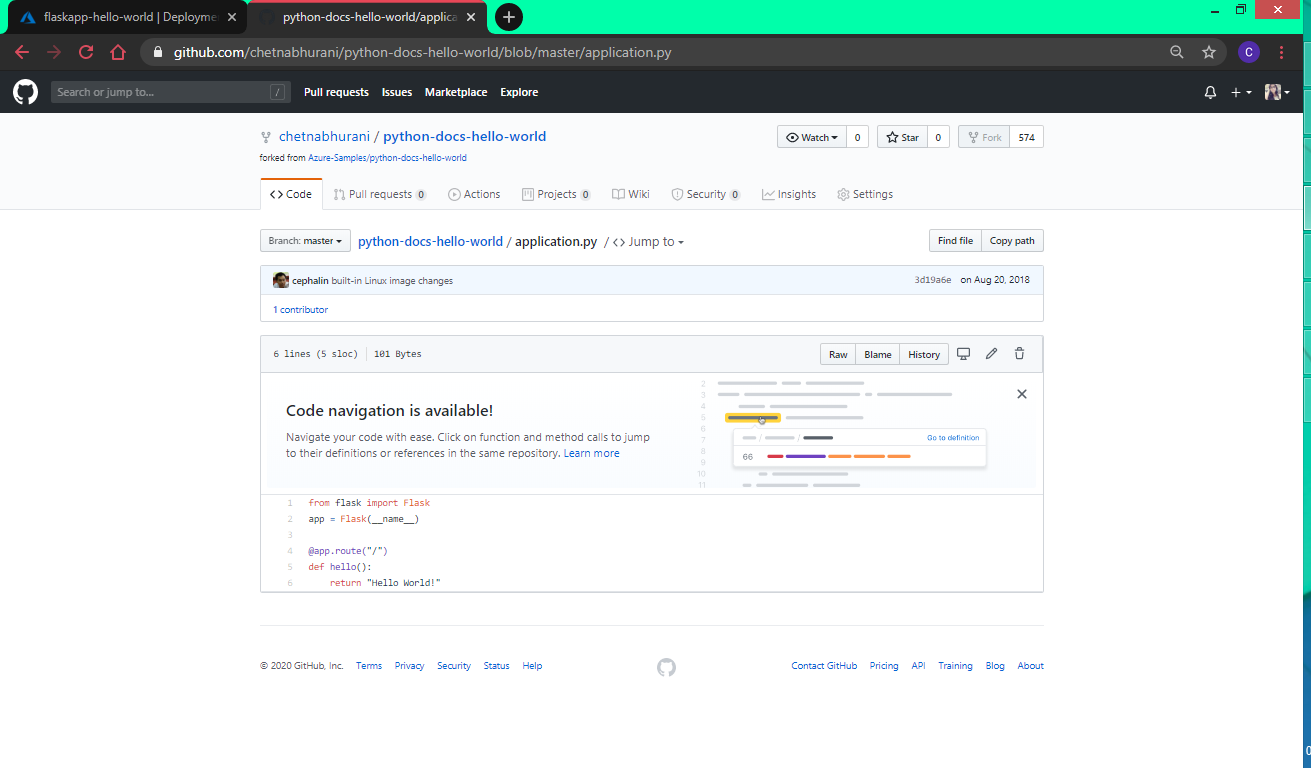
Deploying your application to a non-production slot has the following benefits:

* You can validate app changes in a staging deployment slot before swapping it with the production slot.
* Deploying an app to a slot first and swapping it into production makes sure that all instances of the slot are warmed up before being swapped into production. This eliminates downtime when you deploy your app. The traffic redirection is seamless, and no requests are dropped because of swap operations. You can automate this entire workflow by configuring auto swap when pre-swap validation isn't needed.
* After a swap, the slot with previously staged app now has the previous production app. If the changes swapped into the production slot aren't as you expect, you can perform the same swap immediately to get your "last known good site" back.
  1. **Steps to Create & Deploy an App –**

Step 1- Open Cloud Shell (Bash) on Azure Portal.

****

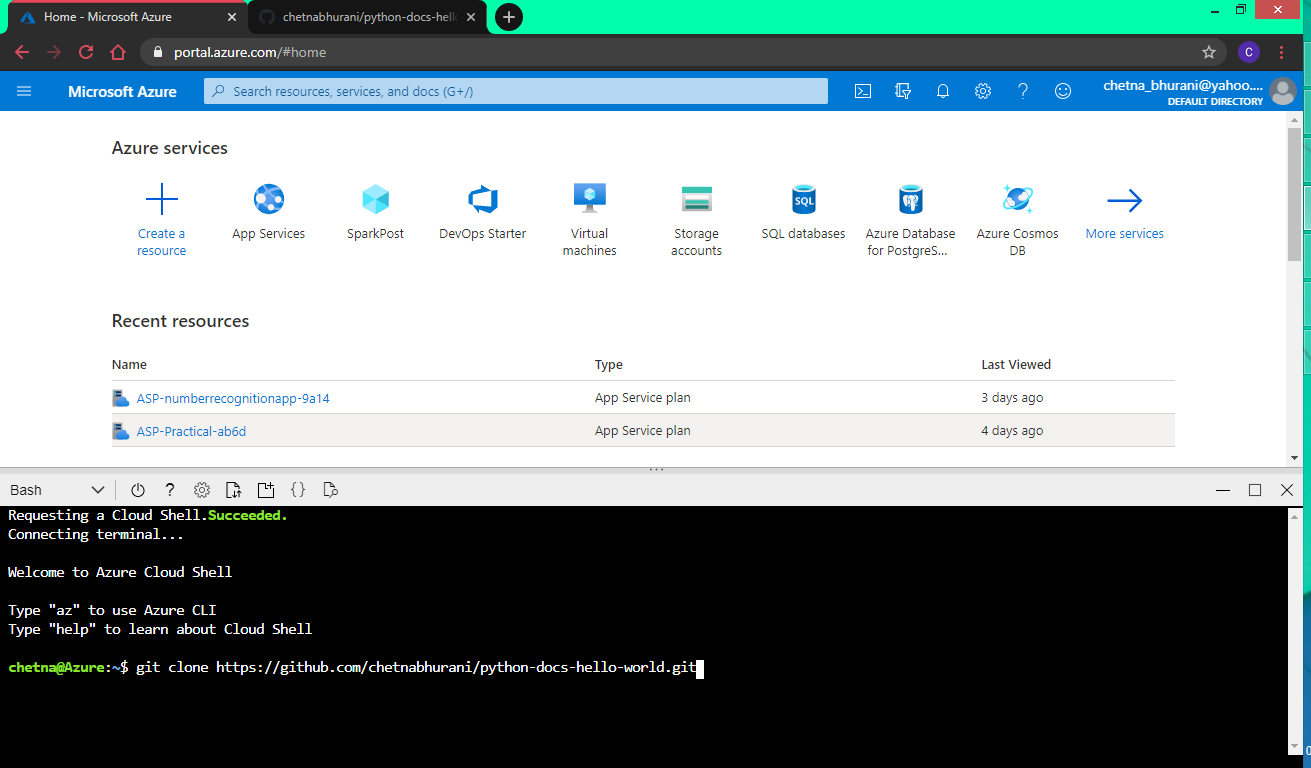
Step 2- Open the GIT repository which you want to clone. Click on “Clone or Download” option and copy the link.

****

Step 3 – In Cloud Shell, enter

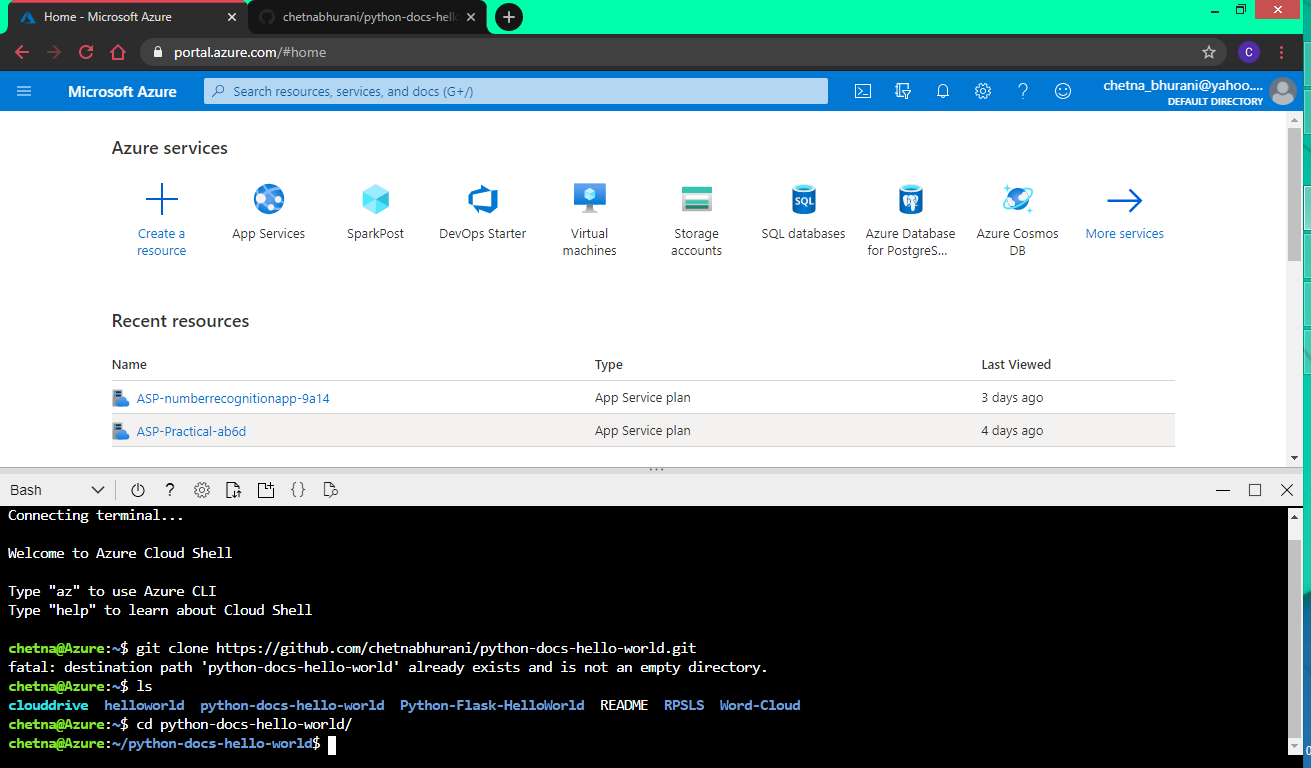
*git clone <copied url from previous step>*

and Enter.

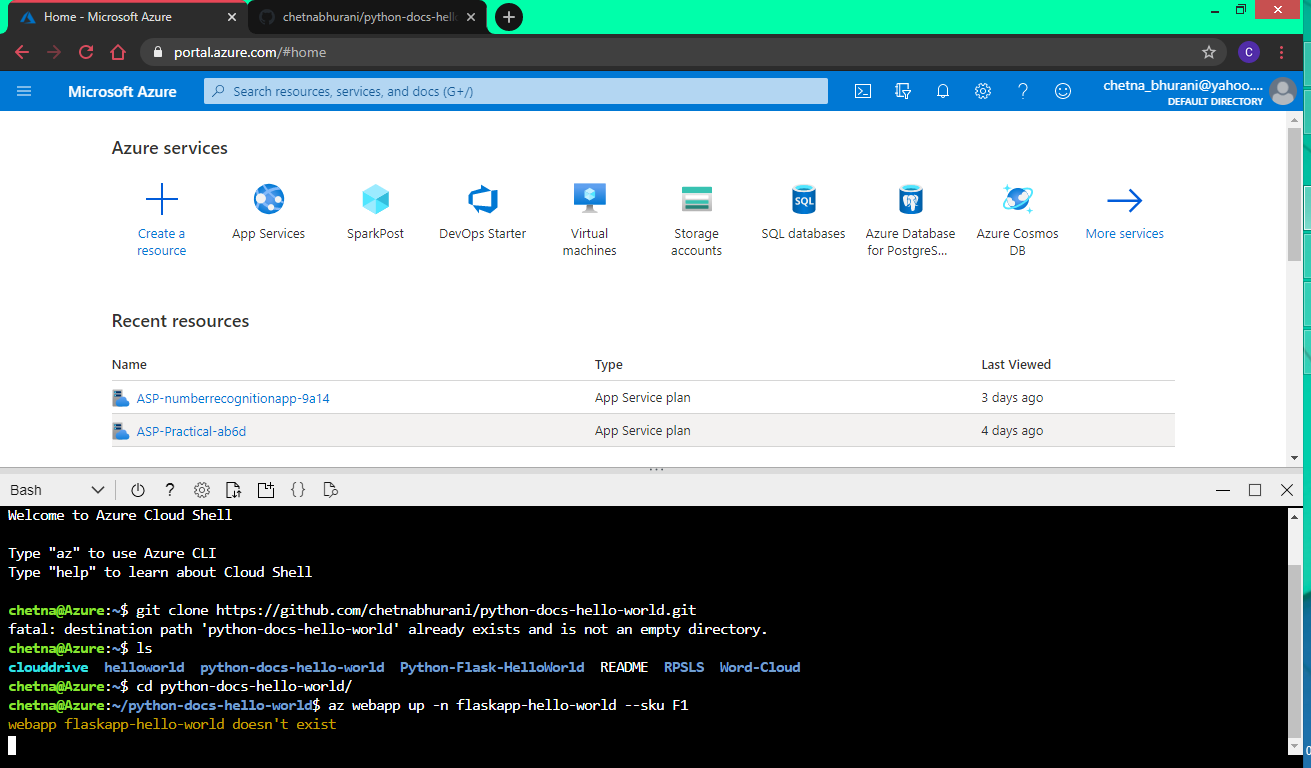
****

Step 4 – Using *cd* command, enter the repository directory.

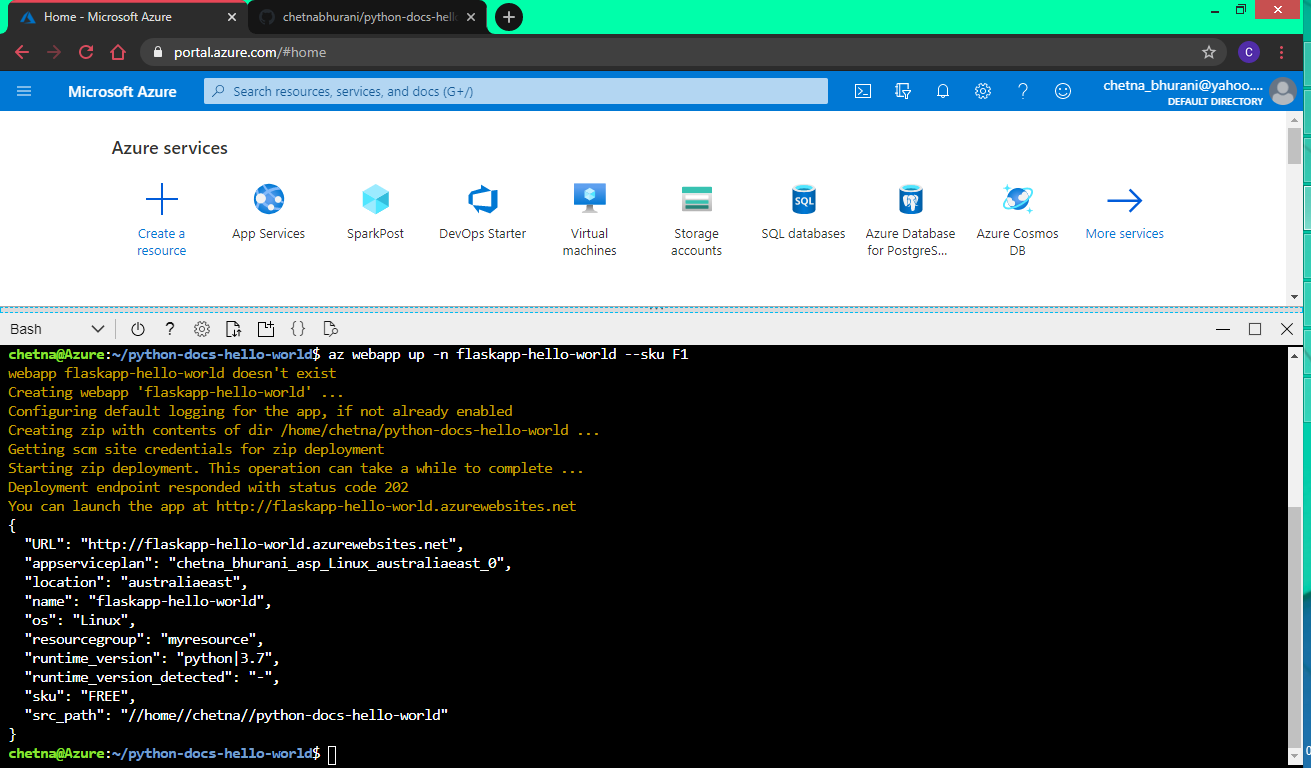
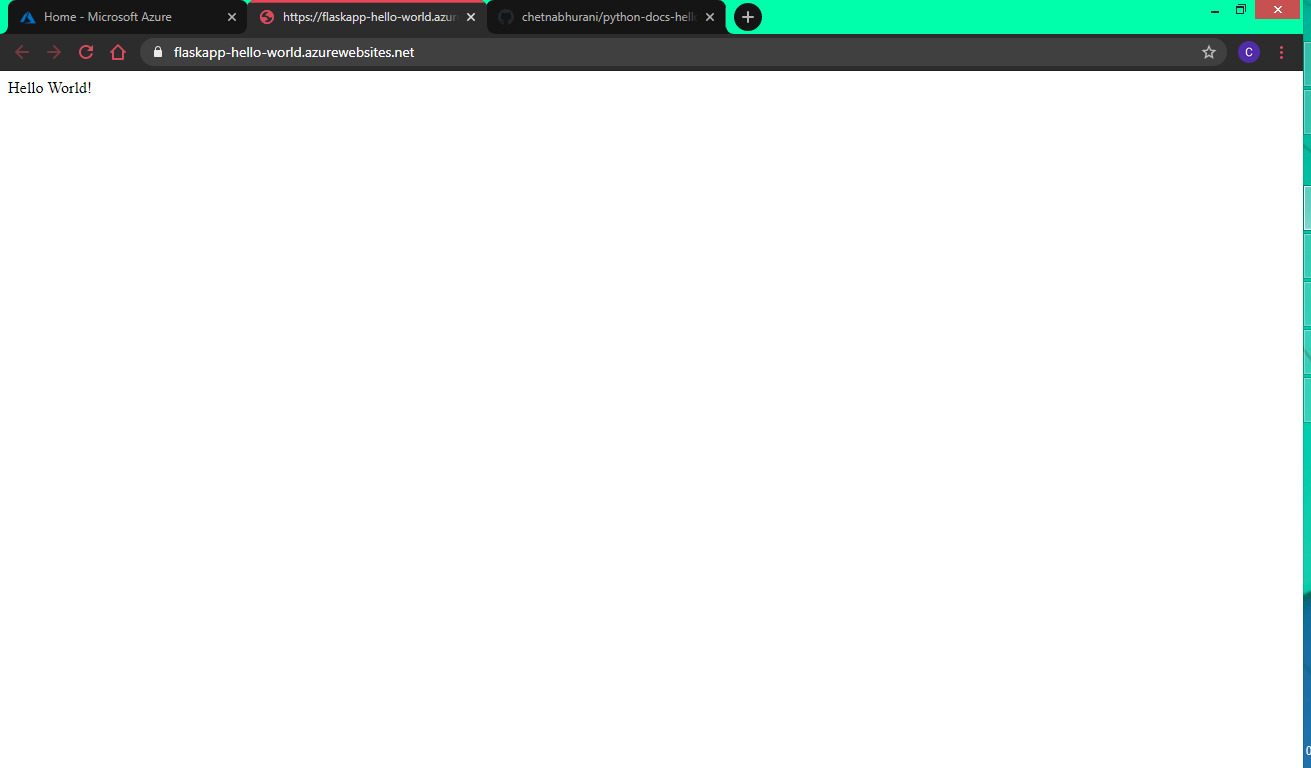
*cd <repository-name>*

**** Step 5 – To create and deploy the web app,

*az webapp up –n <app-name> --sku <plan-type>--resource-group <resource-grp>*

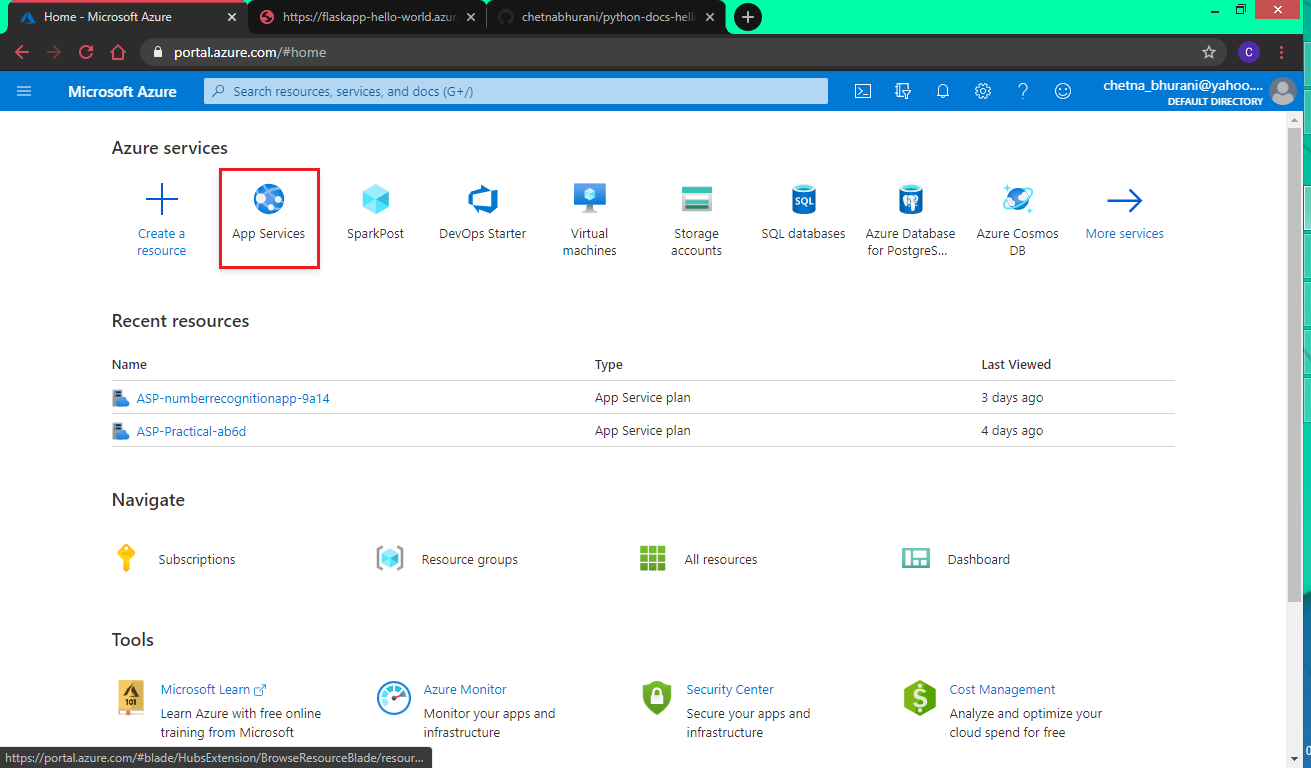
****

Wait for the web app to create and deploy.

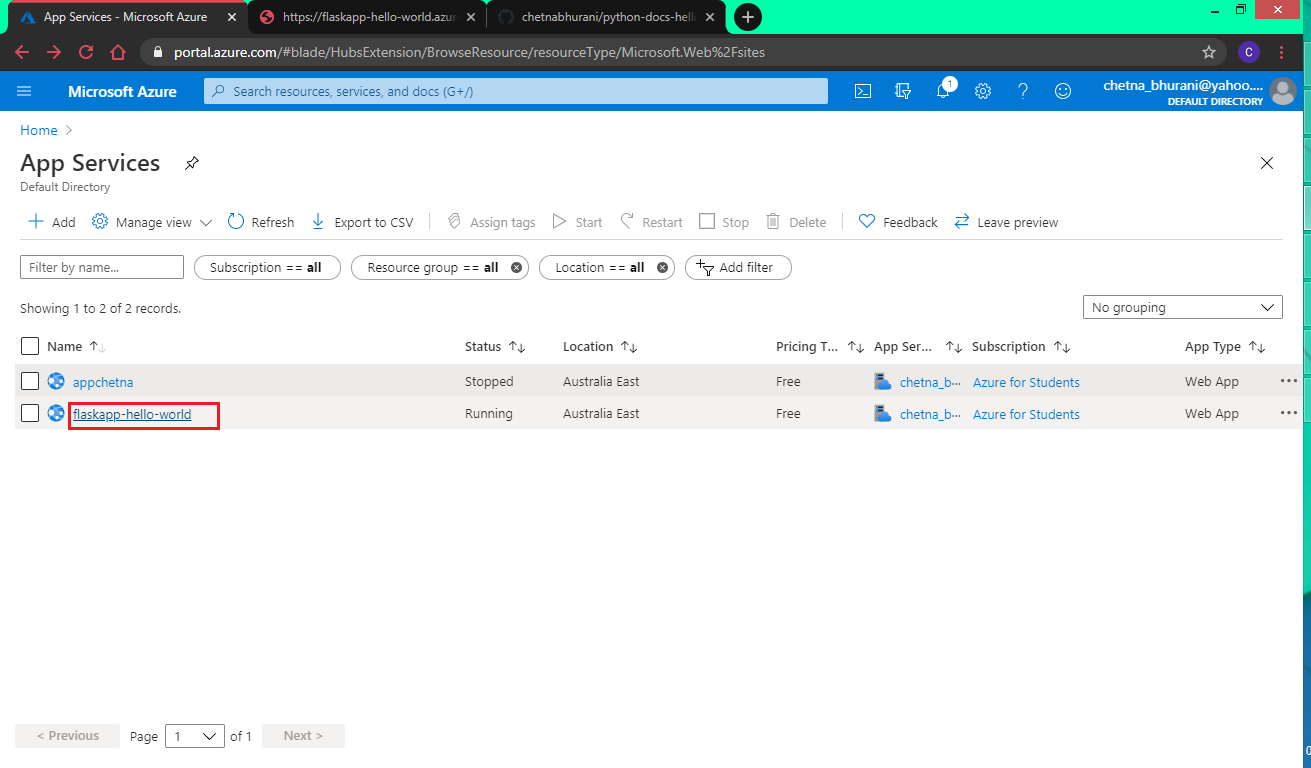
****Step 6 – Click on the webapp link to launch it. ****

* 1. **Steps to create Development Slot-**

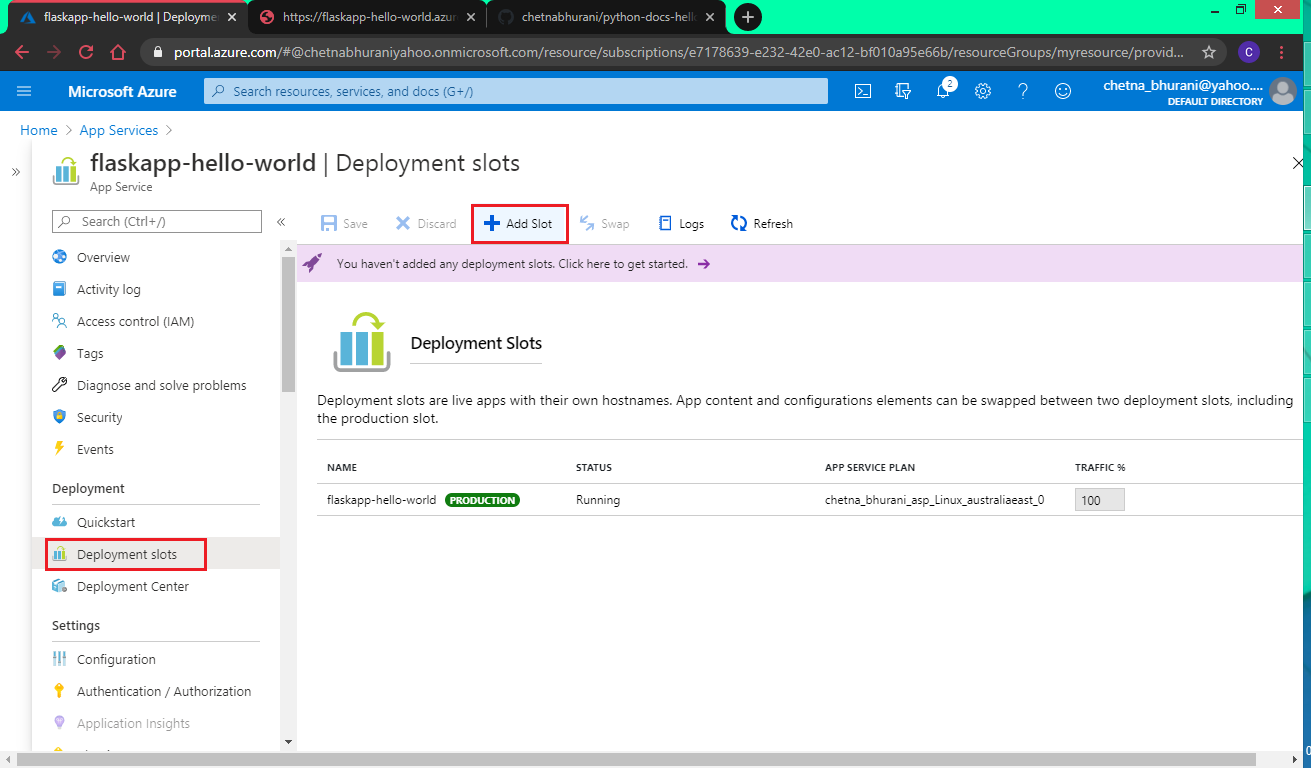
Step 1 – Open App Services Panel on Azure Portal.

****

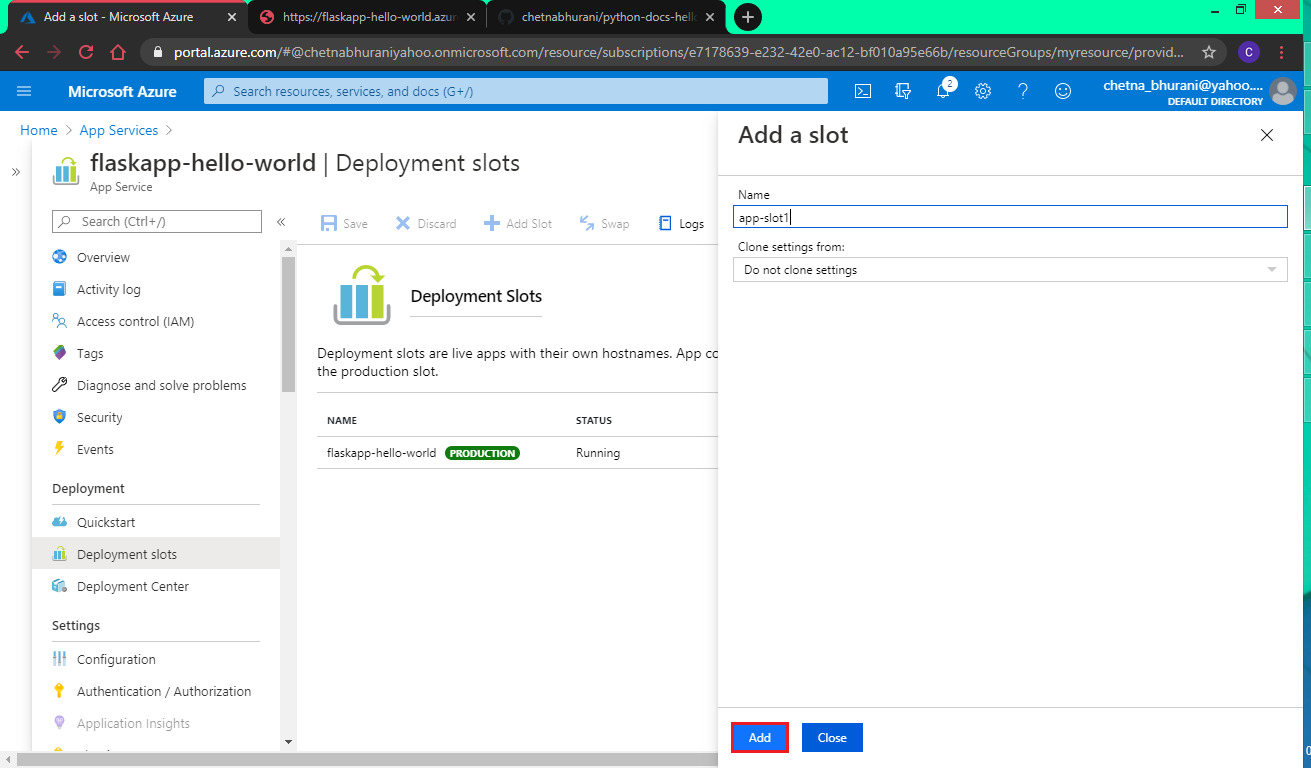
Step 2 – Select the created app.

****

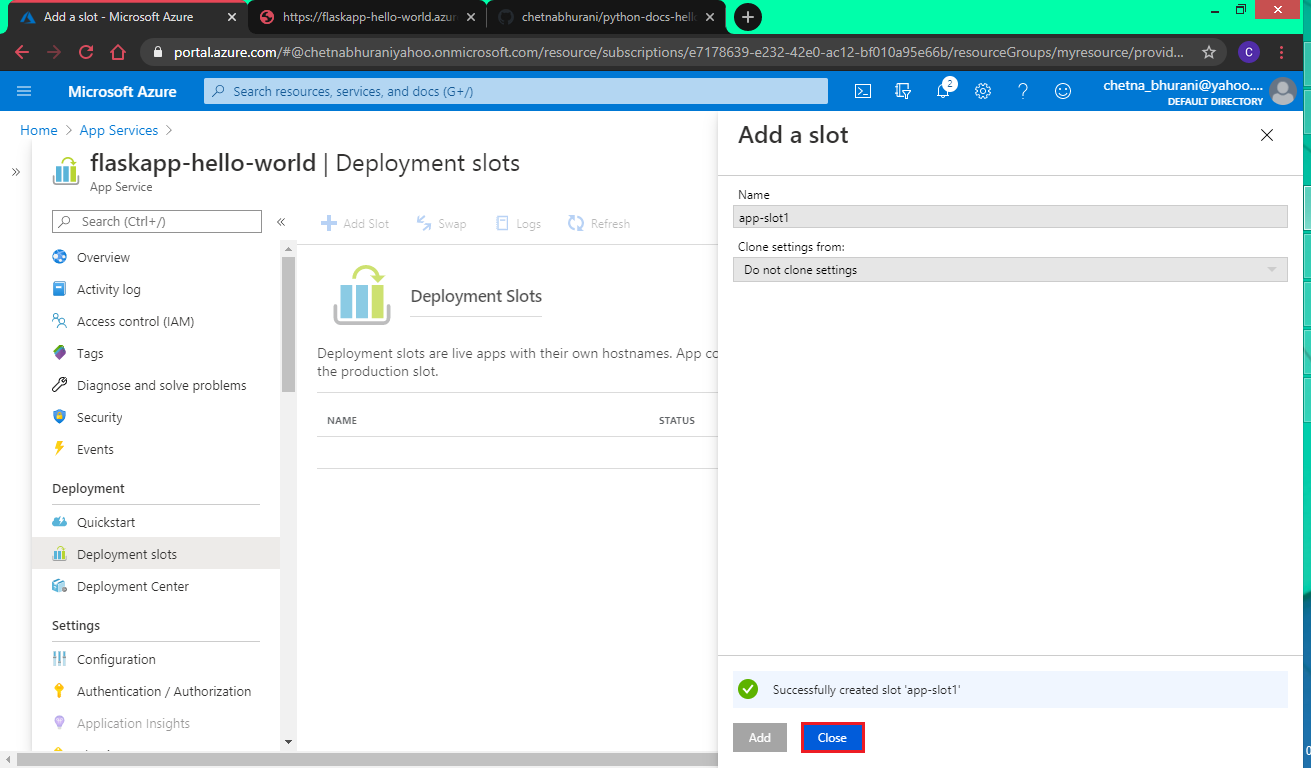
Step 3 – Select “Development Slots” option. Click on “Add Slot” to add a development slot.

****

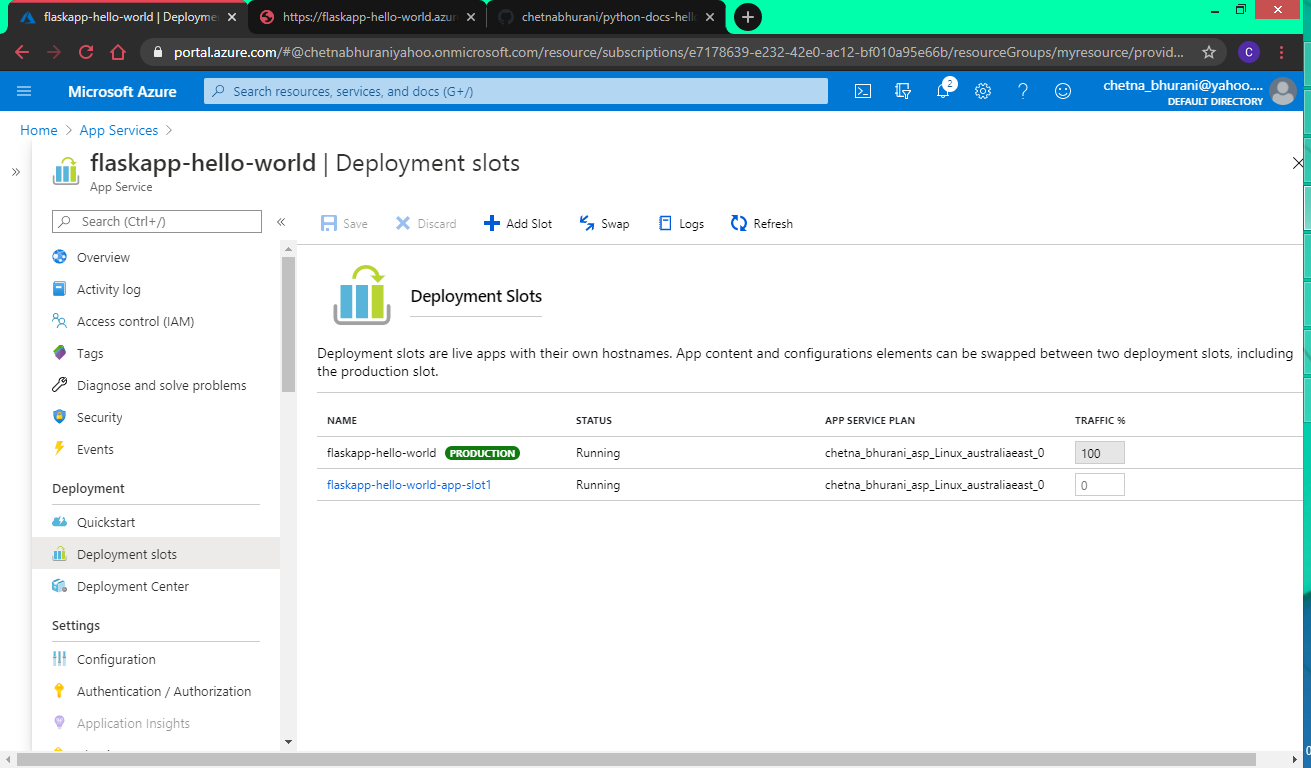
Step 4 – Enter the name of the slot you want to add. And click on “Add” button.

****

Wait for the creation. Click on “Close” button.

****

Development Slot is created successfully.

****

**References-**

* jakeydocs.readthedocs.io
* www.informit.com
* docs.microsoft.com
* microsoft.github.io
* azure.microsoft.com