CHAPTER 10

App Service Environments

Implementing security in some applications requires full isolation from other resources. Azure app service environments (ASE) provide a dedicated fully isolated environment to deploy Windows and Linux web apps, docker containers, and function apps. Essentially, app service apps can run in full isolation, with isolated higher network security and access, high scalability, high memory utilization, and catering for higher requests per second, in an app service environment.

App service environment deployed apps are already in a virtual network without you having to do any additional configurations. Your ASE is a single tenant system that is not shared with anyone else providing isolation fully to your apps. You can even demand that your ASE be deployed into dedicated hardware to have complete isolation.

Version 3 of ASE supports creating up to 200 app service plans, and version 2 supports up to 100 plans. Apps in ASE can access resources in the virtual network of the ASE without you having to perform any additional configurations. With ASE version 3 you can deploy as a zone redundant as well.

P. Herath, *Azure Cloud Security for Absolute Beginners*,
https://doi.org/10.1007/978-1-4842-7860-4_10

Lesson 10.1: Setting Up Azure App Service Environment

With the brief understanding we have about app service environments (ASEs), let's get started by setting up an ASE.

In Azure portal you can search for app service environments (see Figure 10-1).

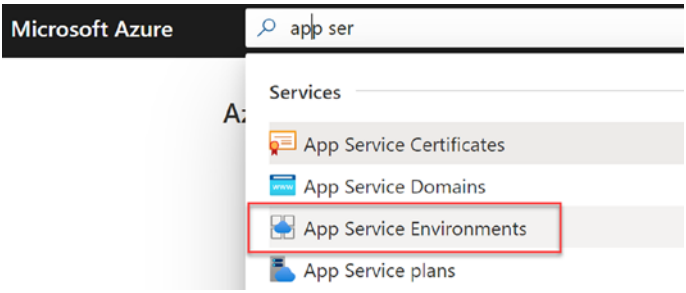

Figure 10-1. *Searching for ASEs*

You can click on Add to get started, creating your first ASE (see Figure 10-2).

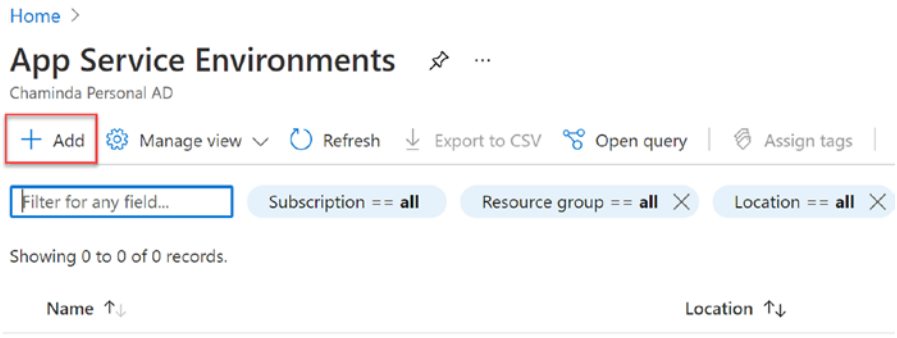

Figure 10-2. *Adding ASE*

In the basic information page of ASE, select an existing resource group or provide a name to create a new resource group. A name for ASE should be provided. You can select the virtual IP of ASE to be internal, which would create it as an internal load-balancer endpoint (see Figure 10-3).

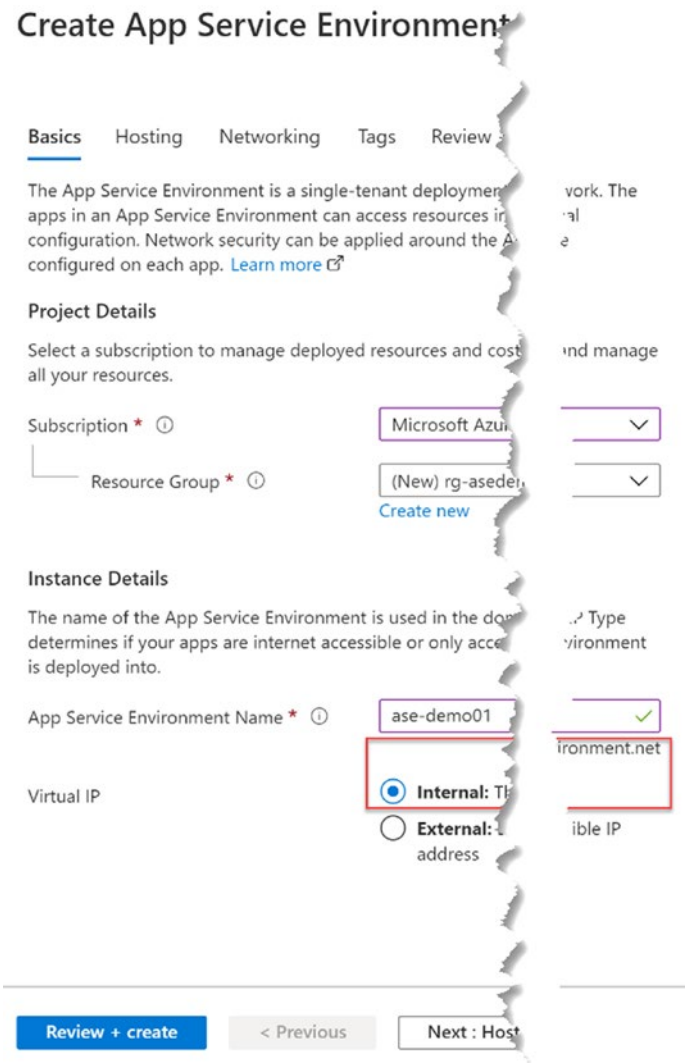

Figure 10-3. *Creating ASE basics*

If you switch virtual IP to external, notice the change in address of the ASE suffix (see Figure 10-4). Let's proceed with External for virtual IP.

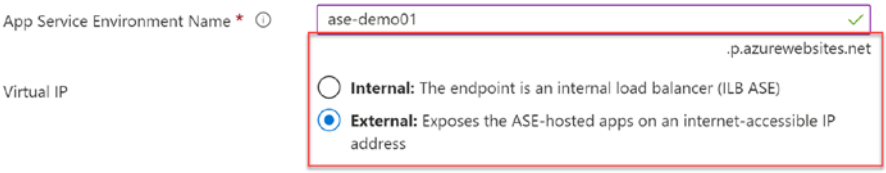

Figure 10-4. *External virtual IP*

In the hosting page, you can select to deploy the ASE to dedicated hosts. Normally ASE is deployed on VMs, which are provisioned on multitenant hypervisors. If you choose to enable dedicated hosts it will deploy on dedicated hardware. However, zone redundancy would not be available for dedicated hosts, and dedicated hosts always be deployed as a pair to ensure redundancy. While deploying the ASE in normal mode you can choose to deploy as a zone redundant (see Figure 10-5).

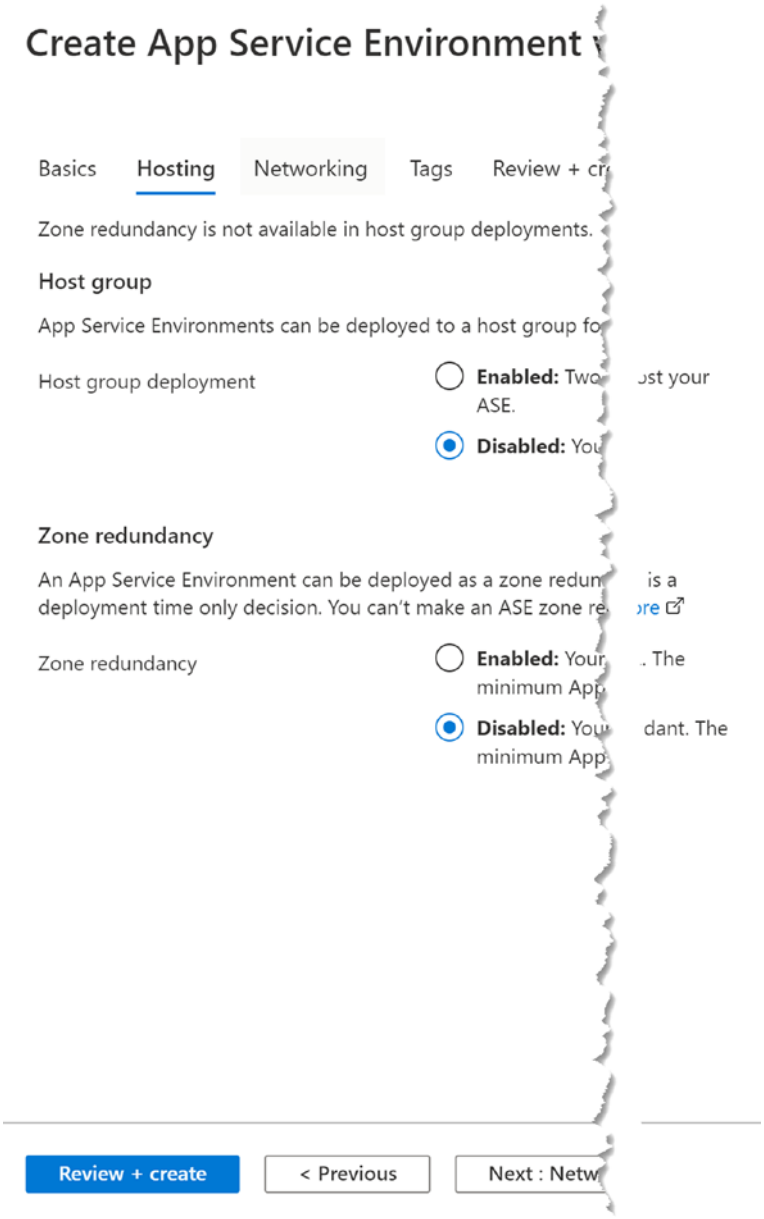

Figure 10-5. *ASE hosting*

We can create a new virtual network for the ASE. You need to provide a name and region. The address block will be automatically selected (see Figure 10-6).

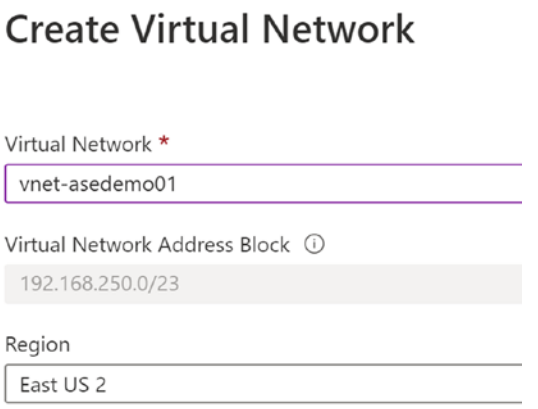

Figure 10-6. *ASE vNet*

We need to define a subnet name as well for the ASE. In the subnet the address block will be with /24 CIDR providing 256 addresses in the subnet of the ASE (see Figure 10-7).

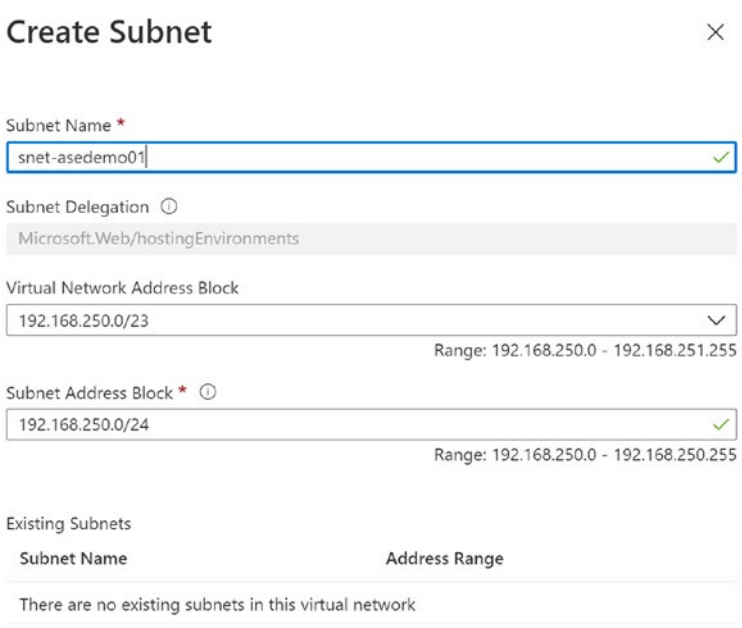

Figure 10-7. *Subnet for ASE*

You can provide tags for the ASE of required and review all configurations before creating the ASE. Click on the create button to create the ASE (see Figure 10-8).

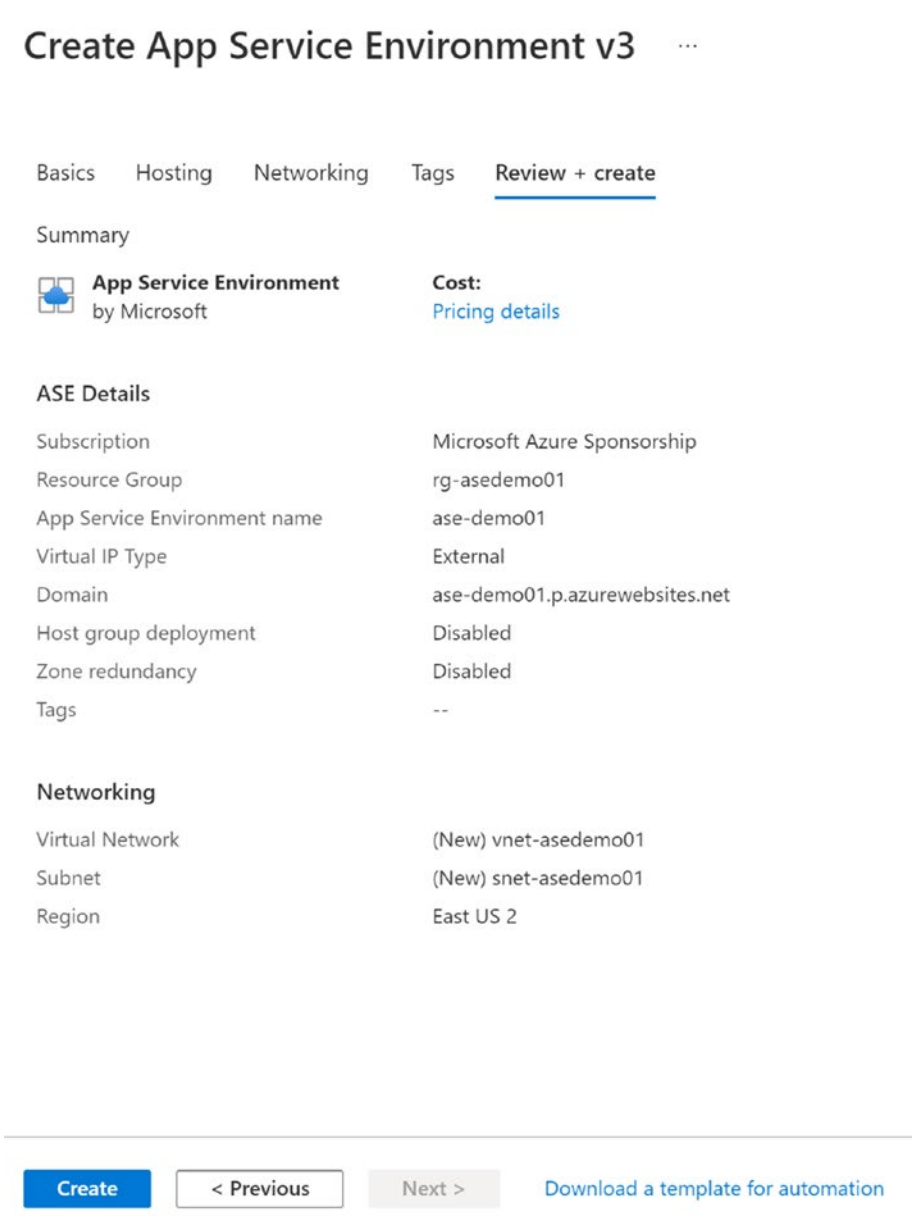

Figure 10-8. *Creating ASE*

It might take an hour or two to get your ASE deployment to complete. Let's explore the created ASE.

In the IP addresses blade, we can see that the inbound IP address and the outbound IP addresses are defined and virtual network and subnet association are visible (see Figure 10-9).

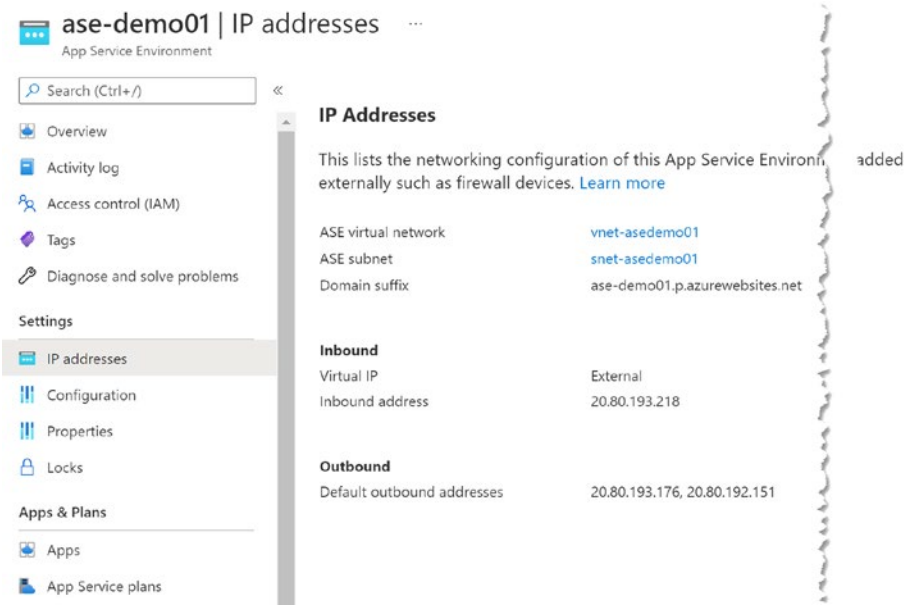

Figure 10-9. *ASE IP addresses*

If we check the Apps or App Service plans blades, we will see that no apps or plans have been added yet. The same information is shown in the overview page of the ASE (see Figure 10-10).

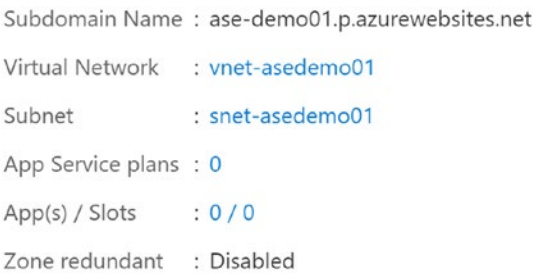

Figure 10-10. *ASE overview*

If we have a look at the resource group where we have set up the ASE, we can see that the ASE and the virtual network are available as resources (see Figure 10-11).

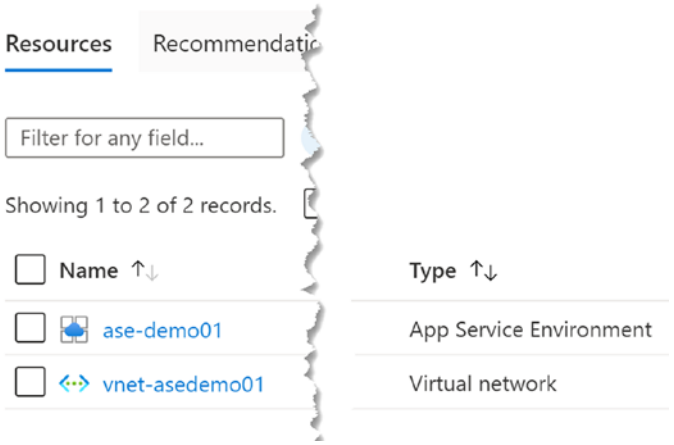

Figure 10-11. *Resources*

We have discussed the steps in setting up an ASE in this lesson.

Lesson 10.2: Creating Apps in App Service Environment

In the previous lesson we have created an ASE. The next step is to understand how we can deploy app service plans and apps to the ASE. If you explore the apps or app service plan blades on ASE, you will not see a way to add your apps or plans to ASE from those blades (see Figure 10-12).

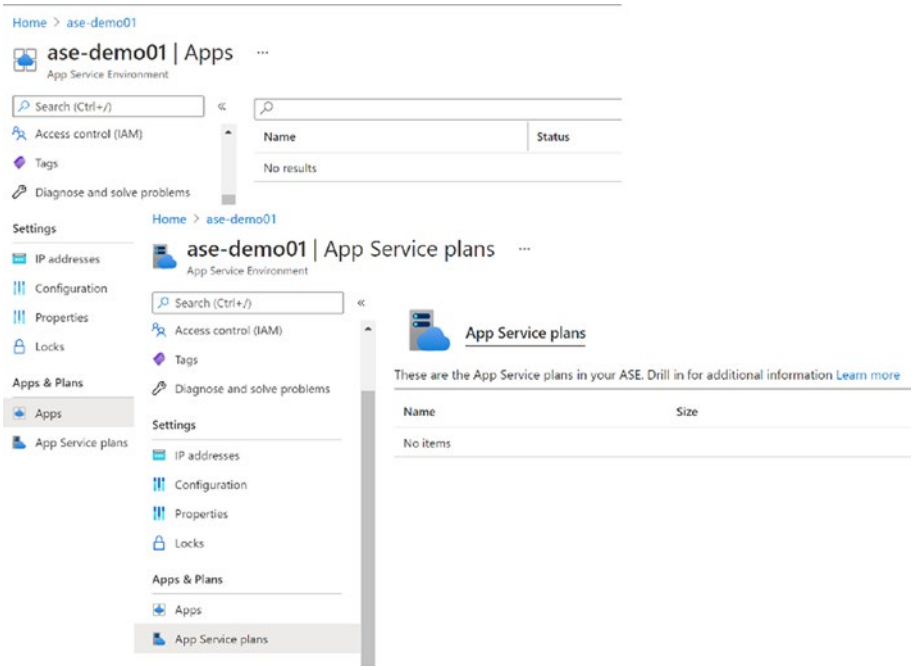

Figure 10-12. *App service plans and apps in ASE*

Let's get started with creating an app service plan in the ASE to understand the steps. Search for an app service plan and click on app service plans in the Azure portal (see Figure 10-13).

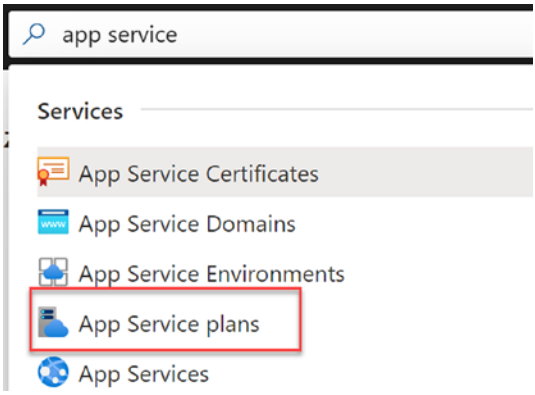

Figure 10-13. *Searching app service plans*

Click create to get started with app service plan creation (see Figure 10-14).

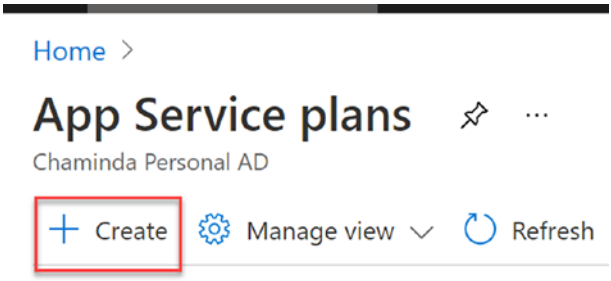

Figure 10-14. *Creating app service plan*

In the app service plan create window, instead of selecting the general Azure region, you need to select your ASE as the region. The resource group can be the same resource group your ASE is created in, or it can be a different resource group within the subscription of your ASE (see Figure 10-15).

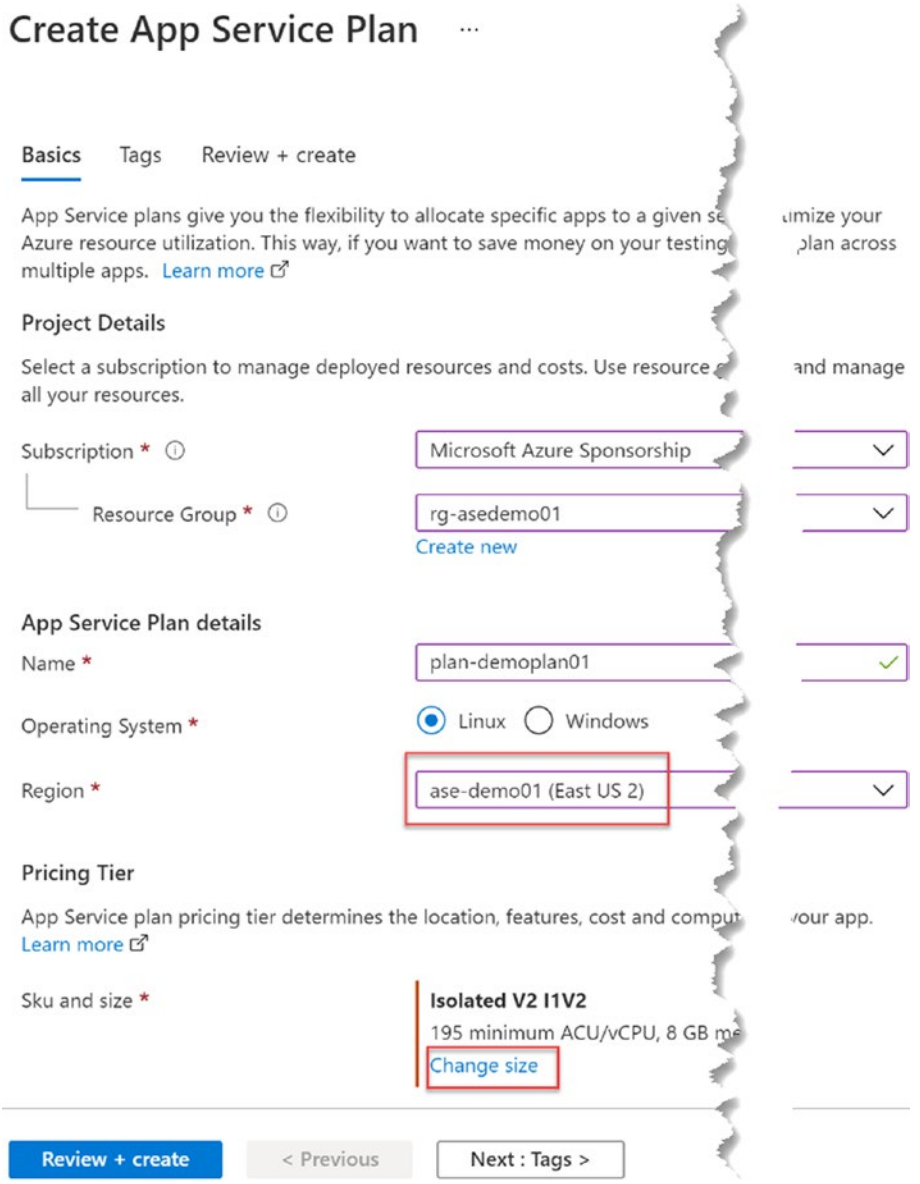

Figure 10-15. *Planning in ASE*

Click on change size in the app service plan create window (see Figure 10-15), and you will be able to see you are only allowed to select the pricing options in Isolated plans, because ASE is an isolated environment (see Figure 10-16).

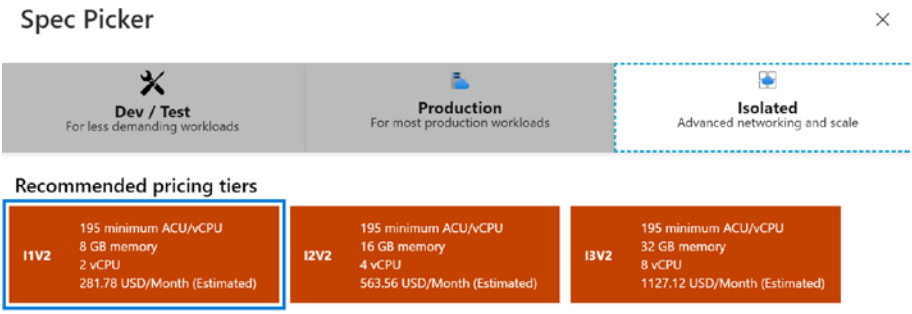

Figure 10-16. *ASE plan sizes*

Proceed with your review and create the plan. You will be able to see the plan appear in your ASE (see Figure 10-17).

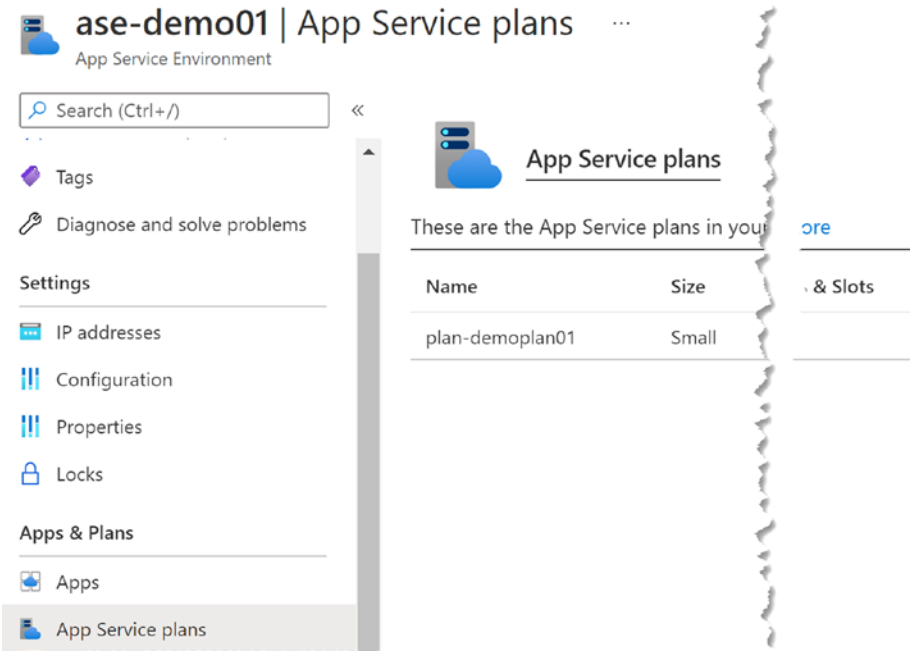

Figure 10-17. *App service plan in ASE*

Now let's see how we can create an app in ASE. You can even create the app in a different resource group from ASE resource group within the subscription. Similar to the app service plan, you need to select the region as ASE in the new app. Once you select ASE as the region you can utilize the previously created app service plan in ASE, or create a new plan with a different isolated size if necessary (see Figure 10-18).

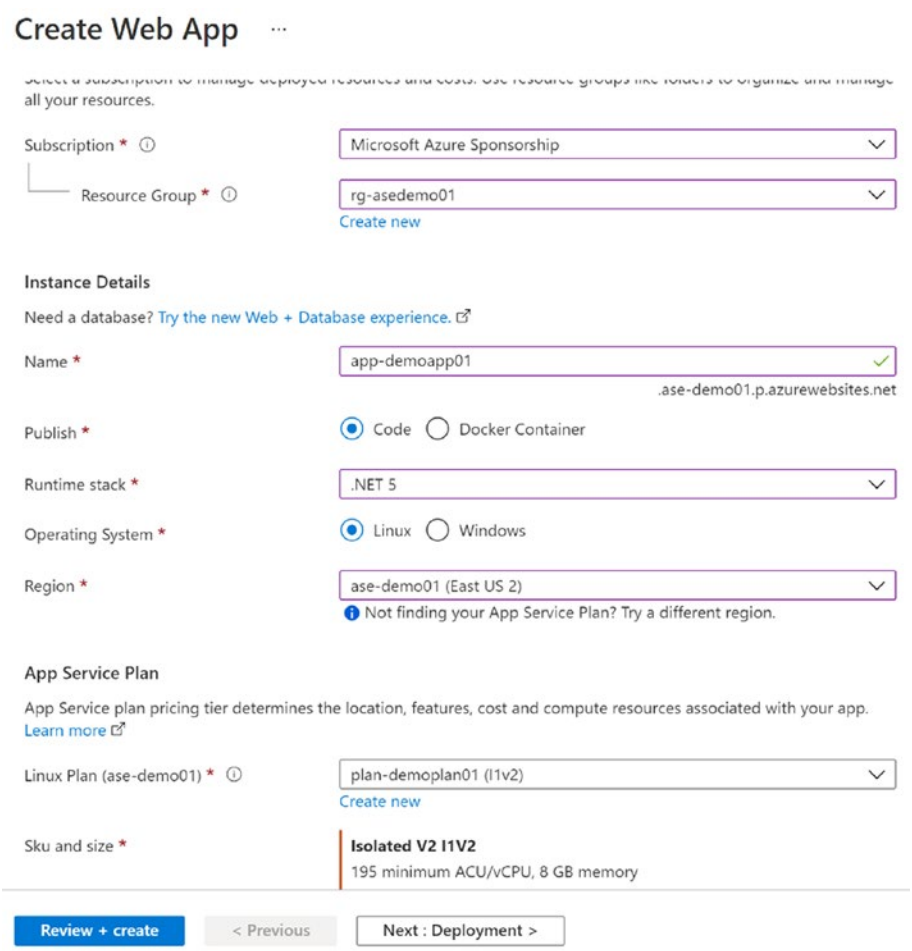

Figure 10-18. *Creating web app in ASE*

Once the app is created, you can see that it is available in the ASE (see Figure 10-19).

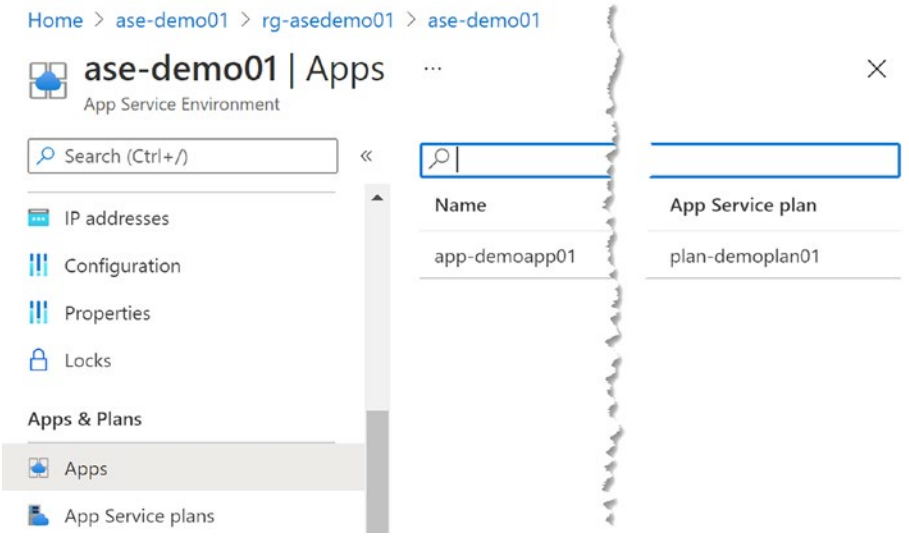

Figure 10-19. *App in ASE*

We have explored the steps to add an app service plan and web app to ASE. You can even add a function app using similar steps to the app service environment. In a function app create page, you can select the ASE as the region (see Figure 10-20).

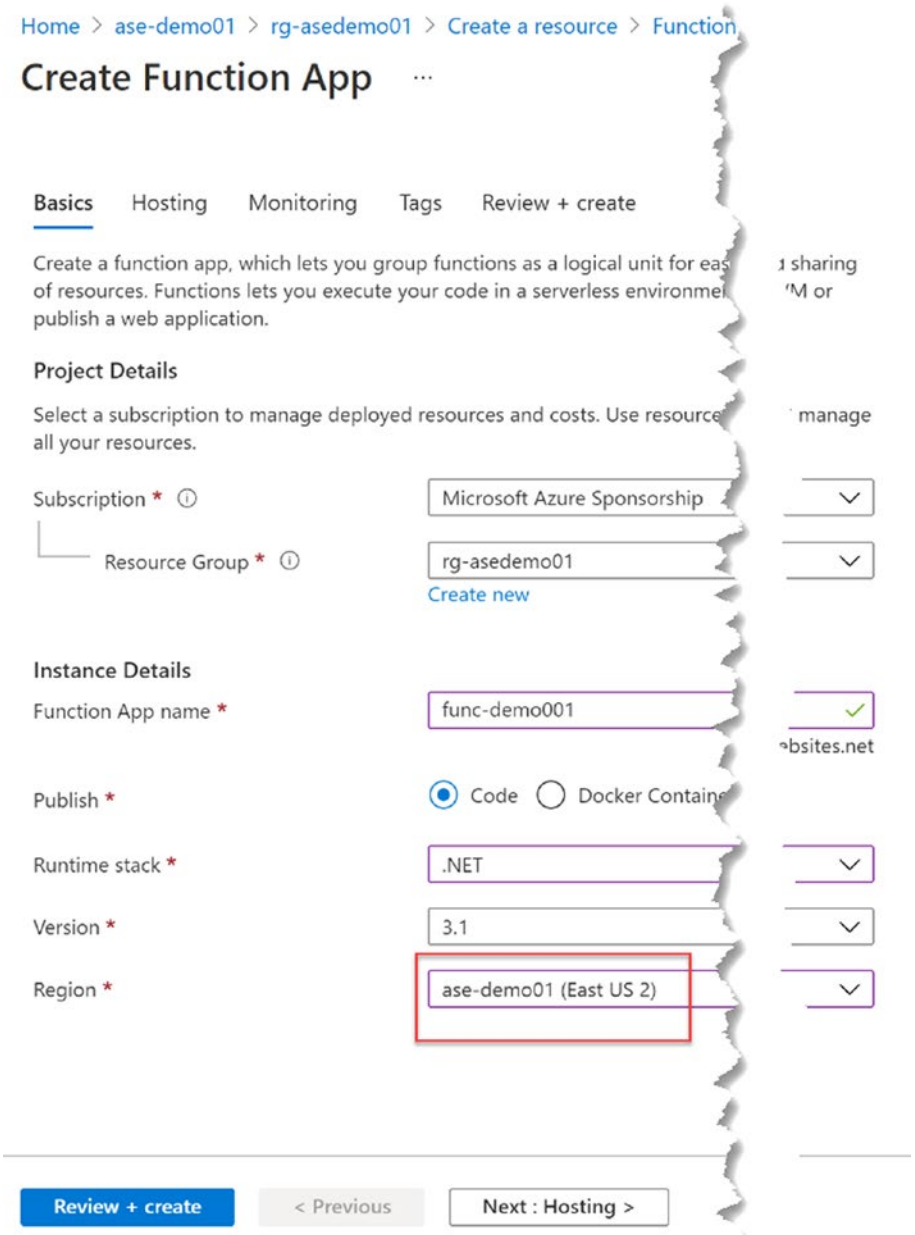

Figure 10-20. *Creating function app in ASE*

In the hosting tab you can select the existing app service plan that is in the ASE, or create a new plan with isolated pricing (see Figure 10-21).

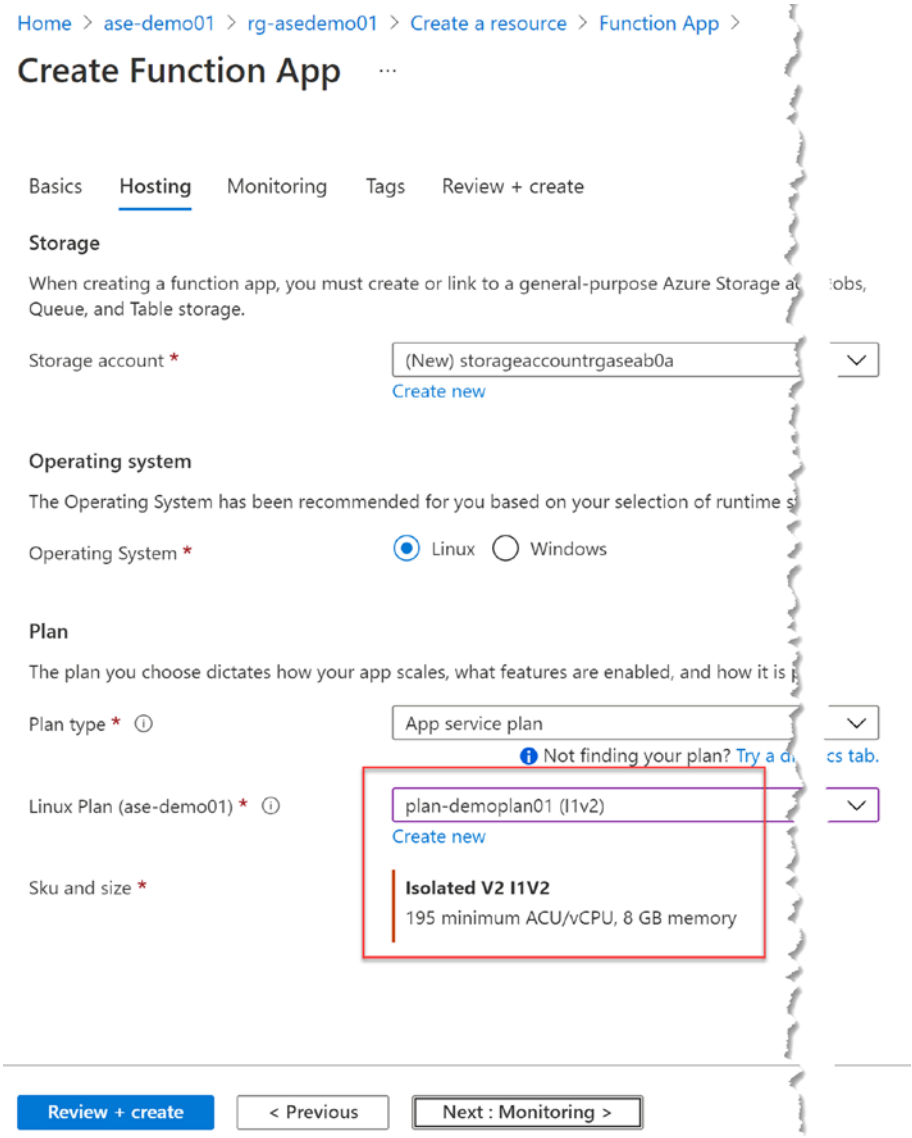

Figure 10-21. *Function app hosting in ASE*

The created function app is also available in the Apps list of the ASE (see Figure 10-22).

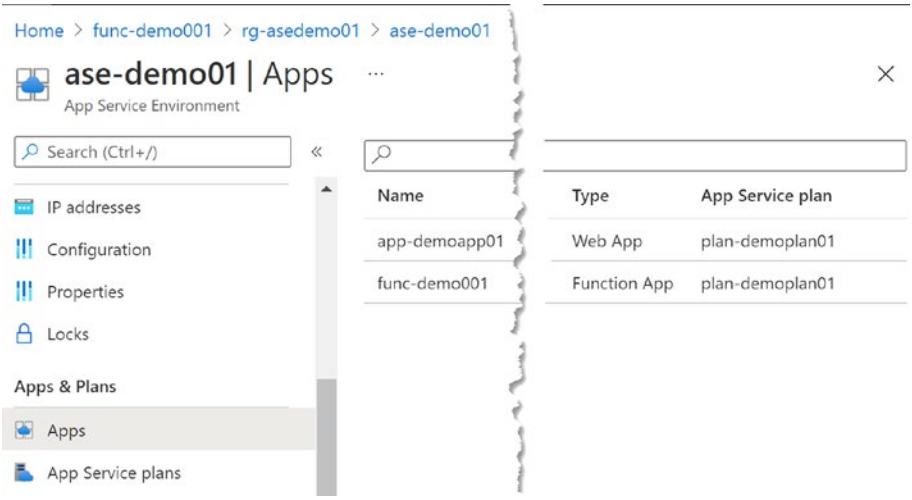

Figure 10-22. *Apps in ASE*

In this lesson, we have explored how to create apps in the app service environment.

Lesson 10.3: Deploying Apps in App Service Environment

In the previous lesson we have created apps in an ASE. Because an ASE is an isolated environment deploying applications, how we should be deploying applications to the app service environment is something worth exploring. In this lesson let's try to understand the steps in deploying a web app to an app hosted in ASE.

You are allowed to deploy to an app in ASE only within the virtual network of the ASE. Therefore, it is mandatory that your deployment

pipeline actions are getting executed within the virtual network of ASE. In order to achieve this requirement, you need to deploy a self-hosted deployment agent to your ASE virtual network. You can set up a virtual machine in a subnet of the virtual network of the ASE. Then that virtual machine can be deployed with, for example, Azure DevOps hosted agent if you are using Azure DevOps to create your deployment pipelines. If it is GitHub, you can set up a self-hosted GitHub action runner in the virtual machine (see Figure 10-23).

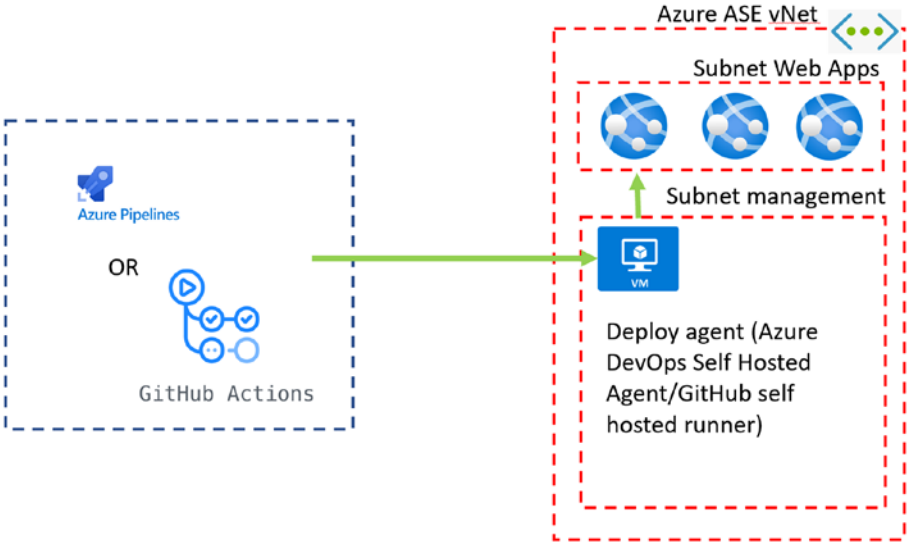

Figure 10-23. *Application deployment for ASE apps*

You need to allow the self-hosted agent or runner machine to have access to Azure DevOps or GitHub in order to obtain the packages to deploy and instructions to deploy based on the pipeline setup for deployment. The virtual machine in the same virtual network of the ASE will be able to reach the deployment URLs of the web apps to deploy the packages supplied by the deployment pipeline.

As you can see from the level of isolation required even for deploying to Apps in ASE, it is pretty much secure. Unlike public web apps, the web apps in ASE will be fully secured in isolation inside a virtual network by default. If a VPN is made from the ASE virtual network to your corporate network, the web apps in ASE will only be available within your corporate network boundary. For this setup ideally you should deploy the ASE with only an internal IP load balancer, as opposed to what we have done in lesson 2 of this chapter.

Even when you want to expose the web app in ASE publicly, it is recommended that you deploy load balancer with only internal IP and then add an application gateway web application firewall (WAF) to protect and expose it to the public with an additional layer of security.

In this lesson we have discussed the deployment consideration in web apps in an ASE.

Summary

In this chapter we have discussed the fully isolated app service environment deployment option in Azure to deploy web and function apps. We also explored how to set up ASE and set up apps inside ASE. The deployment considerations were described to give you an idea of the secure nature of the ASE-based web apps.

In this book, we have explored the security aspect in Azure PaaS services with all the basic details required to get started. The service security aspects, limitations, and enhancing options to ensure your application and data surety is described throughout this book.