## 

## **SPITECH**

## **Version 0.1**

## **Services**

We have used the below services for SPITECH infrastructure

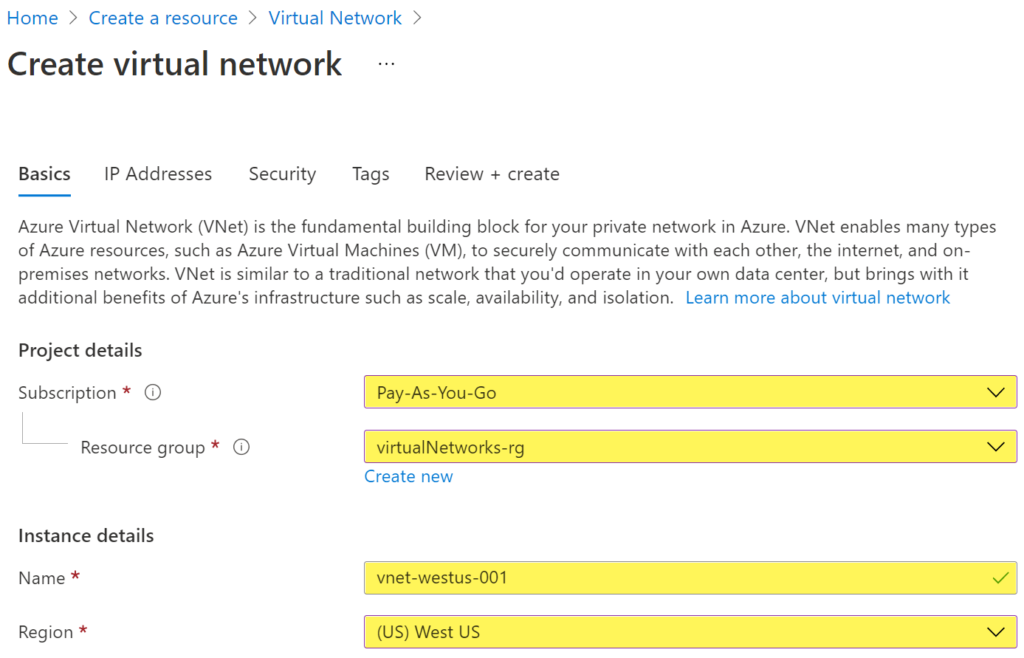
1. Virtual Network
2. Network Security Group
3. AKS
4. Azure Load Balancer
5. API Management
6. Storage account
7. Azure Container Registry
8. Azure SQL Server
9. Azure Key vault
10. Azure Web App
11. Log analytics workspace
12. Application Insight
13. Rabbitmq
14. SEQ
15. Azure DevOps

## **Services Setup**

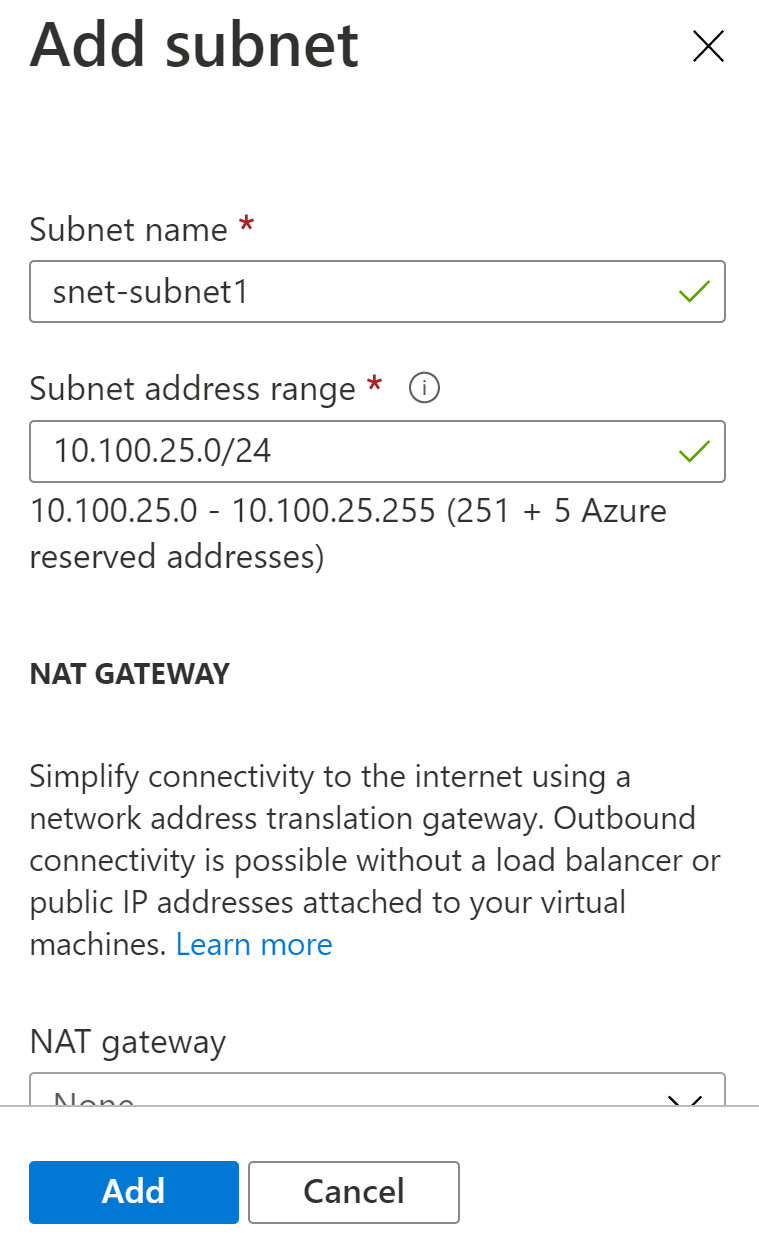
### **Azure Virtual Networks**

To create an Azure Virtual Network using the Azure Portal:

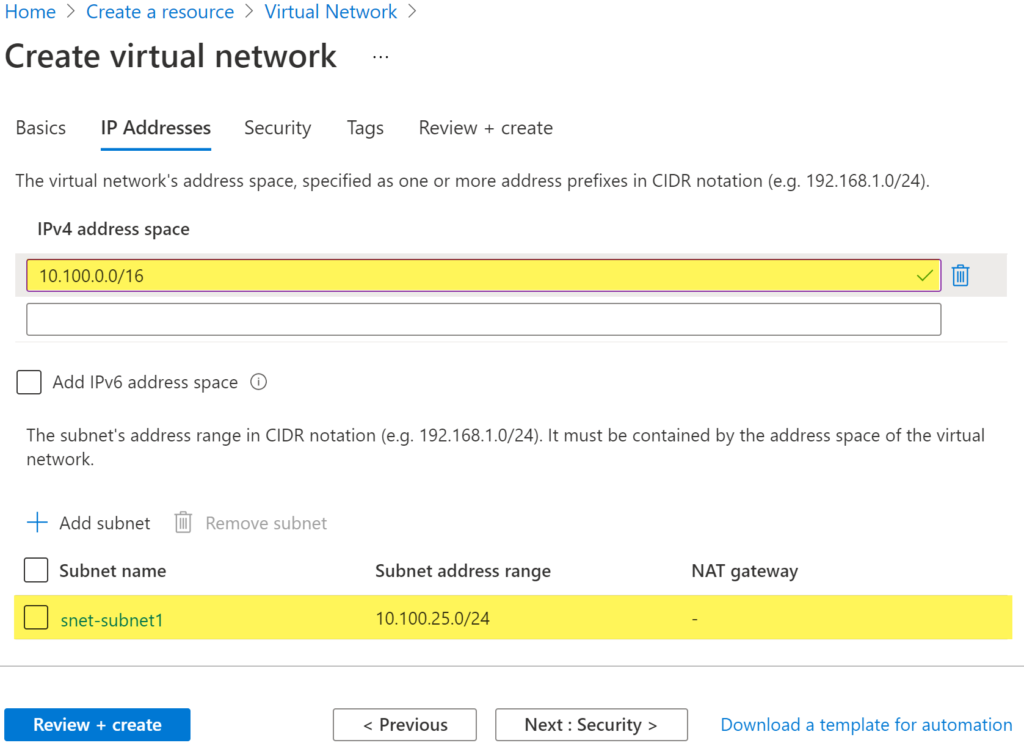
1. Navigate and sign in to the Azure portal.
2. Select Create a resource on the Azure Portal homepage.
3. On the Create a resource page, search the marketplace for virtual networks and select it from the results.
4. On the Virtual Network page, select Create.
5. On the Create virtual network page, configure the information in the Basics tab.
6. Subscription: select the subscription to bill the resource against
7. Resource group: create a new resource group or choose an existing one
8. Name: enter vnet-westus-001
9. Region: select the West US region



1. Select the Next: IP Addresses button at the bottom of the page.
2. In the IPv4 address space section, Azure has pre-populated the address space 10.1.0.0/16. Select this existing address space and change it to 10.100.0.0/16.
3. If you want to add subnets now, select + Add subnet, then enter the subnet name snet-subnet1 and an address range of 10.50.25.0/24.

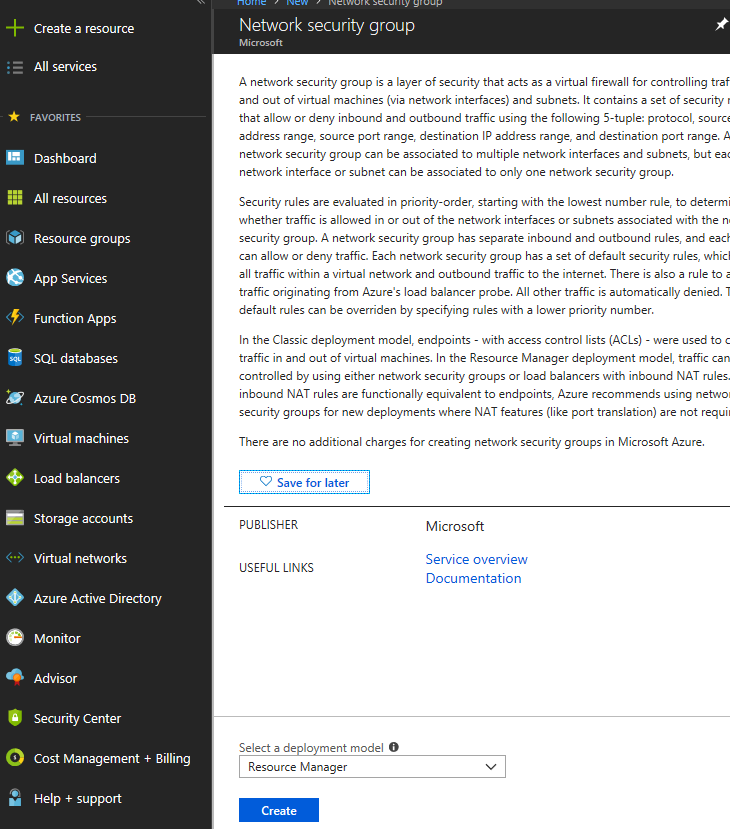


Review the address space and subnets, then select Review + create, then Create after the portal validates the configuration.

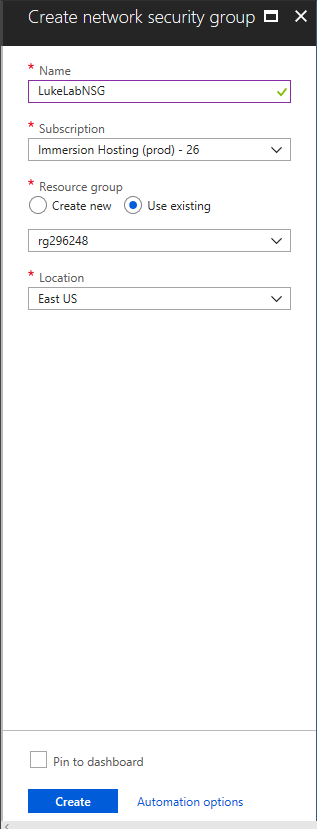


### **Network Security Group**

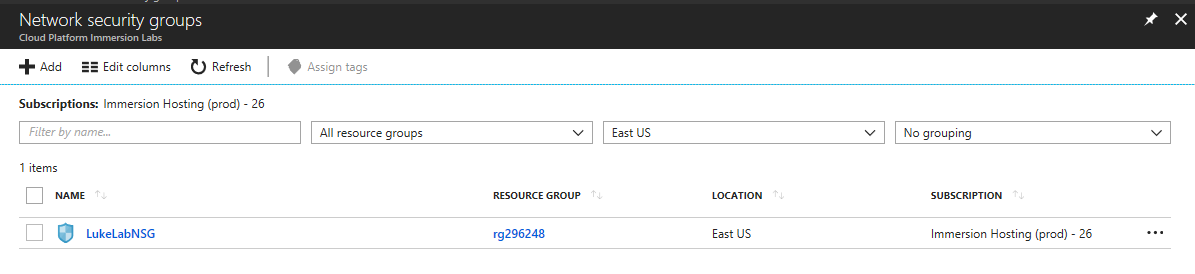
1. Log in to the Azure portal at https://portal.azure.com.
2. In the top-left corner of the portal, select + Create a resource.
3. Select Networking, then select network security group.



1. Enter, or select, the following information

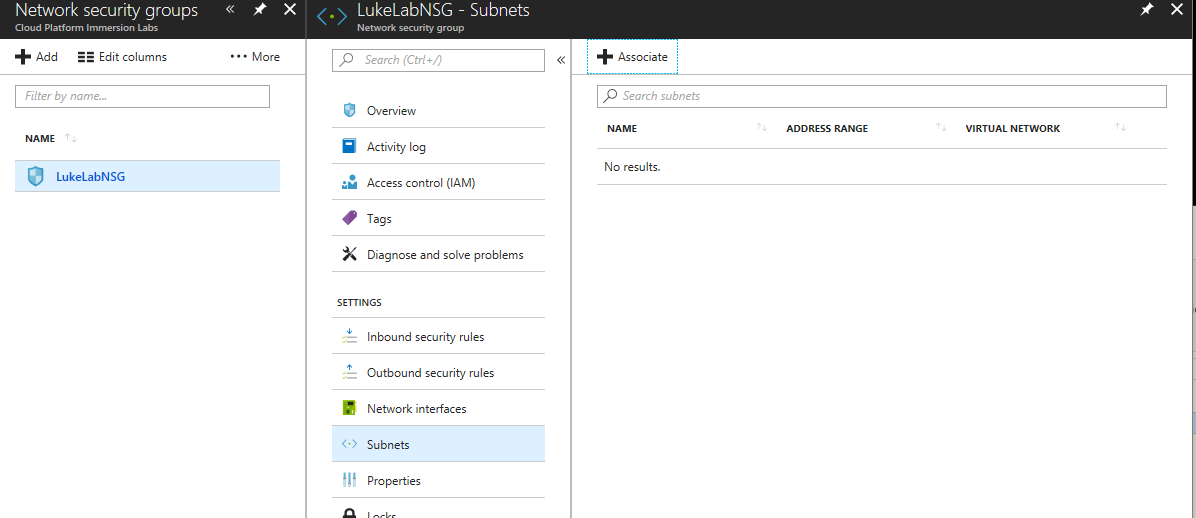


1. Network security group successfully created

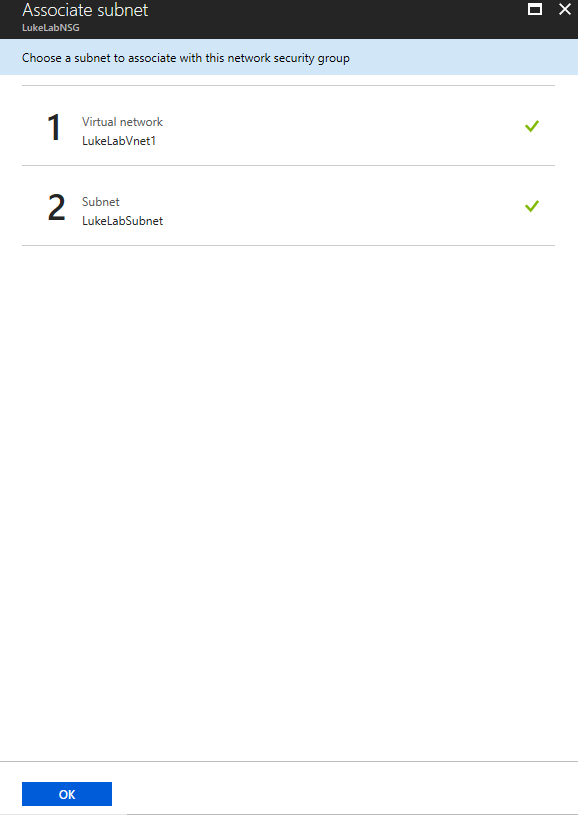


**Associate network security group to subnet**

1. In the Search resources, services, and docs box at the top of the portal, begin typing MyNSG. When MyNSG appears in the search results, select it.
2. Under SETTINGS, select Subnets and then select + Associate

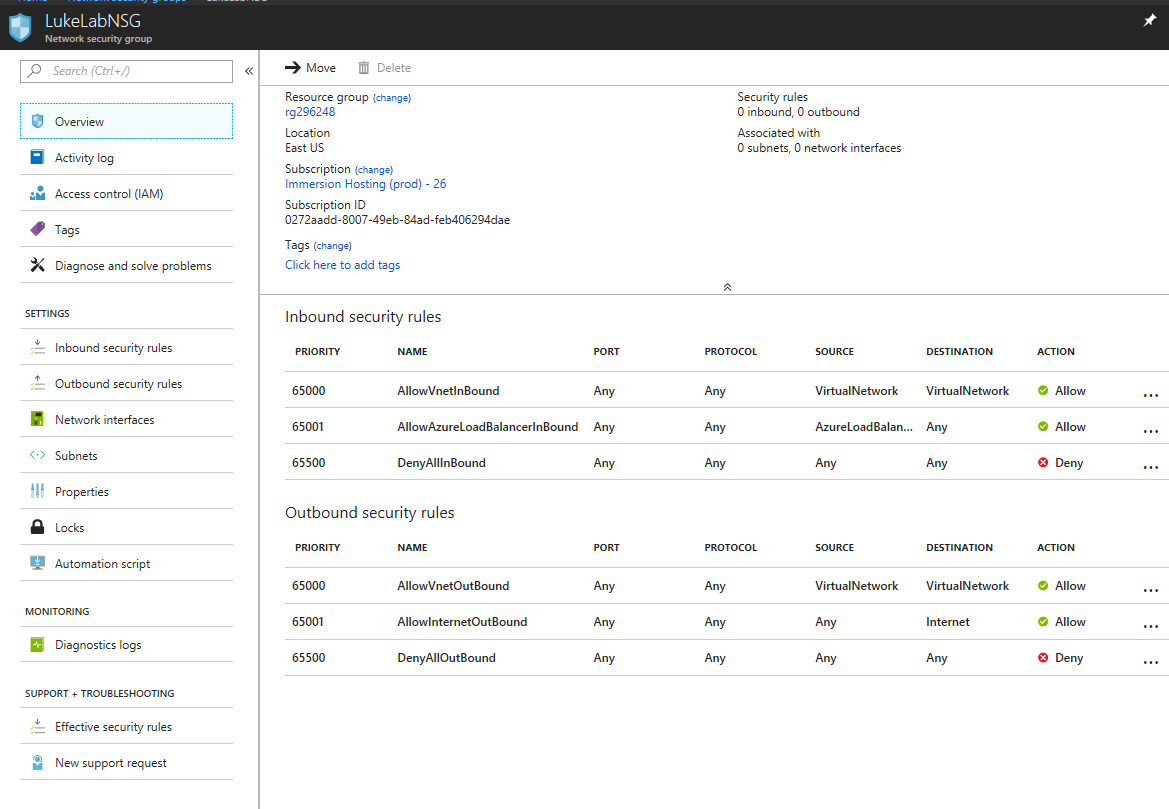


1. Under Associate subnet, select Virtual network and then select MyVNET. Select Subnet, select FrontEndVnet and then select OK.



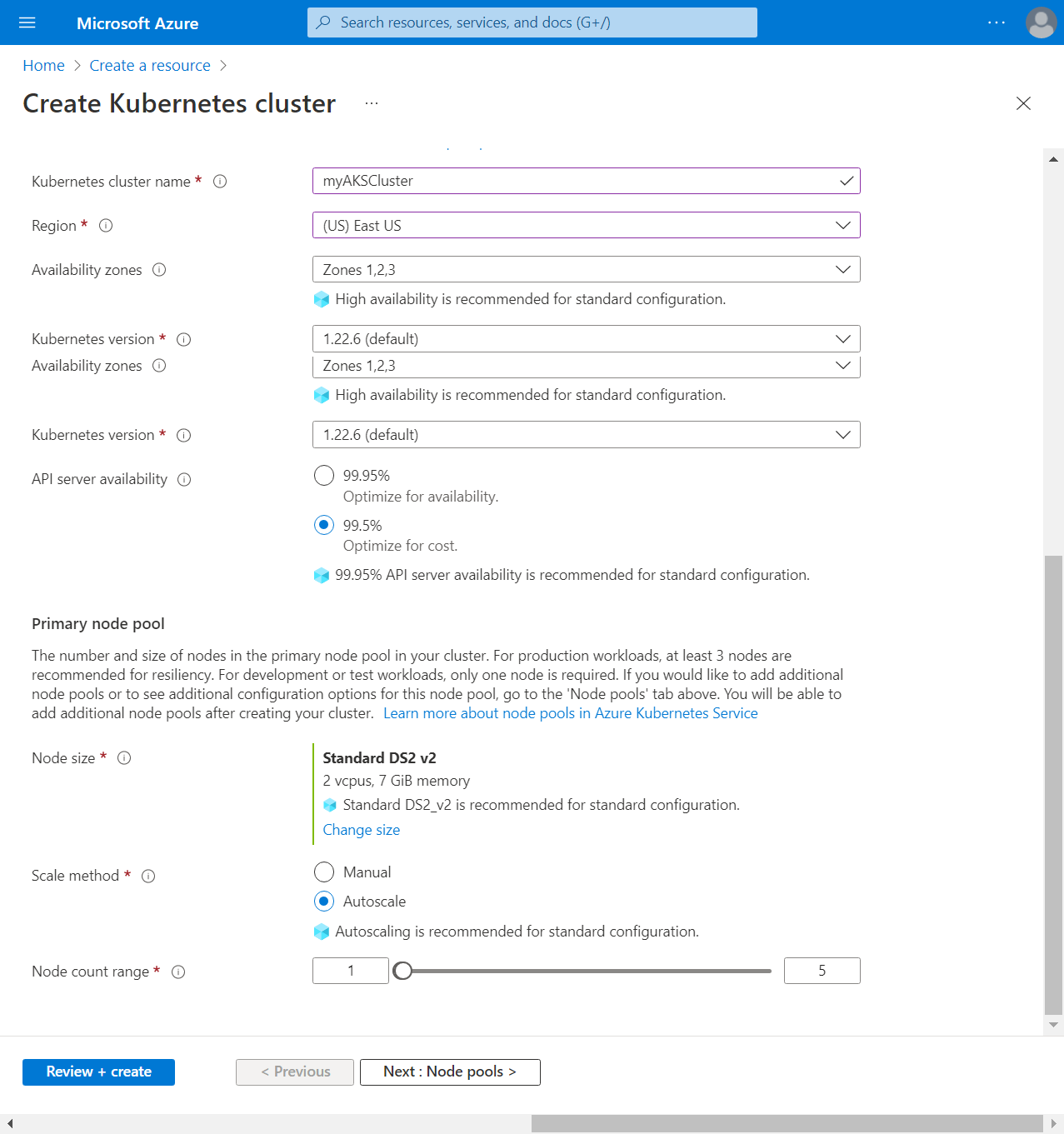
**Create security rules**

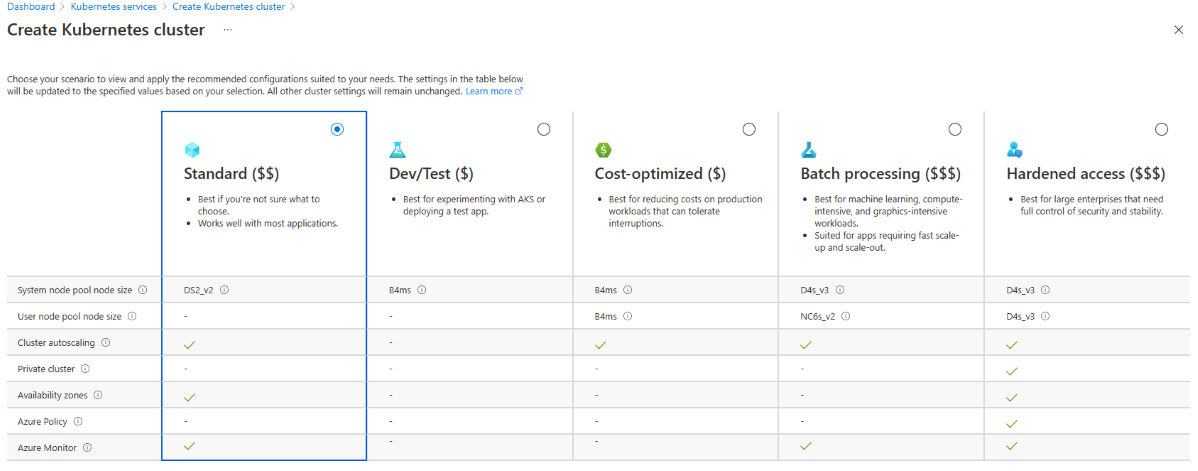
1. Under SETTINGS, select Inbound security rules and then select + Add,
2. Create a security rule that allows ports 80, 443 and 3389
3. Under Add inbound security rule, enter the following values

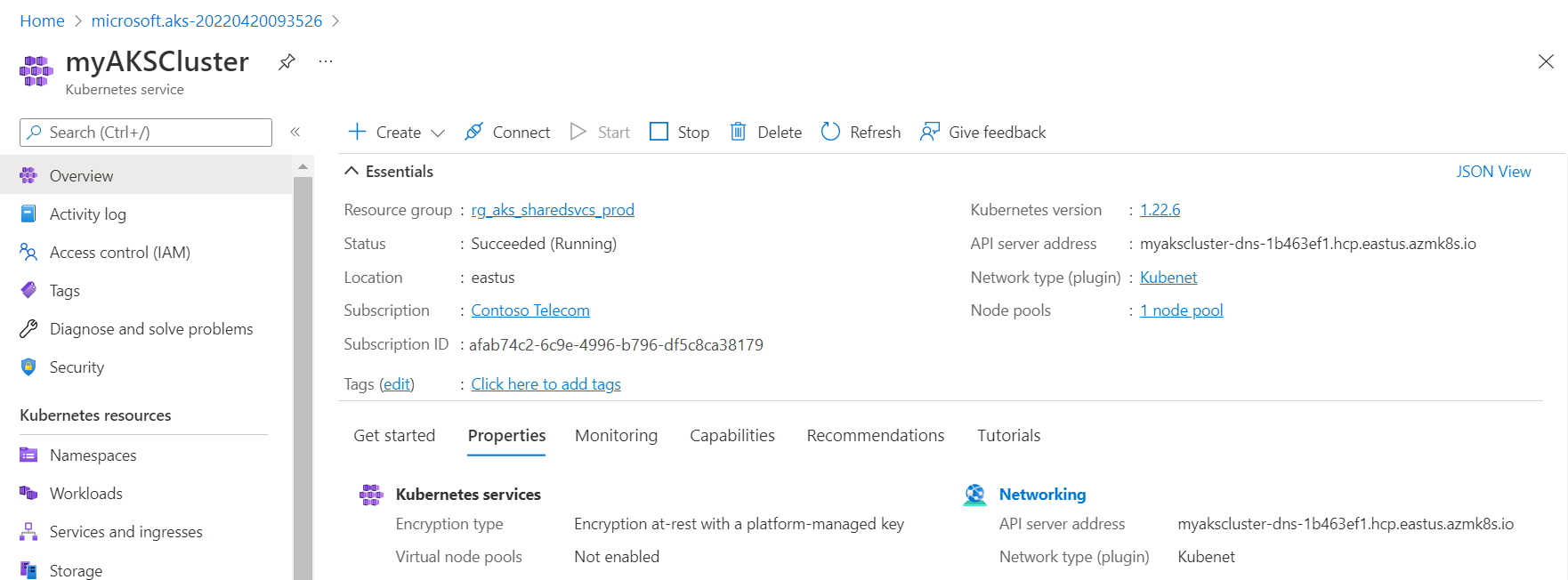


### **Create an AKS cluster**

1. Sign in to the [Azure portal](https://portal.azure.com/).
2. On the Azure portal menu or from the Home page, select Create a resource.
3. Select Containers > Kubernetes Service.
4. On the Basics page, configure the following options:
   1. Project details:
      1. Select an Azure Subscription.
      2. Select or create an Azure Resource Group, such as myResourceGroup.
   2. Cluster details:
      1. Ensure the Preset configuration is Standard ($$).
      2. Enter a Kubernetes cluster name, such as myAKSCluster.
      3. Select a Region for the AKS cluster, and leave the default value selected for the Kubernetes version.
      4. Select 99.5% for API server availability.
   3. Primary node pool:
      1. Leave the default values selected.



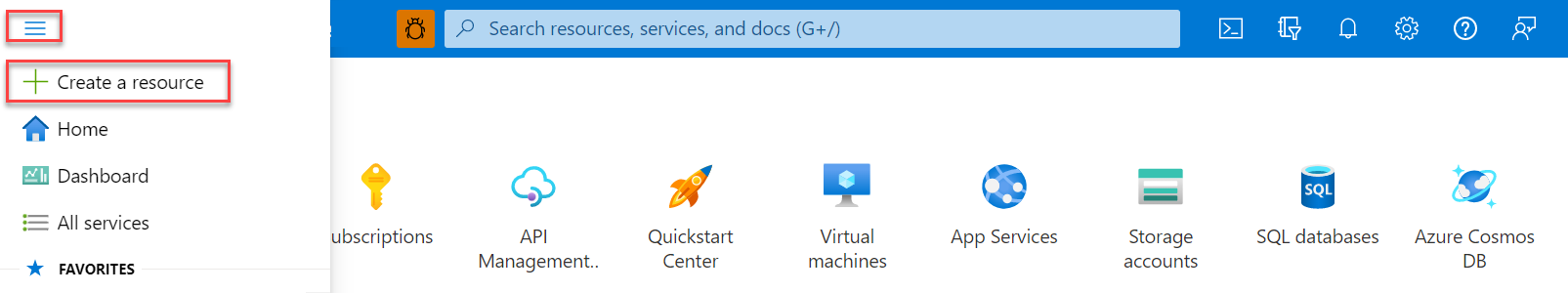
Note: You can change the preset configuration when creating your cluster by selecting Learn more, comparing presets, and choosing a different option. 

1. Select Next: Node pools when complete.
2. Keep the default Node pools options. At the bottom of the screen, click Next: Access.
3. On the Access page, configure the following options:
4. The default value for Resource identity is System-assigned managed identity. Managed identities provide an identity for applications when connecting to resources that support Azure Active Directory (Azure AD) authentication.
5. The Kubernetes role-based access control (RBAC) option is the default value to provide more fine-grained control over access to the Kubernetes resources deployed in your AKS cluster.
6. By default, Basic networking is used, and [Container insights](https://docs.microsoft.com/en-us/azure/azure-monitor/containers/container-insights-overview) is enabled.
7. Click Review + create. When you navigate to the Review + create tab, Azure runs validation on the settings that you have chosen. If validation passes, you can proceed to create the AKS cluster by selecting Create. If validation fails, then it indicates which settings need to be modified.
8. It takes a few minutes to create the AKS cluster. When your deployment is complete, navigate to your resource by either:
9. Selecting Go to resource, or
10. Browsing to the AKS cluster resource group and selecting the AKS resource. In this example, you browse for myResourceGroup and select the resource myAKSCluster.  
    

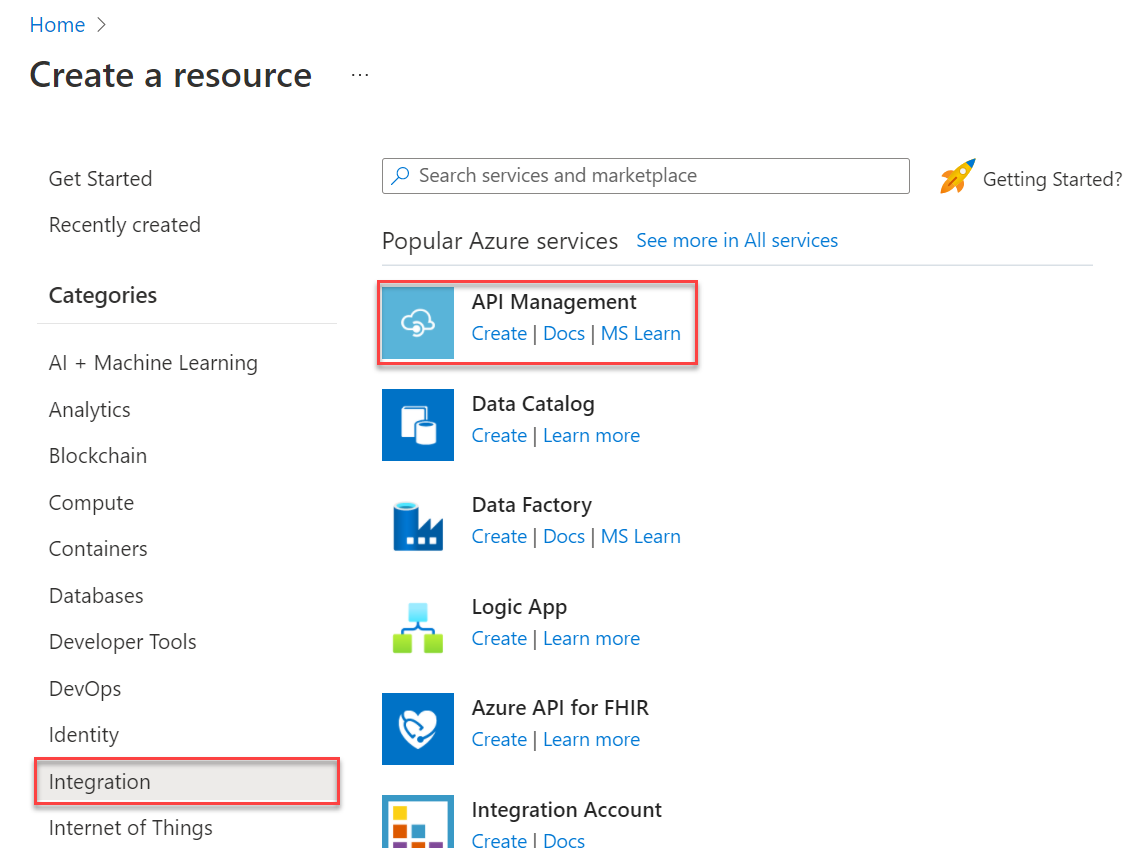
### **Azure Load Balancer**

The azure internal load balancer will automatically be created by the AKS when we deploy a service a load balancer type.

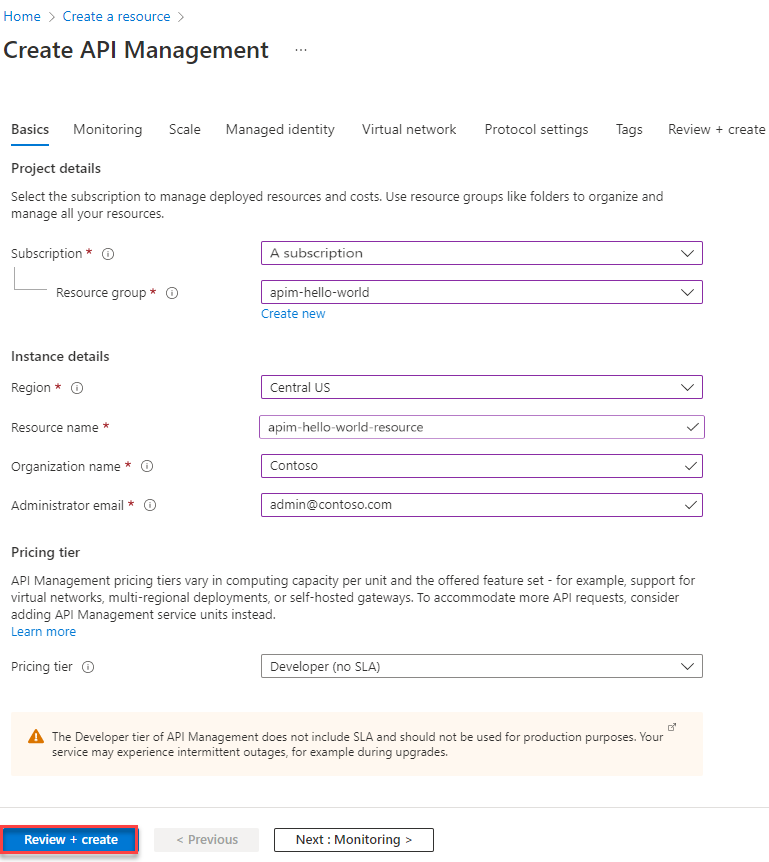
### **Create an API Management service**

From the Azure portal menu, select Create a resource. You can also select Create a resource on the Azure Home page.  


On the Create a resource page, select Integration > API Management.



In the Create API Management page, enter settings.



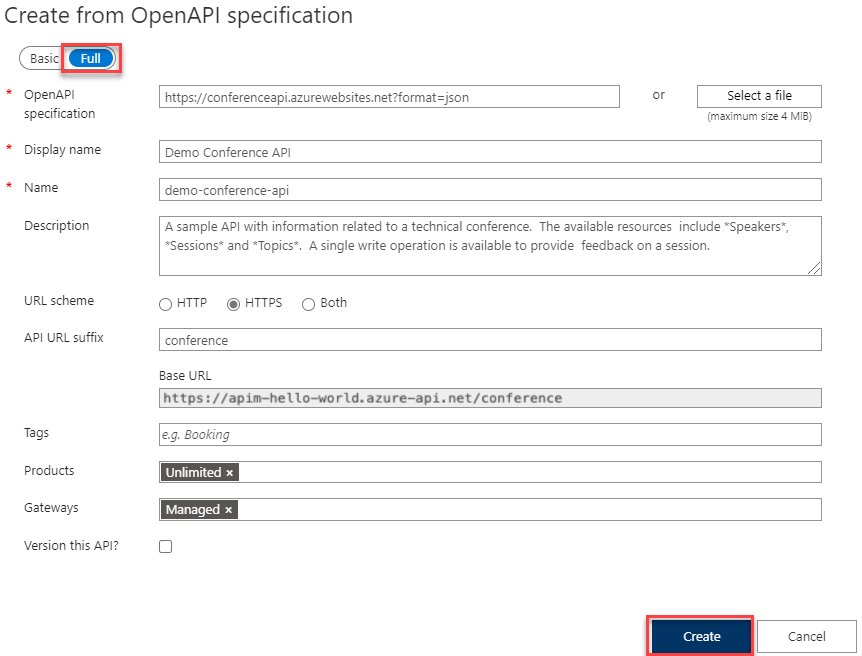
Select Review + create.

Tip: It can take 30 to 40 minutes to create and activate an API Management service in this tier.

### **Import and publish a backend API**

This section shows how to import and publish an OpenAPI Specification backend API.

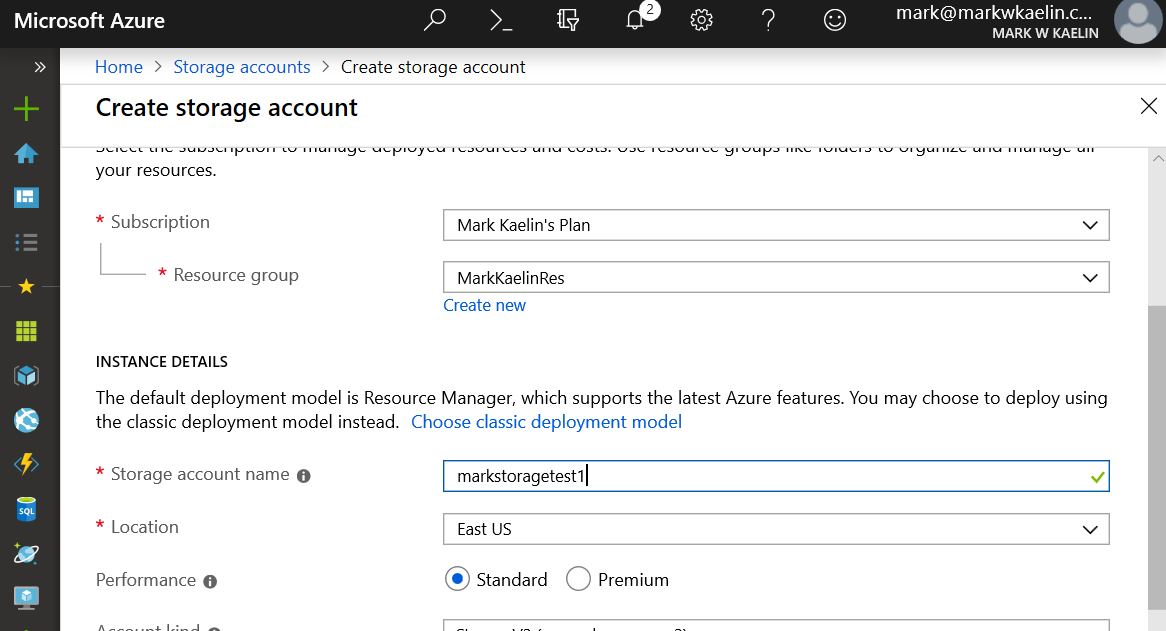
1. In the left navigation of your API Management instance, select APIs.
2. Select the OpenAPI tile.
3. In the Create from OpenAPI specification window, select Full.
4. Enter the values from the following table.  
   You can set API values during creation or later by going to the Settings tab.

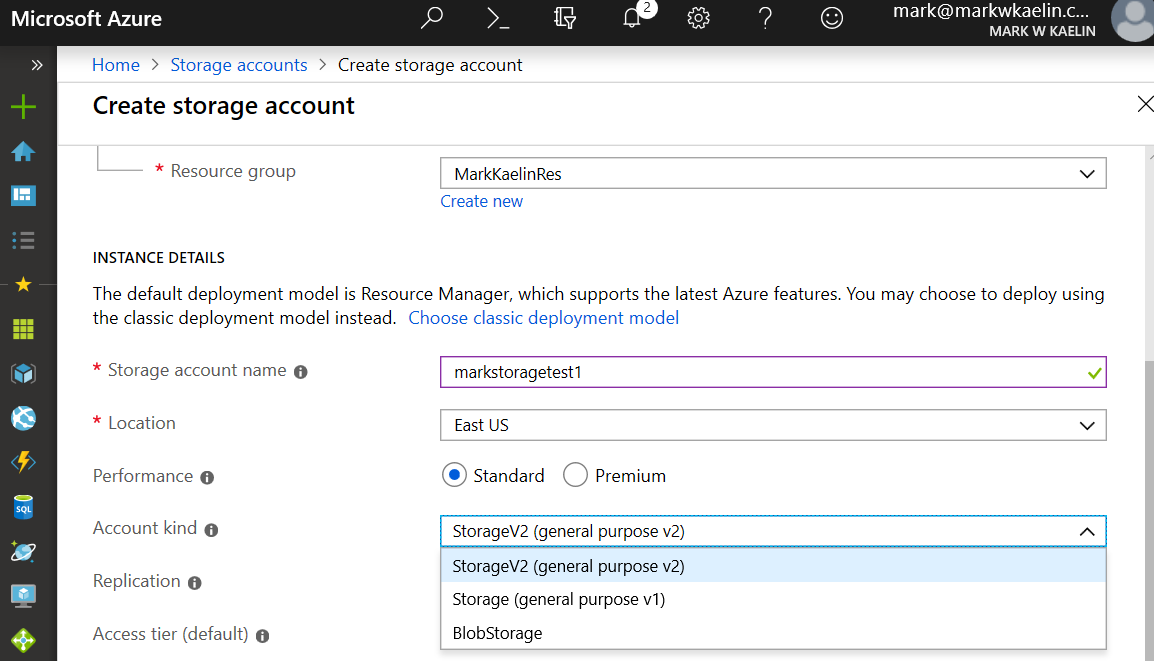
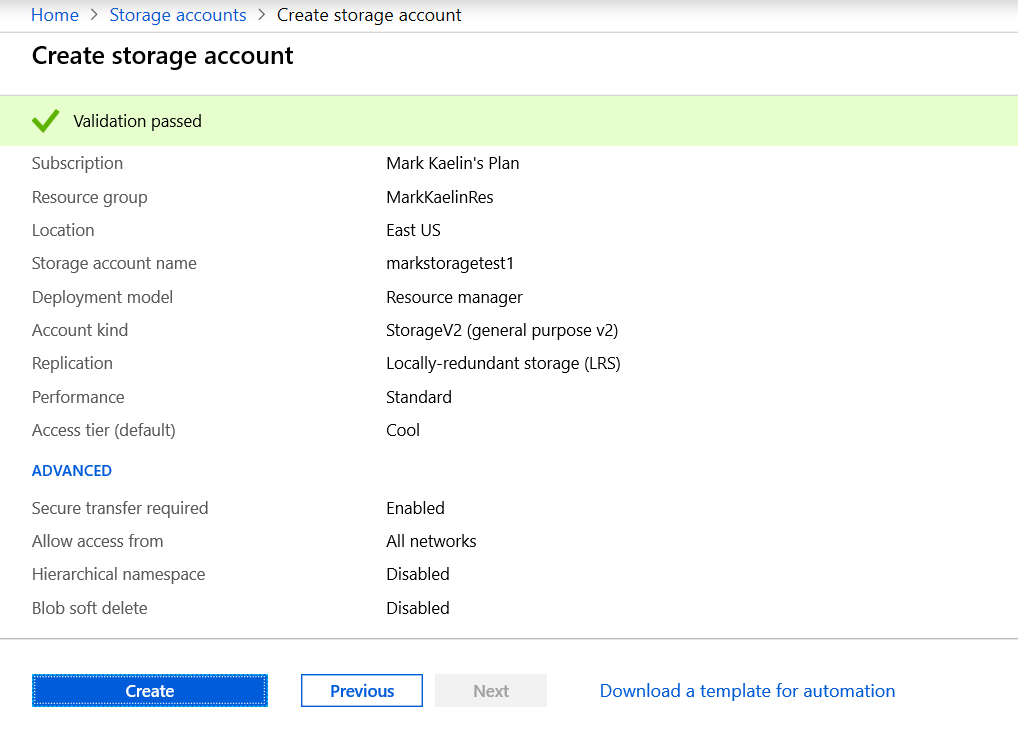


### **Create an Azure storage account**

The outlined procedure assumes you have an active Microsoft Azure account.

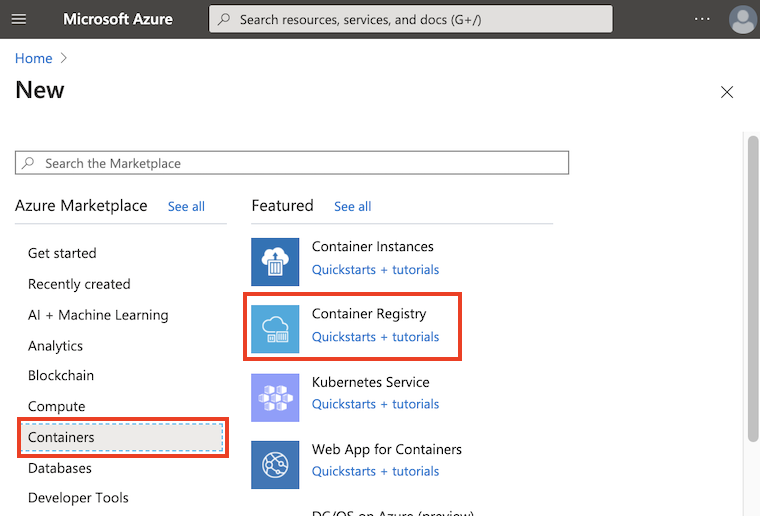
1. Log in to your Azure account and navigate to the Home page.
2. Click on the storage accounts link in the left-hand navigation bar to reach the screen 
3. Click the + Add button to start a new storage account.
4. Choose the appropriate subscription account and resource group, and then chose a unique name for your new storage account. Remember, you will likely have to type this name to access the storage account, so make it something reasonable.



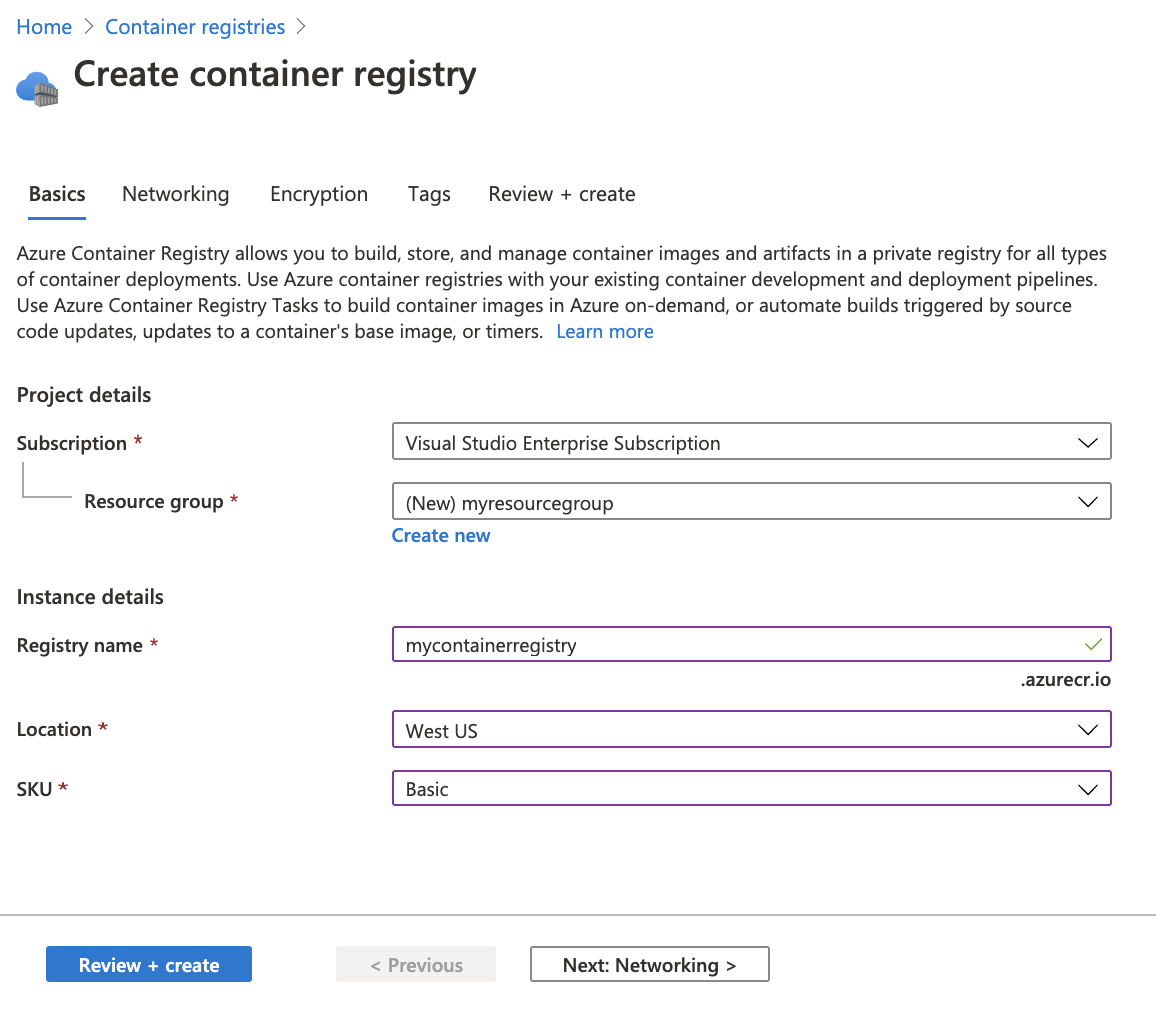
1. Next, choose the location for your server based on your geographical location (closer is better), and then choose a level of service, standard, or premium.
2. The standard setting should be fine for most use cases. For the account kind configuration setting, Microsoft recommends the StorageV2 setting for most applications.
3. Note: StorageV1 is deprecated, and you should only use BlobStorage if your business needs it.
4. 
5. The next configuration setting is for replication, and, unlike the previous settings, this decision can tremendously increase the overall cost of your storage account. If the data you will be storing is business-critical and cannot be lost or inaccessible at any time, then you should opt for the more reliable Geo-redundant storage (GRS) setting. If the data destined for your Azure storage account can be replicated from other sources, then the much less expensive Locally redundant storage (LRS) should be chosen.
6. The last setting on the configuration page allows you to choose an access tier. Choose the Hot setting for data that will be accessed frequently. Likewise, choose the Cool setting for data that will be accessed infrequently.
7. When you have made all of your configuration settings click the Review + Create button to move to the next step. You can review your settings (Figure E) before you click the Create button to complete the process.
8. 
9. Deployment of your new Azure storage account will take a few minutes. When the deployment is complete, you may access your storage account through the Azure Portal.

### **Create a container registry**

1. Select Create a resource > Containers > Container Registry.



1. In the Basics tab, enter values for the Resource group and Registry name. The registry name must be unique within Azure and contain 5-50 alphanumeric characters. For this quickstart create a new resource group in the West US location named myResourceGroup, and for SKU, select 'Basic'.



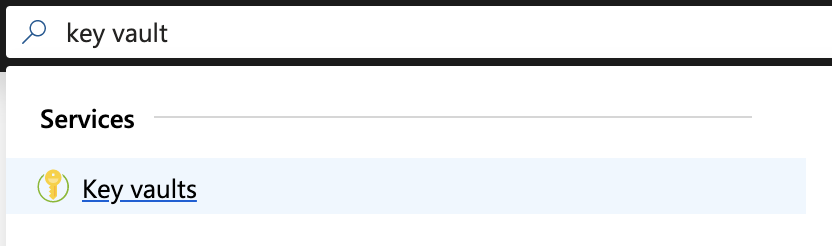
1. Accept default values for the remaining settings. Then select Review + create. After reviewing the settings, select Create.

Tip: In this quickstart, you create a Basic registry, which is a cost-optimized option for developers learning about Azure Container Registry. Choose other tiers for increased storage and image throughput, and capabilities such as connection using a private endpoint.

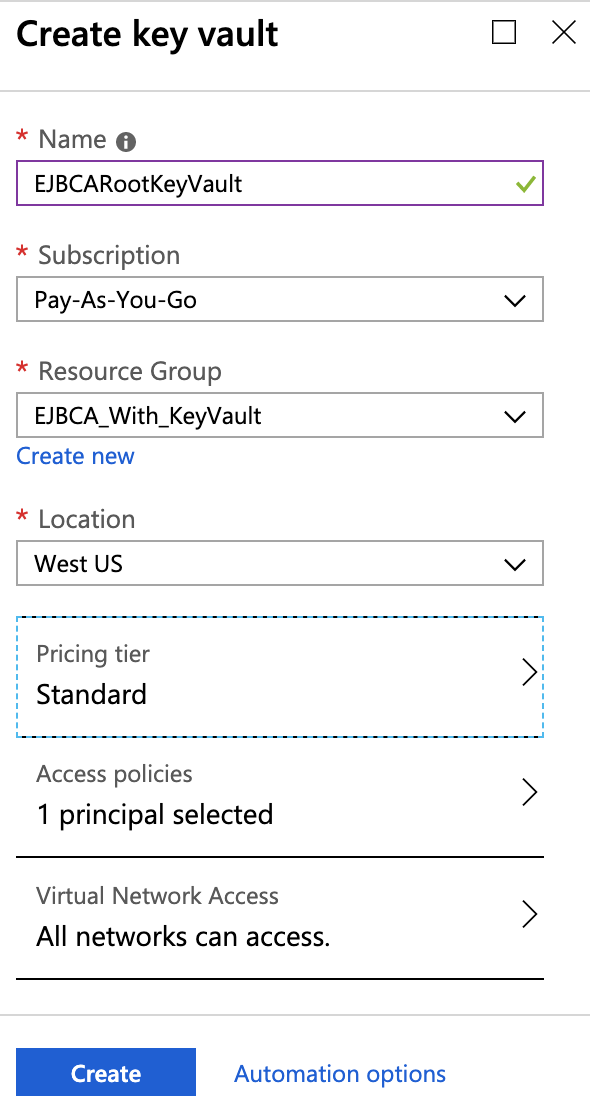
### Creating Azure Key Vault

The following shows how to create the Azure Key Vault.

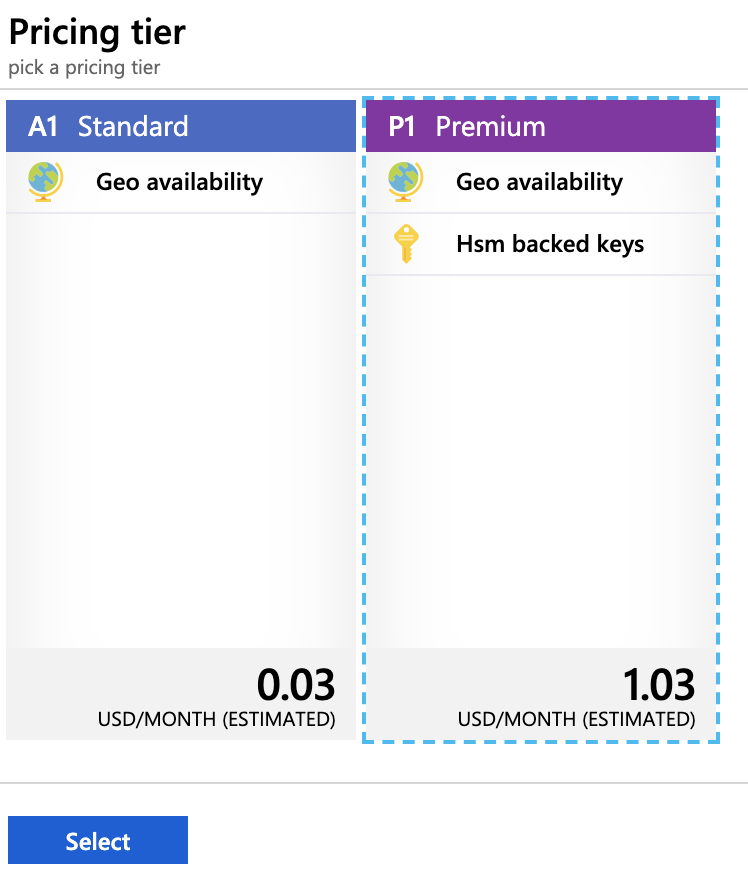
1. Search for Key Vault in the Azure portal and then select Key vaults in the results.



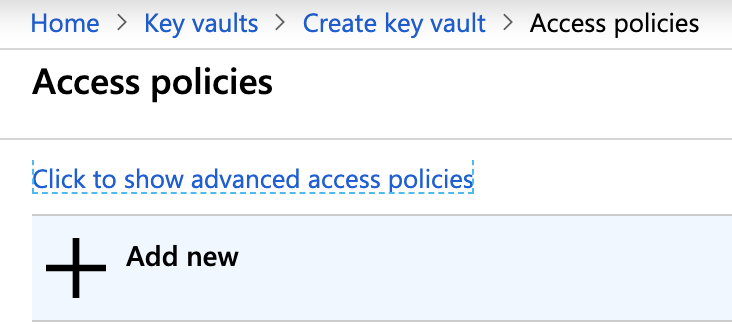
1. Click Add.
2. In the Create key vault section, enter the name for the Key Vault. In this example, create the key vault for the Root CA and use the name EJBCARootKeyVault. Note the name down since it will be required in EJBCA when creating a Crypto Token in the next step [Creating an Azure Key Vault Crypto Token in EJBCA](https://doc.primekey.com/ejbca-cloud/ejbca-cloud-azure/azure-key-vault-integration-guide/creating-an-azure-key-vault-crypto-token-in-ejbca). Then specify the same Resource Group that the EJBCA Instance uses.



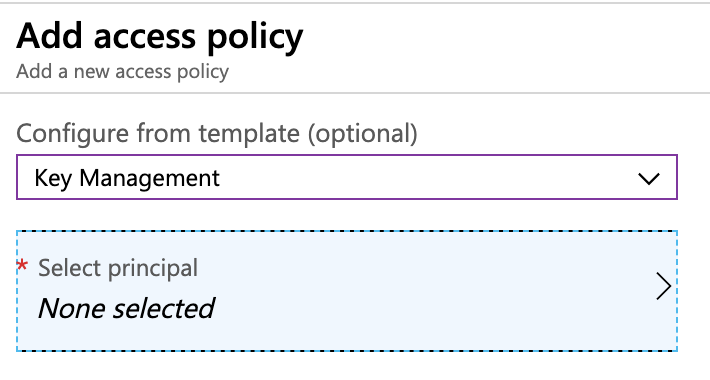
1. Under the Pricing tier, there are two options, Standard and Premium. It is recommended to choose Premium and then click Select.



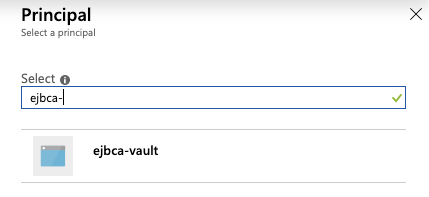
1. Click Access Policies from the Create Key Vault dialog, and then click Add new.



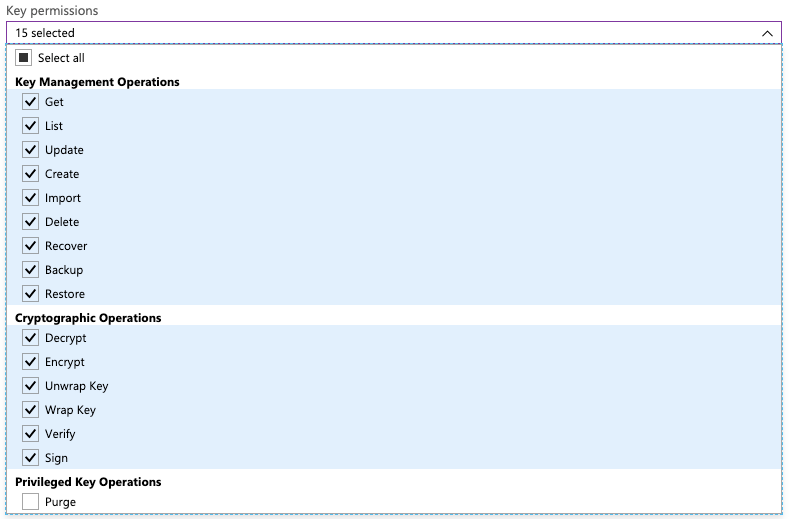
1. Select Key Management from the Configure from the template list.
2. Click Select Principal.



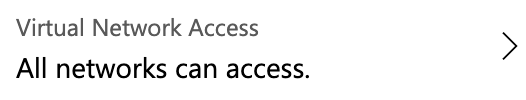
1. Search for the App Registration added in the previous section [Creating an App Registration in Active Directory](https://doc.primekey.com/ejbca-cloud/ejbca-cloud-azure/azure-key-vault-integration-guide/creating-an-app-registration-in-active-directory). In this example, we used the name "ejbca-vault". Select it and click Select.



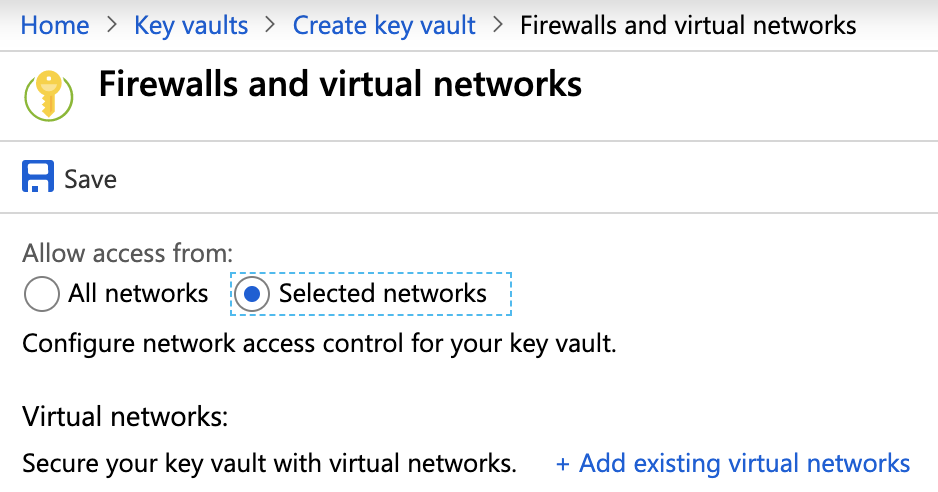
1. Select Key Permissions.
2. Select all of the permissions under Cryptographic Operations.



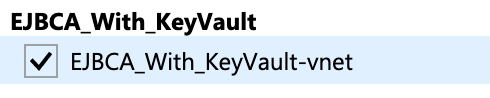
1. Click Add on the Add access policy screen.
2. Click OK on the Access Policies screen.
3. Click the Virtual Network Access section.



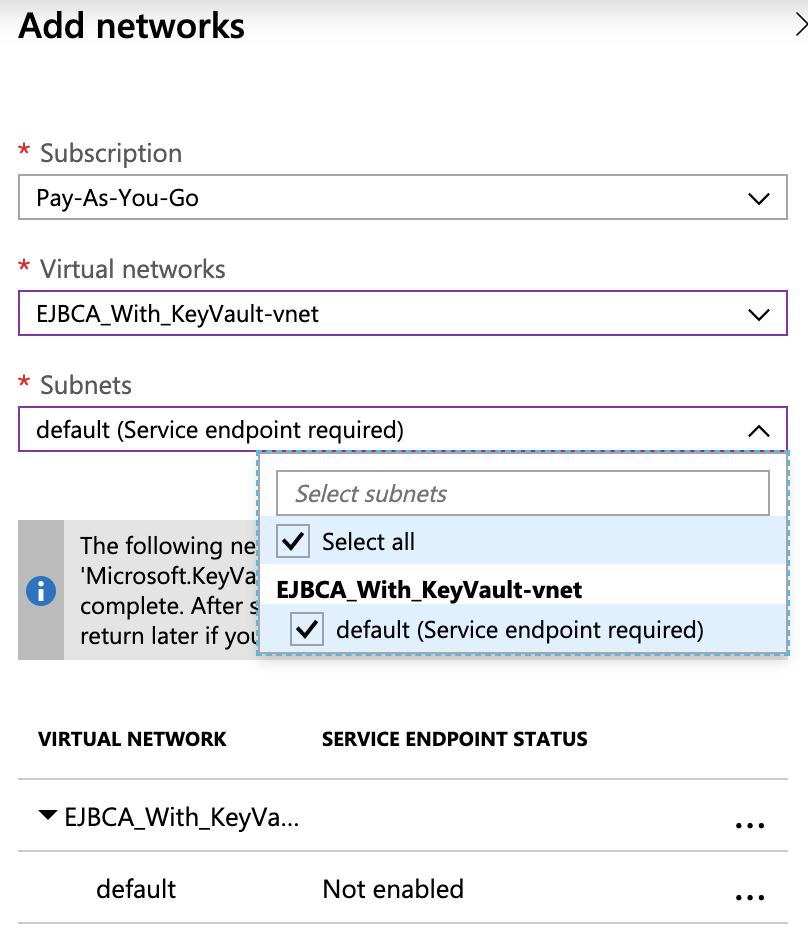
1. Select to Allow access from Selected networks, and then click Add existing virtual networks.



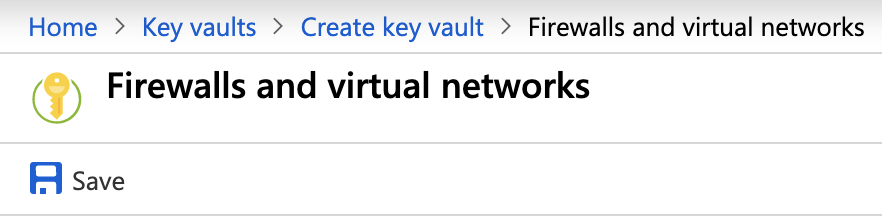
1. From the Virtual networks list, select the virtual network in the resource group that EJBCA Cloud was deployed into. In this example, EJBCA\_With\_KeyVault-vnet.



1. Click Select All under Subnets.



1. Once completed, click Add and then click Save.

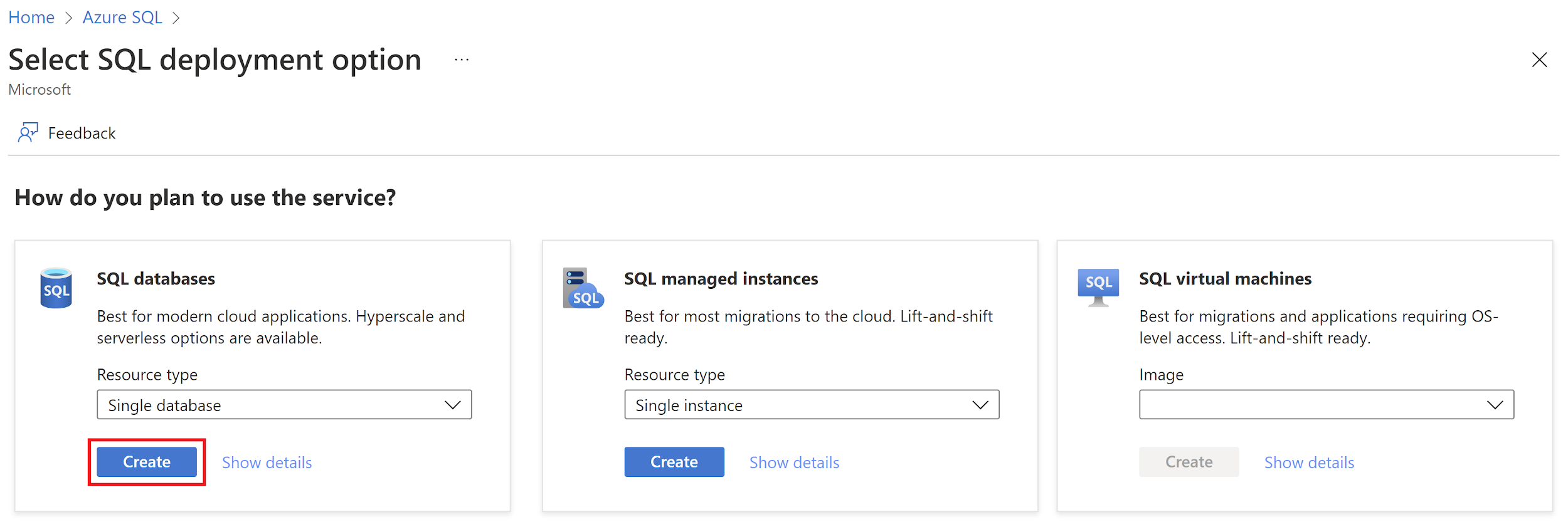


1. Click Create and wait for the deployment to complete.

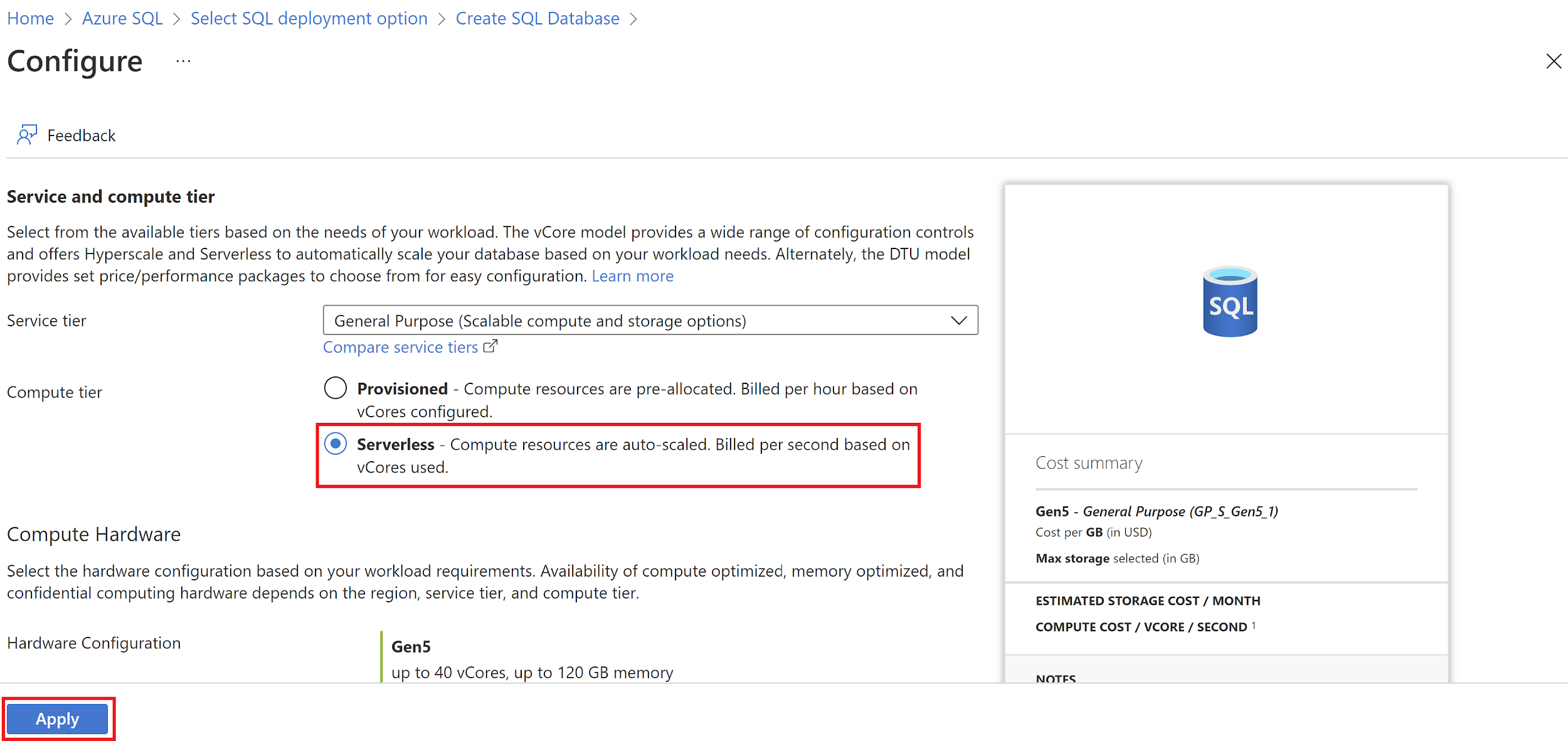
### Create a single database

To create a single database in the Azure portal, this quickstart starts at the Azure SQL page.

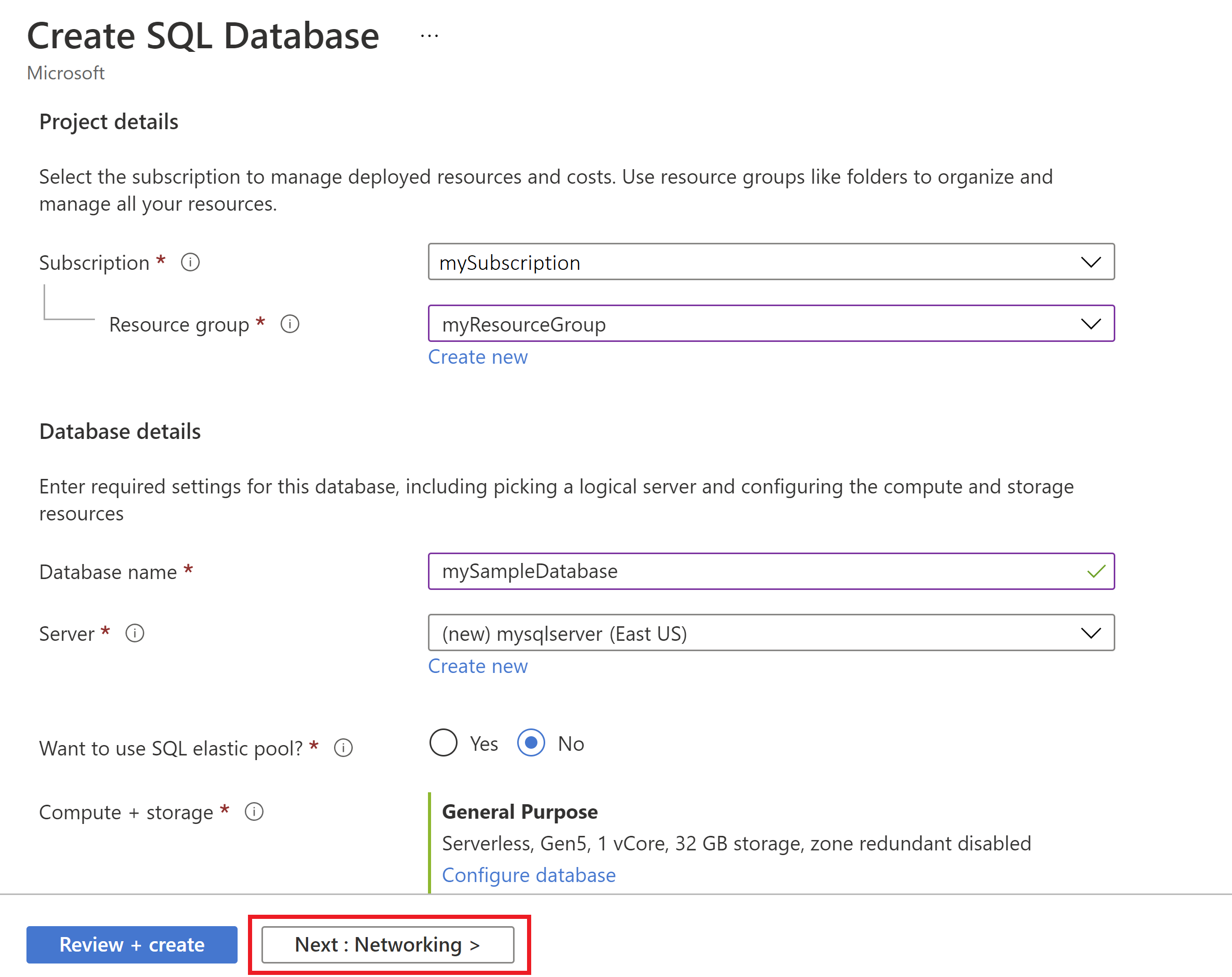
1. Browse to the [Select SQL Deployment option](https://portal.azure.com/#create/Microsoft.AzureSQL) page.
2. Under SQL databases, leave Resource type set to Single database, and select Create.



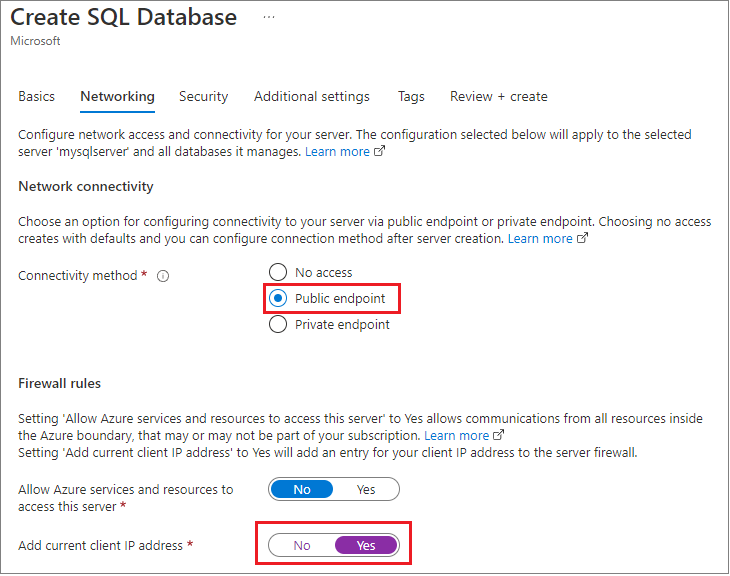
1. On the Basics tab of the Create SQL Database form, under Project details, select the desired Azure Subscription.
2. For Resource group, select Create new, enter myResourceGroup, and select OK.
3. For the Database name, enter my sample database.
4. For Server, select Create new, and fill out the New server form with the following values:
   * Server name: Enter mysqlserver, and add some characters for uniqueness. We can't provide an exact server name to use because server names must be globally unique for all servers in Azure, not just unique within a subscription. So enter something like mysqlserver12345, and the portal lets you know if it's available or not.
   * Location: Select a location from the dropdown list.
   * Authentication method: Select Use SQL authentication.
   * Server admin login: Enter azureuser.
   * Password: Enter a password that meets requirements, and enter it again in the Confirm password field.
5. Select OK.
6. Leave Want to use SQL elastic pool set to No.
7. Under Compute + storage, select Configure database.
8. This quickstart uses a serverless database, so leave the Service tier set to General Purpose (Scalable compute and storage options) and set Compute tier to Serverless. Select Apply.



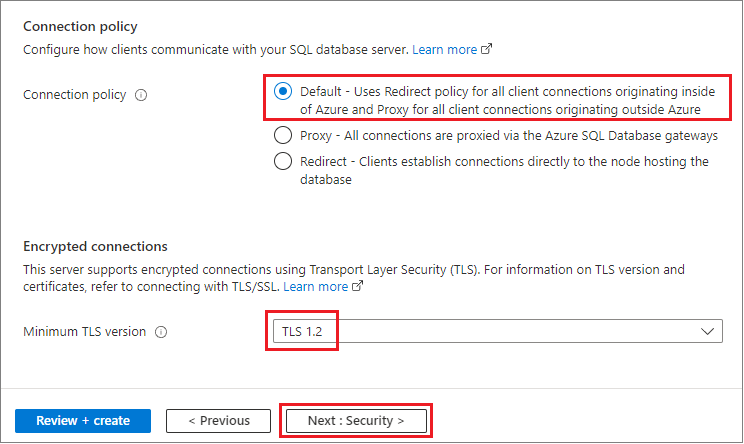
1. Under Backup storage redundancy, choose a redundancy option for the storage account where your backups will be saved. To learn more, see [backup storage redundancy](https://docs.microsoft.com/en-us/azure/azure-sql/database/automated-backups-overview?view=azuresql#backup-storage-redundancy).
2. Select Next: Networking at the bottom of the page.



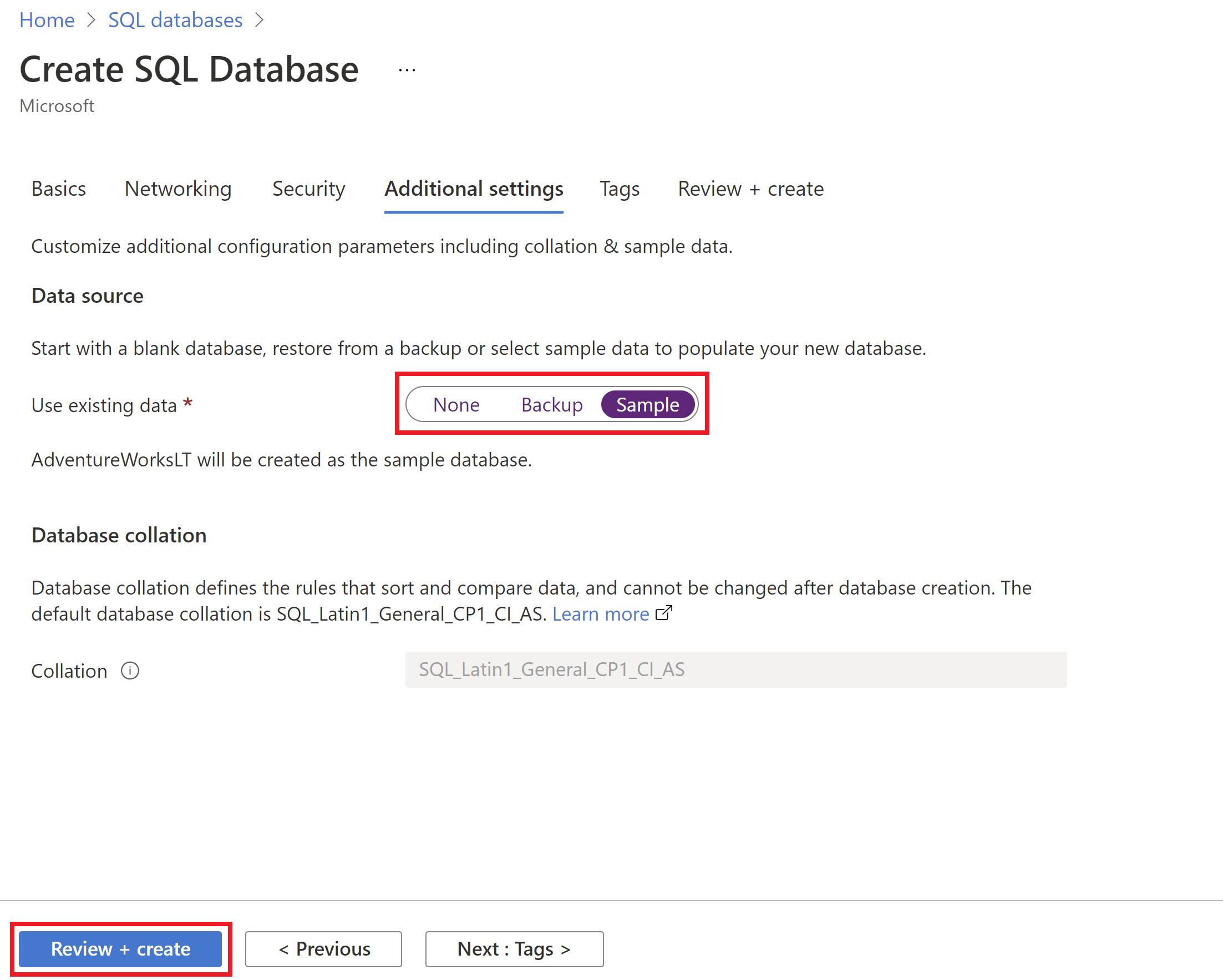
1. On the Networking tab, for the Connectivity method, select the Public endpoint.
2. For Firewall rules, set Add current client IP address to Yes. Leave Allow Azure services and resources to access this server set to No.



1. Under the Connection policy, choose the Default [connection policy](https://docs.microsoft.com/en-us/azure/azure-sql/database/connectivity-architecture?view=azuresql#connection-policy), and leave the Minimum TLS version at the default of TLS 1.2.
2. Select Next: Security at the bottom of the page.



1. On the Security page, you can choose to start a free trial of [Microsoft Defender for SQL](https://docs.microsoft.com/en-us/azure/azure-sql/database/azure-defender-for-sql?view=azuresql), as well as configure [Ledger](https://docs.microsoft.com/en-us/sql/relational-databases/security/ledger/ledger-overview), [Managed identities](https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/overview), and [Transparent data encryption (TDE)](https://docs.microsoft.com/en-us/azure/azure-sql/database/transparent-data-encryption-byok-overview?view=azuresql) if you desire. Select Next: Additional settings at the bottom of the page.
2. On the Additional settings tab, in the Data source section, for Use existing data, select Sample. This creates an AdventureWorksLT sample database so there are some tables and data to query and experiment with, as opposed to an empty blank database. You can also configure [database collation](https://docs.microsoft.com/en-us/sql/t-sql/statements/collations) and a [maintenance window](https://docs.microsoft.com/en-us/azure/azure-sql/database/maintenance-window?view=azuresql).
3. Select Review + create at the bottom of the page:

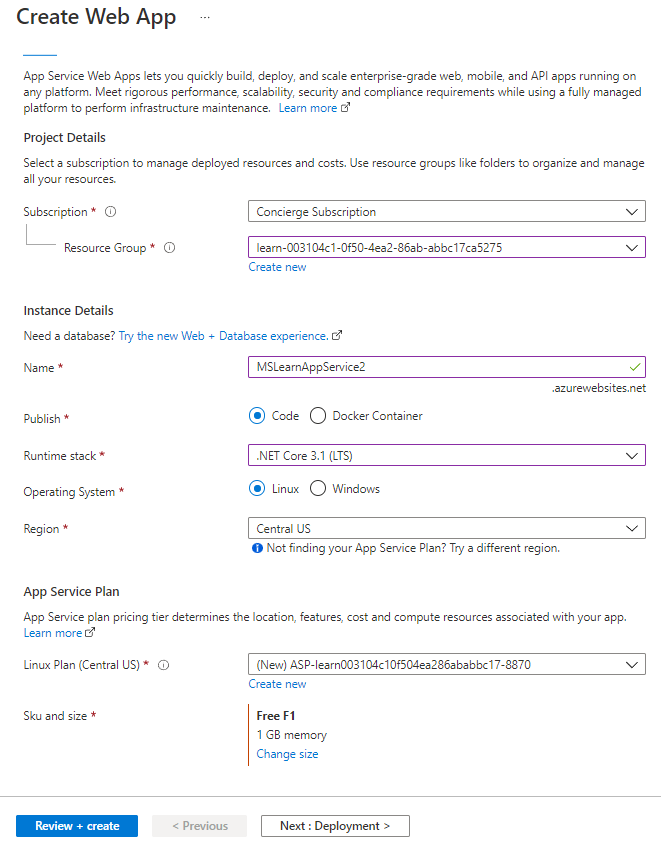


1. On the Review + create the page, after reviewing, select Create.

### Create a web app

Sign in to the [Azure portal](https://portal.azure.com/learn.docs.microsoft.com) using the same account you used to activate the sandbox.

1. On the Azure portal menu, or from the Home page, select Create a resource. Everything you create on Azure is a resource. The Create a resource pane appears.  
   Here, you can search for the resource you want to create or select one of the popular resources that people create in the Azure portal.
2. In the Create a resource menu, select Web.
3. Select Web App. If you don't see it, in the search box, search for and select Web App. The Create Web App pane appears.
4. On the Basics tab, enter the following values for each setting.

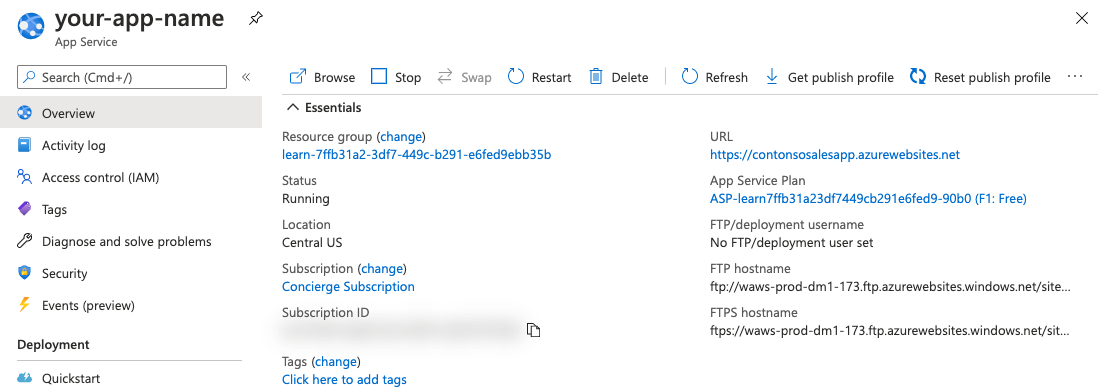


1. Select Review + Create to go to the preview pane, and then select Create. The portal shows the deployment pane, where you can view the status of your deployment.

Note: It can take a minute for deployment to complete.

### Preview your web app

1. When the deployment is complete, select Go to the resource. The portal shows the App Service Overview pane for your web app.

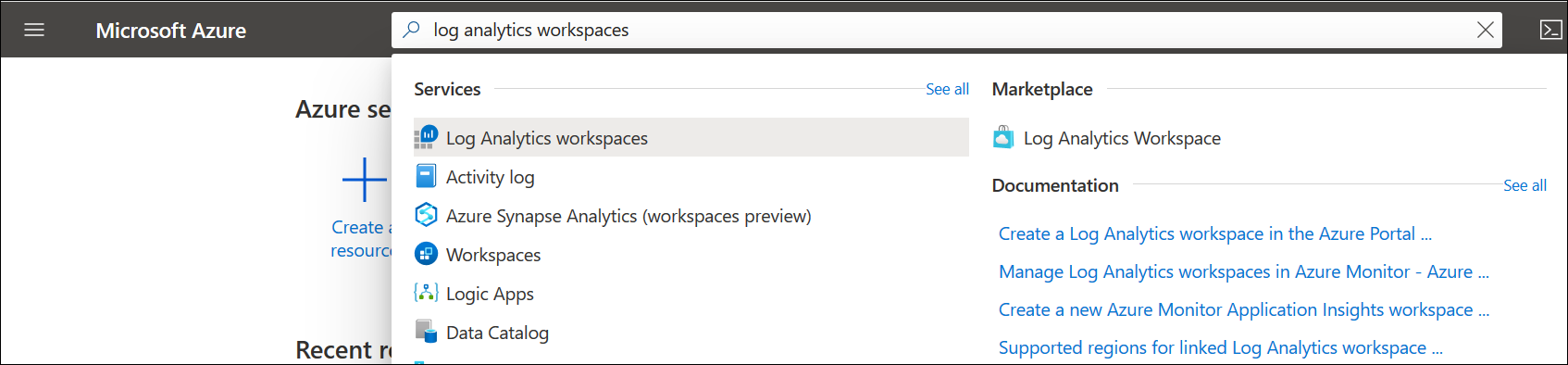


1. To preview your web app's default content, select the URL at the top right.

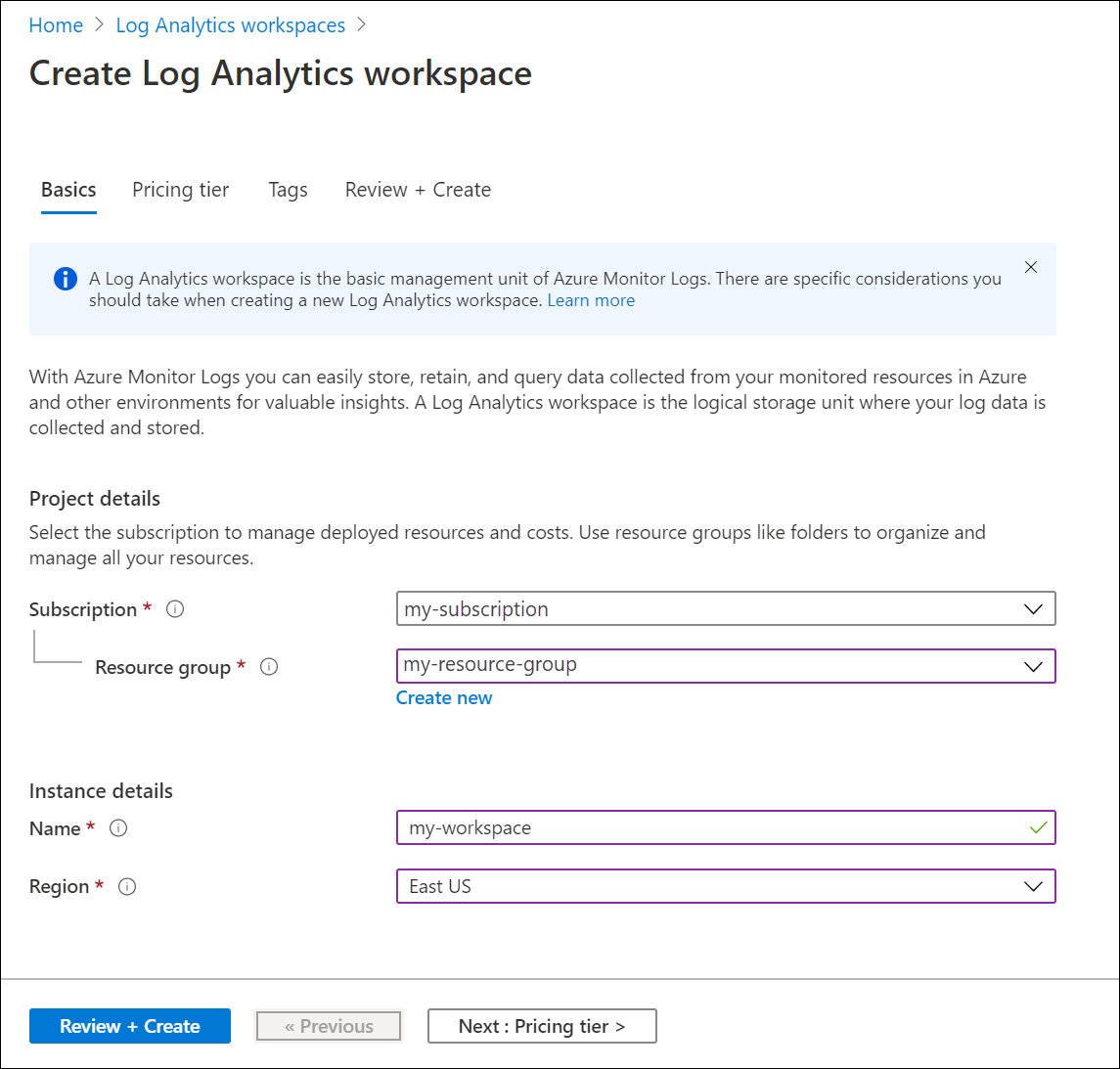
### Create a Log Analytics workspace

Use the Log Analytics workspaces menu to create a workspace.

1. In the [Azure portal](https://portal.azure.com/), enter Log Analytics in the search box. As you begin typing, the list filters based on your input. Select Log Analytics workspaces.



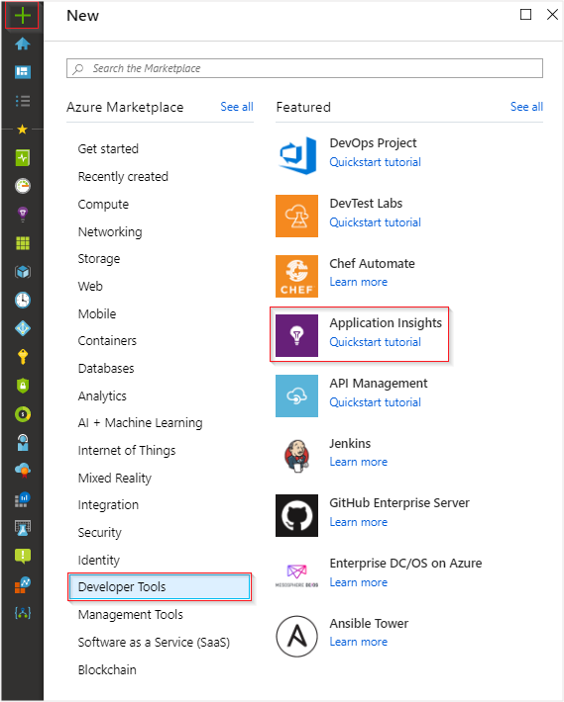
1. Select Add.
2. Select a Subscription from the dropdown.
3. Use an existing Resource Group or create a new one.
4. Provide a name for the new Log Analytics workspace, such as DefaultLAWorkspace. This name must be unique per resource group.
5. Select an available Region. For more information, see which [regions Log Analytics is available in](https://azure.microsoft.com/regions/services/). Search for Azure Monitor in the Search for a product box.



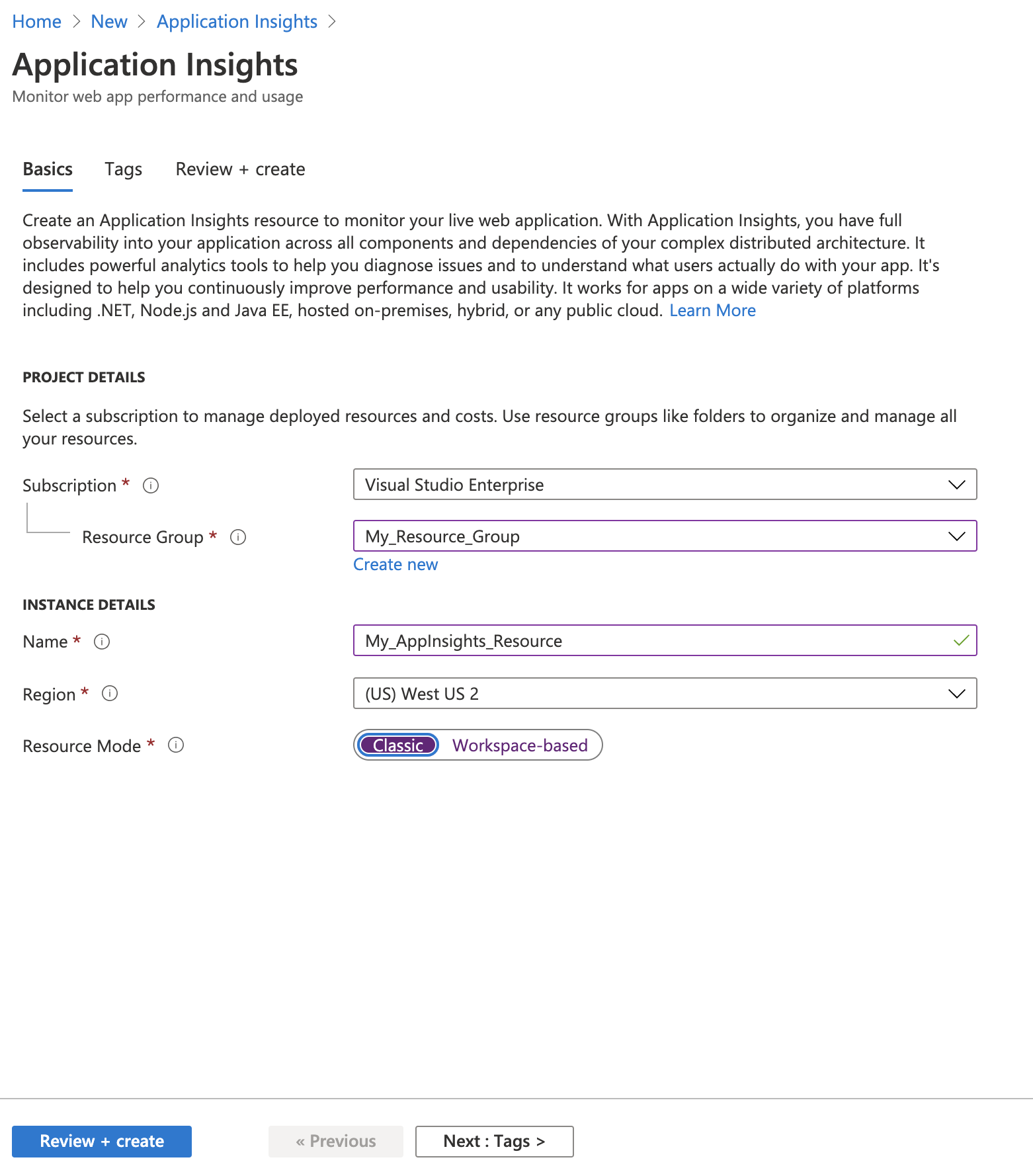
1. Select Review + Create to review the settings. Then select Create to create the workspace. A default pricing tier of pay-as-you-go is applied. No charges will be incurred until you start collecting enough data.

### Create an Application Insights resource

1. Sign in to the [Azure portal](https://portal.azure.com/), and create an Application Insights resource:



1. Note: While you can use the same resource name across different resource groups, it can be beneficial to use a globally unique name. This can be useful if you plan to [perform cross-resource](https://docs.microsoft.com/en-us/azure/azure-monitor/logs/cross-workspace-query#identifying-an-application) [queries](https://docs.microsoft.com/en-us/azure/azure-monitor/logs/cross-workspace-query#identifying-an-application) as it simplifies the required syntax.
2. Enter the appropriate values into the required fields, and then select Review + create.



1. When your app has been created, a new pane opens. This pane is where you see performance and usage data about your monitored application.

### RabbitMQ

RabbitMQ is an established, general-purpose message broker. A broker facilitates the exchange of messages between a set of distributed microservices and allows them to communicate efficiently and reliably.

Kubernetes, with its API-centric approach, is an ideal operator to deploy RabbitMQ instances in a cluster. Kubernetes’ scaling capabilities and automated container deployment enhance RabbitMQ’s dependability in an extended cluster environment.

Please follow the below document to set up the Rabbitmq in AKS Cluster.

<https://git.app.uib.no/caleno/helm-charts/tree/79aa425075232d7024e43ad8f88f3776c68093f5/stable/rabbitmq>

### SEQ

Seq is a real-time search and analysis server for structured application log data. Its carefully-designed user interface, JSON event store, and familiar query language make it an efficient platform for detecting and diagnosing issues in complex applications and microservices.

Please follow the below document to set up the SEQ in AKS Cluster.

<https://docs.datalust.co/docs/using-aks>

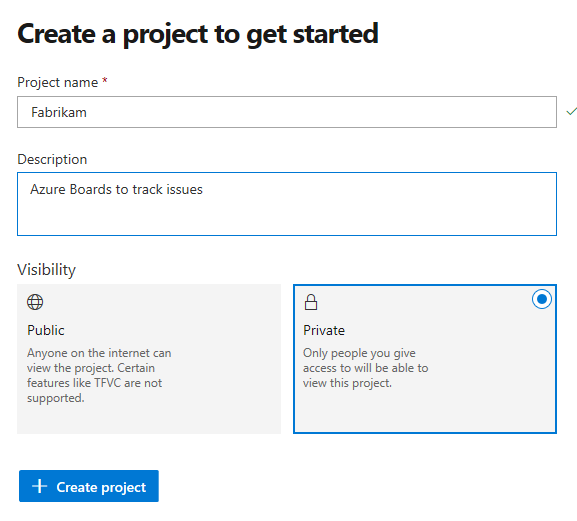
### Azure DevOps

A collection of services given by Microsoft Azure. It provides development services for a team to support, plan, collaborate, build, and deploy applications. It provides integrated features in a browser or an IDE(Integrated Development Environment). Some of the services for developers are:

* Azure Repos
* Azure Pipelines
* Azure Boards
* Azure Test Plans
* Azure Artifacts

#### Create a project

You can create public or private projects.

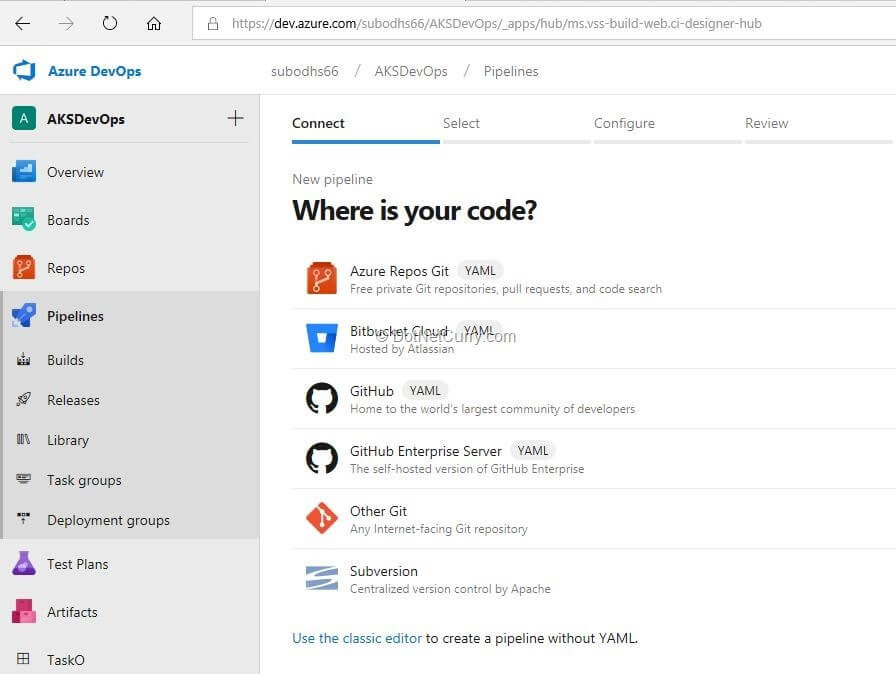
1. Enter a name for your project, select the visibility, and optionally provide a description. Then choose Create project.  
     
   Special characters aren't allowed in the project name (such as / : \ ~ & % ; @ ' " ? < > | # $ \* } { , + = [ ]). The project name also can't begin with an underscore, can't begin or end with a period, and must be 64 characters or less. Set your project visibility to either public or private. Public visibility allows anyone on the internet to view your project. Private visibility is for only people who you give access to your project.
2. When your project is created, if you signed up with a Microsoft account, the wizard to create a new pipeline automatically starts.

#### What is Azure Pipelines?

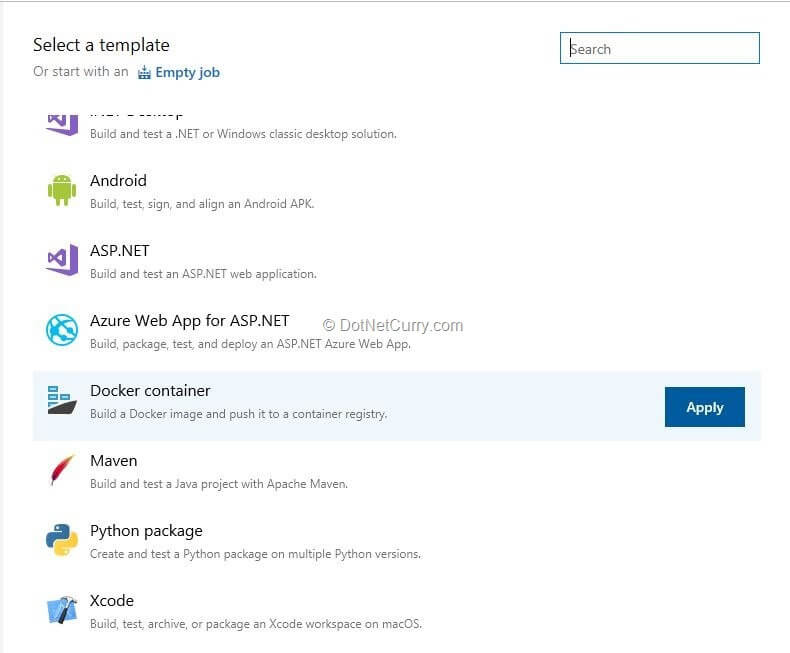
The Azure CI/CD pipeline simplifies continuous integration and continuous delivery (CI/CD) in the application development process. You can start from the source stage with existing code on GitHub or on-premise containers. The Azure Repos can maintain a central repository, and the Azure pipelines maintain build and release pipelines for the given project. The Azure DevOps CI/CD process is a crucial process with all the required dev services.

#### Create a build pipeline

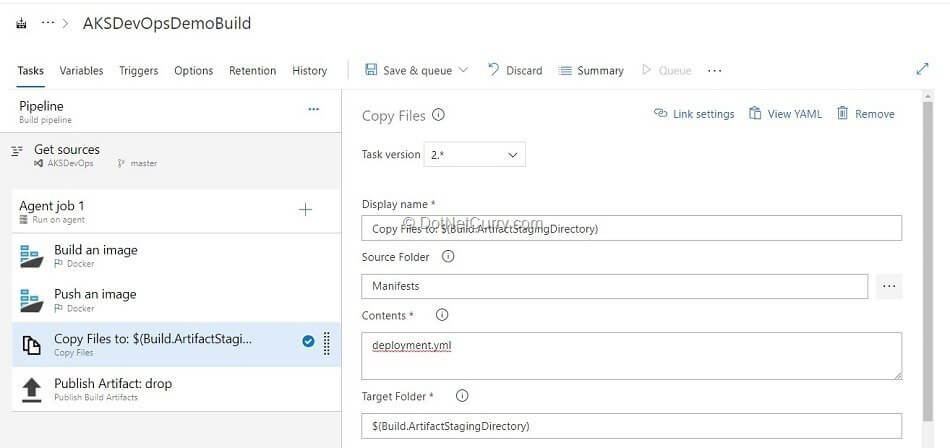
1. we will create a new Build pipeline in which we will create an image based upon the Dockerfile and then Push it to the ACR that we have created earlier.
2. To create the new Build pipeline, open the page of your organization in Azure DevOps https://dev.azure.com//AKSDevOps and then select Builds from the Pipelines section in the left pane.
3. Select the link “Use the classic editor” to create the pipeline without YAML.



1. On the subsequent page, select the “Docker container” template.



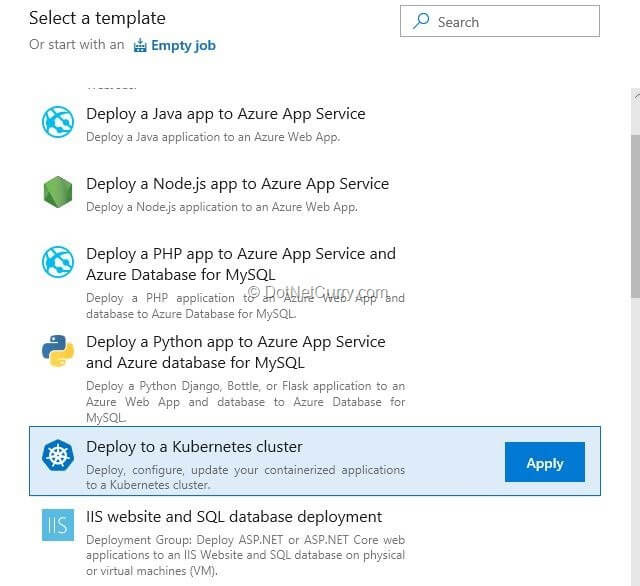
1. This template will provide tasks to create the container image and to push it to the ACR.
2. Before making changes in the parameters of the tasks, open the Pipeline node. On this node, change the name of the pipeline to AKSDevOpsDemoBuild and select the Hosted agent pool. Since we are creating Docker images, agents under this pool support the actions to create and push those images.



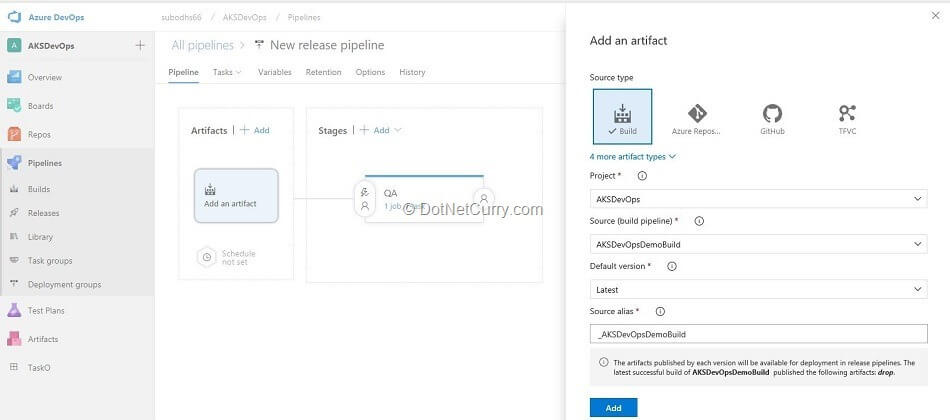
1. After configuring these tasks, we will “Save and Queue” this build. At the success of the build, the image is created as configured in the Dockerfile, pushed to ACR, and the artifact as mentioned earlier, is created and published.

#### Create a release pipeline

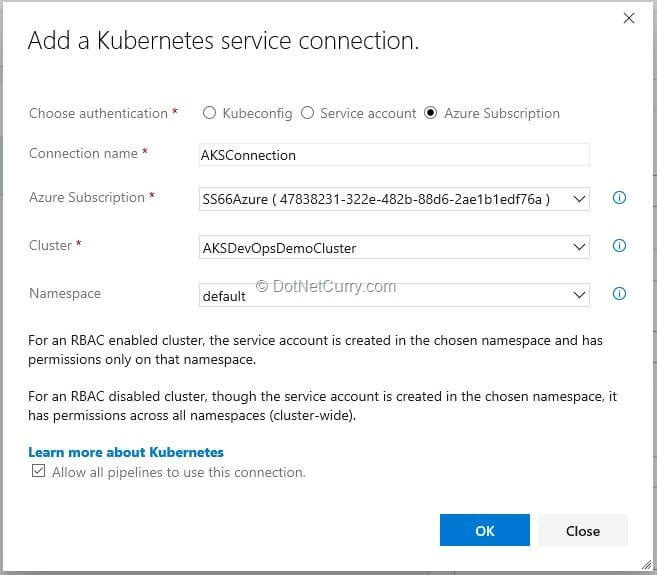
1. We now have to deploy the created image on AKS Cluster. We will do that using the Release Management service.
2. Let’s create a new release pipeline from the Pipelines – Releases section in the left-hand pane of the Azure DevOps page. In this release pipeline, we will add only one stage (for the sake of this example), but normally there may be multiple stages in a release pipeline.
3. We will select the “Deploy to Kubernetes cluster” template for this release pipeline.



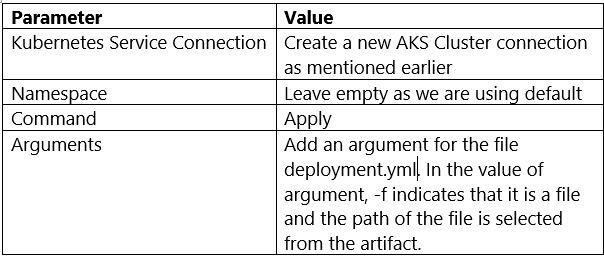
1. Deploy to Kubernetes cluster template for release pipeline
2. This template by default adds one stage, let’s call it “QA”. We will add the build pipeline definition that we created earlier, as the artifact source.

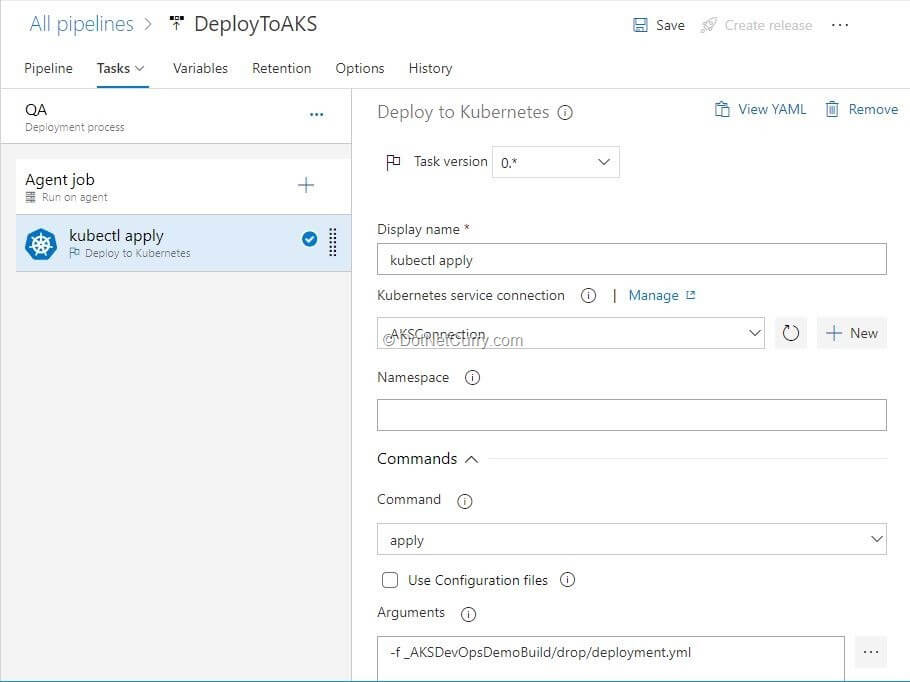


1. Select artifact source and note artifact alias
2. Let’s now set the parameters for the task that is added by the template.
3. That task is of “kubectl”. Before we set other parameters, let’s set up a connection to the AKS Cluster. This is done through the wizard that is started by clicking the New button for the parameter of Kubernetes Service Connection.
4. We will base our connection on the Azure Subscription which is common between Azure and Azure DevOps. Once we select the Azure subscription in this wizard, we can select the AKS Cluster we had created earlier. We will use the “default” namespace. Click OK to create a new connection.



1. Create a connection to AKS Cluster
2. We will set the parameters for this task as shown below:





1. Kubectl task in release pipeline
2. Now create a release that will pull the image that we had built and deploy the containers in the pods on a node in AKS Cluster. Let’s view those pods.

#### View pods and services

1. One excellent tool within the Azure Portal is called CloudShell.
2. It is a shell that can be accessed without going out of the portal (in the browser). Once we open it, we can execute either PowerShell or Bash commands on the command prompt. In this example, let’s select to open the CloudShell in Bash mode. We will now connect to our AKS Cluster by using the command:

**$ az aks get-credentials --resource-group research --name AKSDevOpsDemoCluster**

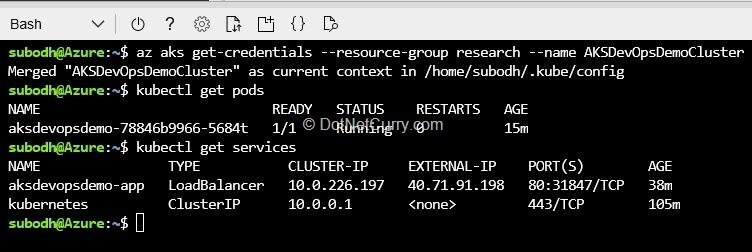
1. In the above-mentioned command, “research” and “AKSDevOpsDemoCluster” are names and may be different in your case.
2. The next command is to get a list of pods:

|  |
| --- |
| **$ kubectl get pods** |

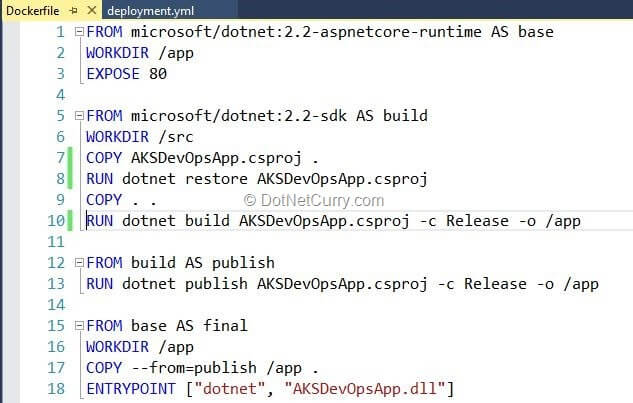
1. This command will list the pods that are created by Azure DevOps Release.
2. Another command is to view the services created:

**$ kubectl get services**

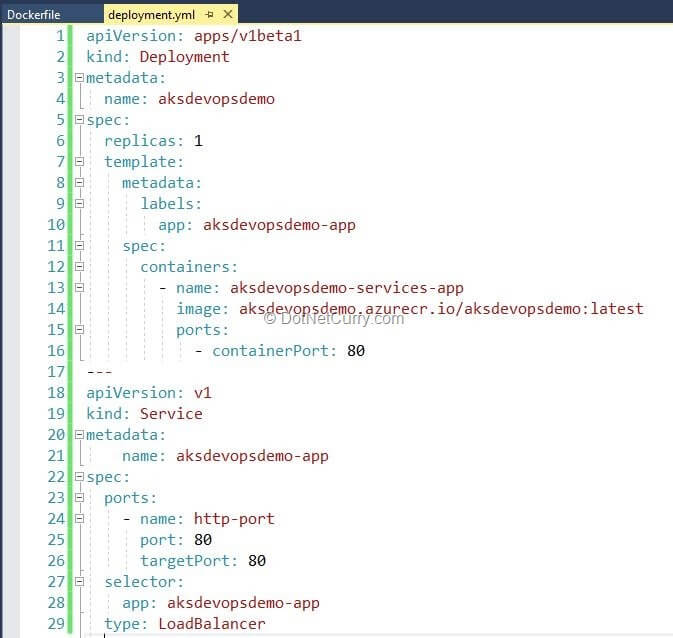
1. This command will list the services including the LoadBalancer service and the service that manages the cluster.



1. Dockerfile



1. Deployment file



1. Reference Document:

<https://www.dotnetcurry.com/devops/1518/aspnet-core-cicd-azure-kubernetes-service>