**Azure Kubernetes Service (AKS)**

**Azure Kubernetes Service (AKS)** is a managed Kubernetes service in which the master node is managed by Azure and end-users manages worker nodes. Users can use AKS to deploy, scale, and manage Docker containers and container-based applications across a cluster of container hosts. One of the best parts about **AKS**is that you only pay for the worker nodes within your clusters, not for the masters. You can create a cluster in the Azure portal, with the Azure CLI, or template-driven deployment options such as Resource Manager templates and Terraform.

The AKS cluster can be accessed from a local machine’s terminal to manage Kubernetes components like **deployments** and [**Kubernetes** **pods**](https://k21academy.com/docker-kubernetes/kubernetes-pods-for-beginners/). It can even be used to create a Kubernetes deployment.

***Note:***Know more about [**Azure Kubernetes Service (AKS)**](https://k21academy.com/microsoft-azure/az-104/video-containers-docker-kubernetes-in-azure-for-beginners/) and [**Kubernetes Architecture**](https://k21academy.com/docker-kubernetes/kubernetes-architecture-components-overview-for-beginners/).

**How To Create Azure Kubernetes Cluster**

**Azure Kubernetes Service (AKS)** is a Kubernetes service that lets you quickly deploy and manage master and worker nodes in clusters. Creating an AKS cluster is easy and there are more than enough manuals who will guide you through the process. It is an effortless process to create a cluster by following the steps given below.

There are 4 ways to deploy an Azure Kubernetes Cluster, which are using:  
**I)** Azure Portal  
**II)** Azure CLI  
**III)** [Azure PowerShell](https://k21academy.com/microsoft-azure/az-900/az-900-azure-cloud-services-management-tools/)  
**IV)** Using template-driven deployment options, like [**Azure Resource Manager templates**](https://k21academy.com/microsoft-azure/az-303/arm-templates/) and [**Terraform**](https://k21academy.com/terraform-iac/terraform-beginners-guide/)

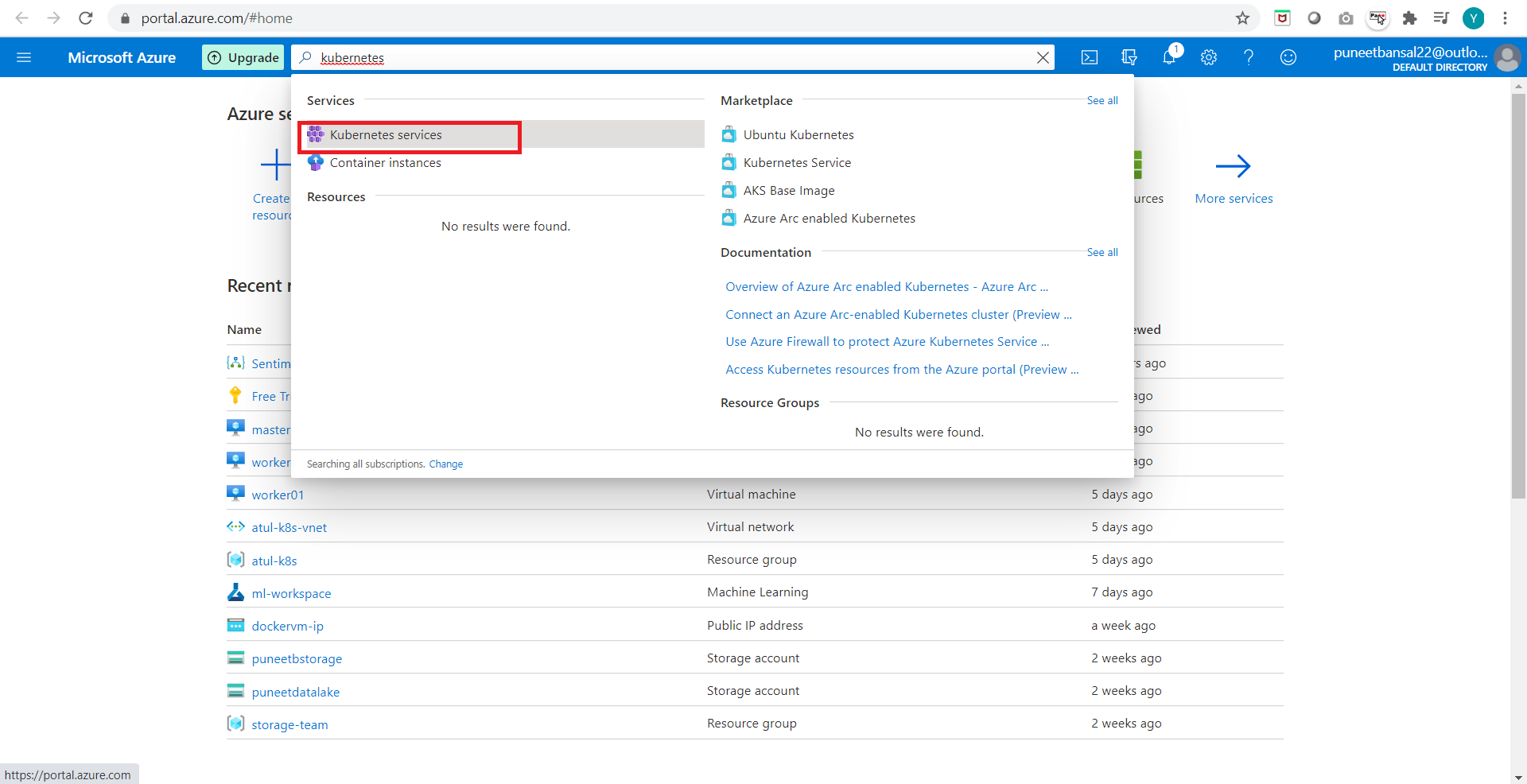
Now let us look at how do we create a cluster using **Azure Portal**. But before all that you should an Azure Cloud account to create a cluster. Create a [**FREE TRIAL AZURE ACCOUNT**](https://k21academy.com/microsoft-azure/create-free-microsoft-azure-trial-account/) to get **$200 free credit** on your Azure portal.

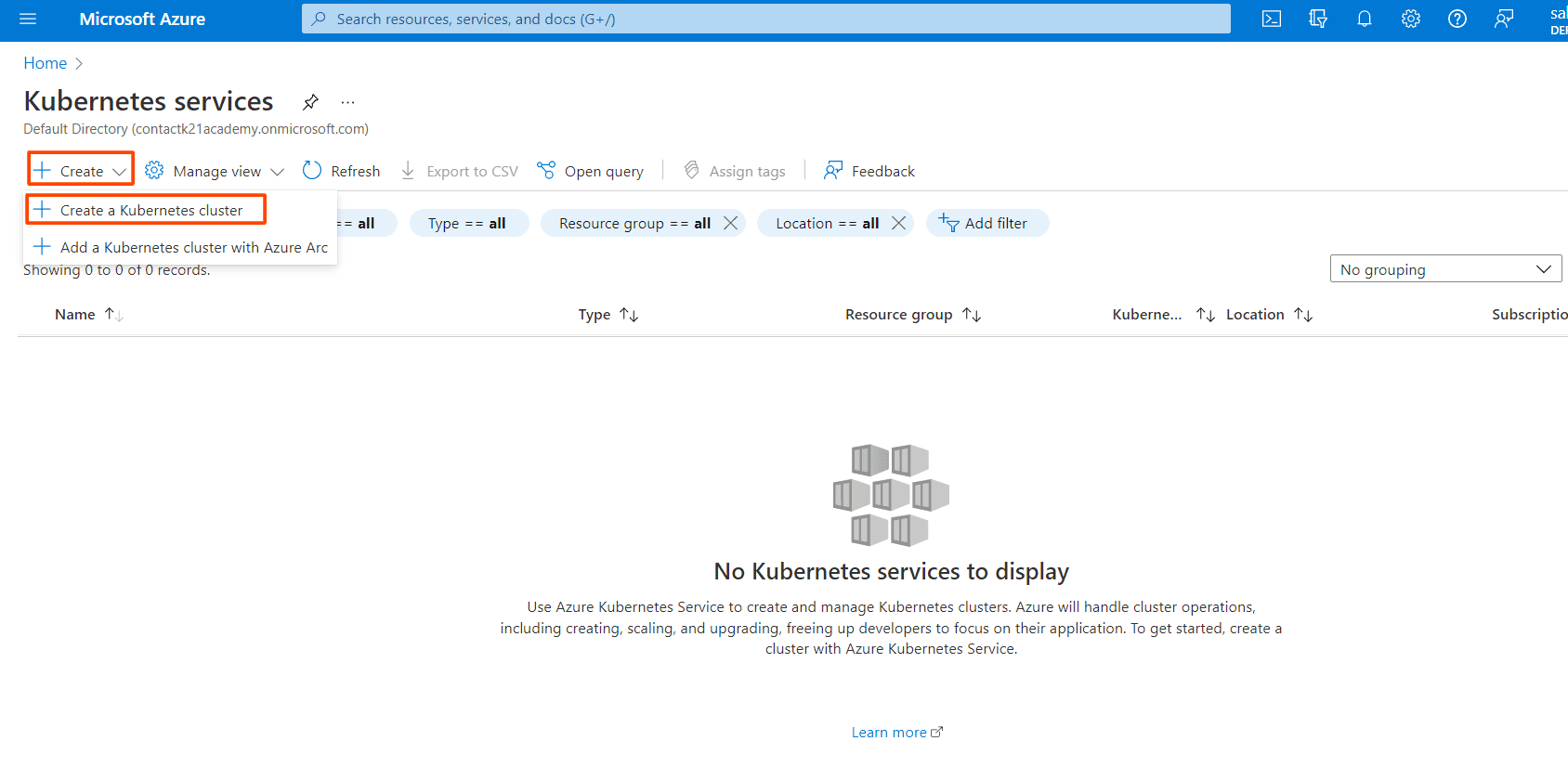
**Steps To Create AKS Cluster**

Once you have created an Azure account, or if you already have one, please follow the steps given below in order to create an Azure Kubernetes Cluster easily.

Cluster Creation Steps

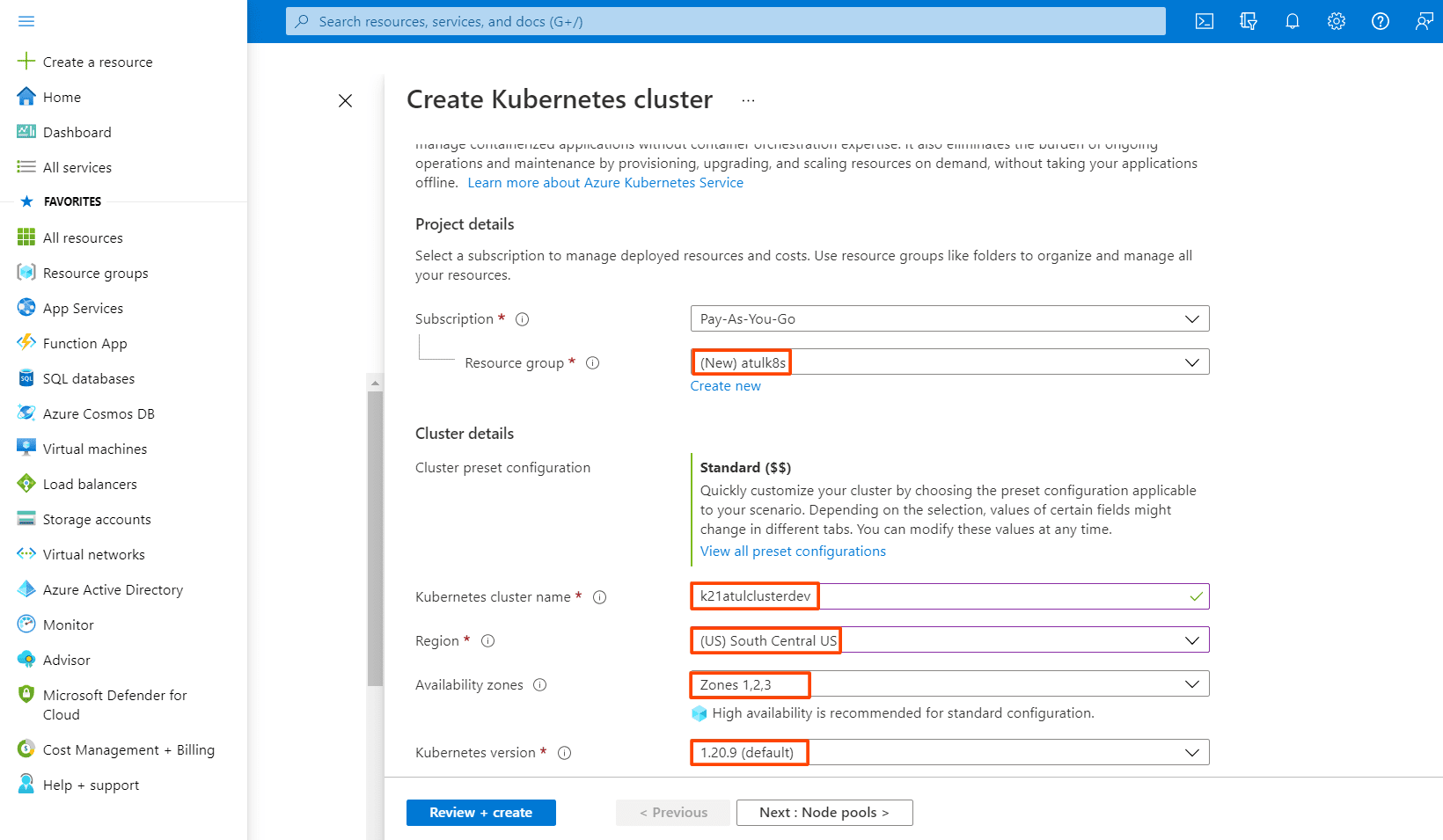
**Step 1 [Kubernetes Services]:** Go to the Azure Portal and search for **Kubernetes Service**in the search bar and click on it.



**Step 2 [Add Kubernetes Cluster]:**Once you reach the Kubernetes Services page, click on **Create**and then **Create Kubernetes Cluster**.

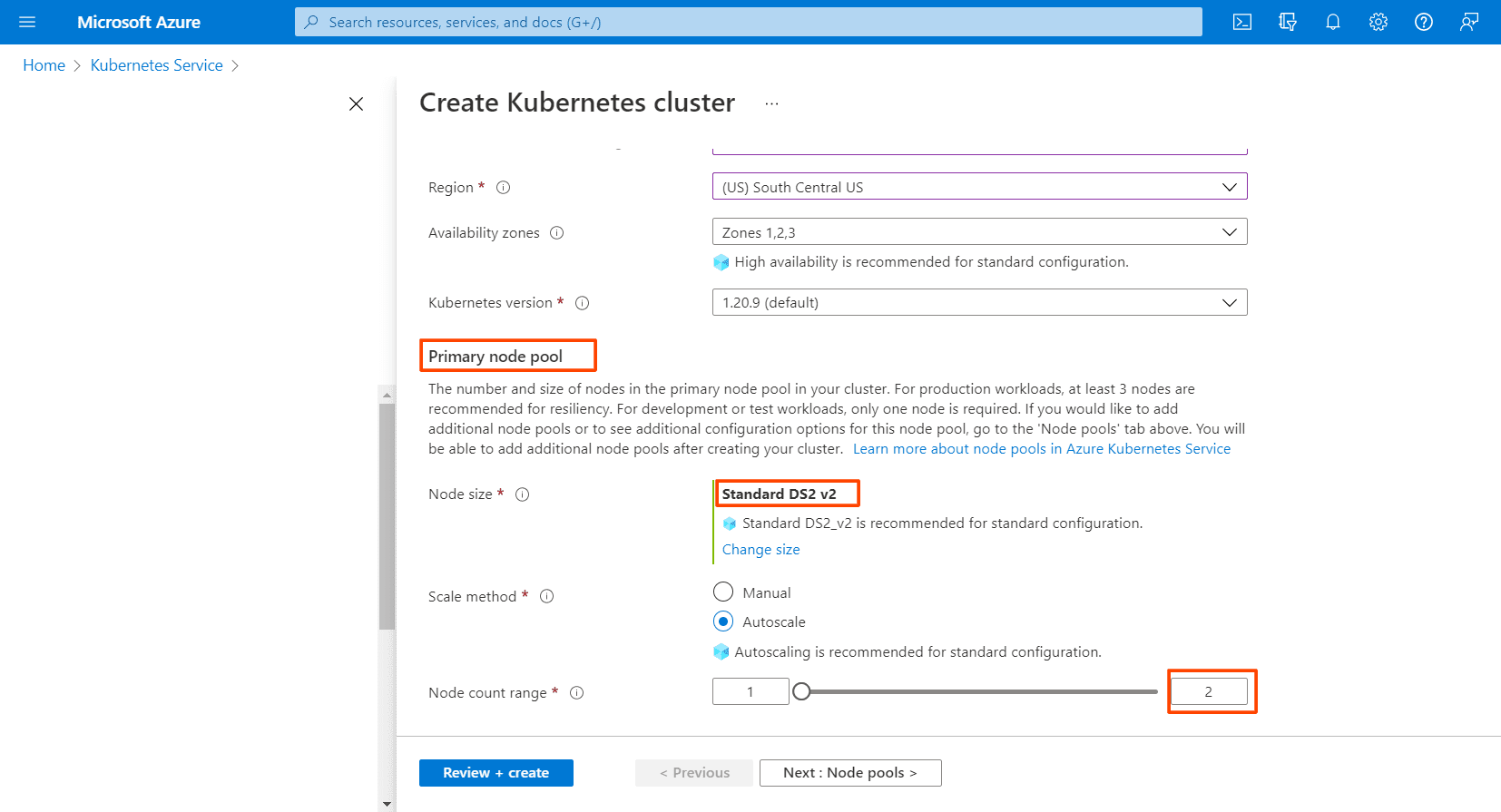
**Step 3 [Basics]:**Once you click on add Kubernetes cluster, the next step is to update the specifications of the cluster. So, click on **Basics**.

* Give the [**Resource Group**](https://k21academy.com/microsoft-azure/az-900/az-900-microsoft-azure-resource-group-resource-manager-arm-template/)name as per your requirement.
* Specify a name to your cluster in**the Kubernetes cluster name**field.
* Choose a **Region** in which you want to create your AKS cluster. In the specified region, our master node will be created.
* Based on the region the select the [**availability zones**](https://k21academy.com/microsoft-azure/az-303/azure-availability-zones-and-regions/).
* Select the **Kubernetes Version.**Here I am choosing the default, i.e., **1.20.9**



Next, comes the size and count of the nodes of the AKS cluster that we are gonna create. These can be updated as per the requirements.

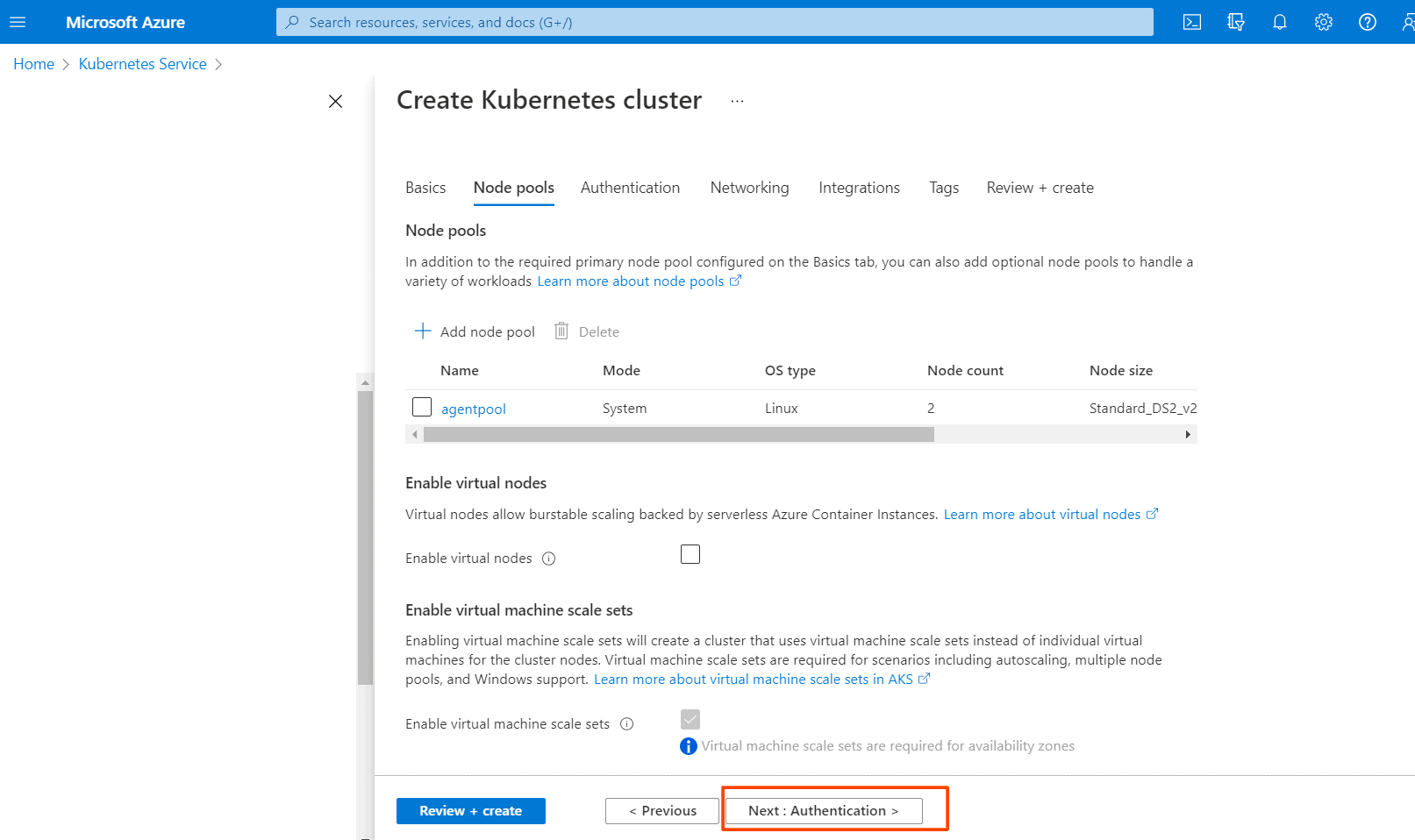
* Select the **Node Size.**We are choosing **Standard Ds2 v2**which has the following configuration: 2 vCPUs, 7 GiB RAM, 8 Data Disks, 14 GiB Temp Storage.
* Give the **Node Count** value which specifies how many Worker Nodes we want.

****

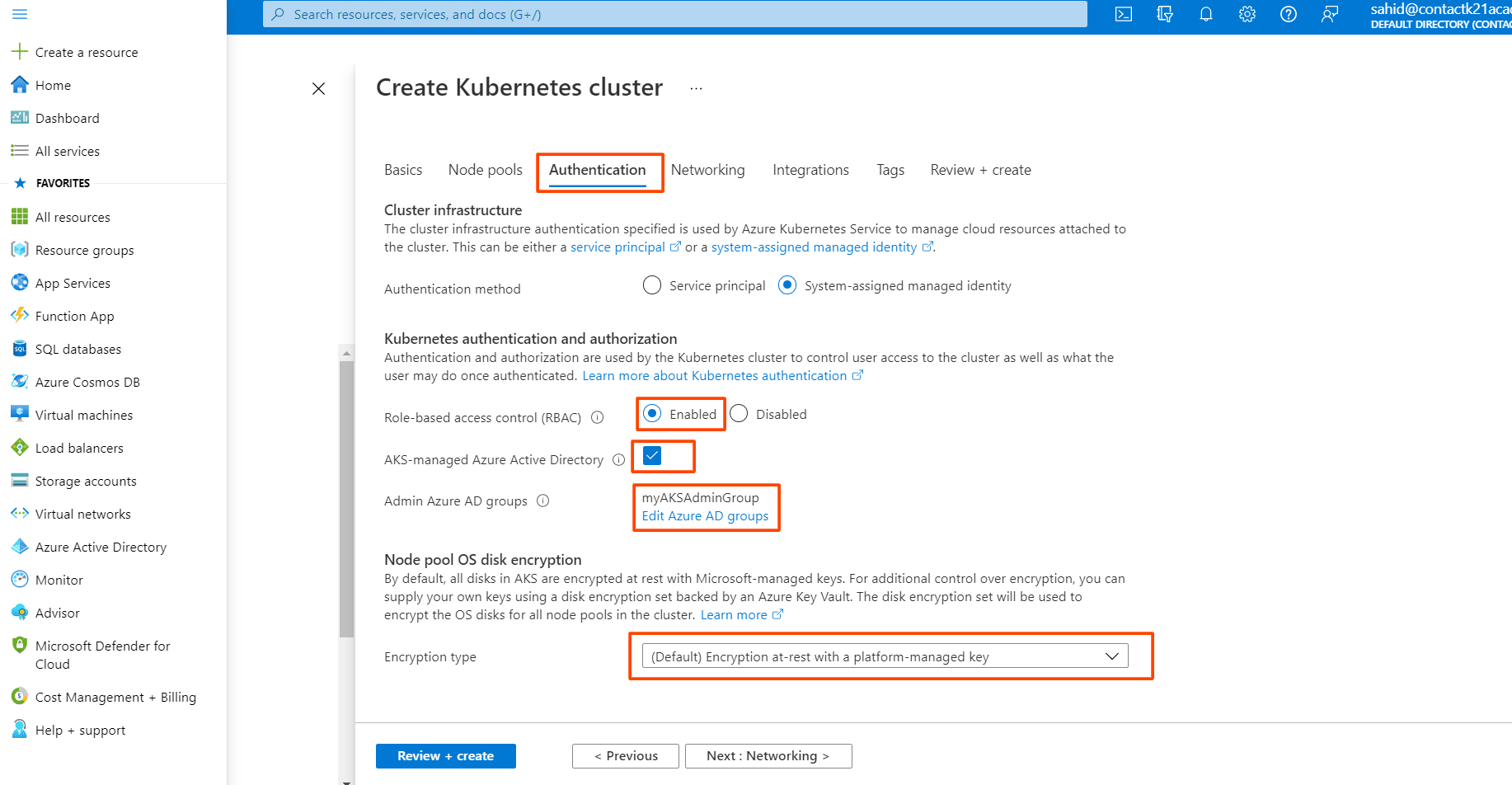
**Also Read:** [What is Kubernetes Cluster in Azure](https://k21academy.com/docker-kubernetes/three-node-kubernetes-cluster/)? Click here

**Step 4 [Node Pools]:**Next comes the **Node Pools**, follow the steps given below:

* In Azure Kubernetes Service (AKS), nodes of the same configuration are grouped together into **node pools**. **Node pools** contain the underlying VMs that run your applications.
* The **Virtual nodes**are a type of Serverless container instance. As we want to create the Worker nodes as Virtual Machines, so we **won’t enable** this option.

****

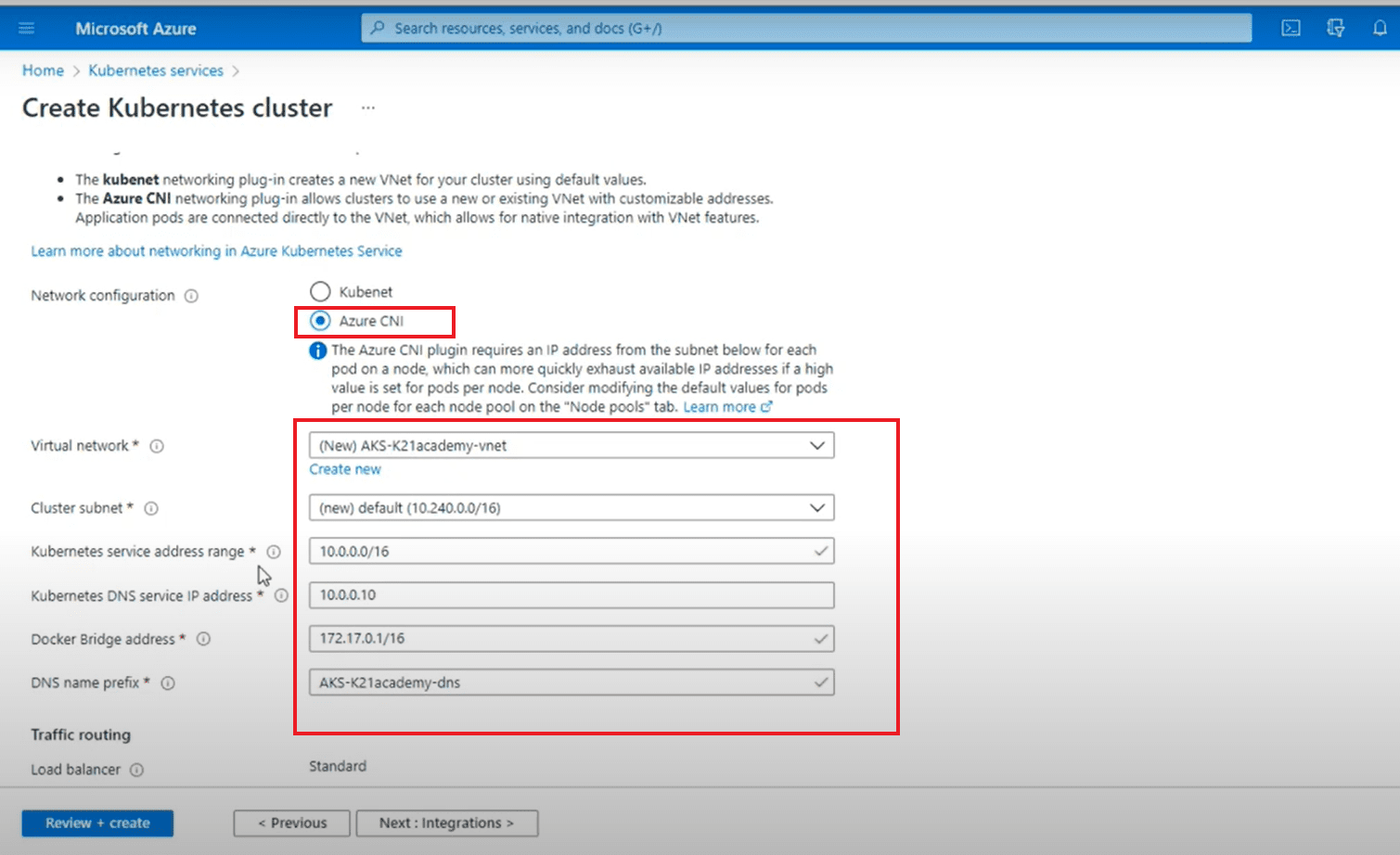
**Step 5 [Authentication]:** Next is to click on **Authentication**.

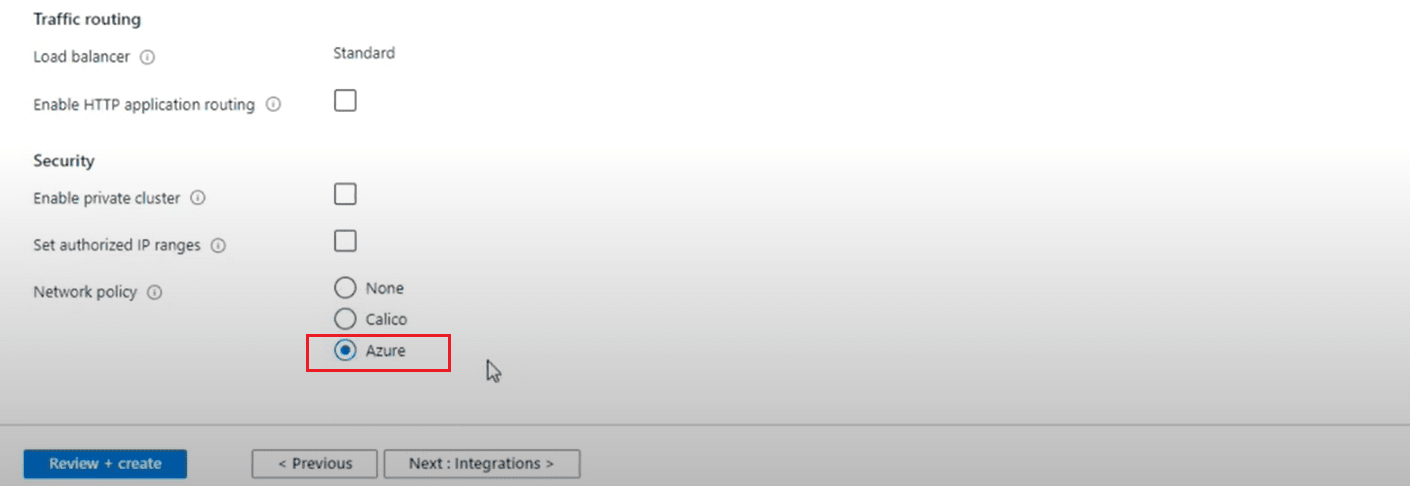
* Choose the**System-assigned managed identity.**
* If you want to go for [**Role-based Access Control**(**RBAC)**](https://k21academy.com/docker-kubernetes/rbac-role-based-access-control/) then select **Enabled.**
* We can use both **RBAC** & [**Azure Active Directory**](https://k21academy.com/microsoft-azure/az-303/azure-active-directory-azure-ad/) for Authentication.
* **Note:** Here in this step-by-step guide, we are using both RBAC & Azure AD for authentication.
* Check the Azure Active Directory option.
* Create a group in Active Directory and add it.
* Choose the **Encryption Type** of your choice, I will use the Default one.

**Also Read:**Our blog post on [**Network Policy Kubernetes**](https://k21academy.com/docker-kubernetes/network-policies-in-kubernetes/)**.**Click here

**Step 6 [Networking]:** Next is the[**Networking**](https://k21academy.com/microsoft-azure/az-304/virtual-networks-in-microsoft-azure-vnet-peeringexpressroutevpn-gateway/)part.

* Select the **Network Configuration.**I will be choosing**Azure CNI**
* The **Cluster Subnet**option is to choose which Subnet you want the Nodes and Containers to be placed in.
* **Kubernetes service address range**is the CIDR notation IP range from which to assign server cluster IPs.
* **Docker Bridge address**is the IP address assigned to Docker Bridge. The Bridge Network is for the container to container communication.
* In **Private Cluster**, the communication between the nodes and the API server happens internally.
* So, I am **Disabling**the Private Cluster.
* Keep the Network Policy to **Azure**.
* Do not enable**HTTP**application routing.

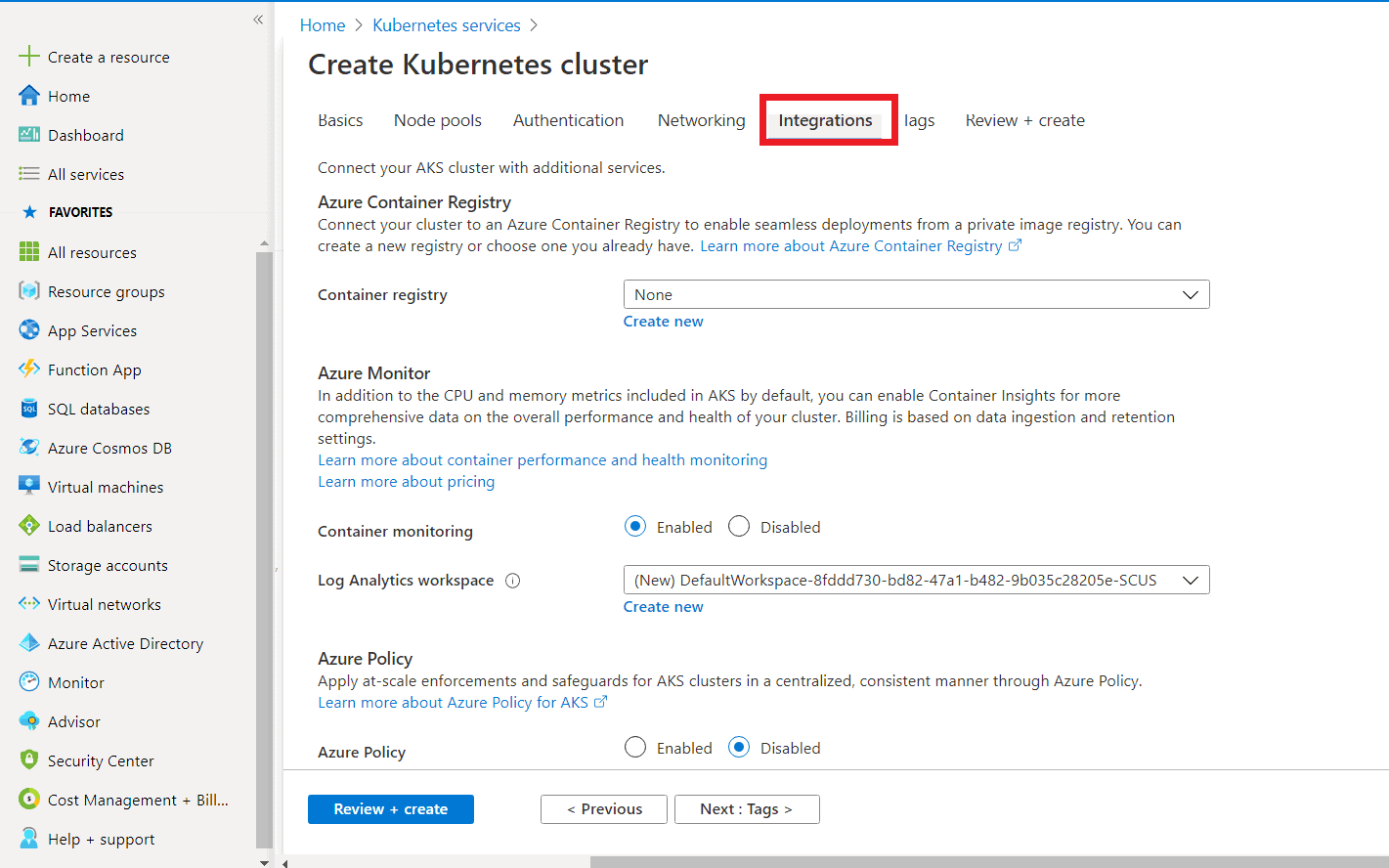
****

****

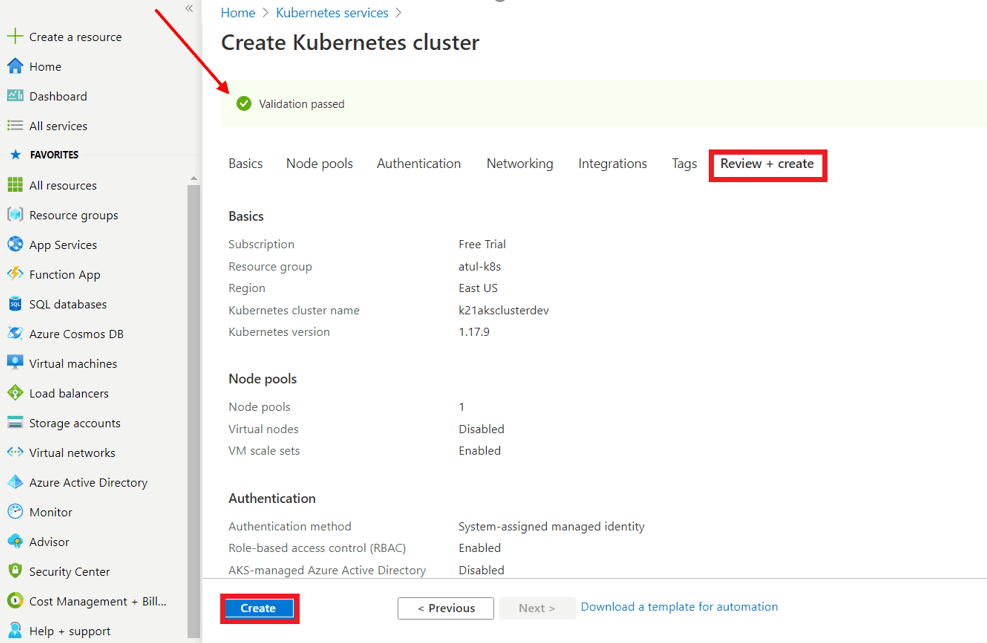
**Also Read:**Our previous blog post on [**Kubernetes networking**](https://k21academy.com/docker-kubernetes/kubernetes-networking/)**.**Click here

**Step 7 [Integration]:**Next is the **Integration.**

Here we keep all settings to default and move to the next step.

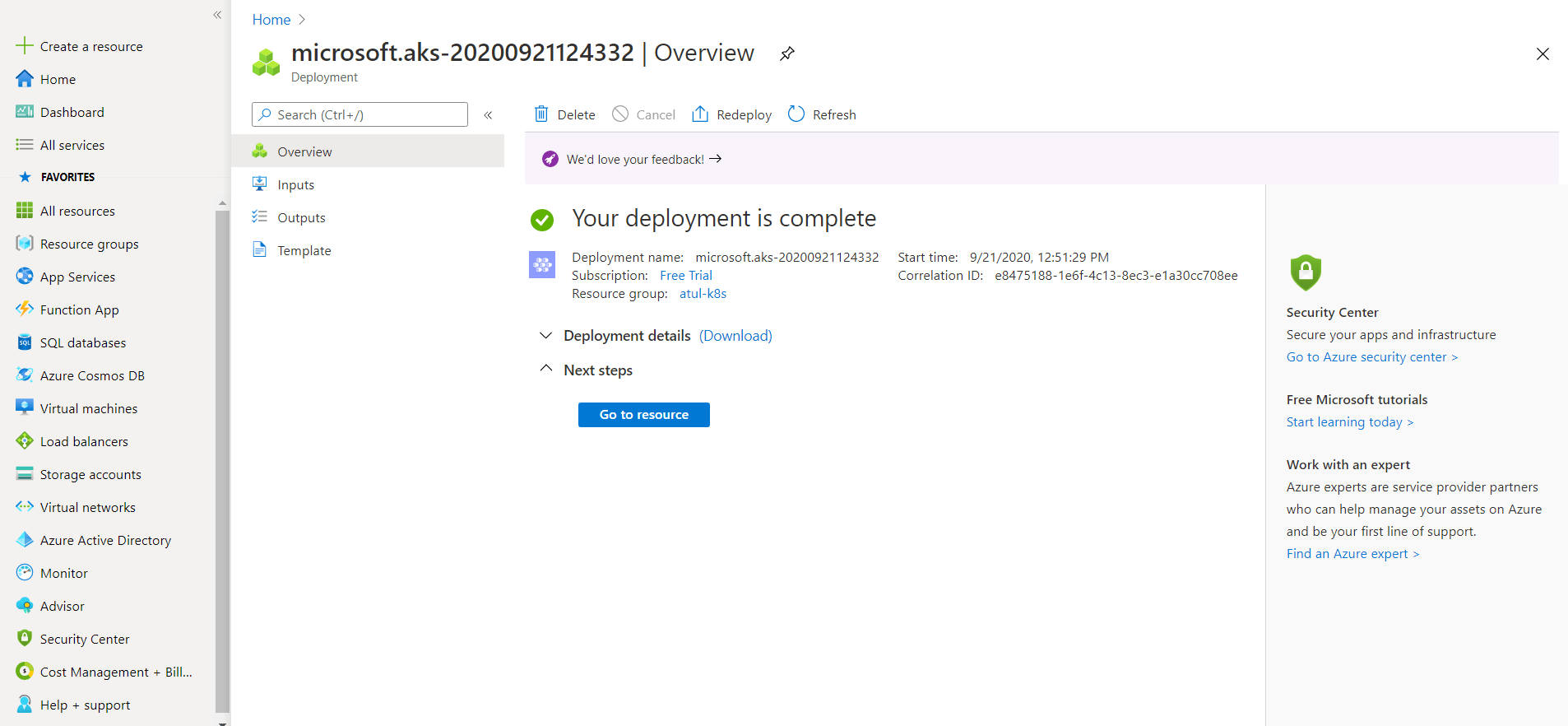


**Step 8 [Review & Create]:**The final step is to click on **Review & Create.** If you click on **Create**, it will first Validate your  AKS Cluster and if everything is fine then the cluster will be created.

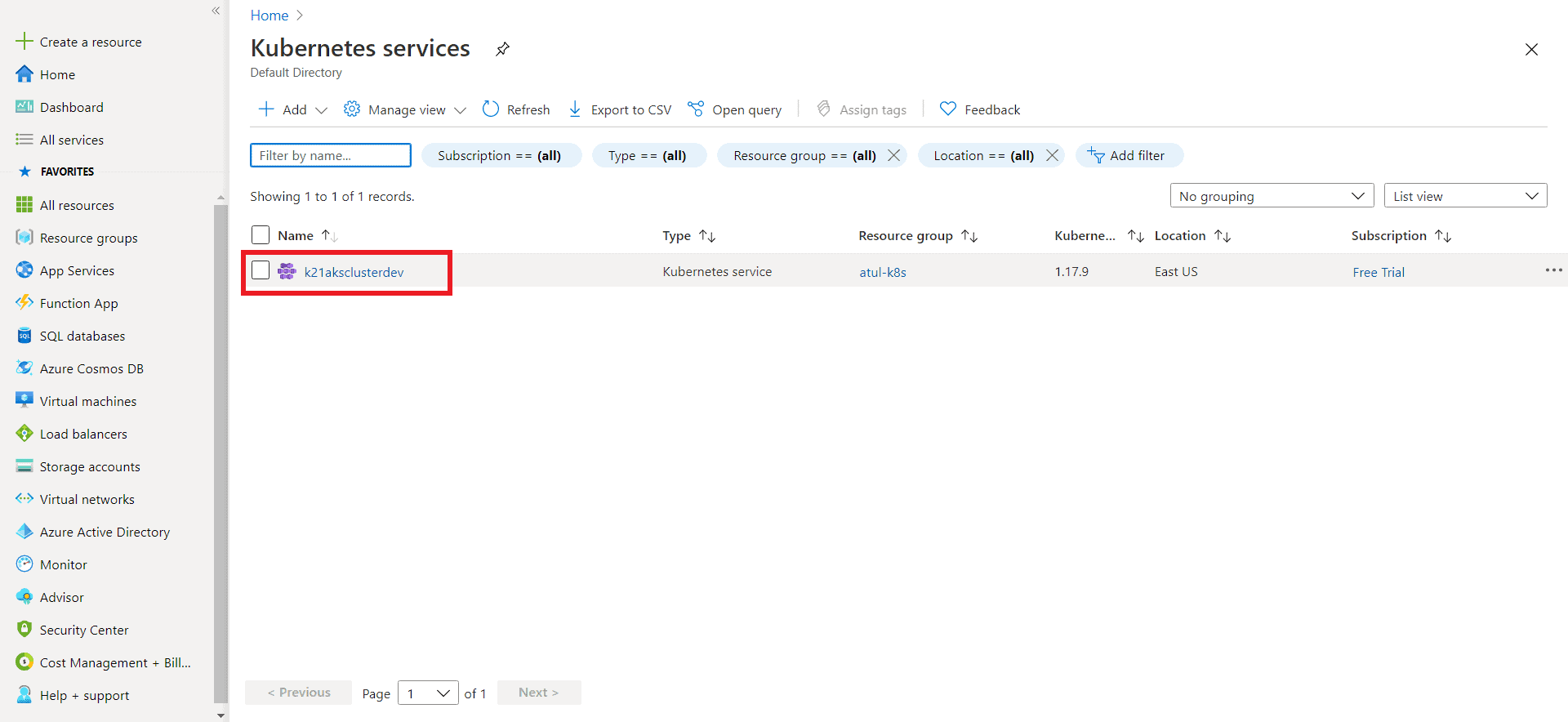


**Also Check:** what is a [Scheduler in Kubernetes](https://k21academy.com/docker-kubernetes/scheduling-in-kubernetes/)?

You can see that our new Azure Kubernetes cluster has been successfully created. Yayyy! But hold on, we will have to connect this cluster, which I am going to cover next.



To view the cluster, go to **Kubernetes services** and there you can access the AKS cluster.

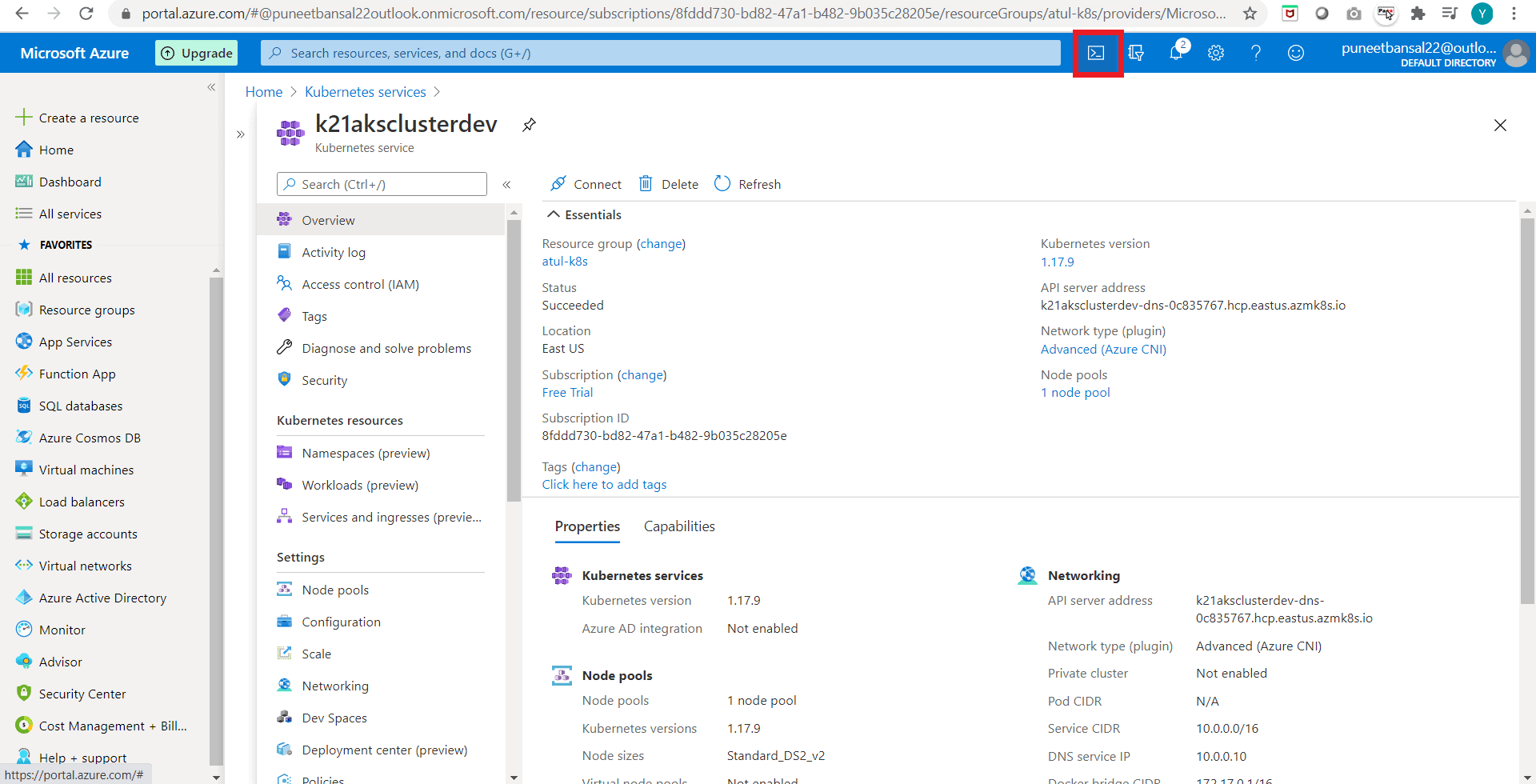


**Also check:** All you need to know about [Kubernetes RBAC](https://k21academy.com/docker-kubernetes/rbac-role-based-access-control/)

**Connect to the Azure Kubernetes Cluster**

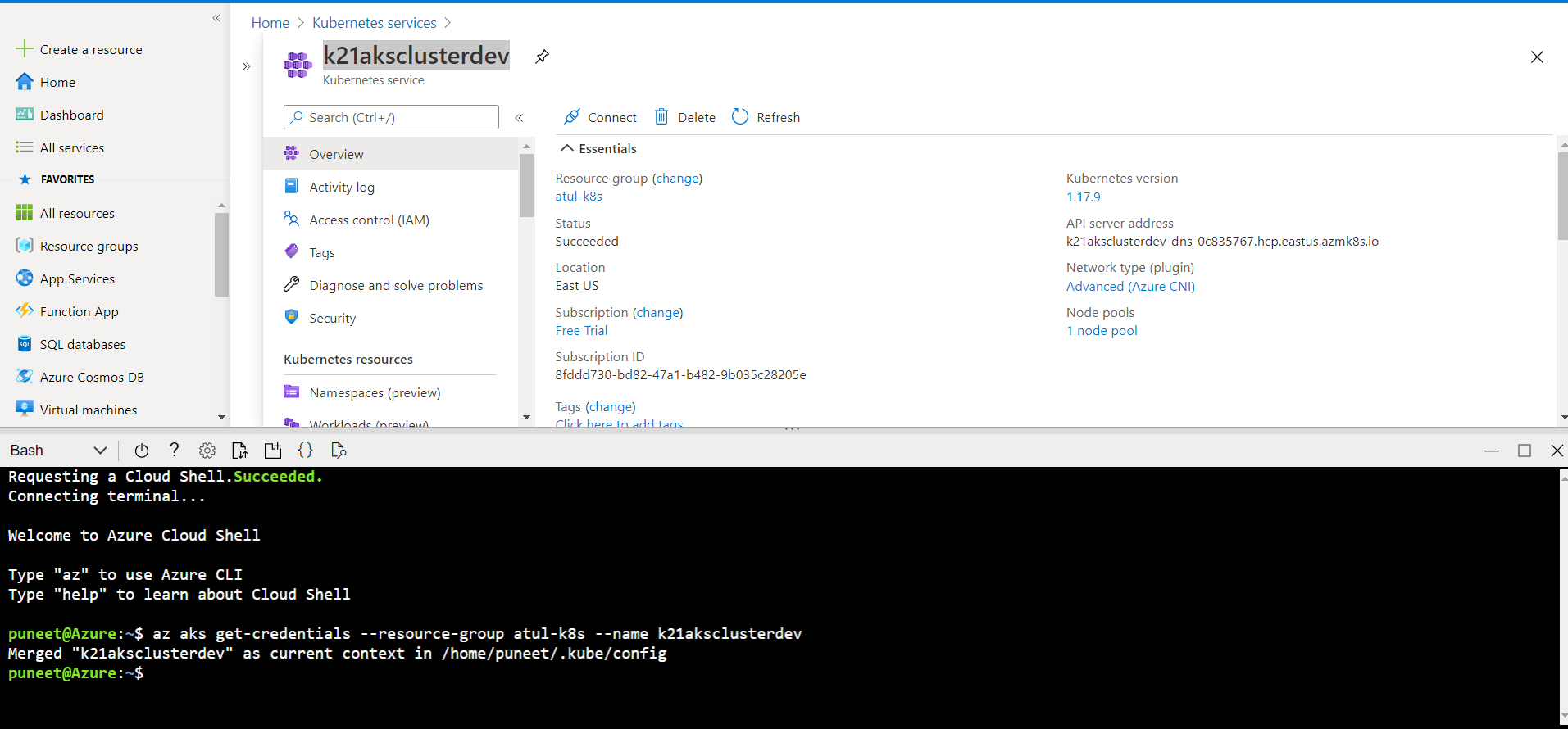
There are two ways to connect the AKS cluster:  
**I)**Using Cloud Shell  
**II)**Using Azure CLI

I am going to cover how do we connect using **Cloud Shell**. We can see the option on top of our screen (marked red in the below image).



**Step 1:**Run the following command, on the Azure bash shell:

$ az aks get-credentials --resource-groups <name of resource group> --name <name of cluster



**Check Out:**[Best Kubernetes Monitoring Tools](https://k21academy.com/docker-kubernetes/monitoring-in-kubernetes/). Click here

**Step 2:**To get the Nodes running in our cluster, run the following command, and you will see all the nodes in your AKS cluster.

$ kubectl get nodes

