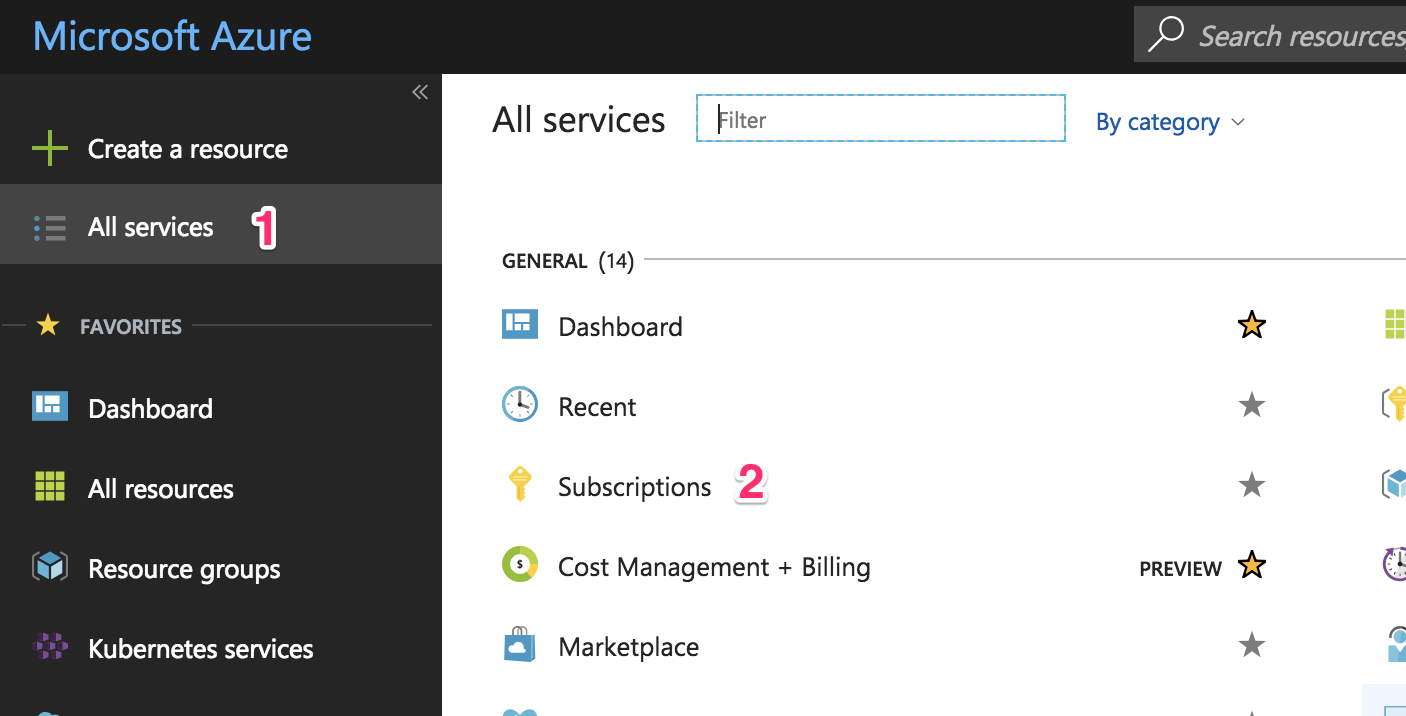
**Prerequisites:**

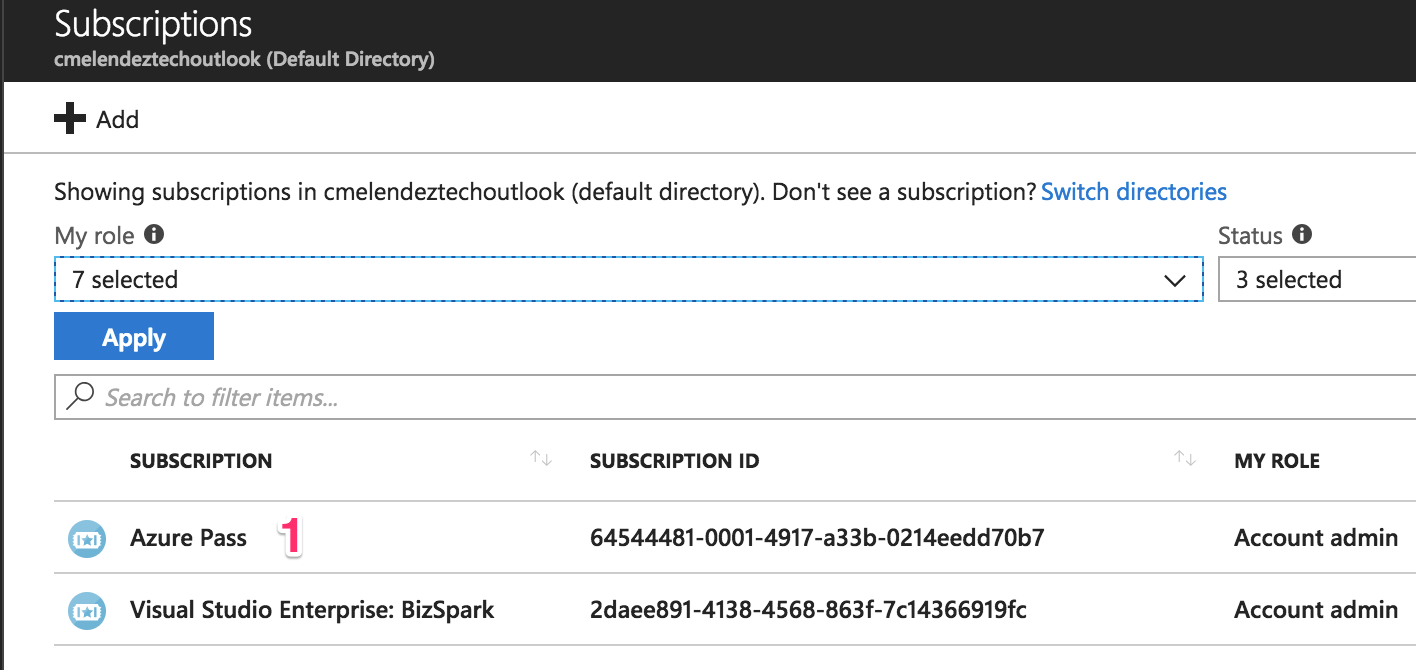
1. An [Azure](https://azure.microsoft.com/en-us/) account with a [subscription](https://theithollow.com/2016/07/11/azure-subscriptions/)
2. [Azure CLI](https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest) installed and configured
3. Kubernetes command-line tool [kubectl](https://kubernetes.io/docs/tasks/tools/install-kubectl/" \t "_blank) installed
4. Lastly, make sure the Azure subscription you use has the following required resources: Storage, Compute, Networking, and ContainerService

Register required resources in the subscription

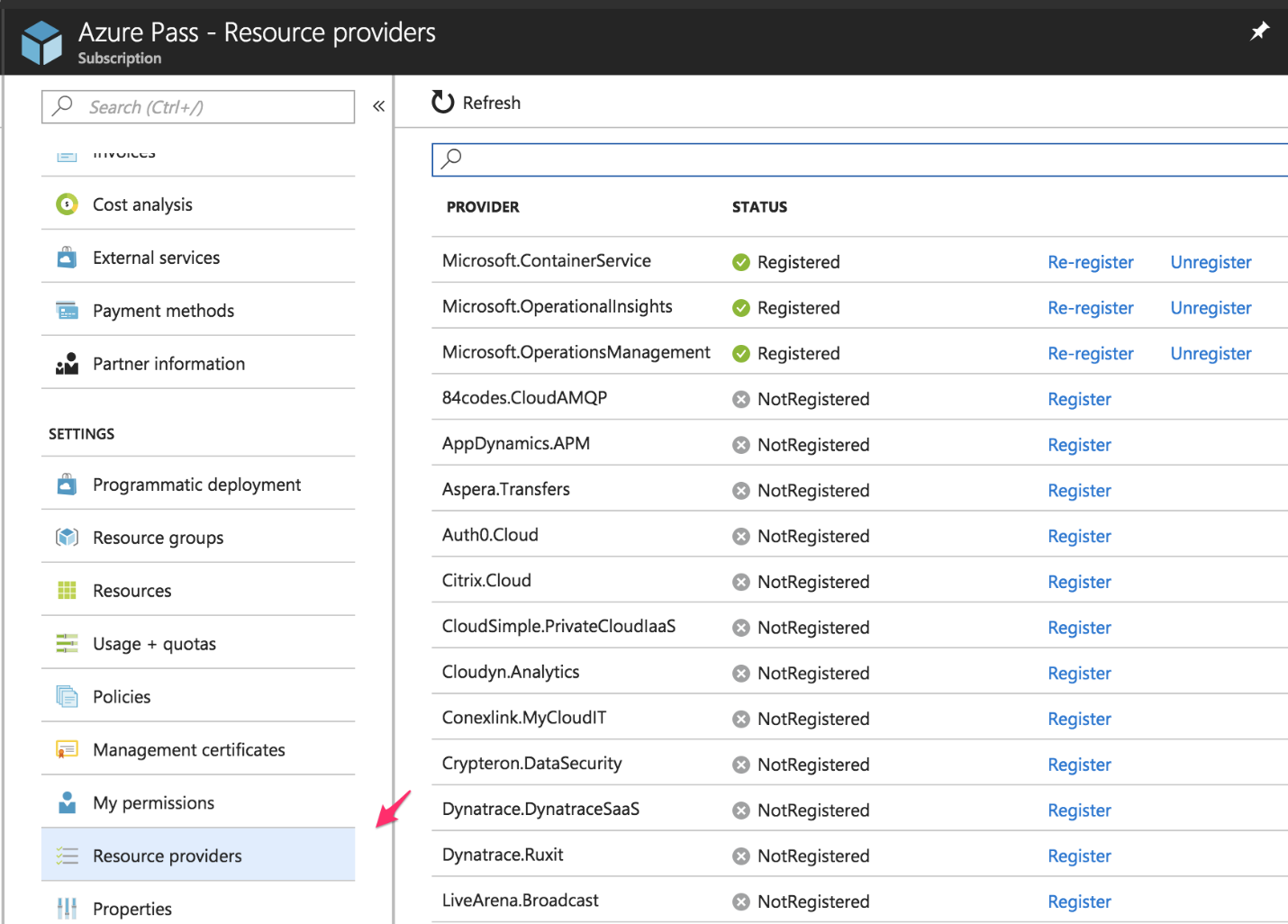
Go to the subscription page by clicking on the “All services” link (1) in the top-left panel, then click on the “Subscriptions” link (2). You can see the steps indicated in the screenshot below.



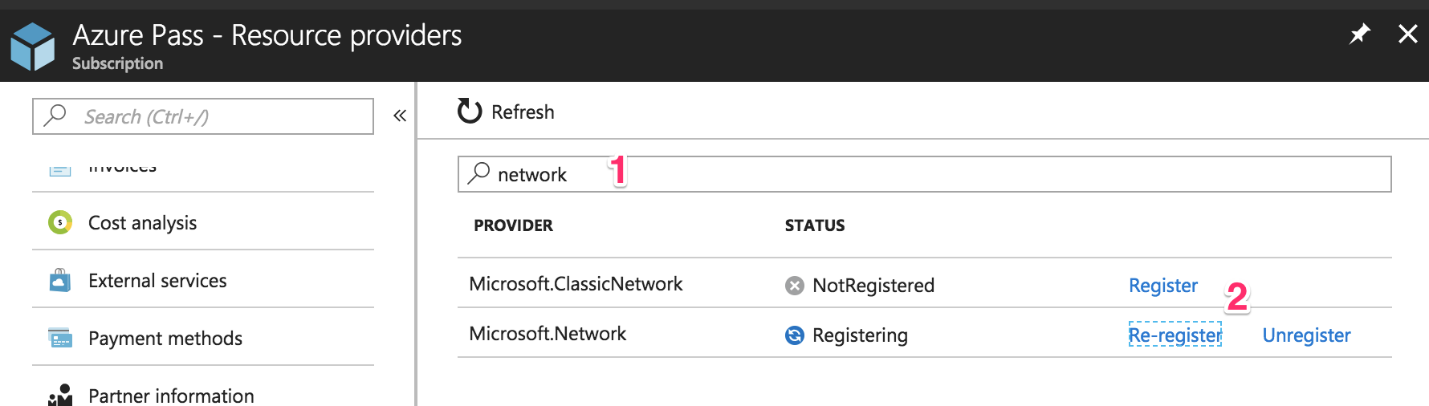
Select the subscription to to create the AKS cluster.



Scroll down and click on the “Resource providers” link to register or review the resources needed.



Search (1) for the resources listed before: Storage, Compute, Networking, and ContainerService. If it’s not registered, click on the “Register” (2) link and wait.



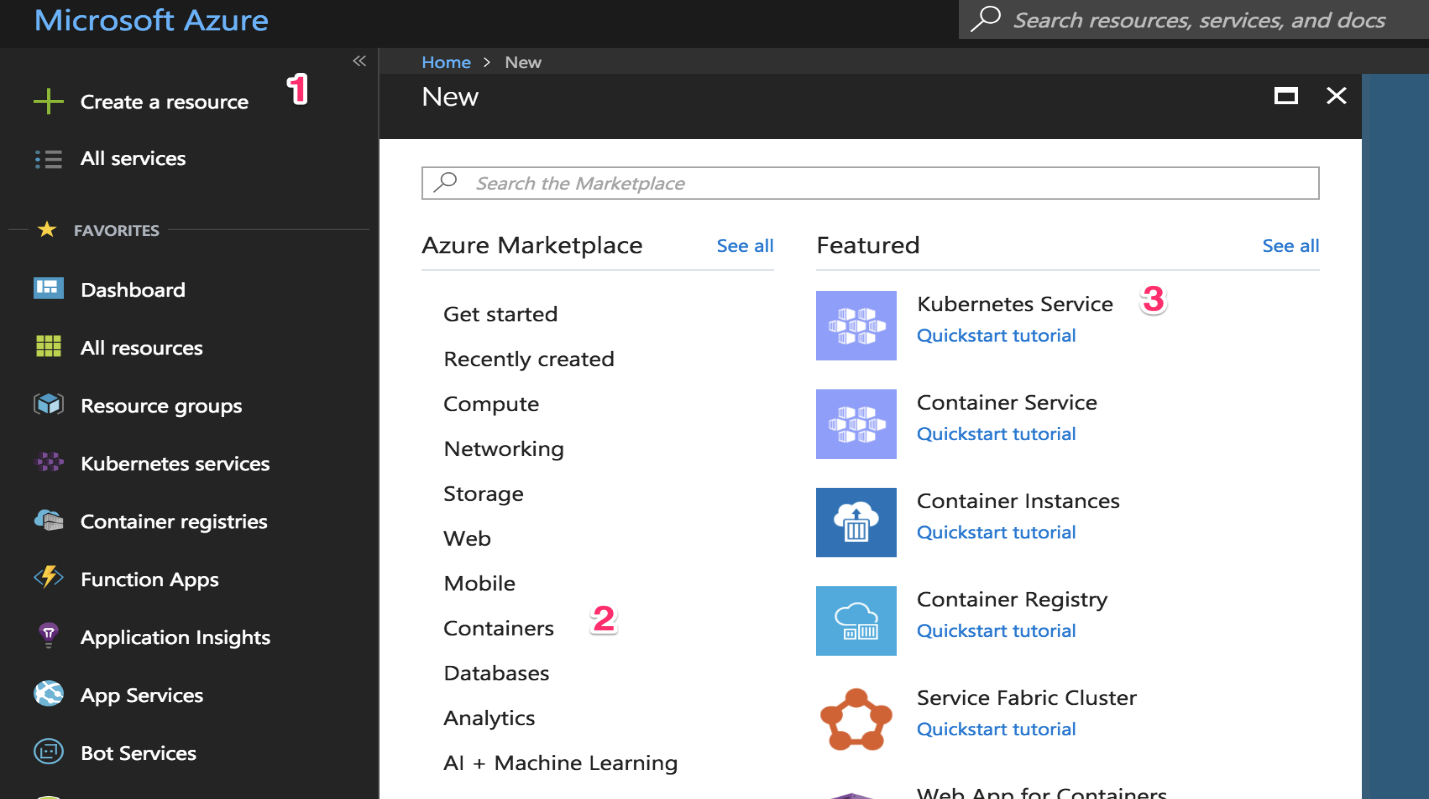
There are two methods that can be followed, which are given below:

Method 1: Creating an AKS/Azure Container Service cluster using the Azure Portal

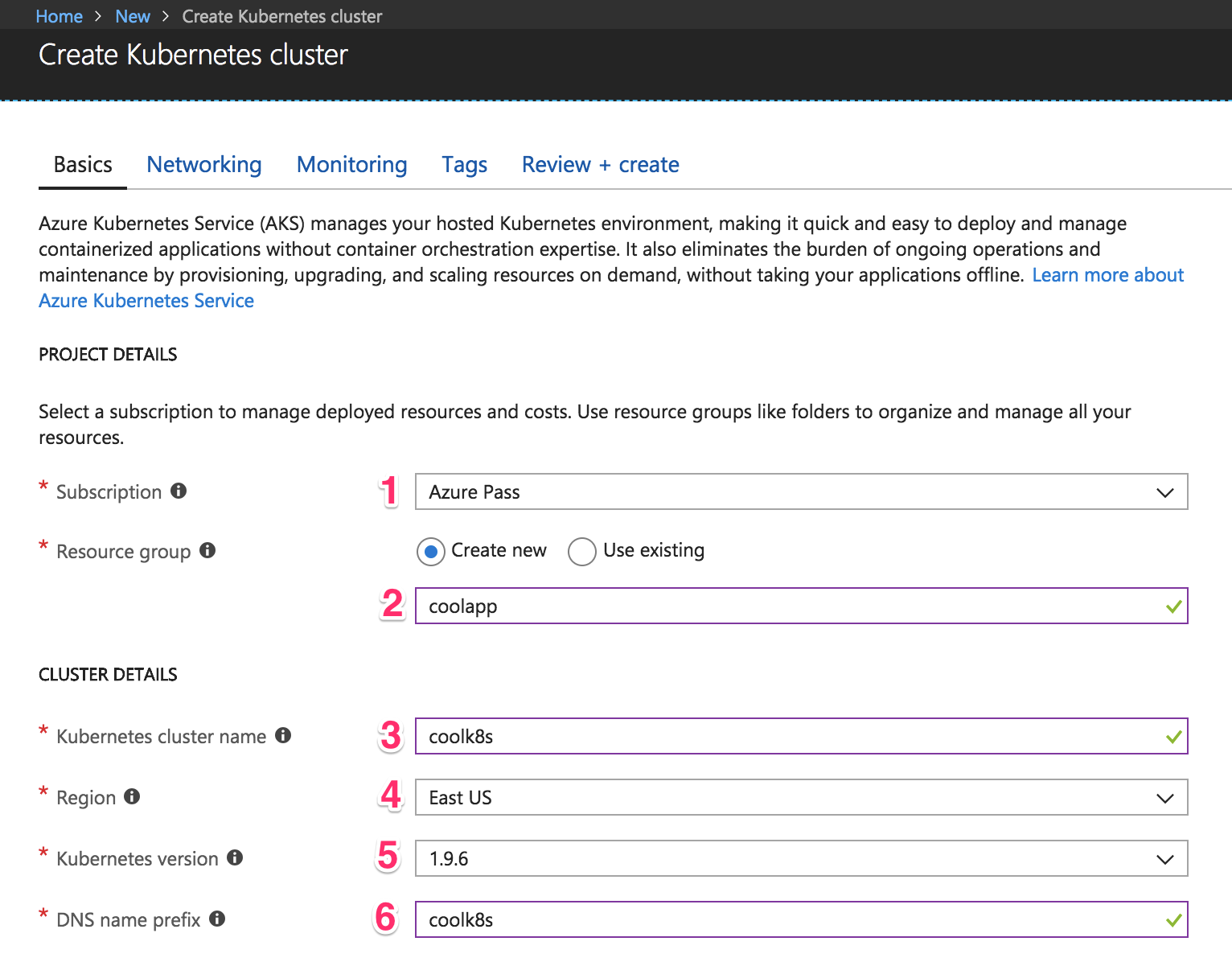
It is recommended to check the official AKS documents before proceeding further.

Create a new Azure resource

Go to your Azure portal and in the top-left panel, click the “Create a resource” (1) button. Then select “Containers” (2) and click on the “Kubernetes Service” (3) link.



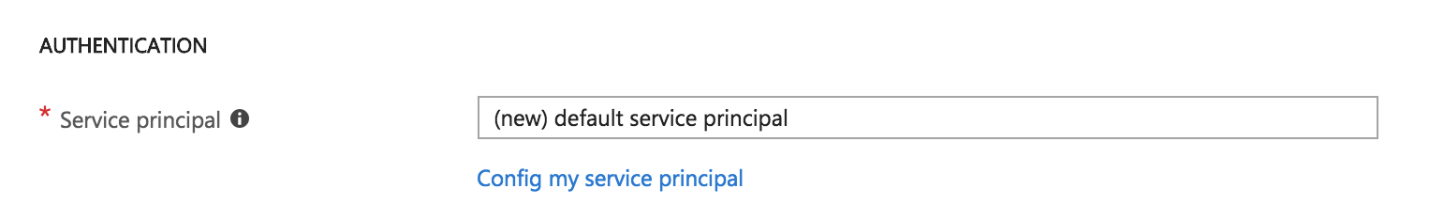
Fill In the Basics configuration:



Choose your subscription and use an existing resource group, but for our use let us just create a new one. A resource group is a way for Azure to keep all the related resources together so that you can make templates, share permissions and policies, or clean out everything by simply deleting the resource group. So choose “Create new” and name it—I put “coolapp” (2).

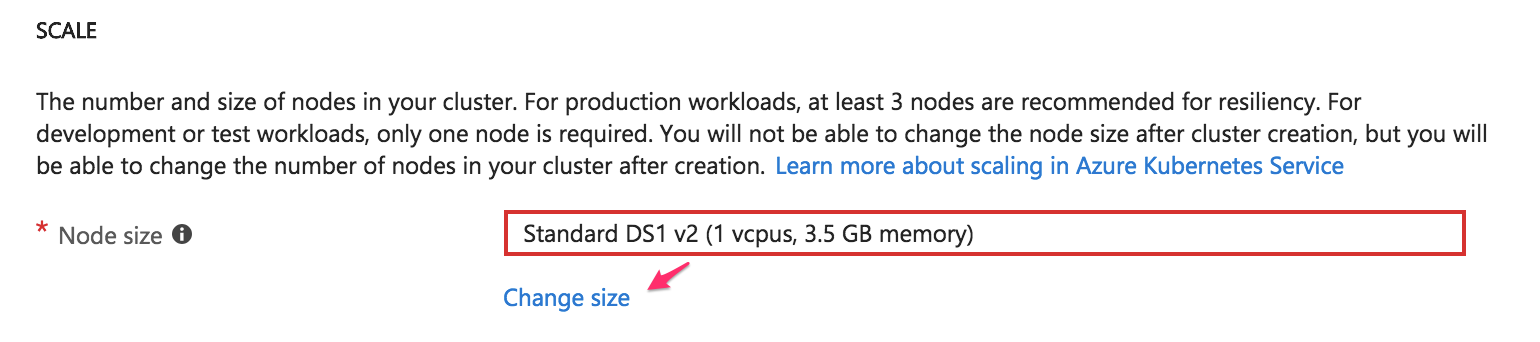
Choose a name for the cluster. I went with “coolk8s” (3). Choose a region where the cluster should be created (4). Ideally, it should be one that’s physically near your users (so if your users are based in the US, you want to create your virtual machine in a US region).

Select the latest version of Kubernetes (5) cluster. Then, write a DNS name (6) to identify your cluster—this should be unique for all the Azure users, so if you’re getting an error, it could be because someone else has already chosen the name you are trying to use. Go to the “Authentication” tab to see this:



Azure has made the service principal integration simpler—in case you’re just starting out, you can leave this option on default. A [service principal](https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-application-objects" \t "_blank) is needed so that AKS can interact securely with Azure to create resources like load balancers. Kubernetes’ services will sometimes need to be configured as load balancers, so AKS will create a real load balancer from Azure. In case you want to have more control and reuse a service principal, you can [create your own, too](https://docs.microsoft.com/en-us/azure/aks/kubernetes-service-principal" \t "_blank).

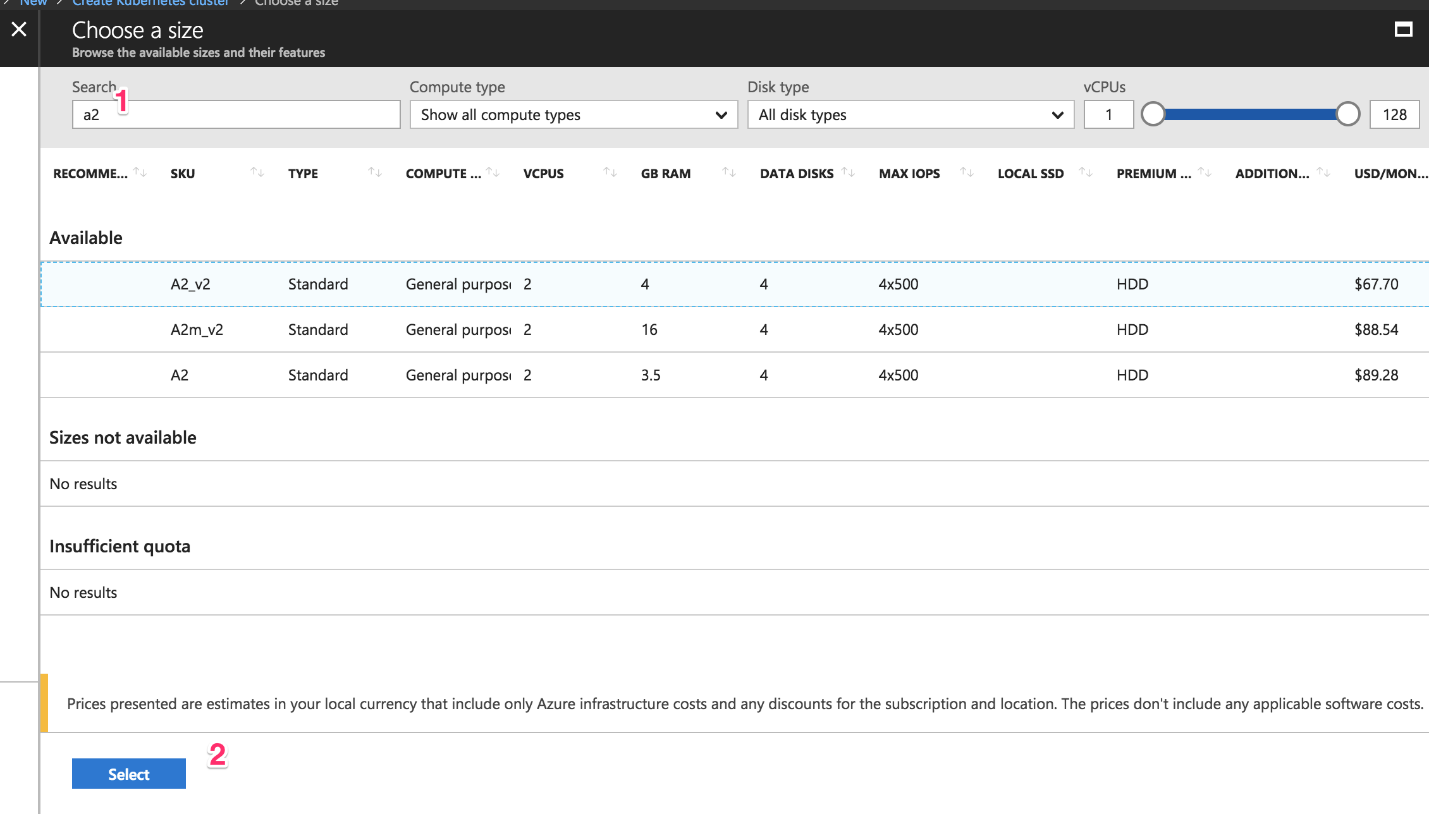
Now it’s time to select which type of instance the cluster will use. You might see an error screen like this:



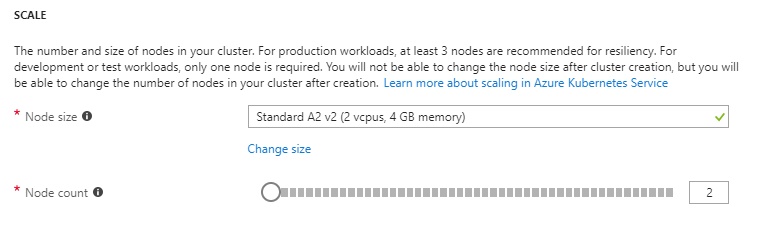
If you are having an “error,” it doesn’t necessarily mean that you are doing something wrong, it can also be because your subscription has some limitations. The reason can be that you have reached the instances limit or that creating the cluster will put you over the limit. All you need to do is to get rid of the error message is click on the “Change size” link and choose a different instance type.

Node Size:

While creating the cluster, if a small instance for AKS, then it can cause an error. In this scenario one would be required to select an instance with at least 3.5 GB of memory, otherwise the cluster would not be created. The given screen should be followed for help:

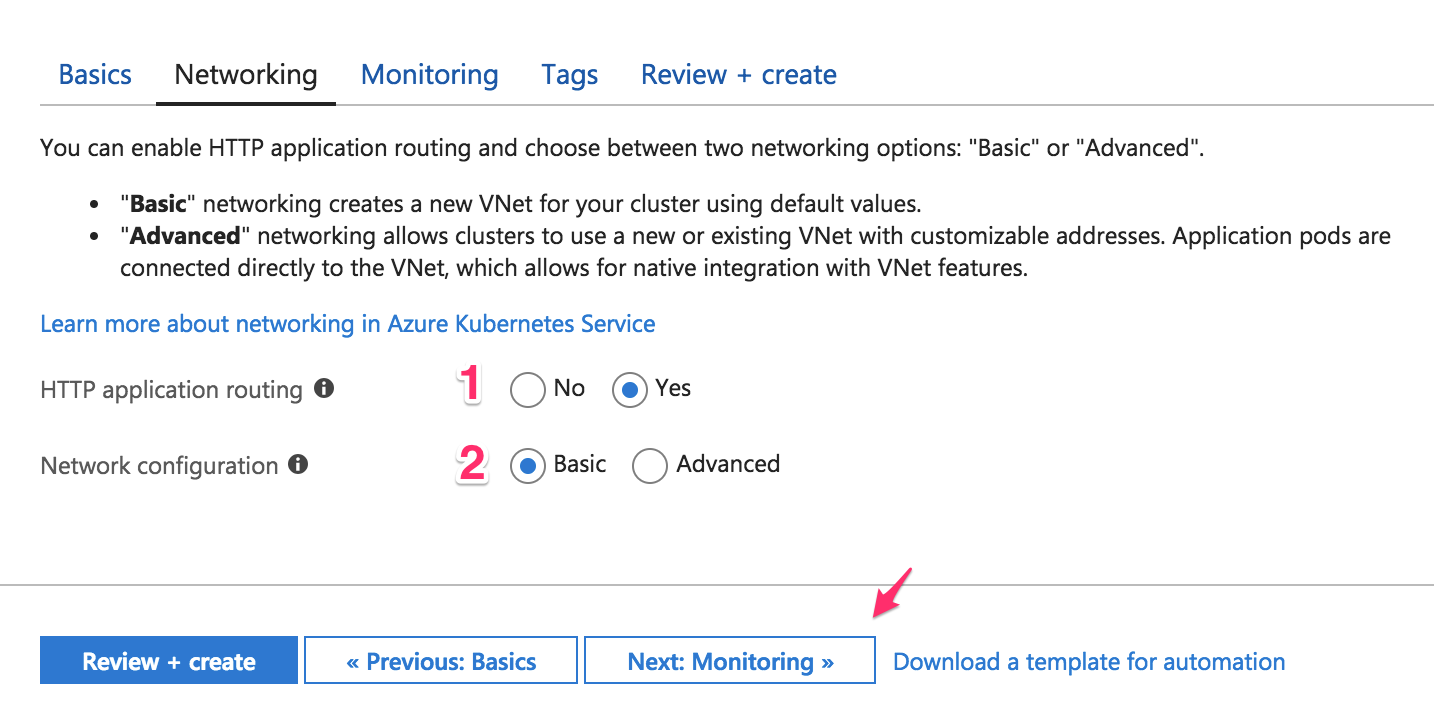


Type “a2” (1) so that the “**Standard A2\_v2″** instance type appears and click it. Next, click on the blue “Select” (2) button. It will take you back to the previous screen.



Set “Node count” option to 2. Once done, click on the “Next: Networking” button.

Fill in the networking configuration

Networking is one of the most important things to configure when you start integrating services, or if you want to create a VPN tunnel. This is the part where you need to avoid any networking conflict to access the Kubernetes nodes that AKS configures. Click on the “Networking” tab. You should now be able to view the below mentioned screen:  


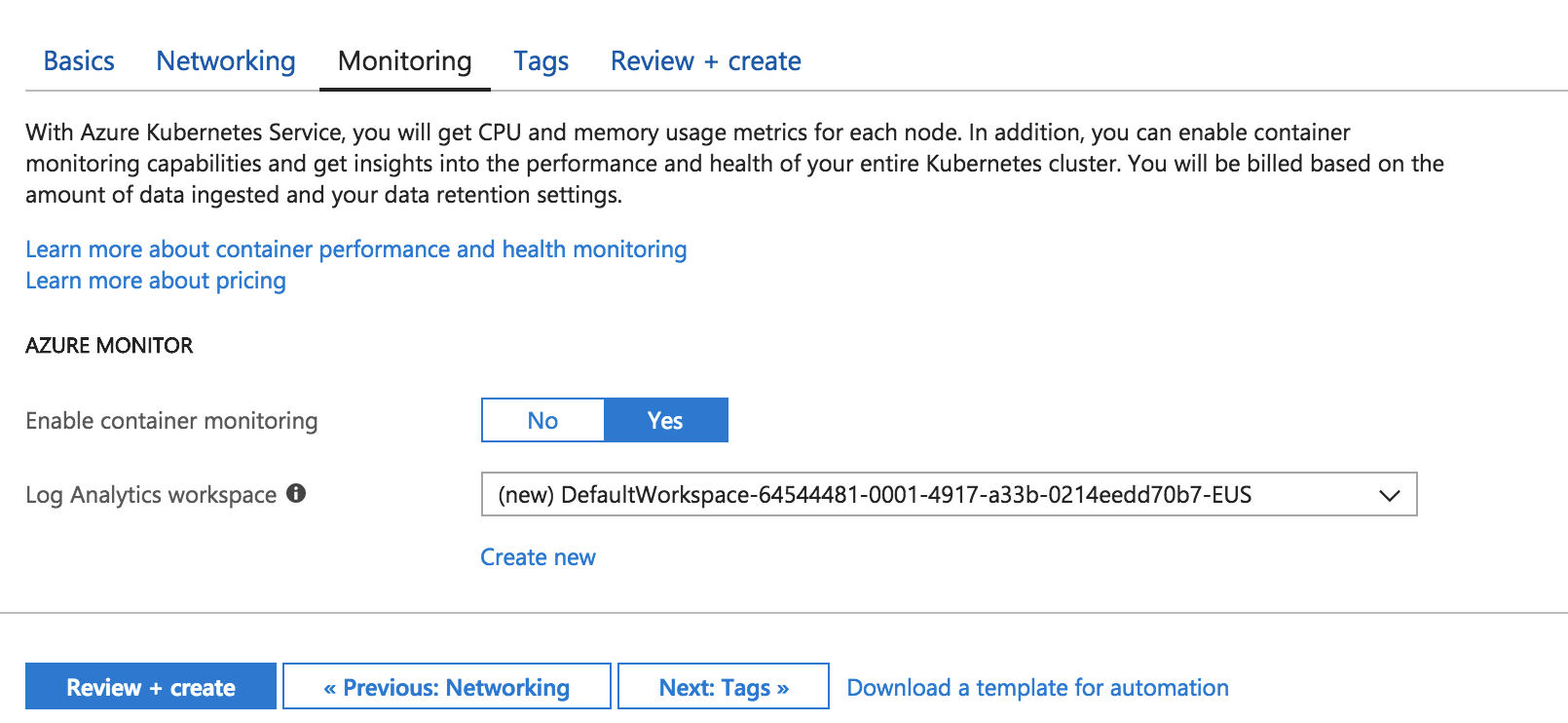
You can leave this section as default and continue with those settings. In case you want to do something specific with networking, then follow the given steps:

Start with the HTTP application routing (1). When you create a Kubernetes “LoadBalancer” service type, a public IP address is assigned to you. At some point, you may encounter a few issues while creating a new service because you’ve reached an Azure [limit](https://docs.microsoft.com/en-us/azure/azure-subscription-service-limits" \t "_blank). Kubernetes has “a collection of rules that allow inbound connections to reach the cluster services” called Ingress. Ingress will allow you to have SSL termination and DNS endpoints for your services.

For networking configuration (2), choose “Basic” for now and let Azure configure the networking for you. This is where you can define network ranges to allocate IPs—usually known as [CIDR](https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing" \t "_blank). This section is crucial for avoiding network conflicts with your on-prem network or other network resources in Azure. Click on the “Next: Monitoring” button to continue.

Fill in the monitoring configuration

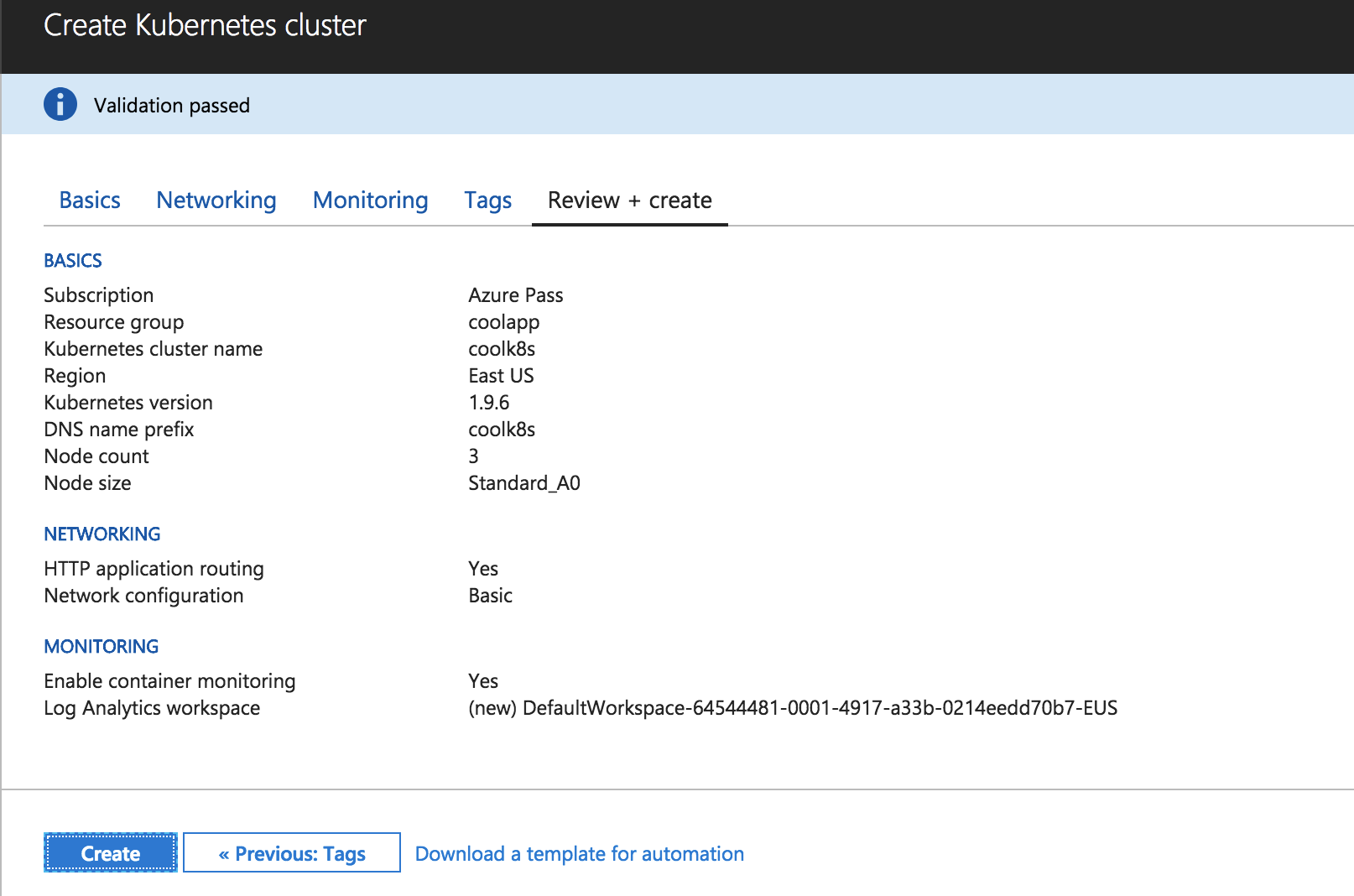
By default, AKS will give you metrics about the performance of the cluster, and if you choose, you can get performance metrics for containers. You may access all the logs to troubleshoot. These monitoring features are invaluable when you are integrating [APM](https://stackify.com/avoid-apm-vanity-metrics/" \t "_blank) into your application.

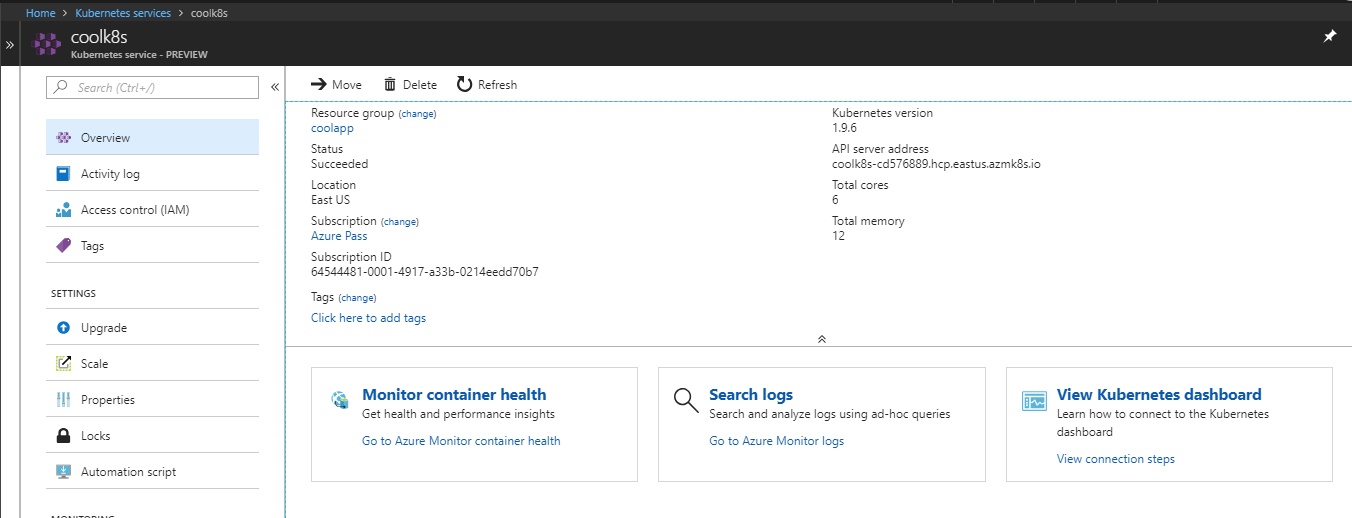


Finally, click on the blue “Review + create” button to continue.

Review and create

Azure will start creating the service principal and will validate the information you have entered. The image given below may be used for reference:



Click on the blue “Create” button to create the cluster. 

The Kubernetes-managed cluster is created in Azure.

Method 2: Creating an Azure Container Service cluster using the command line

The UI has been changing throughout the time, but the command line is constant. The GUI gives us a chance to understand visually how to create the cluster. On the other hand, using the CLI will helps to automate the process or use tools like Terraform.

Before we start creating the cluster using the command line make sure you delete or change the names of the resources in the commands you have used earlier.

Create a new resource group

A resource group is a way for Azure to keep all related resources together so that you can make templates, share permissions and policies, or clean out everything by simply deleting the resource group. So run the following command to create it:

az group create --name coolapp --location eastus

Create the AKS cluster

Run the following command to create the cluster:

az aks create --resource-group coolapp --name coolk8s --node-count 2 --node-vm-size Standard\_A2\_v2 --generate-ssh-keys

The parameters used are explained below:

* **–resource-group** is the name of the resource group we just created
* **–name** is the name of the cluster to identify it
* **–node-count** is the number of nodes we want for our cluster
* **–node-vm-size** is the name of the instance type we choose
* **–generate-ssh-keys** will generate SSH keys on your local machine so it’s easier for you to connect to any node if need be

Specifying only those arguments means that Azure will use the default values for things like networking or monitoring. Finally Azure will create the cluster.

Accessing the Kubernetes UI locally

Whether you created the cluster using the portal, the command line, or both, instructions will are to be followed to access the Kubernetes dashboard. It is required to have the latest version of [Azure CLI](https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest" \t "_blank) and [kubetcl](https://kubernetes.io/docs/tasks/tools/install-kubectl/" \t "_blank) installed and configured.

Download cluster credentials

Firstly download the cluster credentials to your computer by running the below given command:

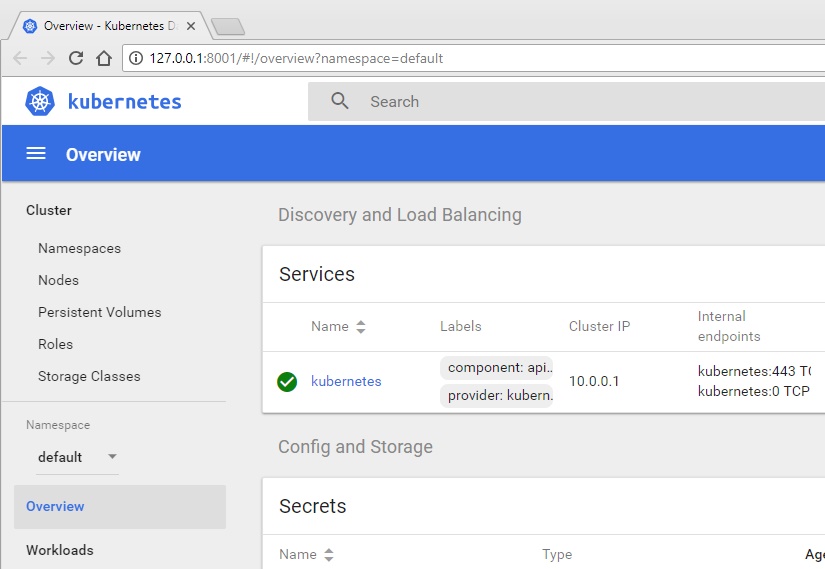
az aks get-credentials --resource-group=coolapp --name=coolk8s

Specify the resource group with the **–resource-group** parameter and the name of the cluster with the**–name** parameter. This makes toggling between different Kubernetes clusters effortless.

Browse the cluster

Run the following command. A new browser tab or window will open with the Kubernetes dashboard automatically.

az aks browse --resource-group coolapp --name coolk8s



If the screen appears similar to the one mentioned above, then the cluster is functioning with ease.

Clean up your resources

Delete the AKS cluster by running the following command:

az aks delete --resource-group coolapp --name coolk8s

If you want to delete the resource group, run the below given command:

az group delete -n coolapp