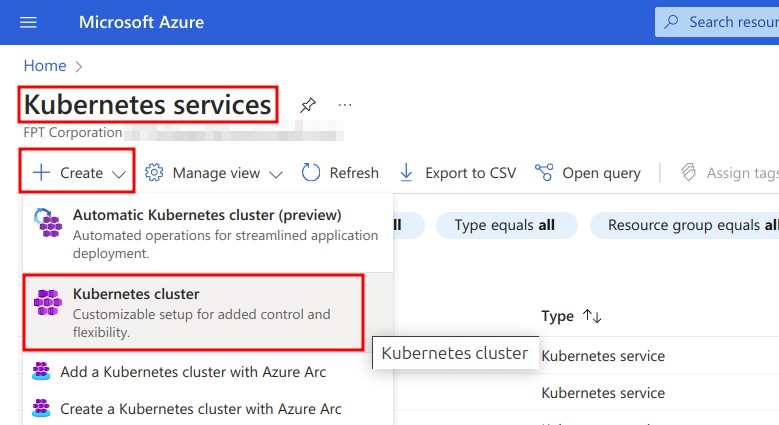
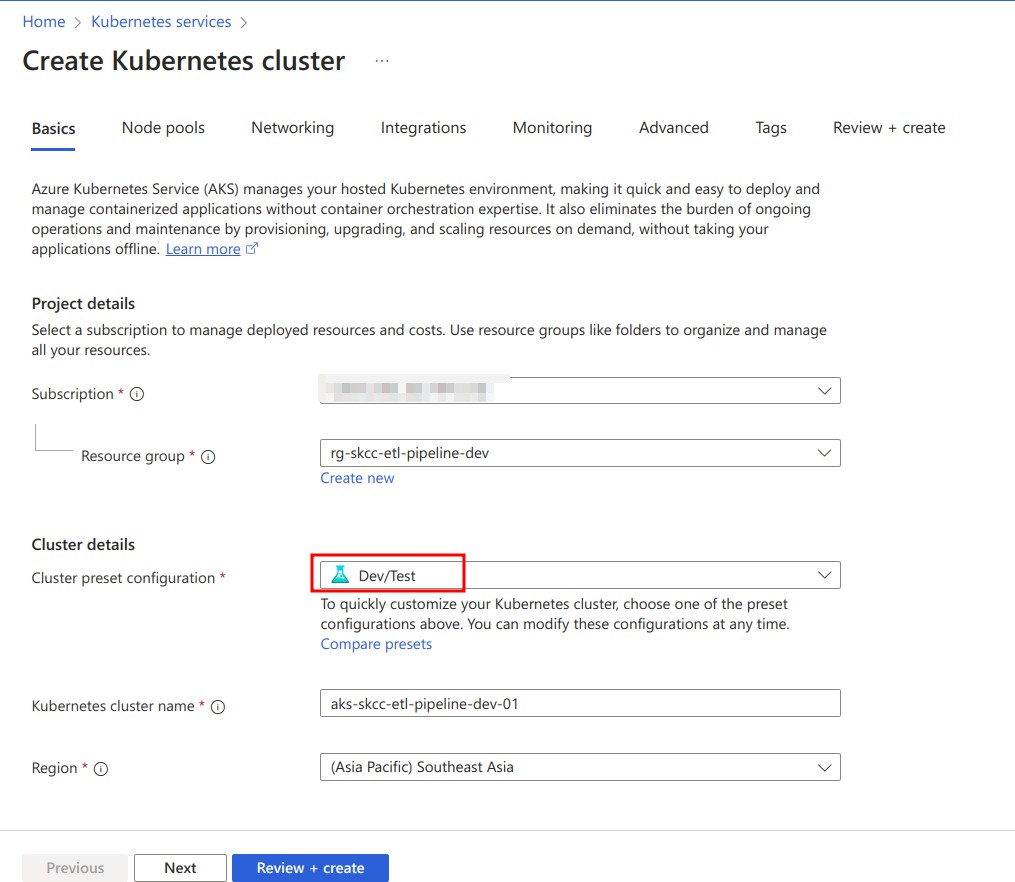
Guide to setup a new AKS cluster for developing environment

# Create a new AKS cluster

**Step 1:** Go to Azure Portal, Kubernetes services, and select “Create \* Kubernetes clusterˮ.

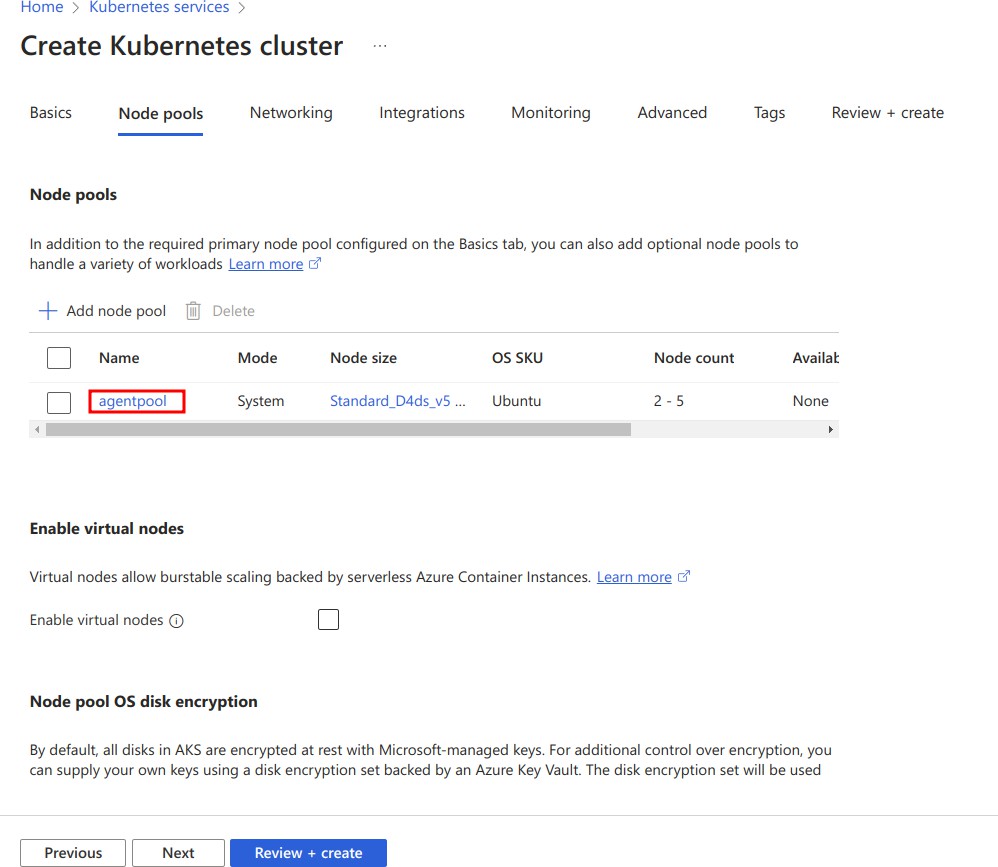


**Step 2:** Choose a subscription, and resource group(create a new one in case does not have one). For developing purposes, set up “Cluster preset configurationˮ with the value “Dev/Testˮ to optimize cost. Then, fill in a cluster name and choose an expected region for the new cluster. We can keep other options by default values.

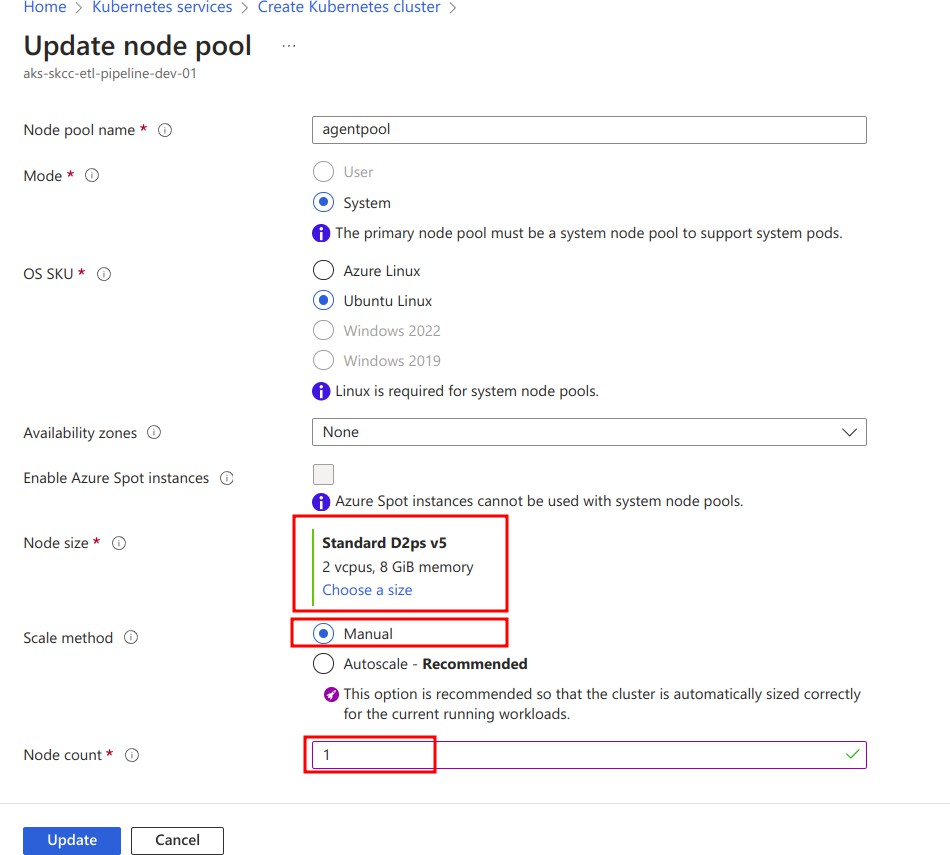


**Step 3:** Config node pools

In the Node pools section, click “agentpoolˮ to edit the default pool.

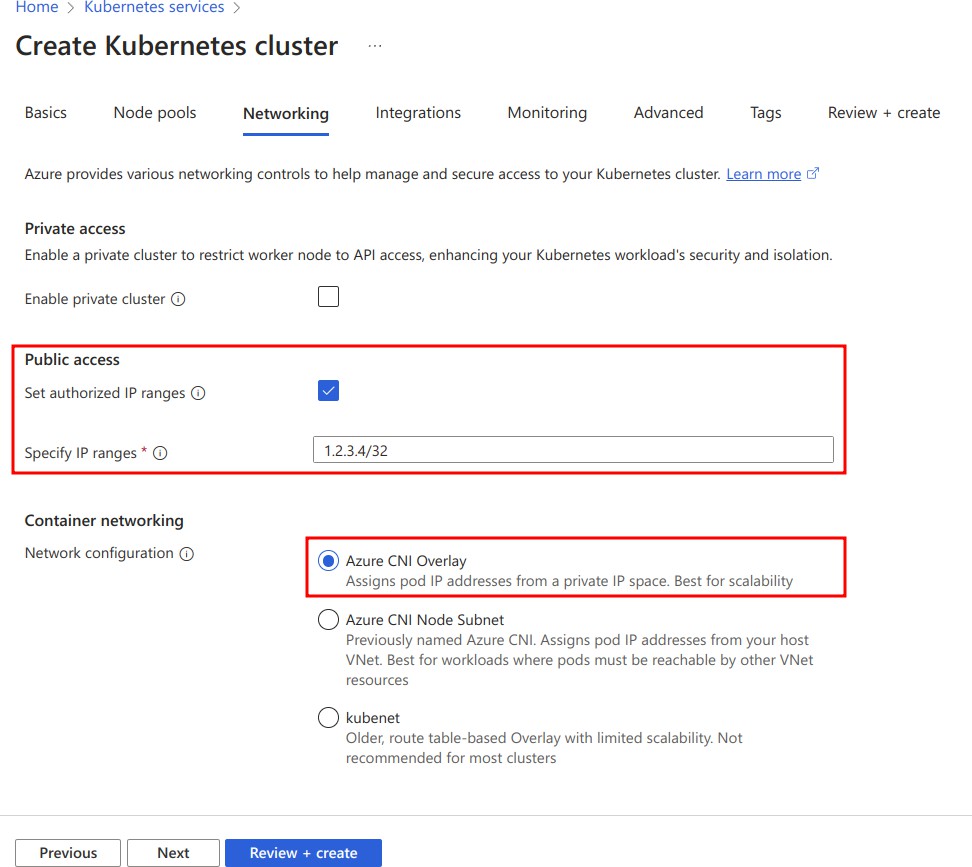


Then, select the node size that fits for developing purposes, here I choose D2ps\_v5 with 2 vcpus and 8 GiB memory for example. Note that for the dev environment, we should use the manual scale method and set the node count manually. Click Update when done.

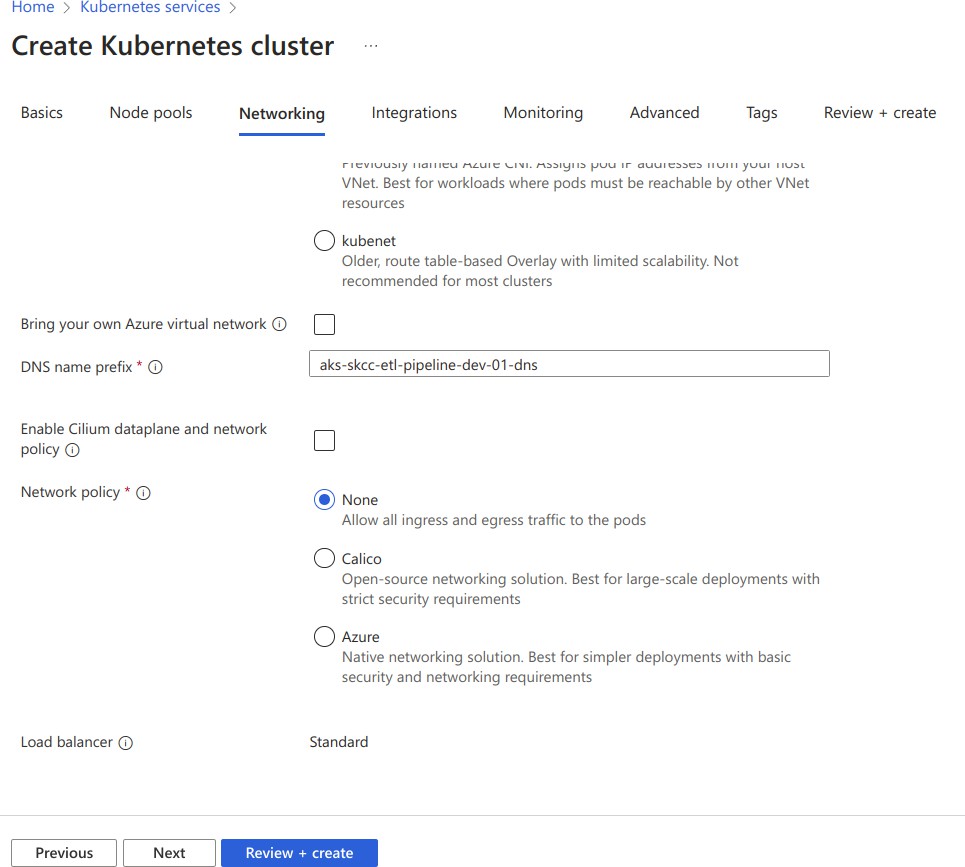


**Step 4:** Config networking

To access our cluster from the internet, enable public access by choosing “Set authorized IP rangesˮ and fill in the IP ranges that we want to connect to the cluster api-server. For the CNI plugins, keep the default option as “Azure CNI Overlayˮ.

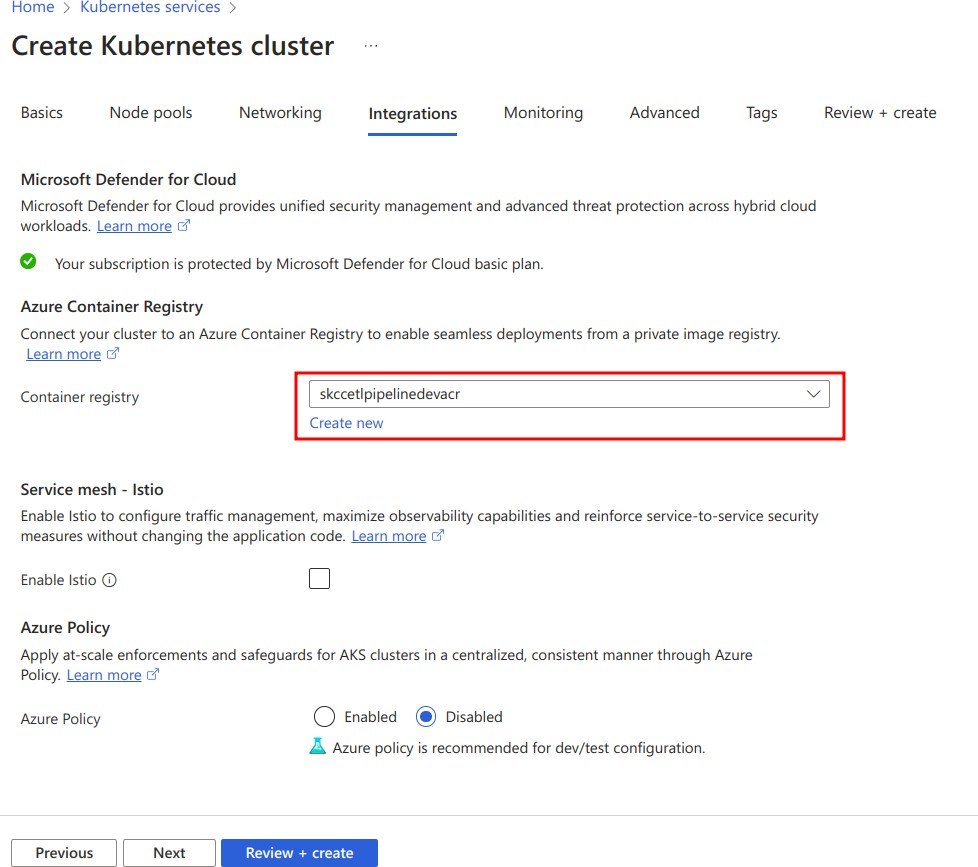


We can keep all other configurations with default values. But in case we need to deploy Network Policy to our cluster, choose Calico or Azure if we want to use one of two built-in solutions.



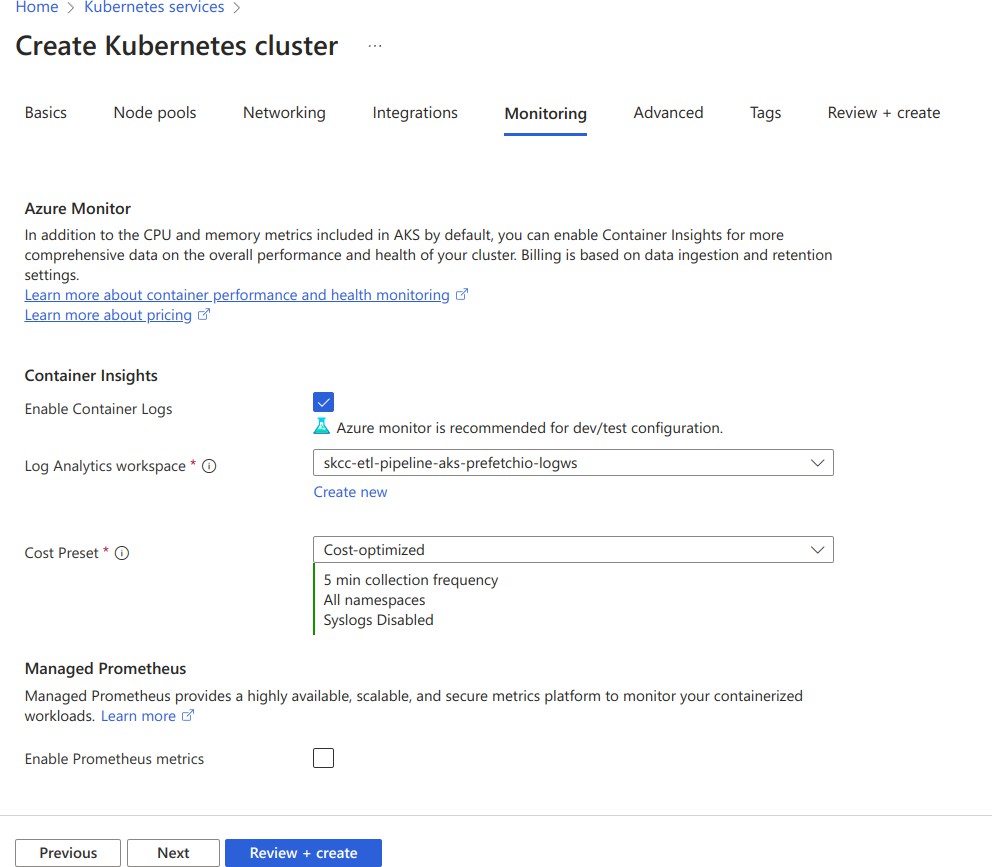
**Step 5:** Config Integrations

If we need to use ACR to store images that will be deployed into the cluster, select an existing registry or create a new one.



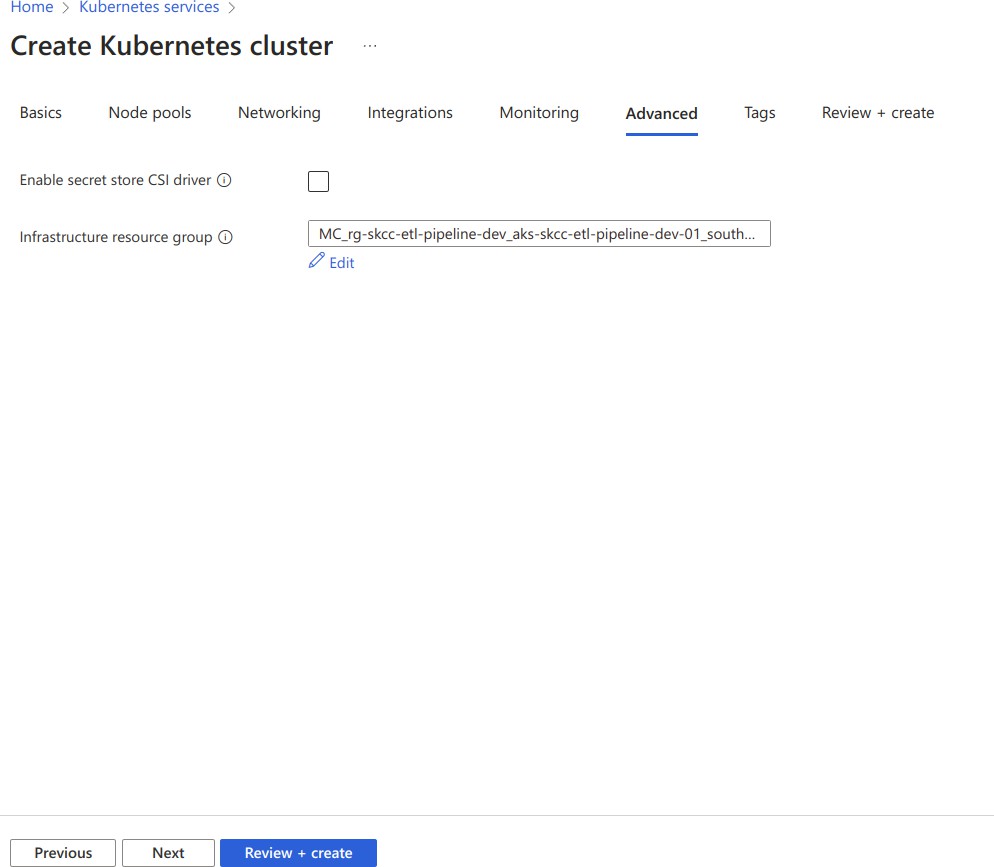
**Step 6:** Config monitoring

For the monitoring section, we can set up Container Insights if needed. Other options like Prometheus and Grafana should be turned off to optimize costs.



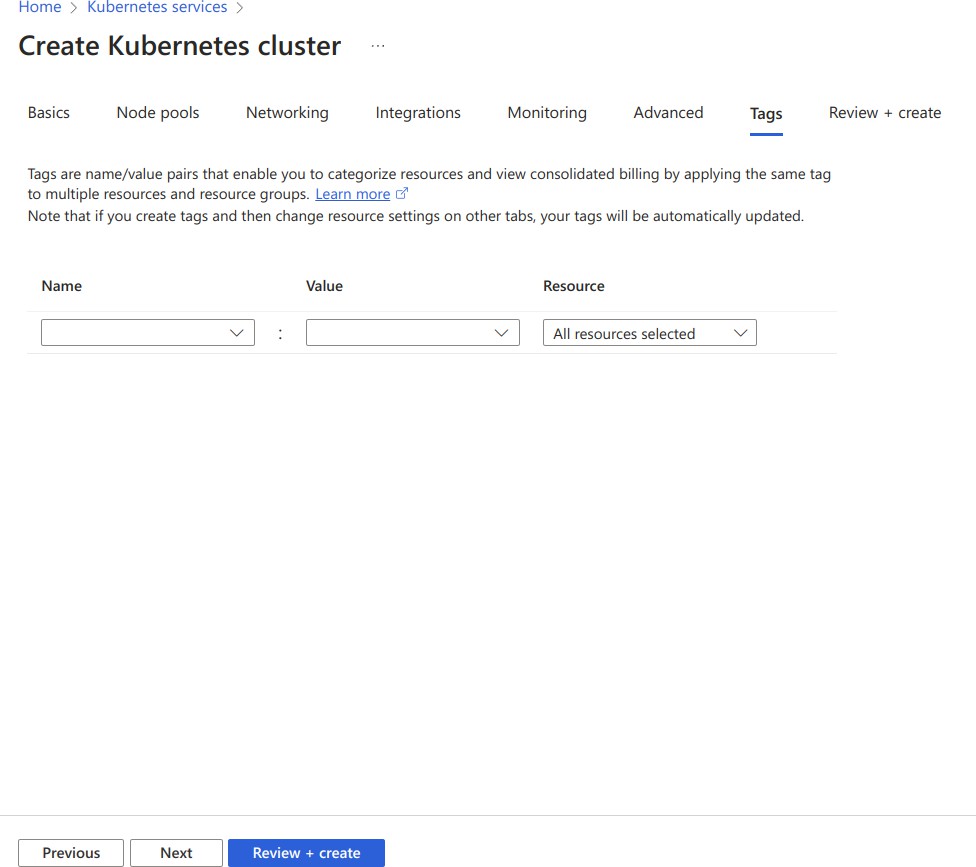
**Step 7:** Config advanced settings

For the dev environment, we can consider not using CSI drivers to cut down the cost.



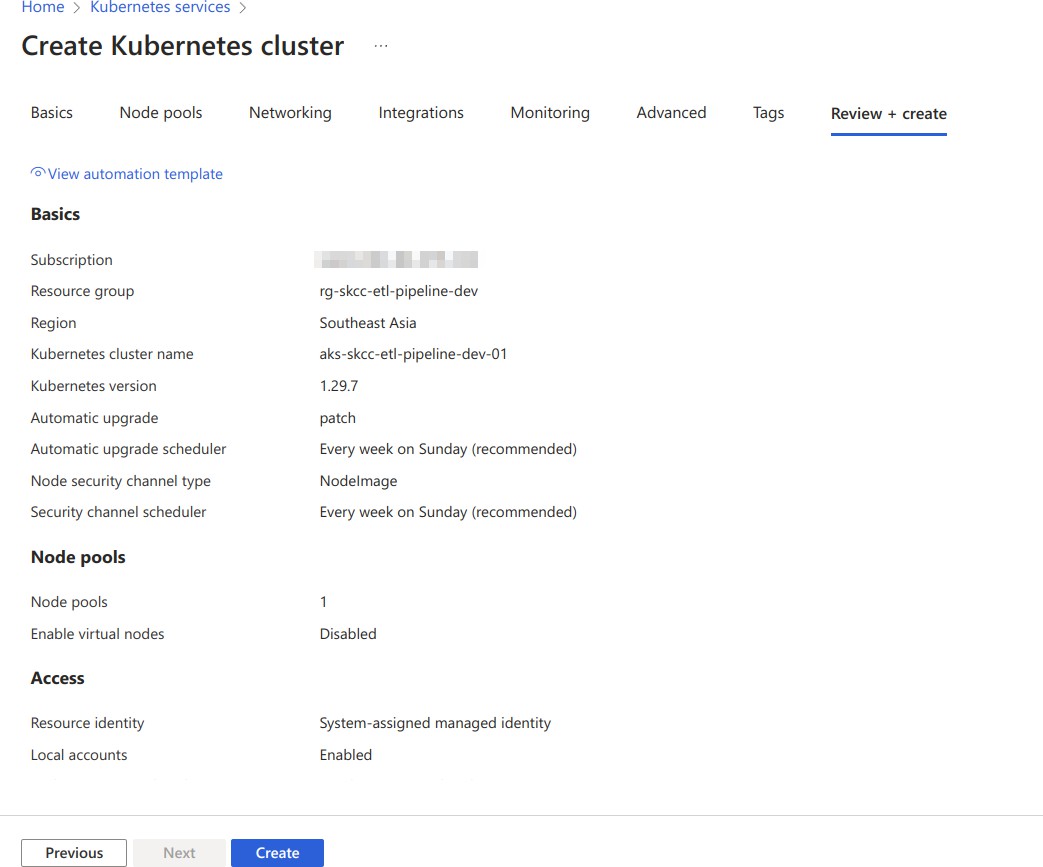
**Step 8:** Tagging

We can add tags to our cluster for management purposes if needed.



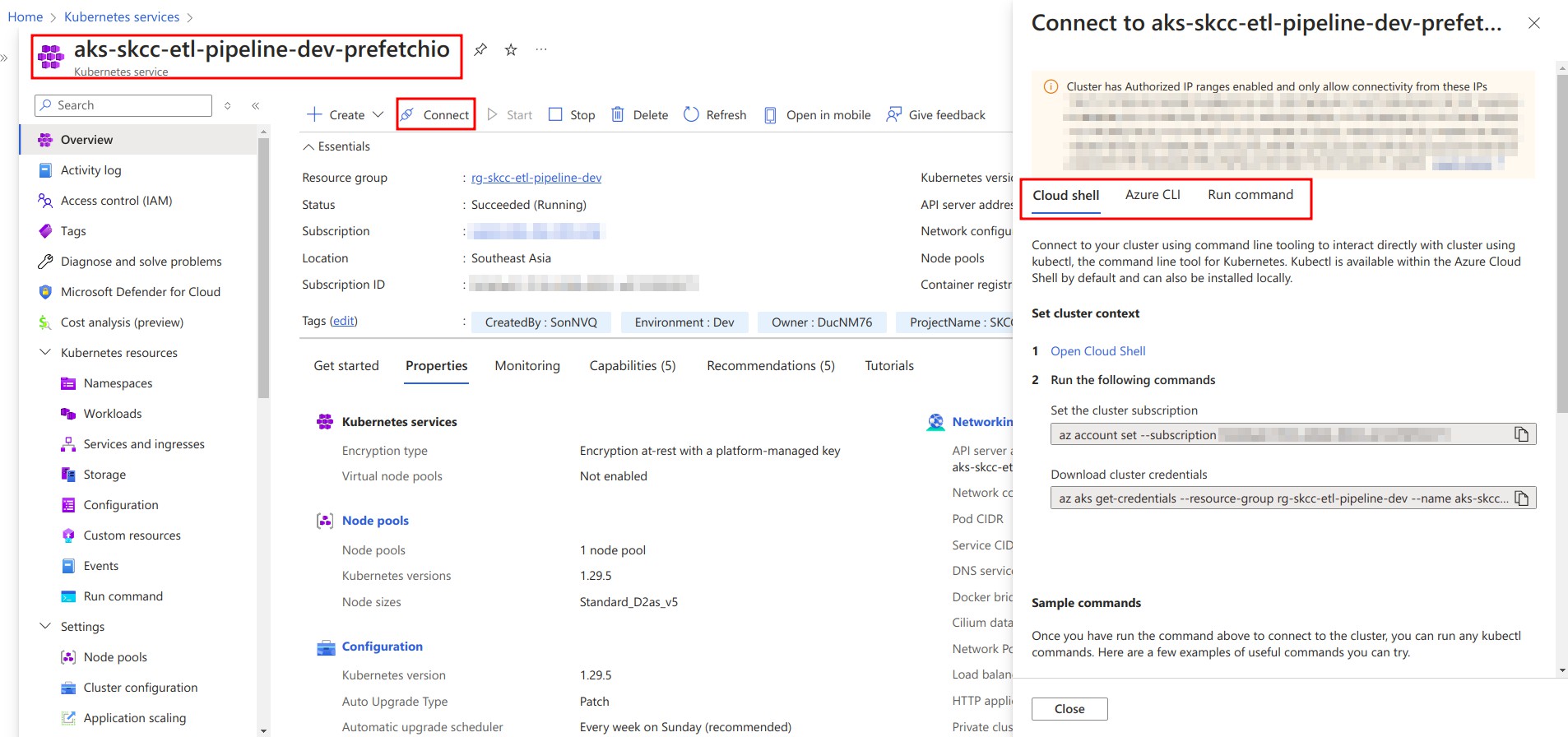
**Step 9:** Review + create

Carefully review our configurations and click Create to deploy our new AKS cluster.

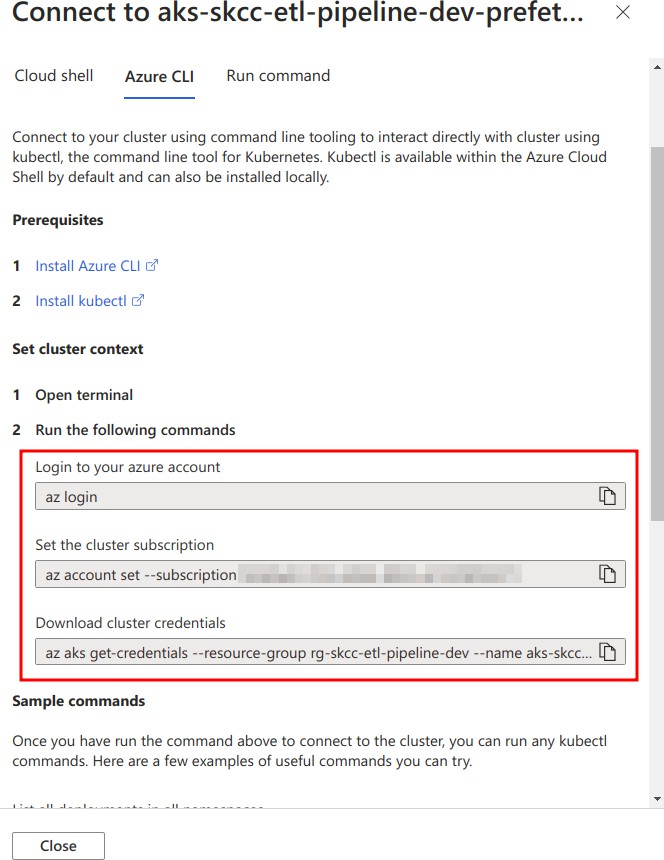


# Connect to the new cluster

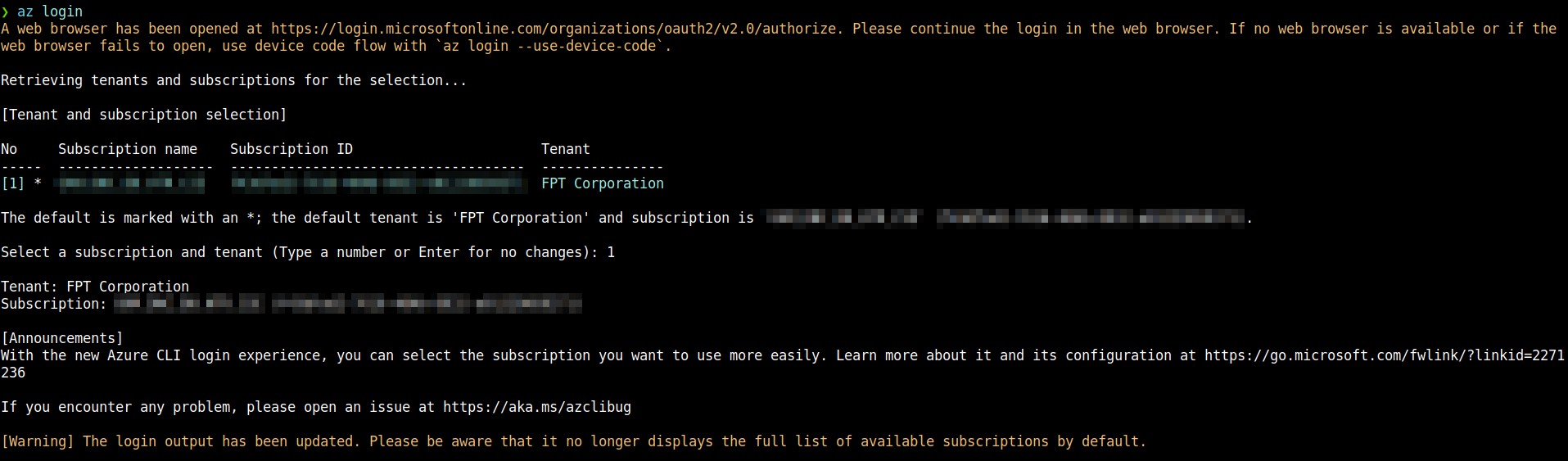
After the new cluster is deployed, go to our cluster and click Connect. Here we have many options to access our cluster like Cloud shell, Azure CLI, and Run command. I will demonstrate the process of using Azure CLI as a demo.



First, we need to install Azure CLI as a prerequisite and make sure that our machine also has kubectl.



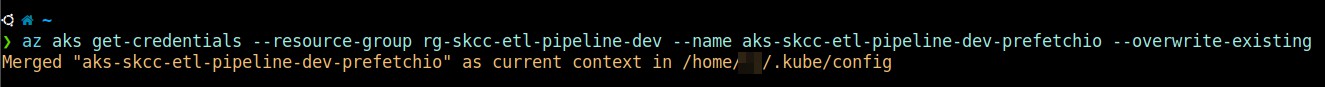
After having Azure CLI on our machine, using 3 commands provided to get started.



az login

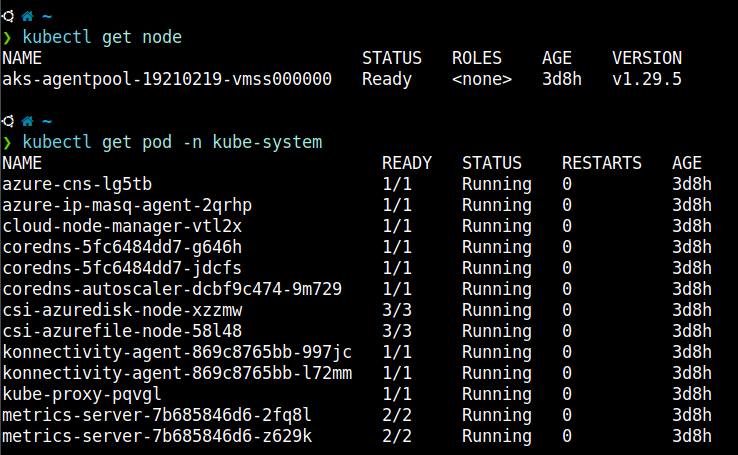
az account set --subscription xxxx





az aks get-credentials --resource-group rg-skcc-etl-pipeline-dev --name aks-skcc-etl-pipeline-d

Thatʼs all! Now we can use kubectl to interact with our cluster. But make sure that the IP of our machine must be in the authentication IP ranges.



In case you have a connection time-out error or something similar, please check your machine IP and add it to authentication ranges to give access rights.

