A

REPORT ON

Project

On

**AKS Cluster**

Submitted in partial fulfilment of the requirements

for the award of the Degree of

Bachelor of Computer Applications

Of

Poornima University

Submitted By:

**Divyank Tiwari**

**Aman Singhal**

**Harendra Singh Khangarot**

**ABHINAV Bhatra**

**Ashish Gurjar**

II Year, BCA

Submitted To:

**Celebal Technologies**

**ACKNOWLEDGEMENT**

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of my project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank **Ms. Anuradha Raheja**, for providing me an opportunity to do the project work and giving us all support and guidance which made me complete the project duly. I am extremely thankful to her for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs.

We are thankful to **Dr. Manoj Gupta, ProPresident, Poornima University** for providing us a platform to carry out this activity successfully.

We are also very grateful to **Mr. Shamneesh Sharma(HOD, BCA Deaprtment)** for his kind support and guidance.

We are thankful and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs which helped us in successfully completing our project work.

Divyank Tiwari

Aman Singhal

Ashish Gurjar

Harendra Singh Khangarot

Abhinav Bhatra

**Abstract**

This project report is basically about how to create AKS Cluster.

Microsoft Azure, commonly referred to as Azure, is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers. It provides software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) and supports many different programming languages, tools, and frameworks, including both Microsoft-specific and third-party software and systems.

Azure Kubernetes Service (AKS) is the quickest way to use Kubernetes on Azure. ... One of the biggest advantage to use AKS is that instead of creating resources in cloud you can create resources and infrastructure inside Azure Kubernetes Cluster through Deployments and Services manifest files.

**TABLE OF CONTENT**

Cover Page

Acknowledgment

Abstract

1.1 What is AKS Cluster……………………………………………………

1.2 AKS features and benefits……………………………………………

1.3 AKS use cases……………………………………………………….

1.4 AKS security, monitoring and compliance……………………………..

1.5 AKS vs. ACS.………………………………………..........................

1.6 AKS availability and costs………………………………………..

1.7 Steps to Create AKS Cluster………………………………………..

1.8 Screen shots of steps………………………………………….…….

1.9 References

***Project -* AKS Cluster**

Azure Kubernetes Service (AKS) is a managed container orchestration service, based on the open source Kubernetes system, which is available on the Microsoft Azure public cloud. An organization can use AKS to deploy, scale and manage Docker containers and container-based applications across a cluster of container hosts.

**AKS features and benefits**

The primary benefits of AKS are flexibility, automation and reduced management overhead for administrators and developers. For example, AKS automatically configures all Kubernetes masters and nodes during the deployment process, and handles a range of other tasks, including Azure Active Directory integration, connections to monitoring services and configuring advanced networking features, such as HTTP application routing.

Since AKS is a managed service, Microsoft handles all Kubernetes upgrades for the service, as new versions become available. Users can decide whether and when to upgrade the Kubernetes version in their own AKS cluster to reduce the possibility of accidental workload disruption.

In addition, AKS nodes can scale up or down to accommodate fluctuations in resource demands. For additional processing power, AKS also supports node pools enabled by graphics processing units (GPUs). This can be vital for compute-intensive workloads, such as scientific applications.

Users can access AKS via an AKS management portal, an AKS command-line interface (CLI), or using templates through tools such as Azure Resource Manager. The service also integrates with the Azure Container Registry (ACR) for Docker image storage, and supports the use of persistent data with Azure Disks.

**AKS use cases**

AKS usage is typically limited to container-based application deployment and management, but there are numerous use cases for the service within that scope.

For example, an organization could use AKS to automate and streamline the migration of applications into containers: first, it could move the application into a container, register the container with ACR, and then use AKS to launch the container into a preconfigured environment. Similarly, AKS can deploy, scale, and manage diverse groups of containers, which helps with the launch and operation of microservices-based applications.

AKS usage can complement agile software development paradigms, such as continuous integration (CI), continuous delivery/continuous deployment (CD) and DevOps. For example, a developer could place a new container build into a repository, such as GitHub, move those builds into ACR, and then rely on AKS to launch the workload into operational containers.

Other uses for AKS involve the Internet of Things (IoT). The service, for instance, could help ensure adequate compute resources to process data from thousands, or even millions, of discrete IoT devices. Similarly, AKS can help ensure adequate compute for big data tasks, such as model training in machine learning (ML) environments.

In general, AKS is most frequently used by software developers and IT operations staff.

**AKS security, monitoring and compliance**

AKS supports role based access control (RBAC) through Azure Active Directory (AD), which enables an administrator to tailor Kubernetes access to AD identity and group associations. Admins can monitor container health using processor and memory metrics collected from containers, Kubernetes nodes and other points in the infrastructure. Container logs are also collected and stored for more detailed analytics and troubleshooting. Monitoring data is available through the AKS management portal, AKS CLI and application programming interfaces (APIs).

AKS meets the regulatory requirements of System and Organization Controls (SOC) and is compliant with major regulatory bodies including ISO, HIPAA and HITRUST. Finally, AKS is certified as Kubernetes conformant by the Cloud Native Computing Foundation (CNCF), which oversees open source Kubernetes.

**AKS vs. ACS**

Prior to the release of AKS, Microsoft offered Azure Container Service (ACS), which supported numerous open source container orchestration platforms, including Docker Swarm and Mesosphere's DC/OS, as well as Kuberenetes. With AKS, the focus is exclusively on the use of Kubernetes. ACS users with a focus on Kubernetes can potentially migrate from ACS to AKS.

However, AKS poses numerous differences that a user must address before a migration from ACS. For example, AKS uses managed disks, so a user must convert unmanaged disks to managed disks before assigning them to AKS nodes. Similarly, a user must convert any persistent storage volumes or customized storage class objects associated with Azure disks to managed disks.

In addition, stateful applications can be impacted by downtime and data loss during a migration from ACS to AKS, so developers and application owners should perform detailed due diligence before making the move.

**AKS availability and costs**

AKS is a free Azure service, so there is no charge for Kubernetes cluster management. AKS users are, however, billed for the underlying compute, storage, networking and other cloud resources consumed by the containers that comprise the application running within the Kubernetes cluster.

**Steps to Create & Deploy**

1. On the Azure portal menu or from the Home page, select Create a resource.
2. Select Containers > Kubernetes Service.

3. On the Basics page, configure the following options:

* Project details: Select an Azure Subscription, then select or create an Azure Resource group, such as myResourceGroup.
* Cluster details: Enter a Kubernetes cluster name, such as myAKSCluster. Select a Region, Kubernetes version, and DNS name prefix for the AKS cluster.
* Primary node pool: Select a VM Node size for the AKS nodes. The VM size can't be changed once an AKS cluster has been deployed. - Select the number of nodes to deploy into the cluster. For this quickstart, set Node count to 1. Node count can be adjusted after the cluster has been deployed.

4. On the Scale page, keep the default options. At the bottom of the screen, click Next: Authentication

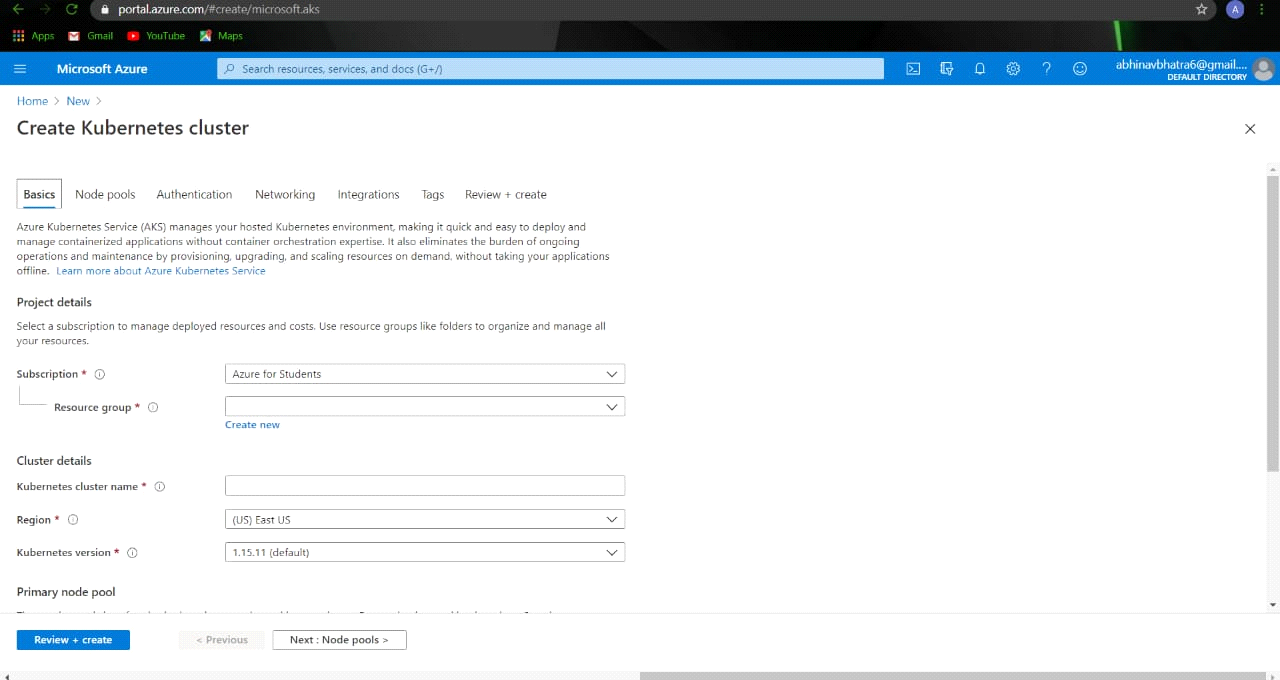
5. On the Authentication page, configure the following options

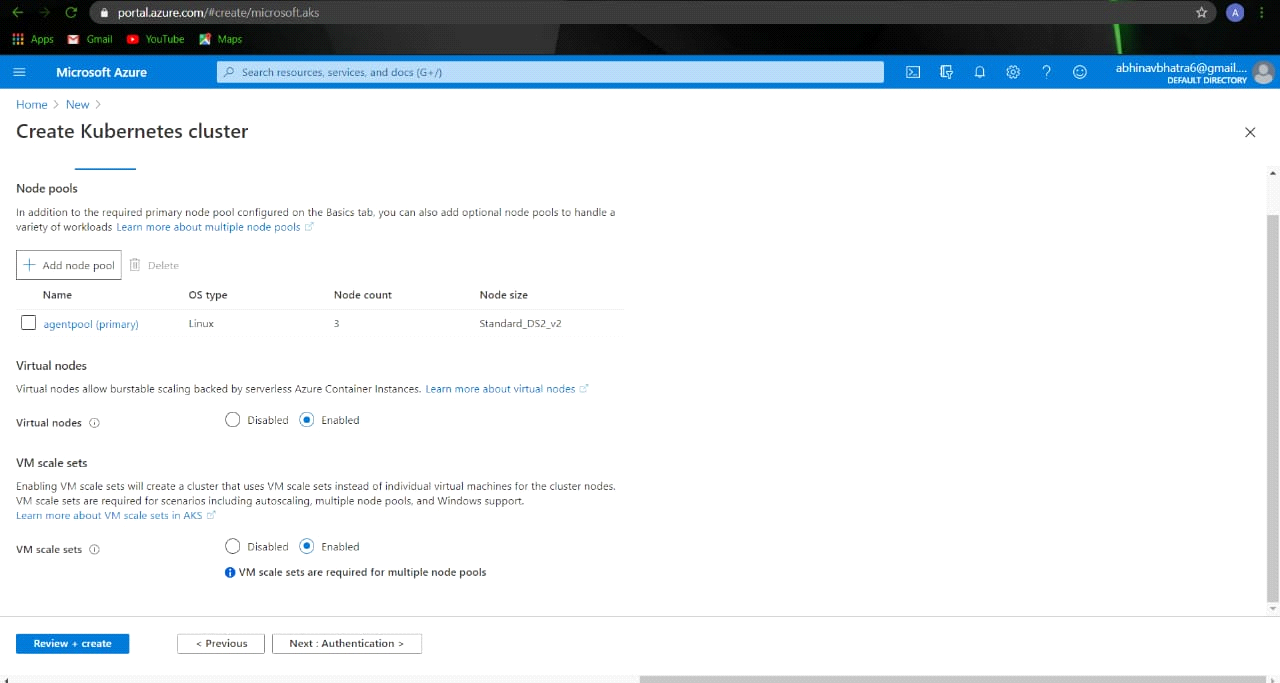
* Create a new service principal by leaving the Service Principal field with (new) default service principal. Or you can choose Configure service principal to use an existing one. If you use an existing one, you will need to provide the SPN client ID and secret.
* Enable the option for Kubernetes role-based access controls (RBAC). This will provide more fine-grained control over access to the Kubernetes resources deployed in your AKS cluster.

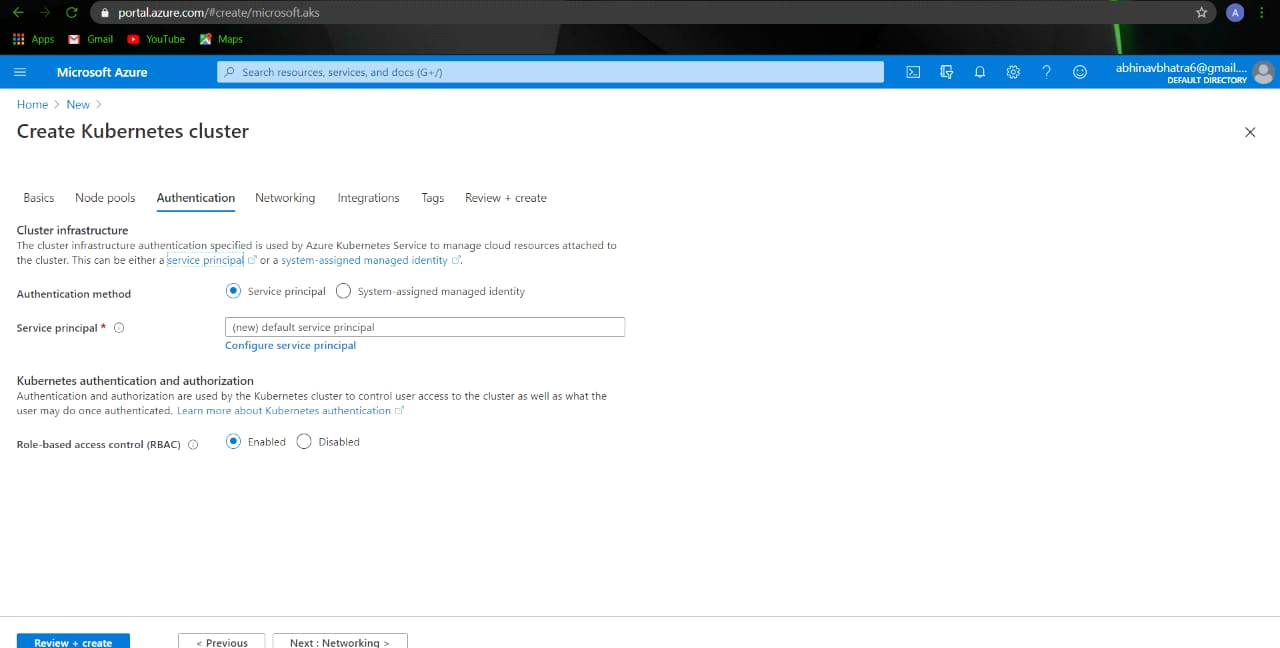
By default, Basic networking is used, and Azure Monitor for containers is enabled. Click Review + create and then Create when validation completes.

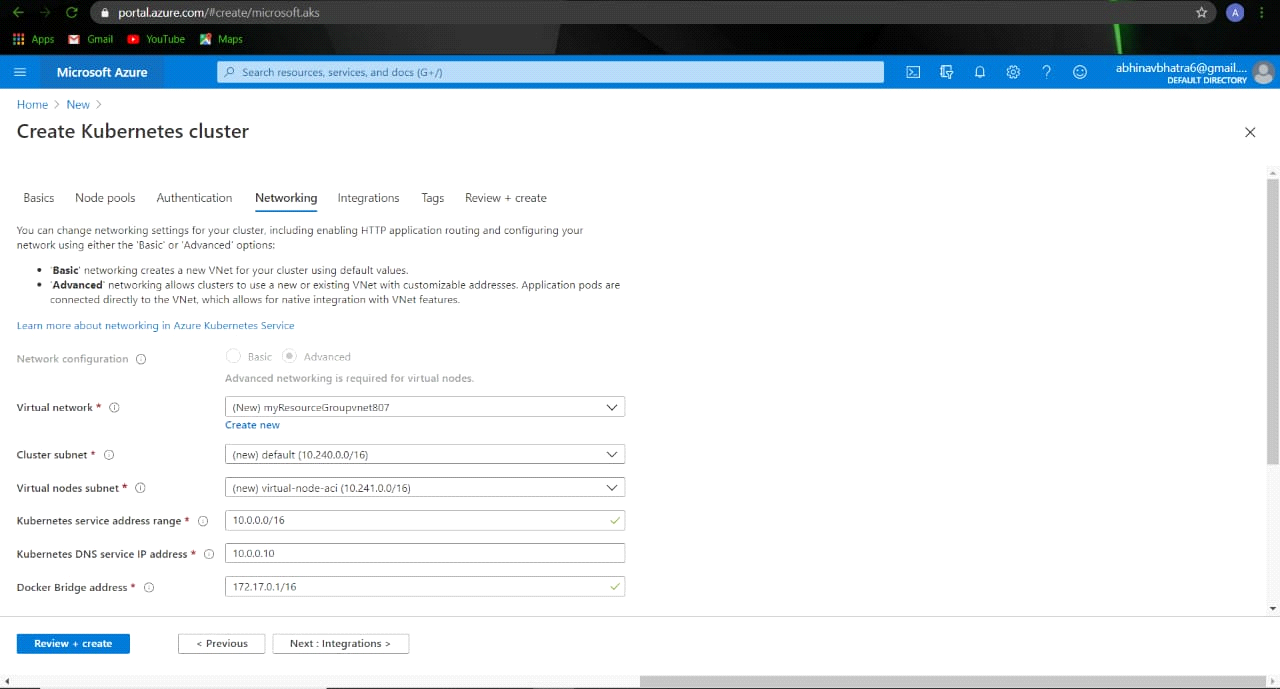
It takes a few minutes to create the AKS cluster. When your deployment is complete, click Go to resource, or browse to the AKS cluster resource group, such as myResourceGroup, and select the AKS resource, such as myAKSCluster.

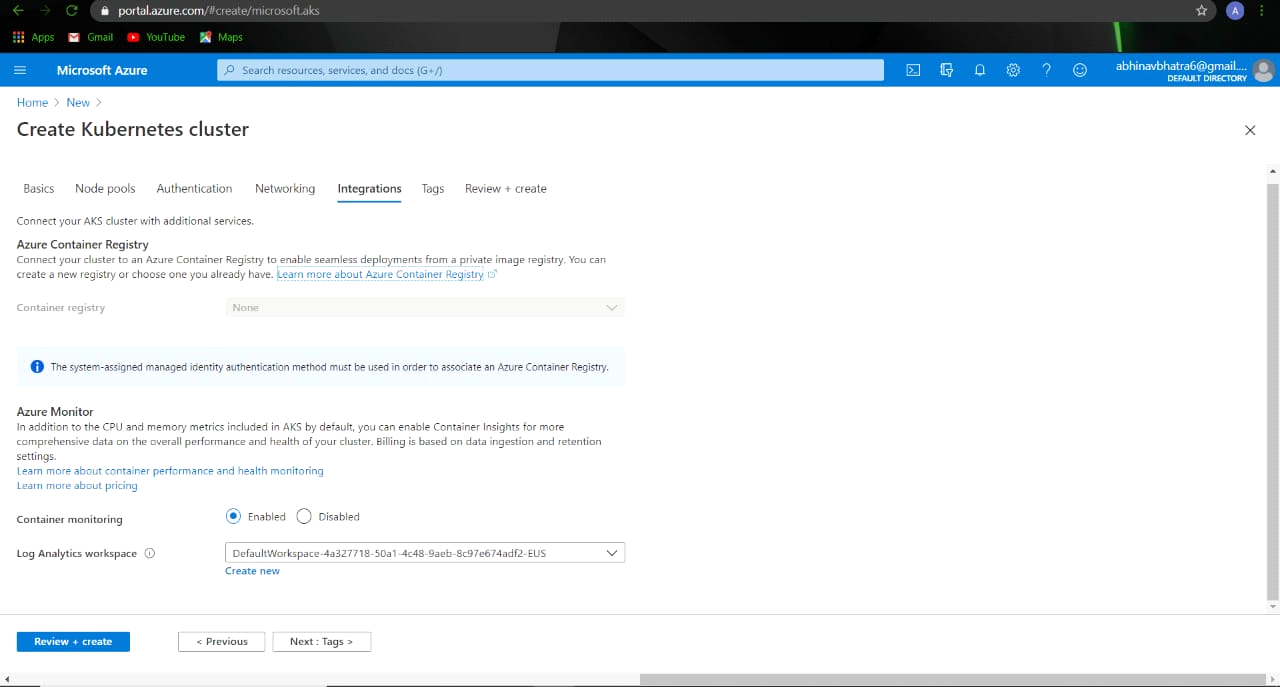
**SCREENSHOTS OF STEPS:**

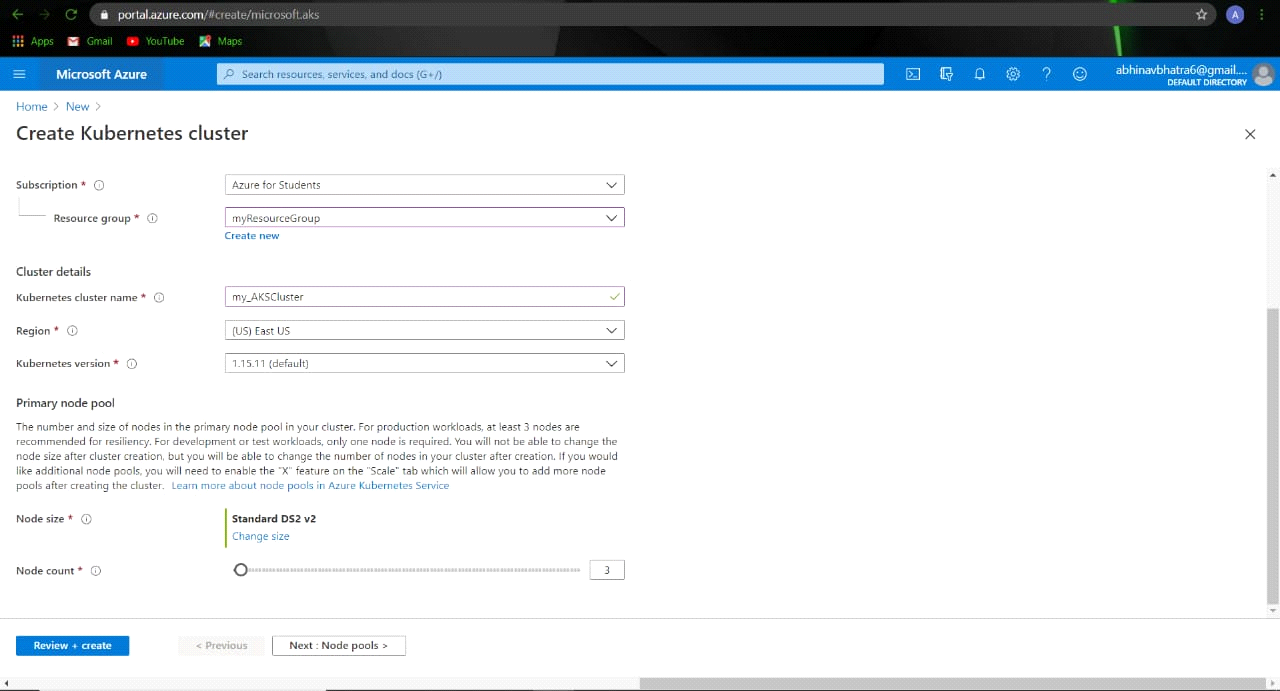


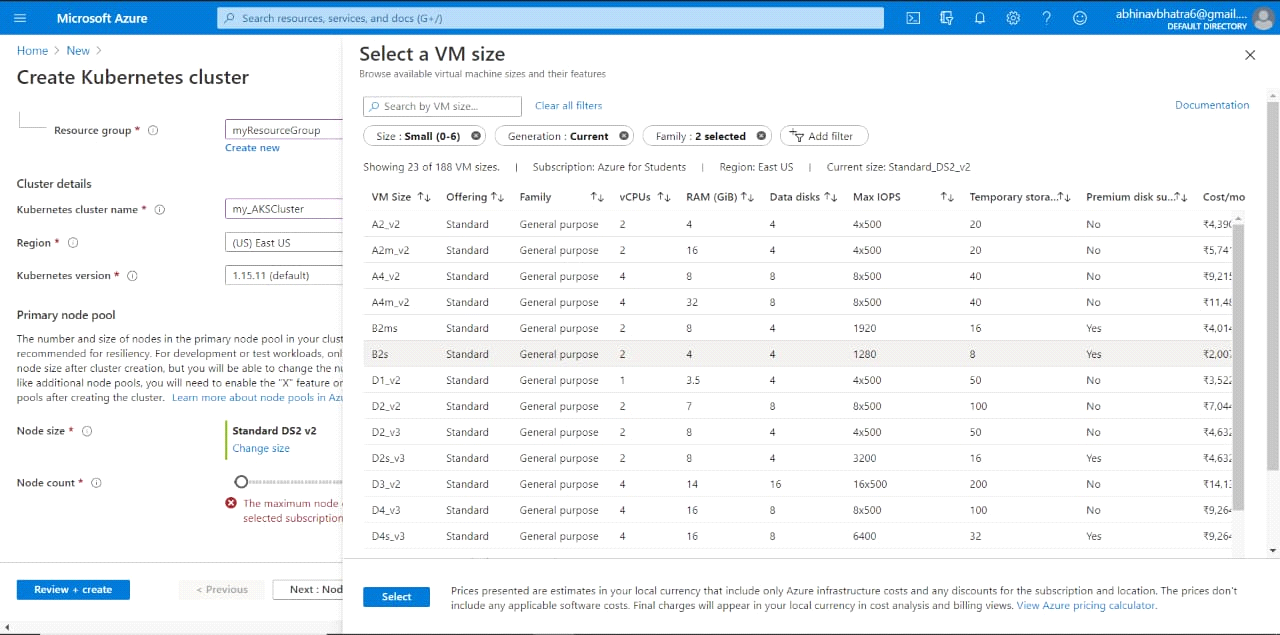


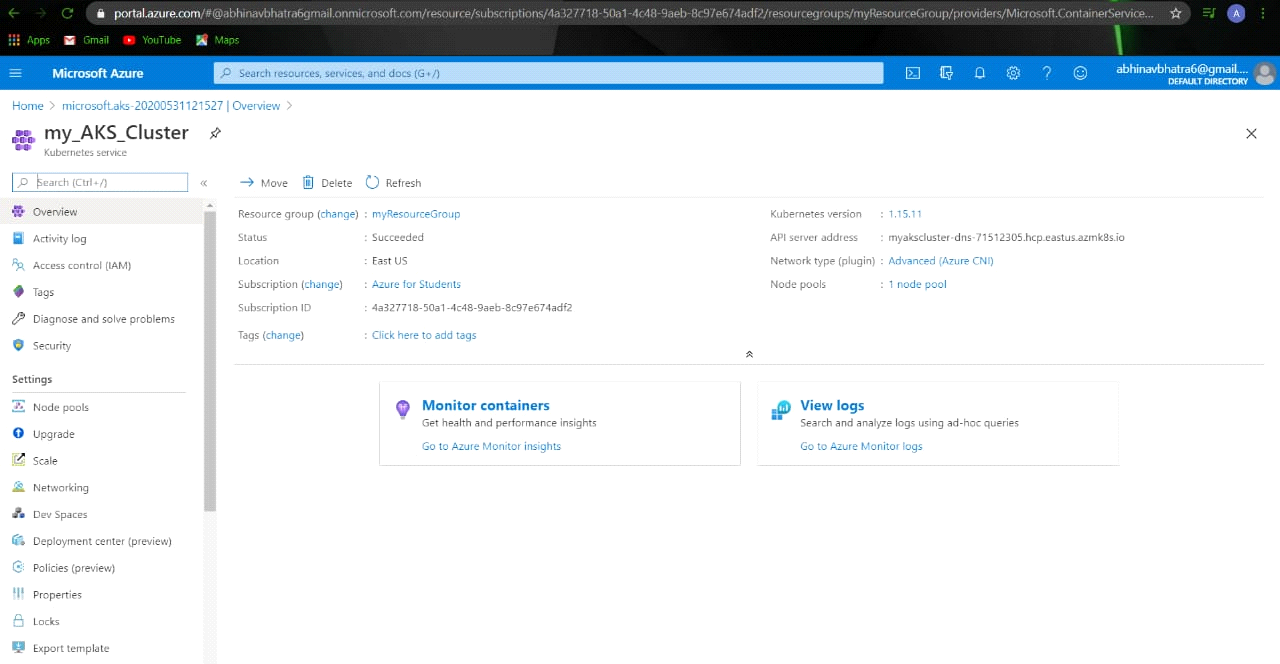


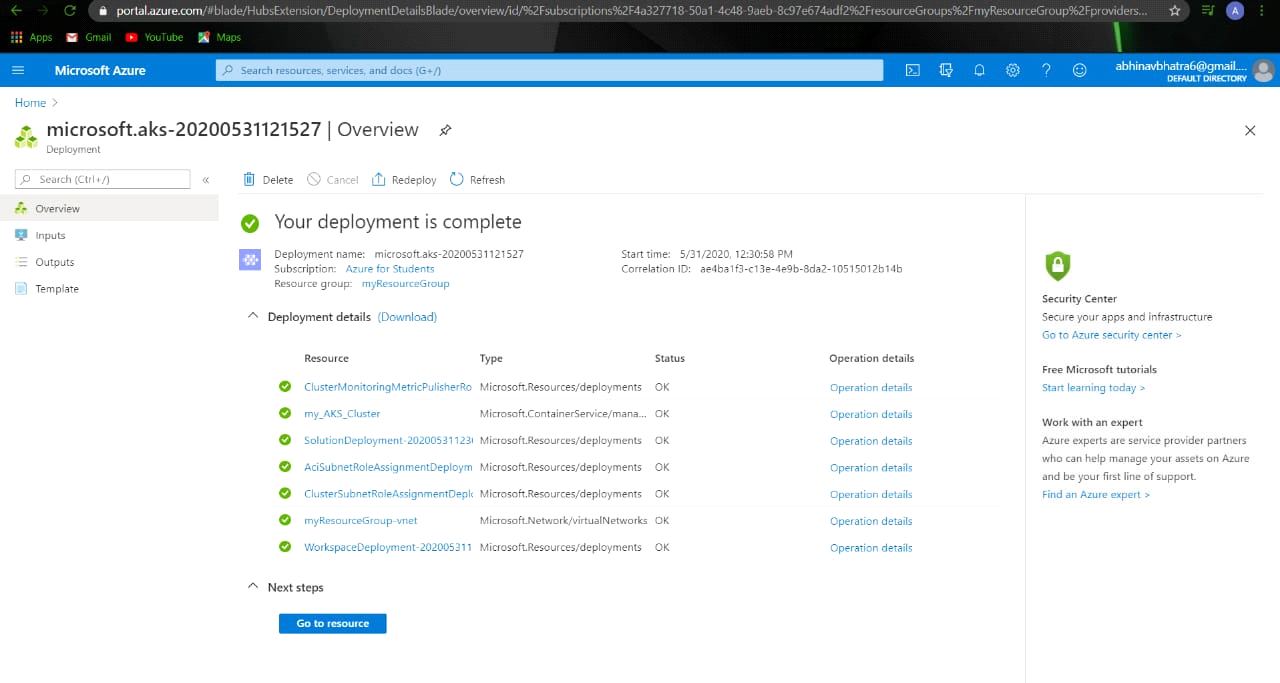










**REFERENCES:**

* docs.microsoft.com
* microsoft.github.io
* azure.microsoft.com