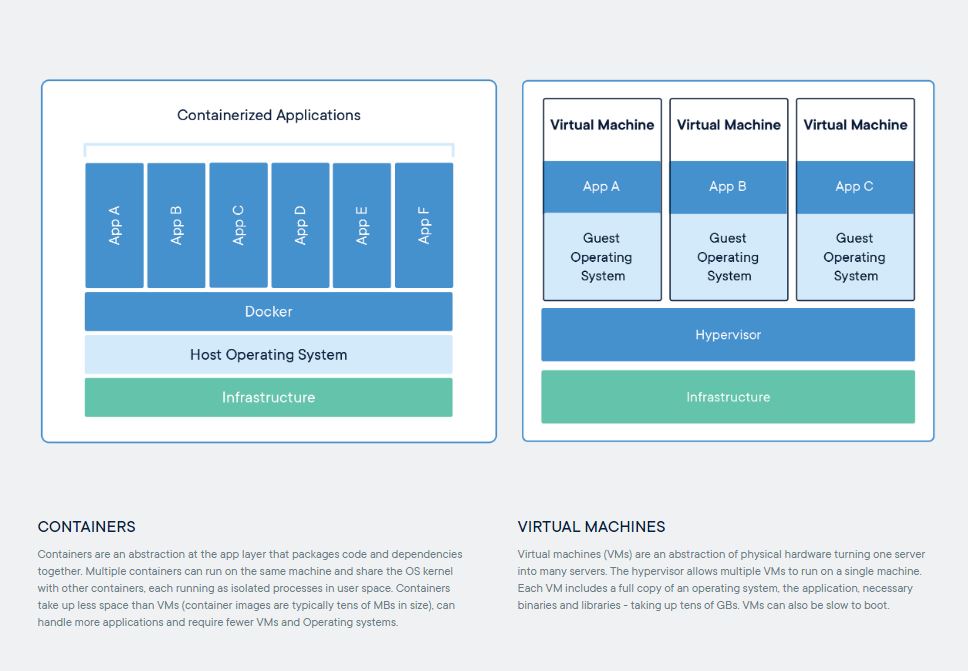
Goals

1. Learning Docker
2. Learning Kubernetes
3. Learning Prometheus
4. Learning Grafana
5. Learning Azure Kubernetes Service (AKS)
6. **Learning Docker**

**Definition :** *Docker is an*[*open source*](http://searchenterpriselinux.techtarget.com/definition/open-source)*software platform to create, deploy and manage virtualized application containers on a common operating system, with an ecosystem of allied tools. Docker Inc., the company that originally developed Docker, supports a commercial edition and is the principal sponsor of the open source tool.*



*Docker Vs Virtual Machines*

**Benefits of Docker:**

* **Containers are small compared to VMs**
* **Containers uses less resources**
* **Fast boot**
* Deployment with quick and easy rollbacks (due to image immutability).
* **Eliminating the “Works on My Machine” situation**
* **They work well in DevOps and CI/CD**
* Return on Investment and Cost Savings: Dockers first advantage is ROI. Especially for large, established companies, which need to generate steady revenue over the long term, the solution is only better if it can drive down costs while raising profits.

**Important Concepts:**

1. **Dockerfile:**

Docker can build images automatically by reading the instructions from a Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build users can create an automated build that executes several command-line instructions in succession.

1. **Docker-Compose:**

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application’s services. Then, with a single command, you create and start all the services from your configuration.

Using Compose is basically a three-step process:

1. Define your app’s environment with a Dockerfile so it can be reproduced anywhere.
2. Define the services that make up your app in docker-compose.yml so they can be run together in an isolated environment.
3. Run docker-compose up and Compose starts and runs your entire app.
4. **Learning Kubernetes**

[Kubernetes (K8s)](https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/) is an open-source system for automating deployment, scaling, and management of containerized applications.

In a production environment, you need to manage the containers that run the applications and ensure that there is no downtime. For example, if a container goes down, another container needs to start. Wouldn’t it be easier if this behavior was handled by a system?

That’s how Kubernetes comes to the rescue! Kubernetes provides you with a framework to run distributed systems resiliently. It takes care of your scaling requirements, failover, deployment patterns, and more. For example, Kubernetes can easily manage a canary deployment for your system.

**Benefits of Kubernetes:**

* + Better management of apps through modularity, allows decomposing apps into smaller parts
  + Eliminates infrastructure lock-ins, offers core capabilities without imposing too many restrictions
  + Enables businesses to deploy and update software at scale across Pods
  + Better version control, rolling updates and canary deployments
  + Auto horizontal scaling, autoscalers size number of Pods automatically
  + Large open-source community backing the project helps keep pace with today’s requirements
  + High-velocity updates without the downtimes associated with such updates
  + Immutable infrastructure, users can build and easily deploy new container images, rollback if needed without much hassle
  + Declarative configuration, far less error prone
  + Self-healing, brings replicas back if destroyed
  + More efficient, tasks can be packed onto fewer machines
  + Separation of concerns, small teams can achieve big results

**Important Concepts:**

1. **Manifest:** When you are going to deploy to kubernetes, or create a kubernetes resources like a pod, replica-set, configmap, secret, deployment, etc, you need to write a file called manifest that describe that object and its attributes either in yaml or json. It's basically a Kubernetes "API object description". A config file can include one or more of these.
2. **Helm Charts: Helm is the package manager** (analogous to yum and apt) and **Charts are packages** (analogous to debs and rpms). The home for these Charts is the [Kubernetes Charts repository](https://github.com/kubernetes/charts) which provides continuous integration for pull requests, as well as automated releases of Charts in the master branch. It basically makes it simple to package and deploy common applications on Kubernetes.
3. **Learning Prometheus:**

[Prometheus](https://github.com/prometheus) is an open-source system monitoring and alerting toolkit originally built at [SoundCloud](https://soundcloud.com/).

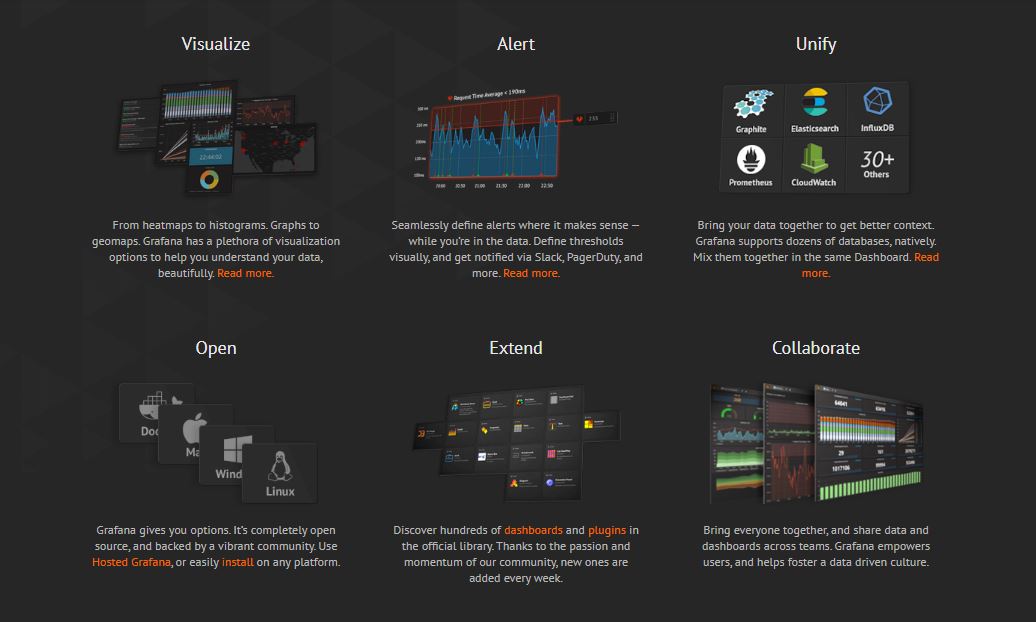
**Benefits of Prometheus:**

* a **multi-dimensional** data model (timeseries defined by metric name and set of key/value dimensions)
* a **flexible query language** to leverage this dimensionality
* no dependency on distributed storage; **single server nodes are autonomous**
* timeseries collection happens via a **pull model** over HTTP
* **pushing timeseries** is supported via an intermediary gateway
* targets are discovered via **service discovery** or **static configuration**
* multiple modes of **graphing and dashboarding support**
* support for hierarchical and horizontal **federation**

1. **Learning Grafana**

Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored.

**Benefits of Grafana:**



Benefits of Grafana

1. **Learning Azure Kubernetes Service (AKS)**

Azure Kubernetes Service (AKS) is a managed container orchestration service, based on the open source [Kubernetes system](https://searchitoperations.techtarget.com/definition/Google-Kubernetes), which is available on the Microsoft Azure public cloud. An organization can use AKS to deploy, scale and manage Docker [containers](https://searchitoperations.techtarget.com/definition/container-containerization-or-container-based-virtualization) and container-based applications across a cluster of container hosts.

**Benefits of AKS:**

## **1. ACCELERATED APP DEVELOPMENT**

[75 percent](https://coralogix.com/log-analytics-blog/this-is-what-your-developers-are-doing-75-of-the-time-and-this-is-the-cost-you-pay/) of developer’s time is typically spent on bug-fixing. AKS removes much of the time-sink (and headache) of debugging by handling the following aspects of your development infrastructure:

* Auto upgrades
* Patching
* Self-healing

Through AKS, **container orchestration is simplified**, saving you time and enabling your developers to remain more productive. It’s a way to breathe life into your application development by combatting one of developer’s biggest time-sinks.

## **2. SUPPORTS AGILE PROJECT MANAGEMENT**

As [this PWC report](https://www.pwc.com/gx/en/actuarial-insurance-services/assets/agile-project-delivery-confidence.pdf) shows, agile projects yield strong results and are typically 28 percent more successful than traditional projects.

This is another key benefit to AKS – it **supports agile development**programs, such as continuous integration (CI), continuous delivery/continuous deployment (CD) and dev-ops. This is done through integration with Azure DevOps, ACR, Azure Active Directory and Monitoring. An example of this is a developer who puts a container into a repository, moves the builds into Azure Container Registry (ACR), and then uses AKS to launch the workload.

## **3. SECURITY AND COMPLIANCE DONE RIGHT**

Cyber security must be a priority for all businesses moving forward. Last year, almost half of UK businesses suffered a cyber-attack and, according to [IBM’s study](https://www-05.ibm.com/services/europe/digital-whitepaper/security/growing_threats.html), 60 percent of data breaches were caused by insiders. The threat is large, and it often comes from within.

AKS **protects your business** by enabling administrators to tailor access to Azure Active Directory (AD) and identity and group identities. When people only have the access they need, the threat from internal teams is greatly reduced.

You can also rest assured that AKS is totally compliant. AKS meets the regulatory requirements of System and Organization Controls (SOC), as well as being compliant with **ISO, HIPAA and HITRUST.**

## **4. USE ONLY THE RESOURCES YOU NEED**

AKS is a **fully flexible**system that adapts to use only the resources you need. Additional processing power is supported via graphic processing units (GPUs) – processor intensive operations, such as scientific computations, enables on-top processing power. If you need more resources, it’s as simple as clicking a button and letting the elasticity of **Azure container instances**do the rest.

When you only use the resources you need, your software (and your business) enjoys the following benefits:

* **Reduced cost**– no extra GPUs need to be bought and integrated onsite.
* **Faster start-up speed** compared to onsite hardware and software which takes time to set-up.
* **Easier scaling**– get more done now without worrying about how to manage resources.

## **SCALE AT SPEED WITH AKS**

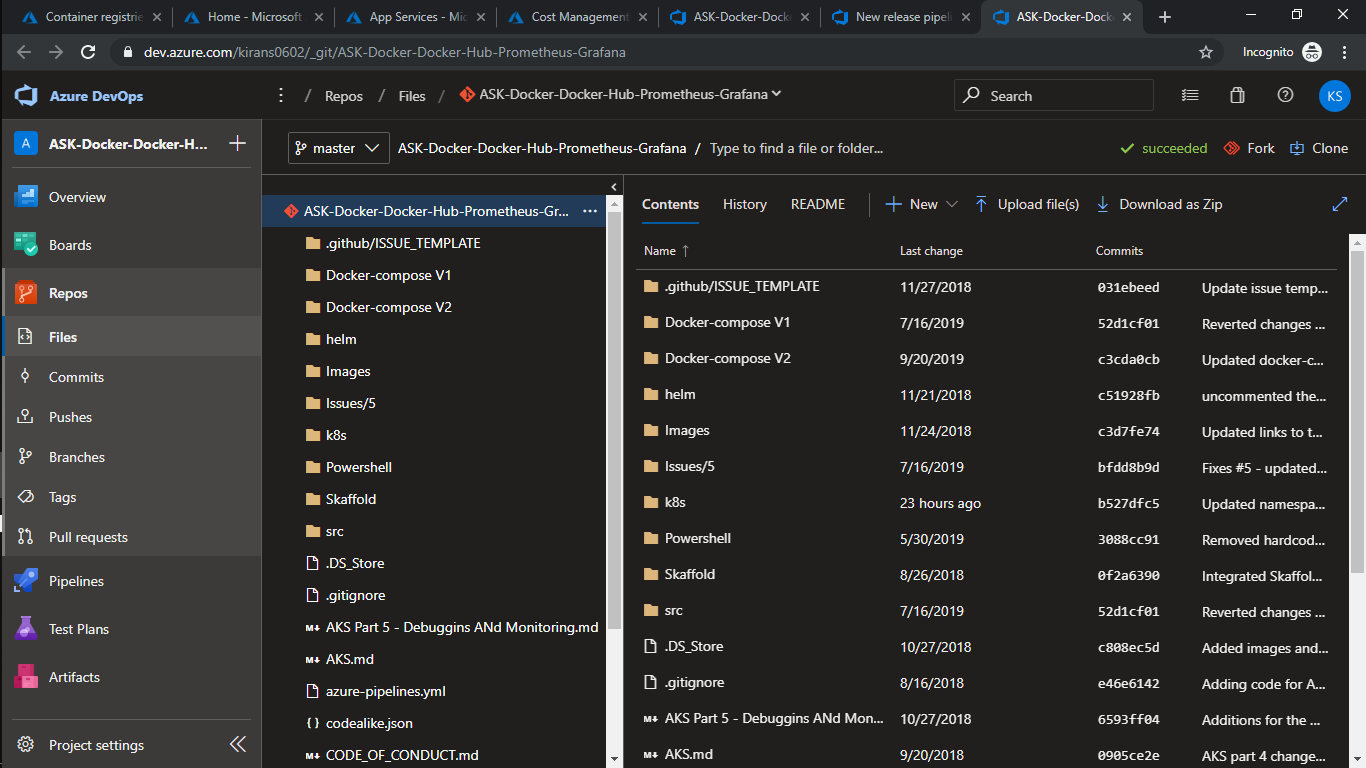
The world of applications moves fast. For example, [6140 Android apps](https://www.statista.com/statistics/276703/android-app-releases-worldwide/) were released in the first quarter of 2018 alone. Ambitious companies can’t afford the risk of slowing down. Free up time and simplify the application of containerization by implementing AKS and take your software development to the next level.

**Tools Used**

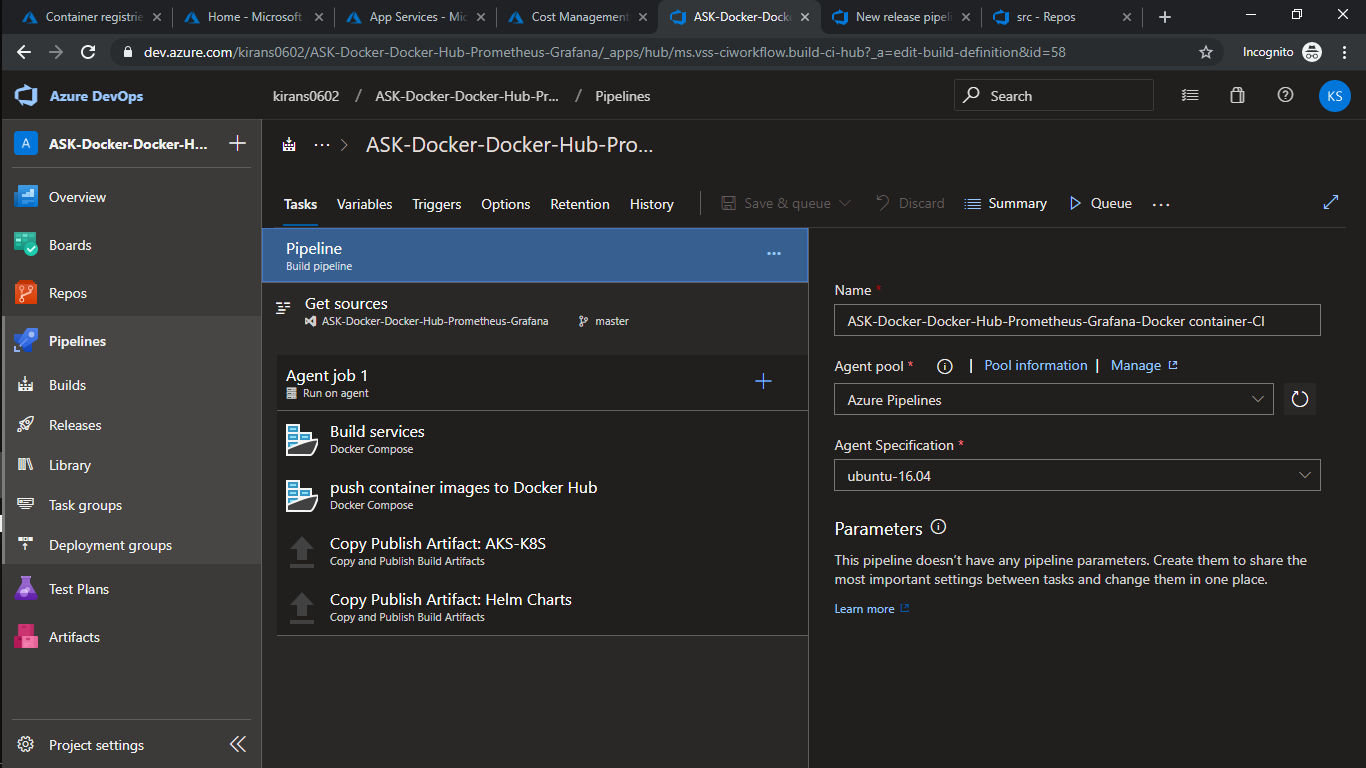
1. Git Hub
2. Docker
3. Docker Hub
4. AKS (Azure Kubernetes Service)
5. Helm Charts
6. Prometheus
7. Grafana
8. Azure Monitoring Service (Insight)
9. Azure Container Registry
10. Azure DevOps Services

**Project Steps**

1. **Creating CI/CD pipeline**

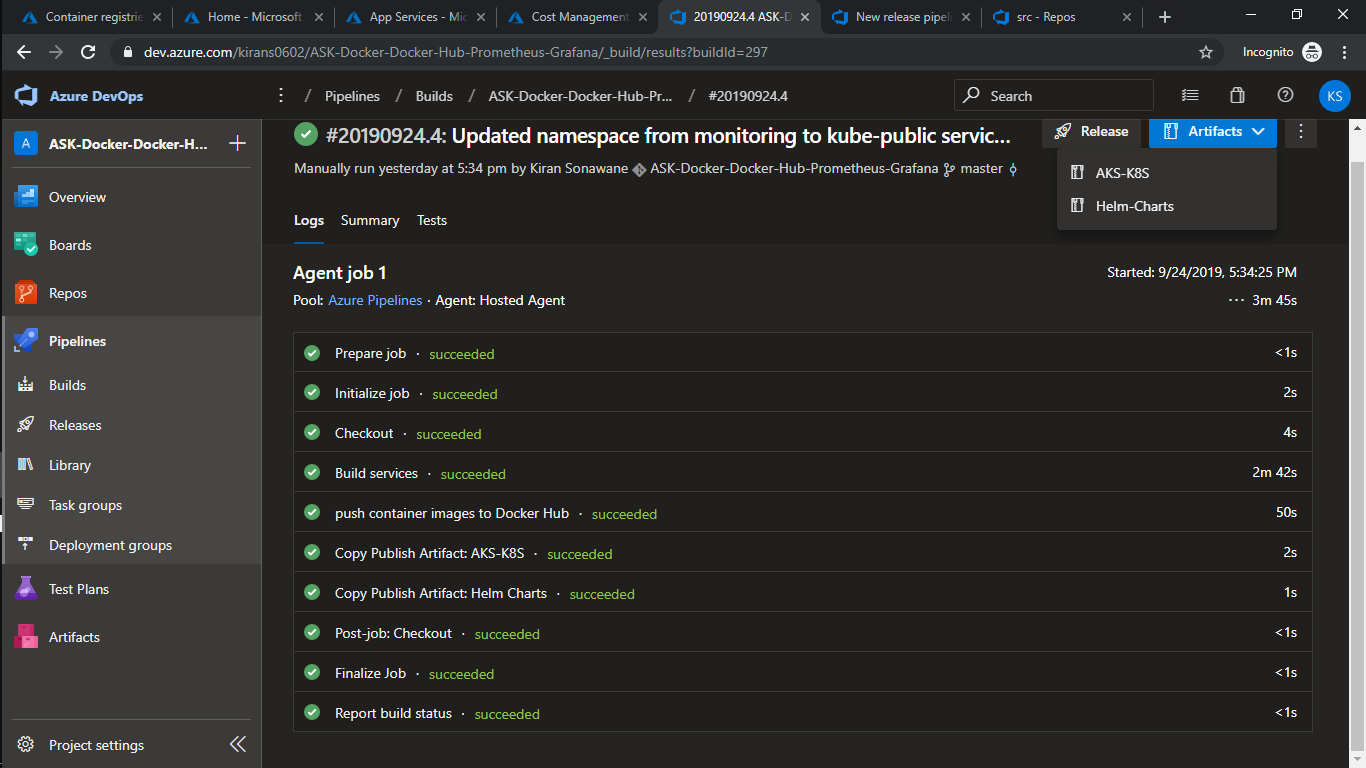


* 1. **Create Build Pipeline**

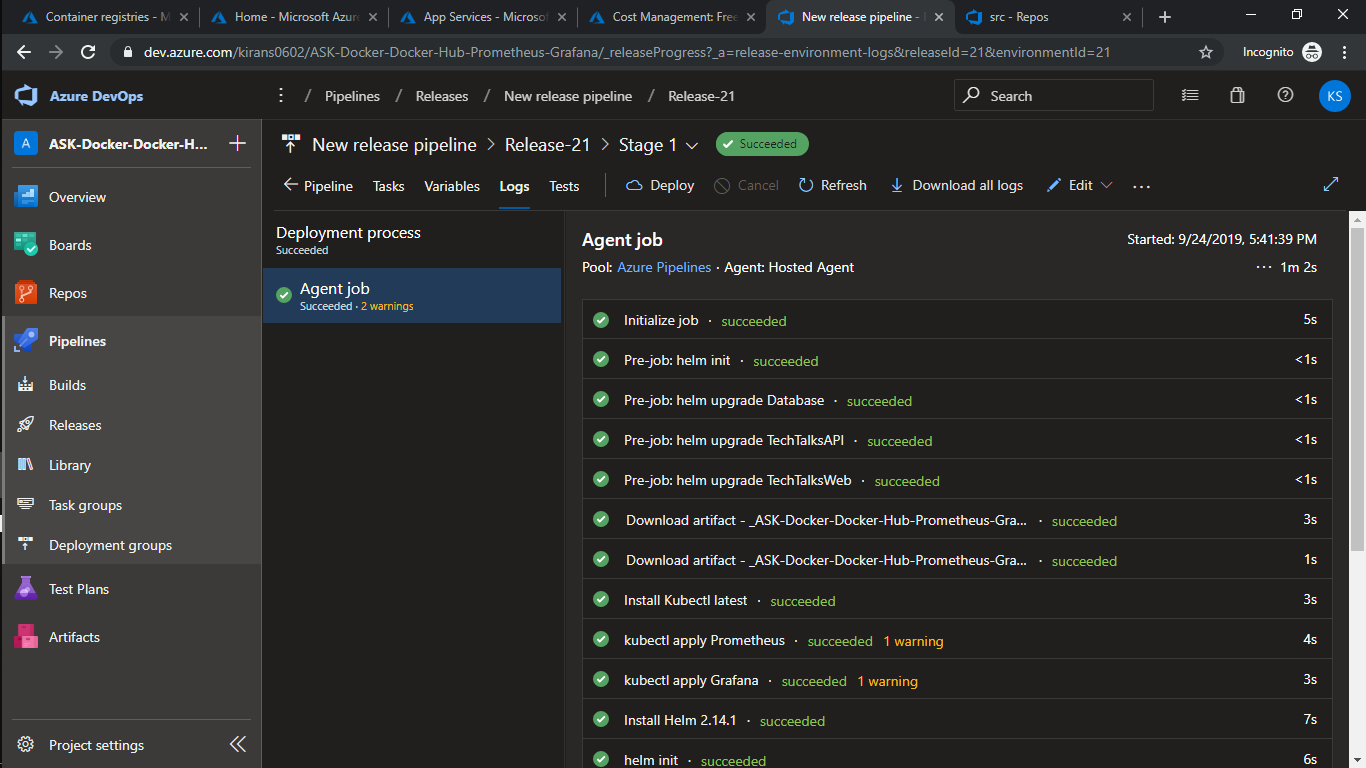


Here we create docker compose file which will help us to upload our docker images on Docker Hub or Azure Container registry. Later we will use these images to update with our techtalks app.

Here we also copied our AKS Files which contains Kubernetes manifest which help us to deploy our monitoring service images on AKS. Also, we copied Helm Charts which will help us to set up an .net environment in Pods to create and maintain running replicas of our app.

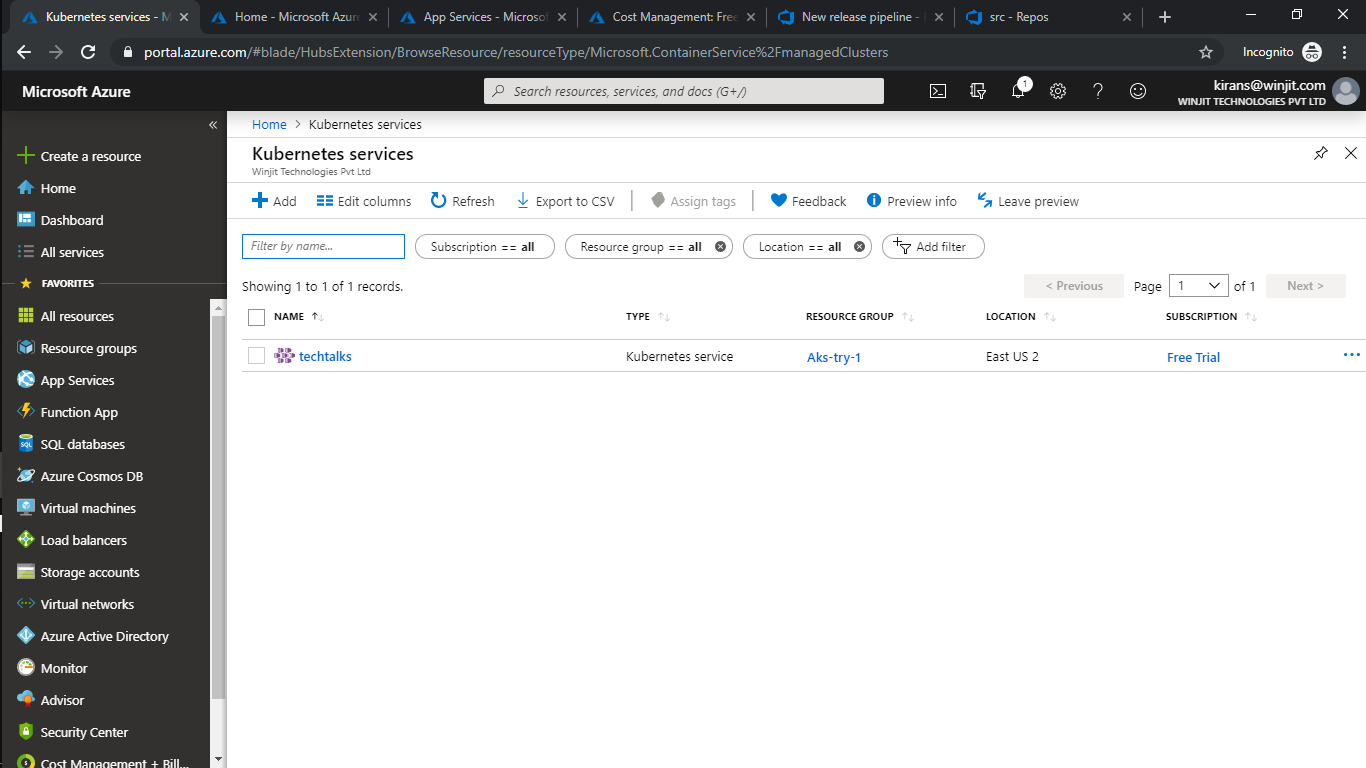


* 1. **Release Pipeline**



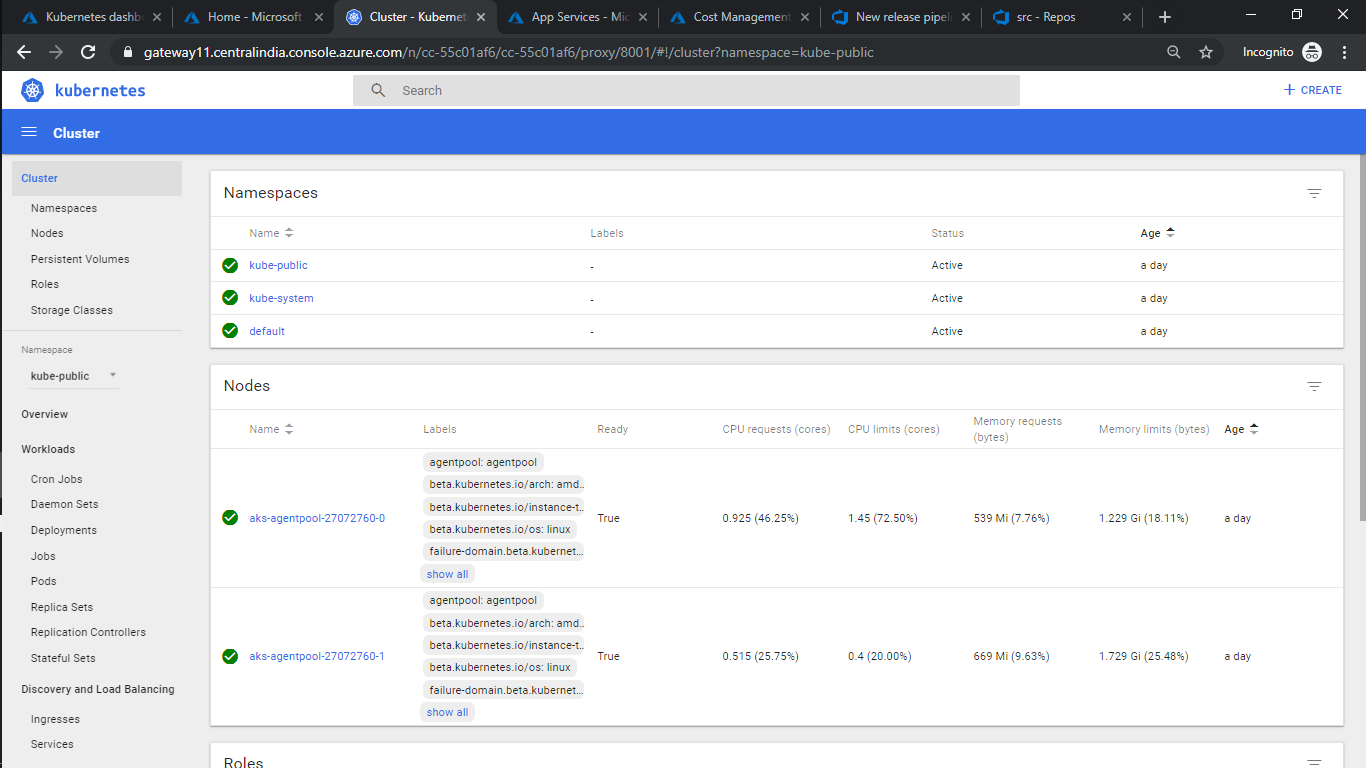
In release pipeline we include kubectl to deploy an Prometheus and Grafana Images on AKS cluster where they will help to monitor our Web App replicas. To deploy Web App on AKS cluster we’ll use helm charts which will update only when there is difference in our previous image and current image also it’ll keep our dependencies and other packages updated.

1. Azure

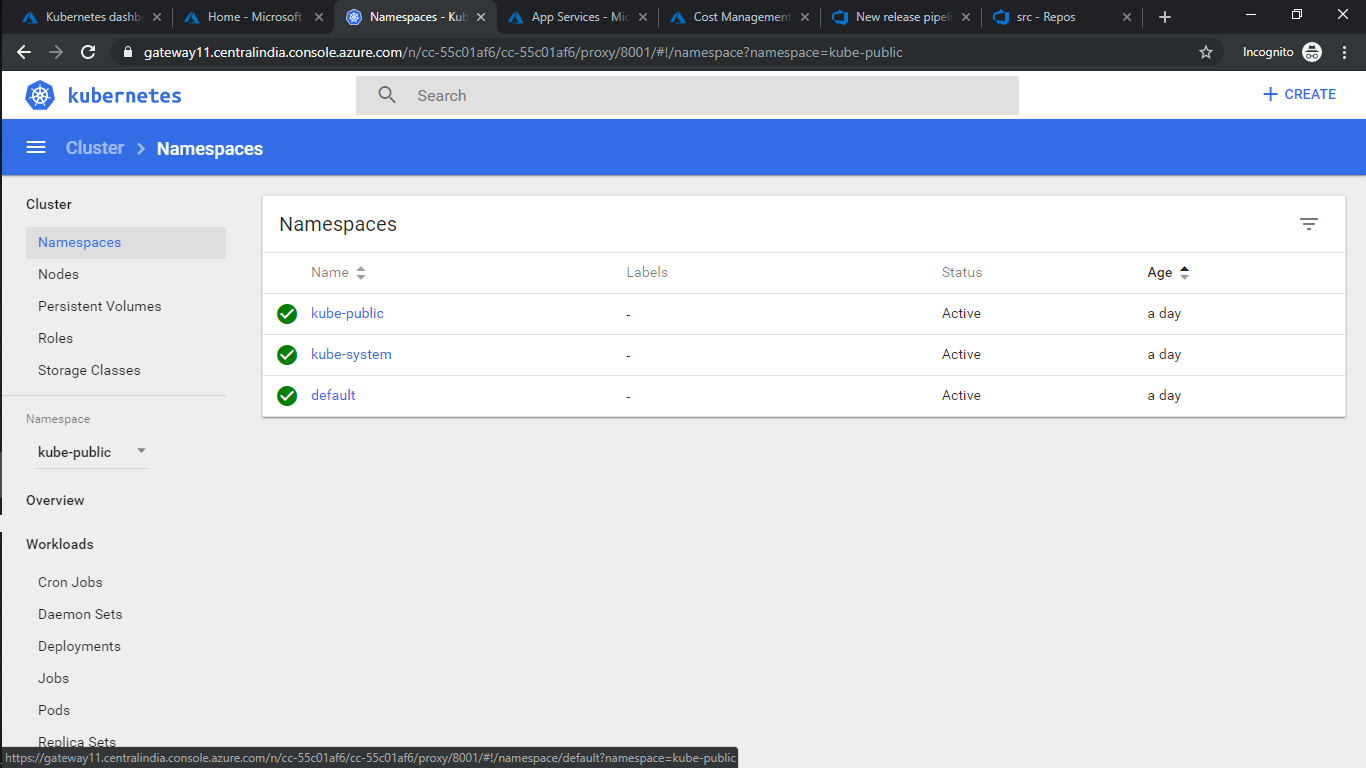


Create AKS cluster techtalks (2 nodes with of 4 cores and 14GB memory).

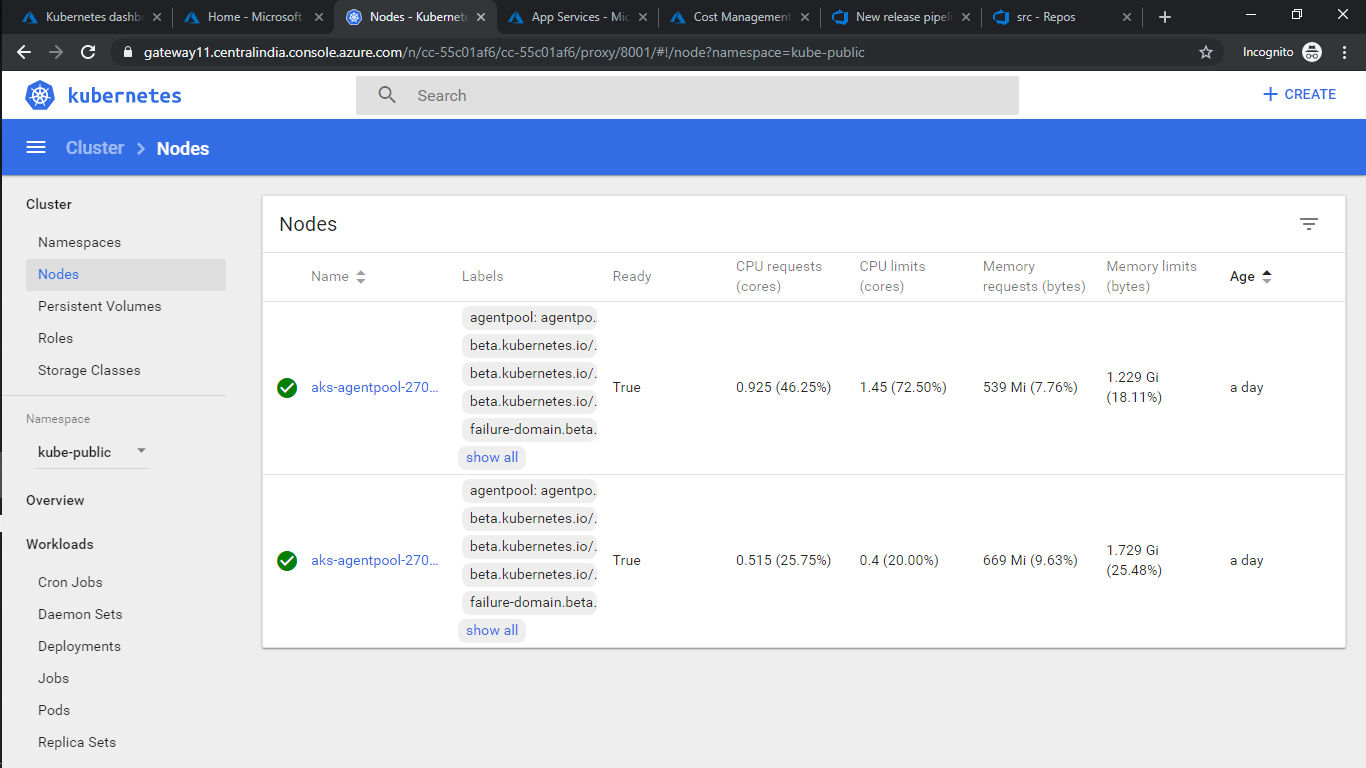
* 1. **Open AKS dashboard**



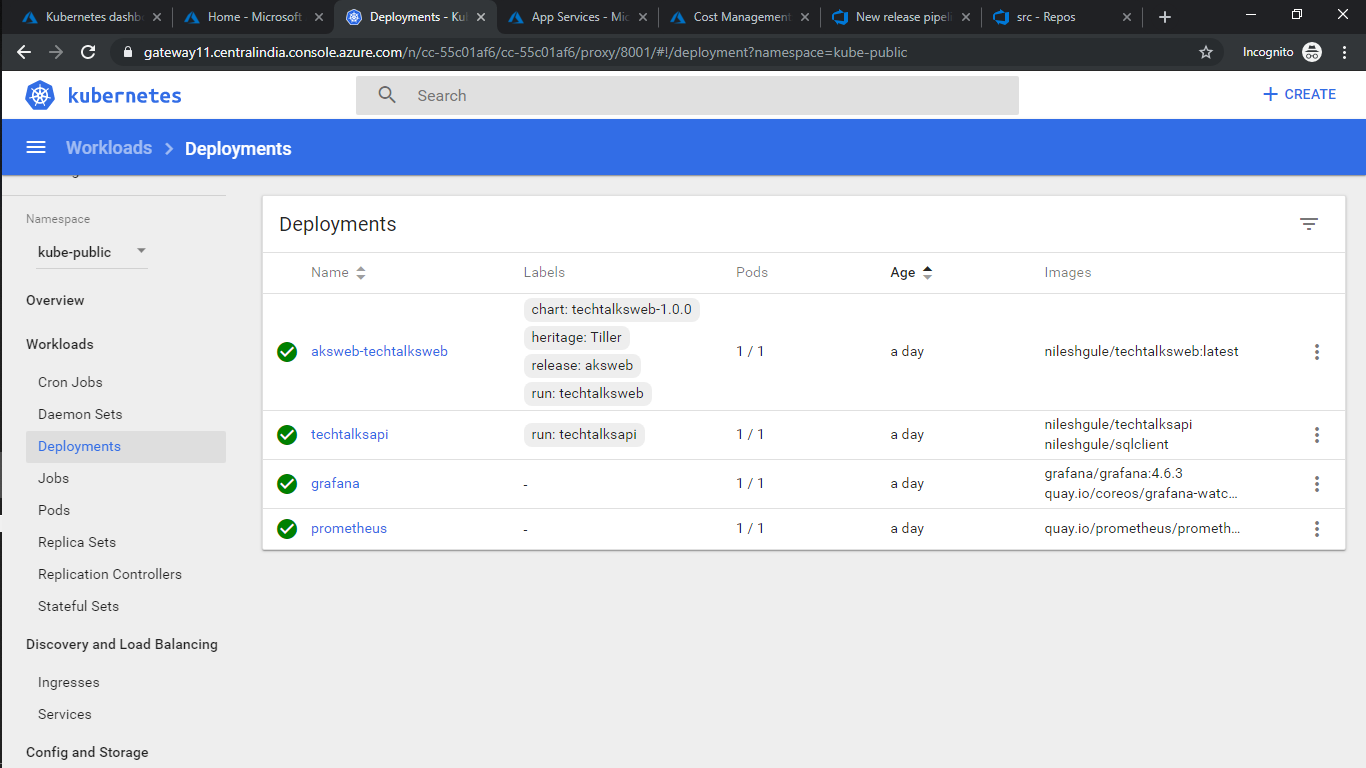
AKS Dashboard



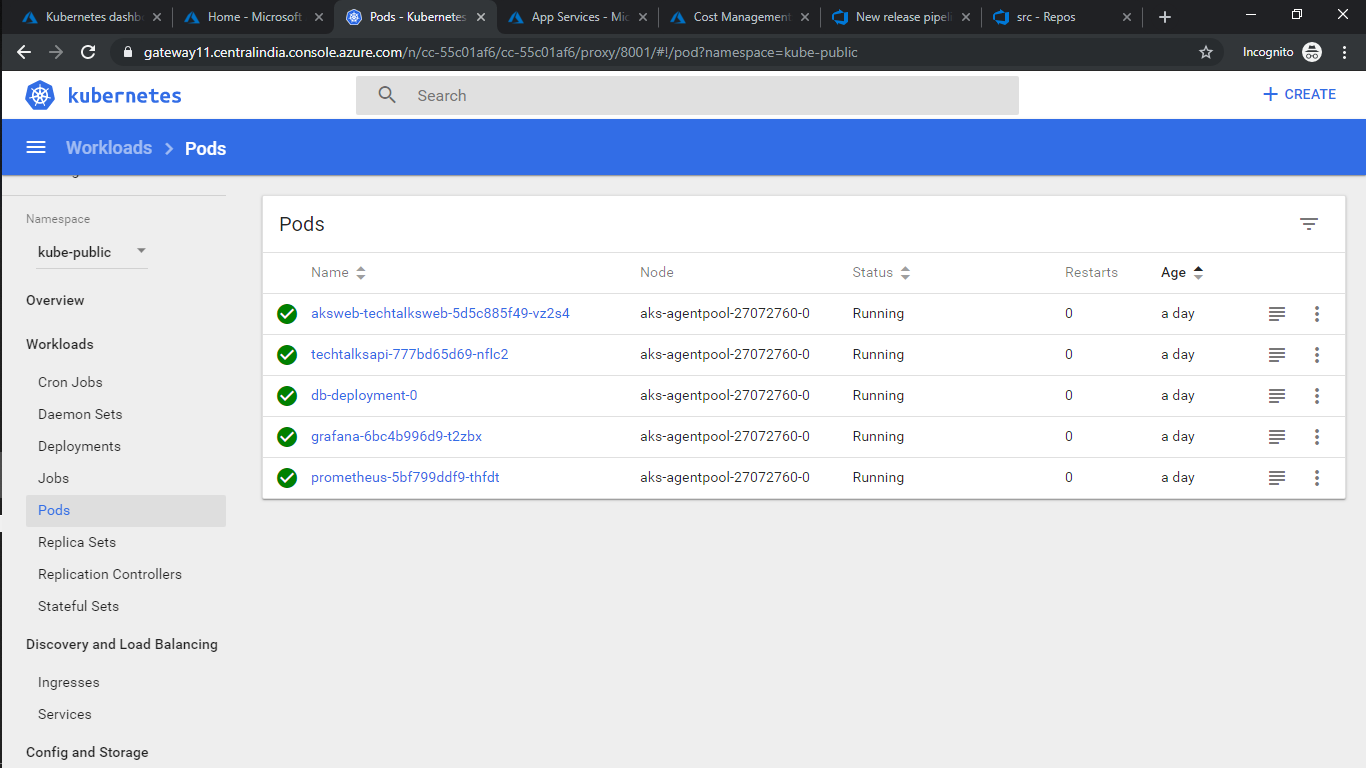
AKS Namespace



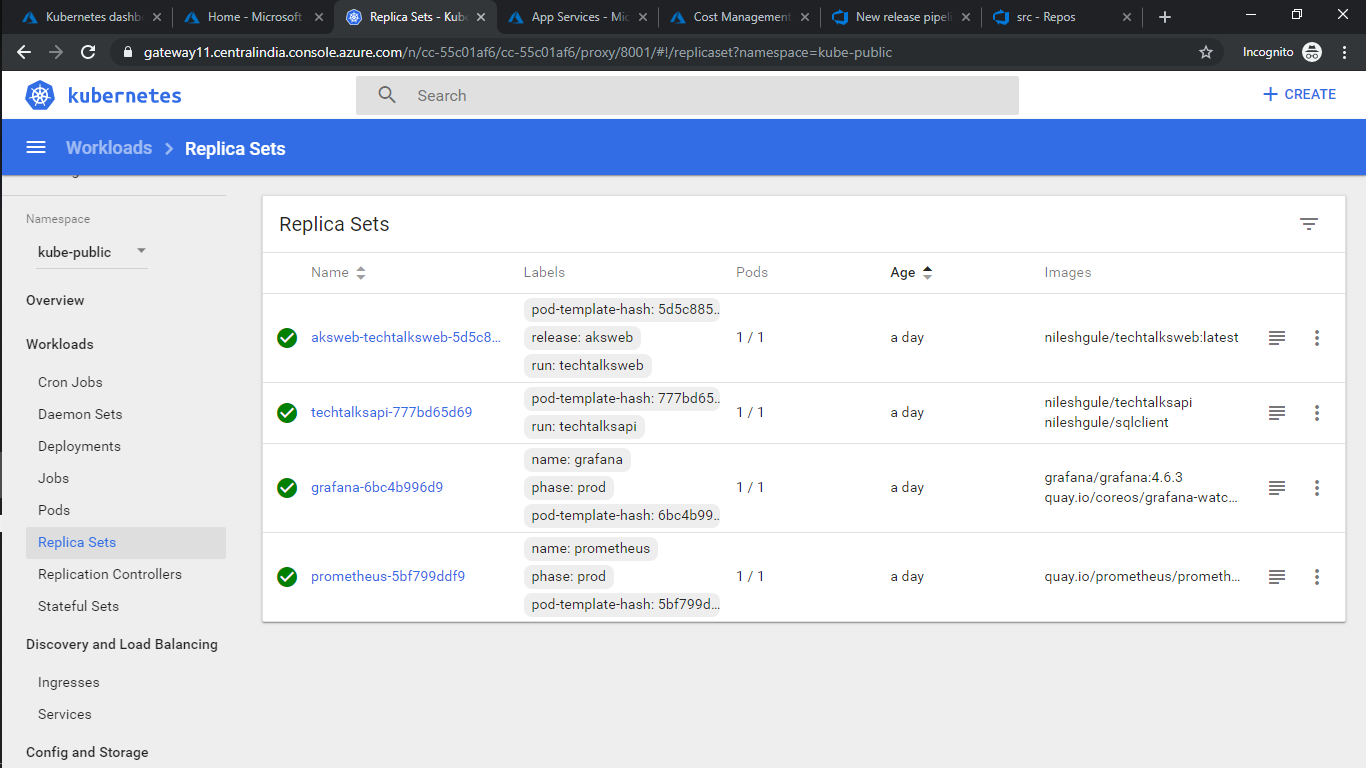
AKS Nodes



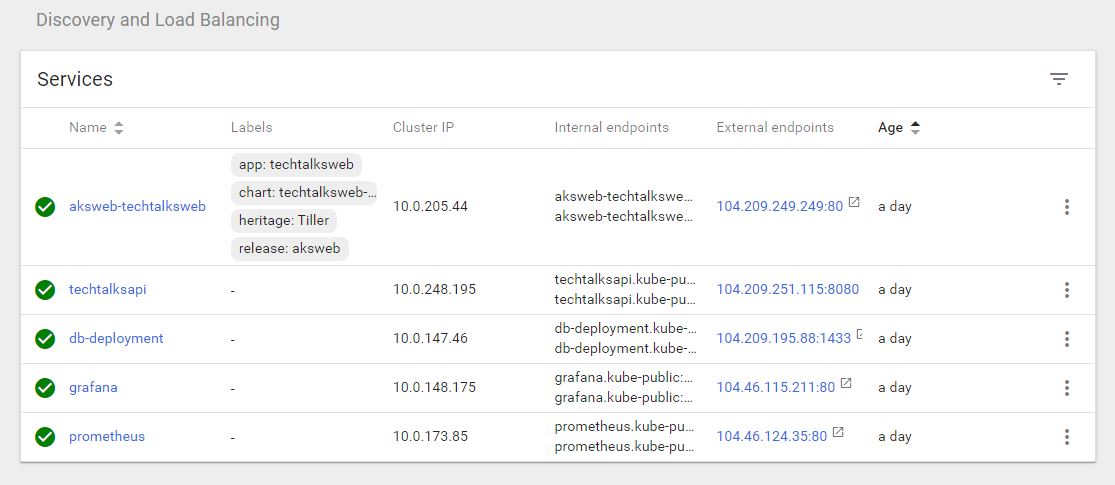
AKS Deployments



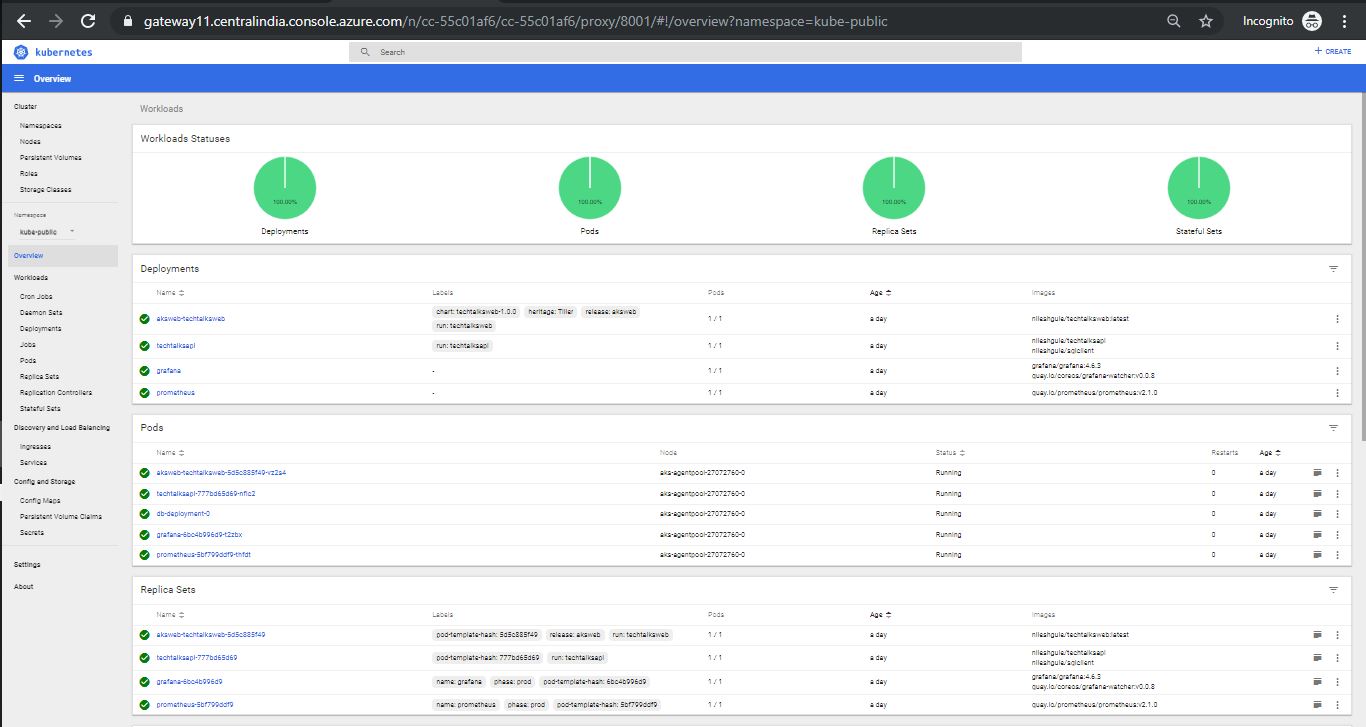
AKS Pods



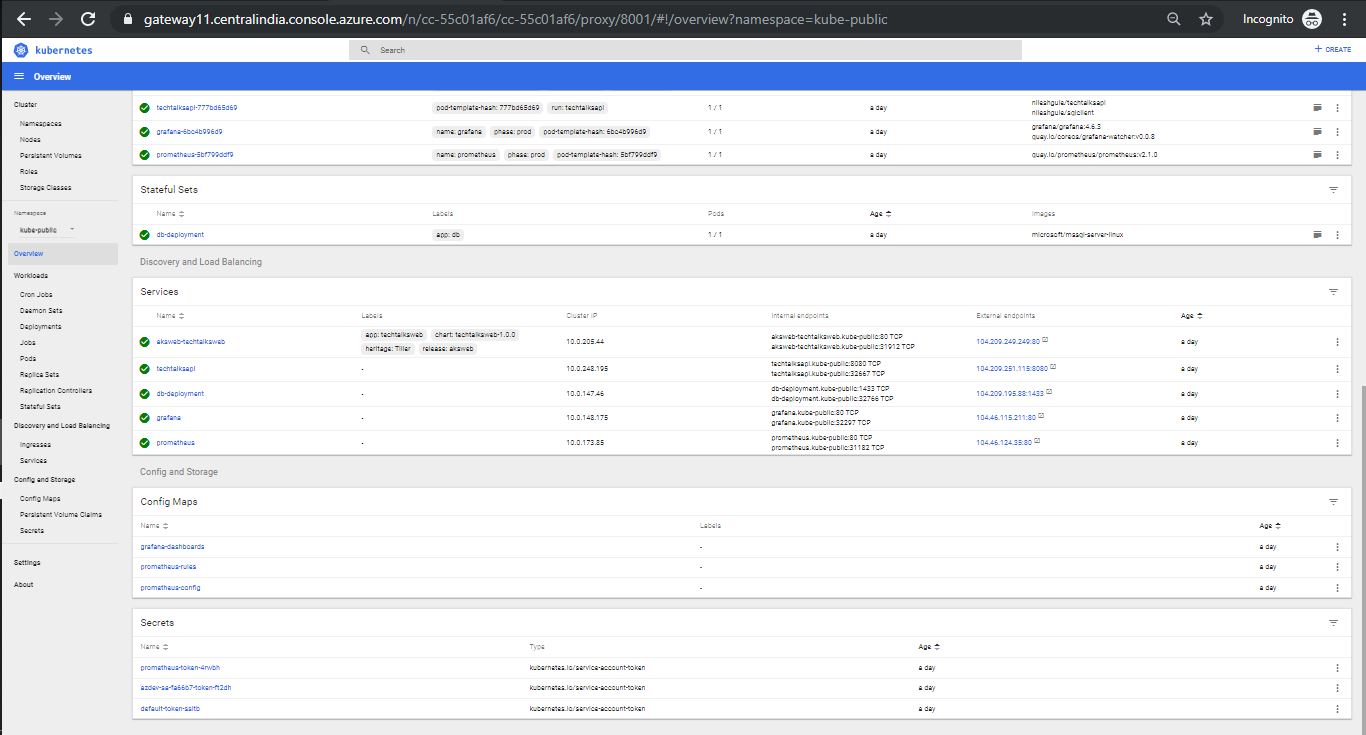
AKS Replica Sets



Discovery and Load balancing

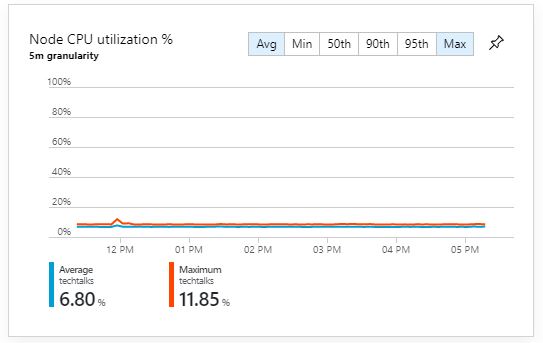


Overview of AKS Dashboard1

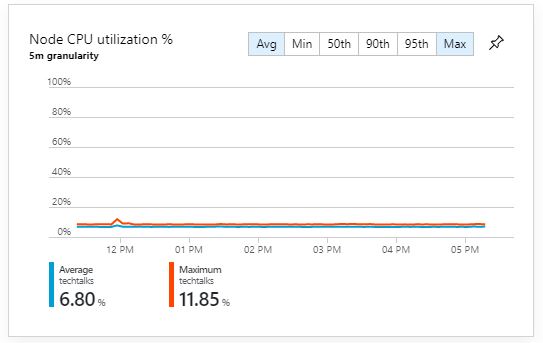


Overview of AKS Dashboard2

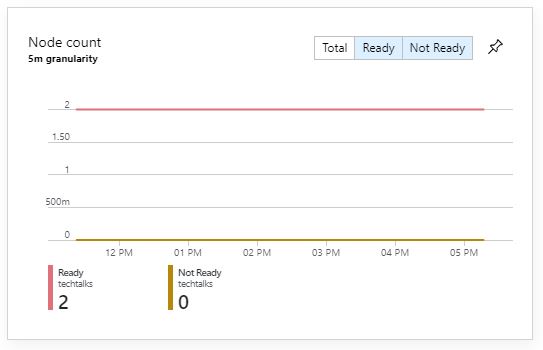
* 1. **Azure Insights** 
     1. **Techtalks Insights Cluster Metrics**



Node CPU utilization %

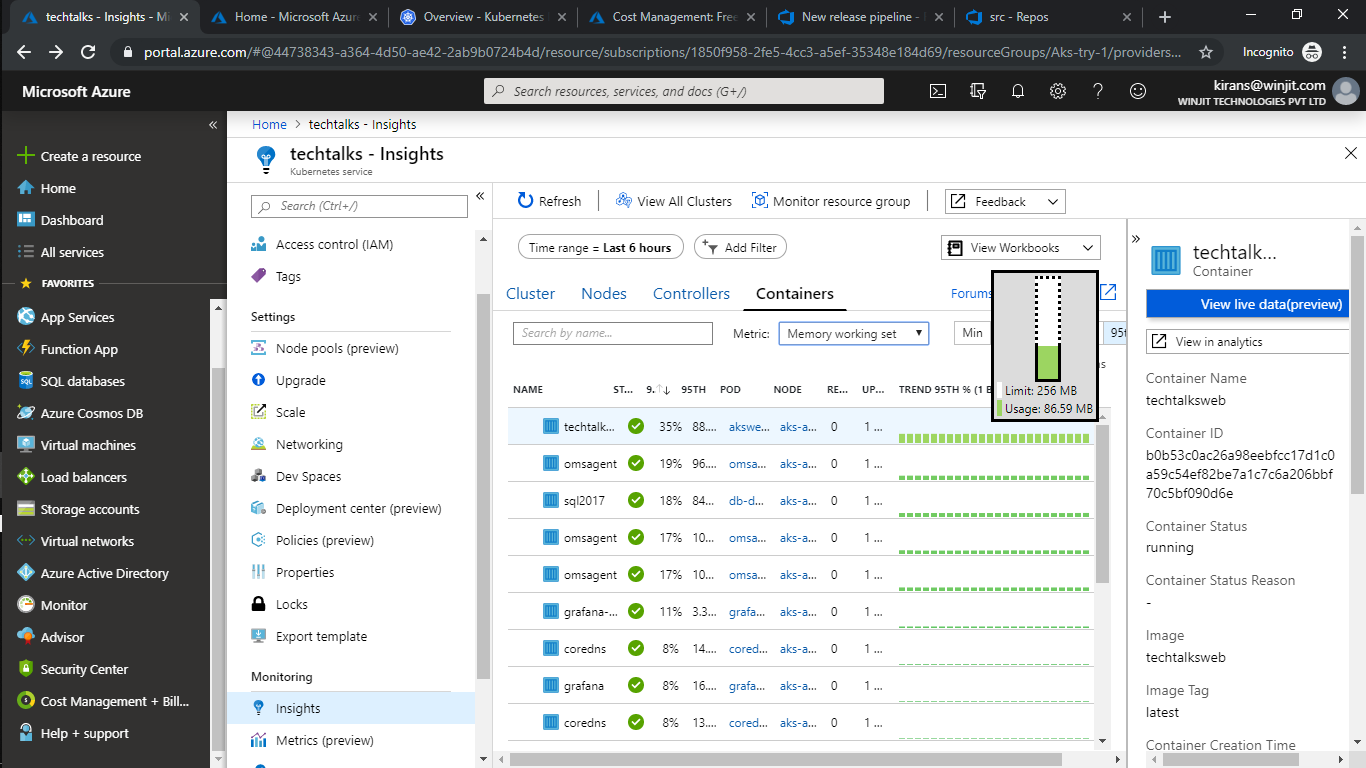


Node Memory Utilization Percentage

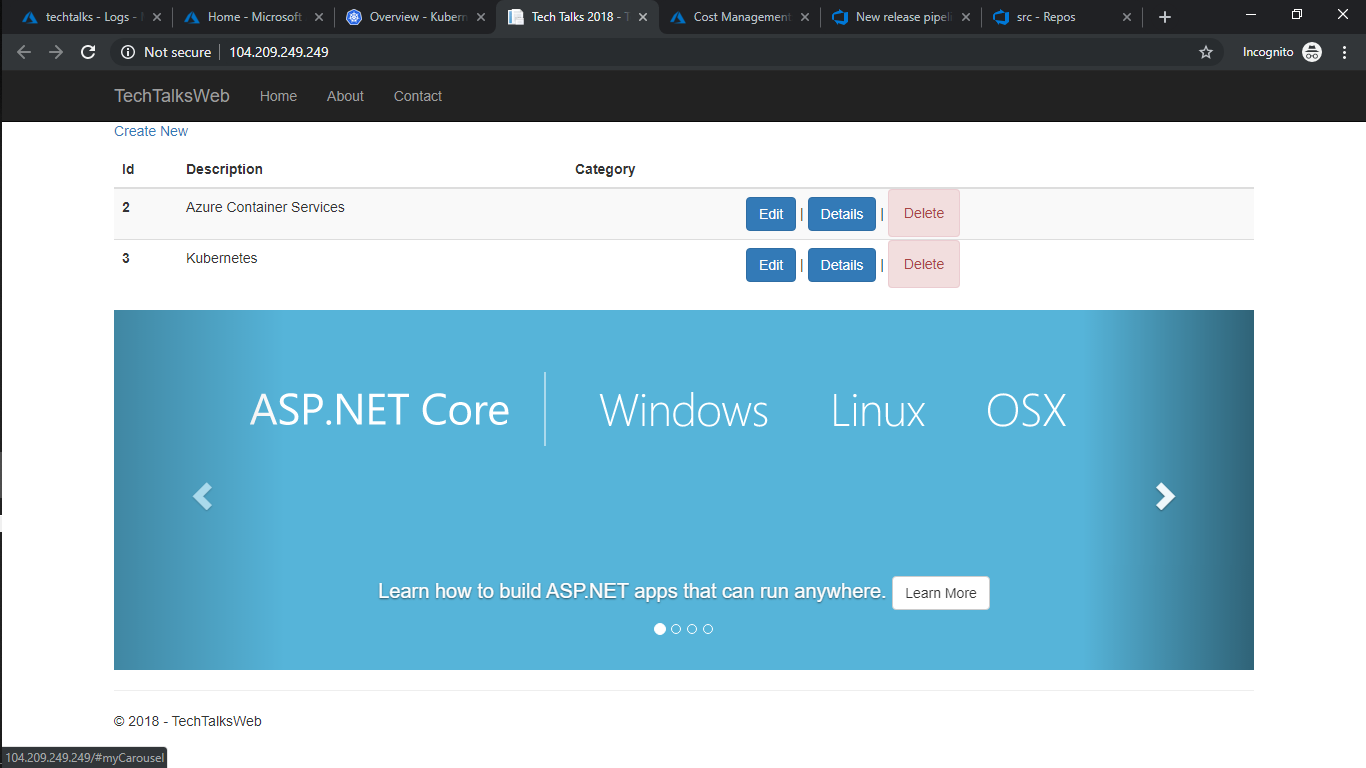


Node Count

* + 1. **Techtalks Insights Cluster Metrics**

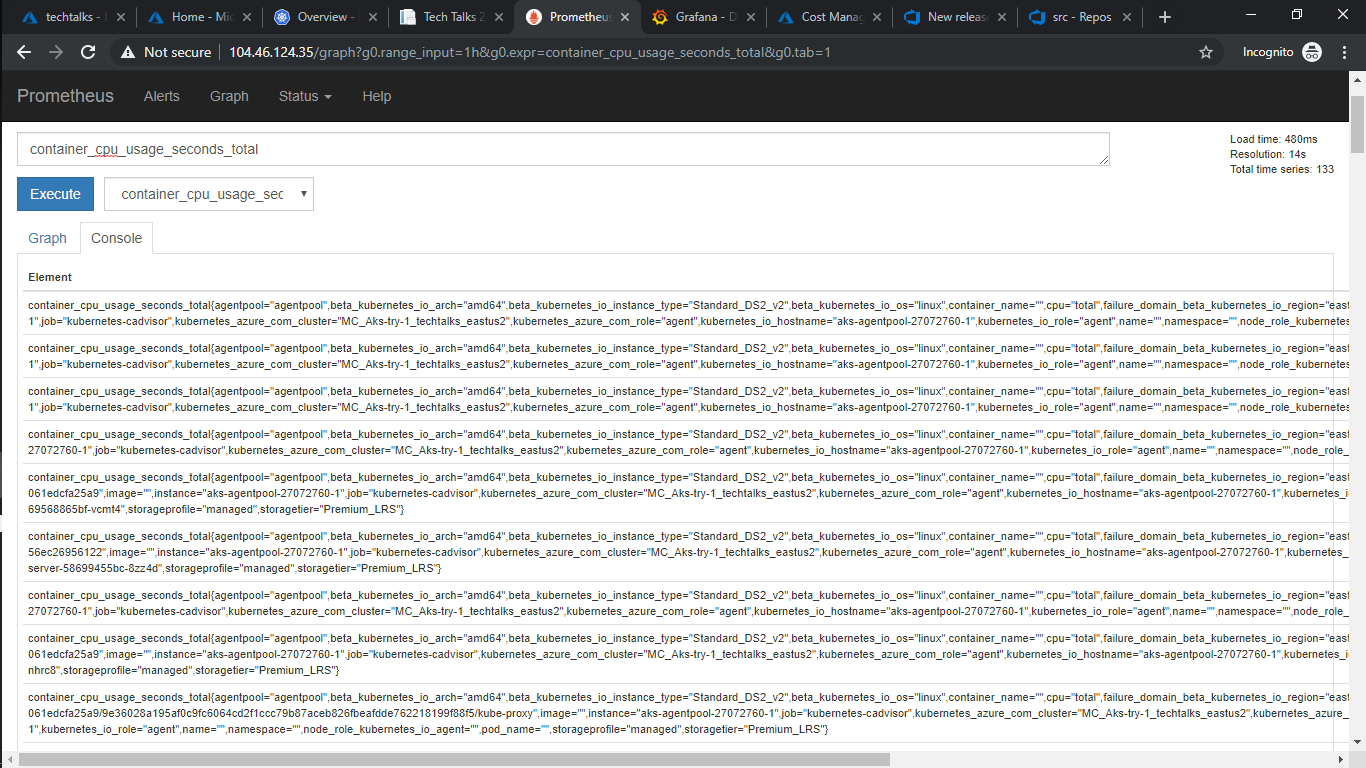


* + 1. **Web App in AKS**

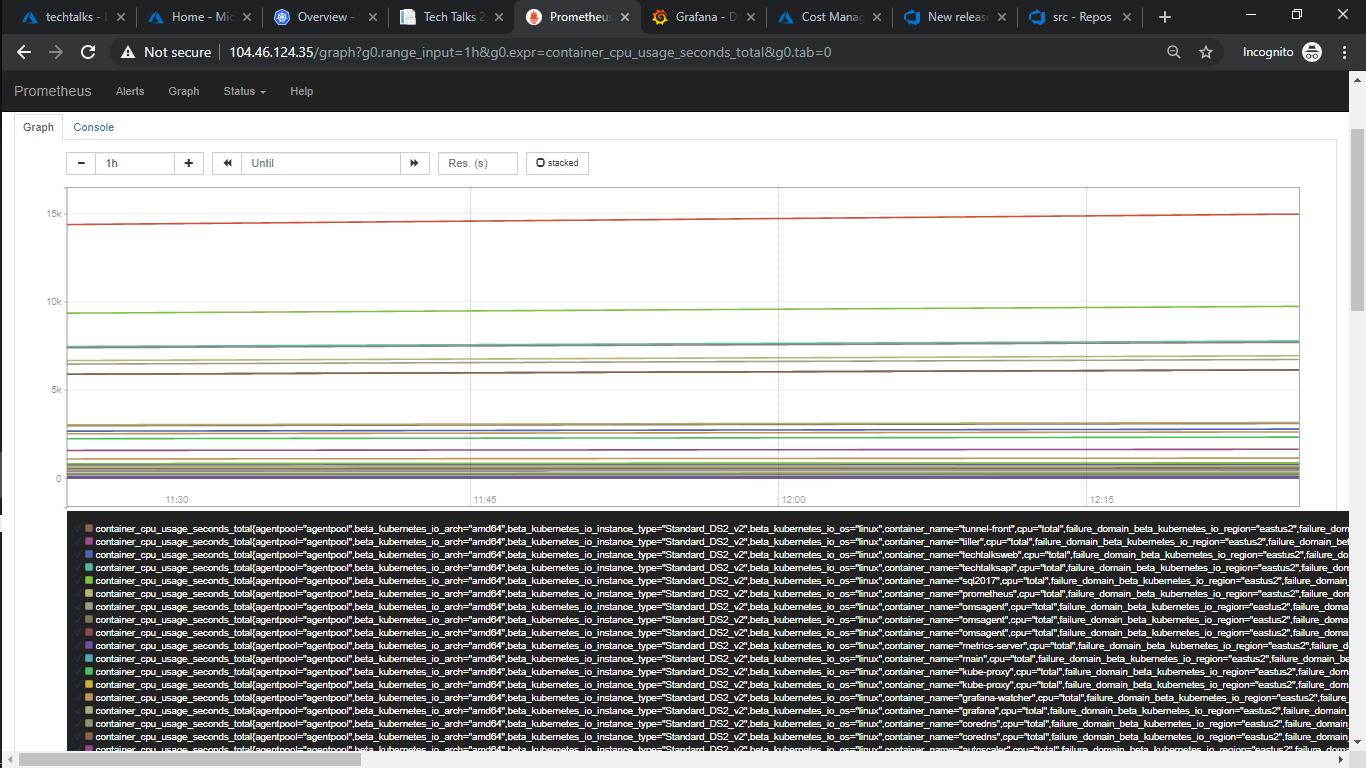


1. Monitoring

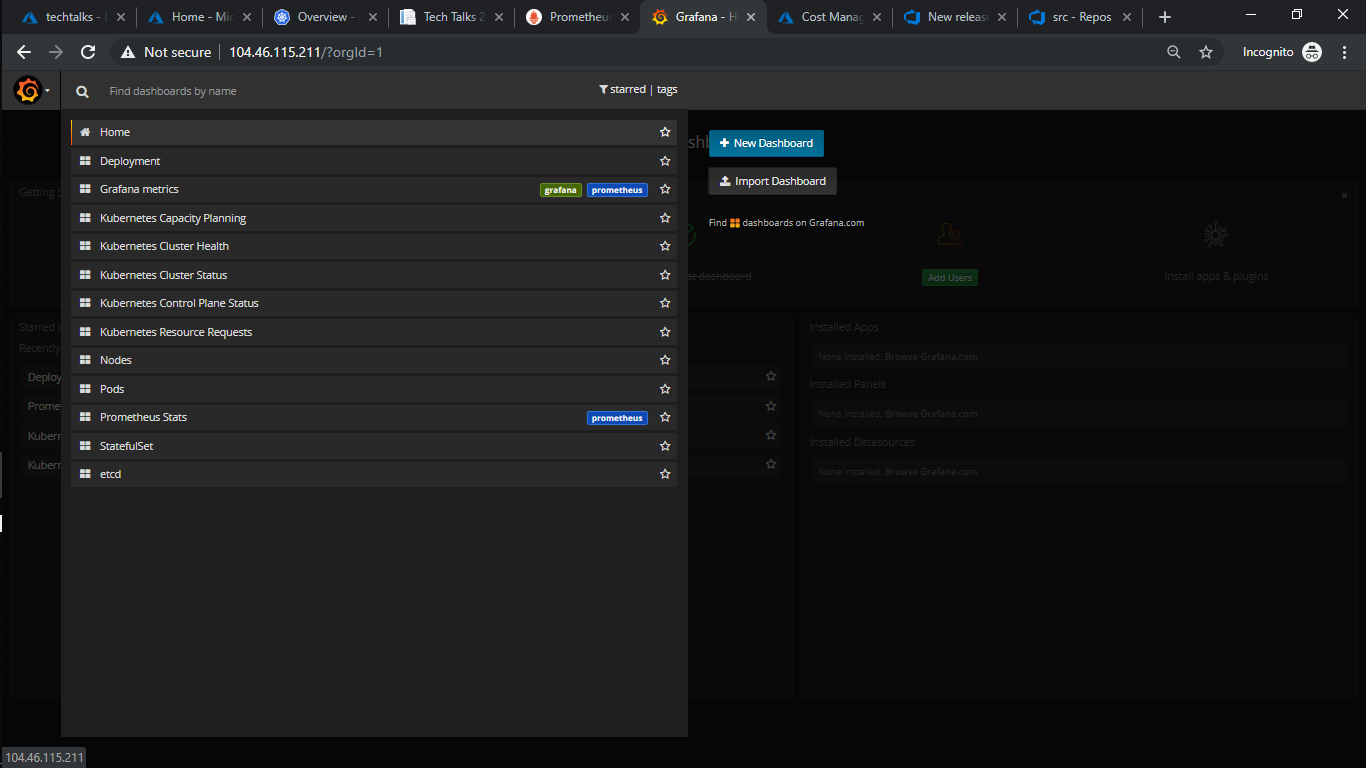
We use Prometheus to collect all the real time data of the container and for visualization we’ll use Grafana.



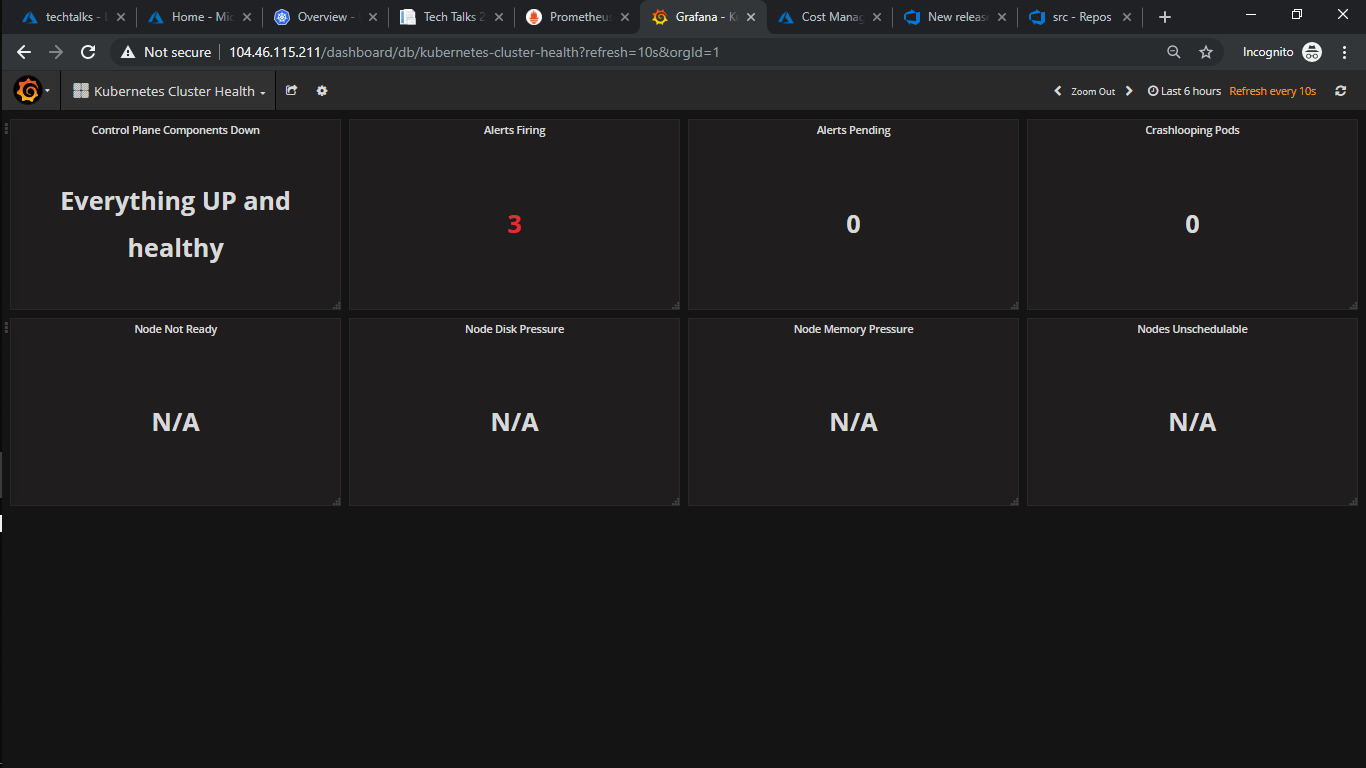
Prometheus container\_cpu\_usage\_seconds\_total



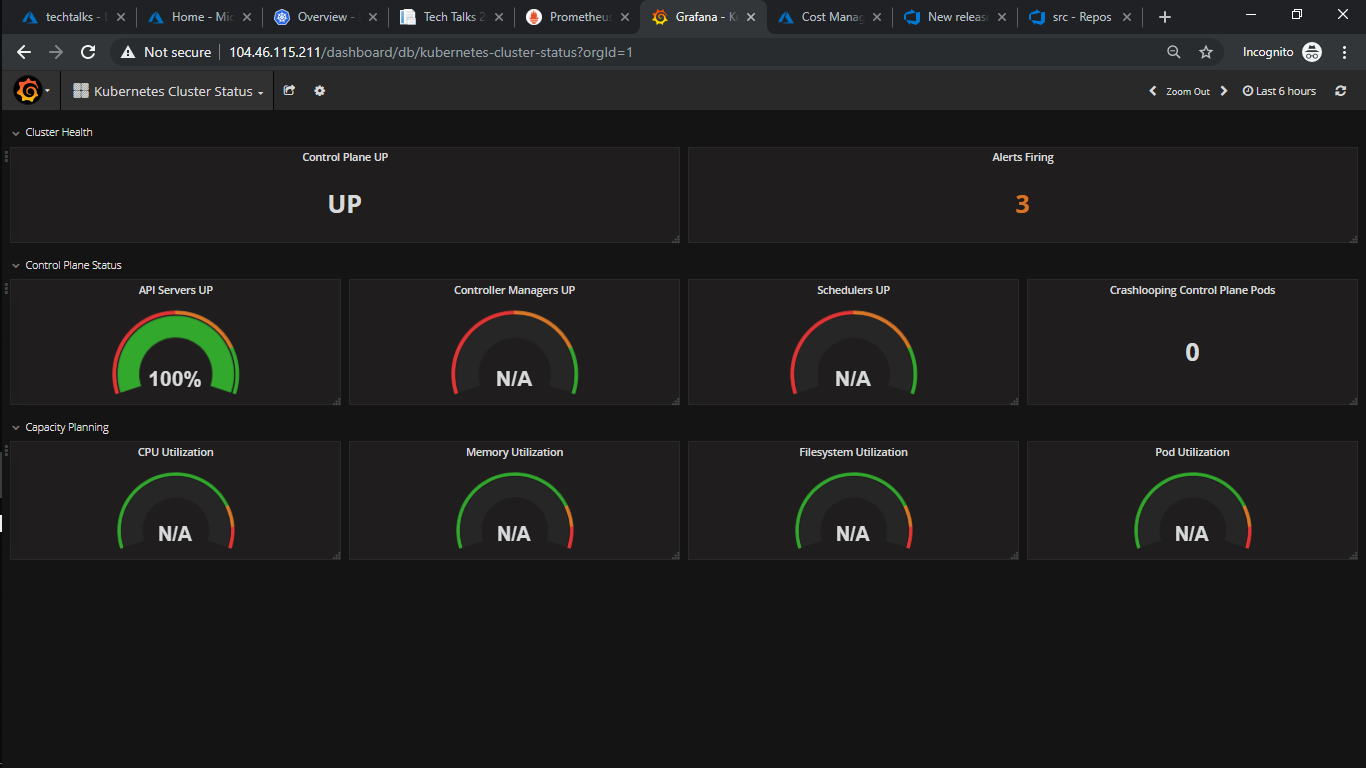
Prometheus container\_cpu\_usage\_seconds\_total Graph



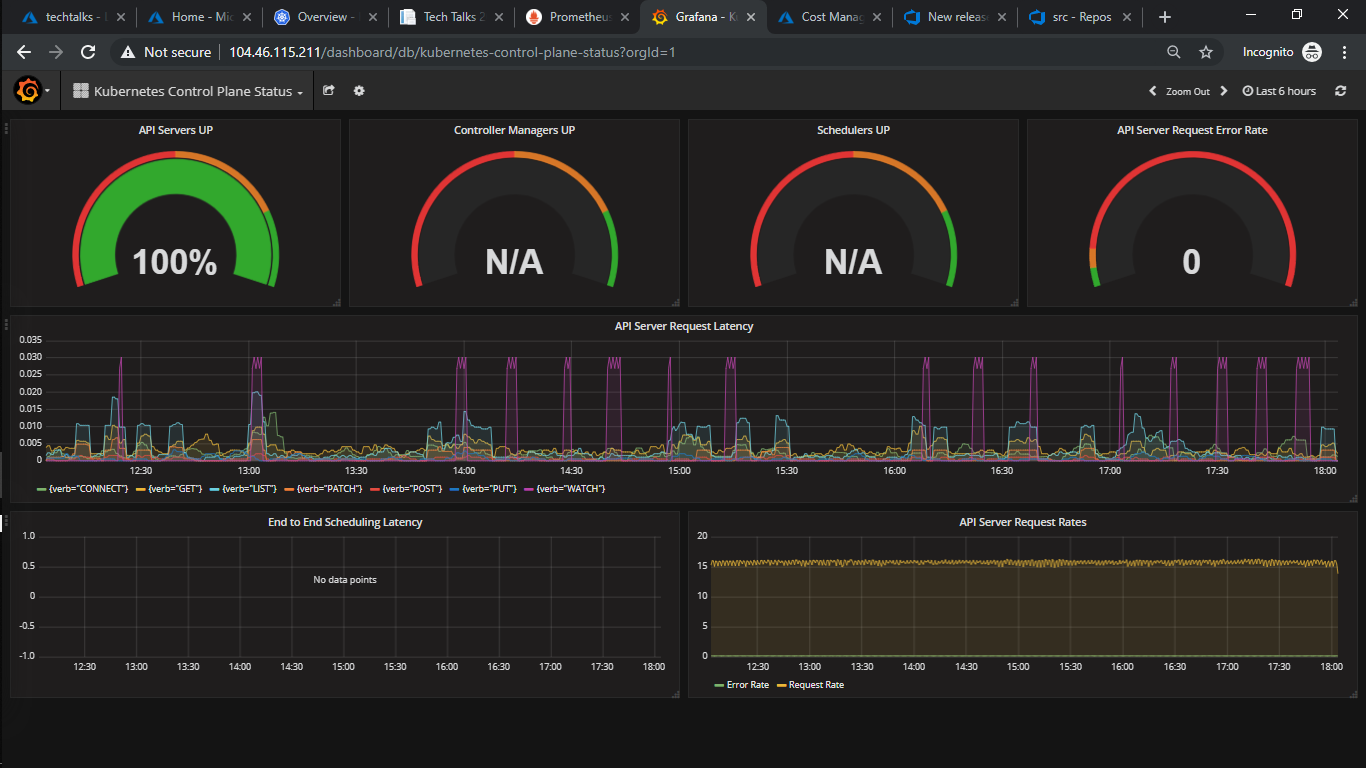
Grafana Different Dashboard Metrics



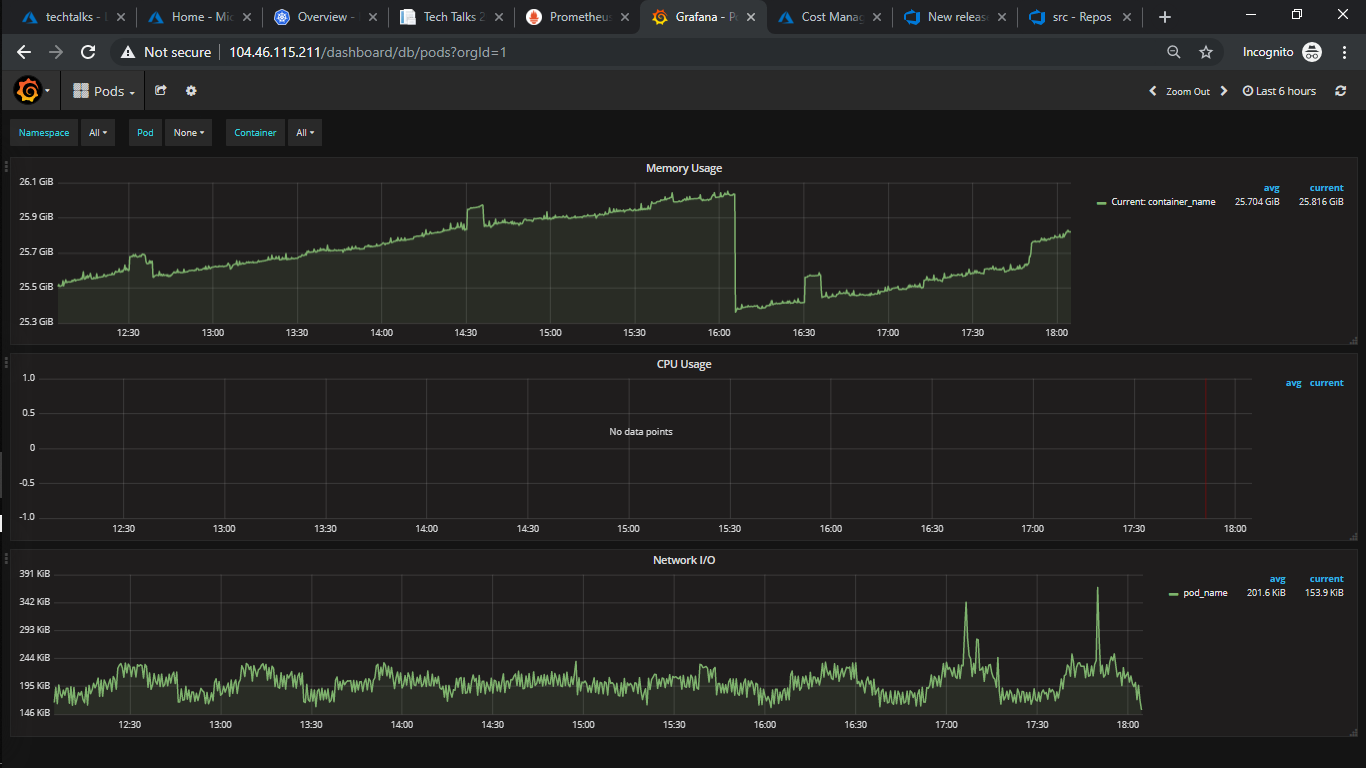
AKS Cluster Health



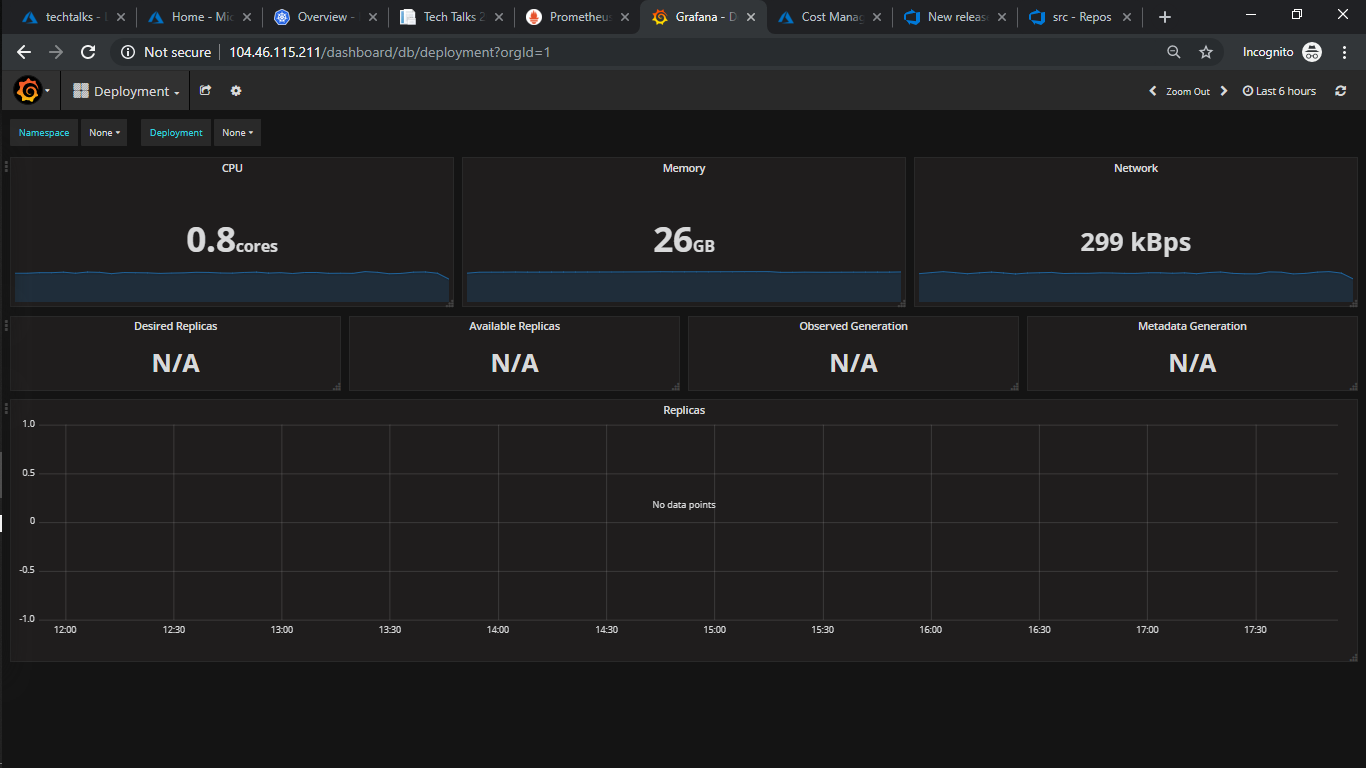
AKS Cluster API Status



AKS Control panel Status



AKS Pods Status



Grafana Deployment