Management Processes for application specific AKS clusters

|  |  |
| --- | --- |
| Document type | Template |
| Document title | Management Processes for application specific AKS clusters |
| Approved by |  |
| LOB |  |
| Company / Unit in charge |  |
| Valid from | xx.10.2022 |
| Valid to | Next revision |
| Next test |  |
| Template Owner |  |
| Document Owner |  |
| Version | 1.0 |
| Status |  |
| Valid until |  |

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change Description |
| 1.0 | 08.10.2022 | Sureshbabu Nadendla |  |

|  |  |  |
| --- | --- | --- |
| **Content** | **Version history** | **Author** |
| Order Namespace |  |  |
| Delete Namspace |  |
| Upgrade Kured |  |  |
| Upgrade Nginx Ingress |  |  |
| Upgrade CSI driver |  |  |
| Upgrade Resource quota |  |  |
| Upgrade Kubernetes version |  |  |

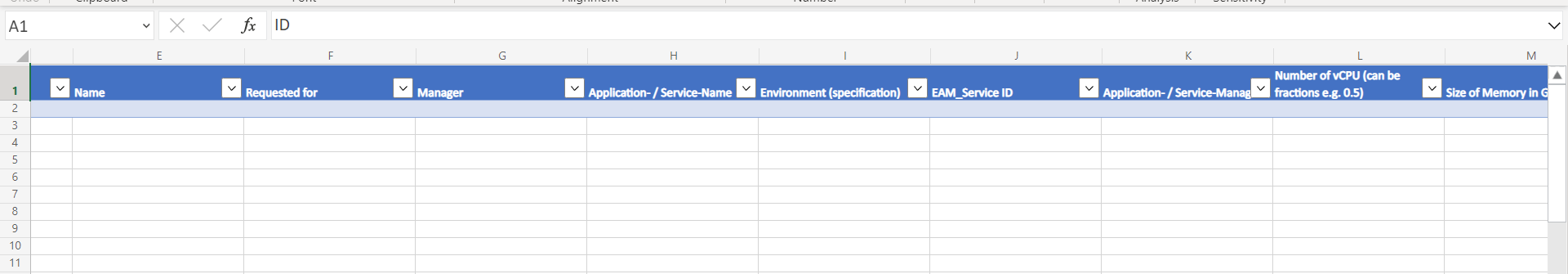
1. Namespace Order Process

For additional cluster the automated order process for namespace creation is not in place yet. Raise requests for namespaces for this cluster directly via mail to the HaCT AKS team [UIT-HACT-AKS@uniper.energy](mailto:UIT-HACT-AKS@uniper.energy) or [hact\_development@uniper.energy](mailto:hact_development@uniper.energy)

Details are required from Application team to order a Namespace through below listed form.

* AAD Group name
* Number of vCPU
* Size of Memory in GB
* Number of vCPU (limit)
* Size of Memory in GB (limit)
* Security Environment
* Cluster Name
* ADO Project Name for Service connection
* Environment tag
* Namespace naming convention should be like: applicationname-EAM-id-env, etc ..

Application team needs to update above details in the below excel form received via email from HaCT AKS team



Once the form is updated with the required information HaCT team will proceed to create the namespace.

Application team should ensure the following details for their Namespace.

* **RBAC**   
  Each namespace will be assigned to a namespace owner (an individual or an Active Directory group). The namespace owner can define namespace roles and assign it to additional users or teams within the assigned namespace.
* **Network policies**  
  Namespace are by default isolated from other namespace by a "Block all" network policy. This network policy can be edited by application teams to enable specific communication. It is strongly recommended to allow specific communication and not open communication in general or even delete the network policy.
* **Azure security policies**  
  Azure security policies control the permissions of a pod on the underlying node so that pods can’t execute operations that have an impact on other pods running on the same node. Pod security policies cannot be changed by application teams. Please find below wiki URL for polices
* [Azure Policy built-in definitions for Azure Kubernetes Service - Hosting Framework (uniper.energy)](https://wiki.intranet.uniper.energy/sales/display/CF/Azure+Policy+built-in+definitions+for+Azure+Kubernetes+Service)
* **Verify the newly created Namespace and Resource quota**

**Note:** You will get ‘No resources found in applicationname-EAMid-env namespace.’ If you do not have even a single resource types.

// Pods  
$ kubectl get pods -n namespace\_name

// Fetch details of capacity you have requested to us we set this is ‘Resource Quota’   
$ kubectl describe quota requests-resourcequota --namespace=namespace\_name

1. Namespace deletion process

Namespace administrator has the overall responsibility of managing Namespace but restricted with namespace deletion and HaCT AKS Team is responsible for Namespace deletion.

HaCT AKS will delete the Namespace as based on the Namespace deletion SNOW catalogue [Service Catalog - Uniper Service Portal (service-now.com)](https://uniperprod.service-now.com/unipersp?id=sc_cat_item_uni&sys_id=ca5613dadbf82c10d1ebbaf1f3961961), Once the request in place we will process within 2 to 3 working days.

HaCT AKS will update the comments section of ServiceNow ticket after successful deletion of Namespace which triggers an email regarding Namespace deletion status.

**Key Stakeholder**

Requester (Namespace Owner)

HaCT AKS Team

Line Manager

**Input Requirements**

Requested for (Name will be auto filled)

Line manager (Name will be auto filled)

Namespace name

User ID (KID) of Namespace Administrator

Additional Contact Information (If any)

Additional Information (Example: Cluster Name.)

After successful deletion of Namespace, AKS Team will update the requester via email.

1. HaCT AKS components (kured, nginx, CSI driver) Upgrade process

HaCT AKS will update the Application team through mail 6 weeks prior of the Prod clusters Patching and 4 weeks prior to the other environments.

HaCT AKS team will send email notifications to the corresponding teams for upgrading the components and its versions table and do changes in the latest versions.

**Upgrade Kured:**

Kured (Kubernetes Reboot Daemon) is a Kubernetes daemonset that performs safe automatic node reboots when the need to do so is indicated by the package management system of the underlying OS.

**Example Cluster name: cmcaks-poc-001**

Login to cmcaks-poc-001 cluster

$ az account set --subscription d4665e52-c01b-4bdf-aa11-3da2eac23394

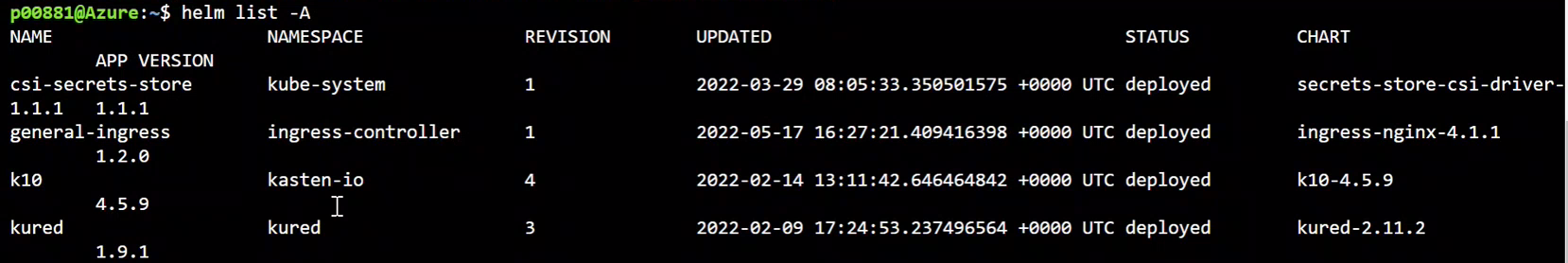
$ az aks get-credentials --resource-group cmcaks-paas-poc-rgp-001 --name cmcaks-poc-001

First, we need to upgrade some of the applications before cluster upgrade

These applications deployed using HELM Package manager

$ helm list -A

$ helm repo list



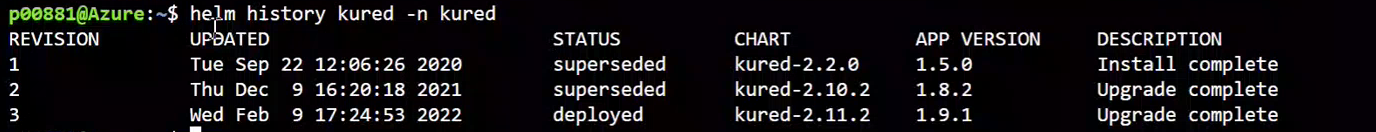
Check the history of kured application

$ helm history kured -n kured

$ k get ds -n kured

$ k get po -n kured

$ kubectl get pods -A --field-selector status.phase!=Running -o=custom-columns=NAME:.metadata.name,STATUS:.status.phase,NAMEDPACE:.metadata.namespace



Delete kured repo if it is existed already

$ helm repo remove kured

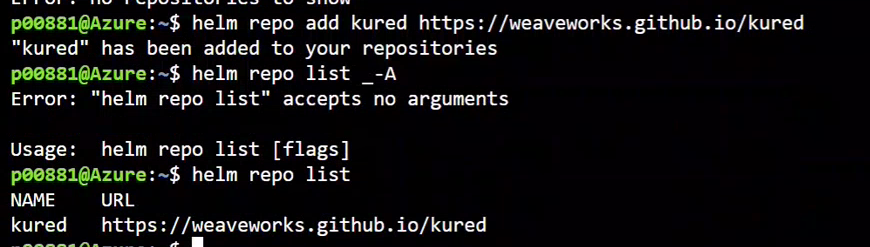
Add repo from Application URL

$ helm repo add kured <https://kubereboot.github.io/charts>

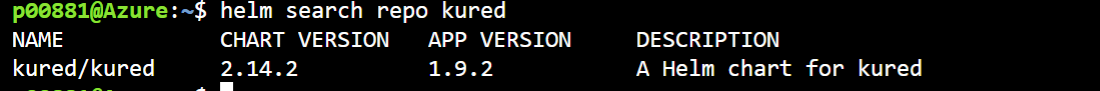
if unable to delete namespace in prod do this one

$ kubectl get apiservice | grep kas

$ k delete apiservice v1alpha1.apps.kio.kasten.io v1alpha1.vault.kio.kasten.io



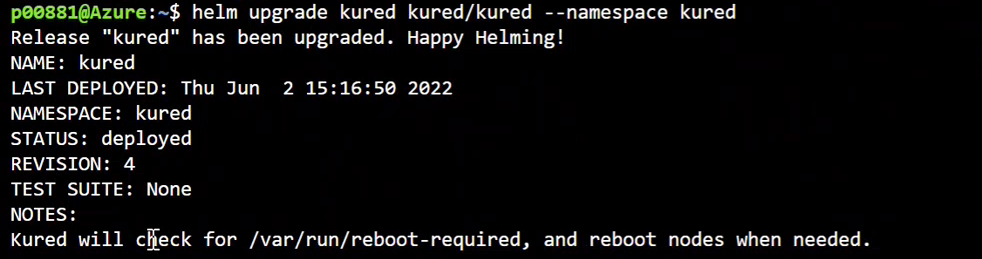
Upgrade kured application to latest one



$ helm upgrade <REPO NAME> <CHART NAME> -N <NAME SPACE NAME>

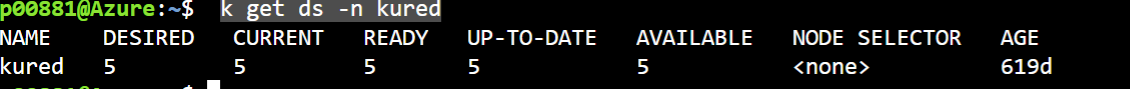
$ helm upgrade kured kured/kured --namespace kured

$ helm install kured kured/kured -n kured



Validate kured daemonsets up and running

$ k get ds -n kured



Validate kured application upgraded successfully or not and check all the pods are newly created or not.

$ Kubectl get pods -n kured

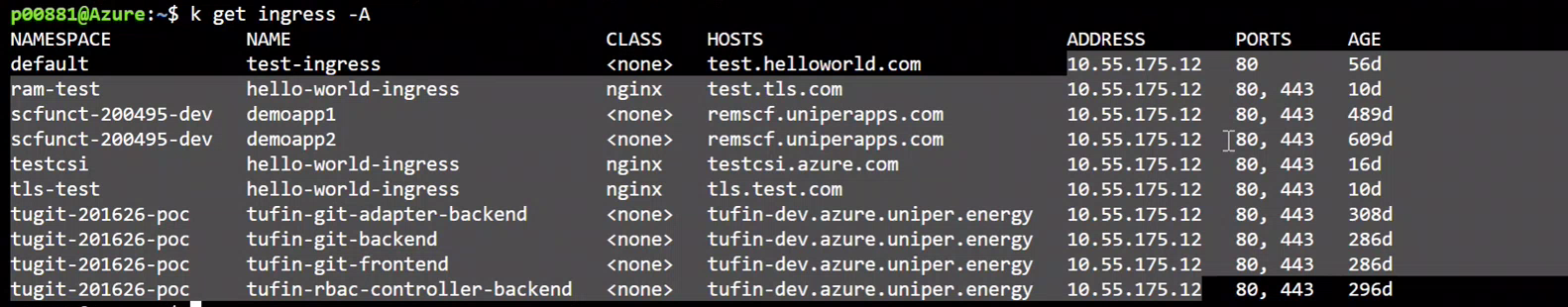


* **Note:-**No Downtime require for this and no need to communicate to Application Team for this.

**Upgrade Nginx:**

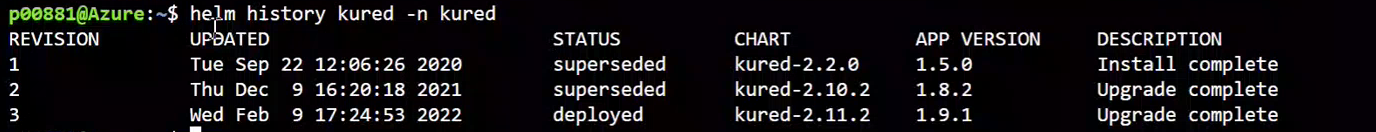
First check all the ingress controller is running or not

$ kubectl get ingress -A



Check the history of general-ingress application

$ helm history general-ingress -n ingress-controller



Delete general-ingress repo if it is existed already

$ helm repo remove general-ingress (DON’T REMOVE)

Add repo from Application URL

$ helm repo add general-ingress https://kubernetes.github.io/ingress-nginx (DON’T REMOVE)

JUST DO

$ helm repo upgrade

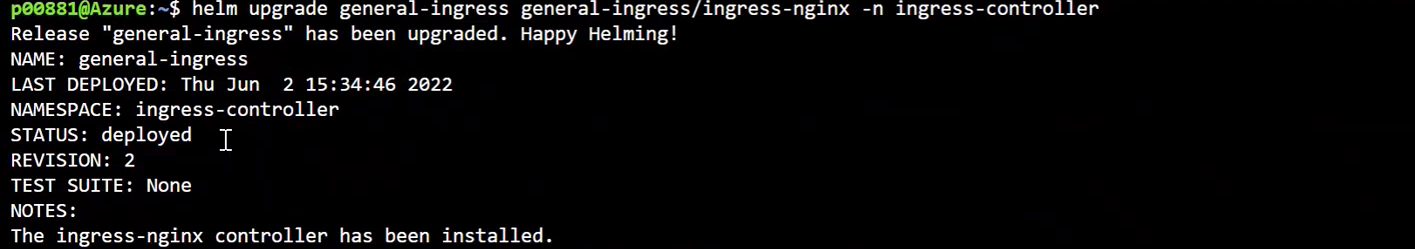


Upgrade general-ingress application to latest one

$ helm upgrade <REPO NAME> <CHART NAME> -N <NAME SPACE NAME>

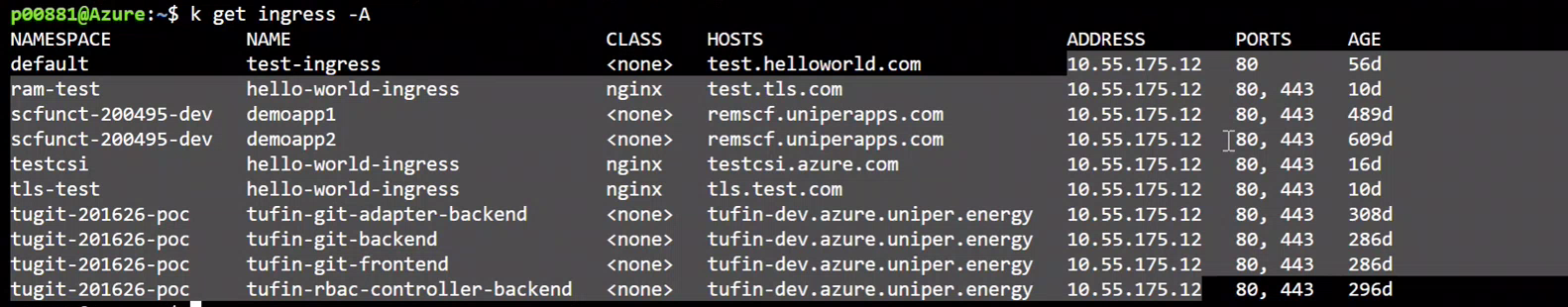
$ helm upgrade general-ingress general-ingress/ingress-nginx -n ingress-controller

$ helm upgrade ingress-nginx general-ingress/ingress-nginx -n ingress-controller[WORKING]

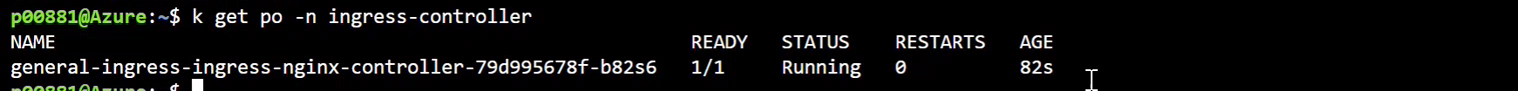


Validate general-ingress up and running

$ kubectl get ingress -A



$ k get po -n ingress-controller



**How does it relate to the cluster patching?**

We need to check the compatibility of the ingress controllers with Kubernetes version. So, we can upgrade this as per required version for Kubernetes Version.



**Upgrade Secret Store CSI driver:**

*helm upgrade* to the latest chart release in the repo will update the Secret Store CSI Driver to the compatible versions.

$ helm upgrade csi-secrets-store secrets-store-csi-driver/secrets-store-csi-driver --namespace=NAMESPACE

where the driver was originally installed, (i.e. kube-system)

**How and when are users informed?**

HaCT will initiate email to Application team regarding cluster upgrade 4 weeks prior to upgradation. For Production cluster we inform 6 weeks earlier.

**Who has which responsibility in the process?**

HaCT AKS team have responsibility to upgrade the cluster.

Note:- HaCT AKS team will include any one of the application team during upgradation time and will ask them for validation of their application after Cluster upgradation.

**How does the communication work?**

After successful upgradation AKS team communicate to Application team via email.

**How does it relate to the cluster upgradation?**

HaCT AKS team can check the Compatibility of drivers and Providers with AKS.

Note:-Timeline between non-prod and production upgradation is 4-6 weeks.

**Request for the resource quota update:**

Based on the Application team request HaCT AKS team can upgrade the resource quota by updating the below catalogue request : <https://uniperprod.service-now.com/unipersp?id=sc_cat_item_uni&sys_id=2a59125387118910e1b1a8e70cbb355f>

Default section - sets up the default limits for a container in a pod. If you set these values in the limit Range, any containers that don’t explicitly set these themselves will get assigned the default values.

Default Request section - sets up the default requests for a container in a pod. If you set these values in the limit Range, any containers that don’t explicitly set these themselves will get assigned the default values.

Max section - Sets up the maximum limits that a container in a Pod can set. The default section cannot be higher than this value. Likewise, limits set on a container cannot be higher than this value. It is important to note that if this value is set and the default section is not, any containers that don’t explicitly set these values themselves will get assigned the max values as the limit.

Min section - Sets up the minimum Requests that a container in a Pod can set. The default Request section cannot be lower than this value. Likewise, requests set on a container cannot be lower than this value either. It is important to note that if this value is set and the default Request section is not, the min value becomes the default Request value too.

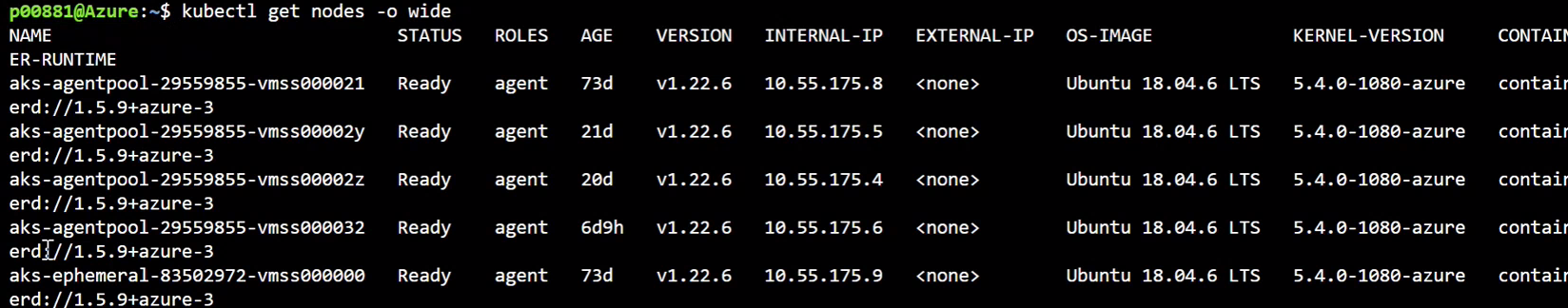
1. AKS cluster upgrade process

HaCT AKS team will update to Application team through mail 6 weeks prior of the Prod clusters Upgrade and 4 weeks prior to the other environments.

HaCT AKS team will send email notifications to the corresponding teams for upgrading the AKS cluster and its versions table and major changes in the new version.

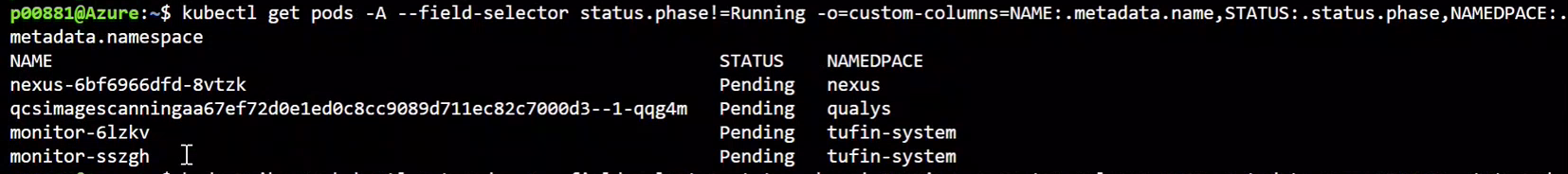
HaCT AKS team will check all the nodes, pods and other resources up and running before starting the upgrade process.

$ kubectl get nodes -o wide

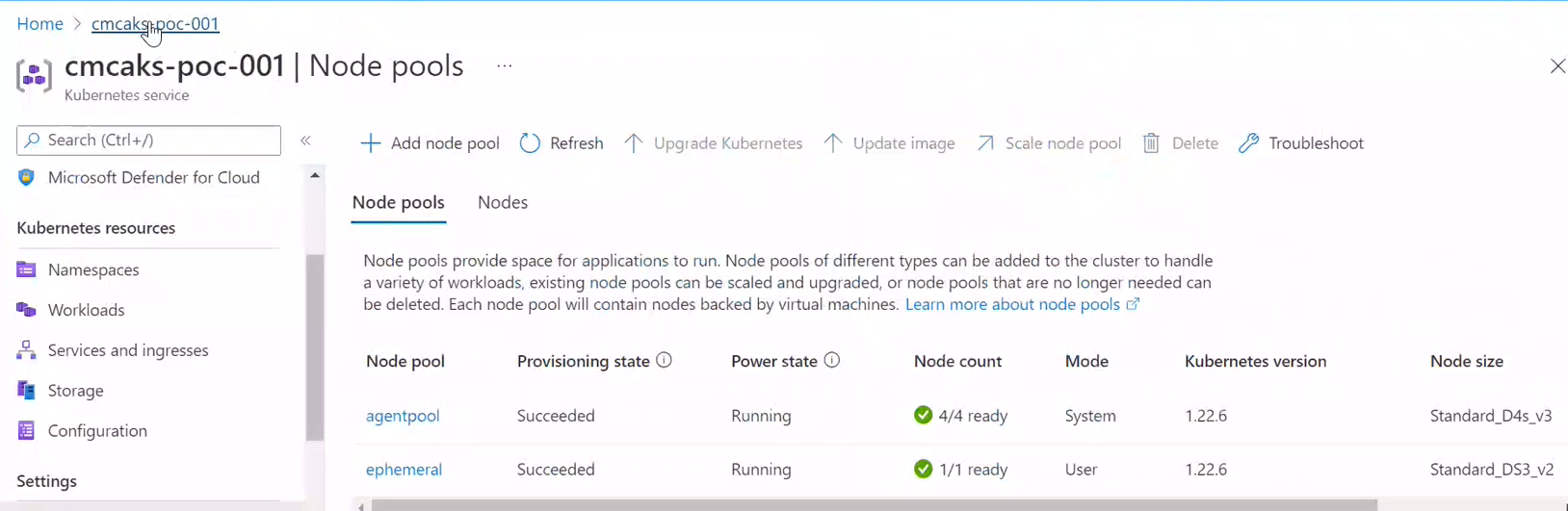


Validate the pods which are not running and get the Namespace of that POD

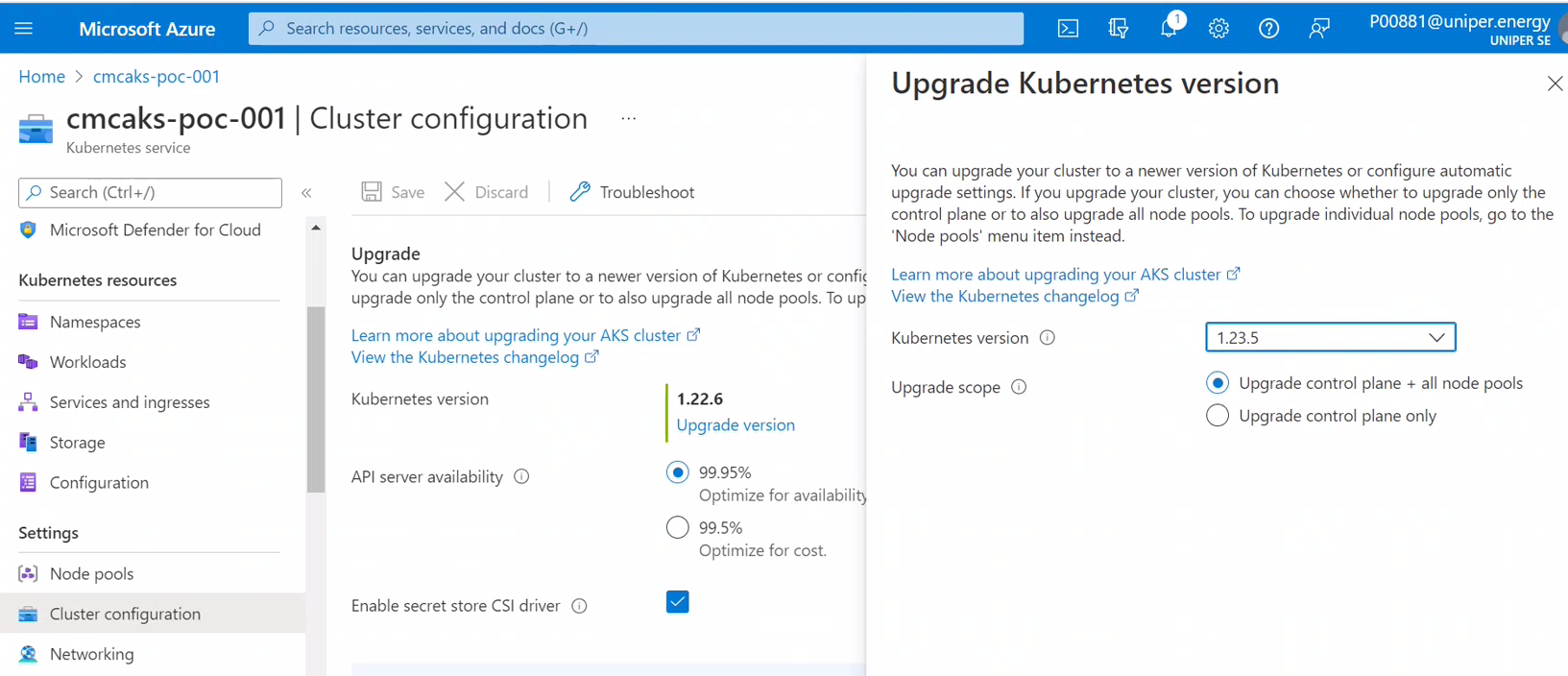
$ kubectl get pods -A --field-selector status.phase!=Running -o=custom-columns=NAME:.metadata.name,STATUS:.status.phase,NAMEDPACE:.metadata.namespace



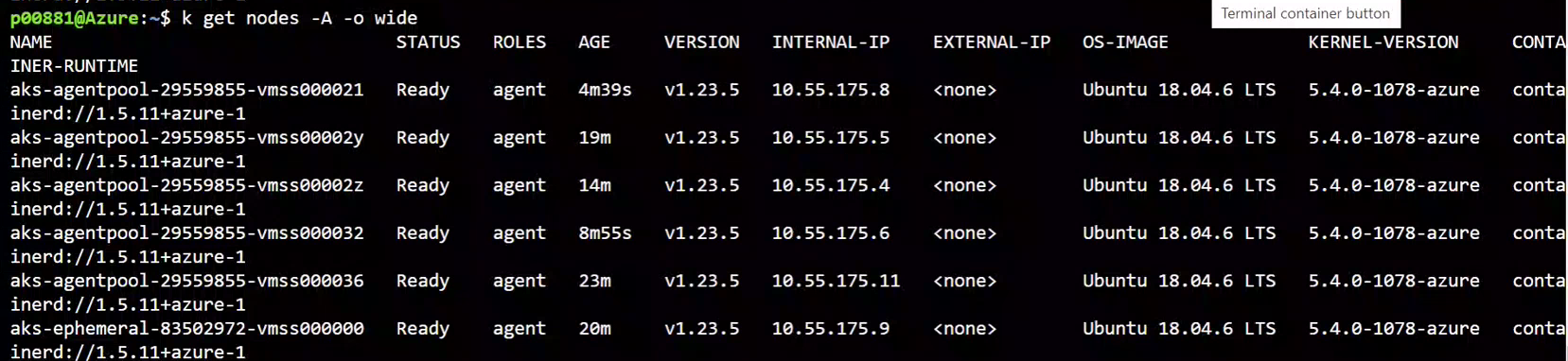
And validate the nodes are running or not in azure portal as well



Upgrade the Kubernetes version as shown below screenshot



After successful of the Kubernetes version upgrade, need to validate the nodes by passing below command.



HaCT team will send email to application teams.

For production environment before start upgradation, HaCT team will RAISE CHANGE REQUEST.

A screenshot of a computer

Description automatically generated