EGS: Admin Guide

Introduction

Elastic GPU Service platform provides a system and workflows for effective resource management of GPUs across one or more Kubernetes clusters.

EGS supports two different personas: Admin and User (Data Scientist, ML engineer/Team)

Admin
Admin is responsible for the installation and administration of EGS platform. EGS provides an Admin portal to perform the Day 0/1/2 operations. EGS also supports YAML (manifests) based admin workflows for these operations so that these workflows can be integrated with CI/CD or MLOps pipelines.

User
User (can be a Data Scientist, Researcher or ML engineer) uses EGS User portal to create and manage the life-cycle of GPU provisioning requests for User's Slice(s). Uses the portal to get deep visualization of the AI workloads and associated GPUs metrics and other data.

EGS Documents

This document describes the EGS Admin operations related workflows:

- For EGS platform overview please see the documentation on the website <Link>
- For User guide please see the documentation on website <link>
- For Installation guide please see the documentation on github repo <link>

Installation

EGS provides a combination of Helm charts, CLI and shell scripts for easy installation procedure.

- git clone the `egs-installation` repo - https://github.com/kubeslice-ent/egs-installation

- Follow the Installation steps specified in the EGS Installation Guide.

Note: Installation scripts create a default project workspace and register worker cluster(s).

Admin Access Token

Admins can access EGS UI portal using two different methods:

1. Using Access Token
2. Using IDP Access Token (when cluster and EGS is enabled with Idp integration)

By default an admin access token will be created in the system as part of the installation.

Run the following script (from installation guide) to get the Admin access token:

```
Unset
##%./egs-get-admin-access-token.sh -h

%kubectl get secret kubeslice-rbac-rw-admin -o jsonpath="{.data.token}" -n
kubeslice-avesha | base64 --decode
```

Note down the admin access token

Access EGS UI Portal

```
Unset
%/home/user/egs-installation$ kubectl get svc -n kubeslice-controller
NAME                                                    TYPE
CLUSTER-IP    EXTERNAL-IP   PORT(S)         AGE
```

```
gpr-manager                                                    ClusterIP
10.7.44.212   <none>        8088/TCP        46h
kubeslice-api-gw                                               ClusterIP
10.7.38.173   <none>        8080/TCP        46h
kubeslice-controller-controller-manager-metrics-service   ClusterIP
10.7.46.137   <none>        8443/TCP        46h
kubeslice-controller-release-prometheus-service           ClusterIP
10.7.43.250   <none>        9090/TCP        46h
kubeslice-controller-webhook-service                      ClusterIP
10.7.40.229   <none>        443/TCP         46h
kubeslice-ui                                                   ClusterIP
10.7.42.178   <none>        80/TCP          46h
kubeslice-ui-proxy                                             LoadBalancer
10.7.45.163   34.19.39.39   443:31322/TCP   46h
kubeslice-ui-v2                                                ClusterIP
10.7.39.69    <none>        80/TCP          46h
```

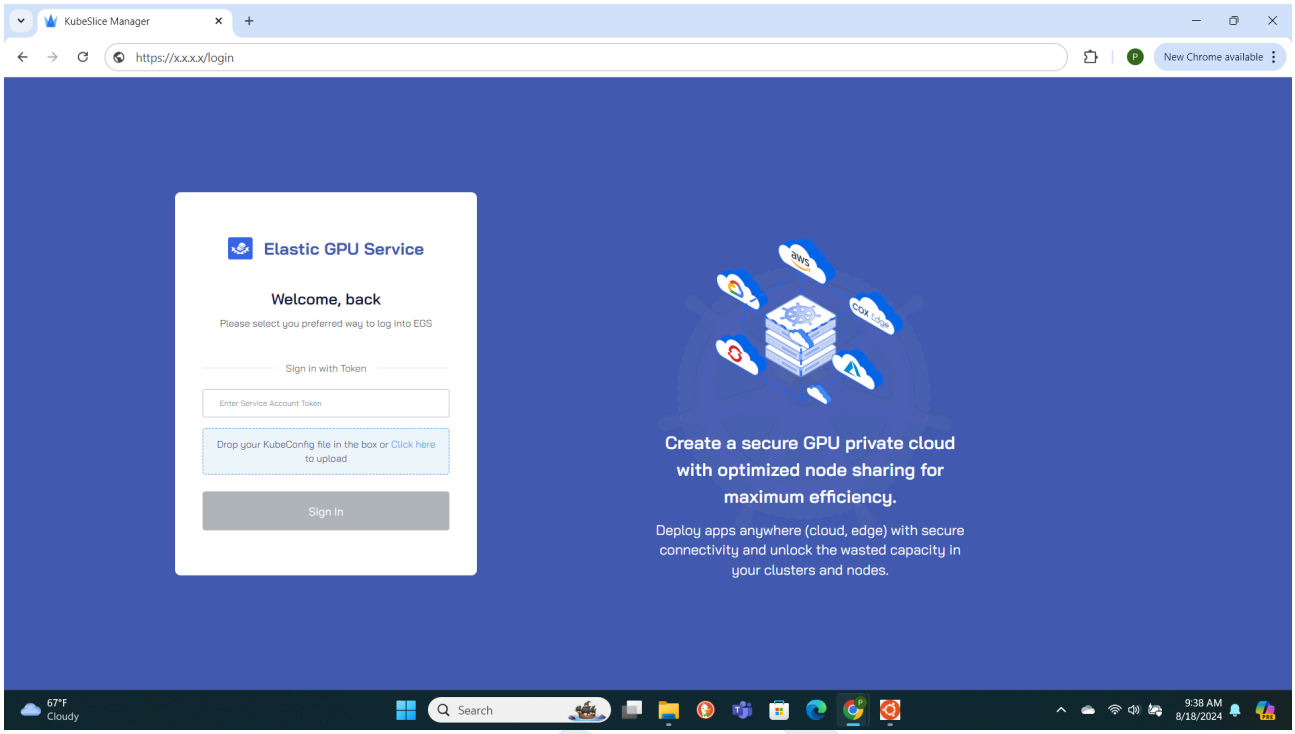
- Note down the LoadBalancer external IP for the kubeslice-ui-proxy pod. This IP will be used to access the EGS Admin/User portals.

UI Portal: **https://<ui-proxy-ip>**

Login to Admin Portal

Login to the EGS UI portal with the URL from the previous step.

Use the Admin Access token to Login to the Admin Portal.



Create User Slice (Workspace) Workflow

Admin is responsible for setting up a Slice workspace for a User (or a team).

Ensure that you have User name, namespace, email ID before performing the following steps.

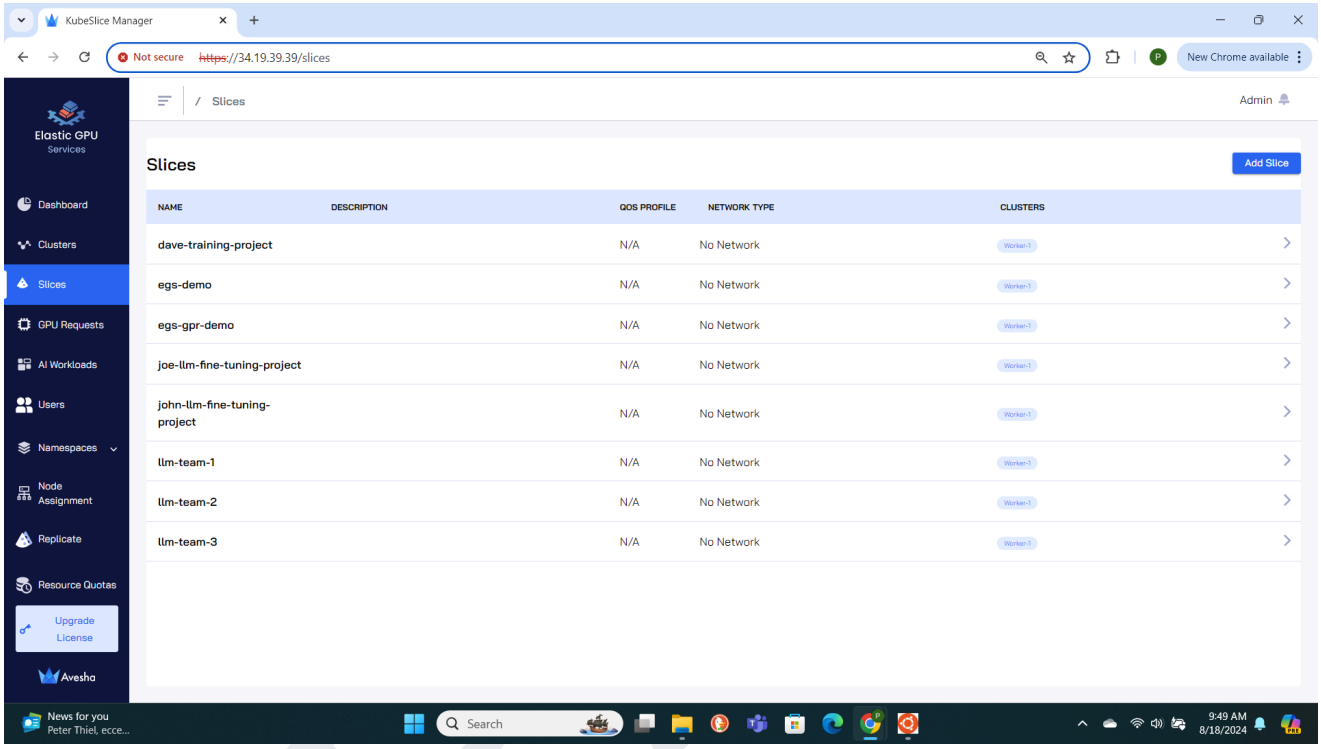
Perform the steps below to set up a User Slice (workspace).

1. Create a Slice for the User (with team name or user name)
2. Assign a namespace to the User Slice
3. Add User to the Slice
4. Provision Slice RBAC for User
    1. Add role and role bindings for the Slice namespace
5. Get UI Portal Access token for the User
    1. User can access the UI portal using the token
6. Download KubeConfig for the Slice
    1. Kubeconfig will have SA token to access the User namespace
7. Send the User UI portal Access token to User
8. Send the KubeConfig to User
    1. User can access the namespace on the cluster
    2. User RBAC scope is associated namespace

Create Slice for User

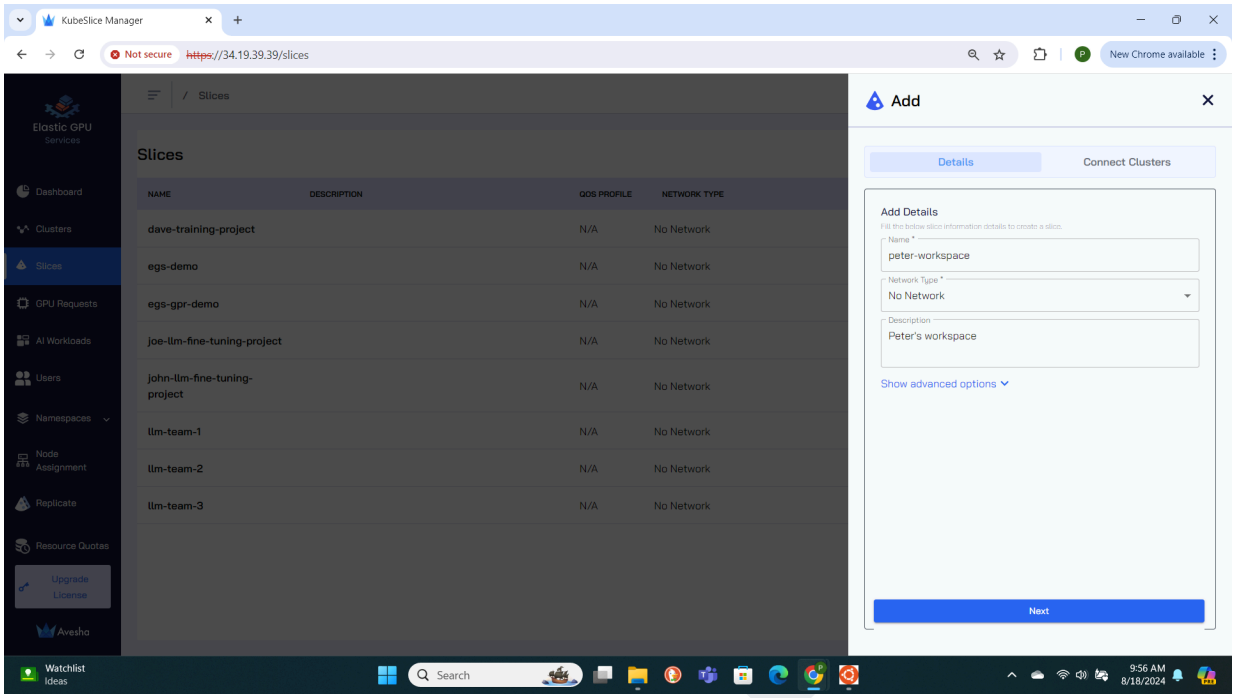
Perform the following steps to create a Slice.

Select Slices on the left panel.



Click on **Add Slice**

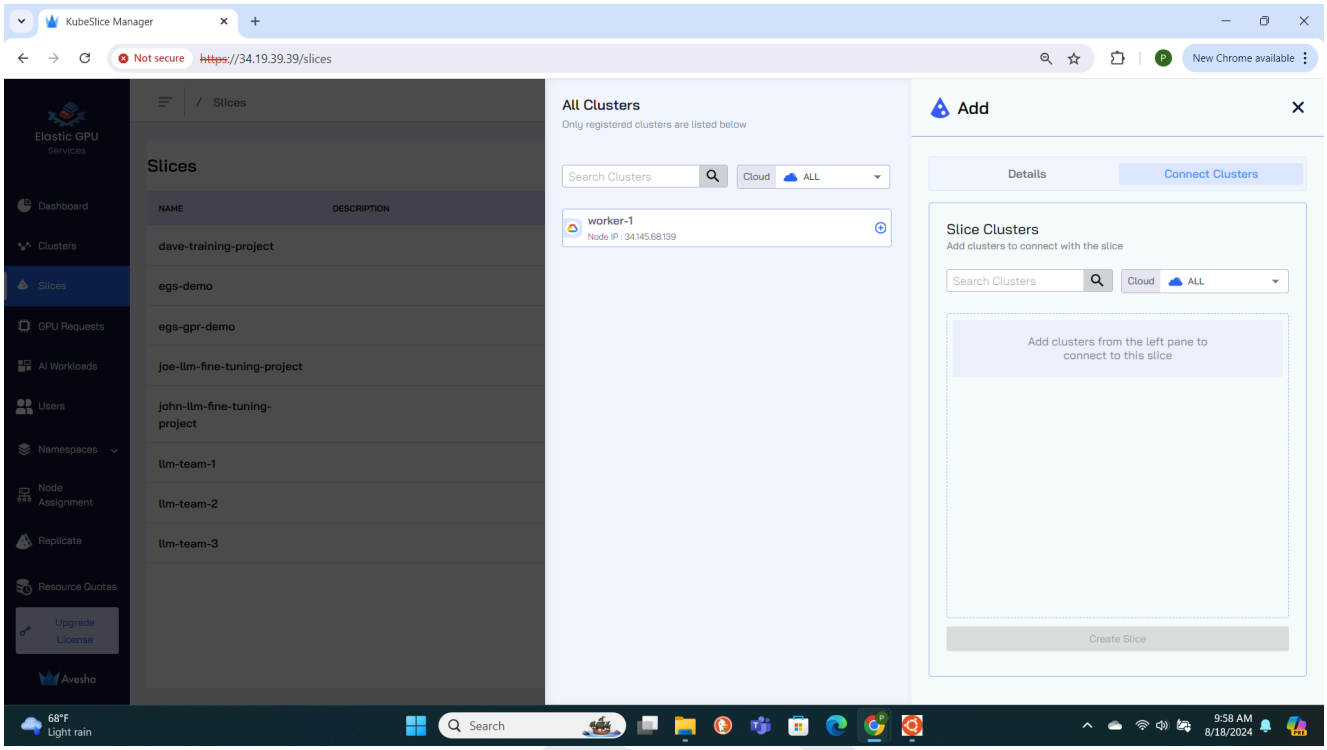
Elastic GPU Service



Add slice details:

Add User name: (example peter-workspace)
Select No-Network for Network Type
Click Next

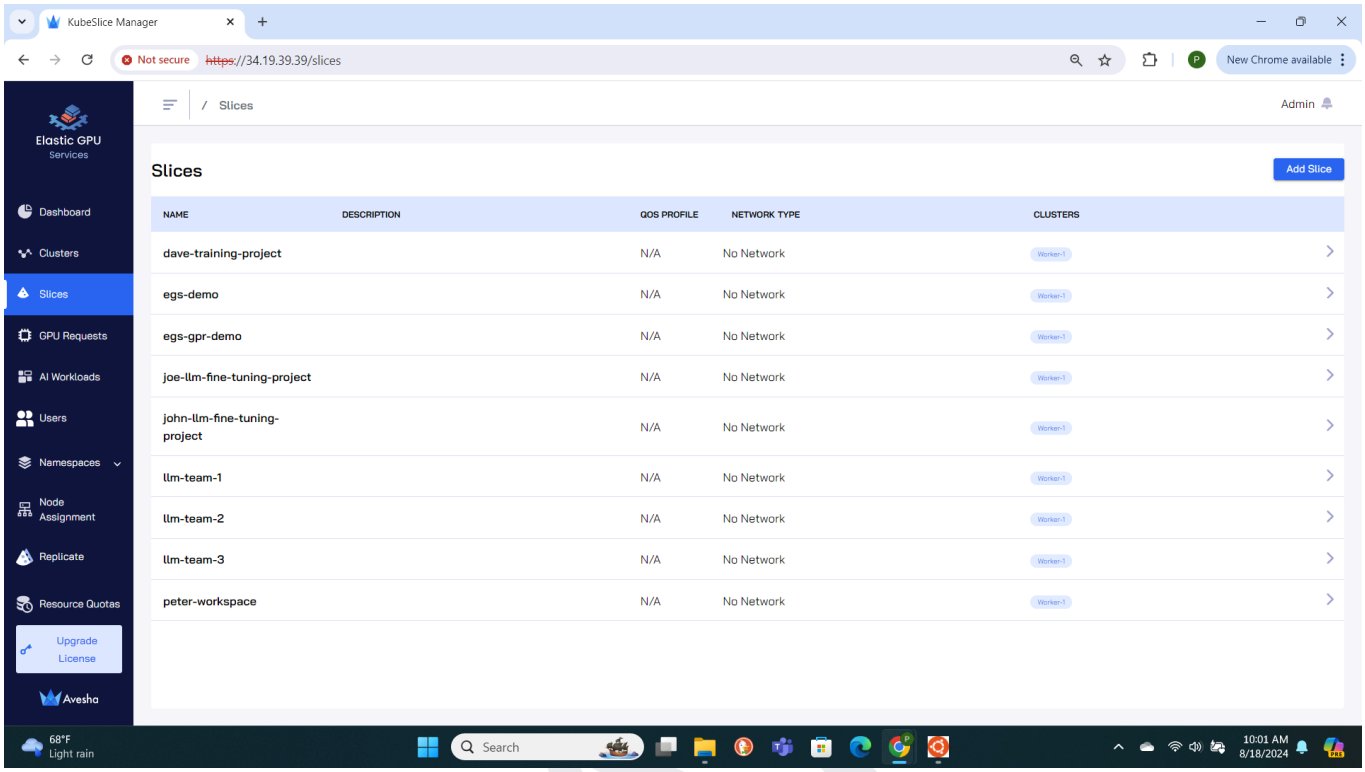
Elastic GPU Service



Add Cluster to the Slice

Click on + sign to add worker-1 cluster to the User Slice.
Click Create Slice.

You should see the newly created slice in the Slices list.

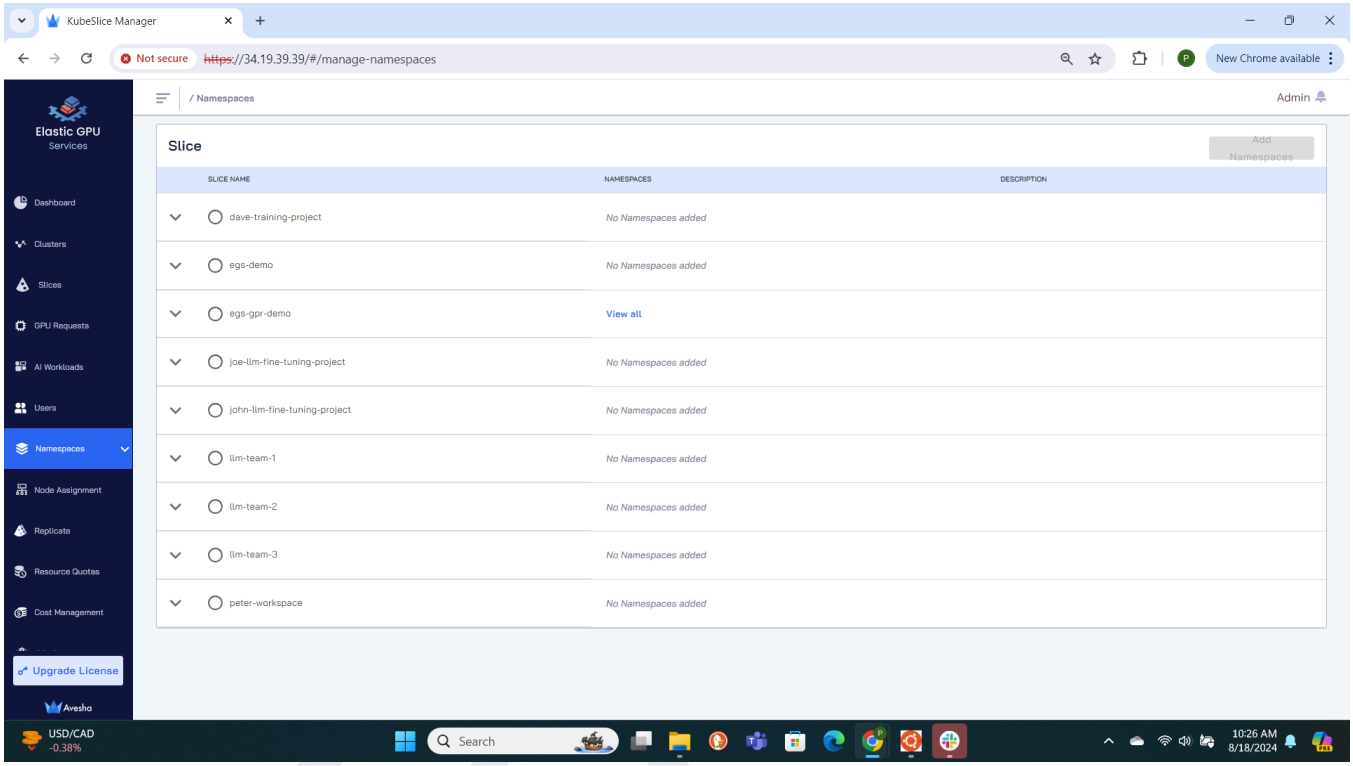


Assign a namespace to the User Slice

If the User has supplied the namespace for the workspace, use that namespace to associate with the User's Slice. Otherwise, create a new namespace for the user.

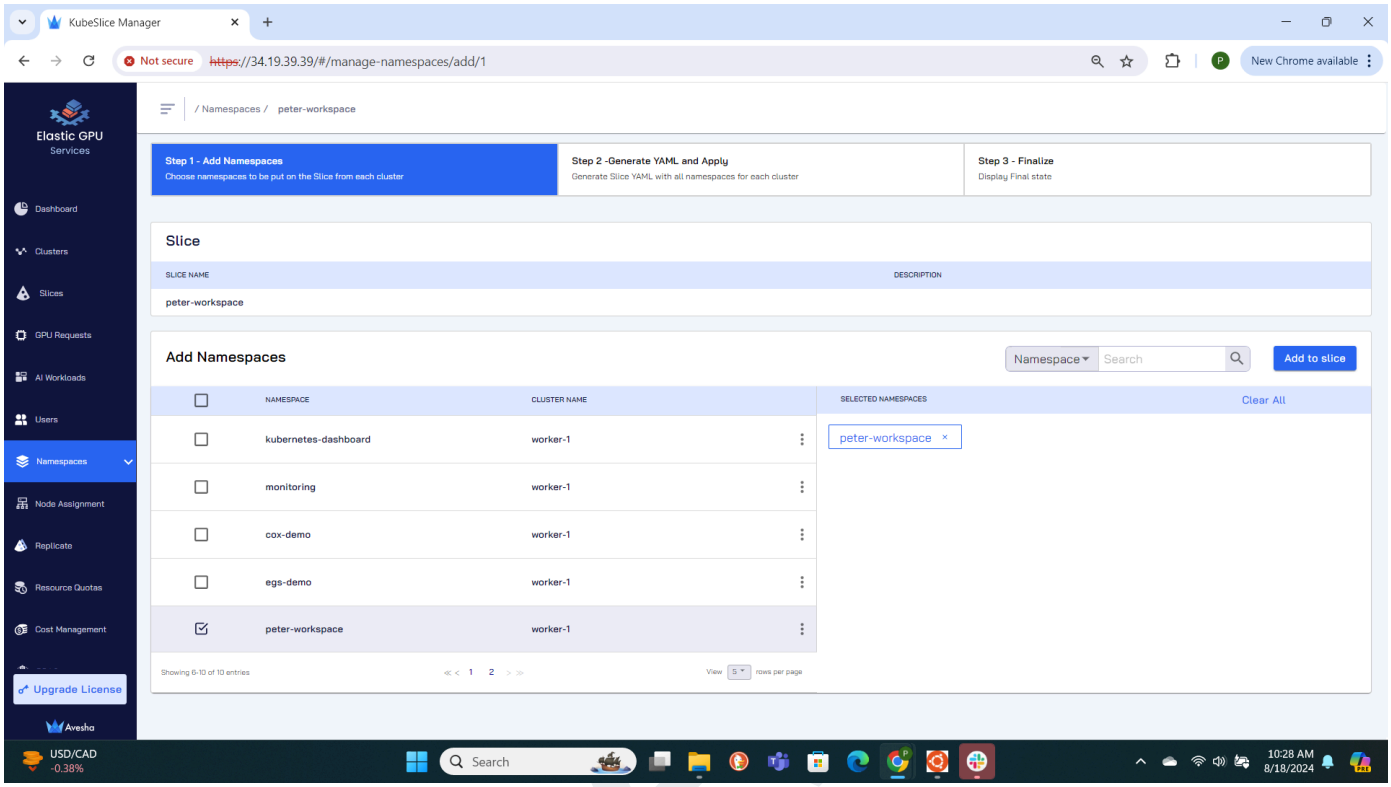
```
Unset
/home/user/egs-installation$ kubectl create ns peter-workspace
namespace/peter-workspace created

/home/user/egs-installation$ k get ns peter-workspace
NAME              STATUS   AGE
peter-workspace   Active   8s
```

Select namespaces on the left panel



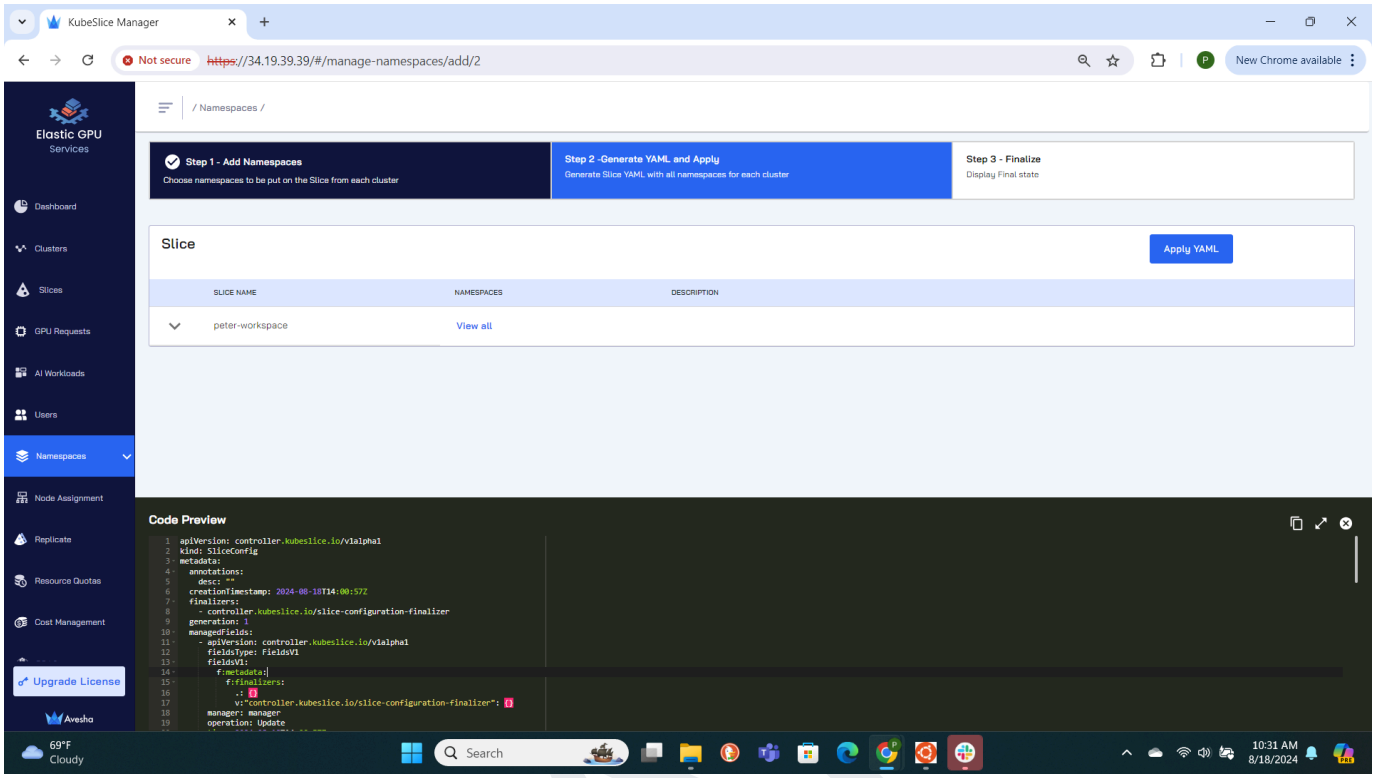
1. **Select the User Slice**
2. **Click on Add Namespaces**
3. **Select the user-workspace namespace**
4. **Click on Add to Slice**

Elastic GPU Service



1. **Click on Apply YAML**

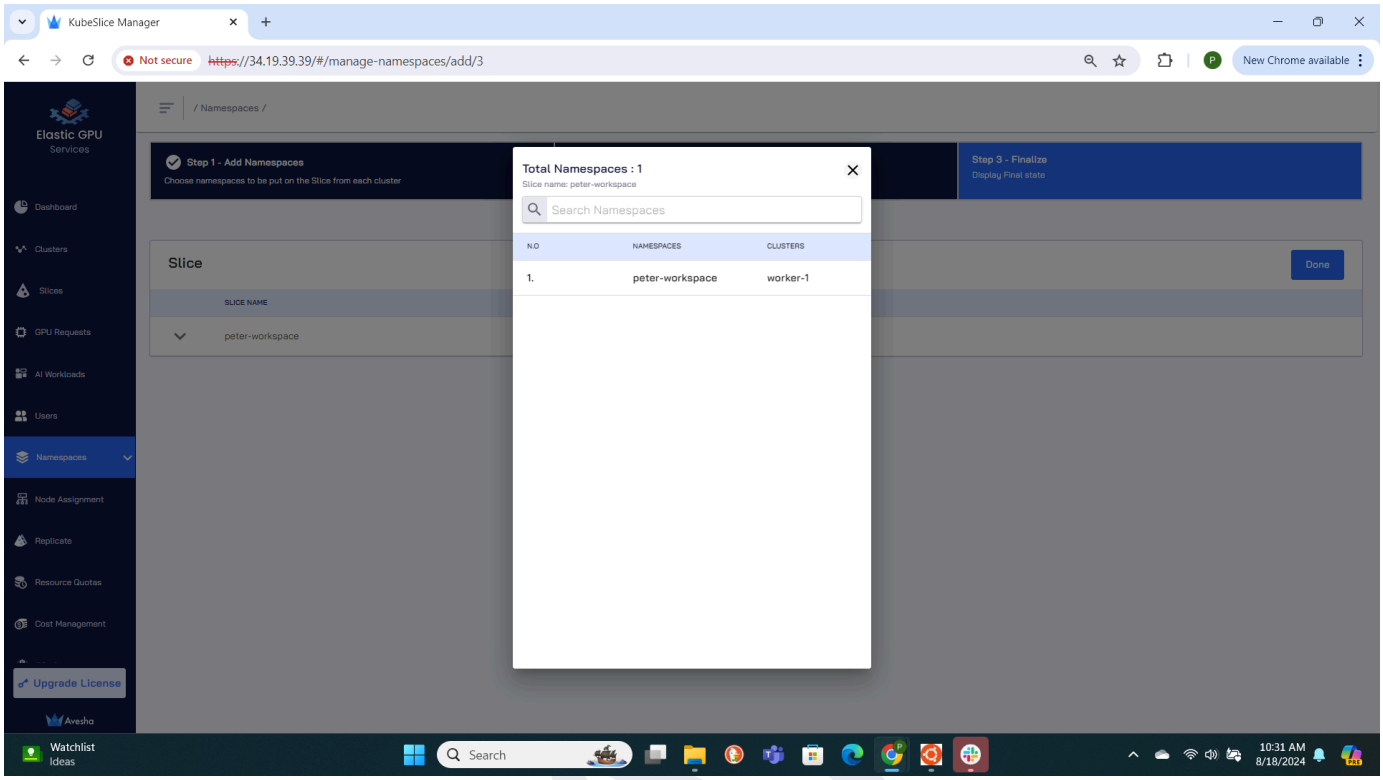
Elastic GPU Service



View namespaces

Select Slice

Click view_all on the Slice bar to see the namespaces associated with User Slice.



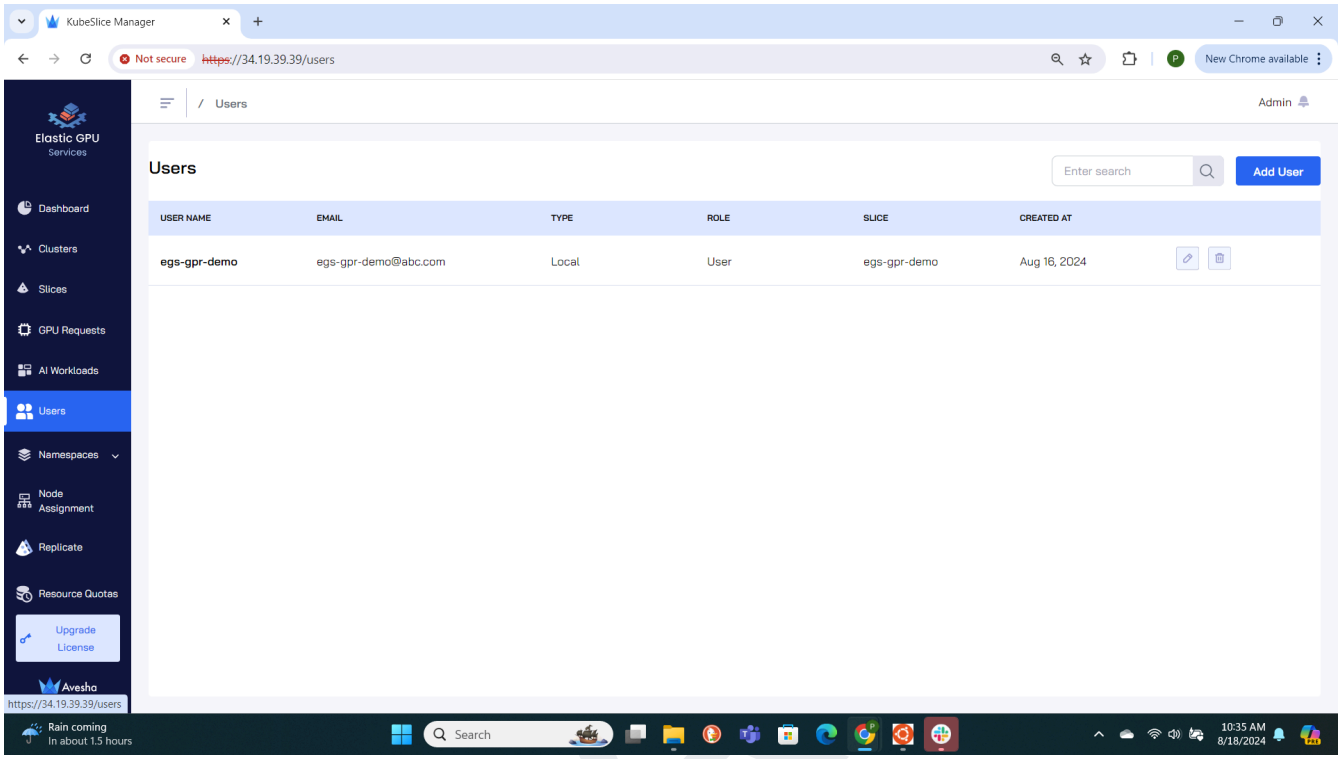
Add User to the Slice

Admin can add, remove, edit user details in the Users panel.

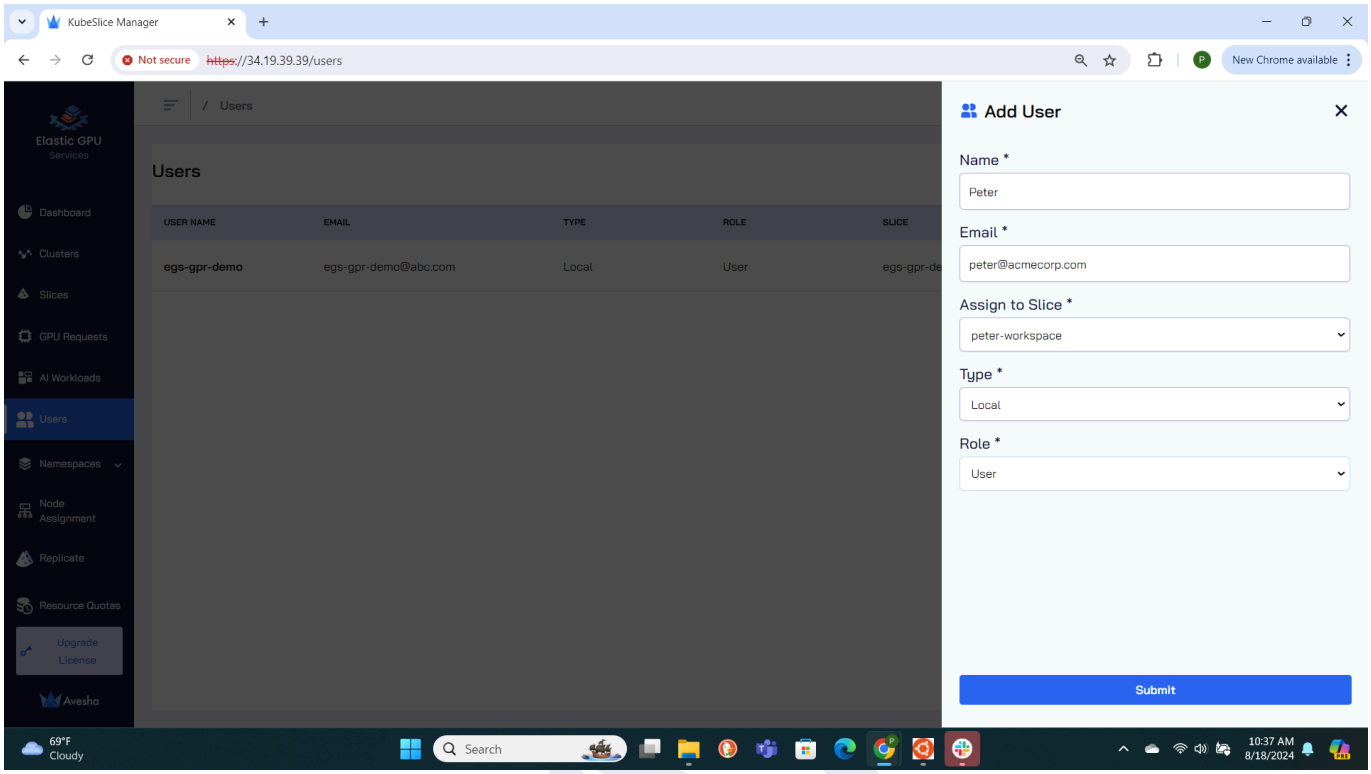
To add a User to the Slice perform the following steps.

Elastic GPU Service

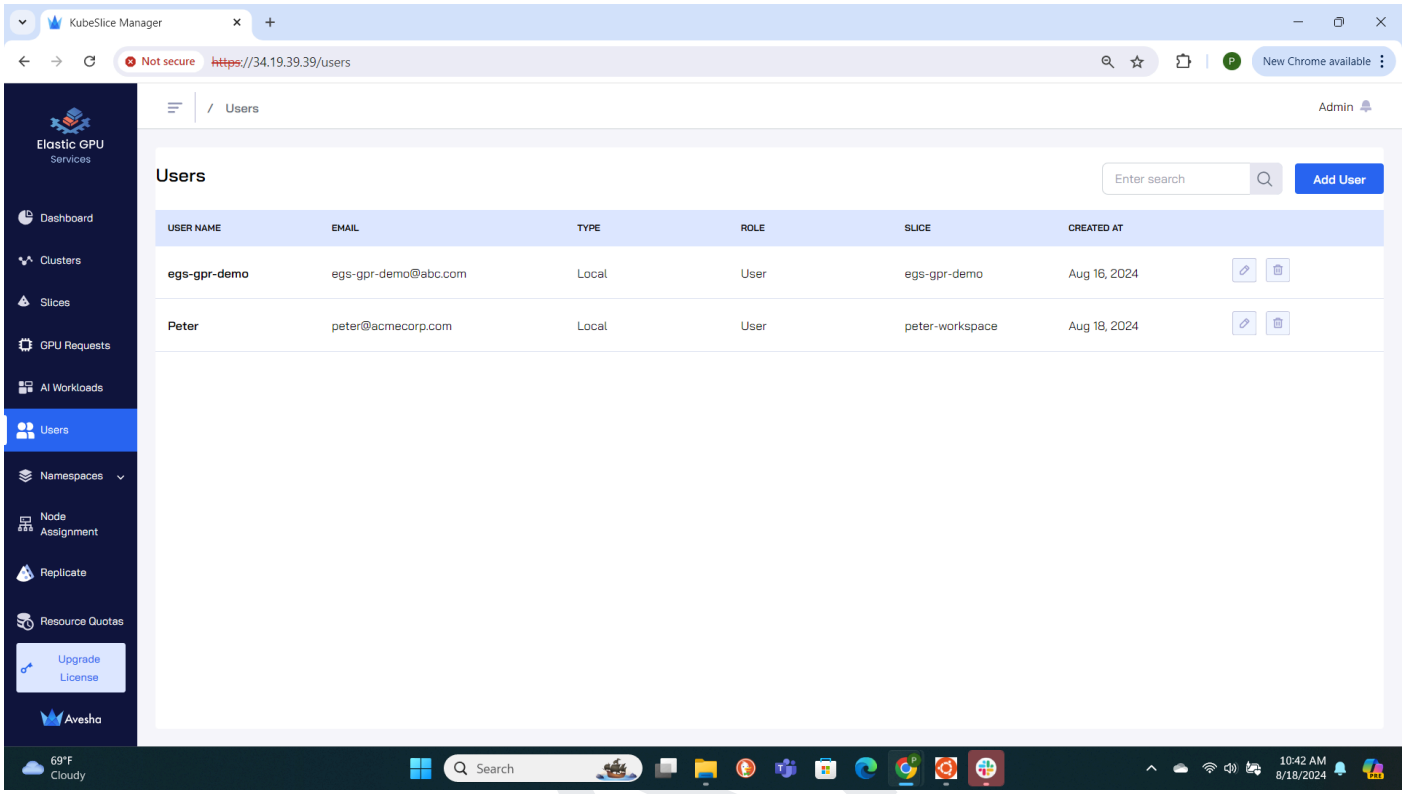
Select **Users** on the left panel.



1. Click on **Add User**
2. Add user details
    a. Name
    b. Email-id
    c. Select the slice name
    d. Select type “local” (or “Idp group” - will be supported soon)
    e. Select Role “User”
    f. Click **“submit”**



You should see added User details in the Users list.



User Access token for UI Access

Users need an Access token to access the UI portal.
Perform the following steps on the cluster to generate the access token.

```
Unset
%./fetch_egs_slice_token.sh -h
%./fetch_egs_slice_token.sh -k <kubeconfig> -s <slice name> -p avesha

Note: default project name is "avesha" - change for different projects
```

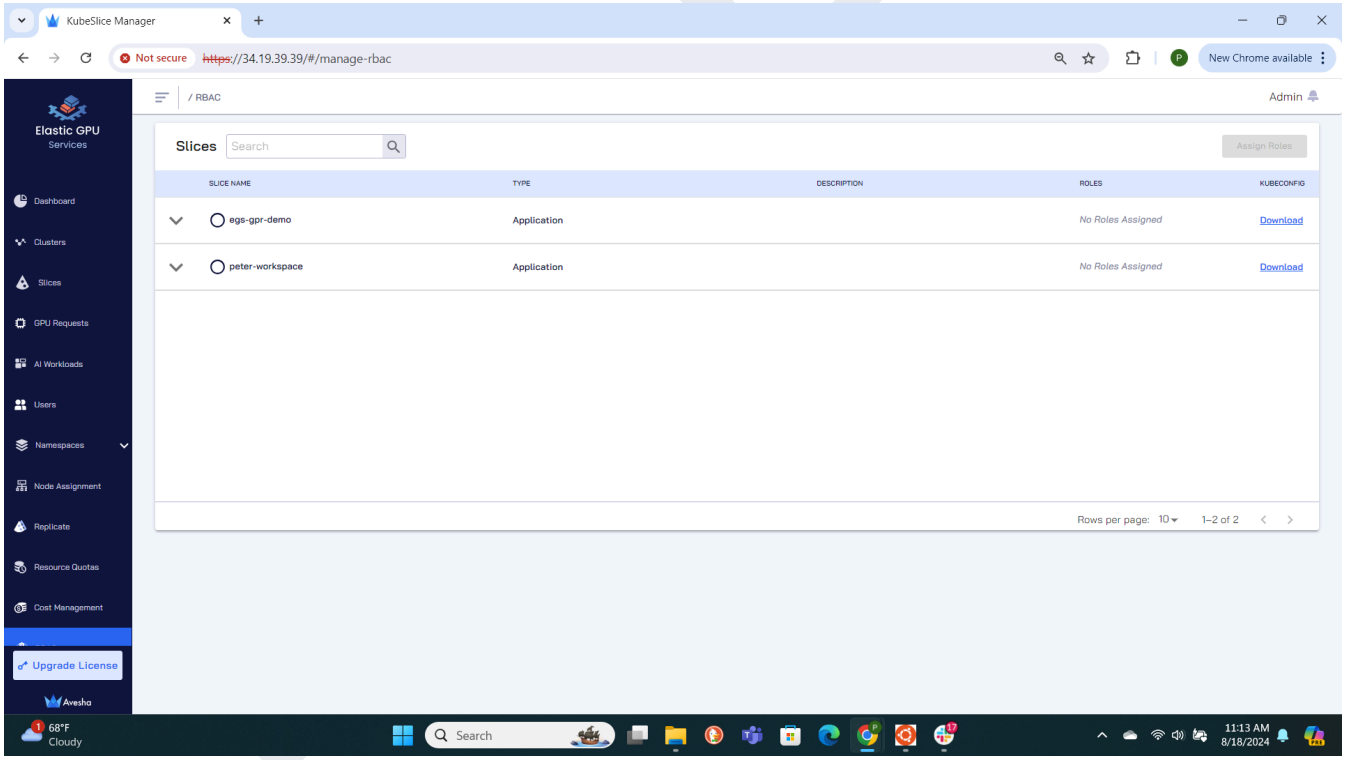
Send the EGS UI portal (ui-proxy) URL and the token to User to access the User portal.

Provision Slice RBAC for User

Perform the following steps to create an RBAC role for User - to restrict User access to worker cluster user namespace(s).

RBAC for slice

Select **RBAC** on the left panel
Select User Slice
Click **Assign Roles**

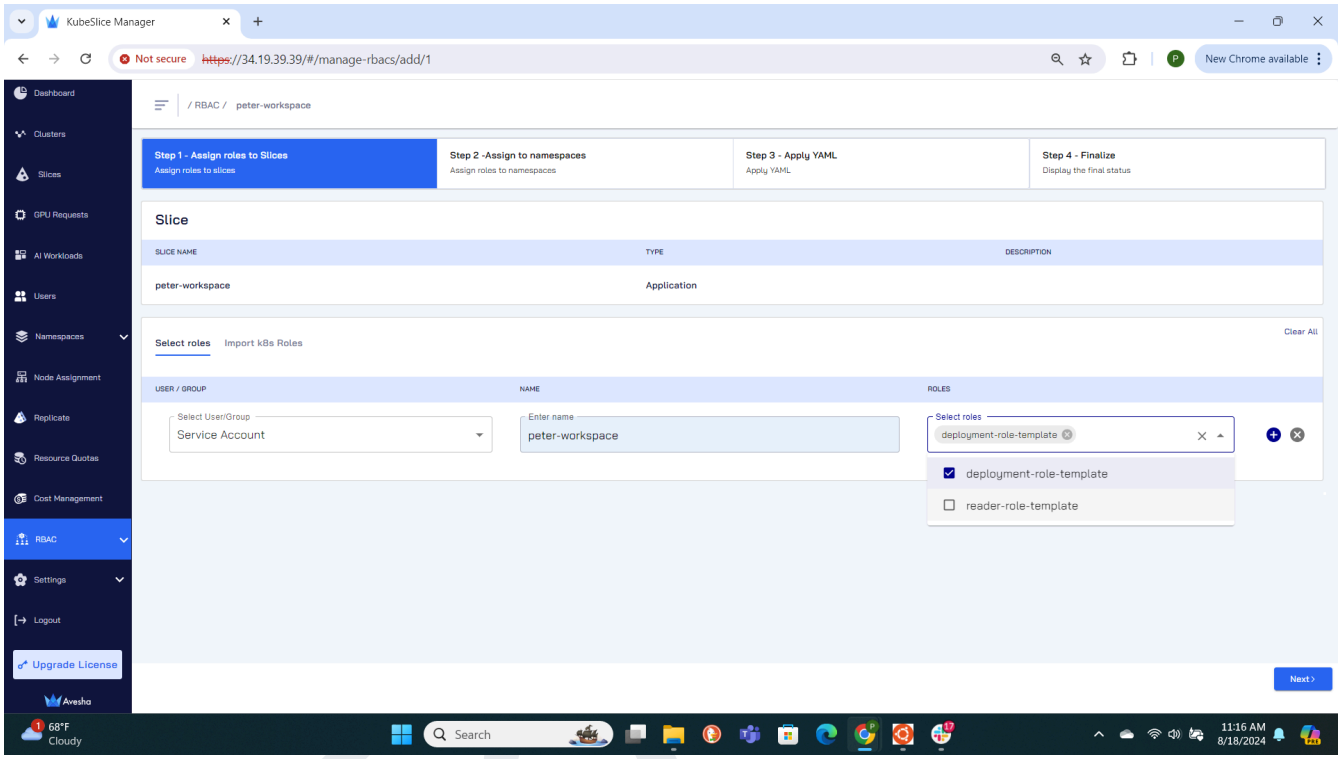


Add Role details

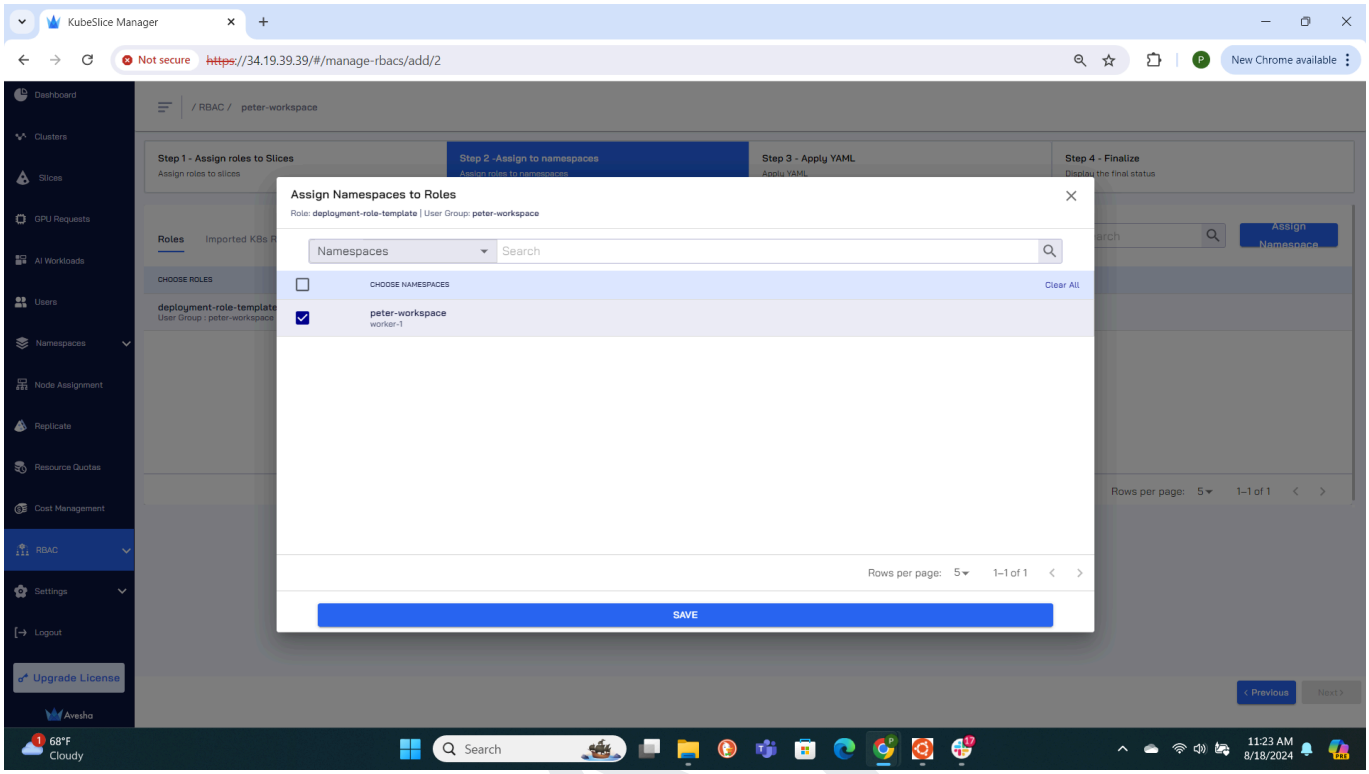
Select “Service Account” for User/Group
Enter name
For “Select roles” - pick deployment-role-template
Click **Next**

Note: Admin can edit/modify the default role templates, add new roles and use them.

For more details on the RBAC and roles - please see the RBAC documentation.

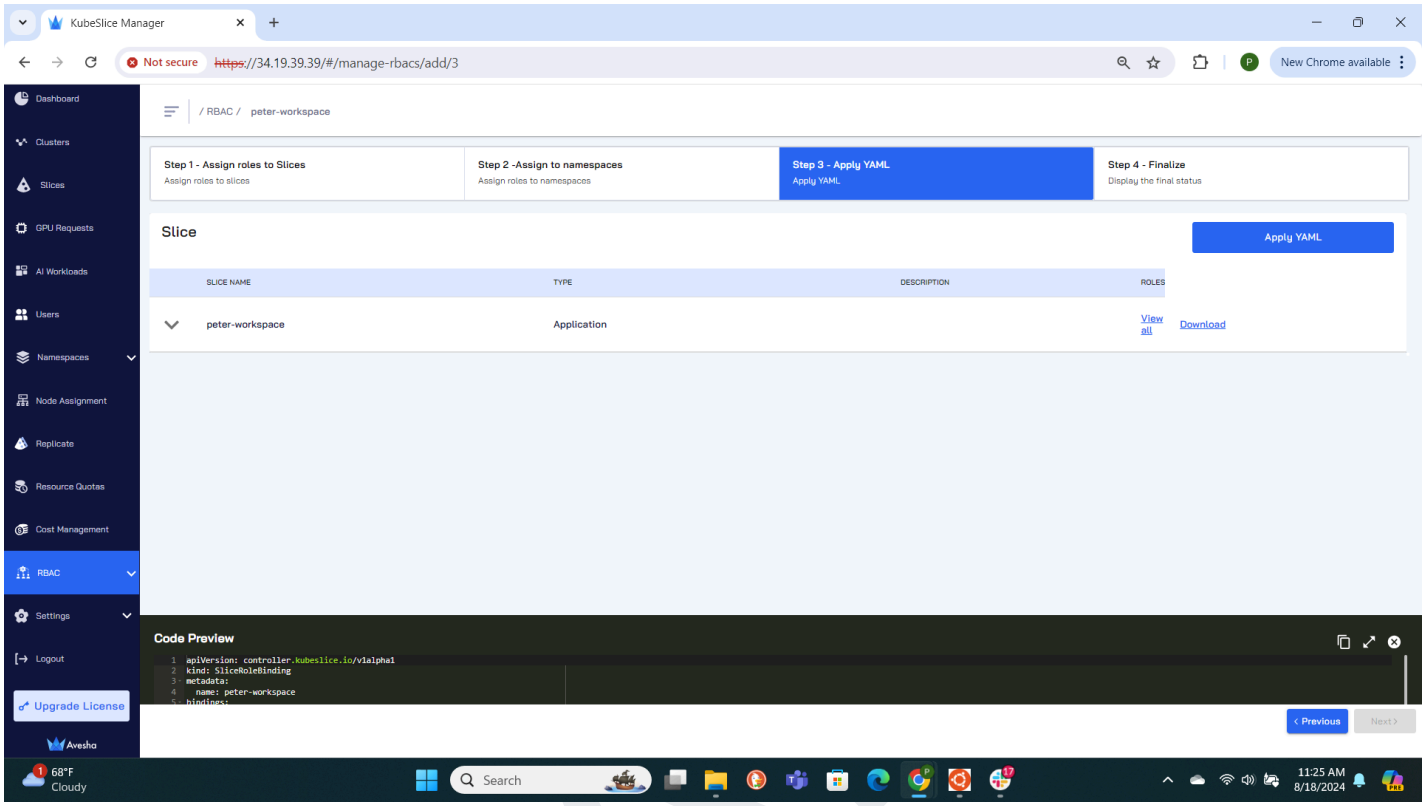


Select role from the list
Click on Assign Namespace
Select namespace
Click **Save**
Click **Next**



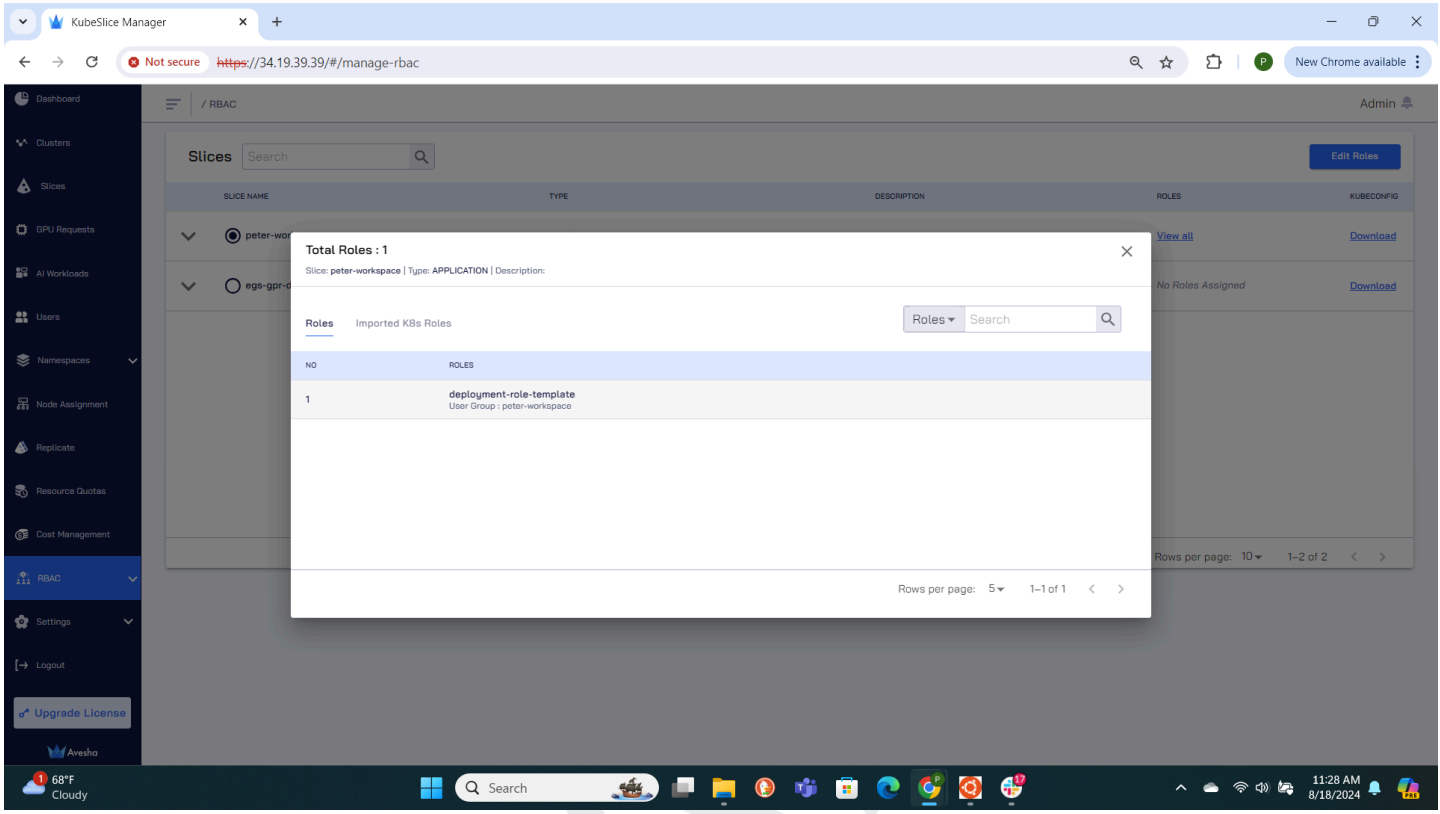
Click on **Apply YAML**
Click **Done**

Elastic GPU Service



On Slice bar click on **view_all** to see the roles for the Slice.

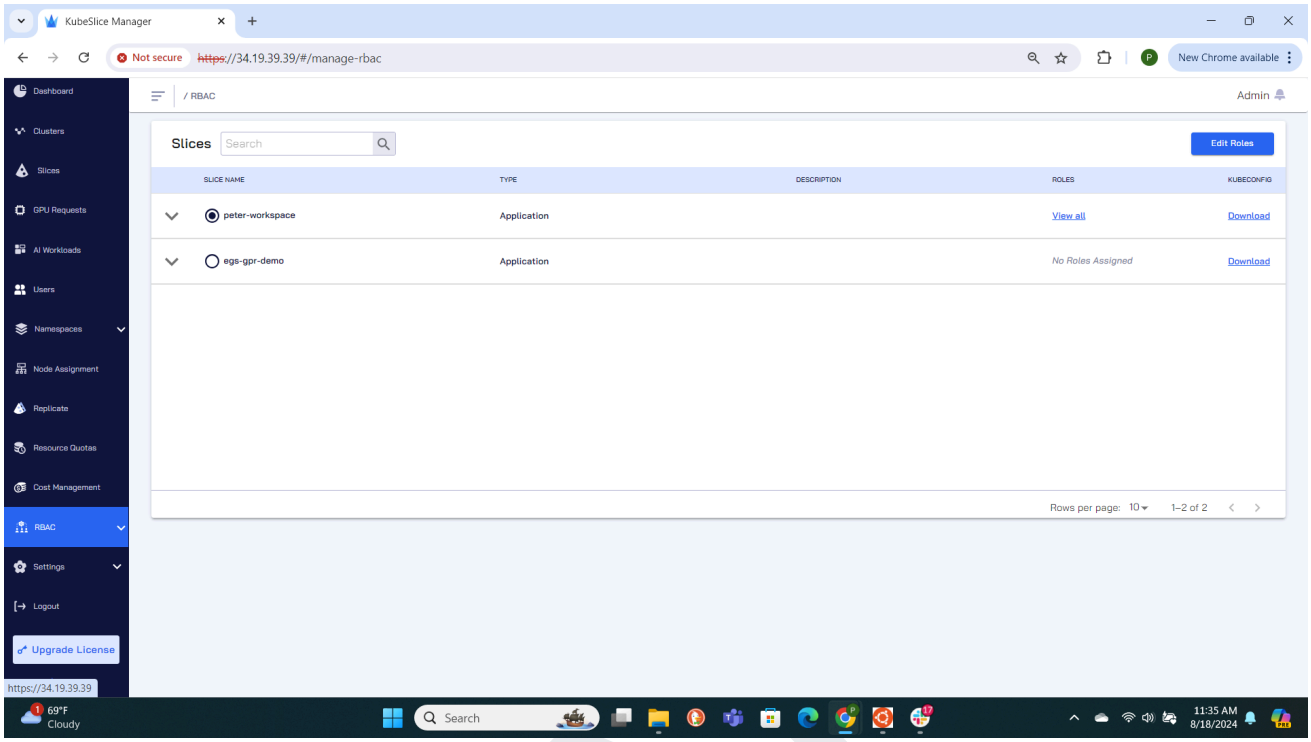
Click on the Download button to download the Kubeconfig file for worker cluster with User specific RBAC.



Download KubeConfig for User

Admin can download the User RBAC Kubeconfig yaml file from the RBAC panel.

Select RBAC on the left panel.
Select the User Slice.
Click on **Download** button - to download the User namespace scoped Kubeconfig for worker cluster.
Note: User Slice RBAC role needs to be provisioned to download the kubeconfig yaml file.



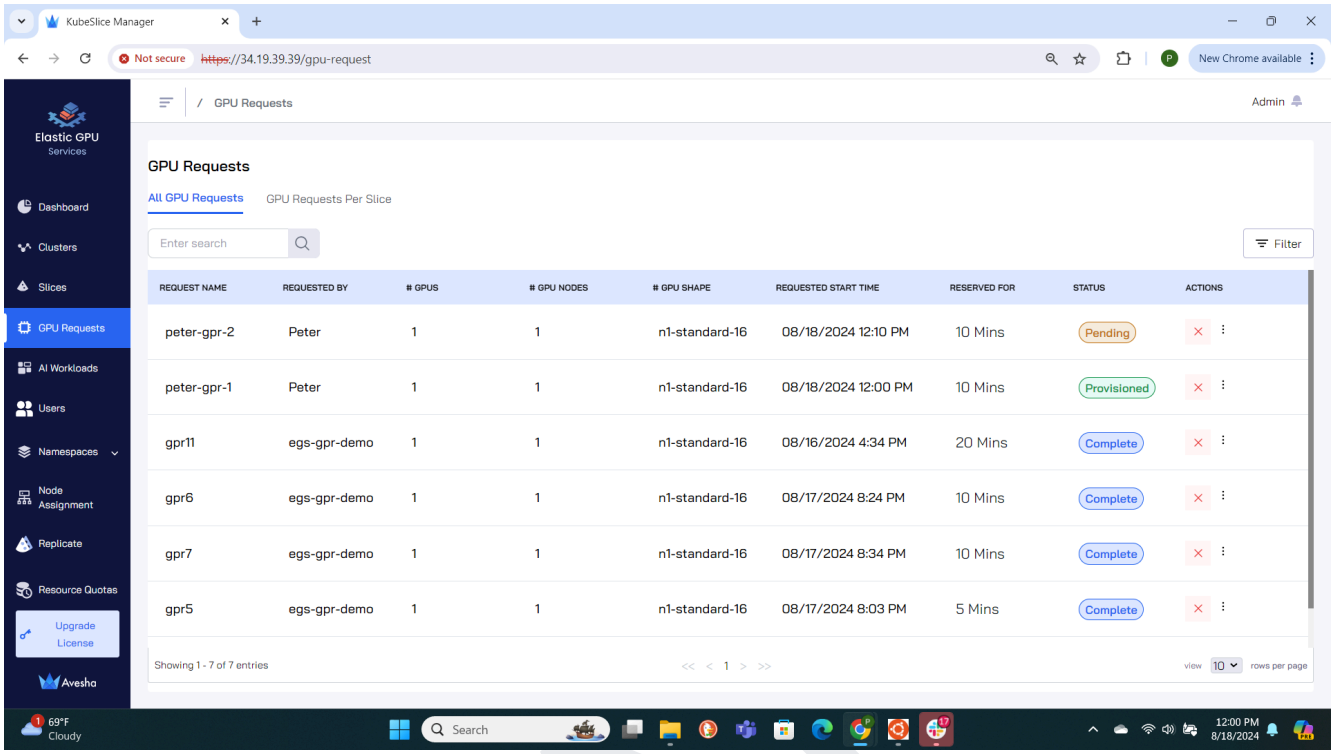
Manage GPR queue

Admin can view and manage the User GPU provision requests (GPRs) queue.

View GPRs across all the Users

Select **GPU requests** on left panel

All GPU Requests tab show all the GPU requests across all the Slices.
Admin can use **search or Filters** to filter the GPRs.



Admin can view the GPRs specific to a Slice.dd

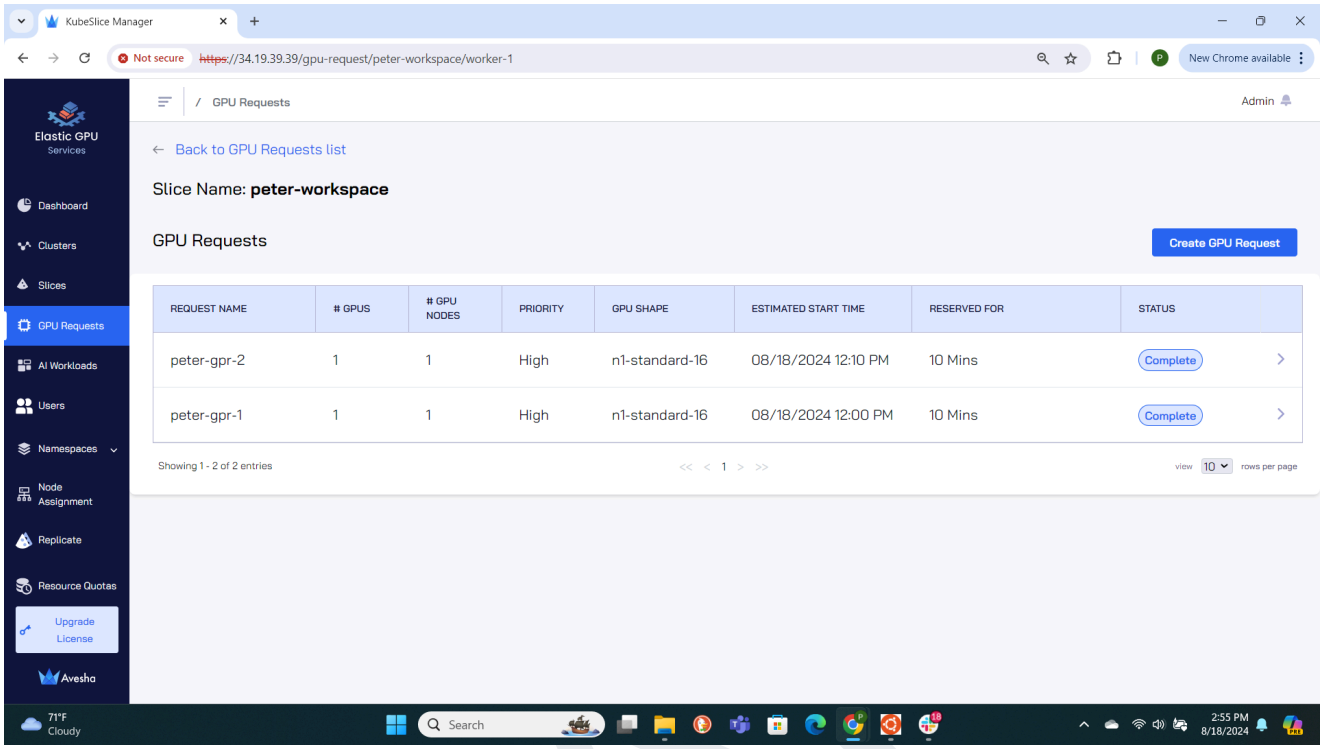
Select “**GPU Requests Per Slice**” tab and click on the Slice to see the GPRs specific to the Slice.

GPR request for a User

Admins can create GPRs on behalf of the Users. Perform the following steps to create a GPR.

Select the User Slice in “**GPU Requests Per Slice**” tab.
Select the User Slice.
Click on **Create GPU Request**

Elastic GPU Service



In the “Create GPU Request” page - fill in the request details.

Enter GPR name, Number of GPU nodes
Select GPU shape
Select priority
Select Reserve for duration

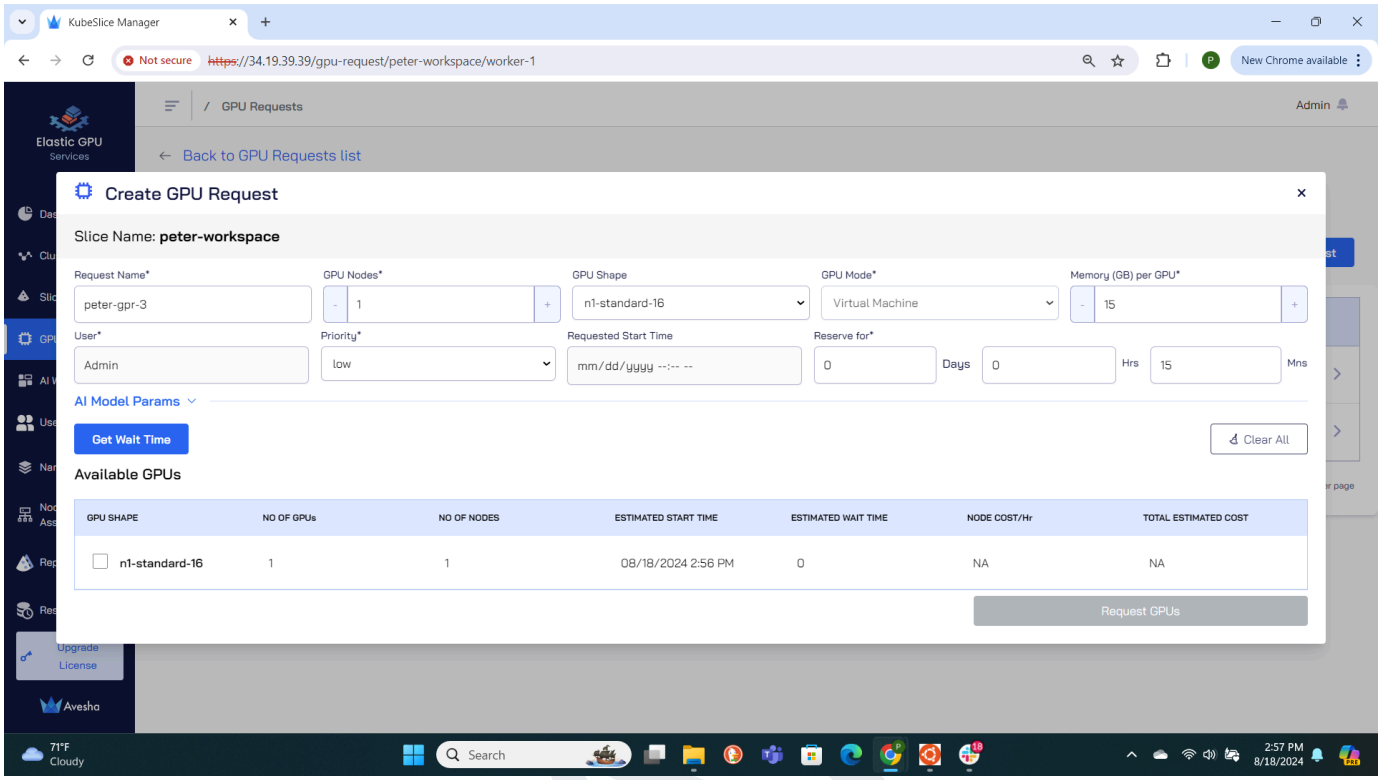
Click “**Get Wait Time**” button

EGS shows the estimated wait time for the GPU nodes provisioning.

Select the GPU in the “Available GPUs” table with acceptable estimated wait time.

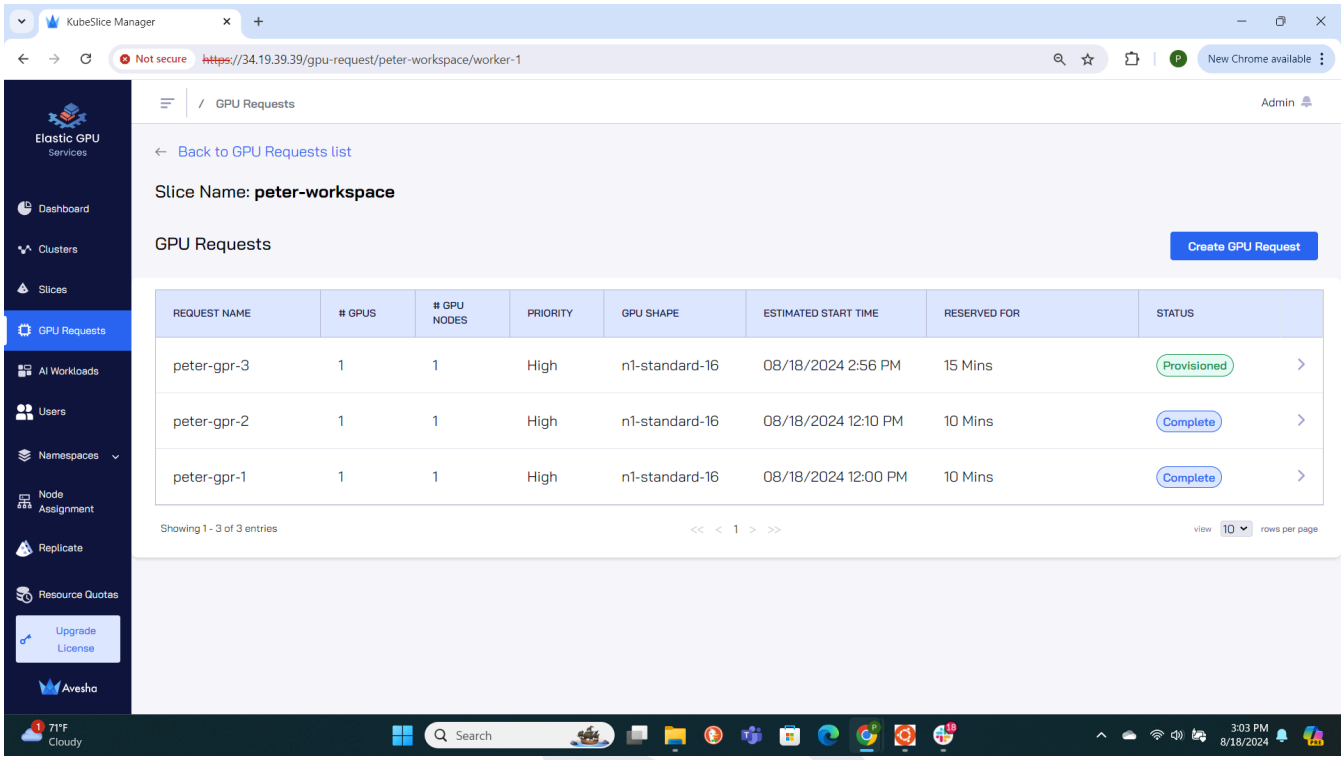
Click **“Request GPUs”**

Elastic GPU Service

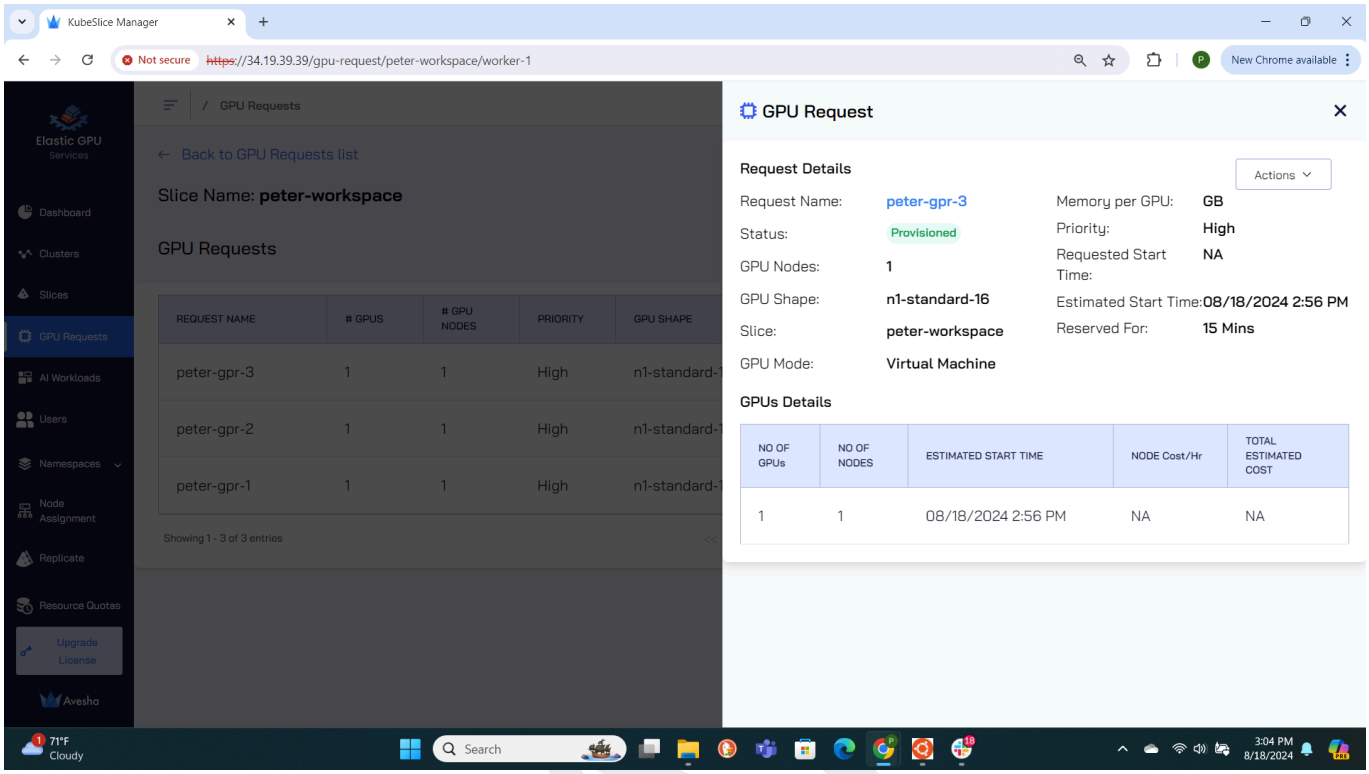


View the GPR in the User's GPU requests queue.

Elastic GPU Service



Click on the GPR to view request details.



Additional operations

Admin can perform the following operations on the GPR queue.

GPR tables, queues view and actions

1. Admin can view the GPR queue
2. Admin can change the priority of GPR and move a GPR up/down the queue
3. Admin can edit, delete, early-release a GPR
4. GPR eviction

Adjust GPR priority

Admin can adjust the priority of a GPR in the queue. Admin can select a GPR and increase the priority number (low 0-100, med: 100-200, high: 200 and above) to move a GPR higher in the queue. When a GPR is moved to the top of the queue it will be provisioned when the resources are available to provision the GPR.

GPR eviction
Admin can early release provisioned GPRs and make needed nodes available for the high priority top GPR to be provisioned.

EGS shows Admin a list of GPRs that needs to be evicted to provision the top GPR. Admin can manually early-release the GPRs to make room for the top GPR.

Approve GPRs in pending-approval state

Admin will approve the GPRs that are in pending approval state. A GPR will be in pending approval state for any of the following conditions:

1. Over the quota for the Slice
2. Over the GPU nodes limit per GPR (request for large number of GPU nodes)
3. Note: Most GPRs are auto-approved

Admin will use the Actions pull down menu to Approve/Deny the GPR.

Early release a provisioned GPR

Admin can early-release a provisioned GPR. Early release of a GPR will remove the associated GPU nodes from the Slice VPC.

Admin can use this workflow to free up GPUs to provision a higher priority GPR. Admin can use this workflow for any other admin reasons or under utilization of GPU resources, User/Manager request etc.

Note: Adjust priority and GPR eviction workflows will be available in the next release.

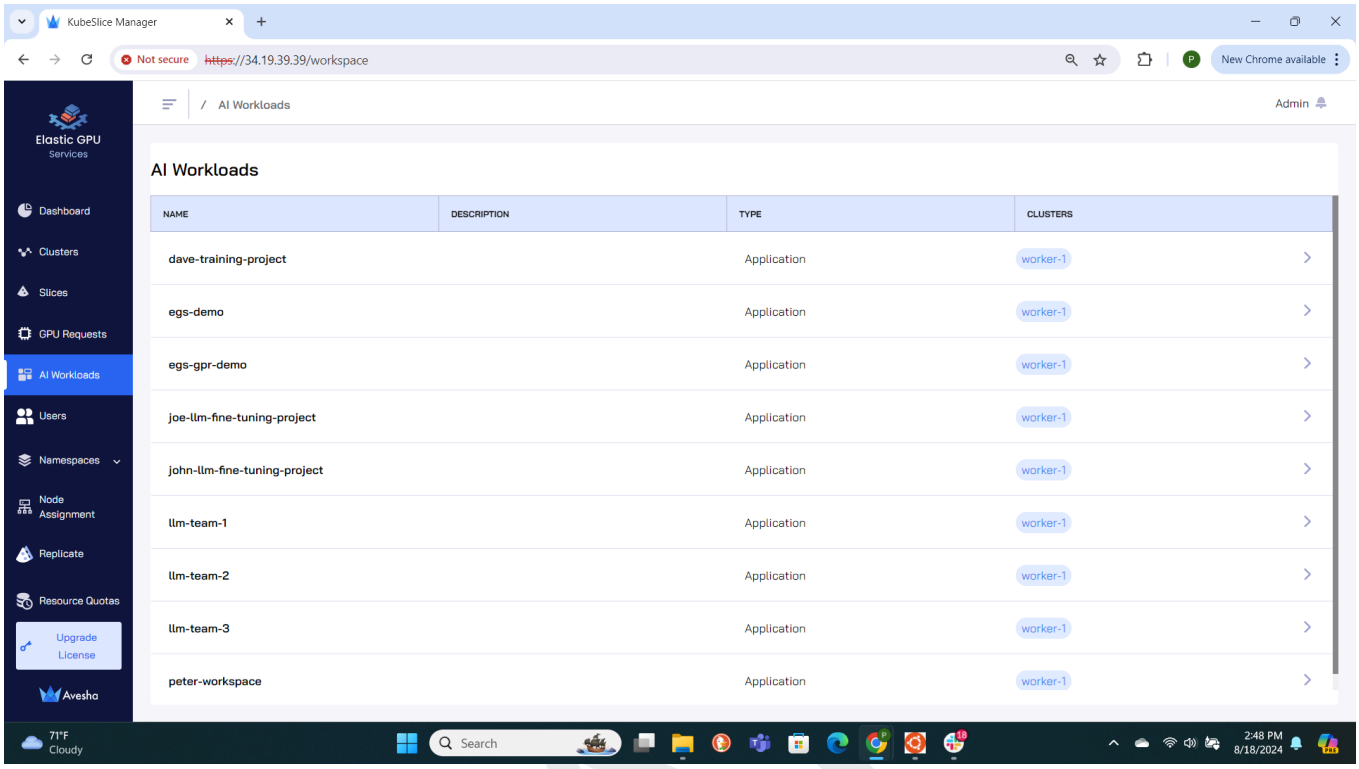
View AI workloads

Admin can view all Slice workspaces AI workloads.

Select **AI Workloads** from the left panel.

Select User Slice to see the AI workloads for the slice.

Elastic GPU Service

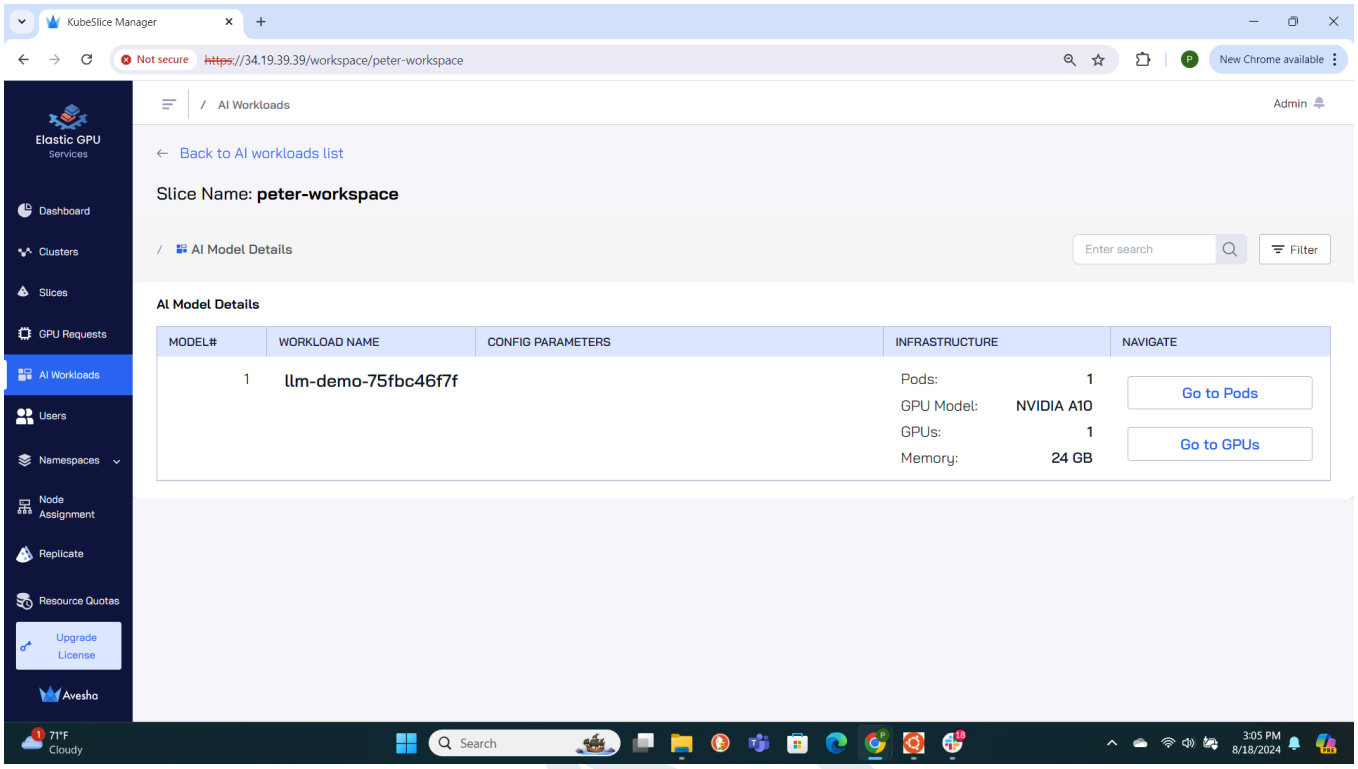


Model details

Select User Slice to see AI workloads for User workspace

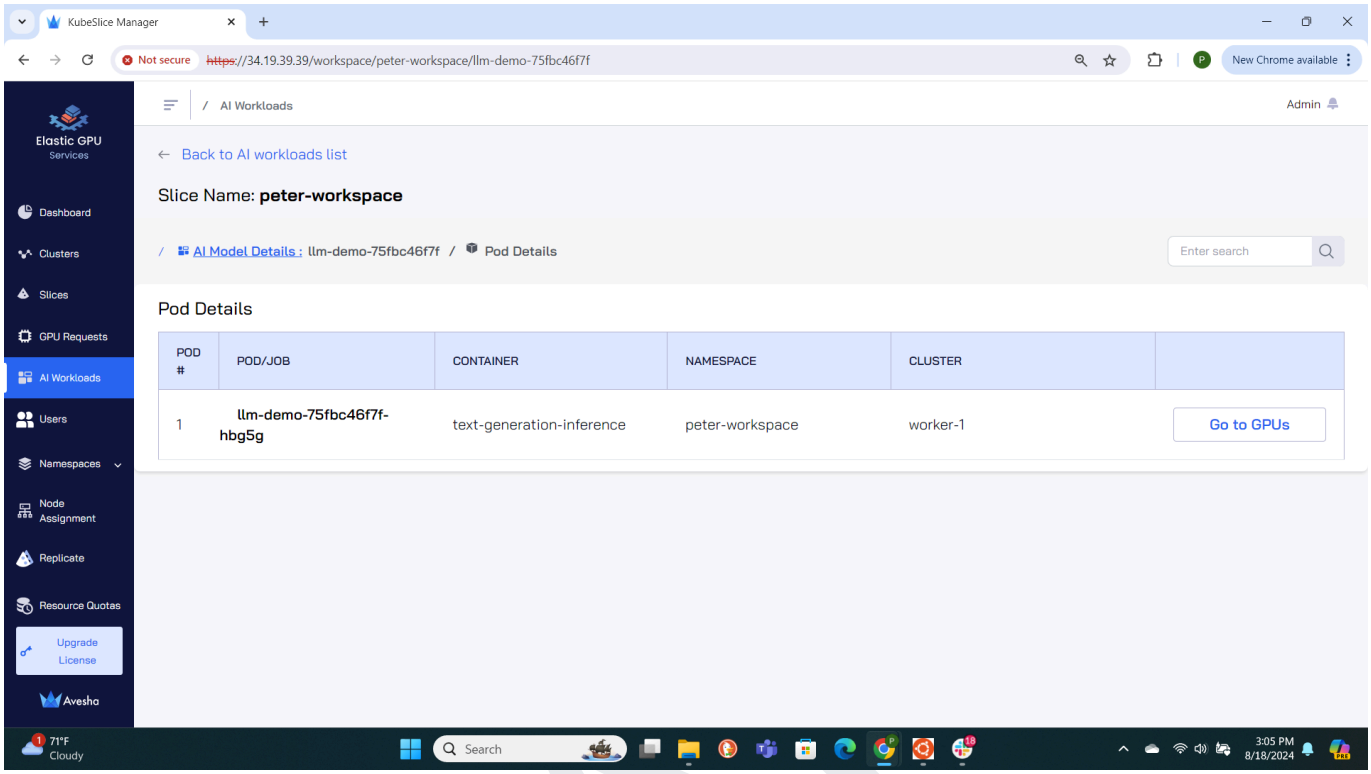
Shows the model details, GPU infrastructure committed to the workload.
Show model summary - high power GPU, high temp GPU and Average Utilization values.

Elastic GPU Service



View Pods

Click on **Go to Pods** to see the pods running with GPUs in the workspace.

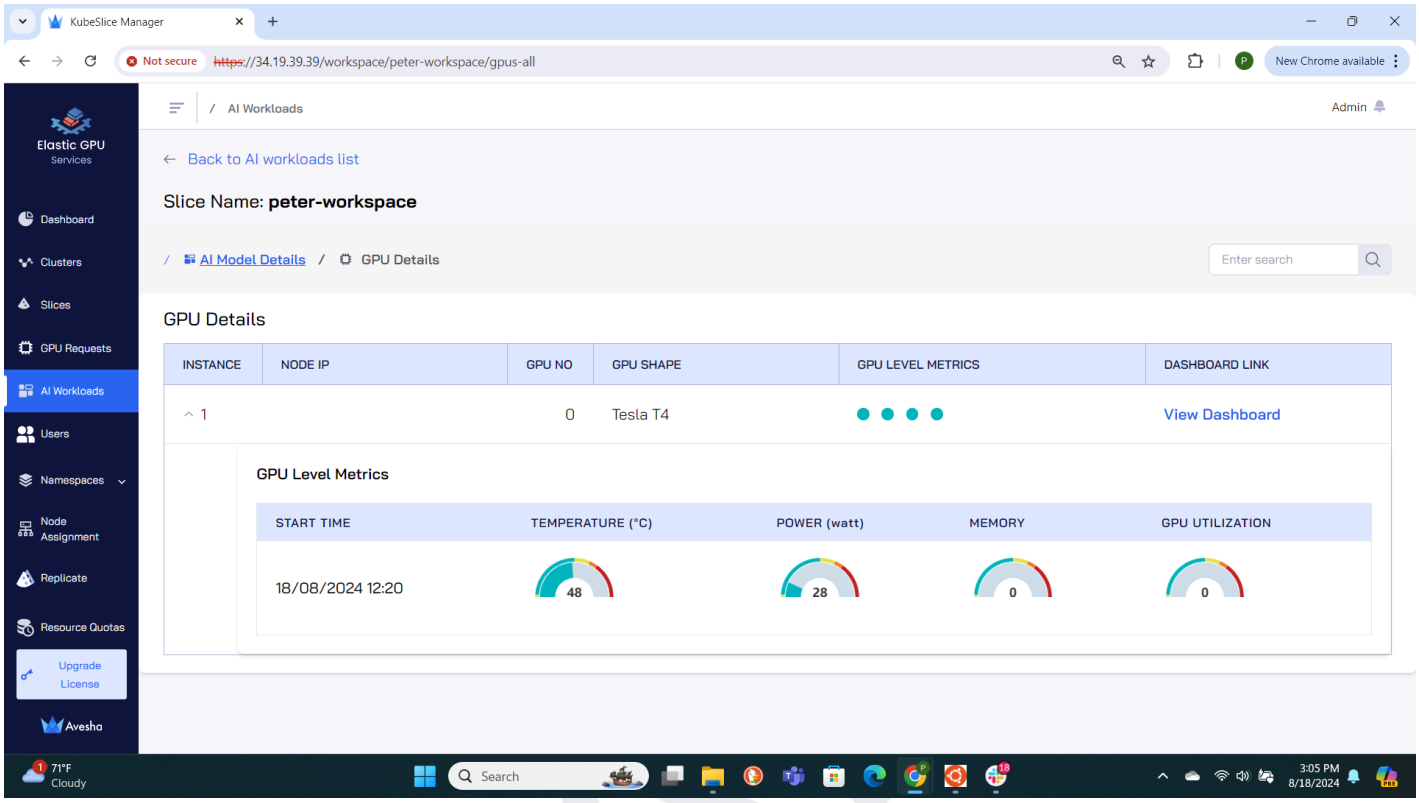


View GPUs

From AI Model Details page Click on “Go to GPUs” to see GPU table page
Shows sorted list of GPUs with high power, temperature GPUs at the top for quick access.
Shows the hotspot GPUs.

Click on “View Dashboard” to view time-series data for the selected GPU device.

Elastic GPU Service



Alerts and Events

Create Project

During installation a default project workspace will be created. A single project namespace can be used to manage one or more clusters across one or more users/teams. In a project one or more slices can be created. A user can be associated with one or more Slices. Each slice provides a workspace for a user or a team. Additional projects can be created to provide multi-tenancy across orgs or large departments - where a pool of clusters are managed by the different departments.

For more details on how to create projects - please see <link>

Registers cluster(s)

During installation, performed using the installation script, for single cluster deployment, the script registers the worker cluster.

Admin can add additional clusters to the EGS control plane to manage the GPU resources on the clusters.

For more details on how to register a cluster please see <link>

Admin Dashboard

1. Use dashboard to monitor key metrics related to the GPU pool
    1. GPRs and GPU utilization, allocation, Errors, Alerts
2. Use dashboard to access Cluster kubernetes dashboard pages

Day-to-Day Operations

Admin can perform the following operation for managing the day-to-day operations to manage the EGS platform.

1. Admin can add/edit/delete User Slices (workspaces)
2. Admin can add/remove namespaces to Slice
3. Admin can add/remove/update the Slice RBAC and regenerate the KubeConfig for User - for cluster access.
4. Generate access tokens for Users for UI access
5. View GPRs queue across all the Users/Slices.
6. Edit/Delete GPRs - delete or early-release a provisioned GPR.
7. View AI workloads across all the Users/Slices.
    1. View GPU dashboards for AI workloads. Access time-series data for all workload GPUs.
    2. View pods, model details.
    3. Access top power/temp GPUs. Filter/search GPUs.
8. Register additional Clusters
    1. add/remove clusters
9. Create additional Projects
    1. add/remove projects

The following workflows will be available in the next release:

1. View admin Dashboards
    1. Error/Warning alerts, top GPUs, top Users, top utilization, etc. Adjust GPR priority, GPR eviction
2. Inventory schedule table views
3. Approve pending GPRs
4. View GPR pre-provisioning checklist/verification (visibility into the GPR pre-checks and logs)
    1. pre-provision the nodes and run checks
        i. Node health check, Node PV/PVC check
        ii. Nodes network check - RDMA subnet check, NCCL latency check
        iii. Nodes connectivity check
5. View GPR post-exit operations (visibility into GPR post-checks and logs)
    1. Node draining, Network check, NCCL test checks
    2. PVC/PV check, Labels check