

# 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

### 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.

A 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 Workspace(s).

The GPU provision requests (GPRs) can be created and managed using EGS APIs or YAML/GPR custom resources by User's CI/CD or RAG pipelines or an external system/service or an application service in the cluster as well. EGS User specific UI portal provides 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](#)
- For User guide please see the [documentation on website](#)
- For Installation guide please see the documentation on [github repo](#)

# Table of contents

<b>Introduction.....</b>	<b>1</b>
EGS Documents.....	1
<b>Table of contents.....</b>	<b>2</b>
<b>Installation.....</b>	<b>4</b>
<b>Admin Access Token.....</b>	<b>4</b>
<b>Access EGS UI Portal.....</b>	<b>5</b>
<b>Login to the Admin Portal.....</b>	<b>6</b>
<b>Create User Slice (Workspace) Workflow.....</b>	<b>7</b>
Create Slice for User.....	8
Assign a Namespace to the User Slice.....	12
Select Namespaces on the Left Panel.....	12
View Namespaces.....	15
Add User to the Slice.....	16
Provision Slice RBAC for User.....	19
RBAC for Slice.....	19
Add Role Details.....	19
User Access Token for UI Access.....	22
Download KubeConfig for User.....	22
Share the Token with the User.....	22
<b>Manage GPR Queue.....</b>	<b>24</b>
View GPRs across all the Users.....	24
View the GPRs Specific to a Slice.....	24
GPR Request for a User.....	25
Additional Operations.....	29
GPR tables, Queues View and Actions.....	29
Adjust GPR Priority.....	29
Approve GPRs in Pending-Approval State.....	29
Early Release a Provisioned GPR.....	30
<b>View AI Workloads.....</b>	<b>31</b>
Model Details.....	31
View Pods.....	32
View GPUs.....	33
<b>Alerts and Events.....</b>	<b>35</b>

---

## Elastic GPU Service

---

<b>Create Project.....</b>	<b>35</b>
<b>Registers Clusters.....</b>	<b>35</b>
<b>Admin Dashboard.....</b>	<b>35</b>
<b>Day-to-Day Operations.....</b>	<b>36</b>

## 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 the 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 the installation guide) to get the Admin access token:

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

or use the following command

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

---

Note down the admin access token to use it on the Admin Portal .

# 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      <none>       8088/TCP      46h  
kubeslice-api-gw    <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      <none>       80/TCP        46h  
kubeslice-ui-proxy    <none>      443:31322/TCP  46h  
10.7.45.163     34.19.39.39  443:31322/TCP  46h  
kubeslice-ui-v2      <none>       80/TCP        46h
```

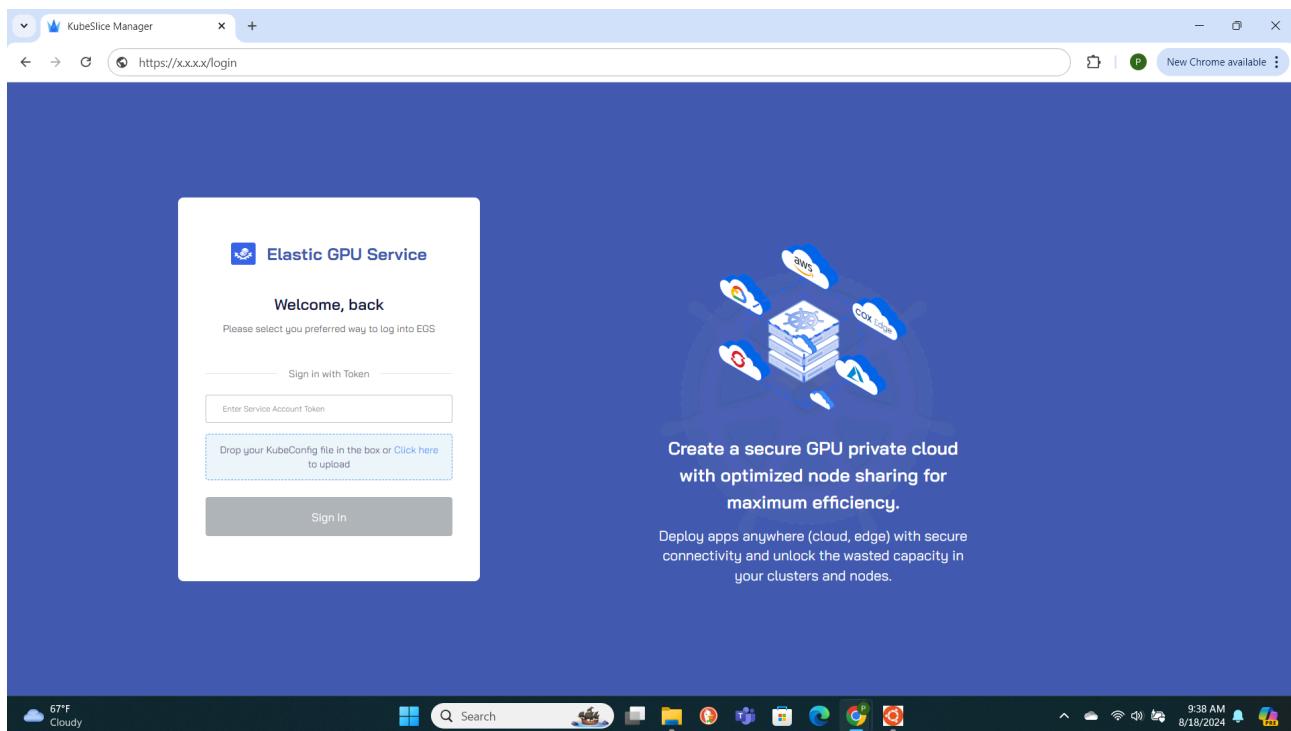
- Note down the LoadBalancer external IP for the kubeslice-ui-proxy pod. This IP will be used to access the EGS Admin/User portals. In the above example: 34.19.39.39 is external IP.

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

# Login to the Admin Portal

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

Use the Admin Access token to log in 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 (workspace).
3. Add User to the Slice workspace.
4. Provision Slice RBAC for User.
  - Add role and role bindings for the Slice namespace.
5. Get UI Portal Access token for the User .
  - User can access the UI portal using the token.
6. Download KubeConfig for the Slice (from where?).
  - Kubeconfig will have SA token to access the User namespace
7. Send the UI portal Access token to the user.
8. Send the KubeConfig to the user.
  - User can access the namespace on the cluster.
  - User RBAC scope is associated namespace.

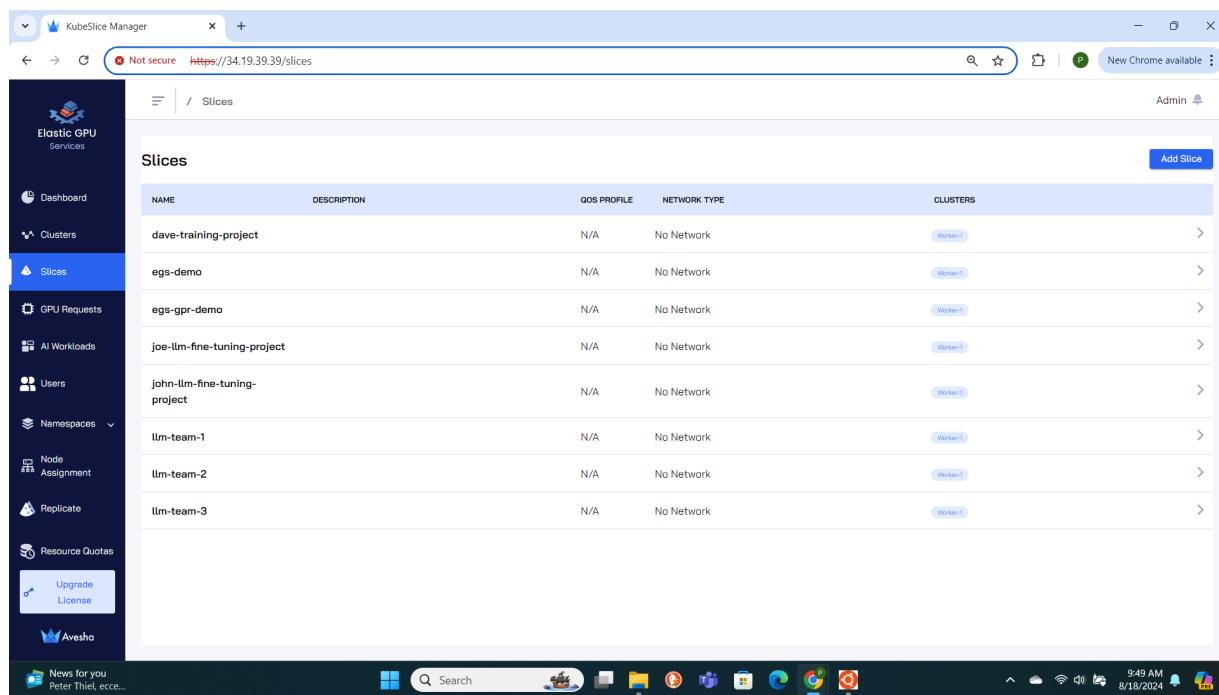
## Elastic GPU Service

---

### Create Slice for User

Perform the following steps to create a Slice.

**Select Slices on the left panel.**



The screenshot shows the KubeSlice Manager web interface. The left sidebar has a dark theme with white icons and text. The 'Slices' option is selected, highlighted in blue. The main content area is titled 'Slices' and contains a table with the following data:

NAME	DESCRIPTION	QOS PROFILE	NETWORK TYPE	CLUSTERS
dave-training-project		N/A	No Network	[Worker-1]
egs-demo		N/A	No Network	[Worker-1]
egs-gpr-demo		N/A	No Network	[Worker-1]
joe-llm-fine-tuning-project		N/A	No Network	[Worker-1]
john-llm-fine-tuning-project		N/A	No Network	[Worker-1]
llm-team-1		N/A	No Network	[Worker-1]
llm-team-2		N/A	No Network	[Worker-1]
llm-team-3		N/A	No Network	[Worker-1]

The top navigation bar shows the URL as <https://34.19.39.39/slices>. The browser status bar indicates it's not secure. The bottom taskbar shows various application icons and the system clock at 9:49 AM, 8/18/2024.

**Click Add Slice.**

# Elastic GPU Service

The screenshot shows the KubeSlice Manager interface. On the left, there's a sidebar with various navigation options: Dashboard, Clusters, Slices (which is selected), GPU Requests, AI Workloads, Users, Namespaces, Node Assignment, Replicate, Resource Quotes, Upgrade, License, and Watchlist Ideas. The main area displays a table titled 'Slices' with columns: NAME, DESCRIPTION, QOS PROFILE, and NETWORK TYPE. The table lists several slices: dave-training-project, egs-demo, egs-gpr-demo, joe-llm-fine-tuning-project, john-llm-fine-tuning-project, llm-team-1, llm-team-2, and llm-team-3. All slices have 'N/A' in the QoS Profile column and 'No Network' in the Network Type column. To the right, a modal window titled 'Add' is open under the 'Details' tab. It has fields for 'Name' (peter-workspace), 'Network Type' (No Network), and 'Description' (Peter's workspace). A 'Next' button is at the bottom of the modal.

## Add slice details:

1. Add **User name**. (example peter-workspace)
2. Select **No-Network** for **Network Type**.
3. Click **Next**.

## Elastic GPU Service

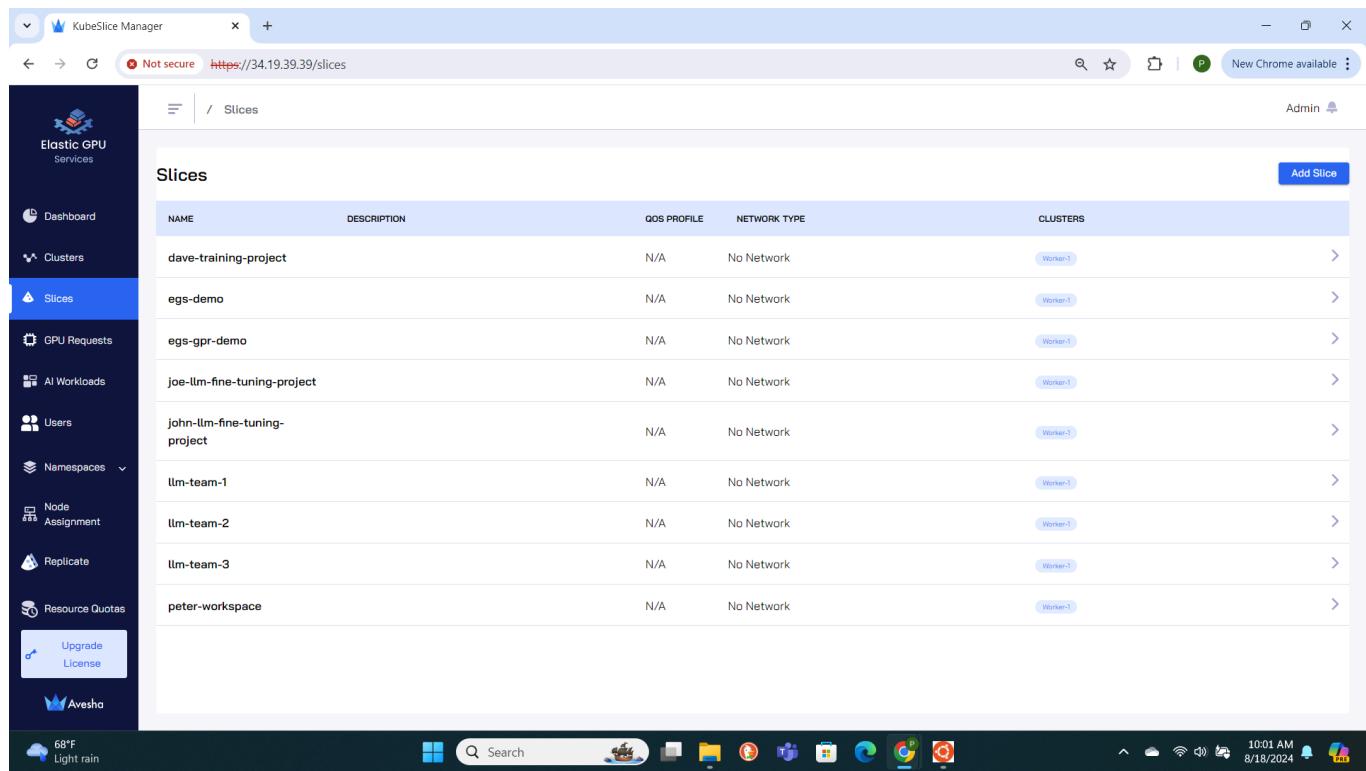
The screenshot shows the KubeSlice Manager interface. On the left, a sidebar menu includes options like Dashboard, Clusters, Slices (selected), GPU Requests, AI Workloads, Users, Namespaces, Node Assignment, Replicate, Resource Quotas, Upgrade License, and Avesha. The main area has two panes: 'Slices' on the left and 'Add' on the right. The 'Slices' pane lists several slices: dave-training-project, egs-demo, egs-gpr-demo, joe-llm-fine-tuning-project, john-llm-fine-tuning-project, llm-team-1, llm-team-2, and llm-team-3. The 'Add' pane shows 'All Clusters' with one entry: worker-1 (Node IP: 34.145.68.139). Below this is a 'Slice Clusters' section with a placeholder message: 'Add clusters from the left pane to connect to this slice'. At the bottom right of the 'Add' pane is a 'Create Slice' button.

### 4. Add Cluster to the Slice

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

## Elastic GPU Service

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



The screenshot shows the KubeSlice Manager interface for managing GPU slices. The left sidebar has a dark theme with the Avesha logo at the bottom. The main area is titled 'Slices' and lists ten entries:

NAME	DESCRIPTION	QOS PROFILE	NETWORK TYPE	CLUSTERS
dave-training-project		N/A	No Network	[Worker-1]
egs-demo		N/A	No Network	[Worker-1]
egs-gpr-demo		N/A	No Network	[Worker-1]
joe-llm-fine-tuning-project		N/A	No Network	[Worker-1]
john-llm-fine-tuning-project		N/A	No Network	[Worker-1]
llm-team-1		N/A	No Network	[Worker-1]
llm-team-2		N/A	No Network	[Worker-1]
llm-team-3		N/A	No Network	[Worker-1]
peter-workspace		N/A	No Network	[Worker-1]

The interface includes a search bar, a toolbar with icons for file operations, and a status bar at the bottom showing weather (68°F, Light rain), system icons, and the date/time (10:01 AM, 8/18/2024).

## Elastic GPU Service

# 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

The screenshot shows the KubeSlice Manager web application running in a Chrome browser. The URL is https://34.19.39.39/#/manage-namespaces. The left sidebar has a dark theme with various service icons and navigation links, including 'Dashboard', 'Clusters', 'Slices', 'GPU Requests', 'AI Workloads', 'Users', 'Namespaces' (which is currently selected and highlighted in blue), 'Node Assignment', 'Replicate', 'Resource Quotas', 'Cost Management', and 'Upgrade License'. The main content area is titled 'Namespaces' and contains a table with the following data:

Slice	SLICE NAME	NAMESPACES	DESCRIPTION
dave-training-project	dave-training-project	No Namespaces added	
egs-demo	egs-demo	No Namespaces added	
egs-gpr-demo	egs-gpr-demo	<a href="#">View all</a>	
joe-llm-fine-tuning-project	joe-llm-fine-tuning-project	No Namespaces added	
john-llm-fine-tuning-project	john-llm-fine-tuning-project	No Namespaces added	
llm-team-1	llm-team-1	No Namespaces added	
llm-team-2	llm-team-2	No Namespaces added	
llm-team-3	llm-team-3	No Namespaces added	
peter-workspace	peter-workspace	No Namespaces added	

The status bar at the bottom shows system information: USD/CAD -0.38%, Avesha logo, and a battery icon indicating 81% charge at 10:26 AM on 8/18/2024.

# Elastic GPU Service

---

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

The screenshot shows the KubeSlice Manager interface. The left sidebar has a dark theme with various service icons and the title 'Elastic GPU Services'. The main area is titled 'Step 1 - Add Namespaces' with the sub-instruction 'Choose namespaces to be put on the Slice from each cluster'. It shows a table of namespaces across different clusters. A specific row for 'peter-workspace' in 'worker-1' is selected, indicated by a checked checkbox and highlighted in blue. The 'Selected Namespaces' column contains 'peter-workspace'. At the bottom right of the table is a blue button labeled 'Add to slice'. The top navigation bar shows the URL 'https://34.19.39.39/#/manage-namespaces/add/1' and indicates 'Not secure'. The status bar at the bottom shows system icons like battery level, signal strength, and date/time.

## Elastic GPU Service

1. Go to the Step 2, Generate YAML and apply tab and click **Apply YAML**

The screenshot shows the KubeSlice Manager interface. The left sidebar has a dark theme with various service icons and navigation links. The main area is titled 'Namespaces' and contains three tabs: 'Step 1 - Add Namespaces', 'Step 2 - Generate YAML and Apply', and 'Step 3 - Finalize'. The 'Step 2' tab is currently selected. Below the tabs is a table with one row, 'peter-workspace', under the 'Slice' column. To the right of the table is a blue button labeled 'Apply YAML'. At the bottom of the main area is a 'Code Preview' section containing the following YAML code:

```
1 apiVersion: controller.kubeslice.io/v1alpha1
2 kind: SliceConfig
3 metadata:
4   annotations:
5     kubeslice.io/slice: peter-workspace
6   creationTimestamp: 2024-08-18T14:00:57Z
7   finalizers:
8     controller.kubeslice.io/slice-configuration-finalizer
9   generation: 1
10  managedFields:
11    - apiVersion: controller.kubeslice.io/v1alpha1
12      fieldsType: FieldV1
13      fieldsV1:
14        f:metadata:
15          f:finalizers:
16            f:v:
17              v:"controller.kubeslice.io/slice-configuration-finalizer"
18        manager: manager
19        operation: Update
```

## Elastic GPU Service

---

### View Namespaces

1. Select the **Slice** that you want to view.
2. Click **view\_all** on the Slice bar to see the namespaces associated with User Slice.

The screenshot shows the KubeSlice Manager interface. On the left, there's a sidebar with various navigation options: Dashboard, Clusters, Slices, GPU Requests, AI Workloads, Users, Namespaces (selected), Node Assignment, Replicate, Resource Quotas, Cost Management, Upgrade License, Avesha, Watchlist, and Ideas. The main area has a header "Namespaces /". A modal window titled "Step 1 - Add Namespaces" is open, with the sub-header "Choose namespaces to be put on the Slice from each cluster". It shows a table with one row:

No.	NAMESPACES	CLUSTERS
1.	peter-workspace	worker-1

At the top right of the modal, it says "Total Namespaces : 1" and "Slice name: peter-workspace". There's also a search bar labeled "Search Namespaces". At the bottom right of the modal, there's a "Step 3 - Finalize" button with a "Display Final state" link. The status bar at the bottom right shows the time as 10:31 AM and the date as 8/18/2024.

### Add User to the Slice

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

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

Select **Users** on the left panel.

The screenshot shows the KubeSlice Manager web application. The left sidebar has a dark theme with white icons and text. It includes links for Dashboard, Clusters, Slices, GPU Requests, AI Workloads, and Users (which is highlighted in blue). Below these are Namespace, Node Assignment, Replicate, Resource Quotes, Upgrade, and License. At the bottom of the sidebar is the Avesha logo. The main content area has a light gray header with the URL https://34.19.39.39/users and a search bar. The title is "Users". The table below has columns: USER NAME, EMAIL, TYPE, ROLE, SLICE, and CREATED AT. There is one row with data: egs-gpr-demo, egs-gpr-demo@abc.com, Local, User, egs-gpr-demo, Aug 16, 2024. To the right of the table are edit and delete icons. The top right of the main area shows "Admin". The bottom of the screen shows a Windows taskbar with various pinned icons and the date/time 10:35 AM 8/18/2024.

USER NAME	EMAIL	TYPE	ROLE	SLICE	CREATED AT
egs-gpr-demo	egs-gpr-demo@abc.com	Local	User	egs-gpr-demo	Aug 16, 2024

1. Click **Add User**.
2. Add the following 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 the **User** role.
  - f. Click **Submit**."

## Elastic GPU Service

The screenshot shows a web browser window for the KubeSlice Manager. The URL is <https://34.19.39.39/users>. The page has a dark theme. On the left, there is a sidebar with various options: Dashboard, Clusters, Slices, GPU Requests, AI Workloads, Users (selected), Namespaces, Node Assignment, Replicate, Resource Quotes, Upgrade, License, and Avesha. The main area shows a table titled 'Users' with columns: USER NAME, EMAIL, TYPE, ROLE, and SLICE. One row is visible: egs-gpr-demo, egs-gpr-demo@abc.com, Local, User, egs-gpr-de. To the right of the table is a modal window titled 'Add User' with fields for Name (Peter), Email (peter@acmecorp.com), Assign to Slice (peter-workspace), Type (Local), and Role (User). A 'Submit' button is at the bottom of the modal.

You should see the newly added user details in the **Users** list.

# Elastic GPU Service

The screenshot shows a web browser window titled "KubeSlice Manager" with the URL <https://34.19.39.39/users>. The page displays a table of users under the heading "Users". The table has columns: USER NAME, EMAIL, TYPE, ROLE, SLICE, and CREATED AT. Two rows are visible:

USER NAME	EMAIL	TYPE	ROLE	SLICE	CREATED AT
egs-gpr-demo	egs-gpr-demo@abc.com	Local	User	egs-gpr-demo	Aug 16, 2024
Peter	peter@acmecorp.com	Local	User	peter-workspace	Aug 18, 2024

The left sidebar contains a navigation menu with the following items:

- Dashboard
- Clusters
- Slices
- GPU Requests
- AI Workloads
- Users** (selected)
- Namespaces
- Node Assignment
- Replicate
- Resource Quotas
- Upgrade
- License
- Avesha

The bottom of the screen shows a Windows taskbar with various pinned icons and system status indicators.

### Provision Slice RBAC for User

Perform the following steps to create an RBAC role for user that restricts the user access to worker cluster user namespaces.

### RBAC for Slice

1. Select **RBAC** on the left panel
2. Select the User Slice.
3. Click **Assign Roles**

SLICE NAME	TYPE	DESCRIPTION	ROLES	KUBECONFIG
egs-gpr-demo	Application		No Roles Assigned	<a href="#">Download</a>
peter-workspace	Application		No Roles Assigned	<a href="#">Download</a>

### Add Role Details

1. Select **Service Account** for the User/Group.
2. Enter a name for the Service Account.

## Elastic GPU Service

---

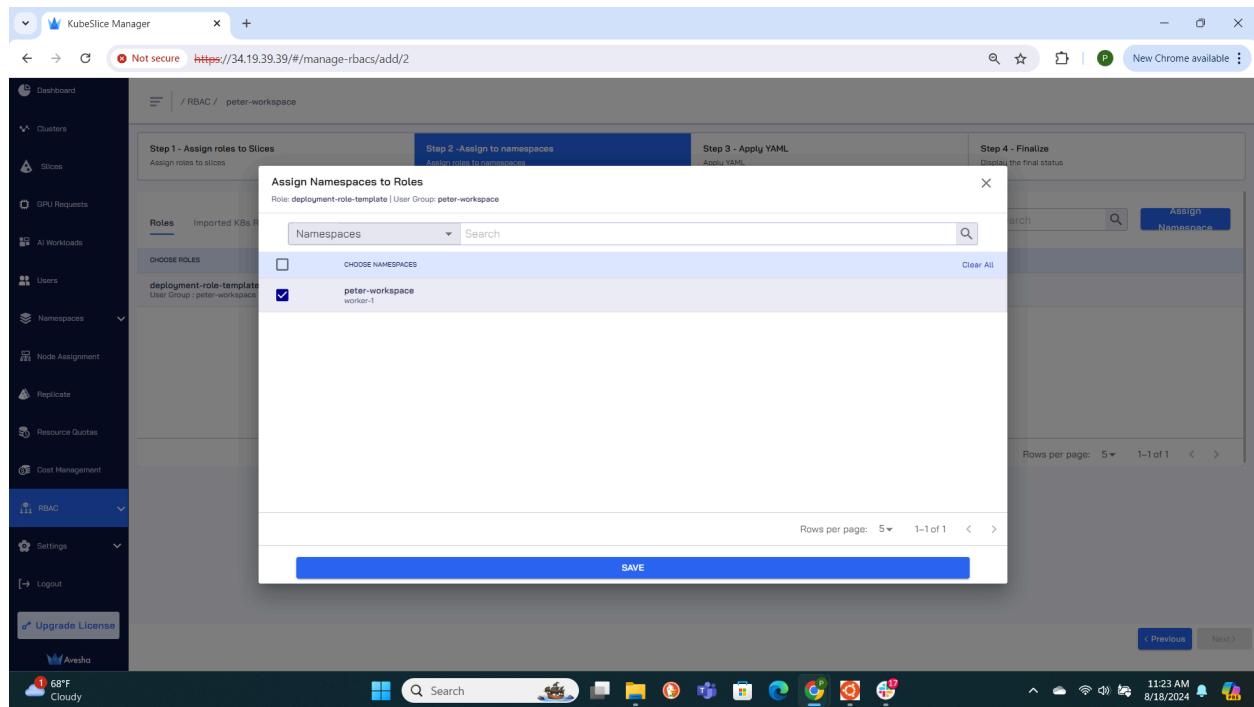
3. From the **Select roles** drop-down list, pick **deployment-role-template**.
4. Click **Next**.

 **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.

Go to the Step 2: Assign Namespace tab and:

1. Select role from the list.
2. Click on Assign Namespace.
3. Select namespace.
4. Click **Save**.
5. Click **Next**.



Go to the **Step 3 Apply YAML** tab and:

1. Click **Apply YAML**
2. Click **Done** to apply the configuration.

# Elastic GPU Service

The screenshot shows the KubeSlice Manager interface for managing RBAC. The left sidebar has a 'RBAC' section selected. The main area shows a 'Slice' table with one entry: 'peter-workspace' (Type: Application). A 'Code Preview' panel displays the YAML configuration for this slice.

SLICE NAME	TYPE	DESCRIPTION	ROLES
peter-workspace	Application		<a href="#">View all</a> <a href="#">Download</a>

```
apiVersion: controller.kubeslice.io/v1alpha1
kind: SliceRoleBinding
metadata:
  name: peter-workspace
  namespace: peter-workspace
```

The screenshot shows the KubeSlice Manager interface for managing RBAC. The left sidebar has a 'RBAC' section selected. A modal window is open, titled 'Total Roles: 1', showing the details for the slice 'peter-workspace'. It lists one role: 'deployment-role-template' (User Group: peter-workspace).

NO	ROLES
1	deployment-role-template User Group : peter-workspace

### 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_efs_slice_token.sh -h  
%./fetch_efs_slice_token.sh -k <kubeconfig> -s <slice name> -p avesha
```

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

### Download KubeConfig for User

Admin can download the User RBAC Kubeconfig YAMLfile from the RBAC panel.

1. Select RBAC on the left panel.
2. Select the User Slice.
3. Click the **Download** button to download the user namespace-scoped Kubeconfig for the worker cluster.

 **Note:** User Slice RBAC role needs to be provisioned to download the KubeConfig YAML file.

Go to the Slice page, on the **Slice** bar, click **view\_all** to see the roles for the Slice.

Click the **Download** button to download the Kubeconfig file on the worker cluster with user-specific RBAC.

### Share the Token with the User

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

# Elastic GPU Service

The screenshot shows the KubeSlice Manager interface for managing RBAC. The left sidebar has a dark theme with various navigation options like Dashboard, Clusters, Slices, GPU Requests, AI Workloads, Users, Namespaces, Node Assignment, Replicate, Resource Quotes, Cost Management, RBAC (which is selected and highlighted in blue), Settings, Logout, Upgrade License, and Avesha.

The main content area is titled "RBAC / peter-workspace". It consists of four tabs: Step 1 - Assign roles to Slices (selected), Step 2 - Assign to namespaces, Step 3 - Apply YAML, and Step 4 - Finalize. Under Step 1, there's a table for "Slice" with one entry: "peter-workspace" (TYPE: Application). Below it is a "Select roles" section where a "Service Account" is chosen, and a search bar shows "peter-workspace". To the right, a "Select roles" dropdown lists "deployment-role-template" (checked) and "reader-role-template". A "Next >" button is at the bottom right of this section.

The second screenshot shows the same interface after the configuration. The main content area is titled "/ RBAC". It displays a table of "Slices" with two entries: "peter-workspace" (Application type, Roles: View\_all, KUBECONFIG: Download) and "egs-gpr-demo" (Application type, Roles: No Roles Assigned, KUBECONFIG: Download). The "Edit Roles" button is visible at the top right of the slice table. The bottom right shows pagination: Rows per page: 10, 1–2 of 2.

# Manage GPR Queue

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

## View GPRs across all the Users

1. Select **GPU requests** on the left sidebar. The **All GPU Requests** tab shows all the GPU requests across all the Slices.

Admin can use **search textbox or Filter** to filter the GPRs.

The screenshot shows the KubeSlice Manager web interface. The left sidebar has a dark theme with white icons and text. It includes links for Dashboard, Clusters, Slices, GPU Requests (which is highlighted in blue), AI Workloads, Users, Namespaces, Node Assignment, Replicate, Resource Quotas, Upgrade License, and Avesha. The main content area has a light background. At the top, it says 'GPU Requests' with tabs for 'All GPU Requests' (which is selected) and 'GPU Requests Per Slice'. Below that is a search bar with placeholder 'Enter search' and a magnifying glass icon. To the right of the search bar is a 'Filter' button. The main table lists seven GPU requests:

REQUEST NAME	REQUESTED BY	# GPUS	# GPU NODES	# GPU SHAPE	REQUESTED START TIME	RESERVED FOR	STATUS	ACTIONS
peter-gpr-2	Peter	1	1	n1-standard-16	08/18/2024 12:10 PM	10 Mins	Pending	X ⋮
peter-gpr-1	Peter	1	1	n1-standard-16	08/18/2024 12:00 PM	10 Mins	Provisioned	X ⋮
gpr11	egs-gpr-demo	1	1	n1-standard-16	08/16/2024 4:34 PM	20 Mins	Complete	X ⋮
gpr6	egs-gpr-demo	1	1	n1-standard-16	08/17/2024 8:24 PM	10 Mins	Complete	X ⋮
gpr7	egs-gpr-demo	1	1	n1-standard-16	08/17/2024 8:34 PM	10 Mins	Complete	X ⋮
gpr5	egs-gpr-demo	1	1	n1-standard-16	08/17/2024 8:03 PM	5 Mins	Complete	X ⋮

At the bottom of the table, it says 'Showing 1 - 7 of 7 entries' and has navigation arrows. To the right, there are buttons for 'view 10 rows per page'. The bottom of the screen shows a Windows taskbar with various icons and the date/time '12:00 PM 8/18/2024'.

## View the GPRs Specific to a Slice

Select the **GPU Requests Per Slice** tab, and click the Slice to see its specific GPRs.

### GPR Request for a User

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

1. Select the User Slice in **GPU Requests Per Slice** tab.
2. Select the User Slice.
3. Click **Create GPU Request**.

The screenshot shows the KubeSlice Manager interface. The left sidebar has a dark theme with white icons and text. It includes sections for Dashboard, Clusters, Slices, GPU Requests (which is currently selected and highlighted in blue), AI Workloads, Users, Namespaces, Node Assignment, Replicate, Resource Quotas, Upgrade License, and Avesha. The main content area has a light background. At the top, it says "Slice Name: peter-workspace". Below that is a table titled "GPU Requests" with columns: REQUEST NAME, # GPUS, # GPU NODES, PRIORITY, GPU SHAPE, ESTIMATED START TIME, RESERVED FOR, and STATUS. Two rows are listed: "peter-gpr-2" and "peter-gpr-1", both marked as "Complete". At the bottom of the table, it says "Showing 1 - 2 of 2 entries" and "view 10 rows per page". The status column for each row contains a blue button labeled "Complete" with a right-pointing arrow. The bottom of the screen shows a Windows taskbar with various icons and the date/time "8/18/2024 2:55 PM".

1. On the **Create GPU Request** page, add the request details:
2. Enter GPR name, Number of GPU nodes.
3. Select GPU shape.
4. Select priority.
5. Select Reserve for duration.
6. Click the **Get Wait Time** button. EGS shows the estimated wait time for the GPU nodes provisioning.
7. Select the GPU in the **Available GPUs** table with acceptable estimated wait time.
8. Click **Request GPUs**.

## Elastic GPU Service

The screenshot shows the KubeSlice Manager interface with the URL <https://34.19.39.39/gpu-request/peter-workspace/worker-1>. The main page displays 'GPU Requests' for the 'peter-workspace' slice. A modal window titled 'Create GPU Request' is open, prompting for details:

- Slice Name:** peter-workspace
- Request Name:** peter-gpr-3
- GPU Nodes:** 1
- GPU Shape:** n1-standard-16
- GPU Mode:** Virtual Machine
- Memory (GB) per GPU:** 15
- User:** Admin
- Priority:** low
- Requested Start Time:** mm/dd/yyyy --::--
- Reserve for:** 0 Days, 0 Hrs, 15 Mns

Below the form, there's a section for 'AI Model Params' with a 'Get Wait Time' button and a 'Clear All' link. A table titled 'Available GPUs' lists one entry:

GPU SHAPE	NO OF GPUs	NO OF NODES	ESTIMATED START TIME	ESTIMATED WAIT TIME	NODE COST/Hr	TOTAL ESTIMATED COST
n1-standard-16	1	1	08/18/2024 2:56 PM	0	NA	NA

A 'Request GPUs' button is located at the bottom right of the table.

View the **GPR** in the **User's GPU Requests** queue.

**Note:** GPRs can be created by additional methods using EGS APIs or by applying GPR custom resource YAML to the KubeApi server. These methods can be invoked by the CI/CD or RAG pipelines or external system/services or application services in the cluster.

# Elastic GPU Service

The screenshot shows the KubeSlice Manager interface for managing GPU requests. The left sidebar has a dark theme with various icons and links: Elastic GPU Services, Dashboard, Clusters, Slices, GPU Requests (selected), AI Workloads, Users, Namespaces, Node Assignment, Replicate, Resource Quotes, Upgrade License, and Avesha. The main content area shows a table of GPU Requests for the slice 'peter-workspace'. The table has columns: REQUEST NAME, # GPUS, # GPU NODES, PRIORITY, GPU SHAPE, ESTIMATED START TIME, RESERVED FOR, and STATUS. Three entries are listed:

REQUEST NAME	# GPUS	# GPU NODES	PRIORITY	GPU SHAPE	ESTIMATED START TIME	RESERVED FOR	STATUS
peter-gpr-3	1	1	High	n1-standard-16	08/18/2024 2:56 PM	15 Mins	Provisioned
peter-gpr-2	1	1	High	n1-standard-16	08/18/2024 12:10 PM	10 Mins	Complete
peter-gpr-1	1	1	High	n1-standard-16	08/18/2024 12:00 PM	10 Mins	Complete

Below the table, it says 'Showing 1 - 3 of 3 entries' and has navigation buttons << < 1 > >>. To the right, there's a 'view 10 rows per page' dropdown. The bottom of the screen shows a Windows taskbar with various icons and the date/time '3:03 PM 8/18/2024'.

Click the **GPR** to view request details.

# Elastic GPU Service

The screenshot shows the KubeSlice Manager interface for managing GPU requests. The main window displays a list of GPU requests under the slice 'peter-workspace'. The details for each request are shown in a table, along with a summary of GPU usage.

**Slice Name:** peter-workspace

**GPU Requests**

REQUEST NAME	# GPUs	# GPU NODES	PRIORITY	GPU SHAPE
peter-gpr-3	1	1	High	n1-standard-1
peter-gpr-2	1	1	High	n1-standard-1
peter-gpr-1	1	1	High	n1-standard-1

**Request Details**

Request Name	Memory per GPU	Status	Priority	GPU Nodes	Requested Start Time	Time
peter-gpr-3	GB	Provisioned	High	1	NA	NA

**GPU Mode:** Virtual Machine

**GPUs Details**

NO OF GPUs	NO OF NODES	ESTIMATED START TIME	NODE Cost/Hr	TOTAL ESTIMATED COST
1	1	08/18/2024 2:56 PM	NA	NA

## 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, or early-release a GPR
4. Admin can do 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, medium: 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 required 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. Exceed the quota for the Slice
2. Exceed the GPU nodes limit per GPR (request for large number of GPU nodes)

 **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 and so on.

 **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.

1. Select **AI Workloads** from the left panel.
2. Select User Slice to see the AI workloads for the slice.

The screenshot shows the KubeSlice Manager web interface. The left sidebar has a dark theme with white icons and text, showing options like Dashboard, Clusters, Slices, GPU Requests, **AI Workloads** (which is selected and highlighted in blue), Users, Namespaces, Node Assignment, Replicate, Resource Quotas, Upgrade License, and Avesha. The main content area has a light background and displays a table titled "AI Workloads". The table has columns: NAME, DESCRIPTION, TYPE, and CLUSTERS. There are nine rows of data:

NAME	DESCRIPTION	TYPE	CLUSTERS
dave-training-project		Application	worker-1
egs-demo		Application	worker-1
egs-gpr-demo		Application	worker-1
joe-llm-fine-tuning-project		Application	worker-1
john-llm-fine-tuning-project		Application	worker-1
llm-team-1		Application	worker-1
llm-team-2		Application	worker-1
llm-team-3		Application	worker-1
peter-workspace		Application	worker-1

The bottom of the screen shows a Windows taskbar with various pinned icons and system status indicators.

## Model Details

Select User Slice to view AI workloads for User workspace. The model details:

- 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

---

The screenshot shows a browser window for the KubeSlice Manager. The URL is <https://34.19.39.39/workspace/peter-workspace>. The page title is "AI Workloads". The left sidebar has a dark theme with the following menu items:

- Elastic GPU Services
- Dashboard
- Clusters
- Slices
- GPU Requests
- AI Workloads** (highlighted)
- Users
- Namespaces
- Node Assignment
- Replicate
- Resource Quotas
- Upgrade License
- Avesha

The main content area displays "Slice Name: peter-workspace" and "AI Model Details". A table shows one entry:

MODEL#	WORKLOAD NAME	CONFIG PARAMETERS	INFRASTRUCTURE	NAVIGATE
1	llm-demo-75fbc46f7f		<p>Pods: 1 GPU Model: NVIDIA A10 GPUs: 1 Memory: 24 GB</p>	<a href="#">Go to Pods</a> <a href="#">Go to GPUs</a>

The bottom of the screen shows a Windows taskbar with various icons and the date/time: 3:05 PM 8/18/2024.

## View Pods

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

## Elastic GPU Service

The screenshot shows the KubeSlice Manager interface for managing AI workloads. The left sidebar has a dark theme with various icons and sections like Dashboard, Clusters, Slices, GPU Requests, and AI Workloads (which is currently selected). The main content area shows a slice named 'peter-workspace'. Below it, a table titled 'Pod Details' lists one pod entry:

POD #	POD/JOB	CONTAINER	NAMESPACE	CLUSTER	
1	llm-demo-75fbc46f7f-hbg5g	text-generation-inference	peter-workspace	worker-1	<a href="#">Go to GPUs</a>

The browser address bar shows the URL <https://34.19.39.39/workspace/peter-workspace/llm-demo-75fbc46f7f>. The system tray at the bottom right shows the date and time as 8/18/2024 3:05 PM.

## View GPUs

On the **AI Model Details** page, click **Go to GPUs** to view the GPU table page. The GPU table:

- Shows sorted list of GPUs with high power, temperature GPUs at the top for quick access
- Shows the hotspot GPUs

Click **View Dashboard** to view time-series data for the selected GPU device.

# Elastic GPU Service

The screenshot shows the KubeSlice Manager interface for managing AI workloads. The main navigation bar at the top indicates the URL is <https://34.19.39.39/workspace/peter-workspace/gpus-all>. The left sidebar has a dark theme with the 'Elastic GPU Services' logo and a list of options: Dashboard, Clusters, Slices, GPU Requests, **AI Workloads** (which is selected), Users, Namespaces, Node Assignment, Replicate, Resource Quotas, Upgrade License, and Avesha. The main content area shows the 'Slice Name: peter-workspace'. Below this, the 'GPU Details' section displays a table with one row. The table columns are: INSTANCE, NODE IP, GPU NO, GPU SHAPE, GPU LEVEL METRICS, and DASHBOARD LINK. The data in the table is: INSTANCE (~ 1), NODE IP (0), GPU NO (Tesla T4), GPU SHAPE (represented by four green dots), and DASHBOARD LINK (a blue link labeled 'View Dashboard'). Below the table, there is a section titled 'GPU Level Metrics' with five metrics: START TIME (18/08/2024 12:20), TEMPERATURE (°C) (48), POWER (watt) (28), MEMORY (0), and GPU UTILIZATION (0). The bottom of the screen shows a Windows taskbar with various icons and the system tray indicating the date and time as 3:05 PM on 8/18/2024.

## 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 organizations 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 Clusters

During an 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 the dashboard to monitor key metrics related to the GPU pool.
  - o GPRs and GPU utilization, allocation, Errors, Alerts
2. Use the 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. Admin can generate access tokens for Users for UI access
5. Admin can view GPRs queue across all the Users/Slices.
6. Admin can edit/delete GPRs - delete or early-release a provisioned GPR.
7. Admin can view AI workloads across all the Users/Slices.
  - o View GPU dashboards for AI workloads. Access time-series data for all workload GPUs.
  - o View pods, model details.
  - o Access top power/temp GPUs. Filter/search GPUs.
8. Admin can register additional Clusters
  - o add/remove clusters
9. Admin can create additional Projects
  - o Add/remove projects

The following workflows will be available in the next release:

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