

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.

Elastic GPU Service

- 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
```

Elastic GPU Service

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.

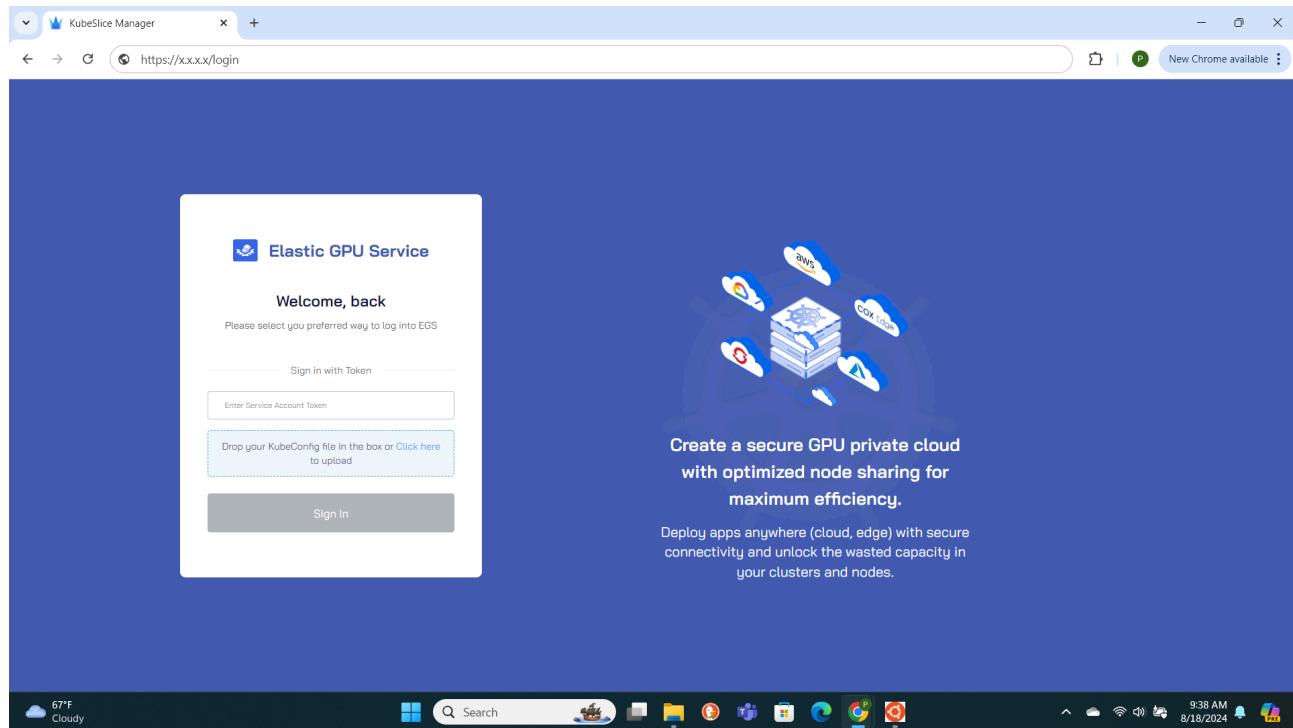
UI Portal: <https://<ui-proxy-ip>>

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.

Elastic GPU Service



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).

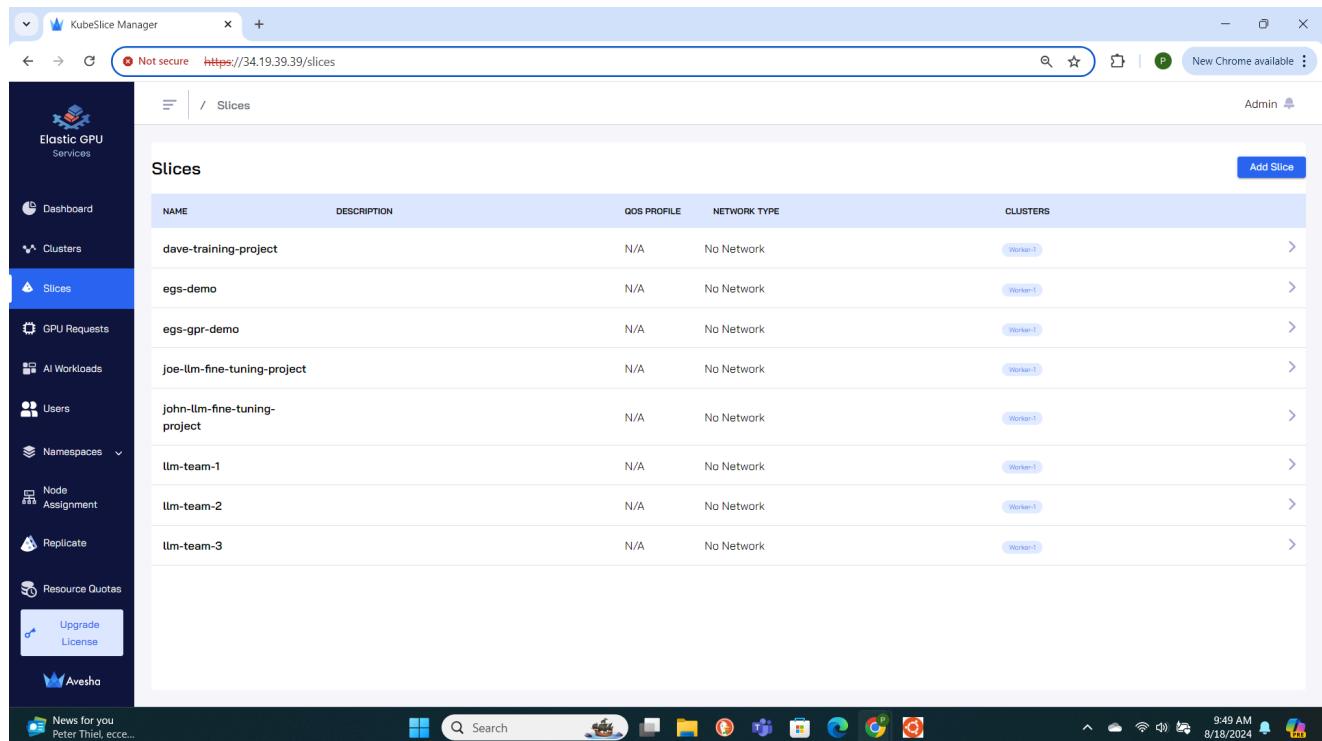
1. Create a Slice for the User (with team name or user name)
2. Assign a namespace to the User Slice
3. Add User to the Slice
4. Provision Slice RBAC for User
 1. Add role and role bindings for the Slice namespace
5. Get UI Portal Access token for the User
 1. User can access the UI portal using the 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

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 interface. On the left, there is a sidebar with the following menu items:

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

The main content area is titled "Slices" and displays 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-lilm-fine-tuning-project		N/A	No Network	[Worker-1]
john-lilm-fine-tuning-project		N/A	No Network	[Worker-1]
lilm-team-1		N/A	No Network	[Worker-1]
lilm-team-2		N/A	No Network	[Worker-1]
lilm-team-3		N/A	No Network	[Worker-1]

At the top right of the main content area, there is a blue button labeled "Add Slice". The browser address bar shows "https://34.19.39.39/slices". The status bar at the bottom indicates "9:49 AM 8/18/2024".

Click on Add Slice

Elastic GPU Service

The screenshot shows the KubeSlice Manager web 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 Watchlist Ideas. The main content area displays a table titled 'Slices' with columns: NAME, DESCRIPTION, QOS PROFILE, and NETWORK TYPE. Several slices are listed, such as 'dave-training-project', 'egs-demo', 'egs-gpr-demo', 'joe-llm-fine-tuning-project', 'llm-team-1', 'llm-team-2', and 'llm-team-3'. To the right, a modal window titled 'Add' is open, showing a 'Details' tab where users can input slice information. Fields include 'Name' (set to 'peter-workspace'), 'Network Type' (set to 'No Network'), and 'Description' (set to 'Peter's workspace'). A 'Next' button is at the bottom of the modal.

Add slice details:

Add User name: (example peter-workspace)

Select No-Network for Network Type

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 Quotes, Upgrade License, and Avesha. The main area displays a list of 'Slices' with columns for NAME and DESCRIPTION. The slices listed are: 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. To the right, a modal window titled 'Add' is open, showing 'All Clusters' with a single entry: 'worker-1' (Node IP: 34.145.68.139). Below this, the 'Slice Clusters' section has a placeholder message: 'Add clusters to connect with the slice'. At the bottom of the modal is a 'Create Slice' button.

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.

Elastic GPU Service

The screenshot shows the KubeSlice Manager web interface. The left sidebar has a dark theme with icons for Dashboard, Clusters, Slices (selected), GPU Requests, AI Workloads, Users, Namespaces (with a dropdown arrow), Node Assignment, Replicate, Resource Quotas, Upgrade License, and Avesha. The main content area is titled 'Slices' and lists ten entries:

NAME	DESCRIPTION	QOS PROFILE	NETWORK TYPE	CLUSTERS	Actions
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 bottom status bar shows it's 68°F Light rain, the system tray has various icons, and the date/time is 10:01 AM 8/18/2024.

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
```

Elastic GPU Service

Select namespaces on the left panel

The screenshot shows the KubeSlice Manager interface. On the left, there is a sidebar with various navigation options: Dashboard, Clusters, Slices, GPU Requests, AI Workloads, Users, Namespaces (which is currently selected), Node Assignment, Replicate, Resource Quotas, and Cost Management. Below the sidebar, there are two status indicators: 'USD/CAD -0.38%' and the user 'Avesha'. The main content area is titled 'Namespaces' and contains a table with the following data:

SLICE NAME	NAMESPACES	DESCRIPTION
dave-training-project	No Namespaces added	
ega-demo	No Namespaces added	
egs-gpu-demo	View all	
joe-ilm-fine-tuning-project	No Namespaces added	
john-ilm-fine-tuning-project	No Namespaces added	
ilm-team-1	No Namespaces added	
ilm-team-2	No Namespaces added	
ilm-team-3	No Namespaces added	
peter-workspace	No Namespaces added	

At the top right of the main content area, there is a button labeled 'Add Namespaces...'. The browser address bar shows 'Not secure https://34.19.39.39/#/manage-namespaces'. The top right corner of the window shows the user 'Admin'.

- 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

The screenshot shows the KubeSlice Manager interface. On the left is a dark sidebar with various service icons and a 'Namespaces' section currently selected. The main area has three tabs at the top: 'Step 1 - Add Namespaces', 'Step 2 - Generate YAML and Apply', and 'Step 3 - Finalize'. The 'Step 1' tab is active, showing a table of namespaces from different clusters. A row for 'peter-workspace' is selected, highlighted with a blue border. The table includes columns for 'NAMESPACE', 'CLUSTER NAME', and '...'. A search bar and an 'Add to slice' button are visible on the right of the table. The 'SLICE NAME' field contains 'peter-workspace'.

1. Click on Apply YAML

Elastic GPU Service

The screenshot shows the KubeSlice Manager web interface. The left sidebar has a dark theme with icons for Dashboard, Clusters, Slices, GPU Requests, AI Workloads, Users, Namespaces (selected), Node Assignment, Replicate, Resource Quotas, Cost Management, and Upgrade License. The main area has a light theme. At the top, there are three tabs: Step 1 - Add Namespaces (disabled), Step 2 - Generate YAML and Apply (selected), and Step 3 - Finalize. The Step 2 tab has sub-instructions: "Choose namespaces to be put on the Slice from each cluster" and "Generate Slice YAML with all namespaces for each cluster". Below this is a "Slice" table with one row: "SLICE NAME" (peter-workspace), "NAMESPACES" (View all), and "DESCRIPTION". A blue "Apply YAML" button is at the top right of the table. On the left, under the Namespaces section, there is a "Code Preview" block containing a snippet of YAML code:

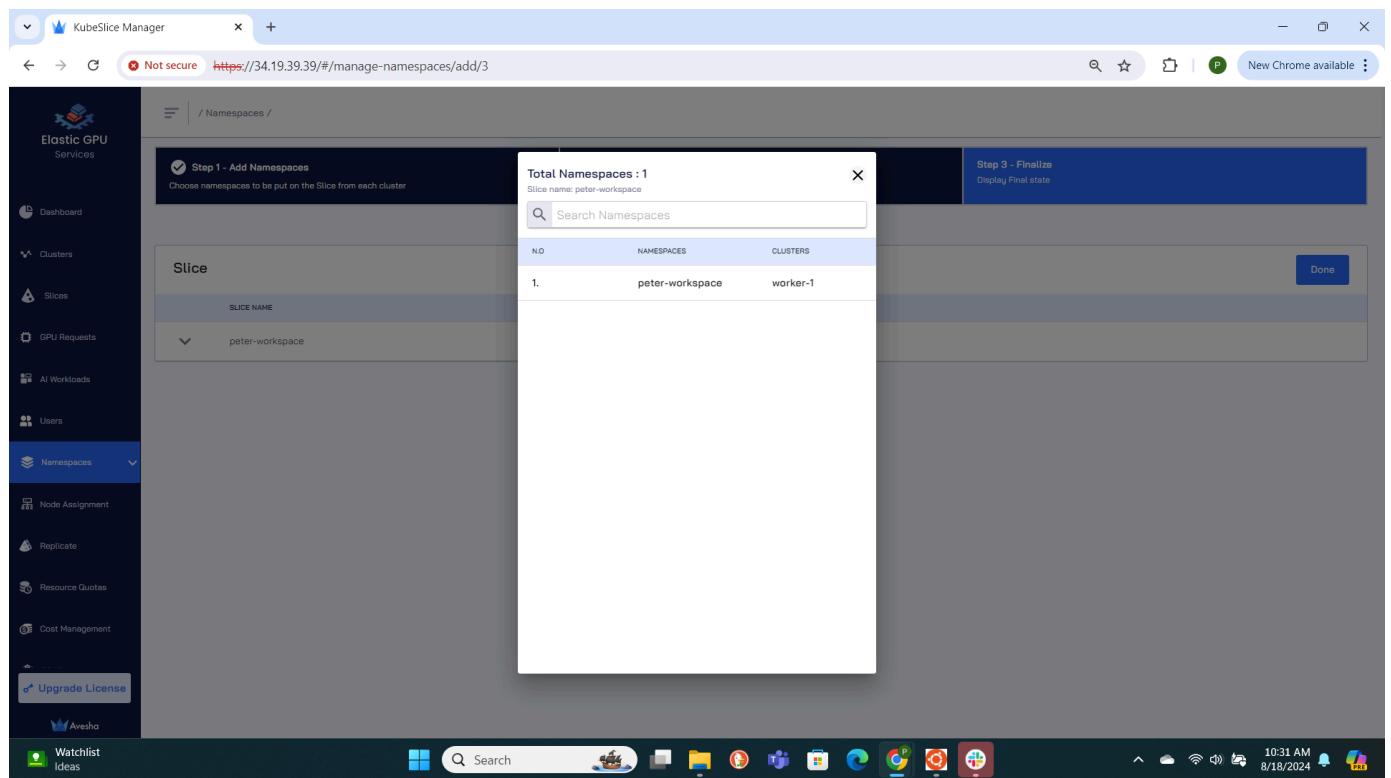
```
1 apiVersion: controller.kubeslice.io/v1alpha1
2 kind: SliceConfig
3 metadata:
4   annotations:
5     v1alpha1:
6       creationTimestamp: 2024-08-18T14:00:57Z
7   finalizers:
8     controller.kubeslice.io/slice-configuration-finalizer
9   generation: 1
10  managedFields:
11    - apiVersion: controller.kubeslice.io/v1alpha1
12      fieldsV1:
13        fieldV1:
14          fmetadata:
15            ffinalizers:
16              f:
17                v: "controller.kubeslice.io/slice-configuration-finalizer"
18      manager: manager
19      operation: Update
```

View namespaces

Select Slice

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

Elastic GPU Service



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.

The screenshot shows the KubeSlice Manager interface. On the left, there is a dark sidebar with various navigation options: Dashboard, Clusters, Slices, GPU Requests, AI Workloads, and a highlighted 'Users' option. Below 'Users' are sub-options: Namespaces (with a dropdown arrow), Node Assignment, Replicate, Resource Quotas, Upgrade, and License. At the bottom of the sidebar is the Avesha logo and the URL https://34.19.39.39/users. The main content area has a header 'Users' with a search bar and an 'Add User' button. The table below lists one user: egs-gpr-demo, with details: Email: egs-gpr-demo@abc.com, Type: Local, Role: User, Slice: egs-gpr-demo, and Created At: Aug 16, 2024. There are edit and delete icons next to the user row. The browser status bar at the bottom shows 'Not secure' and the URL https://34.19.39.39/users. The system tray at the bottom right shows the date and time as 8/18/2024 10:35 AM.

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”

Elastic GPU Service

The screenshot shows a web browser window titled "KubeSlice Manager" with the URL "https://34.19.39.39/users". The left sidebar contains navigation links for Dashboard, Clusters, Slices, GPU Requests, AI Workloads, Users (selected), Namespaces, Node Assignment, Replicate, Resource Quotes, Upgrade License, and Avesha. The main content area has a header "Users" and a table with columns: USER NAME, EMAIL, TYPE, ROLE, and SLICE. One row is visible: "egs-gpr-demo" with email "egs-gpr-demo@abc.com", type "Local", role "User", and slice "egs-gpr-de". To the right, a modal window titled "Add User" is open, containing 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.

USER NAME	EMAIL	TYPE	ROLE	SLICE
egs-gpr-demo	egs-gpr-demo@abc.com	Local	User	egs-gpr-de

Add User

Name *
Peter

Email *
peter@acmecorp.com

Assign to Slice *
peter-workspace

Type *
Local

Role *
User

Submit

You should see added User details in the Users list.

Elastic GPU Service

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

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

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

Elastic GPU Service

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**



SLICE NAME	TYPE	DESCRIPTION	ROLES	KUBECONFIG
egs-gpr-demo	Application		No Roles Assigned	Download
peter-workspace	Application		No Roles Assigned	Download

Add Role details

Select “Service Account” for User/Group

Enter name

For “Select roles” - pick deployment-role-template

Click **Next**

Elastic GPU Service

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.

The screenshot shows the KubeSlice Manager interface for managing RBAC. The left sidebar has a 'RBAC' section selected. The main area is titled 'Step 1 - Assign roles to Slices' with a sub-section 'Assign roles to slices'. It shows a table for a 'Slice' named 'peter-workspace' which is an 'Application'. Below the table, there's a search bar for 'NAME' and a dropdown for 'USER / GROUP' set to 'Service Account'. A modal window is open, showing a list of roles: 'deployment-role-template' (checked) and 'reader-role-template'. At the bottom right of the main area is a 'Next >' button.

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

Elastic GPU Service

The screenshot shows the KubeSlice Manager interface for managing RBAC. The main navigation bar includes 'Dashboard', 'Clusters', 'Slices', 'GPU Requests', 'AI Workloads', 'Users', 'Namespaces', 'Node Assignment', 'Replicate', 'Resource Quotas', 'Cost Management', 'RBAC', 'Settings', and 'Logout'. The current page is 'RBAC / peter-workspace'. A modal window titled 'Assign Namespaces to Roles' is open, showing a table with one row: 'peter-workspace' (worker-1). The 'SAVE' button at the bottom is highlighted.

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

Elastic GPU Service

The screenshot shows the KubeSlice Manager interface. The left sidebar has a dark theme with various options like Dashboard, Clusters, Slices, GPU Requests, AI Workloads, Users, Namespaces, Node Assignment, Replicate, Resource Quotas, Cost Management, RBAC (which is currently selected), and Settings. The main content area has a light theme. At the top, it says 'Step 1 - Assign roles to Slices' and 'Step 2 - Assign to namespaces'. The current step, 'Step 3 - Apply YAML', is highlighted in blue. Below that is 'Step 4 - Finalize'. The main table is titled 'Slice' and lists one item: 'peter-workspace' (TYPE: Application). There are 'View all' and 'Download' buttons. A 'Code Preview' window shows the following YAML code:

```
1 apiVersion: controller.kubeslice.io/v1alpha1
2 kind: sliceRoleBinding
3 metadata:
4   name: peter-workspace
5   namespace: default
6   sliceName: peter-workspace
7   sliceType: Application
8
9 bindings:
10
```

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.

Elastic GPU Service

The screenshot shows the KubeSlice Manager interface with the RBAC module selected. A modal dialog is open for a user slice named 'peter-workspace'. The dialog displays the total number of roles (1) and provides a breakdown of the role type (Imported K8s Roles). The single role listed is 'deployment-role-template' under the 'Imported K8s Roles' category, associated with the user group 'peter-workspace'. The main RBAC page in the background shows other slices like 'ege-gpr-d' and various role management options.

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.

Elastic GPU Service

The screenshot shows the KubeSlice Manager interface with the URL <https://34.19.39.39/#/manage-rbac>. The left sidebar has a 'RBAC' section selected. The main area displays a table of 'Slices' with the following data:

SLICE NAME	TYPE	DESCRIPTION	ROLES	KUBECONFIG
peter-workspace	Application	View all	Edit Roles	Download
egs-gpr-demo	Application	No Roles Assigned		Download

At the bottom, there are links for 'Rows per page: 10' and '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

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.

Elastic GPU Service

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 (selected), 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" and "All GPU Requests". Below that is a search bar with "Enter search" and a "Filter" button. A 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, it says "Showing 1 - 7 of 7 entries" and has a page navigation bar with arrows. On 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 "12:00 PM 8/18/2024".

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

The screenshot shows the KubeSlice Manager interface for the 'peter-workspace' slice. On the left, a sidebar menu includes options like Dashboard, Clusters, Slices, GPU Requests (which is selected), AI Workloads, Users, Namespaces, Node Assignment, Replicate, Resource Quotes, Upgrade License, and Avesha. The main content area displays a table of GPU Requests:

REQUEST NAME	# GPUS	# GPU NODES	PRIORITY	GPU SHAPE	ESTIMATED START TIME	RESERVED FOR	STATUS
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 - 2 of 2 entries". The status bar at the bottom shows "71°F Cloudy", system icons, and the date/time "8/18/2024 2:55 PM".

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

The screenshot shows the KubeSlice Manager interface for creating a GPU request. The main window displays a "Create GPU Request" dialog with the following 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 dialog, there is a section titled "Available GPUs" with a table:

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 large watermark "DRAFT" is overlaid across the center of the page.

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

Elastic GPU Service

The screenshot shows the KubeSlice Manager interface for managing GPU requests. The left sidebar has a dark theme with white icons and text, showing options like Dashboard, Clusters, Slices, GPU Requests (selected), AI Workloads, Users, Namespaces, Node Assignment, Replicate, Resource Quotes, Upgrade License, and Avesha. The main content area has a light blue header with the URL https://34.19.39.39/gpu-request/peter-workspace/worker-1. Below it, a sub-header says "Slice Name: peter-workspace". The "GPU Requests" section contains a table with the following data:

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

At the bottom of the table, it says "Showing 1 - 3 of 3 entries" and has navigation buttons <<, <, >, >>. 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 on the GPR to view request details.

Elastic GPU Service

The screenshot shows the KubeSlice Manager interface for managing GPU requests. On the left, there's a sidebar with various options like Dashboard, Clusters, Slices, GPU Requests (which is selected), AI Workloads, Users, Namespaces, Node Assignment, Replicate, Resource Quotes, Upgrade License, and Avesha. The main area is titled "GPU Requests" and shows a list for slice "peter-workspace". There are three entries in the table:

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

Below the table, it says "Showing 1 - 3 of 3 entries". To the right, under "Request Details", it shows:

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

Other details include GPU Shape: n1-standard-16, Slice: peter-workspace, GPU Mode: Virtual Machine, and Estimated Start Time: 08/18/2024 2:56 PM. The "Actions" dropdown menu is also visible.

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.

Elastic GPU Service

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

The screenshot shows the KubeSlice Manager interface for managing AI workloads. The left sidebar has a dark theme with the Avesha logo at the bottom. The main area is titled 'AI Workloads' and displays a table of workloads. The columns are NAME, DESCRIPTION, TYPE, and CLUSTERS. All workloads listed are of type 'Application' and are assigned to 'worker-1'. The table includes rows for 'dave-training-project', 'egs-demo', 'egs-gpr-demo', 'joe-lm-fine-tuning-project', 'john-lm-fine-tuning-project', 'lm-team-1', 'lm-team-2', 'lm-team-3', and 'peter-workspace'.

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

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

The screenshot shows the KubeSlice Manager interface for the 'peter-workspace' slice. The left sidebar has a dark theme with various options like Dashboard, Clusters, Slices, GPU Requests, and AI Workloads (which is selected). The main content area shows 'AI Model Details' for a workload named 'llm-demo-75fbc46f7f'. A table provides infrastructure details:

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

The status bar at the bottom shows system information including the date (8/18/2024) and time (3:05 PM).

View Pods

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

Elastic GPU Service

KubeSlice Manager

Not secure https://34.19.39.39/workspace/peter-workspace/llm-demo-75fbc46f7f

AI Workloads

Slice Name: **peter-workspace**

/ AI Model Details: llm-demo-75fbc46f7f / Pod Details

Pod Details

POD #	POD/JOB	CONTAINER	NAMESPACE	CLUSTER	
1	llm-demo-75fbc46f7f-hbg5g	text-generation-inference	peter-workspace	worker-1	Go to GPUs

Enter search

Dashboard

Clusters

Slices

GPU Requests

AI Workloads

Users

Namespaces

Node Assignment

Replicate

Resource Quotas

Upgrade License

Avesha

71°F Cloudy

Search

3:05 PM 8/18/2024

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

Slice Name: **peter-workspace**

GPU Details

INSTANCE	NODE IP	GPU NO	GPU SHAPE	GPU LEVEL METRICS	DASHBOARD LINK
~ 1	0	Tesla T4	● ● ● ●	View Dashboard	

GPU Level Metrics

START TIME	TEMPERATURE (°C)	POWER (watt)	MEMORY	GPU UTILIZATION
18/08/2024 12:20	48	28	0	0

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>

Elastic GPU Service

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