Elastic GPU Service (EGS): Overview

Effective resource management of GPUs across Kubernetes Clusters

Introduction

Rapid increase in the development of fine-tuned SLM/LLM models-based GenAI Apps has created a scarcity of specialized GPUs like A100/H100 (or new B100/B200 resources) across the providers/regions, resulting in a need for efficient resource and schedule management of these GPU nodes in one or more clusters to maximize the availability of the GPUs to the Users/Teams and improve the utilization of GPUs.

EGS documents

This document provides an overview of the EGS platform

- For Admin guide please see the documentation on website

- For User guide please see the documentation on website

- For Installation guide please see the documentation on github repo

Keywords and Definitions

- GPU Provision Request (GPR)
  - Resource allocation request for one or more GPU nodes.
- Wait-time-for-deployment
  - Real-time estimation of the wait time for requested GPR node deployment to a Slice VPC.
- EGS - Elastic GPU Service
  - Set of APIs to manage and provision - the GPU provision requests, workspace provision requests, model job deployment requests
  - Platform for effective management of GPU resources across one or more clusters

Table of contents

Elastic GPU Service (EGS)

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

The Elastic GPU Service (EGS) represents a significant opportunity to fill a critical gap in the current landscape of LLM-Ops tools and schedulers. While most tools in the LLM ecosystem focus on managing the lifecycle of large language models, they often neglect the essential aspect of GPU scheduling and resource management across users and clusters. Existing schedulers are similarly limited, primarily addressing job scheduling within in-cluster GPU resources without considering broader resource management needs across multiple users and clusters. This creates a substantial demand among cloud providers, especially those serving large and medium-sized customers, for a robust provision management and automation toolset. EGS can meet this need by making pre-configured GPU nodes and pools readily available for fine-tuning jobs, thereby enhancing GPU utilization and driving monetization. Additionally, EGS empowers cloud providers to offer a premium, white-glove service to their larger customers, while also providing a self-service portal that simplifies and streamlines GPU resource management for a broader user base.

Concepts

- GPU cluster time-slicing
- EGS Slice VPC
- Dynamic GPU provisioning in a Slice VPC
- GPU provision requests management
- GPU inventory schedule management
- AI workload/GPU observability
- GPU monitoring and remediation
- Multi-Cloud multi-cluster EGS Slice VPC
- Insights and trends from GPR queues
- Workspace provision requests
- Dynamic Node pools and nodes

GPU Cluster Time Slicing: Enhancing GPU Utilization Across Clusters

As organizations increasingly rely on GPU-accelerated computing for a variety of applications, effective GPU resource management has become crucial to maximizing performance and minimizing waste. One of the most promising strategies for achieving this is GPU cluster time slicing. This approach allows for the dynamic allocation and reallocation of GPU resources across multiple users and workloads, ensuring that GPU clusters are utilized more efficiently.

Time slicing enables the effective use of GPU resources within one or more clusters by abstracting the GPU infrastructure and dynamically provisioning GPU resources, nodes, and networks. This abstraction layer allows GPU resources to be associated with specific user slices (VPCs) and namespaces, enabling seamless provisioning and removal of GPU nodes from these slices (VPCs). By automating the preparation and setup of GPU nodes, including the configuration of GPUs, plugins, operators, custom resources (CRs), and networks, the time and effort required by Site Reliability Engineers (SREs) for manual provisioning are significantly reduced.

One of the key benefits of GPU cluster time slicing is the ability to manage a large set of GPU nodes across multiple clusters, providing users with access to available resources wherever they are needed. This approach not only improves GPU utilization but also ensures better isolation through GPU slices, which are specifically tailored to user needs. Comprehensive dashboards further enhance the user experience by offering detailed insights into slices, users, workloads, GPUs, and associated metrics. These dashboards include features such as hot-spot detection, event monitoring, and alerts, allowing for constant monitoring and proactive management of GPU resources.

Moreover, constrained GPU resource management and scheduling are critical components of time slicing. This includes managing GPU provision requests, scheduling for each GPU node, and predicting wait times for deployment. By automating the lifecycle of GPU resource and pool provisioning, organizations can improve overall utilization, dynamically configure and reconfigure nodes and networks, and effectively manage GPU resources across one or more Kubernetes clusters. The ability to insert or remove nodes and pools from slices, coupled with MIG/vGPU plugin reconfiguration, ensures that GPU clusters remain agile and responsive to changing demands, ultimately leading to reduced GPU wastage and increased efficiency across the board.
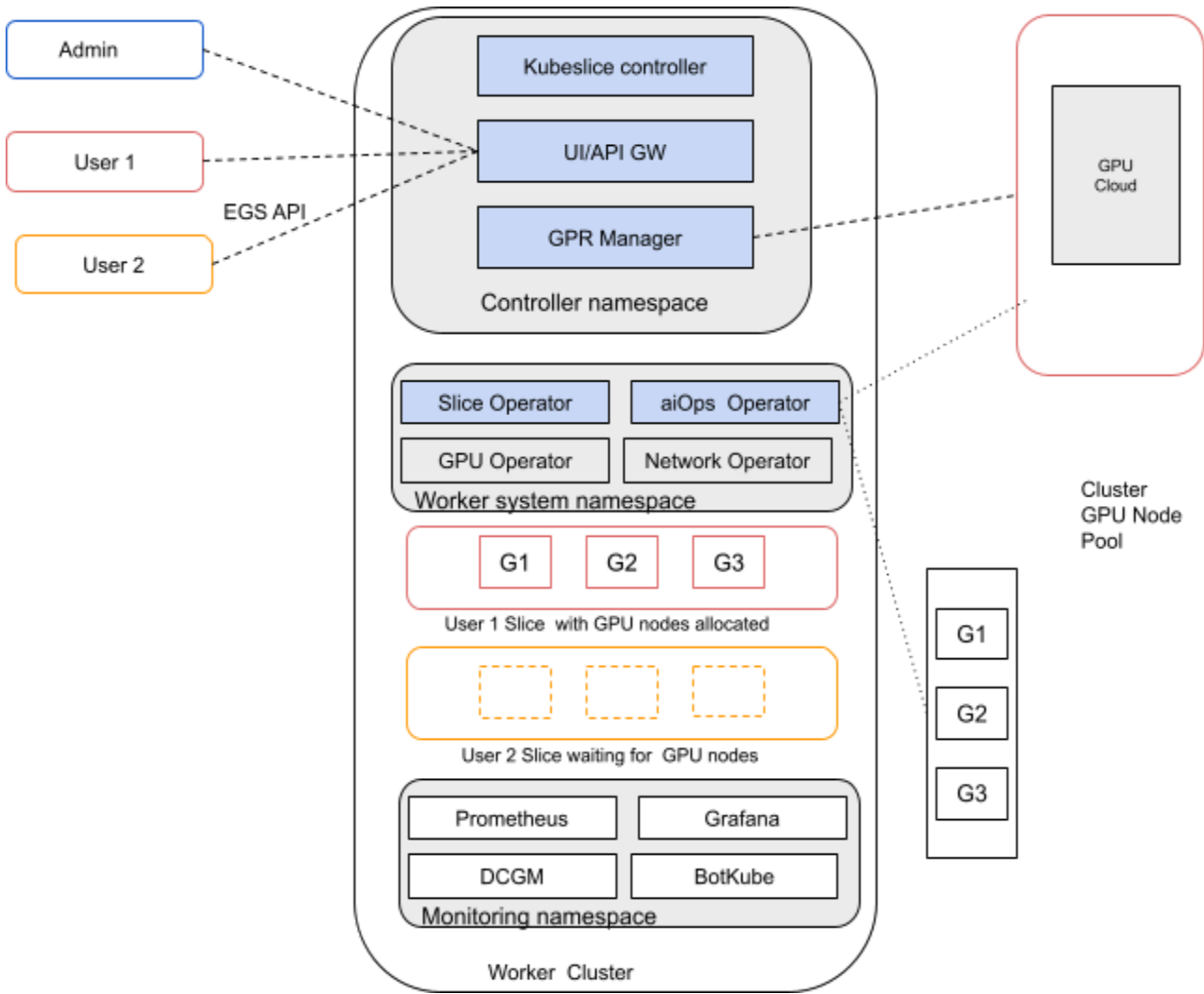
Architecture

Single cluster deployment

Figure 1: Single Cluster EGS reference model

Multi-cluster deployment
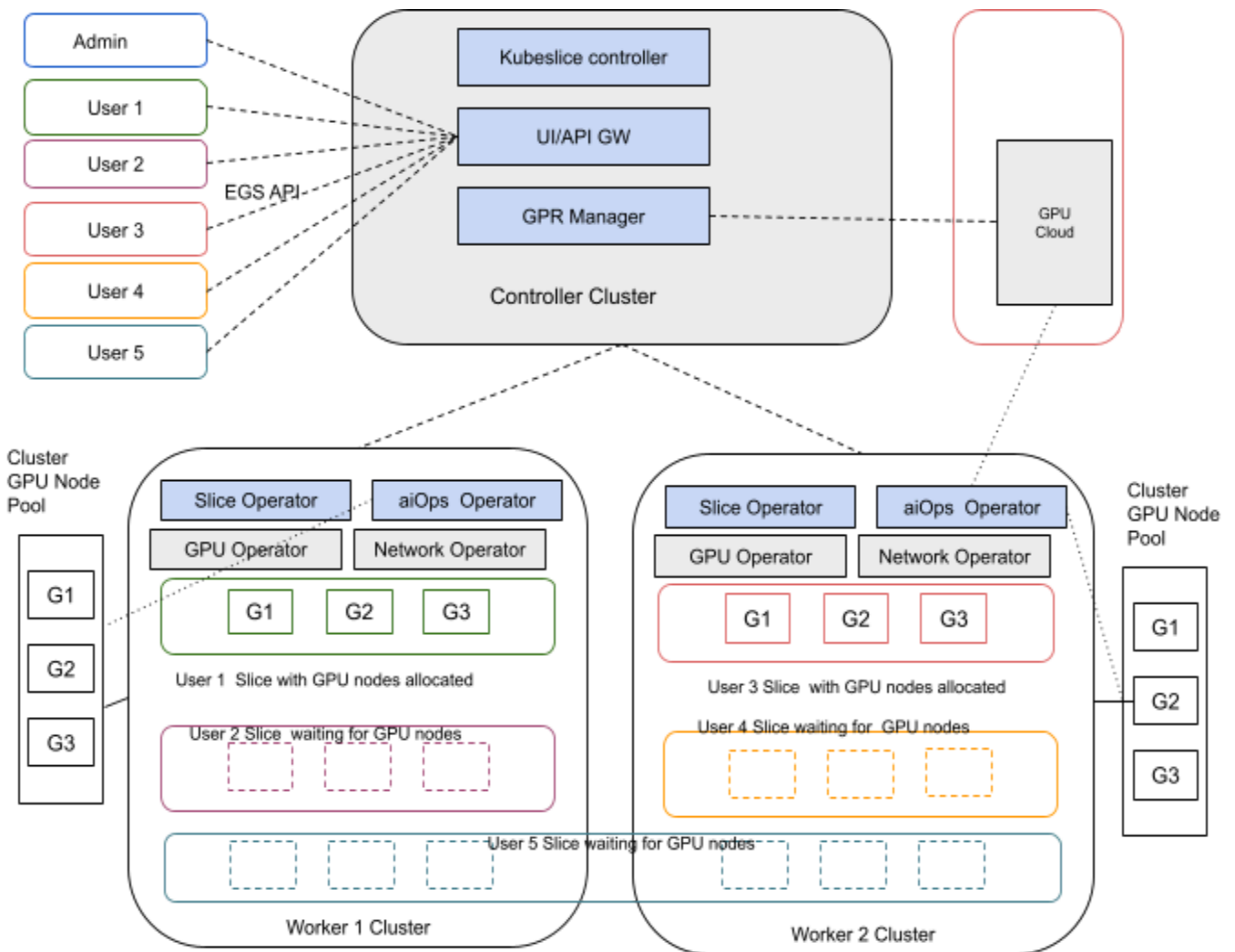
Figure 2: Multi-cluster (multi-cloud) EGS reference model

EGS Slice VPC

EGS Slice VPC is a logical boundary for a User (or a team) workspace. The Slice can be viewed as a VPC that spans one or more Kubernetes clusters. The Slice VPC workspace will be associated with one or more namespaces where user(s) can deploy their AI workloads. Users can deploy non AI workloads (CPU workloads) any time. AI workloads that need GPUs need to have GPU nodes provisioned in the Slice VPC. Without the GPU nodes provisioned in the Slice VPC the AI workloads (Pods/etc) will go

into the pending state - waiting for the GPU resources. Users need to create GPU request(s) for GPU nodes to be provisioned to the slice.

GPU Request

Users create GPU requests for the Slice using the EGS User portal. The EGS control plane manages the queue of the GPU requests across the Slices/Users (teams) and Clusters. During the GPU request creation time User will be given an estimated provisioning time by EGS control plane (backend).

Typically, for constrained GPU resources (like A100/H100, etc.) in most (org) clusters, there will be a wait time to get the GPU nodes provisioned for the User.
A User can create one or more GPU requests. Each request will have an estimated provisioning time. Users can specify the GPU shape, number of nodes, priority, model params, network requirements, and reservation duration while creating a GPU request. Users will be prompted with the estimated wait time for the GPU request. If the User is fine with the estimated wait-time, he can proceed to submit the request.

If the cluster has more than one GPU shape type (multiple node pools), a User can query the wait time across all the GPU shapes available on the cluster. EGS will show the wait times for various GPU shapes and the User can pick the one that fits the job that he wants to deploy.

Users can edit or delete a request before it is provisioned. Users can early-release a provisioned GPU request to remove the provisioned GPU nodes from the Slice VPC.

GPU requests management

The EGS control plane GPU requests (GPR) manager manages the life cycle of the GPU requests. GPU requests are inserted into the pending requests queue based on a number of considerations. Enqueue logic determines the position and the estimated wait time for the request. The enqueue logic takes into account - the inventory available, allocations, priority, fairness, and other requests in the queue to place the request in the queue.

The EGS Queue Manager (QueueMgr) component maintains a queue of pending GPR requests based on an integer priority number. It provides a rich and efficient set of APIs that other components in the EGS system could use to interact with the QueueMgr. The

GPR manager in the EGS system moves GPR requests in and out of the queue while managing the GPU resource inventory and its allocation.

Priority Queue

In EGS, GPRs with different priorities could be admitted and stored in the queue, and GPRs should be processed in the order of their priorities, meaning that the higher priority GPRs are serviced first before lower priority ones. The Admin in the EGS system has the ability to increase the priority of any GPR present in the queue. The QueueMgr is expected to store the GPRs in a way that extracting and rearranging them based on their priorities is performed efficiently and does not cause processing delays in the system.

A priority queue would be used to store GPRs based on the priorities assigned to them. The priority queue performs time and space efficient operations to get, insert, update, delete and rearrange elements in the queue.

Dynamic GPU Provisioning in a Slice

The GPR manager periodically checks with Inventory and Queue managers to get the next GPR allocation for provisioning. Once the GPR is allocated the resources needed to provision the request will be identified. The GPR manager works with the worker cluster EGS component - aiOps Operator - to complete the provisioning. The GPR manager creates appropriate CRs for the aiOps operator to complete the provisioning of the GPU nodes into the worker cluster Slice VPC.

Note:
In single cluster EGS deployments, the EGS control plane components and EGS worker components (aiOps Operator) are deployed in the same cluster in different namespaces.

The EGS worker component (aiOps operator) is responsible for the life-cycle management of GPR provisioning in the cluster. It manages the entry and exit of the

GPU nodes from Slice VPC. It constantly monitors the GPU nodes, node metrics -like temperature, power and utilization. It generates events for error threshold checks for important metrics.

During entry GPU provisioning, it is responsible for the node prep, network prep and latency check. During pre-provisioning, it makes sure the nodes are working properly, networked properly, runs NCCL tests to test the latency. It constantly monitors the (NCCL) latency check across the nodes in the Slice VPC. The nodes are added to the Slice VPC by provisioning appropriate affinity/anti-affinity, and so on. to the Slice namespaces. Once the GPUs are provisioned to the Slice, Users can run AI workloads requesting GPUs in the Slice associated namespaces. Any workloads (pods/jobs/etc) pending will then get the GPU resources requested and the scheduler moves them from pending to running state. EGS sends notifications (on Portal and Slack) when the provisioning is complete. K8s events are generated as well.

A Slice VPC can have only one GPR provisioned at any given time.

The aiOps Operator generates K8S events and Slack notifications (if configured) during the life of the provisioned GPR. This gives users a sense of how much more time is left with this GPR provisioning.

At the provision exit time (based on the reservation duration and start time) the GPU nodes will be removed from the Slice. The nodes will be drained at the exit to prep them for using them for allocation to another Slice VPC. During exit any running workloads (pods/jobs/etc.) that are still using the GPUs will restart and go to pending state. It is expected that AI workloads have periodically generated checkpoints and in most cases the training or fine tuning (or other function) is complete and is safe to remove the GPU nodes from the Slice.

GPU Inventory Schedule Management

EGS inventory schedule management service maintains details about the GPU nodes and GPU devices and their attributes. It also maintains the network and other related configuration information as well. It also maintains the detailed schedule of the GPU nodes. It provides a number of APIs to the GPU requests manager (and other services) to get the nodes, GPU details and their schedules.

EGS keeps track of all the GPU nodes across all the clusters that are associated with EGS projects. Worker aiOps operator constantly updates the node details to the inventory schedule management service.

EGS inventory schedule management provides information to get a good real-time wait time estimate for a GPU request.

AI Workload/GPU Observability

EGS User portal offers a detailed view of AI workloads that are running in the User workspace (namespaces). In addition, a user-focused dashboard shows key metrics across Users Slices, workloads, and GPUs. It provides detailed events and notifications information for various alerts for GPUs, workloads, GPU requests, and so on.

The **Model Details** page shows workload model details, the GPU infrastructure committed for the model, LLM training or LLM fine-tuning, and other jobs. It also shows the GPU with high power consumption, high temperature, and average utilization values. For training or fine-tuning jobs with a lot of workers/GPUs committed, this page offers a quick view of the GPU hotspots.

The **Pods Details** page shows the pod and associated one or more GPU devices details. A GPU dashboard link shows the time-series data specific to the GPUs that are used by the Pods. This helps Users to get to the GPU DCGM metrics for the Pods.

The GPU table shows all the GPUs that are used by a model (LLM training/fine-tuning job) in one table. The table is sorted to show the high power/temperatur GPUs at the top to get a quick view of the GPU hotspots. The tables provide filters/search options to narrow down the GPUs.

GPU Monitoring and Remediation

EGS Worker aiOps operator constantly monitors the key metrics such as power, temperature, utilization for all the GPU nodes across all the Slice VPCs. It generates events, alerts, and notifications for any metrics threshold violations. It constantly monitors for the configuration (or desired state) drift, generates alerts, and works to bring the Slice VPC back to the desired state.

The constant monitoring looks for GPU errors (or error trends) to proactively remediate a potentially failing GPU node. The GPR node-replacement workflow will be used to remove a node and add a new node to the Slice VPC with minimum disruption to enable Users to continue with the training or tuning jobs.

Multi-Cloud, Multi-Cluster Slice VPC

The EGS Control Plane provides workflows to register one or more public cloud (or edge or data center) clusters with a project. EGS Admins then can create a slice that spans across multiple clusters. Users can take advantage of a workspace (namespaces) that spans across the cluster and can decide based on the requirements and GPU provision requests in any of the associated clusters.
EGS provides GPU resource management and visibility across multiple clusters.

Workflows

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, who uses the EGS User Portal to create and manage the life-cycle of GPU provisioning requests for User's Slice(s). He uses the portal to get deep visualization of the AI workloads and associated GPUs metrics and other data.

Admin Workflows

Installation

EGS provides a combination of Helm charts, CLI and scripts for easy installation procedure. For Installation guide please see the documentation on github repo
For Admin guide please see the documentation on website

Day 1 Operations

1. Create Project
    i.   During installation, a default project workspace will be created.
    ii. A single project namespace can be used to manage one or more clusters across one or more users/teams.
    iii. In a project, one or more slices can be created.
    iv. A user can be associated with one or more Slices.
    v.   Each slice provides a workspace for a user or a team.
    vi. 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.
2. Admin registers clusters

      i.   For single cluster deployments, during installation, an Admin registers a cluster with an EGS control plane.

      ii.   For multiple clusters

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

      i.   Creates a Slice (with team name or user name)

      ii.   Assigns one or more namespaces to a Slice

      iii.   Adds User (or team) to the Slice

      iv.   Provisions Slice RBAC for User

          1. Adds role and role bindings for the Slice namespaces

      v.   Downloads KubeConfig for the Slice

          1. Kubeconfig will have SA token to access the namespaces

      vi.   Sends the KubeConfig to User

          1. User can access the namespace(s) on the cluster

          2. User RBAC scope is associated namespace

4. Admin can add/remove namespaces to Slice
5. Admin can add/delete Slices
6. Admin can add/remove/update the Slice RBAC and regenerate the KubeConfig for User - for cluster access.
7. Dashboard
8. GPR tables, queues view and actions

      i.   Admin can view the GPR queue

      ii.   Admin can change the priority of GPR and move a GPR up/down the queue

      iii.   Admin can edit, delete, or early-release a GPR

9. Inventory table views

Day 2 Operations

1. Use dashboard to monitor key metrics related to the GPU pool

      i.   GPRs and GPU utilization, allocation, Errors, Alerts

2. Manage GPR queue

      i.   Adjust priority of a GPR if needed.

      ii.   Approve GPRs that are in pending approval due to policy:

          1. Exceeding the quota for the Slice

          2. Exceeding the GPU nodes limit per GPR (request for large number of GPU nodes)

             📒 **Note**: Most GPRs are auto-approved

      iii.   Early release a provisioned GPR

1. Early release of a GPR will remove the associated GPU nodes from the slice
3. Adds/Deletes Slices for Users

Additional day-to-day operations
1. Adjust GPR priority, GPR eviction - to make room for high priority GPR
2. Inventory schedule table views
3. Approve pending GPRs
4. View GPR pre-provisioning checklist/verification (visibility into the GPR pre-checks and logs)
   ○ 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
5. View GPR post-exit operations (visibility into GPR post-checks and logs)
   ○ Node draining, Network check, NCCL test checks
   ○ PVC/PV check, Labels check
6. Enable provisioning of dynamic node pools and nodes for Slice VPCs.

User Workflows

User (Data Scientist or ML engineer or Team) uses EGS portal to view User's Slices to create and manage the GPU provision requests (GPRs), and view the AI workload and associated GPU details/metrics.

For User guide please see the documentation on website

Getting a Slice Provisioned

A User (or a team) can have one or more Slices (workspaces). Slices can be single cluster scoped or can span across multiple clusters

User works with the EGS Admin to get a Slice created for him/her or team. User provides name, email, Slice name, and one or more namespace names to the Admin to

create a Slice.

Admin creates the Slice and associates the user name and namespaces with the Slice. Admin creates appropriate RBACs and sends an Access token (if IDP is not enabled on the cluster) for User to access the EGS User Portal. Admin also sends KubeConfig for the worker cluster to access the namespaces with KubeCtl and other CLI tools.

EGS User Portal

Users can access the EGS portal with *Access* token (or with IDP credentials if IDP is enabled on the cluster).

EGS Portal enables workflows to manage the life-cycle of GPU provision requests (GPRs), deep GPU observability for User AI workloads.

GPU Provision Requests

By default, no GPUs will be assigned to User's Slice VPC. To run AI workloads (in the namespaces that are associated with the Slice) that require one or more GPUs, User needs to use the Portal to create a GPU provision request.

GPR features:

- Users can create one or more GPU provision requests
- Only one GPR will be provisioned in to the Slice at a given time
- GPR has strict entry and exit times for GPU nodes from Slice VPC
- Isolation of GPU nodes per Slice VPC
- Other Slices (or Users) cannot use the GPUs allocated to the User's Slice VPC inadvertently
- Self-service mechanism for GPU provision requests
- Visibility into wait-time for GPUs
- Users can delete/edit GPRs before they are provisioned
- Users can early-release GPR if they no longer need the GPUs in their Slice VPC

GPU Observability

Users can view AI workloads and associated GPU details that are running in their slice namespaces (workspace).

EGS provides highly granular visualization for every AI workload and associated GPUs:

- AI workloads lists model, model configuration, and infrastructure committed for the workload (LLM training or fine-tuning job).
- Visibility into high power usage GPU, high temperature GPU
  - Generates alerts on high power/utilization levels
- Visibility into GPU metrics - dashboards for Users AI workloads parameters/GPU metrics