Rafay Platform

GPU PaaS Deployment Guide

RAFAY

Version History

Version	**Date**	**Updates**
0.1	July 3, 2024	Initial version based on Rafay docs
1.0	Nov 15, 2024	Standalone version

TABLE OF CONTENTS

Introduction

This document gives a brief overview of Rafay's GPU Cloud PaaS platform and describes the onboarding and deployment process for GPU Cloud Service Providers (CSP). It assumes that the CSP has already deployed Nvidia-certified infrastructure following Nvidia's NCP reference architecture documentation.

They can use this document to deploy the Rafay platform as a fully white-labeled solution and provide GPU Cloud services to their customers. CSPs can also integrate the Rafay GPU PaaS Platform capabilities into their user facing portals and systems leveraging the Rafay platform's API.

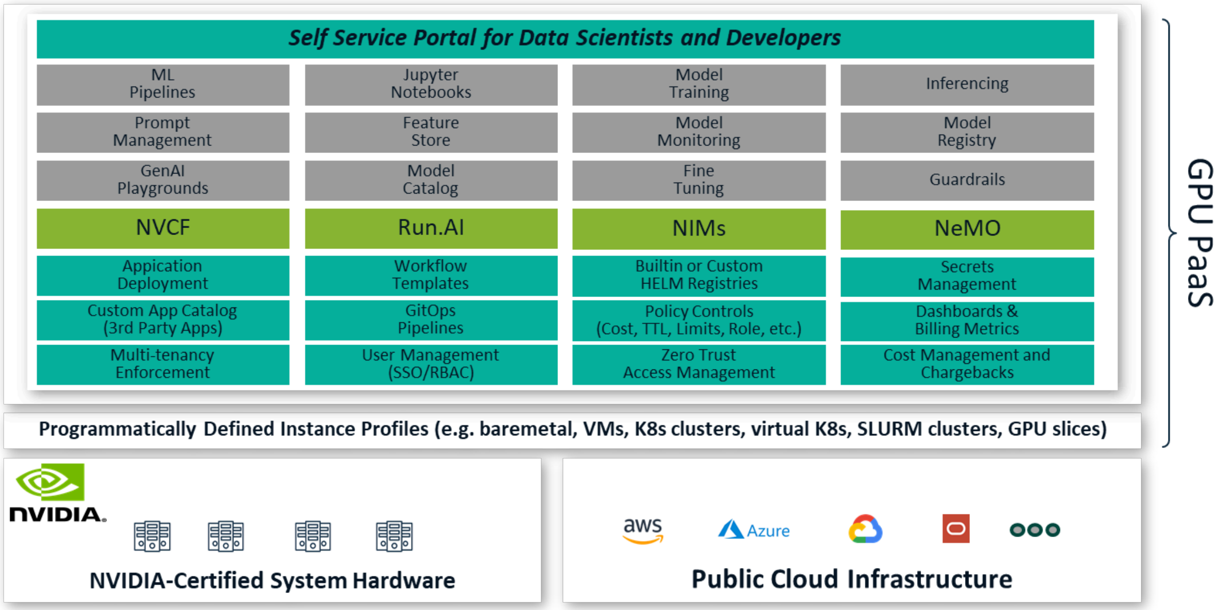
Rafay Platform's GPU PaaS Offering

The Rafay Platform is a multi-tenant platform that enables CSPs to deliver GPU, compute resources, AI-ML/Gen tools and services to their customers on their GPU infrastructure in a Platform-as-a-Service (PaaS) fashion. The Rafay platform allows CSPs to partition their GPU infrastructure and allocate it to multiple organizations.

With Rafay, CSPs can offer various flavors and sizes of GPU and compute resources (e.g. bare metal nodes, VMs, Kubernetes Clusters, Virtual Kubernetes clusters with one or more GPUs, and fractional GPUs). They can also provide their users with turnkey services (such as Jupyter notebooks), AI/ML tools (such as Kubeflow for MLOps and Ray) and GenAI playgrounds for LLM evaluation, fine-tuning, and inference.

Apart from the capabilities above, the platform provides integrated capabilities for user management, single sign on (SSO), multi-tenancy, billing, cost management and integrated dashboards. CSPs are provided with an operator portal that they can use for tenant administration, infrastructure management and operations.

Rafay's multi-tenant GPU PaaS solution delivers all these key capabilities as a turnkey offering to accelerate GPU Cloud deployments and to provide a competitive GPU Cloud offering.



Self Service Portal for Data Scientists and Developers

ML Pipelines	Jupyter Notebooks	Model Training	Inferencing
Prompt Management	Feature Store	Model Monitoring	Model Registry
GenAI Playgrounds	Model Catalog	Fine Tuning	Guardrails
NVCF	Run.AI	NIMs	NeMO
Appication Deployment	Workflow Templates	Builtin or Custom HELM Registries	Secrets Management
Custom App Catalog (3rd Party Apps)	GitOps Pipelines	Policy Controls (Cost, TTL, Limits, Role, etc.)	Dashboards & Billing Metrics
Multi-tenancy Enforcement	User Management (SSO/RBAC)	Zero Trust Access Management	Cost Management and Chargebacks

Programmatically Defined Instance Profiles (e.g. baremetal, VMs, K8s clusters, virtual K8s, SLURM clusters, GPU slices)

GPU PaaS

NVIDIA-Certified System Hardware

Public Cloud Infrastructure

As shown above, the Rafay GPU PaaS platform works seamlessly with CSP provided infrastructure. It also works with all major cloud provider infrastructures for additional capacity requirements (e.g. burst to cloud).

GPU Cloud providers should be able to provide various flavors of GPU and compute instances, such as virtual servers, Kubernetes clusters, Virtual Clusters, Virtualized GPUs, etc., to their customers. Rafay Cloud PaaS provides foundational capabilities such as multi-tenant administration, SSO/RBAC, zero trust access, GitOps, Policy management, dashboards, and several other critical capabilities for enterprise deployments. The platform also provides many out-of-the-box integrations, including container registries, helm repositories, data integrations, Hugging Face, Nvidia services, etc., necessary for enterprise application development and deployment.

Concepts

The Rafay GPU Cloud PaaS platform is broadly categorized into three functional areas:

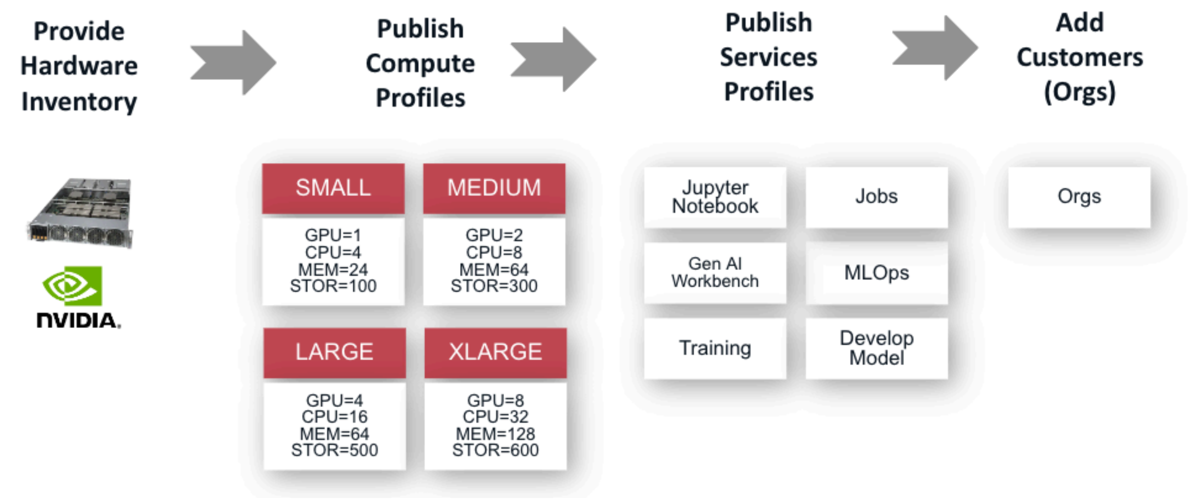
- Platform Operations
- Tenant Administration
- End User

Platform Operations

This is primarily aimed at the operations personnel of the CSP. It provides capabilities to manage their infrastructure and customer configurations. With this part of the platform, the CSP will perform the following:

- Define the infrastructure inventory,
- Specify the SKUs and managed services they wish to provide, and
- Make them available to all/select customers through this interface.

CSPs can also perform administrative tasks such as adding new customers (i.e. tenants) to the platform, adding premium SKUs for certain customers etc. The image below shows the high level steps that a typical CSP has to follow to become operational.


Provide Hardware Inventory
Publish Compute Profiles
Publish Services Profiles
Add Customers (Orgs)
NVIDIA.
SMALL
GPU=1 CPU=4 MEM=24 STOR=100
MEDIUM
GPU=2 CPU=8 MEM=64 STOR=300
LARGE
GPU=4 CPU=16 MEM=64 STOR=500
XLARGE
GPU=8 CPU=32 MEM=128 STOR=600
Jupyter Notebook
Jobs
Gen AI Workbench
MLOps
Training
Develop Model
Orgs

Inventory

The operations team will provision and manage the underlying hardware, servers and networking in the CSP’s datacenter. They will need to maintain and provide the inventory of resources to the Rafay Platform so that it can use it to provision resources such as Kubernetes clusters, storage etc.

Profile Catalog

Rafay provides a number of default “compute” and “service” profiles that the CSP can extend, customize and offer to their customer orgs as SKUs. In addition to the default templates provided by Rafay, CSPs can create and manage “custom” compute and service profiles.

Once published, these SKUs are made available to end users via a catalog style experience. Users just need to select a SKU and deploy it to get access to it.

Compute Profiles

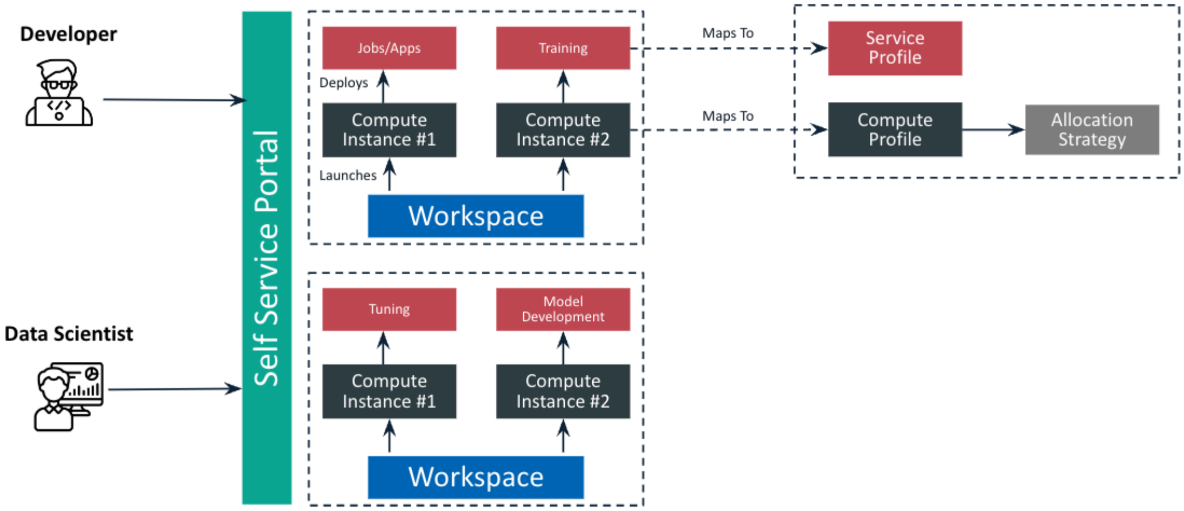
A compute profile is a "predefined configuration" specifying compute resources such as CPU, GPU, RAM, and storage. For example, a compute profile may comprise 1 GPU, 1 vCPU, 16 GB memory and 100 GB storage.

End users will use the compute profile as a template to launch compute instances that they can then use for their work.

Service Profile

A service profile encapsulates a complex software stack that can be deployed by the end users in just a click. For example, data scientists can deploy a Jupyter notebook with Tensorflow and CUDA into a GPU based compute instance.

The image below showcases the relationship between a service profile and the service instance operating in a workspace. It also highlights the relationship between the compute profile and the associated compute instance.


Developer
Data Scientist
Self Service Portal
Jobs/Apps
Training
Deploys
Compute Instance #1
Compute Instance #2
Launches
Workspace
Tuning
Model Development
Compute Instance #1
Compute Instance #2
Workspace
Maps To
Maps To
Service Profile
Compute Profile
Allocation Strategy

Billing

CSPs are expected to annotate compute and service profiles with cost metadata. The metadata will comprise “key:value” pairs and will typically comprise the following information.

- Currency (e.g. US $)
- Value (e.g. 4.23)
- Time (e.g. per hour, per month etc)

Instances spawned by users will be automatically tagged with cost metadata in the Rafay platform’s database. For billing purposes, CSPs can programmatically retrieve both “cost metadata” and “running time” for both compute and service instances.

CSPs can use this data in their existing billing systems to calculate how they wish to charge their customer. For example, assume the “small” compute profile costs $1/hr and the user operated an instance of this for 30 days (i.e. 720 hours). When the CSP’s billing platform queries Rafay for details about the instance, it will return back “$1/hr and 720 hours run time”.

The CSP’s billing system will then use the data to calculate the charges (i.e. $1 * 720 hours = $720) for the time period. The CSP’s billing system can also apply special discounts and promotions in their billing system.

On-Demand vs Reserved Instances
This approach also allows CSPs to support SKUs with both On-demand and Reservations.

Prepaid vs Postpaid Billing
This approach ensures that the CSPs have complete control and flexibility over how they would like to bill their customers i.e. via a prepaid or postpaid billing model.

Tenant Administration

Multi-Tenancy

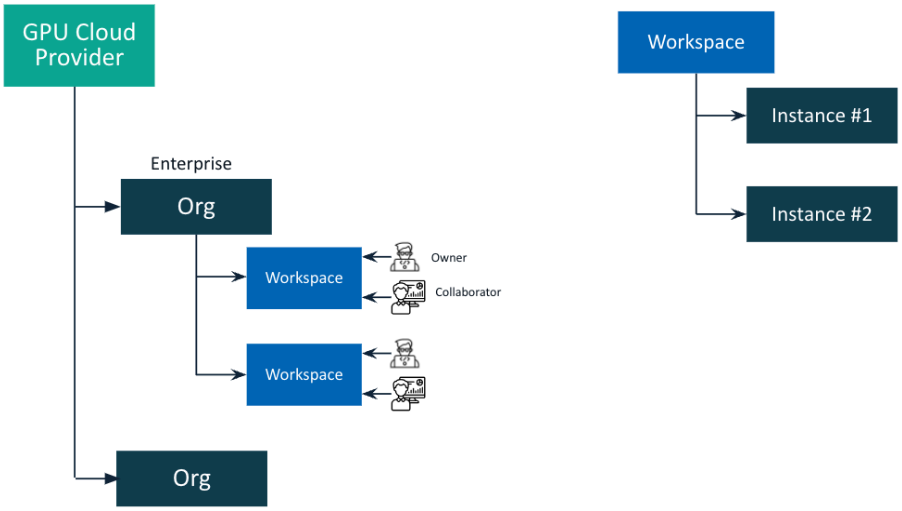
The Rafay platform has deep support for multitenancy. A CSP can manage multiple customers in the same Rafay platform.

Every customer is provided with at least one Org (short for “organization”) is a dedicated instance of Rafay. Each Rafay org represents an isolated tenant with its own secure environment. In the image below, you can see that the CSP is managing “multiple” customers with their own dedicated Orgs. Every customer’s org in turn can host multiple tenants underneath called workspaces.

In the example below, you can see that the CSP is managing two separate customers in their Orgs and the first Org is hosting multiple workspaces in it with multiple users accessing it.

Also, note that inside a workspace, the end user can launch and use “multiple” instances. The maximum number of instances that can be launched by an end user can be controlled via limits specified in a policy.

Note
There are no limits to the number of Orgs in the Rafay platform. In addition, there is also no limit to the number of workspaces that can be managed in an Org.



In order to be cost efficient, instances from multiple users and customers can be deployed on shared, multi-tenant Kubernetes clusters operating on the CSP’s infrastructure. As a result, it is critical to ensure that there is extremely strong “isolation” implemented by default to ensure that containers deployed by a user cannot compromise another user or the CSP’s infrastructure.

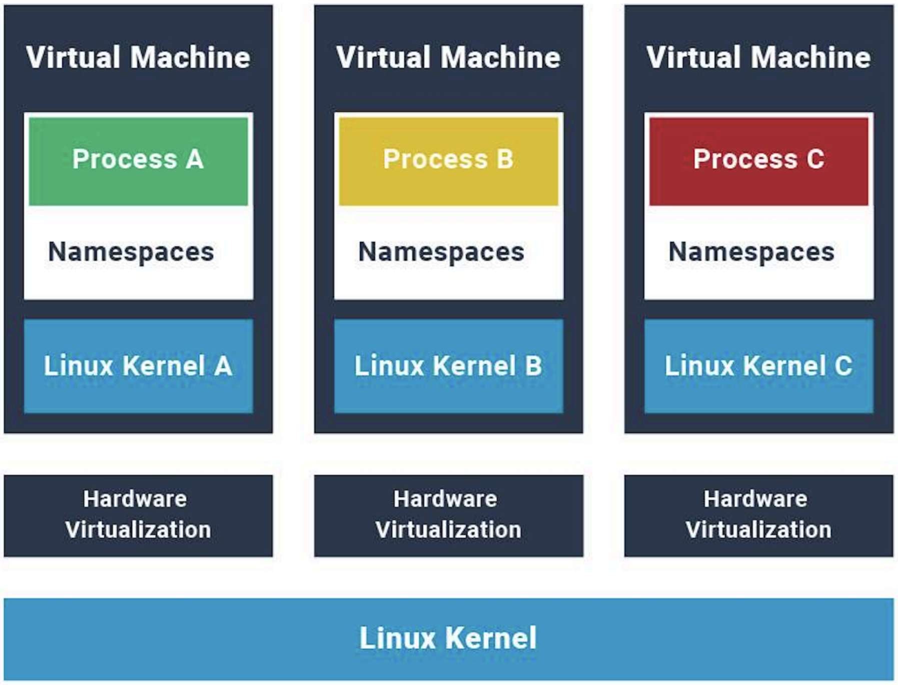
For every instance in a customer Org, a number of critical security controls are automatically implemented and enforced to ensure that risks associated with threats such as lateral escalation etc can be blocked. The table below describes the various threats and associated controls that are implemented by default to secure against the threat.

#	Threat Vectors	Security Controls
1	Prevent malicious containers from lateral escalation inside the cluster.	Isolated Containers
2	Prevent malicious containers from using the data center network for lateral escalation to other hosts/services in the data center.	Network Policy
3	Prevent malicious containers from escaping the container by becoming root etc.	Cluster Policy
4	Prevent users from being able to access resources that are not theirs.	Role based Access Control (RBAC)
5	Ensure only authenticated and authorized users can access resources	Secure Remote Access
6	Prevent users from using more resources than allocation	Resource Quotas
7	Ensure there is an immutable and centralized audit trail for every action	Centralized Audit Logging

Isolated Containers

Containers operating on Kubernetes clusters share the Linux kernel (i.e. default behavior). The image below shows three containers using the same Linux kernel. For shared, multi-tenant Kubernetes clusters especially with users that the service provider cannot control, there is always the risk of container escapes.

Kata containers are a secure alternative to address the issue above. This is a container runtime technology designed to provide the security advantages of virtual machines (VMs) while maintaining the lightweight performance and agility of containers. The image below visually shows how Kata containers are different from a regular container runtime.



VM-Level Isolation
Kata Containers run each container inside a lightweight virtual machine, providing strong isolation between containers. This VM-level isolation mitigates the "noisy neighbor" problem and reduces the risk of security breaches spreading from one container to another.

Protection from Kernel Vulnerabilities
Since each container has its own kernel instance inside the VM, Kata Containers protect against vulnerabilities in the host kernel. Even if a container is compromised, it cannot directly affect the host or other containers running on the same host.

The Rafay platform automatically enforces Kata containers by default for all workloads deployed onto the user's instance. Rafay deploys an admission controller on the host cluster that enforces the use of the Kata runtime in the user's allocated instance.

Network Policy

In a multi-tenant environment, different teams, departments, or organizations may share the same Kubernetes cluster. Network policies help enforce isolation between tenants by controlling which pods can communicate with each other across different namespaces. Without network policies, there could be unintended or unauthorized cross-tenant communication, potentially leading to data leaks or security breaches.

Network Policies are a mechanism to control network traffic flow within and from/to Kubernetes clusters. The Rafay platform makes sure that all namespaces are locked down with a default network policy that blocks all resources in the namespace to exchange network traffic with "other namespaces" and "outside the cluster".

Cluster Policy

In a Kubernetes based multi-tenant environment, cluster policies are crucial for enforcing security, resource management, and compliance across different tenants. It is critical to enforce security best practices for pods, such as running as non-root, restricting the use of privileged containers, and controlling the use of hostPath volumes. This ensures that tenants cannot deploy potentially insecure workloads that might compromise the cluster.

The Rafay platform automatically enforces a default cluster level policy (based on OPA Gatekeeper) to control what users can/cannot do on the cluster. This also ensures that the clusters are always in compliance with centralized policies. Note that the cluster policies are closely coordinated with network policies to ensure there is defense in depth and completeness.

Role based Access

Kubernetes RBAC is a critical security control to ensure that users and workloads only have access to resources required to execute their roles. By default, end users only have access to the virtual cluster that is deployed into the namespace where it operates. The user is automatically mapped to a "ClusterRole" for the virtual cluster.

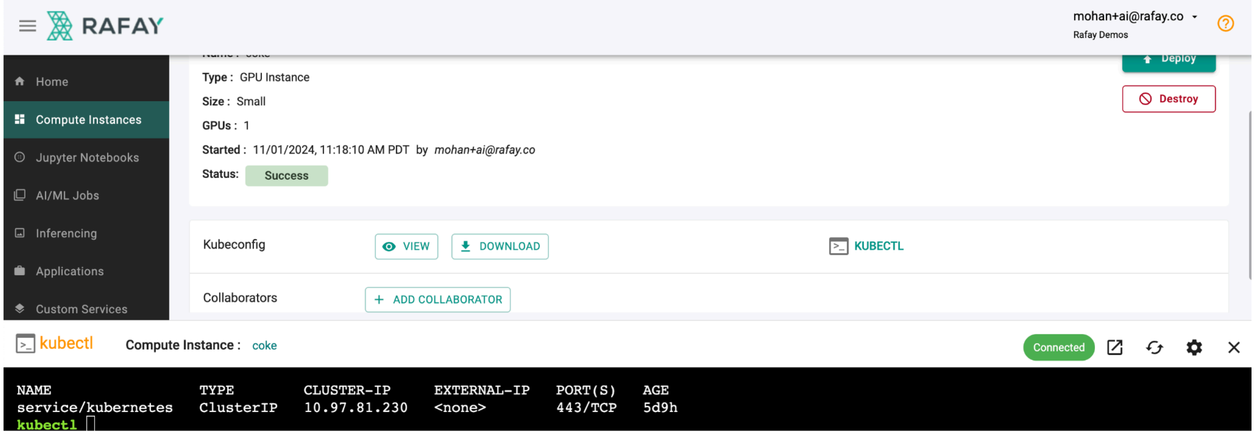
Note that although the virtual cluster is operating inside a namespace (with a role) in the host cluster, the user is not provided access to the namespace itself.

Secure Remote Access

Secure remote access is critical for enhancing security, governance, and operational efficiency. Here are some reasons why it is required:

- Enforce strict identity verification before granting access. Every request is authenticated and authorized, ensuring that only legitimate users with the appropriate permissions can interact with the cluster.
- No user or device is inherently trusted, even if they are within the network perimeter. This reduces the risk of insider threats or compromised credentials leading to unauthorized access.
- Allows administrators to enforce the principle of least privilege by defining granular roles and permissions. Users are granted only the minimal level of access necessary for their tasks, which limits the potential impact of a compromised account.
- Comprehensive and centralized logging of all actions performed within the cluster. This is crucial for meeting compliance requirements, as it ensures that all access and modifications are fully traceable and auditable.

To ensure highest levels of security, with the Rafay platform, all users are required to centrally authenticate using the configured Identity Provider (IdP). Once successfully authenticated, an ephemeral service account for the user is federated on the remote cluster in a Just in Time (JIT) manner. Users are provided with the means to remotely access their instance and perform Kubectl operations using the Kubectl CLI or an integrated browser based shell.



Type : GPU Instance
Size : Small
GPUs : 1
Started : 11/01/2024, 11:18:10 AM PDT by *mohan+ai@rafay.co*
Status: Success

Kubeconfig

Collaborators

Compute Instance : coke

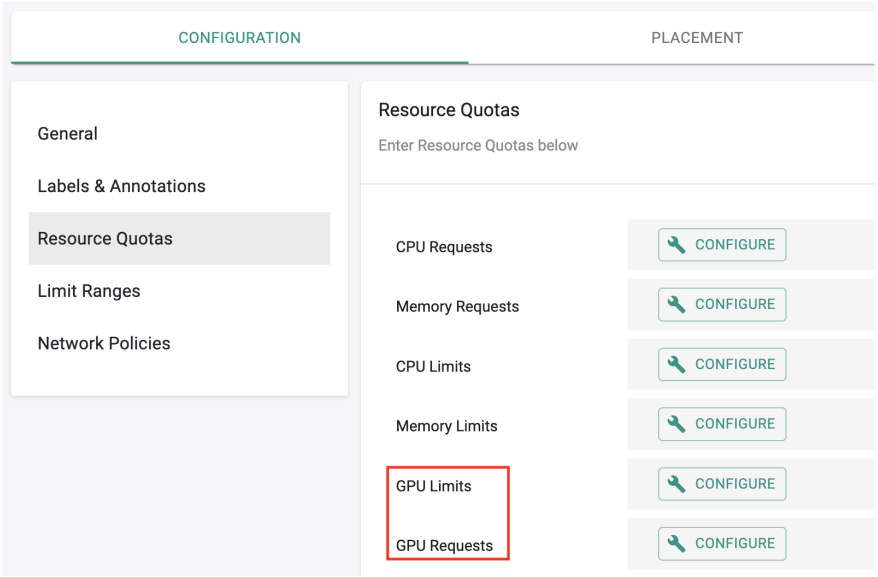
```
NAME                 TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.97.81.230   <none>        443/TCP   5d9h
kubectl
```

Resource Quotas

Resource quotas are essential for managing and controlling the allocation of resources such as CPU, memory, and storage. Here's why they are necessary:

- In a multi-tenant environment where multiple users or teams share the same Kubernetes cluster, resource quotas ensure that no single namespace or application can consume all available resources, which could lead to resource exhaustion and destabilize the entire cluster.
- Quotas help ensure a fair distribution of resources among different teams or applications, preventing situations where a single team could monopolize resources, leaving others with insufficient capacity.
- By setting resource quotas, administrators can prevent users from over provisioning resources (e.g., requesting excessive CPU or memory) that might not be fully utilized, leading to unnecessary costs.
- Quotas encourage efficient use of resources, as users must request only the resources they actually need, optimizing the overall utilization of the cluster.
- In environments where different teams have allocated budgets or resource limits, quotas enforce these limits, ensuring that teams do not exceed their allowed resource consumption.

With the Rafay Platform, resource quotas and limits are automatically implemented and enforced. This prevents one tenant's workloads from affecting the performance or availability of another tenant's workloads. Shown below is an example of resource quotas for GPUs.

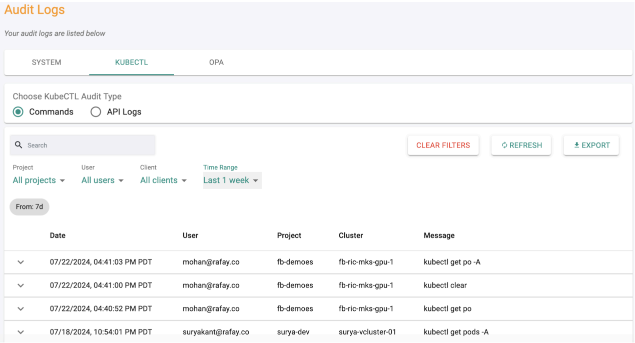


Centralized Audit Logging

Centralized audit logging is crucial for multi-tenant environments for several important reasons:

- Centralized audit logs provide a complete and consistent record of all activities, including user actions, API requests, and changes to resources. This is essential for monitoring suspicious activities, detecting security incidents, and responding to breaches.
- In the event of a security incident, centralized audit logs enable detailed forensic analysis to understand the scope and impact of the breach. This helps identify the root cause, the sequence of events, and any affected resources.
- Many regulatory frameworks (such as GDPR, HIPAA, PCI-DSS) require organizations to maintain an audit trail of all access and changes to sensitive systems. Centralized audit logging ensures that these requirements are met by capturing all relevant actions across the entire cluster.
- Centralized logs make it easier to generate compliance reports and verify that policies and procedures are being followed, as all logs are aggregated in one place and can be queried or analyzed systematically.
- Centralized audit logs provide a unified view of all activities within the tenant, making it easier for administrators, security teams, and auditors to understand what is happening across the environment. This visibility is crucial for maintaining control over complex and dynamic environments.
- In multi-tenant environments, centralized logging ensures that activities from all tenants are logged and can be reviewed. This promotes transparency and accountability, ensuring that tenants' actions are visible and can be audited if necessary.

In the Rafay platform, a centralized and immutable audit trail is automatically maintained for all activity performed by the users via all supported interfaces (UI and programmatic). Admins are provided with centralized access to the audit logs. The audit logs can also be configured to be streamed in real time to a configured SIEM.



Your audit logs are listed below

Choose KubeCTL Audit Type

● Commands ○ API Logs

Project: All projects

User: All users

Client: All clients

Time Range: Last 1 week

From: 7d

Date	User	Project	Cluster	Message
07/22/2024, 04:41:03 PM PDT	mohan@rafay.co	fb-demoes	fb-ric-mks-gpu-1	kubectl get po -A
07/22/2024, 04:41:00 PM PDT	mohan@rafay.co	fb-demoes	fb-ric-mks-gpu-1	kubectl clear
07/22/2024, 04:40:52 PM PDT	mohan@rafay.co	fb-demoes	fb-ric-mks-gpu-1	kubectl get po
07/18/2024, 10:54:01 PM PDT	suryakant@rafay.co	surya-dev	surya-vcluster-01	kubectl get pods -A

User Management and RBAC

Every user in an Org is associated with a unique email address. A user can be associated with one or more Orgs as well. Users can either be "Local Users" or "IdP Users". Click here to learn more about user management in the Rafay platform.

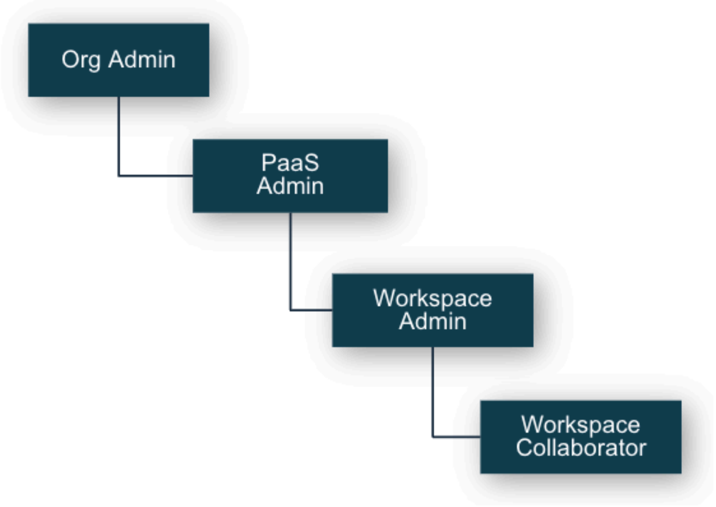
Local Users

- Lifecycle of these users is fully managed in the Rafay Platform.
- Typically limited to privileged super/root users such as Org Admins.
- Are locally authenticated by the Rafay platform.

IdP Users

- Lifecycle of these users is fully managed in the customer's Identity Provider (IdP).
- Typically for all non-privileged users such as developers, operations personnel etc
- Authentication and MFA is performed by the configured Identity Provider (e.g. Okta, Azure AD etc)

Every user is generally associated with one/more roles in the Rafay platform. For GPU PaaS, the following roles and associated permissions are relevant and the image below shows the hierarchy of roles as well.



Workspace Admin

Users with the workspace admin role can perform the following tasks:

- Create, view, update and delete compute instances based on published compute profiles in the catalog

- Create, view, update and delete services based on published service profiles in the catalog
- Create, view, update and delete applications

Workspace Collaborator

Workspace Collaborators are “invited” to a workspace as part of a team. They can be assigned “read and write” or “read only” privileges. They can perform the following tasks:

- View existing compute instances
- Create, view, update and delete services based on published service profiles in the catalog
- Create, view, update and delete applications

Kubernetes Management

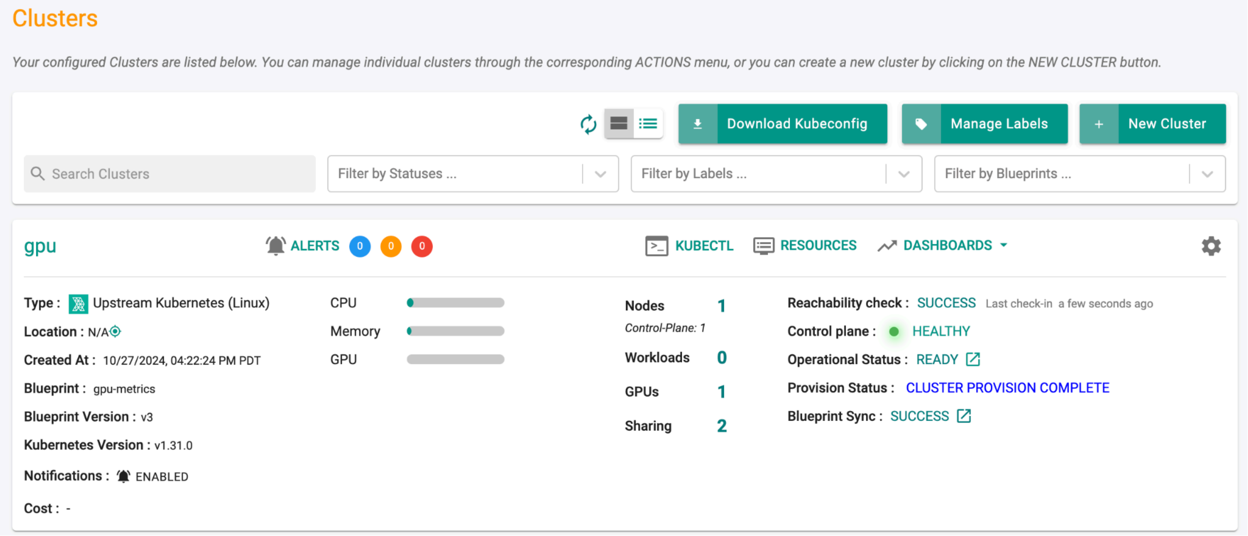
The Rafay platform can also be used to provision and manage the lifecycle (scaling, upgrades etc) of Kubernetes clusters in both datacenter and cloud environments i.e.

- Rafay’s MKS Distribution for Bare Metal
- Rafay’s MKS Distribution for virtual environments (i.e. VMware, OpenStack, Nutanix etc)
- Managed Kubernetes Cloud Providers (EKS, AKS, GKE, OKE etc.)

Rafay’s MKS Distribution is based on Upstream Kubernetes

NOTE: CSPs that may have existing Kubernetes clusters or have an alternate preference can import the clusters into the Rafay platform. However, the lifecycle management (add/remove nodes, Kubernetes upgrades, decommission etc) is the responsibility of the CSP.

Shown below is an example of a Rafay MKS cluster running on a bare metal system. As you can see this cluster is currently running Kubernetes v1.31.0, but was upgraded in-place from an older version.



Your configured Clusters are listed below. You can manage individual clusters through the corresponding ACTIONS menu, or you can create a new cluster by clicking on the NEW CLUSTER button.

gpu

ALERTS 0 0 0

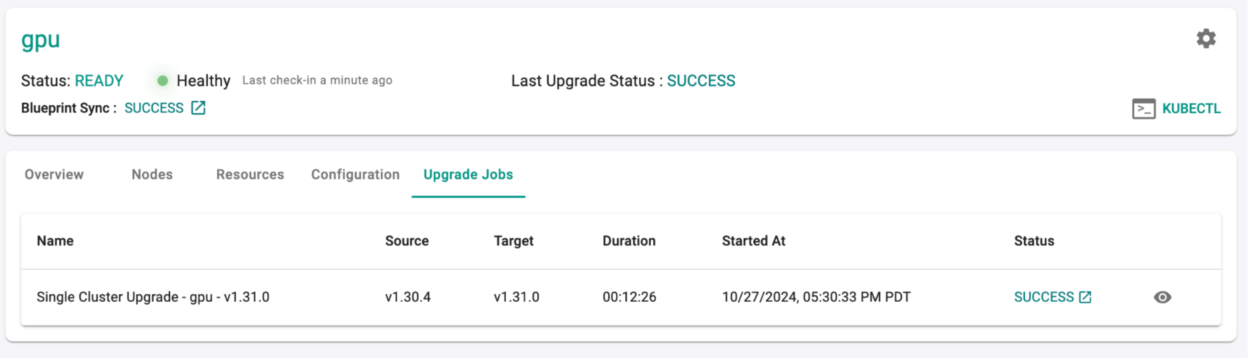
Type : Upstream Kubernetes (Linux)
Location : N/A
Created At : 10/27/2024, 04:22:24 PM PDT
Blueprint : gpu-metrics
Blueprint Version : v3
Kubernetes Version : v1.31.0
Notifications : ENABLED
Cost : -

CPU
Memory
GPU

Nodes	1
Control-Plane: 1	
Workloads	0
GPUs	1
Sharing	2

Reachability check : SUCCESS Last check-in a few seconds ago
Control plane : ● HEALTHY
Operational Status : READY
Provision Status : CLUSTER PROVISION COMPLETE
Blueprint Sync : SUCCESS

The image below shows the Kubernetes upgrade history. In this case, the CSP administrator can quickly check that this was upgraded from Kubernetes v1.30 to 1.31.



gpu

Status: READY ● Healthy Last check-in a minute ago

Last Upgrade Status : SUCCESS

Blueprint Sync : SUCCESS

Name	Source	Target	Duration	Started At	Status
Single Cluster Upgrade - gpu - v1.31.0	v1.30.4	v1.31.0	00:12:26	10/27/2024, 05:30:33 PM PDT	SUCCESS

Security

The Rafay platform is available in both SaaS and software form factors. The SaaS, multi-tenant controller is built on a zero trust security model that only requires outbound Internet connectivity on TCP port 443 from the CSP's datacenters to the Internet based SaaS controller.

Due to security and customer requirements (e.g. sovereign cloud), some CSPs may be required to deploy and operate the Rafay Platform in their infrastructure. The self hosted controller can be deployed in "fully air gapped" environments. All software and dependencies are "pre-packaged" into the installer. Once the software is installed, all activity can be performed without requiring any software downloads from the Internet.

NOTE
Detailed information about security is available in our online product documentation.

Network Security

The CSP is responsible for providing network security for their on-prem data centers where the infrastructure is provisioned. In a public cloud environment, the infrastructure will sit behind a customer's firewall such as AWS security groups.

User Authentication and Authorization

All users in an Org are required to verify access to their registered email address before they are allowed access. Org admins can optionally enable and enforce the use of MFA for all users in their organization.

Centralized Audit Logs

All actions performed by authorized users are audited and aggregated in a centralized, immutable audit logging system. A reverse chronological audit log is available for users via the Console.

Role based Access Control (RBAC)

The platform supports a number of roles that can be associated with users to ensure that user access to privileges can be tightly controlled.

Key Management

The Controller utilizes a central key vault for managing secrets across the entire infrastructure. Secrets are programmed into local cluster secret stores from the central vault when necessary.

Data Encryption

The disks are encrypted per-Org with a unique key for the organization.

Transport Encryption

All communication channels between the managed infrastructure and the Rafay Platform are done over TLS. A private PKI is utilized to generate certificates for inter component mTLS. All customer and partner facing API and Web Console utilize TLS as well.

Security White Paper

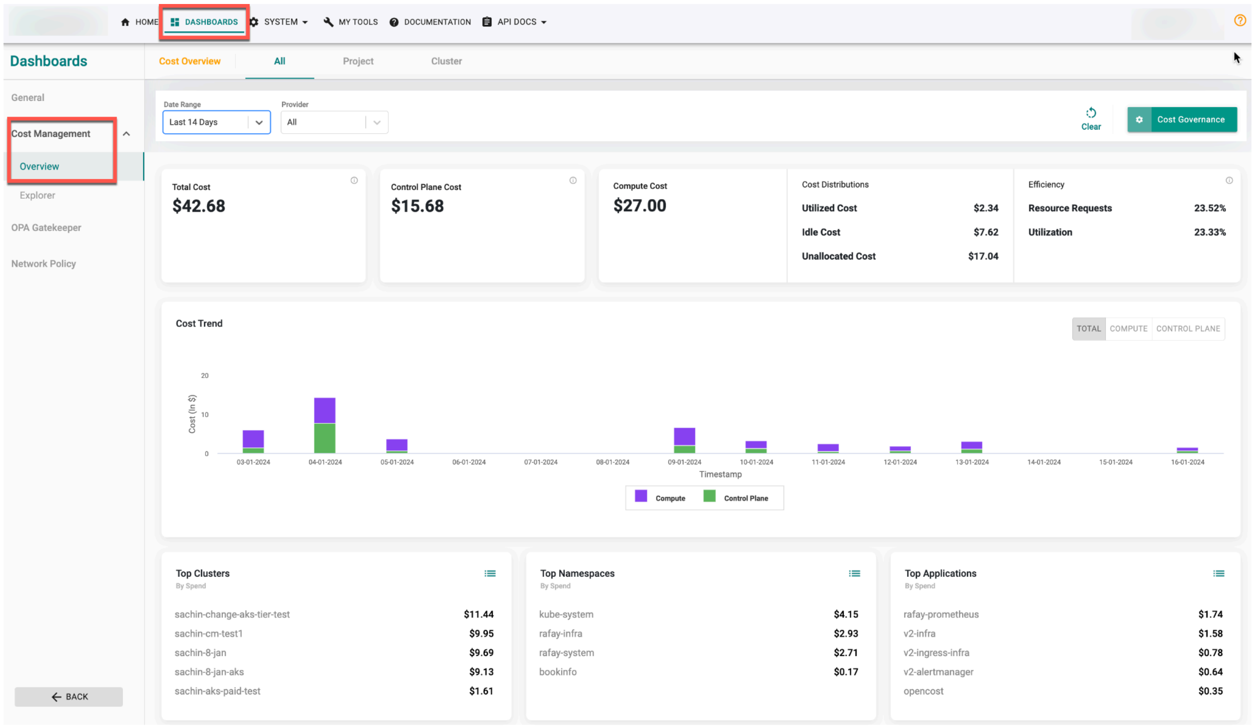
Our security team maintains a comprehensive Security Technical White Paper. Please contact us at security@rafay.co if you would like to receive a copy under NDA.

Cost Visibility and Chargebacks

The Rafay platform provides an integrated cost management and allocation engine that allows admins to effectively visualize, track and allocate costs across teams in a shared environment. This is performed by collecting and aggregating fine grained resource utilization metrics from the Kubernetes clusters.

CSPs can configure and attach custom cost profiles to the clusters. The data in the cost profiles is then associated with the resource utilization metrics to generate cost data for every resource.

Cost Overview dashboards provide visibility and insights into various metrics generated at cluster/project level. They provide a bird's eye view of costs and efficiencies for Projects, Clusters and Namespaces. Access is controlled by the assigned role. Longer-term retention of historical cost information makes it possible to anticipate future expense and can be leveraged for planning purposes.



Date Range: Last 14 Days

Provider: All

Total Cost: $42.68

Control Plane Cost: $15.68

Compute Cost: $27.00

Cost Distributions	
Utilized Cost	$2.34
Idle Cost	$7.62
Unallocated Cost	$17.04

Efficiency	
Resource Requests	23.52%
Utilization	23.33%

Top Clusters
By Spend

sachin-change-aks-tier-test	$11.44
sachin-cm-test1	$9.95
sachin-8-jan	$9.69
sachin-8-jan-aks	$9.13
sachin-aks-paid-test	$1.61

Top Namespaces
By Spend

kube-system	$4.15
rafay-infra	$2.93
rafay-system	$2.71
bookinfo	$0.17

Top Applications
By Spend

rafay-prometheus	$1.74
v2-infra	$1.58
v2-ingress-infra	$0.78
v2-alertmanager	$0.64
opencost	$0.35

Click here for detailed information on the cost visibility, chargebacks and governance capabilities of the Rafay platform.

End Users

End users in a customer's Org are typically data scientists, developers or researchers that need seamless access to GPU, compute and turnkey AI/ML tooling to do their jobs.

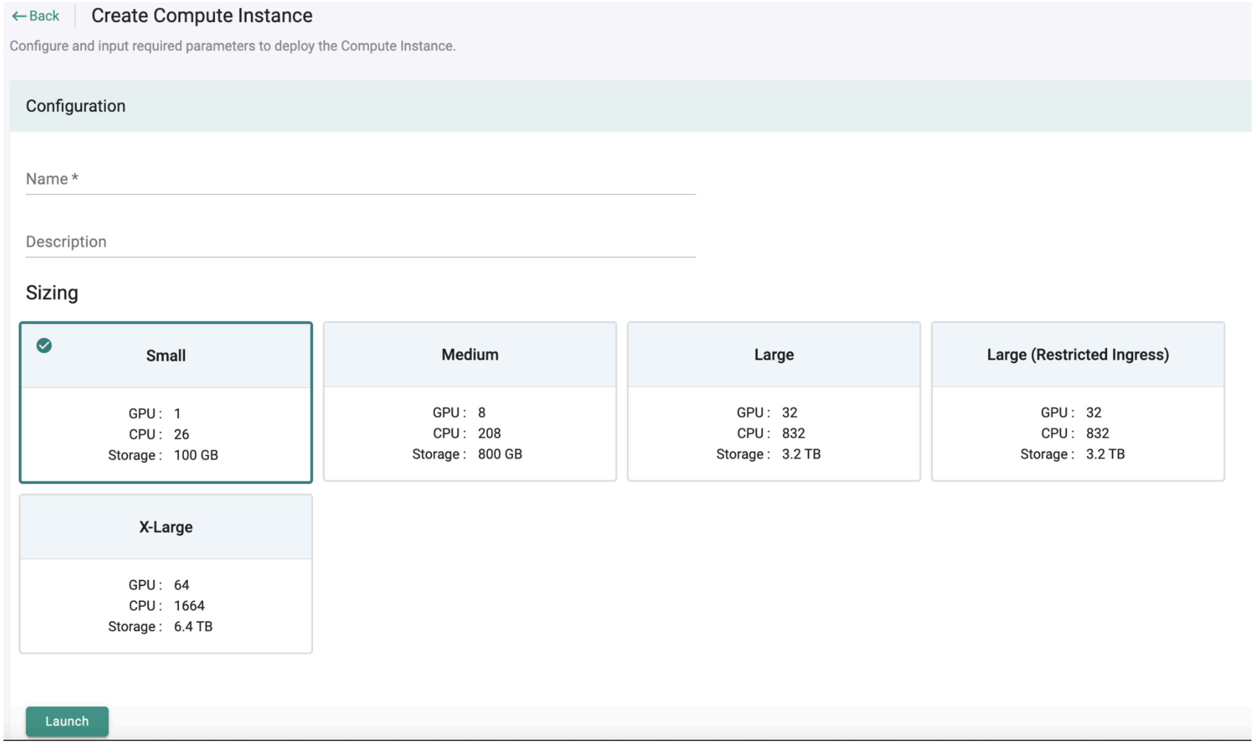
Workspaces

Every customer's Org can have 10s or 100s of workspaces underneath as isolated environments for users. A workspace is typically assigned to a user or a team with multiple users i.e. a team.

Once the tenant admin adds the end user to the platform, the user will have access to their assigned workspace(s). The end user can then access the self service portal to request and use available instances and services.

Compute Instances

A compute instance is an instantiation of a published "compute profile" that is made available by the CSP for their customer orgs. The end user can select and launch compute instances from the list of options available to them. The end user is provided with a vending machine style experience with SKUs that are curated and managed by the tenant administrator and the CSP. A simple, illustrative example is shown below where the end user can select from three options (small, medium, large etc) for their compute instance.



Create Compute Instance

Configure and input required parameters to deploy the Compute Instance.

Configuration

Name *

Description

Sizing

✓ Small
GPU : 1
CPU : 26
Storage : 100 GB

Medium
GPU : 8
CPU : 208
Storage : 800 GB

Large
GPU : 32
CPU : 832
Storage : 3.2 TB

Large (Restricted Ingress)
GPU : 32
CPU : 832
Storage : 3.2 TB

X-Large
GPU : 64
CPU : 1664
Storage : 6.4 TB

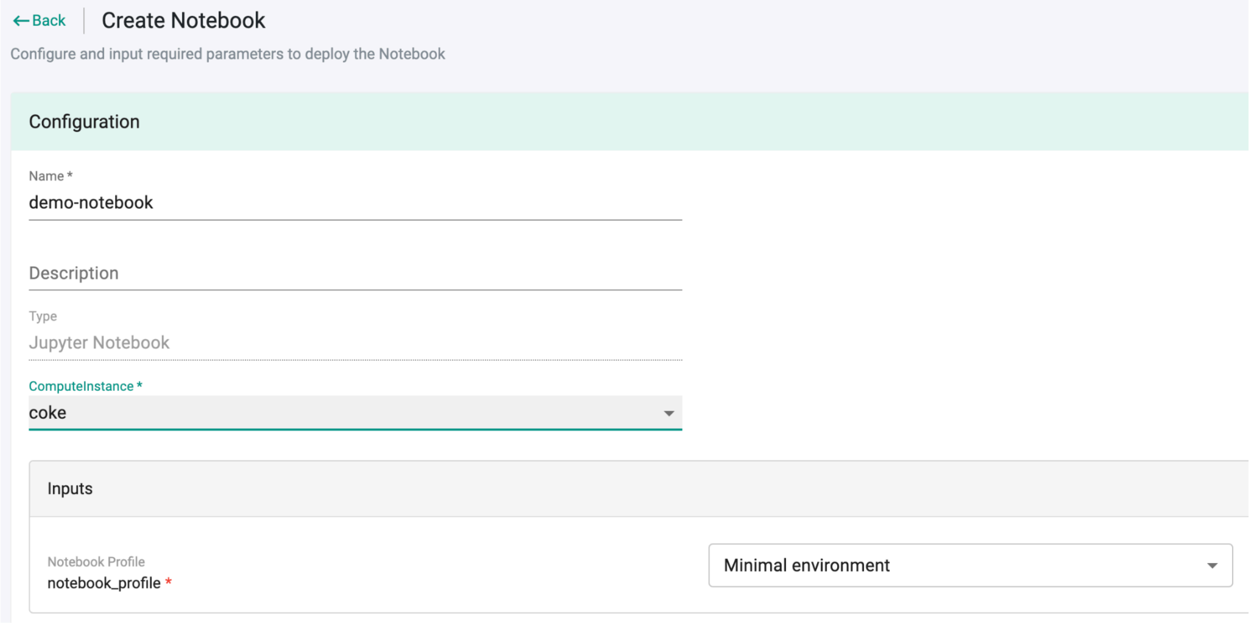
Notebooks

Notebooks are interactive, web-based environments where AI/ML practitioners can write and execute code, visualize data, and document insights in a single, cohesive platform. Notebooks provide a streamlined interface for experimenting with machine learning models, making it easy to test, iterate, and share results in real time. Notebooks have built-in support for Python and popular libraries which are essential tools for data science and AI, enabling both collaboration and reproducibility.

The Rafay platform provides first class support for notebooks via a default service profile. The Rafay platform also provides out of box support for a number of profiles for the notebook. The profiles ensure that the notebook has instant access to common tools and frameworks that can be used by the end user. Shown below is a list of profiles available by default to CSPs.

- Minimal
- Datascience
- Spark
- Tensorflow
- Tensorflow with CUDA
- PyTorch
- PyTorch with CUDA

An illustrative example is shown below where the end user is creating a notebook with a profile selector.



Create Notebook

Configure and input required parameters to deploy the Notebook

Configuration

Name *
demo-notebook

Description

Type
Jupyter Notebook

ComputeInstance *
coke

Inputs

Notebook Profile
notebook_profile * — Minimal environment

Jobs

A Job is a predefined set of processing actions that the user submits to the system to be carried out with little or no interaction from the user. A job will keep retrying their execution until a certain number of them have been completed successfully.

The Rafay platform currently provides end users with turnkey workflows for two types of jobs (a) Training Job and (b) Fine Tuning. These jobs provide users the means to quickly perform these tasks without having to learn low level infrastructure aspects.

NOTE
Additional types of Jobs will be provided by default based on user feedback.

Inferences

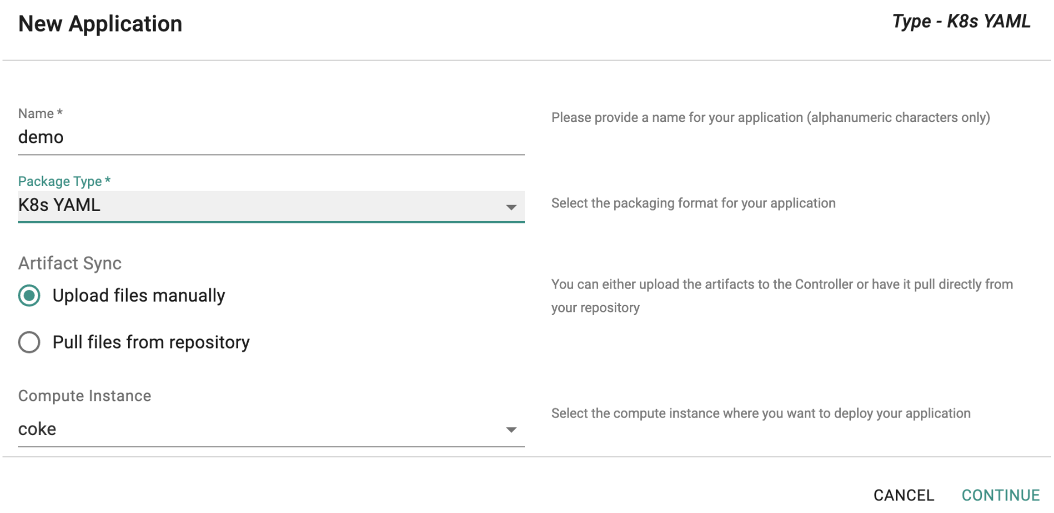
This service is meant for developers to quickly prototype with open-source models and deploy them in production. Users just need to select the serving framework and the model to deploy an inference/serving endpoint that they can use in a few minutes. Serving frameworks currently supported are:

- Ray Serve
- K-Serve
- Nvidia NIM

Applications

End users may wish to develop, deploy and test their containerized AI/ML applications to their instance. The Rafay platform provides an extremely convenient workflow for users to do this for their custom applications that are packaged as Kubernetes YAML or Helm Charts.

Shown below is an example where the end user has specified Kubernetes YAML as the package format for their application and will upload the YAML file as part of the workflow.



New Application

Type - K8s YAML

Name * demo

Please provide a name for your application (alphanumeric characters only)

Package Type * K8s YAML

Select the packaging format for your application

Artifact Sync

● Upload files manually

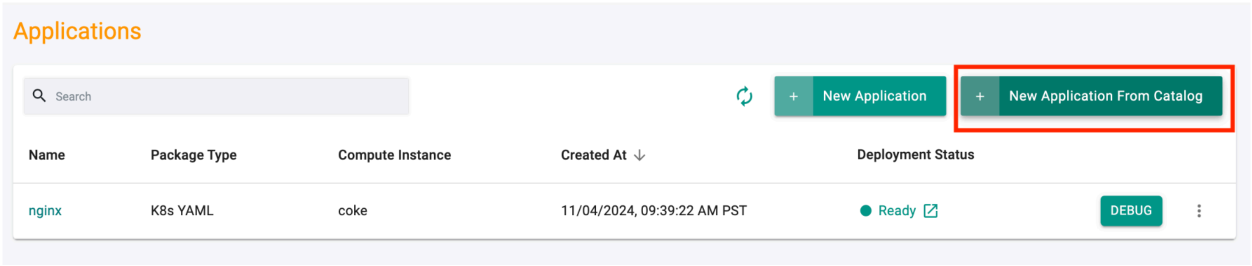
○ Pull files from repository

You can either upload the artifacts to the Controller or have it pull directly from your repository

Compute Instance coke

Select the compute instance where you want to deploy your application

The Rafay platform also supports 1-click deployment of 3rd party applications from an application catalog. This allows CSPs to operate a marketplace of applications that they can provide their users as a convenience. Shown below is an illustrative screenshot showing how the end user can create a new application from the built in application catalog.



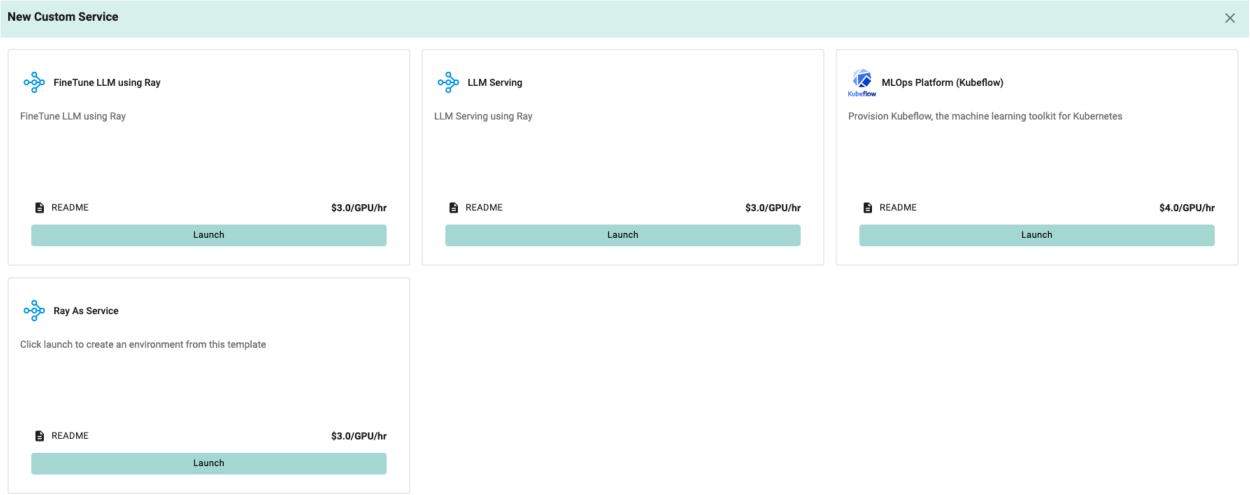
Applications

Name	Package Type	Compute Instance	Created At	Deployment Status
nginx	K8s YAML	coke	11/04/2024, 09:39:22 AM PST	● Ready

Services

The Rafay platform provides a number of AI/ML platforms as integrated, turnkey services. End users can deploy these to their instances in just a single click and start using them. Additional services will be added on an ongoing basis by the Rafay team. The illustrative example below shows the default list that is available for CSPs to offer to their customers.

- MLOps Platform based on Kubeflow + MLflow + Feast + KServe
- Ray as a Service
- LLM Serving as a Service
- Fine Tuning as a Service



New Custom Service

FineTune LLM using Ray

FineTune LLM using Ray

README $3.0/GPU/hr

Launch

LLM Serving

LLM Serving using Ray

README $3.0/GPU/hr

Launch

MLOps Platform (Kubeflow)

Provision Kubeflow, the machine learning toolkit for Kubernetes

README $4.0/GPU/hr

Launch

Ray As Service

Click launch to create an environment from this template

README $3.0/GPU/hr

Launch

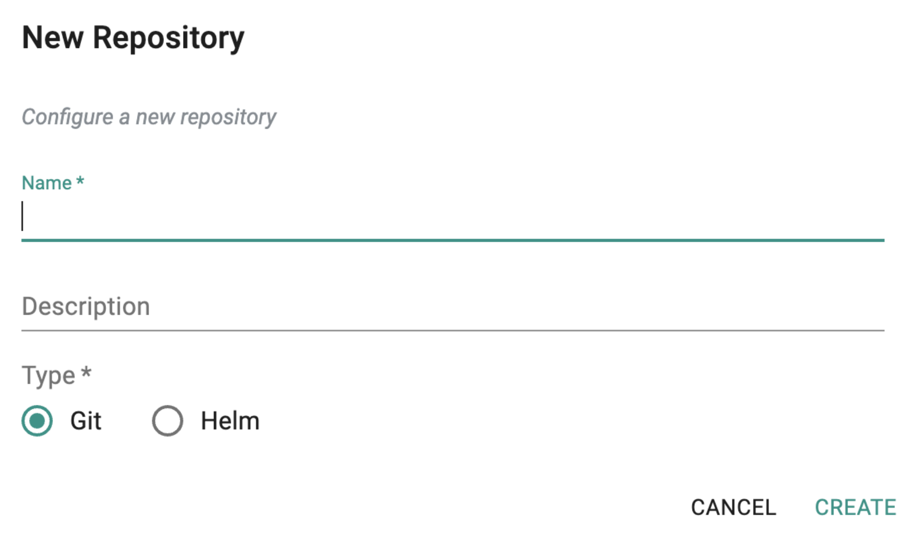
Integrations

The Rafay Platform provides a number of turnkey integrations for the benefit of end users. Two of these integrations are described below.

Repositories

Users generally store their application artifacts in a version controlled repository (i.e. Git or Helm). The Rafay platform provides a turnkey integration with both Git and Helm repositories so that users can manage their artifacts in these repos and do not have to deal with the burden of uploading the artifacts for every deployment. With this integration, the Rafay platform can pull the artifacts directly from the configured repository during the deployment process.

Users can configure both Git and Helm type repositories. Note that the Rafay platform also supports repositories that may be hosted in private networks behind firewalls. Shown below is an example of a repository with the type selected as “Git”.



New Repository

Configure a new repository

Name *

Description

Type *

● Git ○ Helm

CANCEL CREATE

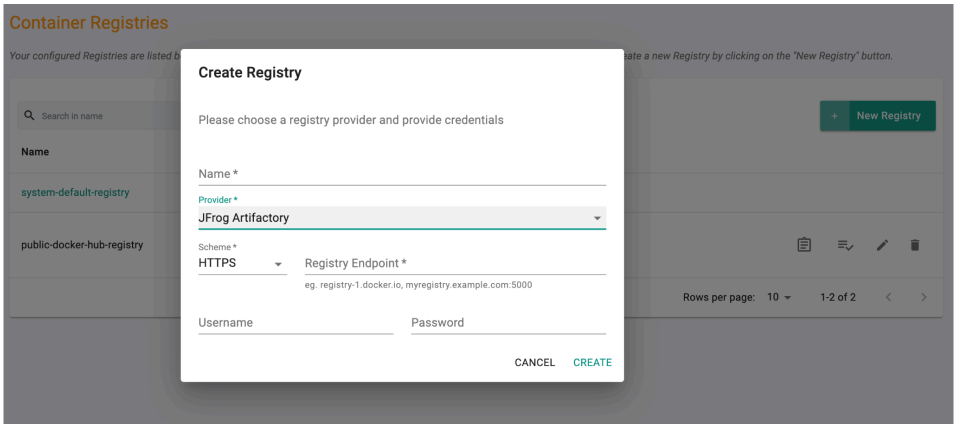
Container Registry

A container registry is a centralized repository where container images are stored, managed, and distributed. It enables developers to pull images to deploy applications across various environments, ensuring version control and easy collaboration in containerized workflows.

The Rafay platform provides a turnkey integration with both public and private container registry providers such as DockerHub (public, private), Amazon ECR, Google GCR, JFrog, Nexus etc. Administrators can securely create and manage imagePullSecrets for their private container registries that can be leveraged as references in end user workloads. The advantages of this integration are:

- No manual handling of imagePullSecrets by developers
- No need to embed imagePullSecrets in Kubernetes yaml files
- Secure (encrypted) delivery of imagePullSecrets to instances
- Automated provisioning and deprovisioning of imagePullSecrets on instances where workloads are deployed
- No dangling or orphaned imagePullSecrets on instances

Shown below is an illustrative example of the registry integration with JFrog Artifactory.



Container Registries

Your configured Registries are listed b … *ate a new Registry by clicking on the "New Registry" button.*

Create Registry

Please choose a registry provider and provide credentials

Name *

Provider *: JFrog Artifactory

Scheme *: HTTPS

Registry Endpoint *

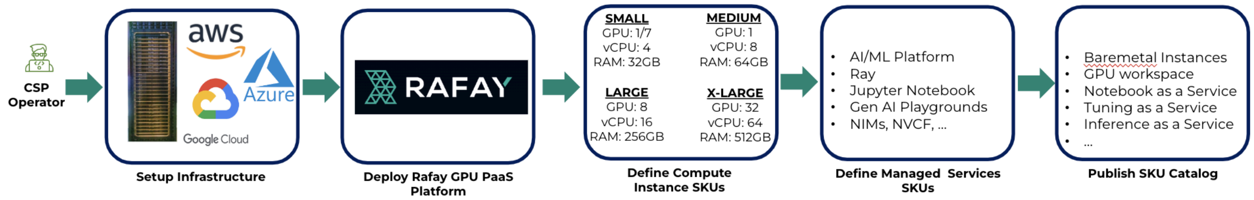
eg. registry-1.docker.io, myregistry.example.com:5000

Username

Password

Rafay GPU PaaS Platform Deployment

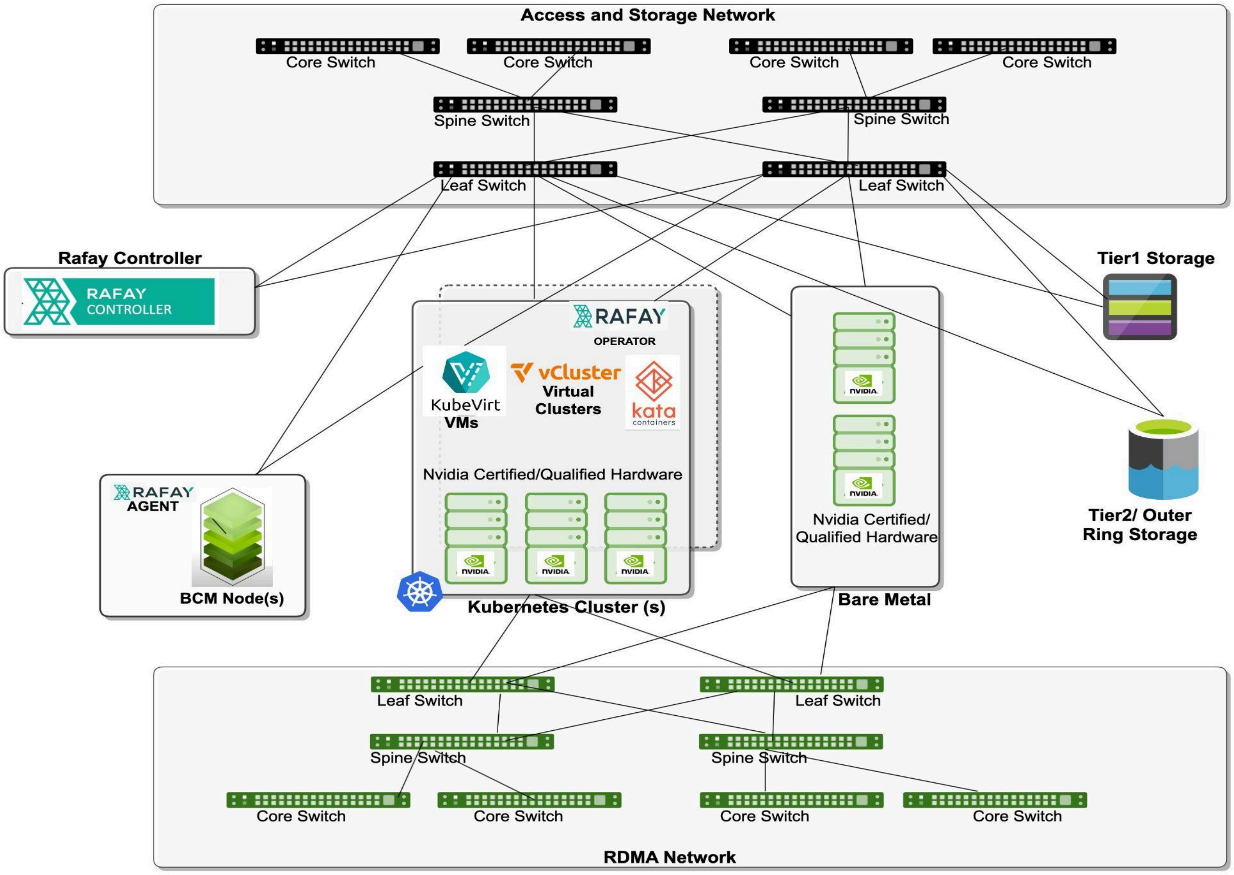
For a CSP, the process of bringing up GPU Cloud Service and deploying it in production can be broadly categorized into five steps as shown below.



Step 1: Setup Infrastructure

As a first step, the CSP deploys the infrastructure in their data center following the reference Nvidia-certified system hardware reference.

NOTE
If the CSPs hardware will require time to setup and is not ready or they need to burst to the public cloud for additional capacity, CSPs can optionally start the deployment with public cloud infrastructure such as GCP.


Access and Storage Network
Core Switch
Core Switch
Core Switch
Core Switch
Spine Switch
Spine Switch
Leaf Switch
Leaf Switch
Rafay Controller
RAFAY CONTROLLER
Tier1 Storage
RAFAY OPERATOR
KubeVirt VMs
vCluster Virtual Clusters
kata containers
Nvidia Certified/Qualified Hardware
RAFAY AGENT
BCM Node(s)
Kubernetes Cluster (s)
Nvidia Certified/ Qualified Hardware
Bare Metal
Tier2/ Outer Ring Storage
Leaf Switch
Leaf Switch
Spine Switch
Spine Switch
Core Switch
Core Switch
Core Switch
Core Switch
RDMA Network

- **Access & Storage Network**: A high-speed Ethernet network connecting users, compute, and storage, supporting low-latency access. This network is used for management tools (including the Rafay Controller, the Rafay Agent(s), and BCM), end-user access, and data transfers to and from storage during operations such as loading datasets or transferring model outputs. Implement this network using Nvidia-certified/qualified Ethernet networking hardware.

- **RDMA Network**: A high-throughput interconnect (InfiniBand/RoCE) for fast datapath data transfers. The RDMA network enables direct memory access between nodes without involving the operating system, reducing CPU overhead and significantly increasing data transfer speeds. RDMA is critical for workloads like distributed training, ensuring minimal latency and maximum throughput. Implement this fabric using Nvidia-certified/qualified networking hardware.

- **NVIDIA Certified/Qualified Compute Hardware** offered in multiple deployment options:
    - *Kubernetes*: GPU workloads can be deployed using Kata Containers within vClusters or as VM-based workloads via KubeVirt. GPU resources can be allocated flexibly, either as full GPUs or partitioned (using MiG/vGPU).
    - *Bare Metal Nodes*: Provide direct GPU access for performance-sensitive workloads that require maximum efficiency.

- **Storage**:
    - *Tier 1*: High-performance storage for data requiring rapid access. This tier is essential for workloads like neural network training or real-time inferencing, where fast I/O and low-latency data access are critical. Actively used data, such as training datasets and model weights, reside here. Use a storage solution from the Nvidia partner ecosystem.
    - *Tier 2/Outer Ring*: This storage serves as a repository for colder, less frequently accessed data, commonly used for archival, ETL workflows, or storing raw, unprocessed datasets. While not as fast as Tier 1, it is highly scalable and cost-efficient. Use a storage solution from the Nvidia partner ecosystem.

- **Rafay Controller and Agent(s)**: Responsible for the lifecycle management of multi-tenant Kubernetes clusters, automating the provisioning and scaling of GPU-enabled clusters. They orchestrate the deployment of containerized and VM workloads on Kubernetes clusters, facilitating policies for scheduling, monitoring, and enforcing resource quotas. This ensures efficient GPU allocation and minimizes resource contention. Rafay's GPU slicing capabilities, leveraging MiG or vGPU, ensure optimal GPU utilization while maintaining tenant isolation.

- **BCM**: Nvidia's Base Command Manager (BCM) manages the underlying bare metal infrastructure. BCM provides tools for provisioning and managing physical bare metal GPU nodes, handling tasks like node discovery, configuration, and lifecycle management.

Step 2: Deploy Rafay Platform

In this step, the CSP deploys the Rafay platform. The Rafay GPU PaaS platform can be completely white-labeled. It can be consumed by the CSP either as a SaaS solution or deployed in the CSP's infrastructure.

SaaS Option
For CSPs interested in using Rafay's SaaS deployment option, Rafay's Customer Success team will help configure and enable your administrators as part of an onboarding session.

This step involves setting up the CSP as a partner in the Rafay GPU PaaS platform with their own DNS hostnames, certificates, and white-label configuration, such as logos, etc. This step will also include a basic onboarding session to help the CSP administrators set up the required infrastructure and train them on basic workflows they need to understand to onboard and support their end customers.

NOTE
The SaaS option is the fastest and most cost effective way to get started. A typical CSP can be operational in just a few hours with this deployment option and requires zero management and infrastructure overhead with this approach.

Self Hosted Option
CSPs operating sovereign clouds may not be able to use the SaaS option and may need to deploy and configure the Rafay Platform software in their infrastructure.

NOTE
Please refer to the step-by-step instructions available in the supporting document "Air Gapped Installation" for details.

Step 3: Compute Profiles

In this step, the CSP defines the "compute profiles" aka SKUs that they want to offer to their end customers. The typical SKUs comprise both "types" and "flavors".

- Flavors can include SKUs such as "VMs, bare metal servers, Kubernetes clusters, Virtual Clusters, Fractional GPUs, etc.
- Multiple instances can be made available for a specific flavor. For example, GPU type, storage type etc. The instances can also be modeled in t-shirt sizes based on the number of resources such as GPUs, CPUs, Memory etc.

We recommend that CSPs start with a few instance types and add more flavors based on customer usage and interest.

In this step, the CSP also needs to configure their credentials such as BCM API keys, Public Cloud credentials, SSH keys, etc. in the Rafay PaaS Platform. This will allow the Rafay platform to fully **automate** the provisioning, scaling and configuration of underlying infrastructure as end users request for instances.

NOTE
To help a CSP get started immediately, Rafay provides a number of “reference” flavors and instance types out of the box. The CSP can leverage these to get started quickly and customize them to suit their needs. At steady state, the CSP will create and curate their own instance profiles with detailed billing metadata etc.

Step 4: Service Profiles

Rafay’s GPU PaaS platform provides many out of the box managed services, tooling and integrations that are needed by the data science teams to do their jobs effectively.

In this step, the CSP selects the services and tools that they want to provide to their customers.

By creating custom service profiles, CSPs can also operate a “marketplace” for 3rd party services that can be deployed onto the instances by their customers in just a click.

NOTE
To help a CSP get started immediately, Rafay provides a number of service profiles out of the box. The CSP can leverage these to get started quickly and customize them to suit their needs.

Step 5: Publish Catalog

In this step, the CSP creates a catalog of compute instance SKUs and managed services SKUs and makes it available to their customers. This step may include tasks like setting up default configurations, instance limits, GPU quotas etc.

With the completion of this step, the PaaS platform is fully ready for the cloud provider customer’s use.

GPU Infrastructure Deployment Automation

Before the CSP can onboard any customers, they need to make sure that they have all the required infrastructure and shared software configured and deployed. Here are the steps that they need to follow for this.

Step 1: Create CSP Org

Creates a Rafay Org for the CSP. This is an Org meant for the CSP to model and test the compute instance types and managed service offerings they wish to provide their customers.

Step 2: Deploy Rafay Agents

Create agents to execute Rafay provided templates and workflows in the CSP's Org. The Rafay platform uses the agent deployed in the CSP's network so that it has secure access to the CSP's infrastructure. These agents can be installed in a Kubernetes cluster or as a docker agent on a server.

Step 3: Configure Templates

Rafay provides a wide range of pre-built templates for "compute instances" and "service" as a catalog. These templates include infrastructure templates with different flavors (vCluster, VM, bare metal, K8s cluster) and sizes(small, medium, large, etc.). These templates also include managed services like Kubeflow, notebooks, model training, Ray as a service, RAG as a service, etc.

The CSP selects the infrastructure and service templates they intend to offer their customers and customizes them based on their requirements. CSPs can also develop and publish custom templates. The templates use the agent deployed in the prior step to access the CSP's infrastructure and tooling.

Step 4: Publish Compute & Service Profiles

In this step, the CSP creates compute profiles and service profiles for the templates created in the prior step. Compute profiles and service profiles are the CSP's SKUs that they can offer to their customers.

The CSP will now publish the compute and service profiles to their catalog in the operations portal to make them available to all/select customers.

Onboarding Customers

Here are the steps that a CSP needs to follow to onboard new customers with their own Orgs.

NOTE
Although the steps below can be performed via the Ops Console, we expect this to be primarily performed programmatically using APIs from the CSP's customer mgmt platform.

Step 1: Create Rafay Org

A Rafay Org needs to be created for every new customer. This can be done programmatically via APIs or via the Operations Portal by users with administrative privileges.

You will need to provide the following information:

- Name of the Org
- First Name, Last Name and Email Address for at least one admin for Customer's Org

Once an Org is created for the customer, the org admins can login to the GPU PaaS Platform to onboard their teams and end users.

Each team or environment can be considered a project in the Rafay platform. Separate RBAC, Quotas, Resources, etc. can be configured for each project. Developers, data scientists, researchers, Gen AI developers, ML Ops Engineers, and data engineers of the tenant organization will be given access to one or more projects based on their permissions to use the Rafay end-user portal to get resources and tools.

Step 2: Create Support User

This is an optional step, but recommended for scenarios where the CSP may be requested by the end customer to provide remote support. By default, only users authorized to an Org have access to it based on assigned privileges.