Ray Integration with Kubeflow VMware Distribution

Provides enterprise-grade, autoscaling, observability and intrinsic security.
VMware Private AI helps organizations accelerate returns from Generative AI (GenAI) initiatives while maintaining privacy and governance over sensitive data. It supports data wherever it resides, including on-premises data centers, public clouds, and the edge.

The VMware Private AI reference architecture establishes a scalable foundation for building secure and compliant AI services. It empowers organizations to keep control over confidential corporate information while enabling the use of open source and commercial solutions. Its integrated infrastructure and platform components streamline time-to-value.

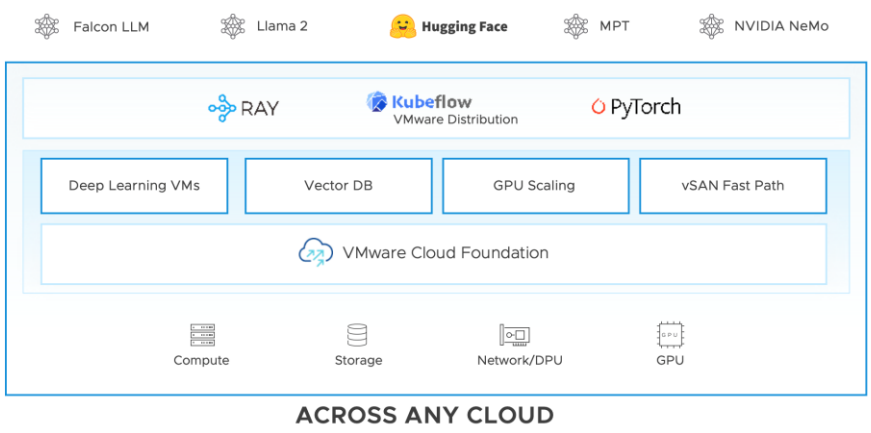

Figure 1. VMware Private AI

At the core, the reference architecture leverages Kubernetes and Kubeflow to provide a flexible MLOps environment. Kubeflow's capabilities manage the full machine learning (ML) lifecycle. Enterprises can seamlessly augment the platform with partner tools. For distributed training at scale, the Ray project's clustering and scheduling capabilities are integrated. This enables rapid scaling of workloads across on-premises infrastructure.

The Kubeflow VMware Distribution delivers an enterprise-grade Kubeflow platform optimized for VMware environments. It unites Kubeflow and VMware Tanzu® Kubernetes Grid™ to enable robust ML workflow deployment and administration on VMware systems. Generative models present new challenges for distributed training, prediction, and inference due to their scope and requirements. The Kubeflow VMware distribution addresses these through Ray integration, leveraging VMware's networking, security, scalability, and management to train and serve large generative models at scale.

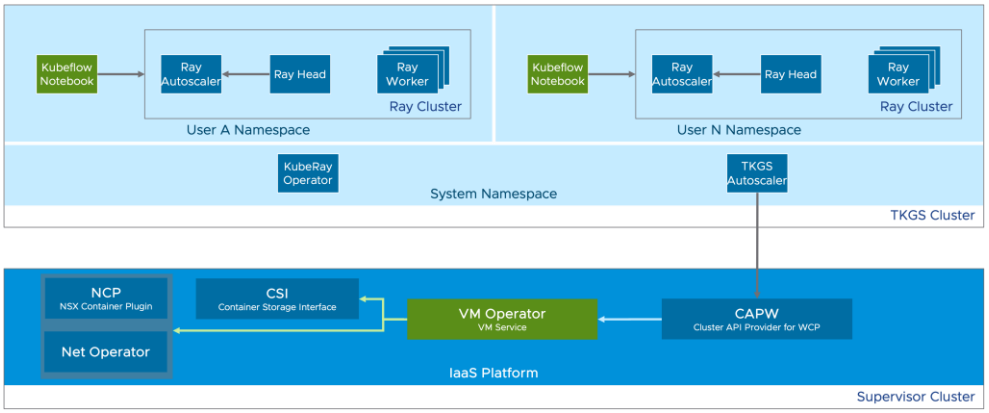

Figure 2. Ray Integration with Kubeflow

Business Values

The integration of Ray with Kubeflow VMware Distribution provides several key benefits for ML workflows:

- Simplified distributed training: It streamlines the notebook experimentation process through single-node building and tuning. Ray then easily scales these models across VMware Tanzu Kubernetes Grid clusters utilizing distributed training APIs. This maximizes available compute resource utilization. Ray also provides fault tolerance for distributed jobs.
- Two-layered autoscaling: Ray clusters running on Kubeflow through the VMware platform can automatically scale nodes up and down based on workload. This provides flexibility and cost optimization.
- Extensible pipelines: Ray integrates smoothly with Kubeflow pipelines via plugins. This allows data scientists to compose end-to-end workflows for training, validation, and model deployment.
- Unified environment: Ray and Kubeflow deliver a single interface and API set for managing ML lifecycles within a governed Kubernetes cluster on VMware. This streamlines workflow development and deployment for data scientists and ML engineers.

Figure 3 illustrates how Kubeflow on vSphere provides an interactive development environment.

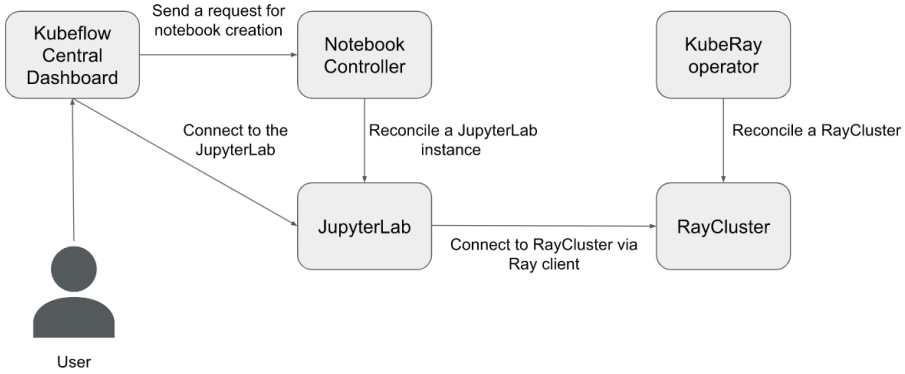

Figure 3. Interactive Development Environment

Simplified Deployment

Ray enables simplified deployment of distributed applications and machine learning workloads on Kubernetes. KubeRay, an open-source project, provides a comprehensive toolkit for deploying Ray on Kubernetes clusters. KubeRay uses Kubernetes custom resources and an operator to abstract complexities of deploying Ray in containers.

Enterprises can take advantage of KubeRay's capabilities by deploying it on the VMware Kubeflow distribution. Following the Install Kubeflow on vSphere Guide provides a turnkey process for deploying Kubeflow on VMware vSphere with Tanzu. To streamline Ray integration, we provide a run.sh script that handles deploying KubeRay alongside Kubeflow. This automated solution avoids manual configuration and supports users to launch and manage Ray applications through Kubeflow.

```
(base)  juanl@juanlEMD6R  ray  kubectl get all -n ray
NAME                                            READY   STATUS    RESTARTS   AGE
pod/kuberay-operator-7fbdbf8c89-n7256           1/1     Running   0          24h
pod/raycluster-kuberay-head-2br5f               1/1     Running   0          24h
pod/raycluster-kuberay-worker-workergroup-j26rw   1/1     Running   0          24h

NAME                                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
  AGE
service/kuberay-operator            ClusterIP   198.54.56.172    <none>        8080/TCP
  24h
service/raycluster-kuberay-head-svc   ClusterIP   198.51.107.123   <none>        8265/TCP,8080/TCP,8000/TCP,10001/TCP,6379/TCP
  24h

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/kuberay-operator    1/1     1            1           24h

NAME                                           DESIRED   CURRENT   READY   AGE
replicaset.apps/kuberay-operator-7fbdbf8c89    1         1         1       24h
```

Two-Layered Autoscaling

The Ray Autoscaler and Kubernetes Cluster Autoscaler work in tandem to dynamically scale infrastructure for ML workloads. The Ray Autoscaler monitors workload and decides when to scale out by creating or removing Ray pods. The Kubernetes Cluster Autoscaler then complements this by provisioning or deleting Kubernetes nodes in response. We provided an example to create an autoscaling Ray Cluster custom resource.

Specifically, when the Ray Autoscaler determines new pods are needed, the Cluster Autoscaler will automatically provision additional worker nodes if the existing nodes lack adequate computing capacity, thereby dynamically resizing the cluster to satisfy the scaling requirements identified by Ray. Likewise, after the idle Ray pods are removed, the underlying nodes can be cleaned up if left vacant. This collaborative autoscaling approach provides flexibility while efficiently utilizing cloud resources. It allows workloads to automatically leverage additional compute on demand, then release nodes no longer required in a coordinated manner.

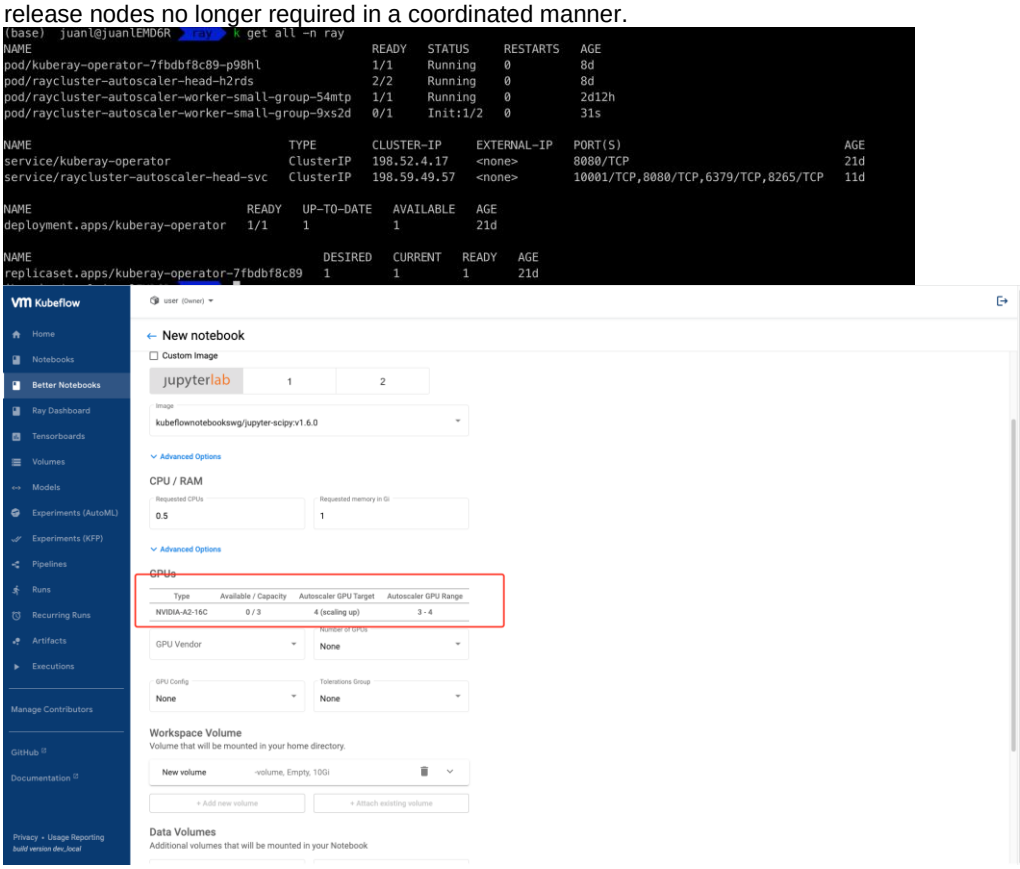

Figure 4. Autoscaling

Unified User Interface

Ray provides a web-based dashboard for monitoring and debugging Ray applications. The Ray dashboard is integrated into Kubeflow VMware distribution's centralized dashboard. The visual representation of the system state allows users to track the performance of applications and troubleshoot issues.

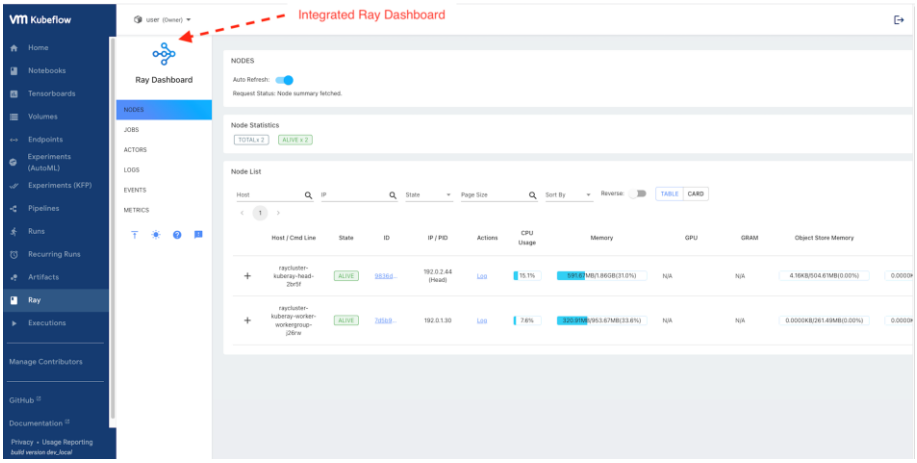

Figure 5. United User Interface

Takeaway

The Kubeflow VMware distribution provides a unified platform for optimizing ML workflows from early research to ongoing production deployment. Its integration of Ray and Kubeflow streamlines the full AI development cycle.

Ray enables accelerated experimental prototyping through efficient single-node model building, tuning, and testing capabilities. This allows developers to quickly iterate on ideas and experiment with different models and parameters. Ray then seamlessly scales workloads over VMware's Tanzu Kubernetes Grid infrastructure for further optimization.

Kubeflow further enhances productivity. It offers granular visibility and governance over resource utilization during large-scale model training and inference. Additionally, Kubeflow's optimized Kubernetes operations and secured multi-tenant namespaces improve deployment control and user isolation. The combined solution significantly accelerates the complete AI initiatives built to run seamlessly from prototype to product on VMware infrastructure.

Reference

For more information, visit:

- https://github.com/vmware/vSphere-machine-learning-extension
- https://docs.vmware.com/en/VMware-vSphere/index.html
- https://www.ray.io
- For VM-based Ray integration，refer to https://octo.vmware.com/enabling-ai-announcing-the-ray-on-open-source-plugin/
- https://core.vmware.com/resource/kubeflow-vmware-distribution

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com