Run:ai Notebooks and Workflows

Christopher Nagy; Mentors: Johnathan Nielsen; Daniel Wild; Cory Blount | HPC-SYS Support Services

Introduction

MOTIVATION

We want to provide LANL scientists an easy-to-use interface for developing and running AI-focused workloads to increase the efficiency of scientific discovery.

GOALS

The interface, from workload creation to development, should be intuitive. We want to provide environments suited to various AI tasks. There should be helpful documentation with tutorials to make onboarding easy. Further, integration to existing workflows and data should be supported to make the transition easier for scientists. Repeatable tasks will be automated using GitLab CI/CD and DevOps practices to make administration easier.

Architecture

OPENSHIFT

OpenShift is an enterprise Kubernetes platform offered by Red Hat with additional tools for managing, securing, and automating containerized applications at scale. Openshift adds administrator and developer features such as integrated CI/CD pipelines and role-based access controls, all in an intuitive user interface.

RUN:AI

Run:ai extends the Kubernetes platform to optimize GPU resource utilization for AI/ML workloads. It introduces dynamic and fractional GPU usage and makes it more efficient to launch training and inference workloads for multiple researchers.

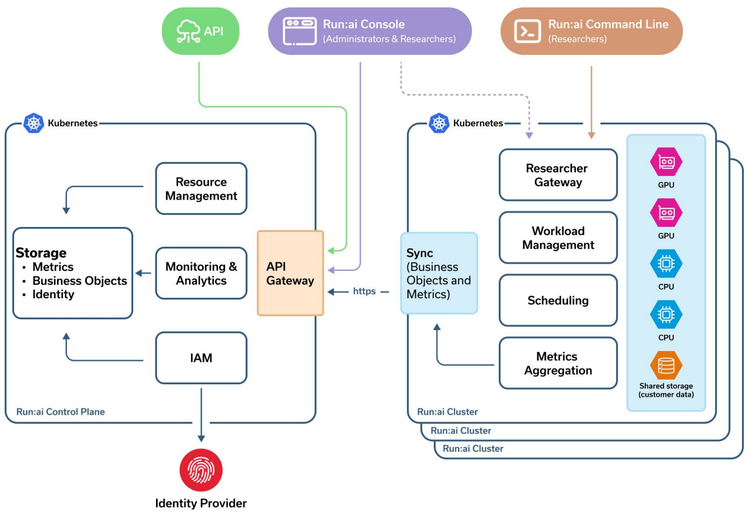

Figure 1. Run:ai architecture. Source: https://docs.run.ai/v2.20/home/overview/

JUPYTER NOTEBOOK

Jupyter Notebook provides an interactive Python IDE in a browser which can be hosted remotely. This removes the burden of installing software and compilers and gives users more compute power than what is available to them locally. Because Jupyter extends Python, development and data visualization is easier for those with less coding experience. A Notebook runs in a single container which ensures a consistent, isolated, and reproducible environment and simplifies management for administrators as we can utilize existing container platforms such as Openshift and Run:ai.

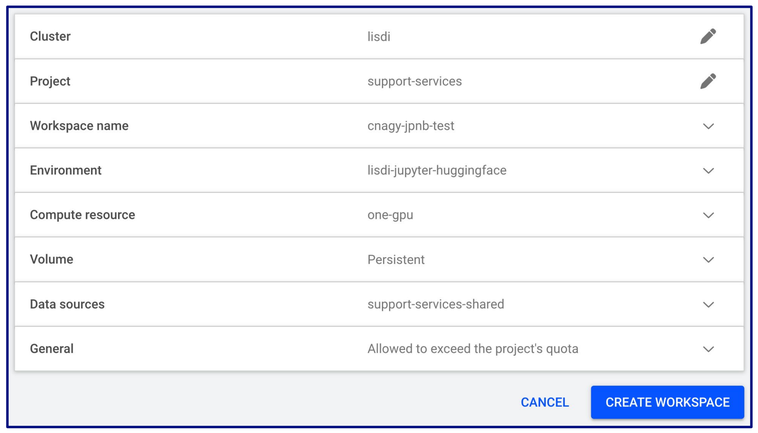

Figure 2. Run:ai web console UI showing a workload creation configuration. Users select a pre-built environment, request resources, and can attach storage or data.

Method and Results

CREATING IMAGES AND ENVIRONMENTS

Due to LANL's secure environment, it is necessary to create custom images that conform to the security constraints. The LANL-specific images were created using a Dockerfile, starting from a base Jupyter Notebook image which already contained the necessary code to run in an outside environment. However, we needed to add and configure LDAP libraries for compatibility with LANL's authentication and security.

Once the images were made, we stored them in an internal quay registry that Run:ai can access. Using Run:ai's web console we created pre-configured environments that pull these custom images. Users create workloads with the environments, specifying GPU resources to match workload demand.

The Run:ai web console connects to the Openshift cluster through an Openshift route, meaning that Run:ai users do not need accounts on the Openshift cluster. As administrators, we assign users their own namespaces and roles in Run:ai with varying permissions, helping to maintain isolation and uphold the principle of least privilege.

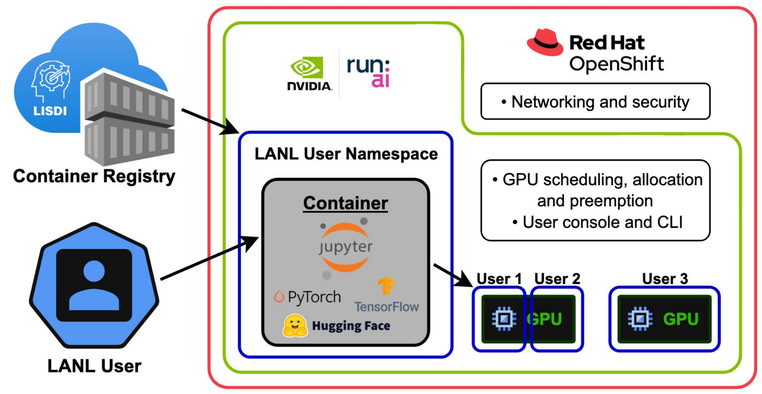

Figure 3. Visualization for how a user interacts with Run:ai. Run:ai pulls the image and schedules the container on a GPU. The user can directly access the container and has no knowledge of Openshift in the background. GPUs can be allocated fractionally.

EARLY USER RESULTS

Through early users and our testing, the Notebook images are functioning well. Creating workloads is seamless, they are able to pull and run models, data persists between sessions, and development is easy especially with the help of AI tools.

Issues and Challenges

LANL SPECIFICITIES AND EARLY USERS

The main challenge to overcome is adhering to LANL network security. Configuring the images to work with our LDAP authentication server was difficult initially, but works smoothly now. Other Run:ai environments, such as VS Code and the vLLM server, also have networking issues stemming from Openshift that are actively being diagnosed.

The other challenge at this stage is the lack of users testing to get feedback from. We cannot precisely predict what features users want, what is unintuitive, and what kinds of workflows they will be running. Thus, we must run tests and strain the system ourselves before we get more users.

Additional Exploration

JUPYTER HUB ON OPENSHIFT

Deploying Jupyter Notebooks through JupyterHub on Openshift was also explored. A proof-of-concept was implemented, but focus shifted to Run:ai as the preferred platform due to its superior management of AI workloads.

OTHER RUN:AI ENVIRONMENTS

Run:ai provides default environments for VS Code, a vLLM server, and a Chatbot UI. These were explored due to their potential to help with user integration and providing an interface to experiment with LLMs. However, these are not yet functional due to conflicts with LANL Openshift networking policies.

Future Work

AUTOMATION AND USER LIFECYCLE

We plan to add GitLab CI/CD automation in line with existing HPC pipelines, such as image mirroring to automate and standardize image updates and creation. We will be writing documentation for the whole user lifecycle from onboarding to retirement. The platform's security will be evaluated to progress towards a wider user base.

Los Alamos NATIONAL LABORATORY
NNSA National Nuclear Security Administration
Managed by Triad National Security, LLC, for the U.S. Department of Energy's NNSA.
ROSY ID: 6846be39