Ezmeral Unified Analytics: Fix problems with running kfp-compiled Kubeflow Pipelines

Issue

Users of EZUA 1.5 can encounter problems with running Kubeflow Pipelines compiled by EZUA's ML Pipeline service because of a Kyverno policy requiring all containers in a pod to specify resource requests and limits. If you find that your Kubeflow Pipeline pod is stuck in Pending state with an event description similar to

```
Warning   PolicyViolation    31s   kyverno-scan                 policy audit-pod-requests-and-limits/check-pod-resourc
```

then please follow the instructions in this KB article to address the problem.

Additionally, if you have any other problems when running Kubeflow Piplelines that you have compiled in EZUA using the KFP client (typically from within a Jupyter Notebook), please download and review the sample at https://hpecp-support.s3.us-east-2.amazonaws.com/ezua_kfp_lightweight_component_example.ipynb , which can be used as a basis for creating your own pipelines. The notebook sets 'install_kfp_package' to 'false', meaning that it will not attempt to install the KFP package into the component image that it builds and is therefore suitable for use from behind a web proxy. Also, the /tmp directory of the component base image is writable, allowing your pipeline code to store its output between steps.

Environment

Ezmeral Unified Analytics 1.5
EZUA 1.5

Cause

The problem occurs because the Kubeflow Pipeline jobs created by the ML Pipeline in EZUA do not specify resource requests and limits and are therefore blocked from being scheduled by a Kyverno policy.

Resolution

1 - If your EZUA installation is airgapped, add these three container images to your airgap registry:

- gcr.io/mapr-252711/kubeflow/ml-pipeline/kfp-driver:ezua-1.5.1-b431bf965
- gcr.io/mapr-252711/kubeflow/ml-pipeline/kfp-launcher:sha256-50151a8615
- gcr.io/mapr-252711/kubeflow/ml-pipeline-api-server:ezua-1.5.1-b431bf965

The images are also freely available from the HPE Greenlake Marketplace registry (prefix the above image tags with **marketplace.us1.greenlake-hpe.com/ezua** to obtain them).

2 - Acting as a Kubernetes administrator on your EZUA workload Kubernetes cluster, edit the *ml-pipeline* deployment in the *kubeflow* namespace (*kubectl -n kubeflow edit deployment ml-pipeline*) and update the tag of the image used by the 'ml-pipeline-api-server' container from ' *ezua-1.5.0-0b57b5445*' to ' *ezua-1.5.1-b431bf965*'

3 - Confirm that the ml-pipeline deployment creates a new pod in the kubeflow namespace and that the pod successfully comes to running state

4 - Test that your Kubeflow Pipeline can be successfully compiled and run (or try the sample notebook linked in this KB article)