Kubeflow

CUJ Multi-user Kubeflow

Contributors: jlewi@google.com
2018-12-20
http://bit.ly/kubeflow_cuj_multi_user

Status

- CUJ Draft Started 2018/12/20
  - Open issues label cuj/multi-user
  - Design Doc (http://bit.ly/kf_jupyter_design_doc)

Objective

- Enable a DevOps engineer to deploy a single Kubeflow cluster to be shared by multiple individuals/teams
- Each individual/team gets their own namespace to use Kubeflow that is isolated from other users
- Admins can set resource quotas on each namespace

Admin deploys kubeflow

Deploy Kubeflow

- Users follow Kubeflow Getting Started Guide to deploy Kubeflow on their platform of choice
- kubeflow/kubeflow#1419 - kfctl CLI
- kubeflow/kubeflow#2094 - support basic auth
- **kubeflow/kubeflow#2177**: Install profiles by default

```
curl
http://go/gh/kubeflow/kubeflow/archive/kfct
l

kfctl init ${KFAPP} --platform none
cd ${KFAPP}

kfctl generate k8s
kfctl apply k8s
```

What namespaces are created?

- Two namespaces are created
  - kubeflow- contains system components of user
  - TODO - should we rename to kubeflow-system?

```
kubectl get namespace
NAME                 STATUS    AGE
kubeflow             Active    1d
```

Customize Default Resource Quotas

- Admin can customize the default ResourceQuotas assigned to each namespace
- profiles is the profiles component

```
cd ${KFAPP}/ks_app
ks param set profiles maxCpu 100
ks param set profiles maxRam 1Tb
ks param set profiles maxGpu 20
Ks apply ${ENV} -c profiles
```

Admin grants permission to create profiles

- profile-creator RBAC role grants create permission for Profiles resource
- A Profile resource creates the namespace and related resources

```
Kubectl -n kubeflow-system create rolebinding
bob-creator \
 --role=profile-creator \
 --user=bob@acme.com
```

Override default resource quotas

- Admin can update ResourceQuotas for an existing namespace to use non-default values

```
Kubectl -n kubeflow-${USER} apply -f quota.yaml
```

Create a namespace for user/team

Create a new namespace CLI

- Users with appropriate RBAC permissions can create a new namespace by creating a new profile resource with Kubectl
- The profile controller then creates the namespace and related resources (RBAC roles & bindings, ResourceQuotas)

```
kubectl -n kubeflow-system create profiles
abe.profile.yaml

apiVersion: kubeflow.org/v1alpha1
kind: Profile
metadata:
  name: abe
  namespace: kubeflow-system
spec:
  template:
    metadata:
      namespace: kubeflow-abe
    spec:
      owner:
        kind: User
        apiGroup: rbac.authorization.k8s.io
        name: abe@foo.com
```

Create a new namespace UI

- We will also provide a UI/web app to create new namespaces
- kubeflow/kubeflow/issues/2178

TODO add mock; simple form

Profile Resource Created

- Profiles package
- TODO: Why do we need apiGroup in User?
- TODO: Is name just the RBAC user name?

```
kubectl -n kubeflow-system get profiles -o yaml

apiVersion: kubeflow.org/v1alpha1
kind: Profile
metadata:
  name: jane
  namespace: kubeflow-system
spec:
  template:
    metadata:
      namespace: kubeflow-jane
    spec:
      owner:
        kind: User
        apiGroup: rbac.authorization.k8s.io
        name: jane@foo.com
```

User uses Kubeflow

User Accesses Jupyter

- User connects to Kubeflow spawner
- Open https://<ENDPOINT>/jupyter
- User may need to setup kubectl port-forward if not exposed

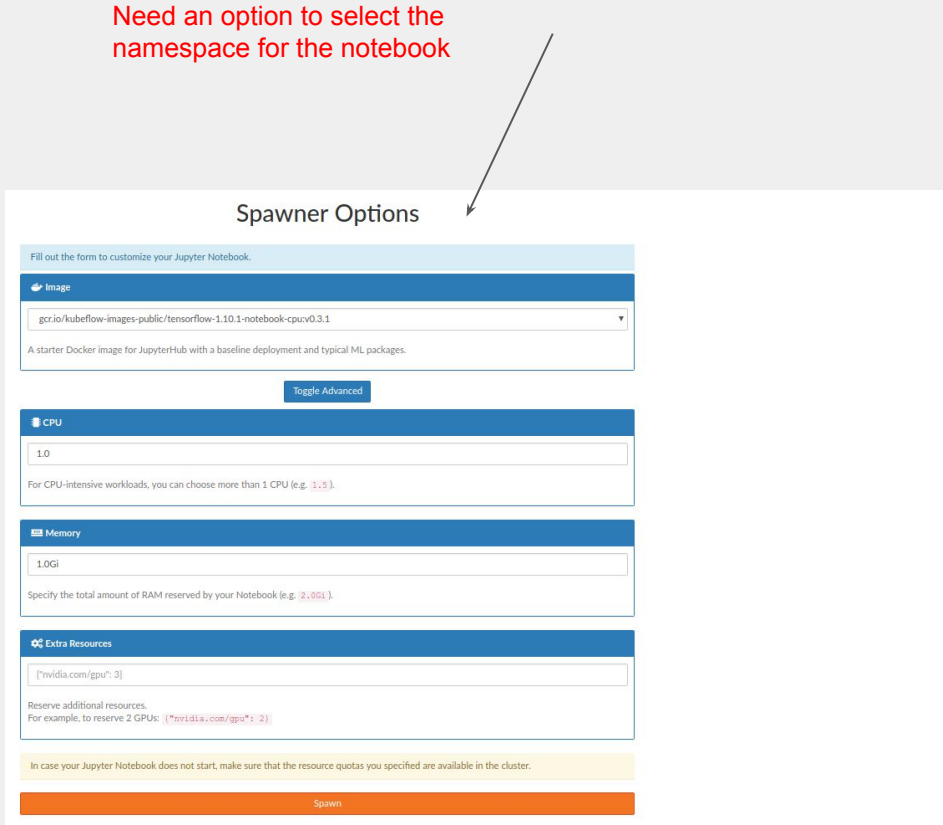
User Is Directed to login

- User logs in using OIDC provider (e.g. Google)
- Implementation will be platform specific
  - E.g. OSS version based on Ambassador & bit.ly/oauth_proxy
  - kubeflow/kubeflow#11
  - Or cloud/vendor specific solution e.g. Google IAP
- Username gets added as JWT header

WebPage for OIDC flow

Create a Jupyter Notebook

- Spawner should allow user to set namespace
  - Default should be based on user as specified in the JWT
- WebApp uses OIDC to authenticate to ApiServer and create resources
  - see http://bit.ly/kf_jupyter_design_doc



Under the hood

- Notebook is spawned in namespace kubeflow-jane
- ISTIO route created for the notebook
- ISTIO configured to reject traffic if JWT doesn't match user who owns the notebook
- See: http://bit.ly/kf_jupyter_design_doc

Create TFJob/PyTorch

- User can create TFJob/PyTorch in namespace
- Rely on RBAC to ensure only in appropriate namespace

```
Kubectl -n kubeflow-jane create -f tfjob.yaml
```

Deploy model

- User deploys model and uses ISTIO to restrict who can access the endpoint
- TODO: Add ISTIO commands/steps for restricting access to the endpoint

Adding Users to an existing namespace

Share namespace with other users

- If Jane wants to share her namespace with other users she just creates appropriate RoleBinding in the namespace
- Name of the binding is defined here
- TODO: Use a better name than default for the role?

```
Kubectl -n kubeflow-jane create rolebinding
bob-kubeflow \
 --role=default \
 --user=bob@acme.com
```

Deleting Profiles

Delete a profile

- Users with RBAC permission profile-deleter can delete profiles

```
Kubectl -n kubeflow-system delete profiles bob
```

References

https://github.com/kubeflow/kubeflow/tree/master/kubeflow/profiles