Если вы видите что-то необычное, просто сообщите мне.

Подключение crossplane к yandex

Установка провайдера

Создаем ключ для доступа в yandex и помещаем его в кубер

```
yc iam service-account create --name crossplane
yc resource-manager folder add-access-binding <folder_id> --service-account-name crossplane --
role editor
yc iam service-account get crossplane
yc iam key create --service-account-id service_account_id --output key.json
kubectl create secret generic yc-creds -n "crossplane-system" --from-
file=credentials=./key.json
```

Создаем конфигурацию

controllerconfig.yml(используется не всегда)

```
apiVersion: pkg.crossplane.io/v1alpha1
kind: ControllerConfig
metadata:
  name: controllerconfig-k8s
  namespace: crossplane-system
spec:
  tolerations:
  - effect: NoSchedule
    key: worker-for-systemapps
    operator: Equal
    value: "1"
```

provider.yml

```
apiVersion: pkg.crossplane.io/v1
kind: Provider
metadata:
  name: provider-ycloud
  namespace: crossplane-system
spec:
  package: cr.yandex/crp0kch415f0lke009ft/crossplane/provider-jet-yc:v0.1.37
  packagePullPolicy: Always
  # используется вместе с controllerconfig
  controllerConfigRef:
    name: "controllerconfig-k8s"
```

providerconfig.yml

```
apiVersion: yandex-cloud.jet.crossplane.io/v1alpha1
kind: ProviderConfig
metadata:
  name: providerconfig-ycloud
  namespace: crossplane-system
spec:
  credentials:
    cloudId: <cloud-resource-id>
    folderId: <cloud-resource-id>
```

```
    source: Secret
    secretRef:
      name: yc-creds
      namespace: crossplane-system
      key: credentials
```

Применяем crossplane ресурсы

Задержка необходима для providerconfig, так как CRD устанавливается не crossplane а ycloud провайдером. Сразу всё применить не получится.

```
# первую команду можно опустить если для запуска проекта не требуется специальных настроек.
kubectl apply -f controllerconfig.yml
kubectl apply -f provider.yml
sleep 10 && kubeclt apply -f providerconfig.yml
```

Разное

В качестве эксперимента можно создать тестовую s3 корзину:

> Внимание! Необходимо использовать директиву: `deletionPolicy: Orphan` в противном случае ресурс будет удален вместе с xrd.

```
apiVersion: iam.yandex-cloud.jet.crossplane.io/v1alpha1
kind: ServiceAccountStaticAccessKey
metadata:
  name: sa-static-key
  namespace: crossplane-system
spec:
```

```
    forProvider:
        description: "static access key for object storage"
        serviceAccountId: <cloud-resource-id>
    providerConfigRef:
        name: providerconfig-ycloud
    writeConnectionSecretToRef:
        name: sa-key-conn
        namespace: crossplane-system


---
apiVersion: storage.yandex-cloud.jet.crossplane.io/v1alpha1
kind: Bucket
metadata:
    name: bucket
    namespace: crossplane-system
spec:
deletionPolicy: Orphan
    forProvider:
        accessKeyRef:
          name: sa-static-key
        secretKeySecretRef:
          name: sa-key-conn
          namespace: crossplane-system
          key: attribute.secret_key
        bucket: "bucket"
        acl: "public-read"
    providerConfigRef:
        name: providerconfig-ycloud
```

```
kubectl apply -f example-s3-bucket.yml
```

После применения идем в веб-интерфейс yandex и проверяем созданный ресурс.

> Если ресурс не появился откройте описание созданного ресурса в k8s, в yml будет отдельное поле с ошибкой.

Импортирование существующего ресурса

Импортировании существующего ресурса производится следующим образом:

```
    metadata:
      annotations:
        crossplane.io/external-name: <cloud-resource-id>
```

Удаление

Для переустановки необходимо удалить пространство crossplane-system. А так же удалить все CRD которые принадлежат crossplane.

Revision #3
Created 2022-11-02 07:23:43 UTC by gasick
Updated 2022-11-02 07:55:16 UTC by gasick