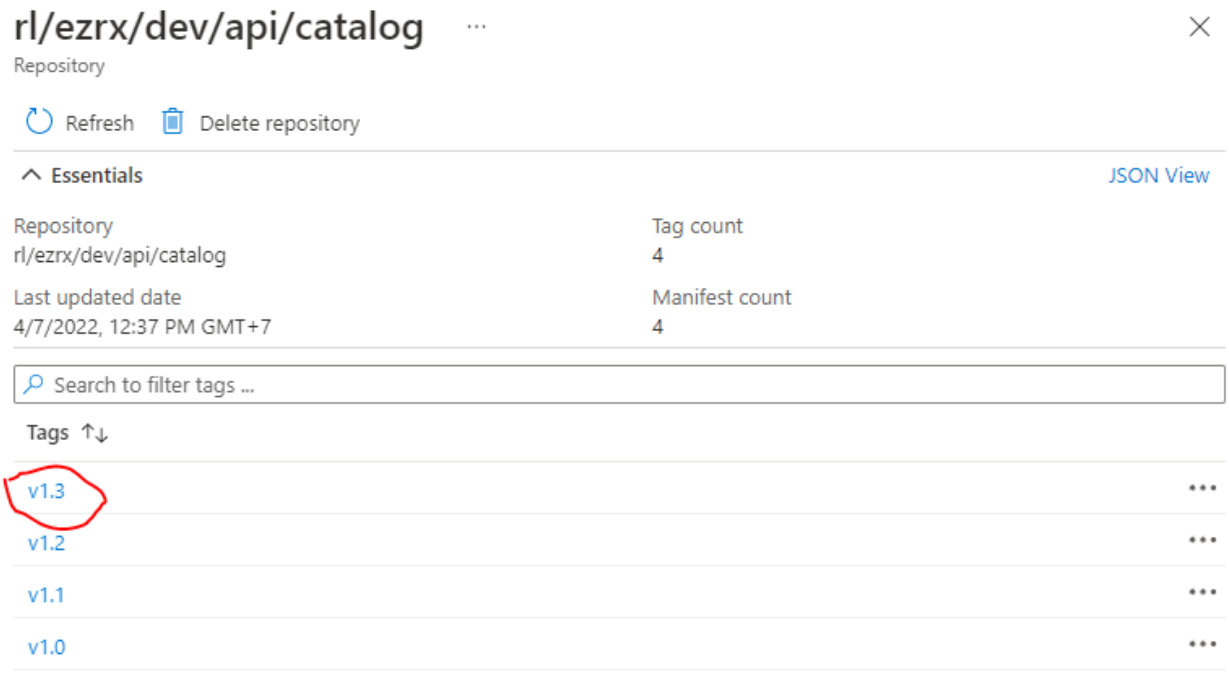
1. Persiapan File Configurasi
    1.1. Buka daftar image catalog di azure
    https://portal.azure.com/#blade/Microsoft_Azure_ContainerRegistries/TagMetadataBlade/id/%2Fsubscriptions%2F1d290f77-e37d-4371-8b28-3ebc6dd0e727%2FresourceGroups%2Frl-ezmic-rg%2Fproviders%2FMicrosoft.ContainerRegistry%2Fregistries%2Frlezmic/repository/rl%2Fezrx%2Fdev%2Fapi%2Fcatalog

Lihat last vesionnya



Dalam kasus ini last versionnya adalah v1.3, maka kita akan membuat versi v1.4

2. Ubah Config

2.1. Buka file apl-ezrx-api\src\Services\Catalog.api\config\config.go

```
// Load reads the settings written to the yml file
func Load() *Config {
    env := "staging-rl"
    if value := os.Getenv("WEB_APP_ENV"); value != "" {
        env = value
    }

    config := &Config{}
    if err := configor.Load(config, "config/application."+env+".yml"); err != nil {
        fmt.Printf("Failed to read application.%s.yml: %s", env, err)
        os.Exit(2)
    }
    config.ENV = env
    return config
}
```

Dalam kasus ini kita akan me-deploy ke staging RL

3. Configurasi version

3.1. Buka file .yaml sesuai dengan environment yang akan kita pakai dalam kasus ini saya membuka file **application.staging-rl.yml** di folder **apl-ezrx-api\src\Services\Catalog.api\config**

3.2. Ubah versi yang akan dibuat

```
api:
  name: Catalog API Staging RL
  port: 80
  version: v1.4
  host: localhost
  baseurl: ezmicbus-aks.radyalabs.id/catalog
auth:
  jwtkey: "EObX23RcZlXz1U6mh2qkwVN7LEivNZK7r8t6U5yw-MaSZJXpI2_2ocTtUGRA6CzX4uzJLGJdodzGRCtHusSA6Ta51Qkq
rpc:
  name: Catalog RPC
  port: 8882
database:
  product:
```

==========Membuat Image di Lokal==========

1. Buat image

docker build . --tag catalog-api:04 <== 04 adalah versi 4

Tunggu sampai muncul seperti dibawah ini

```
[+] Building 365.0s (11/11) FINISHED
 => [internal] load build definition from Dockerfile                                                    1.8s
 => => transferring dockerfile: 32B                                                                     0.4s
 => [internal] load .dockerignore                                                                       2.0s
 => => transferring context: 2B                                                                         0.1s
 => [internal] load metadata for docker.io/library/golang:alpine                                       15.9s
 => [1/6] FROM docker.io/library/golang:alpine@sha256:a2ca4f4c0828b1b426a3153b068bf32a21868911c57a9fc4dccdc5fb  0.1s
 => [internal] load build context                                                                       2.2s
 => => transferring context: 437.59kB                                                                   1.7s
 => CACHED [2/6] RUN apk update && apk add --no-cache git                                               0.0s
 => CACHED [3/6] WORKDIR /cmd                                                                           0.0s
 => [4/6] COPY . .                                                                                      7.4s
 => [5/6] RUN go mod tidy                                                                             173.5s
 => [6/6] RUN go build cmd/main.go                                                                    144.0s
 => exporting to image                                                                                 16.6s
 => => exporting layers                                                                                15.8s
 => => writing image sha256:b42c578c2488102cece5f72ee291ea9feee229eadca5c24a811f0d6a79711c7d            0.1s
 => => naming to docker.io/library/catalog-api:04                                                       0.1s

C:\Users\RizQ\go-path\src\apl-ezrx-api\src\Services\Catalog.api>
```

2. Pastikan image sudah jalan di lokal
   docker run catalog-api:04

```
C:\Users\RizQ\go-path\src\apl-ezrx-api\src\Services\Catalog.api>docker run catalog-api:04
Catalog RPC Starting at  Port:  8882


   ____    __
  / __/___/ /  ___
 / _// __/ _ \/ _ \
/___/\__/_//_/\___/ v4.6.3
High performance, minimalist Go web framework
https://echo.labstack.com
____________________________________O/_______
                                    O\
⇨ http server started on [::]:80
```

3. login ke azure
   az login

Nb : login dengan akun radyalabs masing2. Jika belum punya akses bisa tanya langsung

```
C:\Users\RizQ\go-path\src\apl-ezrx-api\src\Services\Catalog.api>az login
A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please contin
ue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code
flow with `az login --use-device-code`.
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "be9b8878-5c82-47db-bab7-9937544f7cf9",
    "id": "1d290f77-e37d-4371-8b28-3ebc6dd0e727",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Pay-As-You-Go",
    "state": "Enabled",
    "tenantId": "be9b8878-5c82-47db-bab7-9937544f7cf9",
    "user": {
      "name": "rizky@radyalabs.id",
      "type": "user"
    }
  }
]
```

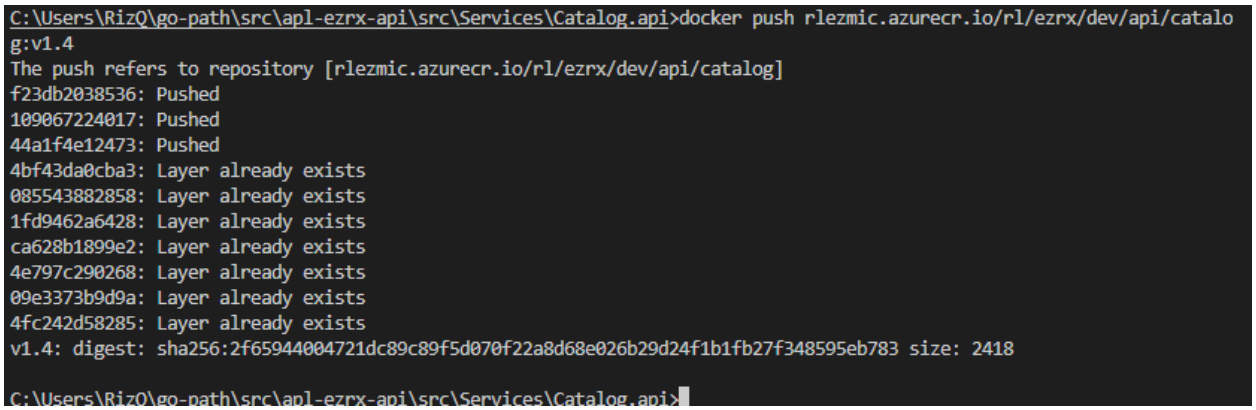
4. login ke rlezmic
   az acr login --name rlezmic

5. Dapatkan rlezmicbus credential

```
az aks get-credentials --resource-group rl-ezmic-rg --name rlezmicbus
```

6. sikron docker local dengan yg di rlezmic
   docker tag catalog-api:04 rlezmic.azurecr.io/rl/ezrx/dev/api/catalog:v1.4

7. push docker ke azure
   docker push rlezmic.azurecr.io/rl/ezrx/dev/api/catalog:v1.4



==========Deploy Aplikasi==========

1. get credential rlezmicbus

   az aks get-credentials --resource-group rl-ezmic-rg --name rlezmicbus

2. masuk ke rlezmicbus

   kubectl config use-context rlezmicbus

```
C:\Users\RizQ\go-path\src\apl-ezrx-api\deploy>kubectl config use-context rlezmicbus
Switched to context "rlezmicbus".
```

3. pastikan config ada

   kubectl config get-contexts

```
C:\Users\RizQ\go-path\src\apl-ezrx-api\deploy>kubectl config get-contexts
CURRENT   NAME         CLUSTER      AUTHINFO                              NAMESPACE
*         rlezmicbus   rlezmicbus   clusterUser_rl-ezmic-rg_rlezmicbus
```

4. ubah version di deployment.yaml

   Buka folder apl-ezrx-api\deploy\k8s\catalog-api

   image: rlezmic.azurecr.io/rl/ezrx/dev/api/**catalog:v1.4** <====== sesuaikan dengan versi image yang sudah dibuat sebelumnya

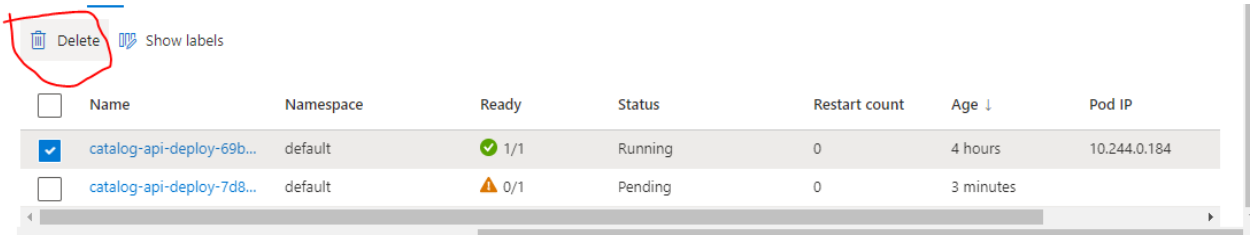
```
    app: catalog-api
replicas: 1
template:
  metadata:
    labels:
      app: catalog-api
  spec:
    containers:
      - name: catalog-api
        image: rlezmic.azurecr.io/rl/ezrx/dev/api/catalog:v1.4
        ports:
        - containerPort: 80
        - containerPort: 8080
        env:
          - name: api__name
            value: "Catalog-API"
          - name: api__port
```

5. apply deployment aplikasi

   kubectl apply -f .\k8s\catalog-api\deployment.yaml

6. Cek pods yang sedang build
Buka url

https://portal.azure.com/#blade/Microsoft_Azure_ContainerService/AksK8ResourceMenuBlade/overview-Service/aksClusterId/%2Fsubscriptions%2F1d290f77-e37d-4371-8b28-3ebc6dd0e727%2FresourceGroups%2Frl-ezmic-rg%2Fproviders%2FMicrosoft.ContainerService%2FmanagedClusters%2Frlezmicbus/resource/%7B%22kind%22%3A%22Service%22%2C%22metadata%22%3A%7B%22uid%22%3A%22765c7d21-2625-4611-8f4f-457dfab94eb9%22%2C%22name%22%3A%22catalog-api-svc%22%2C%22namespace%22%3A%22default%22%7D%7D



Dalam kasus ini pods yg sedang kita buat pending, maka kita harus menghapus pods yang existing dahulu dengan mengklik tombol delete

7. Check pods kita sudah jalan atau belum
kubectl get pods

```
C:\Users\RizQ\go-path\src\apl-ezrx-api\deploy>kubectl get pods
NAME                                     READY   STATUS    RESTARTS   AGE
account-api-deploy-654d6c4b8b-dbrkr      1/1     Running   0          5h35m
catalog-api-deploy-7d84d6f9f5-nw9c7      1/1     Running   0          4m19s
customerapigw-deploy-59f574f884-mdpv9    1/1     Running   0          49d
nginx-ingress-controller-7d84c8d57-7vb75 1/1     Running   0          86d
notification-api-deploy-5ccd7b787d-j72h4 1/1     Running   0          7d3h
qa-ezrx-notification-6bd8b775b6-nnfzp    1/1     Running   13         72d
webcustgw-deploy-759f8cff4f-d7fsb        1/1     Running   0          120d

C:\Users\RizQ\go-path\src\apl-ezrx-api\deploy>
```

8. apply ingress
kubectl apply -f .\k8s\apicust-gw\ingress.yaml

9. Pastikan aplikasi berjalan

Buka http://ezmicbus-aks.radyalabs.id/catalog/swagger/index.html

Pastikan versi yg berhasil di deploy sama dengan yg kita deploy