비밀은 암호, 토큰 또는 키와 같은 소량의 민감한 데이터를 포함하는 오브젝트이다.

시크릿을 사용하면 사용자의 기밀 데이터를 애플리케이션 코드에 넣을 필요가 없다.

**ECR secret 적용**

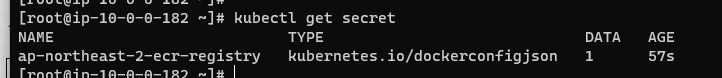
1. 먼저 변수를 지정해야 한다.

|  |
| --- |
| ACCOUNT=7654321XXXXX #사용자 Root 계정의 Account ID 입니다  REGION=ap-northeast-2  SECRET\_NAME=${REGION}-ecr-registry #kubernetes Secret 이름이다.  EMAIL=example123@gmail.com #임의 이메일이다.  TOKEN=`aws ecr --region=$REGION get-authorization-token --output text --query authorizationData[].authorizationToken | base64 -d | cut -d: -f2` |

2. 지정이 완료 되었다면 Kubernetes Secret를 생성해주도록 한다.

|  |
| --- |
| kubectl create secret docker-registry $SECRET\_NAME \  --docker-server=https://${ACCOUNT}.dkr.ecr.${REGION}.amazonaws.com \  --docker-username=AWS --docker-password="${TOKEN}" \  --docker-email="${EMAIL}" |

ap-northeast-2-ecr-registry라는 이름의 시크릿이 생성된 것을 확인할 수 있다.



3. secret 사용하기

|  |
| --- |
| cat << EOF > deployment.yaml  apiVersion: apps/v1  kind: Deployment  metadata:  name: front  labels:  app: ecr  spec:  replicas: 2  selector:  matchLabels:  app: ecr  template:  metadata:  labels:  app: ecr  spec:  containers:  - name: ecr-container  image: xxxxxxxxxxxx.dkr.ecr.ap-northeast-2.amazonaws.com/skills-ecr:latest  ports:  - containerPort: 5000  imagePullSecrets:  - name: ap-northeast-2-ecr-registry  EOF  kubectl apply -f deployment.yaml |

4. Container 상태 확인해보기

|  |
| --- |
| kubectl get pods -A |

