**Minio**

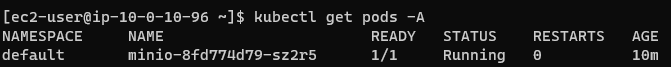
실행하고 있는 서버에서 /data/minio 디렉토리를 생성해준다.

|  |
| --- |
| sudo mkdir -p /data/minio |

우선 deployment를 작성해준다.

|  |
| --- |
| apiVersion: apps/v1  kind: Deployment  metadata:  name: minio  namespace: default  spec:  replicas: 1  selector:  matchLabels:  app: minio  strategy:  type: Recreate  template:  metadata:  labels:  app: minio  spec:  volumes:  - name: storage  hostPath:  path: /data/minio  containers:  - name: minio  image: minio/minio:latest  args:  - server  - --console-address  - ":9001"  - "/storage"  env:  - name: MINIO\_ACCESS\_KEY  value: "minio"  - name: MINIO\_SECRET\_KEY  value: "minio123"  - name: TZ  value: Asia/Seoul  - name: LANG  value: ko\_KR.utf8  ports:  - containerPort: 9000  hostPort: 9000  - containerPort: 9001  hostPort: 9001  volumeMounts:  - name: storage  mountPath: "/storage" |

Pods 조회시 바로 minio가 표시되는 것을 확인할 수 있다.



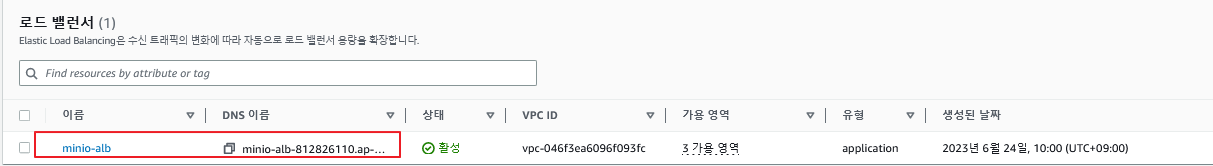
이제 service.yaml을 설정해준다.

|  |
| --- |
| apiVersion: v1  kind: Service  metadata:  name: minio  namespace: default  labels:  run: minio  spec:  ports:  - port: 9000  name: api  - port: 9001  name: ui  selector:  app: minio |

Ingress.yaml을 설정해준다.

|  |
| --- |
| apiVersion: networking.k8s.io/v1  kind: Ingress  metadata:  name: minio-ingress  namespace: default  annotations:  kubernetes.io/ingress.class: alb  alb.ingress.kubernetes.io/scheme: internet-facing  alb.ingress.kubernetes.io/target-type: ip  alb.ingress.kubernetes.io/load-balancer-name: minio-alb  spec:  rules:  - http:  paths:  - pathType: Prefix  path: /  backend:  service:  name: minio  port:  number: 9001 |

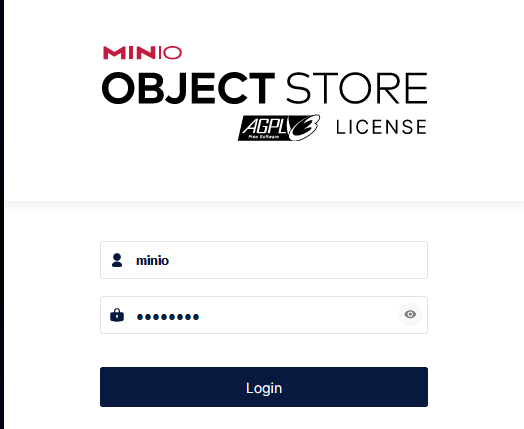
이러면 다음과 같이 로드벨런서가 생성된 것을 확인할 수 있다.



deployment에서 지정하였던 환경변수로 로그인을 진행해준다.

Id: minio

Password: minio123



다음과 같이 성공된 화면이 표시된다.

