Node에 Route53에 대한 권한을 부여해주고 아래와 같이 External DNS를 설치해준다.

|  |
| --- |
| apiVersion: v1  kind: ServiceAccount  metadata:  name: external-dns  labels:  app.kubernetes.io/name: external-dns  ---  apiVersion: rbac.authorization.k8s.io/v1  kind: ClusterRole  metadata:  name: external-dns  labels:  app.kubernetes.io/name: external-dns  rules:  - apiGroups: [""]  resources: ["services", "endpoints", "pods", "nodes"]  verbs: ["get","watch","list"]  - apiGroups: ["extensions", "networking.k8s.io"]  resources: ["ingresses"]  verbs: ["get","watch","list"]  ---  apiVersion: rbac.authorization.k8s.io/v1  kind: ClusterRoleBinding  metadata:  name: external-dns-viewer  labels:  app.kubernetes.io/name: external-dns  roleRef:  apiGroup: rbac.authorization.k8s.io  kind: ClusterRole  name: external-dns  subjects:  - kind: ServiceAccount  name: external-dns  namespace: default  ---  apiVersion: apps/v1  kind: Deployment  metadata:  name: external-dns  labels:  app.kubernetes.io/name: external-dns  spec:  selector:  matchLabels:  app.kubernetes.io/name: external-dns  strategy:  type: Recreate  template:  metadata:  labels:  app.kubernetes.io/name: external-dns  spec:  serviceAccountName: external-dns  securityContext:  fsGroup: 65534  containers:  - name: external-dns  image: bitnami/external-dns:0.13.1  # must specify env AWS\_REGION in AWS china regions  # env:  # - name: AWS\_REGION  # value: cn-north-1  args:  - --source=service  - --source=ingress  - --domain-filter=wsi-korea.org # will make ExternalDNS see only the hosted zones matching provided domain, omit to process all available hosted zones  - --provider=aws  - --policy=upsert-only # would prevent ExternalDNS from deleting any records, omit to enable full synchronization  - --aws-zone-type=public # only look at public hosted zones (valid values are public, private or no value for both)  - --registry=txt  - --txt-owner-id=my-identifier |

그리고 적용해준다.

|  |
| --- |
| $ kubectl apply -f external-dns.yaml  $ kubectl logs -f $(kubectl get po | egrep -o 'external-dns[A-Za-z0-9-]+') |

그리고 ingress를 배포해주자.

|  |
| --- |
| apiVersion: apps/v1  kind: Deployment  metadata:  name: mario  labels:  app: mario  spec:  replicas: 1  selector:  matchLabels:  app: mario  template:  metadata:  labels:  app: mario  spec:  containers:  - name: mario  image: pengbai/docker-supermario  resources:  limits:  memory: 512Mi  cpu: "0.5"  requests:  memory: 256Mi  cpu: "0.2"  ---  apiVersion: v1  kind: Service  metadata:  name: mario  spec:  selector:  app: mario  ports:  - port: 80  targetPort: 8080  protocol: TCP  type: NodePort  ---  apiVersion: networking.k8s.io/v1  kind: Ingress  metadata:  name: ingress  annotations:  kubernetes.io/ingress.class: alb  alb.ingress.kubernetes.io/scheme: internet-facing  alb.ingress.kubernetes.io/target-type: ip  external-dns.alpha.kubernetes.io/hostname: web.wsi-korea.org  external-dns.alpha.kubernetes.io/aws-weight: "100"  external-dns.alpha.kubernetes.io/set-identifier: "3"  spec:  rules:  - http:  paths:  - pathType: Prefix  path: /  backend:  service:  name: mario  port:  number: 80 |

※ 여기서는 파란 색 글씨가 제일 중요하다.



그러면 Route53에 아래와 같이 Records가 추가된 것을 볼 수 있다.

※ 만약 Private Route53 Records를 사용하고자 하면 아래와 같이 args를 수정해주면된다. (아래 파란 색 참고)

|  |
| --- |
| - --source=service  - --source=ingress  - --domain-filter=wsi-korea.internal  - --provider=aws  - --policy=upsert-only  - --aws-zone-type=private  - --registry=txt  - --txt-owner-id=my-identifier |