\*\* EKS - Bastion Host 셋업

1. 루트 어카운트 생성후 로그인하고 eks-admin 사용자 생성

- Mgmt Console 액세스 허용

- 관리자 권한 부여

- 로그인 링크 기억할 것. 다음과 같은 형식

<https://242337484181.signin.aws.amazon.com/console>

- eks-admin 사용자로 로그인(MFA 적용 권장)

2. IAM Role 생성

- EC2에 대한 eks-admin-role 생성

- AdministratorAccess 권한 부여

3. Cloud9 배스쳔 인스턴스 생성

- 서울 리전인지 확인

- Name: eks-bastion

- 나머지는 기본값 : Cost Saving을 위해 자동 종료 시간을 사용하는데 조정 가능함.

4. EC2 인스턴스에서 볼륨 크기 조정, Role 지정

- eks-bastion 이름이 포함된 인스턴스의 세부 항목으로 들어가서 스토리지를 찾음

- vol-xxxxx 형태의 링크를 찾아서 클릭

- EBS 볼륨 화면에서 우측 상단의 '작업' 드롭다운을 클릭하여 '볼륨 수정' 클릭. 10G-->20G로 늘림

- 다시 eks-bastion ec2 인스턴스 화면으로 이동

- 작업 드롭다운에서 보안 - IAM 역할 수정 클릭

- eks-admin-role 연결

- Cloud9 설정 화면으로 이동(우측상단 톱니바퀴 클릭)

- Themes 에서 밝은색으로 선택(예:Flat Theme)

- AWS Settings - Credentials 화면에서 AWS managed temporary credentials를 off 상태로 변경

- 적용 후 터미널에서 다음 명령어 실행하여 기존 영구자격증명 삭제하고 eks-admin-role로 자격증명이 변경되었는지 확인

\* aws configure 명령 실행 후 region과 포맷만 설정 : ap-northeast-2, json

\* rm ~/.aws/credentials

\* aws sts get-caller-identity

5. 유틸리티 파일들 설치

- 몇몇 유틸리티 설치

sudo yum -y install jq bash-completion

\* jq: json 파싱 유틸리티

\* bash\_completion : bash 쉘 자동 완성 설정

- kubectl 설치

sudo curl --location -o /usr/local/bin/kubectl \

https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.9/2023-01-11/bin/linux/amd64/kubectl

sudo chmod +x /usr/local/bin/kubectl

kubectl completion bash >> ~/.bash\_completion

. /etc/profile.d/bash\_completion.sh

. ~/.bash\_completion

- eksctl 설치

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp

sudo mv -v /tmp/eksctl /usr/local/bin

eksctl completion bash >> ~/.bash\_completion

. /etc/profile.d/bash\_completion.sh

. ~/.bash\_completion

6. EKS Cluster 생성

- eks-demo.yaml 파일 생성

|  |
| --- |
| apiVersion: eksctl.io/v1alpha5  kind: ClusterConfig  metadata:  name: demo-cluster  region: ap-northeast-2  version: "1.24"  availabilityZones: ["ap-northeast-2a", "ap-northeast-2b", "ap-northeast-2c"]  managedNodeGroups:  - name: nodegroup  desiredCapacity: 2  instanceType: t3.small  ssh:  enableSsm: true |

- 클러스터 생성

eksctl create cluster -f eks-demo.yaml

7. 클러스터 삭제

eksctl delete cluster --name demo-cluster

8. 평상시 사용하지 않을때 비용 절감을 위해 다음 작업

- 클러스터 삭제하고

- eks-bastion 인스턴스 중지(Stopping) : 종료(terminating) 아님!!