eksctl 명령을 이용해서 AWS EKS 생성하기

ubuntu ec2 instance 생성

- 최소 사양으로 인스턴스 생성
- 아래 관리 도구를 설치

aws cli tool install

- https://docs.aws.amazon.com/ko_kr/cli/latest/userguide/install-cliv2-linux.html

```
sudo apt-get install -y unzip
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
"awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
aws --version
```

aws 계정 연결하기

```
aws configure
# 사용자 계정에서 받은 키 값 넣기
AWS Access Key ID [None]: 액세스키
AWS Secret Access Key [None]: 비밀_액세스키
Default region name [None]: ap-northeast-2
Default output format [None]: <Enter>

# aws 연결 arn 확인
aws sts get-caller-identity
```

eksctl tool install

- EKS설치/운영 툴인 eksctl 설치
- https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/eksctl.html

```
# for ARM systems, set ARCH to: `arm64`, `armv6` or `armv7`
ARCH=amd64
PLATFORM=$(uname -s)_$ARCH
```

```
curl -sLO "https://github.com/eksctl-
io/eksctl/releases/latest/download/eksctl_$PLATFORM.tar.gz"

# (Optional) Verify checksum
curl -sL "https://github.com/eksctl-
io/eksctl/releases/latest/download/eksctl_checksums.txt" | grep $PLATFORM |
sha256sum --check

tar -xzf eksctl_$PLATFORM.tar.gz -C /tmp && rm eksctl_$PLATFORM.tar.gz

sudo mv /tmp/eksctl /usr/local/bin
```

kubectl install

- https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/install-kubectl.html

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.7/2023-11-
14/bin/linux/amd64/kubectl
chmod +x ./kubectl
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export
PATH=$HOME/bin:$PATH
echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc
kubectl version --short --client
```

EKS 구성

- 참고: https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/getting-started-eksctl.html
- Amazon EKS 클러스터에 대해 시간당 0.10 USD
- Kubernetes 워커 노드를 실행하기 위해 생성하는 AWS 리소스(예: EC2 인스턴스, Amazon Elastic Block Store(EBS) 볼륨, Outposts 용량)에 대한 비용
- t3.medium으로 설정할 경우 대당 -> 0.052 USD 시간당
- eksctl 명령을 실행해서 Amazon EKS 클러스터를 생성

```
eksctl create cluster \
    --name k8s-encore \
    --region ap-northeast-2 \
    --with-oidc \
    --ssh-access \
    --ssh-public-key encore_17_key \
    --nodes 3 \
    --node-type t3.medium \
```

```
    --node-volume-size=20 \
    --managed
```

- 참고: Amazon Elastic Kubernetes Service(Amazon EKS)에서 OpenID Connect(OIDC) 호환 자격 증명 공급자를 Kubernetes 클러스터에 대한 사용자 인증 옵션으로 사용할 수 있음
- OIDC 인증을 사용하면 직원 계정의 생성, 활성화 및 비활성화에 대한 조직의 표준 절차를 사용하여 EKS 클러스터에 대한 사용자 액세스를 관리할 수 있습니다.
- https://aws.amazon.com/ko/about-aws/whats-new/2021/02/amazon-eks-clusters-support-user-authentication-oidc-compatible-identity-providers/
- CloudFormation으로 생성되기 때문에 aws에서 cloudformation으로 확인 가능
- 생성되는 시간이 20분정도 걸린다.

```
2024-02-04 17:16:24 [ℹ]  eksctl version 0.170.0
2024-02-04 17:16:24 [ℹ]  using region ap-northeast-2
2024-02-04 17:16:24 [ℹ]  setting availability zones to [ap-northeast-2b ap-
northeast-2a ap-northeast-2d]
2024-02-04 17:16:24 [ℹ]  subnets for ap-northeast-2b - public:192.168.0.0/19
private:192.168.96.0/19
2024-02-04 17:16:24 [ℹ]  subnets for ap-northeast-2a -
public:192.168.32.0/19 private:192.168.128.0/19
2024-02-04 17:16:24 [ℹ]  subnets for ap-northeast-2d -
public:192.168.64.0/19 private:192.168.160.0/19
2024-02-04 17:16:24 [ℹ]  nodegroup "ng-b68cbf6a" will use ""
[AmazonLinux2/1.27]
2024-02-04 17:16:24 [ℹ]  using EC2 key pair "encore_17_key"
2024-02-04 17:16:24 [ℹ]  using Kubernetes version 1.27
2024-02-04 17:16:24 [ℹ]  creating EKS cluster "k8s-encore" in "ap-northeast-
2" region with managed nodes
2024-02-04 17:16:24 [ℹ]  will create 2 separate CloudFormation stacks for
cluster itself and the initial managed nodegroup
2024-02-04 17:16:24 [ℹ]  if you encounter any issues, check CloudFormation
console or try 'eksctl utils describe-stacks --region=ap-northeast-2 --
cluster=k8s-encore'
2024-02-04 17:16:24 [ℹ]  Kubernetes API endpoint access will use default of
{publicAccess=true, privateAccess=false} for cluster "k8s-encore" in "ap-
northeast-2"
2024-02-04 17:16:24 [ℹ]  CloudWatch logging will not be enabled for cluster
"k8s-encore" in "ap-northeast-2"
2024-02-04 17:16:24 [ℹ]  you can enable it with 'eksctl utils update-
cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --
region=ap-northeast-2 --cluster=k8s-encore'
2024-02-04 17:16:24 [ℹ]
2 sequential tasks: { create cluster control plane "k8s-encore",
    2 sequential sub-tasks: {
```

```
        4 sequential sub-tasks: {
            wait for control plane to become ready,
            associate IAM OIDC provider,
            2 sequential sub-tasks: {
                create IAM role for serviceaccount "kube-system/aws-node",
                create serviceaccount "kube-system/aws-node",
            },
            restart daemonset "kube-system/aws-node",
        },
        create managed nodegroup "ng-b68cbf6a",
    }
}
2024-02-04 17:16:24 [ℹ]  building cluster stack "eksctl-k8s-encore-cluster"
2024-02-04 17:16:24 [ℹ]  deploying stack "eksctl-k8s-encore-cluster"
2024-02-04 17:16:54 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-cluster"
2024-02-04 17:17:24 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-cluster"
2024-02-04 17:18:24 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-cluster"
2024-02-04 17:19:24 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-cluster"
2024-02-04 17:20:24 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-cluster"
2024-02-04 17:21:24 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-cluster"
2024-02-04 17:22:24 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-cluster"
2024-02-04 17:23:25 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-cluster"
2024-02-04 17:25:26 [ℹ]  building iamserviceaccount stack "eksctl-k8s-
encore-addon-iamserviceaccount-kube-system-aws-node"
2024-02-04 17:25:26 [ℹ]  deploying stack "eksctl-k8s-encore-addon-
iamserviceaccount-kube-system-aws-node"
2024-02-04 17:25:26 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-addon-iamserviceaccount-kube-system-aws-node"
2024-02-04 17:25:56 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-addon-iamserviceaccount-kube-system-aws-node"
2024-02-04 17:25:56 [ℹ]  serviceaccount "kube-system/aws-node" already
exists
2024-02-04 17:25:56 [ℹ]  updated serviceaccount "kube-system/aws-node"
2024-02-04 17:25:56 [ℹ]  daemonset "kube-system/aws-node" restarted
2024-02-04 17:25:57 [ℹ]  building managed nodegroup stack "eksctl-k8s-
encore-nodegroup-ng-b68cbf6a"
2024-02-04 17:25:57 [ℹ]  deploying stack "eksctl-k8s-encore-nodegroup-ng-
b68cbf6a"
```

```
2024-02-04 17:25:57 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-nodegroup-ng-b68cbf6a"
2024-02-04 17:26:27 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-nodegroup-ng-b68cbf6a"
2024-02-04 17:27:08 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-nodegroup-ng-b68cbf6a"
2024-02-04 17:28:44 [ℹ]  waiting for CloudFormation stack "eksctl-k8s-
encore-nodegroup-ng-b68cbf6a"
2024-02-04 17:28:44 [ℹ]  waiting for the control plane to become ready
2024-02-04 17:28:45 [✔]  saved kubeconfig as "/home/ubuntu/.kube/config"
2024-02-04 17:28:45 [ℹ]  no tasks
2024-02-04 17:28:45 [✔]  all EKS cluster resources for "k8s-encore" have
been created
2024-02-04 17:28:45 [ℹ]  nodegroup "ng-b68cbf6a" has 3 node(s)
2024-02-04 17:28:45 [ℹ]  node "ip-192-168-17-209.ap-northeast-
2.compute.internal" is ready
2024-02-04 17:28:45 [ℹ]  node "ip-192-168-53-241.ap-northeast-
2.compute.internal" is ready
2024-02-04 17:28:45 [ℹ]  node "ip-192-168-76-125.ap-northeast-
2.compute.internal" is ready
2024-02-04 17:28:45 [ℹ]  waiting for at least 3 node(s) to become ready in
"ng-b68cbf6a"
2024-02-04 17:28:45 [ℹ]  nodegroup "ng-b68cbf6a" has 3 node(s)
2024-02-04 17:28:45 [ℹ]  node "ip-192-168-17-209.ap-northeast-
2.compute.internal" is ready
2024-02-04 17:28:45 [ℹ]  node "ip-192-168-53-241.ap-northeast-
2.compute.internal" is ready
2024-02-04 17:28:45 [ℹ]  node "ip-192-168-76-125.ap-northeast-
2.compute.internal" is ready
2024-02-04 17:28:46 [ℹ]  kubectl command should work with
"/home/ubuntu/.kube/config", try 'kubectl get nodes'
2024-02-04 17:28:46 [✔]  EKS cluster "k8s-encore" in "ap-northeast-2" region
is ready
```

확인

```
ubuntu@ip-172-30-0-155:~$ kubectl get nodes
NAME                                               STATUS   ROLES    AGE
VERSION
ip-192-168-17-209.ap-northeast-2.compute.internal   Ready    <none>   67s
v1.27.9-eks-5e0fdde
ip-192-168-53-241.ap-northeast-2.compute.internal   Ready    <none>   66s
v1.27.9-eks-5e0fdde
ip-192-168-76-125.ap-northeast-2.compute.internal   Ready    <none>   72s
v1.27.9-eks-5e0fdde
```

CLI 명령어 완성기능 추가

```
source <(kubectl completion bash)
echo "source <(kubectl completion bash)" >> ~/.bashrc
```

실행 실습

```
# 워커 노드 정보 보기
kubectl get nodes -o wide
```

- Pod 배포 TEST. nginx 컨테이너 5개 실행하고 결과 확인

```
kubectl create  deployment webtest --image=nginx:1.14 --port=80  --
replicas=5
```

```
kubectl get  pods -o wide
```

eks 삭제

```
eksctl delete cluster --name k8s-encore
```

- 삭제하는 시간도 10분 이상 걸림