

AWS 클러스터 생성 (EKS : Elastic Kubernetes Service 생성하기)

```
eksctl create cluster --name (Cluster-Name) --version 1.21 --nodegroup-name standard-workers --node-type t3.medium --nodes 3 --nodes-min 1 --nodes-max 3
```

```
Problems  docker  root@labs-910775232: /home/project/Docker  root@labs-910775232: /home/project  ×  Output
2022-03-17 00:27:16 [i] waiting for CloudFormation stack "eksctl-user003003-eks-cluster"
2022-03-17 00:28:17 [i] waiting for CloudFormation stack "eksctl-user003003-eks-cluster"
2022-03-17 00:29:17 [i] waiting for CloudFormation stack "eksctl-user003003-eks-cluster"
2022-03-17 00:30:17 [i] waiting for CloudFormation stack "eksctl-user003003-eks-cluster"
2022-03-17 00:31:17 [i] waiting for CloudFormation stack "eksctl-user003003-eks-cluster"
2022-03-17 00:32:18 [i] waiting for CloudFormation stack "eksctl-user003003-eks-cluster"
2022-03-17 00:33:18 [i] waiting for CloudFormation stack "eksctl-user003003-eks-cluster"
2022-03-17 00:34:18 [i] waiting for CloudFormation stack "eksctl-user003003-eks-cluster"
2022-03-17 00:35:18 [i] waiting for CloudFormation stack "eksctl-user003003-eks-cluster"
2022-03-17 00:36:18 [i] waiting for CloudFormation stack "eksctl-user003003-eks-cluster"
2022-03-17 00:38:20 [i] building managed nodegroup stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:38:20 [i] deploying stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:38:20 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:38:39 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:38:55 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:39:13 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:39:31 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:39:49 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:40:07 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:40:25 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:40:41 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:41:01 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:41:20 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:41:37 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:41:56 [i] waiting for CloudFormation stack "eksctl-user003003-eks-nodegroup-standard-workers"
2022-03-17 00:41:56 [i] waiting for the control plane availability...
2022-03-17 00:41:56 [✓] saved kubeconfig as "/root/.kube/config"
2022-03-17 00:41:56 [i] no tasks
2022-03-17 00:41:56 [✓] all EKS cluster resources for "user003003-eks" have been created
2022-03-17 00:41:56 [i] nodegroup "standard-workers" has 3 node(s)
2022-03-17 00:41:56 [i] node "ip-192-168-19-0.ap-northeast-2.compute.internal" is ready
2022-03-17 00:41:56 [i] node "ip-192-168-45-115.ap-northeast-2.compute.internal" is ready
2022-03-17 00:41:56 [i] node "ip-192-168-74-130.ap-northeast-2.compute.internal" is ready
2022-03-17 00:41:56 [i] waiting for at least 1 node(s) to become ready in "standard-workers"
2022-03-17 00:41:56 [i] nodegroup "standard-workers" has 3 node(s)
2022-03-17 00:41:56 [i] node "ip-192-168-19-0.ap-northeast-2.compute.internal" is ready
2022-03-17 00:41:56 [i] node "ip-192-168-45-115.ap-northeast-2.compute.internal" is ready
2022-03-17 00:41:56 [i] node "ip-192-168-74-130.ap-northeast-2.compute.internal" is ready
2022-03-17 00:41:59 [i] kubectl command should work with "/root/.kube/config", try 'kubectl get nodes'
2022-03-17 00:41:59 [✓] EKS cluster "user003003-eks" in "ap-northeast-2" region is ready
```

EKS란?

클라우드 또는 온프레미스에서 Kubernetes 애플리케이션을 실행하고 크기를 조정하는 관리형 컨테이너 서비스

AWS 클러스터 토큰 가져오기

```
aws eks --region (Region-Code) update-kubeconfig --name (Cluster-Name)
kubectl get all
# 클러스터 설정확인
kubectl config current-context
```

```
root@labs-910775232:/home/project# aws eks --region ap-northeast-2 update-kubeconfig --name user003003-eks
Added new context arn:aws:eks:ap-northeast-2:979050235289:cluster/user003003-eks to /root/.kube/config
root@labs-910775232:/home/project#
root@labs-910775232:/home/project# kubectl get all
NAME                                TYPE                CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes                  ClusterIP          10.100.0.1    <none>         443/TCP    13m
root@labs-910775232:/home/project# kubectl config current-context
arn:aws:eks:ap-northeast-2:979050235289:cluster/user003003-eks
```

Metric-Server 설치

```
kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
```

```
root@labs-910775232:/home/project# kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
serviceaccount/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
service/metrics-server created
deployment.apps/metrics-server created
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created
```

Metrics Server는 Kubernetes 의 컴포넌트들에 대한 자원 모니터링이 가능해지며 이것을 이용해 Autoscaling 에도 사용이 가능하다.

WorkerNode 확인 : kubectl top node

```
root@labs-910775232:/home/project# kubectl top node
```

NAME	CPU(cores)	CPU%	MEMORY(bytes)	MEMORY%
ip-192-168-19-0.ap-northeast-2.compute.internal	46m	2%	485Mi	14%
ip-192-168-45-115.ap-northeast-2.compute.internal	236m	12%	487Mi	14%
ip-192-168-74-130.ap-northeast-2.compute.internal	45m	2%	486Mi	14%

3개인 이유는?

앞서 AWS 클러스터 생성시 워커노드 3개를 생성하겠다고 함