minikube 
kubeadm 
amazon 
web services 
Google Cloud 

Biz bugüne kadar kubeadm kurulumu yaparak kubernetes kullandık. Minikube ile de kurulum yapılıyor ancak bunların dışında cloud providerların da kubernetes servisleri var.

Amazon un container orchestration olarak iki servisi var. Biri EKS, Kubernetes kullanımını sağlayan servisi. Bir de ECS Elastic Container Service bu da Amazon un kendi container orchestration tool udur.

Kubemetes Control Plane 
kube•controller 
manager 
etcd 
cloud-controller 
manager 
kube•scheduler 
kubelet 
kube• proxy 
Cloud 
kubelet 
kube-proxy 
Kubernetes Nodes 
kubelet 

Kubernetes Control Plane de cloud controller manager ımız da bugün çalışacak bunun görevi cloud ile olan iletişimi sağlamaktır. Cloud a giriş kapımız gibi düşünebiliriz.

Amazon EKS 
Create Kubernetes 
clusters (powered by 
Amazon EKS Distro) 
AWS Fargate 
Deploy serverless 
containers 
Amazon EC2 
Deploy worker 
nodes for your 
EKS cluster 
Run Kubernetes apps 
Amazon EKS 
dashboard in the 
AWS console 
View and explore running 
Kubernetes apps 

Control Plane i EKS yönetiyor biz görmüyoruz arka planda 3 tane instance açıyor bunun için. Bize de worker node olarak instance lar kaldırıyor.

Fargate ise serverless container hizmetidir, biz ec2 görmüyoruz bize podları veriyor.

EKS de bu iki yöntemden birini seçerek uygulamamızı deploy ediyoruz.

Service deployment vs. Gibi bilgileri de buradan alabiliyoruz.

bugünkü kurulumlar lokalde de yapılabilir ancak derste tek düzeliği sağlamak için instance üzerinde yapacağız

K8s with AWS EKS setup 
Home 
AWS EC2 
O 
Network 
install: 
AWS CLI 
kubectl 
ssh connection 
Network 
Control Plane/Masters 
O 
O 
O 
O 
CLARUSWAYC 
interaction with kubectl CLI 
ssh connection 
Workers Pool 
O 
o 
K8s C uster 

Provision of K8s with 
kubectl & aws cli 
configuration 
Provision an EKS cluster 
EKS automatically deploys 
Kubernetes masters 
CLARUSWAYO 
ro REINVENT YOURSELF 
o 
AWS EKS 
Configuring 
Cluster 
Autoscaler 
creating a 
kubeconfig file 
Deploy worker nodes 
Add worker nodes to your 
EKS cluster 
Connect to EKS 
Point your favorite 
Kubernetes tooling at your 
EKS cluster 
Run Kubemetes apps 
Deploy your Kubernetes 
applications to your EKS cluster 

Role 
& Policies 
Provision an EKS cluster 
EKS automatically deploys 
Kubernetes masters 
IAM Role: (jb-eks-cluster) 
EKS - Cluster policy: 
- AmazonEKSCIusterPoIicy 
Deploy worker nodes 
Add worker nodes to your 
EKS cluster 
Connect to EKS 
Point your favorite 
Kubernetes tooling at your 
EKS cluster 
Run Kubemetes apps 
Deploy your Kubernetes 
applications to your EKS cluster 
IAM Role: (jb-k8s-worker-pool) 
EC2 - Common policies 
- AmazonEKSWorkerNodePoIicy 
- AmazonEC2ContainerRegistryReadOnly 
- AmazonEKS-CNl-Policy 

EKS nin IAM role ihtiyacı olacak.

Şimdi EKS servisine gidelim ve cluster oluşturalım. çok kullanılan bir tool.

Add cluster dan create diyoruz.

Containers 
Elastic Kubernetes Service 
(Amazon EKS) 
Fully managed Kubernetes 
control plane 
Add cluster 
Add cluster 
Create 
Register 

gerçek ortamda da en az 3 tane master oluyor AWS bunun için saatlik 0.10 usd ücretlendirme yapıyor.

fargate ise pahalı bir servis. anlık kısa süreli kullanılır. tamamen AWS tarafından yönteilen bir servistir.

EKS te ise control plane i yani master node u AWS yönetiyor.

cluster a bir isim veriyoruz, kubernetes version seçiyoruz. kubernetes te version önemlii çünkü desteğini son 3 version a kadar sağlıyor.

More information in the version skeü' policy document, 
Release History 
1.2E 
Lateşt Releaşe: 1,263 (releaşed: 2023-03-15) 
End of Life: 2024-02-28 
Patch Releases: 1.26.0, 1,26.1, 1.26.2, 1.26.3 
Complete 1 , 26 Schedule and Changelag 
1.25 
Latest Release: 1.25,8 (released: 2023-03-15) 
End of Life: 2023-10-28 
Patch Releases: 1.25.0, 1.25.1, 1.25.2, 1.25.3, 1.25.4, 1.25.5, 1 256, 1.25.7, 1.25.8 
Complete 1.2S Schedule and Changelag 
1.24 
Latest Release: 124,12 (released: 2023-03-15) 
End of Life: 2023-0728 
Patch Releases: 1.240, 1 24.1, 1.24_3, 1 246. 1.248, 1 24 g 

k8s in kendi sayfasında 1.26 var EKS de 1.25 default şu anda.

Kubernetes version Info 
Select th2 Kubernetes version for this Cluster. 
ı.25 
ı.25 
Default 
1.24 
1.23 
1.22 

bu nedenle dikkat etmemiz gerekiyor. yılda bir de kullandığımız versionu güncelleriz.

Release History Version Skew Policy 
Version Skew Policy 
This document describes the maximum version skew supported between various Kubernetes components. Specific cluster 
deployment tools may place additional restrictions on version skew. 
Supported versions 
Kubernetes versions are expressed as x.y.z. where x is the major version, y is the minor version, and z is the patch version. 
following Semantic Versioning terminology For more information, see Kubernetes Release Versioning. 
The Kubernetes project maintains release branches for the most recent three minor releases (1.26, I IS. I .24). Kubernetes I .19 
and newer receive approximately I year of patch support. Kubernetes I. 18 and older received approximately 9 months of 
patch support. 
Applicable fixes, including security fixes, may be Backported to those three release branches, depending on sevariry and 
feasibility, patch releases are cut from those branches at a regular cadence, plus additional urgent releases, when required, 
The Release Managers group owns this decision, 
For more information, see the KLIbernetes patch releases page. 
Supported version skew 

kubernetes componentlerinin de versionları var. bunlar arasında uyumluluk olmalı. Version Skew Policy deniyor buna

Supported version skew 
kube-apiserver 
In highly-available (HA) clusters, the newest and Oldest 
Example: 
is at I .26 
other instances are supported at 1.26 and 1.25 
kubelet 
instances must be within one minor version. 
kubelet must not be newer than kube-apiserver and may be up to two minor versions older. 
Example: 
kube•apiserver• is at 1.26 
is supported at 1.26, 1.25, and 1.24 
Note: If version skew exists between instances in an HA cluster, this narrows the allowed versions. 
Example: 
kube-apiserver 
instances are at 1.26 and 1.25 
kubelet is supported at 1.25, and 1.24 (1.26 is not supported because that would be newer than the kibe-a,piserver 
instance at version 1.25) 
kube-controller-manager, kube-scheduler, and cloud-controller-manager 
, , and 
must not be newer than the 
insta nces 

bu nedenle güncellerken aşama aşama güncelleme yapıyoruz.

bizim için bugün EKS cluster da kendisi kuracak ancak biz instance a kubetl kuracaz bizim de bunun versionuna dikkat etmemiz gerekiyor.

Configure cluster 
Cluster configuration Info 
Enter a unique name for this cluster. This property cannot be changed after the cluster is created. 
emre 
The cluster name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens 
and underscores. Maximum length of 100. 
Kubernetes version Info 
Select the Kubernetcs versicm for this cluster. 
1.25 
Cluster service role Info 
Select the IAM allow the Kubernetcs control plane to manage AWS 02sourccs on your behalf. This property cannot be changed after 
the cluster is created. To create a new role, follow the instructions in the Amazon EKS User Guide@ 
Select role 
c 

IAM role istediği için gidip role oluşturalım. IAM role u EKS kullanacak seçiyoruz:

use case 
Allow an AWS service like EC2, Lambda, or others to pedorm actions in this account 
Common use cases 
O 
EC2 
Allcms EC2 instances to call AWS services on your behatf 
Lambda 
Allcms Lambda functions to call AWS services on your behalf 
Use cases for other AWS services: 
EKS 
O 
EKS 
Allows EKS to manage clusters on your behalf 
O EKS - Cluster 
Allows access to other AWS service resources that are required to operate clusters man 
O 
EKS - Nodegroup 

EKS cluster ı seçiyoruz. amazon servislerini kullanmak için veriyoruz bu rolü.

Add permissions 
Permissions policies 
(I) Info 
The type of role that you selected requires the following policy. 
Policy name t? 
AmazonEKSCluster 
Type 
AL'S 
Attached entities 
Set permissions boundary - optional Info 
Set a permissions boundary to control the maximum permissions this role can have This not a com 

policy zaten tanımlı. ileri gidip role e isim verip create diyoruz.

EKS servisine dönüp role ün yanından refresh edip role ü seçelim.

Secrets encryption Info 
Once turned on, secrets encryption cannot be modified or removed. 
Turn on envelope encryption of Kubernetes secrets using KMS 
Envelope encryption provides an additional layer of encryption for your Kubernetes secrets. 

secret larımız vars encrytpe etmek için kullanılıyor burası.

default vpc seçecez ancak gerçekte kubernetes cluster için ayrı bir vpc olur. subnet te a b c kalacak diğerlerini silelim çünkü yavaş kalıp sistemde sıkıntı çıkarıyorlar.

Amazon EKS security group requirements and considerations 
PDF Rss 
This topic describes the security group requirements Of Amazon EKS cluster _ 
When create a cluster, Amazon ERS creates a security groupTthatS named eks- clu 
Ste r 
-Sg- - This security group has the folltMing default 
Rule type 
Inbound 
Outbound 
Important 
Protocol 
Ports 
Source 
Destination 
0.000/0 ( 
If your cluster doesn't need the Outbound rule, you can it. If you it, you must Still have the minimum rules listed in Restricting cluster traffic. If you remove the inbo 
Amazon EKS recreates it whenever the cluster is updated. 
Amazon EKS adds the following tags to the security gmup_ If you remove the tags, Amazon EKS adds them back to the Secunty group whenever 
cluster is updated 

default olarak EKS cluster olunca bizim için cluster içi iletişimi sağlayan sec grp oluşturuyor. biz istiyorsak ekstradan seçebiliriz. önemli olan HTTPS ve SSH açık olması. defaultu seçip devam edebiliriz.

Cluster endpoint access Info 
Configure access to the Kubernetes API server endpoint 
O public 
The cluster endpoint is accessible from outside of Y)ur VPC_ Worker node traffic will leave your VPC to connect to the endpoint 
C) Public and private 
The cluster endpoint is accessible from outside of Y)ur VPC_ Worker node traffic to the endpoint will stay within Y)ur VPC_ 
C) Private 
The cluster endpoint is only accessible through pur VPC Worker node traffic to the endpoint will stay within pur VPC 
Advanced settings 

public deyince master worker a vpc dışından ve biz master a dışarıdan erişebiliriz.

public and private master public ten worker private tan ulaşılabilsin.

private master ve worker a sadece private erişilebilir.

public and private seçip next diyoruz.

Select add-ons 
from multiple categMie tt-m to 
Amazon EKS add-ons (4) Info 
kube-proxy Info 
@Instzued 
Amazon GuardDuty 
EKS Runtime 
Monitoring 
ud Within clustev. 
within 
CoreDNS 
service eithin 
rki ng 
Amazon vpC CNI info 
pod 
Categtvy 
Canæl 

gerekli add-on lar bizim karışmamıza gerek yok. next next deyip create diyorz.

EKS > Clusters 
Clusters (1) Info 
Q Filter cluster by name, status, kubernetes version, or provider 
c 
Kubemetes version 
125 
Delete 
v 
ster v 
Provider 
O 
Cluster name 
emre 
Status 
Creating 

hazır olması uzun sürer 15 dk kadar.

ec2 muzu da kuralım. lokalden de yapabiliriz bunu.

sadece ssh olan secgrp yeterli.

ec2 muzu update edip aws credentials yükleyecez. CLI zaten yüklü geldi. IAM role ile de yapabiliriz ancak biz aws credentials ile devam edecez.

curl -O <https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.7/2023-03-17/bin/linux/amd64/kubectl>

komutuyla kubectl yüklüyoruz bu komudu AWS EKS docs tan aldık.

chmod +x ./kubectl ile kubectl i executable hale getiriyirouz.

mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin

ile PATH değişkenine tanıttık.

bshrc içinde en alt satıra birkaç kez enter lıyoruz.

echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc

komutunu giriyoruz. sonra kubectl versionuna bakalım:

kubectl version --short --client

halen bir pod ya da node umuz yok. çünkü EKS sadece control plane oluşturdu.

aws credentials girelim aws configure komutuyla.

aws configuration 
bash 
aws configure 
AWS Access Key ID [None] : xxxxxxx 
AWS Secret Access Key [None] : xxxxxxxx 
Default region name [None 
]: us-east-I 
Default output format [None]: json 

bizim user olarak admin role ümüz var. o nedenle ec2 ile EKS nin içindeki cluster ı görebilmemiz lazım:

• ec2-use 
"clusters" . 
" emre" 
aws eks list-clusters 

şimdi kubeconfig dosyası oluşturacağız. adından ankalıkacağı üzere configürasyon dosyası. . ile başlayan dizinle config dosyalarını tutuyor:

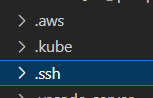
config 
credentials 
v .ssh 
authorized_keys 
> .vscode-server 

kubernetes için de oluşturacağız. çünkü bu makinede kubernetes yüklü değil, master ya da worker node yok. AWS CLI komutu var bunu oluşturmak için. bu komut config dosyası varsa onu update de eder.

aws eks --region <us-east-1> update-kubeconfig --name <cluster\_name>

credentials girdiğimiz için bu makinenin AWS servislerine erişimi var. biz de cluster bölgesi ve ismini girdiğimizde o cluster ı bulabilecek.

- • ec2-use 
o ec2-use 
rap-172-31-82 
aws eks 
-region us-east-I update-kubeconfig - -name emre 
Added new context arn:aws:eks:us-east-1:99571U6521Ø:c1uster/emre to /home/ec2-user/ .kube/config 
-147# 



.kube dosyası geldi. içeriği de şu şekilde config dosyasının:

apiVersion: v1

clusters:

- cluster:

certificate-authority-data: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

server: [https://25E6C551F839A8F7CD144F5F50DE665E.gr7.us-east-1.eks.amazonaws.com](https://25e6c551f839a8f7cd144f5f50de665e.gr7.us-east-1.eks.amazonaws.com/)

name: arn:aws:eks:us-east-1:995714465210:cluster/emre

contexts:

- context:

cluster: arn:aws:eks:us-east-1:995714465210:cluster/emre

user: arn:aws:eks:us-east-1:995714465210:cluster/emre

name: arn:aws:eks:us-east-1:995714465210:cluster/emre

current-context: arn:aws:eks:us-east-1:995714465210:cluster/emre

kind: Config

preferences: {}

users:

- name: arn:aws:eks:us-east-1:995714465210:cluster/emre

user:

exec:

apiVersion: client.authentication.k8s.io/v1beta1

args:

- --region

- us-east-1

- eks

- get-token

- --cluster-name

- emre

command: aws

şimdi konsoldan worker node oluşturacağız. bunun için node group oluşturulması gerekiyor.

EKS > Clusters > 
emre 
em re 
Cluster info Info 
Kubernetes version Info 
1.25 
Overview 
Resources 
Status 
@ Active 
Compute 
Networkinc 

cluster a tıklayıp compute kısmına geldik. burada

Node groups (O) Info 
Group name 
Desired size 
v 
Edit 
V 
Delete 
Add node group 
AMI release version 
No node groups 
Launch template v 
Status 
This cluster does not have any node groups. 
Nodes that are not part of an Amazon EKS managed node group are not shov.m in the AWS console. 
Add node group 

add node group diyoruz.

isim veriyoruz.

burada da bir role gerekecek. readme deki policy leri seçiyoruz. bu kez yetkiyi ec2 ya verecez.

AmazonEKSClusterPolicy 
AmazonEKSWorkerNodePolicy 
AmazonEKSServicePolicy 
AmazonEKS CNI_PoIicy 
îi AWSFaultIn• 
lectłonSimulatorEKSAccess 

Permissions policy summary 
Policy name t? 
Amazon EC2ContainerRegistryReadOnly 
AmazonEKSWorkerNodePolicy 
Type 
AWS managed 
AWS managed 
AWS managed 
Attached as 
Permissions policy 
Permissions policy 
Permissions policy 

EKS node group sayfasına geri dönüp oluşturduğumuz role ü atıyoruz. next diyoruz.

Node group compute configuration 
These properties cannot be changed after the node group is created. 
AMI type Info 
Select the EKS-optimized Amazon Machine Image for nodes. 
Amazon Linux 2 (AL2 x86_64) 
Capacity type 
Select the capacity purchase option for this node group. 
On-Demand 
Instance types Info 
Select instance types you prefer for this node group. 
Select 
t3.medium 
vcpu: 2 vCPUs 
Disk size 
x 
Memory 4 Gig 
Netwnyrk: Up to 5 Gigabit 
Max ENI: 3 
Select the size of the attached EBS volume for each node 

ec2 oluşturacğı için özellikleri soruyor oldupu gibi bırakıyoruz.

Node group scaling configuration 
Desired size 
Set the desired number of nodes that the group should launch with initially. 
nodes 
Minimum size 
Set the minimum number of nodes that the group can scale in to. 
nodes 
Maximum size 
Set the maximum number of nodes that the group can scale out to. 
nodes 

autoscaling i 1-1-3 şeklinde güncelliyoruz. ilk etapta bir tane worker node oluşturacak ihtiyaç olunca 3 e kadar çıakracak.

next diyoruz.

Specify networking 
Node group network configuration 
These properties cannot be changed after the node group is created. 
Subnets Info 
Specify the subnets in Y)ur VPC where your nodes will run.To create a nc%' subnet, go to the corresponding page in the VPC console 
Select subnets 
subnet-oe092123d2a548fda X 
subnet-09df44bf4fea8478a X 
subnet-osf0006e30671473f X 
Configure SSH access to nodes Info 
Cancel 
c 
Previous 

subnetler bu şekilde kalacak. worker node a bağlanmak için configure ssh acces in işaretli olması gerkeir.

O Configure SSH access to nodes Info 
SSH key pair 
Select an SSA key pair to allow secure remote access to your nodes. To create a nmv SSE key pair, go to the corresponding page in the EC2 
console. 
first-key 
Allow SSH remote access from 
Configure the SSH client source IP ranges that can remotely access nodes. 
c 
C) Selected security groups 
Specify security groups 02Strict 
which source IPS can remotely access 
nodes. 
O All 
Do not 02Strict source IPS that can 
remotely access nodes. 

bu derste bunu kullanmayacağız. next diyoruz . review sayfasından create diyoruz.

desired 1 dediğimiz için 1 tane hazır node var:

• ec2-use 
ip-172-31-6 
o ec2-use 
rap-172-31-82-147# 
-43. ec2 
. internal 
kubectl 
STATUS 
Ready 
get no 
ROLES 
<none> 
AGE 
16m 
VERSION 
VI. 25. 7-eks-a59e1fØ 

node ların artırılıp azaltılması için cluster auto scaling servisi var. bunun için cluster auto scaling in kurulması lazım.

bunun için IAM servisinden policy oluşturuyoruz:

{

"Version": "2012-10-17",

"Statement": [

{

"Action": [

"autoscaling:DescribeAutoScalingGroups",

"autoscaling:DescribeAutoScalingInstances",

"autoscaling:DescribeLaunchConfigurations",

"autoscaling:DescribeTags",

"autoscaling:SetDesiredCapacity",

"autoscaling:TerminateInstanceInAutoScalingGroup",

"ec2:DescribeLaunchTemplateVersions"

],

"Resource": "\*",

"Effect": "Allow"

}

]

}

bu policy i yapıştırıp policy oluşturuyoruz.

gelen talebe göre node sayısının azaltlıp artırılması için bir policy bu.

Name* 
escription 
ClusterAutoscalerPolicy 
use alphanumeric and 
ClusterAutoscalerPolicy 
charactery Maximum 128 charactery 
Maximum 1000 characters. use alphanumeric and charactere 
Summary 
Q Filter 
Service 
Access level 
Allow (2 of 370 services) Show remaining 368 
EC2 
EC2 Auto Scaling 
Limited: List 
Full: Read Limited: List, Write 
Resource 
All resources 
All resources 
Request condition 
None 
None 

isim verip create diyoruz. bunu worker node un role üne ekliyoruz.

Permissions 
Trust relationships 
Tags 
Access Advisor 
Permissions policies 
(3) Info 
You can attach up to 10 managed policies. 
Q Filter policies by property or policy name and press enter 
Policy name ? 
AmazonEKSWorkerNodePolicy 
AmazonEC2ContainerRegistryReadO__ 
AmazonEKS CNI_Policy 
Revoke sessions 
Simulate 
Description 
Remove 
Add permissions A 
Attach policies 
Create inline policy 
Type 
AWS managed 
AWS managed 
AWS managed 
This policy allows Amazon EKS worker nodes to connect to Amazon EKS Clusters. 
Provides read-only access to Amazon EC2 Container Registry repositories 
This policy provides the Amazon VPC CNI Plugin (amazon-vpc-cni-k8s) the permi___ 

attach policies diyoruz.

policy i seçip

Q Filter policies by property or policy name and press enter 
"cluster X 
Clear filters 
Policy name t? 
ClusterAutoscalerPolicy 

add permissions diyoruz.

kubectl apply -f <https://raw.githubusercontent.com/kubernetes/autoscaler/master/cluster-autoscaler/cloudprovider/aws/examples/cluster-autoscaler-autodiscover.yaml>

bu komutla github ta duran kurulumu yapacaz. bu manifest dosyası ve objeler oluşturacak. bu sayede autoscaling yapılabilecek.

kubectl apply -f https://raw.githubusercontent.c 
• ec2-use 
ter-autoscaler/ cloud provider/ aws/ examples/ cluster-autos caler-autodiscover. yaml 
serviceaccount/cluster-autoscaler created 
clusterrole. rbac . authorization . k8s . io/cluster-autoscaler created 
role. rbac . authorization . k8s . io/cluster-autoscaler created 
clusterrolebinding. rbac . authorization . k8s . io/cluster-autoscaler created 
rolebinding. rbac . authorization . k8s . io/cluster-autoscaler created 
deployment . apps/cluster-autoscaler created 

kubectl get po -A 
• ec2-use 
rap-172-31-82-147# 
WIESPACE 
READY 
1/1 
1/1 
1/1 
1/1 
1/1 
STATUS 
Running 
Running 
Running 
Running 
Running 
RESTARTS 
kube-system 
kube-system 
kube-system 
kube-system 
kube-system 
aws -node-8k2gn 
cluster-autoscaler 
-69995cc689-pbfrj 
coredns -7975d6fb9b-14qdh 
coredns-7975d6fb9b-zdtn2 
kube-proxy -xvdkp 
AGE 
29m 
42s 
85m 
85m 
29m 

arka planda podları görüyoruz. bu işlemler sürekli yapılan işlemler değil. bir kere kuruluyor ve sonra cluster kullanılıyor. bir sonraki derste de eksctl ile yapacağız o daha kolay.

246* 
READY 
cluster-autoscaler 
1/1 
coredns 
ec2-us 
eru-172-31-81 246* 
k get deploy •n kube•system 
UP-TO-DATE AVAILABLE AGE 

deployment a baktığımızda autoscaling için bir deployment olduğunu görüyoruz. bizim buraya bir anotation eklememiz ve bununla ek bilgi vermemiz gerekiyor.

kubectl -n kube-system annotate deployment.apps/cluster-autoscaler cluster-autoscaler.kubernetes.io/safe-to-evict="false"

komutuyla deployment a safe-to-evict="false" bilgisini görüyoruz. cluster auto scaler a bu annotation a sahip bir pod varsa o pod u remove etme. neden remove eder, arttırdığı node ları tekrar azaltacak. bu anotation ile o pod ları remove etmeyecek.

bunu edit komuyula da yapabiliriz:

kubectl -n kube-system edit deployment.apps/cluster-autoscaler

şimdi edit komutuyla cluster ismi yazacaz ve container spec altına şu ifadeyi girecez:

This conunand will open the yaml file for your editting. Replace 
<CLUSTER NAPIE> value with your 
own cluster name, and add the following command option 
-skip-nodes-with -system-pods—false- 
to the command section under 
containers- 
under 
spec 
. Save and exit the file by pressing 
: wq 
. The changes will be applied. 

- -expander—least -waste 
-node-group-auto-discovery=asg : tag=k8s . io/ cluster-autoscaler/ enabled , k8s . io/ cluster-autoscaler/ emre 
-skip-nodes -with-system-pods—fals 

editörden ekledik. :wq ile kaydedip çıkıyoruz.

şimdi de cluster autoscaler kuracağız ancak versionu uyumlu olmalı.

7. Find an appropriate version of your cluster autoscaler in the [link] (https://github.com/ 
kubernetes/autoscaler/releases). The version number should start with version number of the 
cluster Kubernetes version . 
should find something like 
For example, if you have selected the Kubernetes version 1.23, you 
*1.23.e• 
8. Then, in the following conunand, set the Cluster Autoscaler image tag as that version you 
have found in the previous step. 
bash 
kubectl -n kube-system set image deployment.apps/cluster-autoscaler cluster-autosca1er=k8s .gcr. 
io/ autoscaling/ clu ster-autoscaler : < YOUR -VERSION-HERE > 

bizim kubernetes verisonumuz 1.25 autoscaler için hangi versionu kullanabileceğimiz docs ta mevcut. 1.25.1 i kullanacağız.

kubectl -n kube-system set image deployment.apps/cluster-autoscaler cluster-autoscaler=k8s.gcr.io/autoscaling/cluster-autoscaler:v1.25.1

deplymentın içine gidip image ini update etmiş oldu. bunu neden yaptık version önemli olduğu için yaptık.

kubernetes te yapılacak işlemler için bu şekilde farklı farklı komutları girip konfigürasyonlar yapmamız gerekiyor.

şimdi örnek uygulamamızı çalıştırıp autoscaling i senarize edecez.

kind: Service 
metadata : 
name: container-infi 
-svc 
namespace: my-namespace 
labels : 
app: container- n 
spec : 
type: LoadBa1ancer 
ports : 
protocol: TCP 
port: 3ØØØ 
nodeport: 30300 
targetPort: 80 
selector : 
app: container- n 

app imizde service tipi loadbalancer çünkü bugün provider desteğimiz var.

kubectl apply -f myapp.yml 
• ec2-use 
namespace/my-namespace created 
service/ container-info-svc created 
deployment . apps/ container-info-deploy created 

apply ile objeleri oluşturduk.

onaws . com 3Ø30Ø/TCP 
• ec2-use 
PORT(S) 
service/ container-info-svc 
13s 
kubectl -n 
TYPE 
AGE 
LoadBa1ancer 
my-namespace get svc, deploy,po 
CLUSTER-IP 
10.100. 138.245 
EXTERNAL-IP 
a47øaf2Ø855Ød4cc18f556øaff926e58-10Ø1313474. Us -east -1 
elb. amazonaws . com 3000: 30300/ TCP 1Ø5s 
deployment . apps / container -info-deploy 
READY 
3/3 
UP-TO-DATE 
3 
AVAILABLE 
3 
AGE 
1Ø5s 

burada svc ye baktığımızda external ip kısmında pending yazıyor olabilir, subnetlere tag girerek svc nin bağlanmasını sağlayacağız. (ben tag girmediğim halde elb nin dns ini aldı)

subnete gidip tag veriyoruz:

VPC Subnets > > Manage tags 
Manage tags for subnet-c41 ba589 
Tags 
Key 
Q kuberneteS.iO/duSter/petdinit-Rand 
Q kubemetæio/cluster/ 
Q Name 
Add new tag 
Value . 

cluster için olan kısmımda shared dediğimizde başka cluster lar tarafından kullanılmasına imkan tanır.

Subnets 
(1/6) Info 
Q Filter subnets 
search: vpc-033ab2eeb012575b1 
x 
Clear filters 
Actions 
Share subnet 
Manage tags 
Delete subnet 
Finn 
Create subnet 
rinp 

subneti seçip actions tan manage tags diyoruz.

Tags 
A tag is a label that you assign an AV.'S resource Each tag consists of a key and an optional value. You can use tags to search and filter 
pur resources or track your AWS costs. 
Q kubernetes.io/cluster/emre 
Add new tag 
You can add 49 more tags. 
X 
Value - optional 
Q shared 
x 
Remove 

bu şekilde tag leyip save ediyoruz.

bunu her 3 subnet için de yapıyoruz.

<https://repost.aws/knowledge-center/eks-vpc-subnet-discovery>

bu sayfada bununla ilgili bilgi mevcut.

load balancer kısmında da load balancer ı görebiliyoruz:

EQ > Load balancers 
Load balancers (1) 
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic. 
Q Filter by property or value 
Name 
a470af208550d4cc18f5 
560aff926e58 
DNS name 
0 a470af208550d4cc18f556... 
vpc ID 
033ab2eeb012575b1 
Actions 
Availability Zones 
6 Availability Zones 
v 
load balancer 
Type 
classic 
Apr 

Load balancer ın dns ini service portu (3000) ile girince uygulamayı görüoruz:

CLARUSWAY 
WAY TO REINVENT YOURSELF 
Project - Docker Container Info Demo 
Container Info v1.0 
Host: container-info-deploy-5bS9d4f5f-sqSvf 
Running O S :linux 
Network Information: 
DNS 
This app is developed by DevOps Team. 

şimdi deployment taki replicas sayısını 3 ten 30 a çıkaralım.

apply edelim.

o ec2-use 
container-info-deploy 
container-info-deploy 
container-info-deploy 
container-info-deploy 
container-info-deploy 
container-info-deploy 
container-info-deploy 
container-info-deploy 
container-info-deploy 
kubectl 
-2g9cp 
-2q64q 
-5x5cb 
-6t9tk 
-7chv6 
-7jxsw 
-9t5dw 
-cr76h 
-dns7d 
-gjdnm 
-hdjcm 
-hxdwz 
get po 
READY 
1/1 
1/1 
1/1 
1/1 
1/1 
1/1 
1/1 
-n my-namespace -w 
container-info-deploy-5b89d4f5f 
container-info-deploy-5b89d4f5f-6g5ng 
container-info-deploy-5b89d4f5f 
container-info-deploy-5b89d4f5f-94wtp 
container-info-deploy-5b89d4f5f-btjrj 
container-info-deploy-5b89d4f5f-dfhtj 
container-info-deploy-5b89d4f5f 
container-info-deploy-5b89d4f5f-hj1pj 
container-info-deploy-5b89d4f5f-j18g6 
-5b89d4f5f 
-5b89d4f5f 
-5b89d4f5f 
-5b89d4f5f 
-5b89d4f5f 
-5b89d4f5f 
-5b89d4f5f 
-5b89d4f5f 
-5b89d4f5f 
STATUS 
Pending 
Running 
Pending 
Pending 
Pending 
Running 
Pending 
Running 
Running 
Pending 
Pending 
Pending 
Pending 
Pending 
Running 
Pending 
Running 
Running 
RESTARTS 
AGE 
42s 
42s 
42s 
42s 
42s 
42s 
42s 
19m 
42s 
42s 
42s 
42s 
42s 
42s 
42s 
42s 
42s 
42s 

pod lara baktığımızda pendingler var çünkü pod sayısı node a fazla geldi, node lar oluştukça podlar running duruma geçecek.

-31-81-246* 
ip-1n-g1-12-1ø 
eco. internal 
ip-172-31-31-28. 
ip-1/2.g1-gg-g 
STATUS 
Ready 
Ready 
ROLES 
AGE 
2m21s 
2,n22s 
VERSIcn 
VI _ 25 _ 7-eks 
-eks 
vi.25.7 
Iner container 
a59e1fØ 
-a59e1fø 

node ların sayısını max capacitye çıkardı.

podları da 30 a çıkaracak.

şimdi replicas ı 3 e düşürüp apply diyebiliriz ancak arttığı kadar hızlı düşmez.

myapp.yml ı siliyoruz. node group u siliyoruz. node group u n silinmesini bekliyoruz, sonra cluster ı siliyoruz. en sonda da bağlandığımız makineyi terminate edebiliriz ya da stop edebiliriz.

EKS de resource larda pod ları görebiliyoruz

