

EKS ENVIRONMENT

CLUSTER

```
> eksctl create cluster \
  --name idc-ex1-q2 \
  --version 1.16 \
  --without-nodegroup
[i] eksctl version 0.20.0
[i] using region us-east-2
[i] setting availability zones to [us-east-2a us-east-2c us-east-2b]
[i] subnets for us-east-2a - public:192.168.0.0/19 private:192.168.96.0/19
[i] subnets for us-east-2c - public:192.168.32.0/19 private:192.168.128.0/19
[i] subnets for us-east-2b - public:192.168.64.0/19 private:192.168.160.0/19
[i] using Kubernetes version 1.16
[i] creating EKS cluster "idc-ex1-q2" in "us-east-2" region with
[i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-2 --cluster=idc-ex1-q2'
[i] CloudWatch logging will not be enabled for cluster "idc-ex1-q2" in "us-east-2"
[i] you can enable it with 'eksctl utils update-cluster-logging --region=us-east-2 --cluster=idc-ex1-q2'
[i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "idc-ex1-q2" in "us-east-2"
[i] 2 sequential tasks: { create cluster control plane "idc-ex1-q2", no tasks }
[i] building cluster stack "eksctl-idc-ex1-q2-cluster"
[i] deploying stack "eksctl-idc-ex1-q2-cluster"
[i] waiting for the control plane availability...
[✓] saved kubeconfig as "/Users/grabin/.kube/config"
[i] no tasks
[✓] all EKS cluster resources for "idc-ex1-q2" have been created
[i] kubectl command should work with "/Users/grabin/.kube/config", try 'kubectl get nodes'
```

```
> kubectl get svc
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes    ClusterIP     10.100.0.1    <none>         443/TCP    9m29s
```

NODE GROUP

```

> eksctl create nodegroup \
--cluster idc-ex1-q2 \
--version 1.16 \
--name idc-ex1-q1-node-group \
--node-type t2.small \
--nodes 2 \
--nodes-min 1 \
--nodes-max 4 \
--node-ami auto
[i] eksctl version 0.20.0
[i] using region us-east-2
[i] nodegroup "idc-ex1-q1-node-group" will use "ami-0edc51bc2f03c9dc2" [AmazonLinux2/1.16]
[i] 1 nodegroup (idc-ex1-q1-node-group) was included (based on the include/exclude rules)
[i] will create a CloudFormation stack for each of 1 nodegroups in cluster "idc-ex1-q2"
[i] 2 sequential tasks: { fix cluster compatibility, 1 task: { 1 task: { create nodegroup "idc-ex1-q1-node-group" } } }
[i] checking cluster stack for missing resources
[i] cluster stack is missing resources for Fargate
[i] adding missing resources to cluster stack
[i] re-building cluster stack "eksctl-idc-ex1-q2-cluster"
[✓] all resources in cluster stack "eksctl-idc-ex1-q2-cluster" are up-to-date
[i] building nodegroup stack "eksctl-idc-ex1-q2-nodegroup-idc-ex1-q1-node-group"
[i] deploying stack "eksctl-idc-ex1-q2-nodegroup-idc-ex1-q1-node-group"
[i] no tasks
[i] adding identity "arn:aws:iam::089769435595:role/eksctl-idc-ex1-q2-nodegroup-idc-ex1-q1-nodegroup-1D6R8RZK4LX4J" to auth ConfigMap
[i] nodegroup "idc-ex1-q1-node-group" has 1 node(s)
[i] node "ip-192-168-6-130.us-east-2.compute.internal" is not ready
[i] waiting for at least 1 node(s) to become ready in "idc-ex1-q1-node-group"
[i] nodegroup "idc-ex1-q1-node-group" has 2 node(s)
[i] node "ip-192-168-6-130.us-east-2.compute.internal" is not ready
[i] node "ip-192-168-69-182.us-east-2.compute.internal" is ready
[✓] created 1 nodegroup(s) in cluster "idc-ex1-q2"
[✓] created 0 managed nodegroup(s) in cluster "idc-ex1-q2"
[i] checking security group configuration for all nodegroups
[i] all nodegroups have up-to-date configuration

```

```

> kubectl get nodes

```

NAME	STATUS	ROLES	AGE	VERSION
ip-192-168-6-130.us-east-2.compute.internal	Ready	<none>	97s	v1.16.8-eks-e16311
ip-192-168-69-182.us-east-2.compute.internal	Ready	<none>	97s	v1.16.8-eks-e16311

DEPLOYMENT

```

> kubectl apply -f ex1_q2_deployment.yaml
deployment.apps/ex1-q2 created

```

```

> kubectl get deployments.apps

```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
ex1-q2	3/3	3	3	30s

PODS -3 REPLICAS

```
> kubectl get pods -l 'app=ex1-q2' -o wide | awk {'print $1 " " $3 " " $6'} | column -t
```

NAME	STATUS	IP
ex1-q2-647c796685-7cwwc	Running	192.168.25.72
ex1-q2-647c796685-dm9vt	Running	192.168.89.7
ex1-q2-647c796685-hdqq9	Running	192.168.26.9

SERVICE - LOAD BALANCER

```
> kubectl describe svc
```

Name: ex1-q2-lb
Namespace: default
Labels: app=ex1-q2
Annotations: <none>
Selector: app=ex1-q2
Type: ClusterIP
IP: 10.100.68.29
Port: <unset> 80/TCP
TargetPort: 80/TCP
Endpoints: 192.168.17.78:80,192.168.69.196:80,192.168.83.137:80
Session Affinity: None
Events: <none>

WEBSITE

a7cd276dc7ad24e8e807e24687f21ef3-661517096.us-east-2.elb.amazonaws.com

[Pipelines - content](#) [Content-project](#) [Demisto](#) [IDC](#) [google](#) [No internet](#) [Pluralsight](#) [Repos](#)

Upload photo

Key to upload:

4da56c6f-a110-42d4-bbeb-751b6cb6c8a4.png

x-amz-storage-class:

STANDARD_IA

X-Amz-Credential:

AKIAIL3YY5HF7C6FWO2Q/20200601/us-east-2/s3/aws4_request

X-Amz-Algorithm:

AWS4-HMAC-SHA256

X-Amz-Signature:

44e983e1aae650afd2a87e14909c7b30371d47ebcfc0bb5d02ab7ee565f8921

X-Amz-Date:

20200601T080646Z

X-Amz-Date:

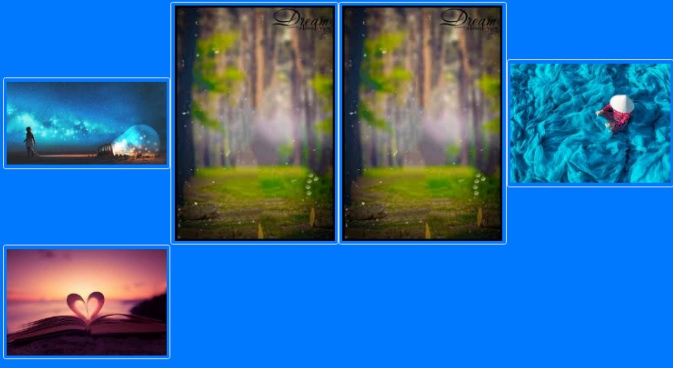
eyJleHBpcmF0aW9uIjoiMjAyMC0wNi0wMVQwOTowNjo0Ni0iLCJlb25kaXRpb25zIjpbeyJrZXkiOiI0ZGE1NmM2Zi1hMTewLTQyZDQTYmJiYiO3NTI






File to upload:

[Choose File](#) [No file chosen](#)

[Upload to Amazon S3](#)

Photos in S3 bucket



<input type="checkbox"/>	Name ▾	Last modified ▾	Size ▾	Storage class ▾
<input type="checkbox"/>	 1.jpeg	Jun 1, 2020 10:59:14 AM GMT+0300	7.4 KB	Standard
<input type="checkbox"/>	 7d359ee1-9652-4174-bf10-294818cd693a.png	Jun 1, 2020 11:04:50 AM GMT+0300	6.2 KB	Standard-IA
<input type="checkbox"/>	 a45f57fb-d725-4b61-99c9-44aaab5f9008.png	Jun 1, 2020 11:06:46 AM GMT+0300	6.2 KB	Standard-IA
<input type="checkbox"/>	 dfb46939-841d-4c62-a055-03e34fe9f9bb.png	Jun 1, 2020 11:04:39 AM GMT+0300	10.7 KB	Standard-IA
<input type="checkbox"/>	 fb3e0783-3bba-4359-b77a-b19061dff866.png	Jun 1, 2020 11:04:26 AM GMT+0300	4.5 KB	Standard-IA