eksctl EKS cluster CLI

START HERE

`eksctl` **`create cluster -f`** *`<file>`*
Create from config

`eksctl` **`get cluster`**
List clusters

`eksctl` **`get nodegroup --cluster`** *`<name>`*
List node groups

`eksctl` **`scale nodegroup --cluster`** *`<name>`* **`--name`** *`<ng>`* **`--nodes`** *`<n>`*
Resize nodes

`eksctl` **`utils write-kubeconfig --cluster`** *`<name>`*
Connect kubectl

`eksctl` **`upgrade cluster -f`** *`<file>`* **`--approve`**
Upgrade control plane

`eksctl` **`create cluster --dry-run`**
Print generated config

`eksctl` **`delete cluster -f`** *`<file>`* **`--wait`**
Tear it down

AWS account + region default region from AWS CLI config
CloudFormation stacks eksctl-<cluster>-cluster, -nodegroup-<ng>
EKS control plane + VPC managed by AWS
Node groups managed (default) or self-managed

What eksctl creates

CONCEPTS

eksctl turns each cluster into CloudFormation stacks; a cluster and each node group are separate stacks you can inspect in AWS.

Pass a ClusterConfig file (-f) instead of long flag lists: it is reviewable, repeatable and works for create, upgrade and delete.

CREATE CLUSTER

`eksctl` **`create cluster`**
Default cluster: 2 managed m5.large nodes

`eksctl` **`create cluster -f`** *`<file>`*
Create from a ClusterConfig file

`--name` *`<name>`* **`--region`** *`<region>`*
Name and region of the cluster

`--version` *`<k8s-version>`*
Kubernetes version of the control plane

`--dry-run`
Print the ClusterConfig, create nothing
Options that cannot be written to a config file are rejected together with --dry-run.

`--node-private-networking`
Put nodes in private subnets only

CLUSTERCONFIG FILE

Minimal cluster.yaml

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: <name>
  region: <region>
managedNodeGroups:
  - name: ng-1
    instanceType: m5.large
    desiredCapacity: 3
    minSize: 1
    maxSize: 5
```

`managedNodeGroups: / nodeGroups:`
Managed vs self-managed groups

Cluster and node group names: letters, digits and hyphens, starting with a letter, at most 128 characters.

CLUSTERCONFIG FILE (cont.)

Use eksctl create cluster --dry-run to generate a full config with all defaults filled in, then edit it.

INSPECT CLUSTERS

`eksctl` **`get cluster`**
List clusters in the region

`eksctl` **`version`**
Show the eksctl version

`eksctl` **`utils nodegroup-health --cluster`** *`<name>`* **`--name`** *`<ng>`*
Health issues of a managed group

NODE GROUPS

`eksctl` **`get nodegroup --cluster`** *`<name>`*
List node groups

`eksctl` **`create nodegroup --cluster`** *`<name>`* **`--name`** *`<ng>`*
Add a node group

`eksctl` **`scale nodegroup --cluster`** *`<name>`* **`--name`** *`<ng>`* **`--nodes`** *`<n>`*
Change desired size

`eksctl` **`upgrade nodegroup --cluster`** *`<name>`* **`--name`** *`<ng>`*
Roll a managed group to the cluster version

`eksctl` **`delete nodegroup --cluster`** *`<name>`* **`--name`** *`<ng>`* **`--approve`**
Delete a node group
Without --approve the command only shows a plan.

`--include` *`<glob>`* **`/ --exclude`** *`<glob>`*
Select groups when using a config file

KUBECONFIG

`eksctl` **`utils write-kubeconfig --cluster`** *`<name>`*
Add the cluster to your kubeconfig

create cluster already adds the context to $KUBECONFIG or ~/.kube/config; write-kubeconfig is for a fresh machine or teammate.

UPGRADES

`eksctl` **`upgrade cluster --name`** *`<name>`*
Preview the control-plane upgrade

`--version` *`<k8s-version>`* **`--approve`**
Apply the upgrade, one minor step
Only the current version or one minor version higher is allowed.

`eksctl` **`upgrade cluster -f`** *`<file>`* **`--approve`**
Upgrade using metadata.version

`eksctl` **`utils update-kube-proxy --cluster`** *`<name>`* **`--approve`**
Update the kube-proxy add-on

Upgrade order: control plane, then node groups, then the default add-ons.

IAM AND ADD-ONS

`eksctl` **`utils associate-iam-oidc-provider --cluster`** *`<name>`* **`--approve`**
Enable the OIDC provider for IRSA

`eksctl` **`create iamserviceaccount --cluster`** *`<name>`* **`--name`** *`<sa>`* **`--namespace`** *`<ns>`* **`--attach-policy-arn`** *`<arn>`*
Bind an IAM role to a service account

`eksctl` **`delete iamserviceaccount --cluster`** *`<name>`* **`--name`** *`<sa>`* **`--namespace`** *`<ns>`*
Remove the role and service account
Also deletes Kubernetes service accounts not created by eksctl.

`eksctl` **`get addon --cluster`** *`<name>`*
List installed add-ons

`eksctl` **`create addon --cluster`** *`<name>`* **`--name`** *`<addon>`*
Install an EKS add-on

`eksctl` **`update addon --cluster`** *`<name>`* **`--name`** *`<addon>`* **`--version`** *`<ver>`*
Change an add-on version

IAM AND ADD-ONS (cont.)

`eksctl` **`utils describe-addon-versions --kubernetes-version`** *`<ver>`*
Find available add-on versions

DELETE CLUSTER

`eksctl` **`delete cluster --name`** *`<name>`* **`--wait`**
Delete and report errors
Without --wait only the CloudFormation delete is issued; failures stay hidden.

`eksctl` **`delete cluster -f`** *`<file>`* **`--wait`**
Delete the cluster from its config

`--disable-nodegroup-eviction`
Ignore PodDisruptionBudgets while draining

Deletion is destructive: load balancers, volumes or ENIs created by workloads can block it. Remove them first, or delete the leftover CloudFormation stacks by hand.

RECIPES

Preview, create, connect

```
eksctl create cluster --name
<name> --region <region> \
  --nodes 3 --dry-run >
cluster.yaml
eksctl create cluster -f
cluster.yaml
kubectl get nodes
```

Upgrade a cluster

```
eksctl upgrade cluster -f
cluster.yaml --approve
eksctl upgrade nodegroup --
cluster <name> --name <ng>
eksctl utils update-kube-proxy
--cluster <name> --approve
```

IRSA for an app

```
eksctl utils associate-iam-
oidc-provider --cluster <name>
--approve
eksctl create iamserviceaccount
--cluster <name> \
  --name <sa> --namespace <ns>
\
  --attach-policy-arn
arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess \
  --approve
```