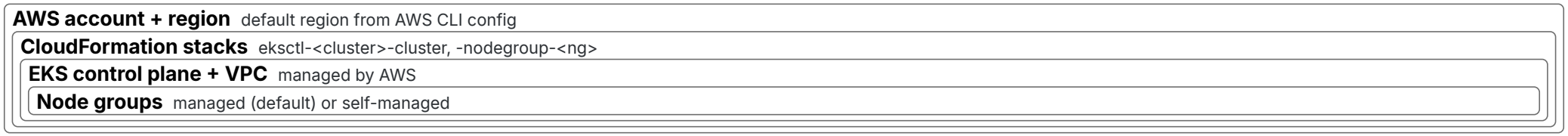
eksctl EKS cluster CLI



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

CREATE CLUSTER (cont.)

--dry-run
Print the ClusterConfig, create nothing

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

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

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

KUBECONFIG

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

UPGRADES

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

--version *<k8s-version>* **--approve**
Apply the upgrade, one minor step

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

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

IAM AND ADD-ONS (cont.)

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

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

DELETE CLUSTER

eksctl delete cluster --name *<name>* **--wait**
Delete and report errors

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

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.