USAGE
- Create VPC from template IPV4 with Cloudformation

- Create role to access the cluster with I AM: eks-cluster-role
- public is to make sure it works locally with your kubeconfig
- private

Use kubeconfig

- aws configure list
- aws eks update-kubeconfig --name eks-cluster-test (create kubeconfig)

EKS can use EC2 or Fargate

- Node groups are the e2 instances you need to manage
- Fargate ec2 instances managed
- EC2 role for node groups with 3 policies (eks_cni_policy, eksworkernode, EC2 containerregistry readonly)
- Create the nodegroup with the role we just created, configure remote access
- Node group should have I am role for autoscaling to work a specific policy
- To have the autoscaling to work it needs to have the following tags:

k8s.io/cluster-autoscaler/eks-cluster-test	owned	Yes
k8s.io/cluster-autoscaler/enabled	true	Yes

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

- kubectl get deployment -n kube-system cluster-autoscaler
- the yaml needs to be updated with some additional values

Fargate

- Fargate is a managed service, limited stateful.
- We can add fargate as an addition to the node group
- Create a role in IAM > fargate pod
-

Jenkins Kubernetes

——————

Install kubectl in docker as root: docker exec -u 0 -it f75be156f36b
curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s

```
https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl; chmod +x ./kubectl; mv ./kubectl /usr/local/bin/kubectl
```

aws-authenticator to talk to the cluster

```
curl -Lo aws-iam-authenticator curl -Lo aws-iam-authenticator https://github.com/kubernetes-sigs/aws-iam-authenticator/releases/download/v0.5.9/aws-iam-authenticator_0.5.9_linux_amd64
 chmod +x ./aws-iam-authenticator
mv ./aws-iam-authenticator /usr/local/bin (so jenkins can use it)
```

- authenticator needs the endpoint url from the cluster in the yaml file to be able to communicate
- certificate authoritay data is in the kubeconfig
- the config needs be available in the jenkinscontainer in the default kube config location as non root. cd ~ to go into the home directory of jenkinscontainer. This is the home of the Jenkinsuser
- copy the created kubeconfig file docker cp config f75be156f36b:/var/jenkins_home/.kube/
- create jenkins user with limited access: cat ./aws/credentials
-

Best practices

- Create jenkins user with it's own SSH and limited access
-

Use dynamic environment variables with envsubst, this is not standard available. Enter as root in the Jenkins container to make it available.