

- Install `aws` and `eksctl` command line tools:

```
brew install aws eksctl
```

- Create an EKS cluster using the following command:

```
eksctl create cluster \
--name my-cluster \
--version auto \
--region eu-west-1 \
--nodegroup-name my-nodes \
--node-type t3.large \
--nodes 1 \
--nodes-min 1 \
--nodes-max 2 \
--managed \
--with-oidc
```

- Create a `trust.json` policy that contains the following (replace `accountID` and `clusterID` with the respective values: your account ID, and the clusterID from the EKS page.)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::accountID:oidc-
provider/oidc.eks.eu-west-1.amazonaws.com/id/clusterID"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.eks.eu-west-1.amazonaws.com/id/clusterID:sub":
            "system:serviceaccount:flux-system:source-controller",

```

```
        "oidc.eks.eu-west-1.amazonaws.com/id/clusterID:aud":  
        "sts.amazonaws.com"  
    }  
  }  
}  
]  
}
```

- Create the role, attach the trust policy and the ECR readonly policy:

```
aws iam create-role --role-name FluxCDECR --assume-role-policy-  
document file://trust.json  
aws iam attach-role-policy --role-name FluxCDECR --policy-arn  
arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
```

- Go to the IAM console and make note of the role's ARN.
- Create a directory for the EKS cluster:

```
cd myfluxrepo/clusters  
mkdir eks
```

- Make sure you are on the `main` branch and pull the latest changes:

```
git checkout main  
git pull
```

- Boot Flux CD with the new cluster:

```
# export the GitLab token  
flux bootstrap gitlab --owner=abohmeed --repository=fluxrepotest --  
branch=main --path=clusters/eks --token-auth --personal
```

- Pull the changes that Flux CD did to the cluster:

```
git pull
```

- In the `flux-system` directory, modify the Kustomization to be as follows:

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
resources:
- gotk-components.yaml
- gotk-sync.yaml
patches:
- patch: |
    apiVersion: v1
    kind: ServiceAccount
    metadata:
      name: source-controller
      annotations:
        eks.amazonaws.com/role-arn:
"arn:aws:iam::accountID:role/FluxCDECR"
    target:
      kind: ServiceAccount
      name: source-controller
```

- Commit and push:

```
git add -A
git commit -m "Adds the AWS Service Account to Flux CD"
git push
```

- Reconcile Flux CD:

```
flux reconcile kustomization flux-system --with-source
```

- Create a new Helm package for Apache:

```
cd ../../../../charts/apache
helm package .
```

- Go to the ECR page and create a new repository called `apache`.
- Login to ECR:

```
aws ecr get-login-password --region eu-west-1 | docker login --
username AWS --password-stdin accountID.dkr.ecr.eu-west-
1.amazonaws.com
```

- Push the Helm package to ECR (notice how we are NOT appending the repository name to the URL):

```
helm push ./apache-0.1.0.tgz oci://accountID.dkr.ecr.eu-west-
1.amazonaws.com
```

- Create a new branch in our Flux repository:

```
git checkout -b "ecr"
```

- Create a new Helm repository file called `ecr.yaml` with the following contents:

```
apiVersion: source.toolkit.fluxcd.io/v1beta2
kind: HelmRepository
metadata:
  name: ecr
  namespace: default
spec:
  type: oci
  interval: 5m0s
  url: oci://accountID.dkr.ecr.eu-west-1.amazonaws.com
  provider: aws
```

- Create a new Helm Release that install Apache and called it `apache-helm-release` and add the following (copy it from the other cluster and make the appropriate changes):

```
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: apache
  namespace: default
spec:
  interval: 5m
  chart:
    spec:
      chart: apache
      version: '0.1.0'
      sourceRef:
        kind: HelmRepository
        name: ecr
        namespace: default
      interval: 1m
```

- Commit and push the changes:

```
git add -A
git commit -m "Adds the ECR repository and the Apache Helm release"
git push --set-upstream origin ecr
```

- Create and merge the MR.
- Reconcile Flux CD.
- Check the OCI repository, the Helm release, and the pods.

```
kubectl get helmrepository
kubectl get helmrelease
kubectl get pods
```

