Mosip Environment Deployment on AWS using EKS(Elastic k8s service)

Overview

- Mosip modules are deployed in the form of microservices in the Kubernetes cluster.
- Wireguard is used as a trust network extension to access the admin, control, and observation pane.
- It's also used for on-the-field registration.
- Mosip uses AWS load balancers for
  - SSL termination
  - Reverse Proxy
  - CDN/Cache management
  - Load balancing
- Kubernetes cluster is administered using the Rancher and EKS.
- In V3 we have two Kubernetes clusters:
  - Observation Cluster - This cluster is part of the observation plane and it helps in administrative tasks. By design, this is kept independent of the actual cluster as a good security practice and to ensure clear segregation of roles and responsibilities. As a best practice, this cluster or its services should be internal and should never be exposed to the external world.
    - Rancher is used for managing the Mosip cluster.
    - Keycloak in this cluster is used for cluster user access management.
    - Its recommended to configure log monitoring and network monitoring in this cluster.
  - Mosip Cluster - This cluster runs all the Mosip components and specific third-party components to secure the cluster, API, and Data.
    - MOSIP External Components
    - Mosip Services

Deployment Repos

- k8s-infra : contains scripts to install and configure Kubernetes cluster with required monitoring, logging and alerting tools.
- mosip-infra : contains deployment scripts to run charts in a defined sequence.
- mosip-config : contains all the configuration files required by the Mosip modules.
- mosip-helm : contains packaged helm charts for all the Mosip modules.

AWS EKS Cluster Setup for Rancher

Prerequisites

- AWS account and credentials with permissions to create an EKS cluster.
- Make sure the AWS user should have all the permission and roles added to create the EKS cluster.
  Amazon EKS cluster IAM role - Amazon EKS
- AWS credentials in `~/.aws/` folder as given here.
- Wireguard setup. (You can refer below section of `Wireguard Setup` to setup wireguard)
- Copy of `~/.kube/config` file with another name. *(IMPORTANT. As in this process,* your existing `~/.kube/config` file will be overridden).
- eksctl utility.
- kubectl utility installed.
- Key `.pem` file from AWS console in `~/.ssh/` folder. (Generate a new one if you do not have this key file).

- aws-iam-authenticator installed.
- istioctl utility.
- Ansible : version > 2.12.4
- rke : version: 1.3.10
- helm client version 3.8.2 and add the below repo's as well to local

  -
    ```
    helm repo add bitnami https://charts.bitnami.com/bitnami
    helm repo add mosip https://mosip.github.io/mosip-helm

    ```
- Create a directory as Mosip in your local and clone `k8's infra` and `mosip-infra` repo with `tag : 1.2.0.1-B2` (tag depends on the latest release) inside the Mosip directory.

  -
    ```
    git clone https://github.com/mosip/k8s-infra -b v1.2.0.1-B2
    git clone https://github.com/mosip/mosip-infra -b v1.2.0.1-B2

    ```
- Set the below-mentioned variables in bashrc

  -
    ```
    export MOSIP_ROOT=<location of mosip directory>
    export K8_ROOT=$MOSIP_ROOT/k8s-infra
    export INFRA_ROOT=$MOSIP_ROOT/mosip-infra

    ```
  - `source .bashrc`
    Note:
    Above mentioned environment variables will be used throughout the installation to move from one directory to another to install scripts.
- Hardware requirements:
  - All the nodes must be in the same VPC. And the below table is just for your reference. Here you are not creating any below-mentioned nodes manually.

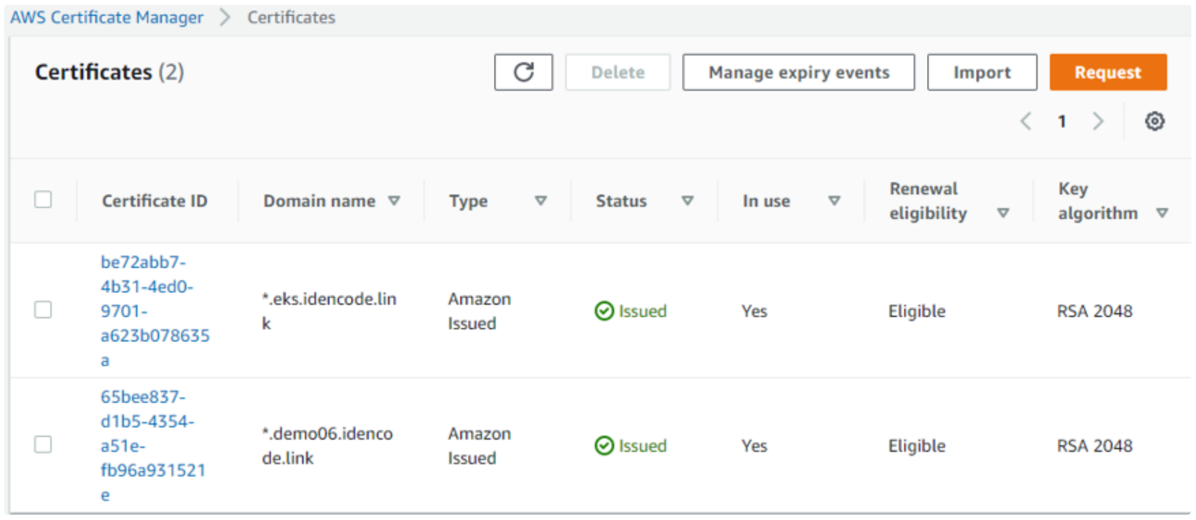
Purpose	vCPUs	RAM	Storage	AWS Instance Type	Number of Nodes
Rancher Cluster nodes	2	8 GB	32 GB	t3a.large	2
Mosip Cluster nodes	6	16 GB	64 GB	t3a.2xlarge	6
Wireguard bastion host	2	4 GB	8 GB	t2.medium	1

Note:

- Only proceed to DNS mapping after the ingress gateways are installed and the Nginx reverse proxy is set up.

Certificate requirements

- Depending upon the above hostnames, at least one wildcard SSL certificate will be required. For example; `*.org.net`.
- More SSL certificates will be required, for every new level of the hierarchy. For example; `*.sandbox1.org.net`.
- Make sure the SSL certificate hostname will be provided with respect to your domain name of rancher and mosip
- you can create SSL certs by ACM for EKS deployments. Find the below image for the same and this needs to be mapped in the route53 in AWS. Make sure the name and values should be taken from the certificate only for routing.


AWS Certificate Manager > Certificates

Certificates (2)

	Certificate ID	Domain name	Type	Status	In use	Renewal eligibility	Key algorithm
☐	be72abb7-4b31-4ed0-9701-a623b078635a	*.eks.idencode.link	Amazon Issued	✓ Issued	Yes	Eligible	RSA 2048
☐	65bee837-d1b5-4354-a51e-fb96a931521e	*.demo06.idencode.link	Amazon Issued	✓ Issued	Yes	Eligible	RSA 2048

Wireguard setup

A Wireguard bastion host (Wireguard server) provides a secure private channel to access the MOSIP cluster. The host restricts public access and enables access to only those clients who have their public key listed in the Wireguard server. Wireguard listens on UDP port51820.

Setup Wirguard VM and Wireguard bastion server

- Create a Wireguard server VM in the AWS console with above mentioned Hardware and Network requirements.
- Edit the security group and add the following inbound rules in the AWS console
  - type ‘custom TCP', port range ‘51820’, and source '0.0.0.0/0’
  - type ‘custom UDP', port range ‘51820’, and source '0.0.0.0/0’
- Open ports and Install docker on Wireguard VM Please make sure you will not find this ports.yaml and docker.yaml script in AWS installation so go to on-prem path and install on only your wireguard machine.
- execute `ports.yml` to enable ports on VM level using ufw:
  - `ansible-playbook -i hosts.ini ports.yaml`
    Note: these ports only needed to be opened for sharing packets over UDP. Take necasary measure on firewall level so that the Wireguard server can be rachable on 51820/udp.
- execute `docker.yml` to install docker and add user to docker group:
  - `ansible-playbook -i hosts.ini docker.yaml`
- Setup Wireguard server
  - SSH to wire guard VM
  - Create a directory for storing Wireguard config files.
    `mkdir -p wireguard/config`
  - Install and start wireguard server using docker as given below:

```
sudo docker run -d \
   --name=wireguard \
   --cap-add=NET_ADMIN \
   --cap-add=SYS_MODULE \
   -e PUID=1000 \
   -e PGID=1000 \
   -e TZ=Asia/Calcutta\
   -e PEERS=30 \
   -p 51820:51820/udp \
   -v /home/ubuntu/wireguard/config:/config \
   -v /lib/modules:/lib/modules \
   --sysctl="net.ipv4.conf.all.src_valid_mark=1" \
   --restart unless-stopped \
   ghcr.io/linuxserver/wireguard

```

Note:
* Increase the no of peers above in case needed more than 30 Wireguard client confs. (`-e PEERS=30`)
* Change the directory to be mounted to Wireguard docker in case needed.
All your Wireguard confs will be generated in the mounted directory. (`-v /home/ubuntu/wireguard/config:/config`)

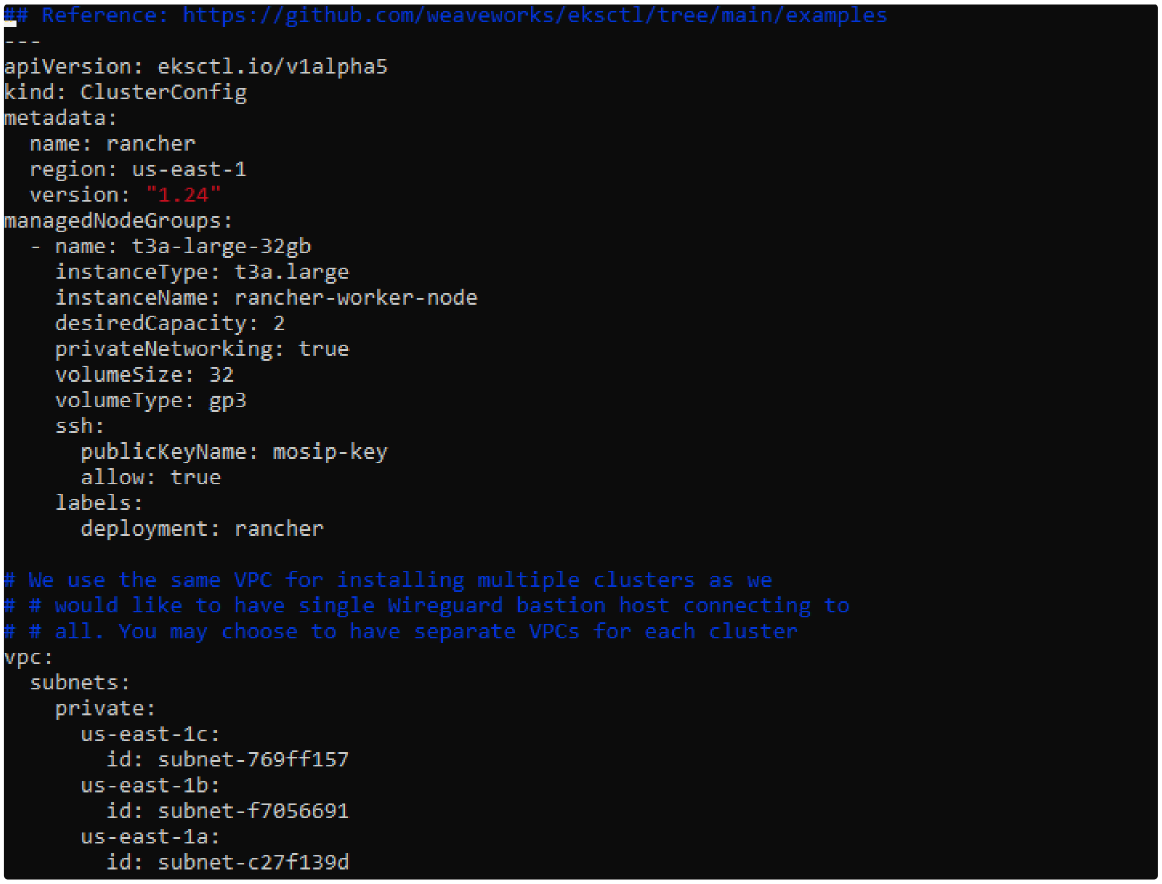
Setup Wireguard Client on your PC

- Install Wireguard client in your PC.
- Assign wireguard.conf:
  - SSH to the wireguard server VM.
  - `cd /home/ubuntu/wireguard/config`
  - assign one of the PR for yourself and use the same from the PC to connect to the server.
    - create `assigned.txt` file to assign and to keep track of peer files allocated and update every time some peer is allocated to someone.

      -
        ```
        peer1 :   peername
        peer2 :   xyz

        ```
    - Use `ls` cmd to see the list of peers.
    - get inside your selected peer directory, and add mentioned changes in peer.conf:
      - `cd peer1`
      - `nano peer1.conf`
        - Delete the DNS IP.
        - Update the allowed IP's to subnets CIDR ip . e.g. 10.10.20.0/23
      - Share the updated `peer.conf` with the respective peer to connect to wireguard server from Personal PC.
- Add the peer in your wireguard and do activate it.
- Once Connected to wireguard you should be now able to login using private ip’s.

Rancher K8s Cluster setup and configuration

- Make sure all the required tools mentioned above should be installed.
- Setup the aws credentials as given in the prerequisites above.
- Setup rancher cluster,
  - `cd $K8_ROOT/rancher/aws`
  - Copy `rancher.cluster.config.sample ` to `rancher.cluster.config`.
  - Review and update the parameters of `rancher.cluster.config` carefully.
    - Update the metadata keys like `region` to the region on aws and `version: "1.24"`
    - Update the `instanceName` , `instanceType` , `desiredcapacity` , `volumeSize` , `volumeType` and the `publicKeyName`
    - Update the VPC subnets from the AWS console either using default VPC subnets or creating new VPC.
    - Below is the ex for config file to create the rancher cluster.



```
## Reference: https://github.com/weaveworks/eksctl/tree/main/examples
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: rancher
  region: us-east-1
  version: "1.24"
managedNodeGroups:
  - name: t3a-large-32gb
    instanceType: t3a.large
    instanceName: rancher-worker-node
    desiredCapacity: 2
    privateNetworking: true
    volumeSize: 32
    volumeType: gp3
    ssh:
      publicKeyName: mosip-key
      allow: true
    labels:
      deployment: rancher

# We use the same VPC for installing multiple clusters as we
# # would like to have single Wireguard bastion host connecting to
# # all. You may choose to have separate VPCs for each cluster
vpc:
  subnets:
    private:
      us-east-1c:
        id: subnet-769ff157
      us-east-1b:
        id: subnet-f7056691
      us-east-1a:
        id: subnet-c27f139d
```

- Install with the below command.
  - `eksctl create cluster -f rancher.cluster.config`
- It takes around 30 minutes to create (or delete a cluster).
- The last line of output is similar to the following example line.
  - `[✓] EKS cluster "my-cluster" in "region-code" region is ready`
- In case cluster failed due to some error delete the cluster with below command and recreate it.
  - `eksctl delete cluster --region=(your region) --name=rancher`
- The config file for the new cluster will be created on `~/.kube/config`
- After creating cluster make a backup copy of `config` with a suitable name in `~/.kube/` folder, eg. `rancher_config` because if you create cluster again using `eksctl` it will override existing `~/.kube/config`. Set file permission to `chmod 400 ~/.kube/rancher_config` to avoid any accidental changes or deletions.
- Test cluster access:
  - `kubect get nodes`
  - The above command will result in details of the nodes of the rancher cluster and also you can verify from AWS-EKS.

Rancher K8s Cluster Ingress

Once the rancher cluster is ready we need ingress and storage class to be set for other applications to be installed.

- Nginx Ingress Controller: used for ingress in rancher cluster.

  -
    ```
    cd $K8_ROOT/rancher/aws
    helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
    helm repo update
    helm install \
      ingress-nginx ingress-nginx/ingress-nginx \
      --namespace ingress-nginx \
      --create-namespace  \
    ```

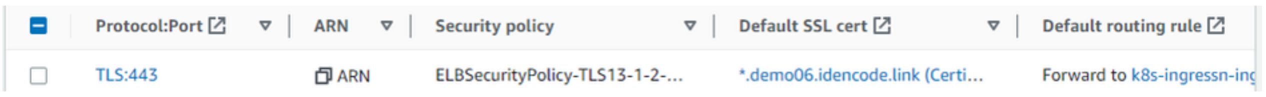
```
-f nginx.values.yaml

```

The above will automatically spawn an Internal AWS Network Load Balancer (L4).

- Check the following on AWS console:
  - An NLB has been created. You may also see the DNS of NLB with

```
kubectl -n ingress-nginx get svc
```

  - So now go to LB section in AWS, and select rancher LB and come down you will find listners. Edit listner "443". Select "TLS". Do select the target group of listner 80 in 443 listner(`Default Action` section). And under that select the `` `ACM SSL certificate `` that you have created for you rancher domain. You can find the below image for the same.



	Protocol:Port	ARN	Security policy	Default SSL cert	Default routing rule
☐	TLS:443	ARN	ELBSecurityPolicy-TLS13-1-2-...	*.demo06.idencode.link (Certi...	Forward to k8s-ingressn-ing

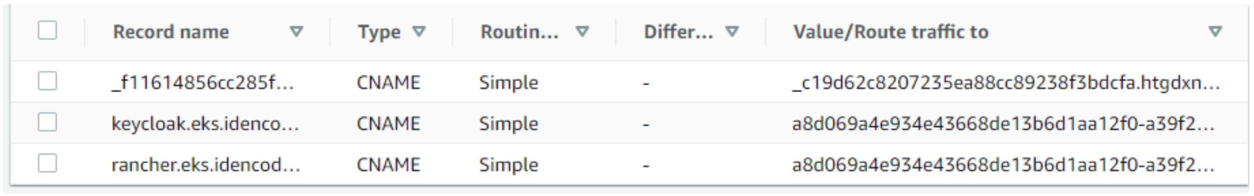
  - Basically, we want TLS termination at the LB and it must forward HTTP traffic (not HTTPS) to port 80 of ingress controller. So
    - Input of LB: HTTPS
    - Output of LB: HTTP --> port 80 of ingress nginx controller
  - Enable "Proxy Protocol v2" in the target group that you have selected in `443`. find below path for the same.
EC2 > Target groups > k8s-ingressn-ingressn-cc567feb37 > Edit target group attributes
  - Make sure all subnets are selected in LB -->Description-->Edit subnets.
  - Check health check of target groups.
  - Remove listner 80 from LB as we will receive traffic only on 443.

DNS Requirements for rancher cluster

Create the following domain names:

- Rancher: `rancher.xyz.net`
- Keycloak: `keycloak.xyz.net`

Point the above to the **internal** IP address of the NLB. This assumes that you have a Wireguard Bastion Host has been installed. On AWS this is done on Route 53 console.



	Record name	Type	Routin...	Differ...	Value/Route traffic to
☐	_f11614856cc285f...	CNAME	Simple	-	_c19d62c8207235ea88cc89238f3bdcfa.htgdxn...
☐	keycloak.eks.idenco...	CNAME	Simple	-	a8d069a4e934e43668de13b6d1aa12f0-a39f2...
☐	rancher.eks.idencod...	CNAME	Simple	-	a8d069a4e934e43668de13b6d1aa12f0-a39f2...

Rancher K8s Cluster Apps Installation and Storage class setup

- **Storage class setup**: You can go with the default gp2 EBS volume attached to all the nodes or else you can install LongHorn also but longhorn will not provide you the multipath option for PVC. So go with default gp2 one as default keep longhorn as optional storage class. And to mount this gp2 PVCs you need to install `aws-ebs-csi-driver-on-aws-eks-for-persistent-storage` follow this document here .
  - After installing make sure storage class should be default one. To check run this command `kubectl get sc` it will show all the storage classes and if your storage class is not set as default one go with this document to make it default one.
- You can Install LongHorn also.
- **Rancher UI** : Rancher provides full CRUD capability of creating and managing Kubernetes cluster.

- Install rancher using Helm, update `hostname` in `rancher-values.yaml,` and run the following command to install.

  -
    ```
    cd $K8_ROOT/rancher/rancher-ui
    helm repo add rancher-latest https://releases.rancher.com/server-charts/latest
    helm repo update
    helm install rancher rancher-latest/rancher \
      --namespace cattle-system \
      --create-namespace \
      -f rancher-values.yaml


    ```
- Login:
  - Open Rancher page `https://rancher.org.net`.
  - Get a Bootstrap password using
    ```
    kubectl get secret --namespace cattle-system bootstrap-secret -o go-template='{{ .data.bootstrapPassword
    ```
  - Assign a password. IMPORTANT: makes sure this password is securely saved and retrievable by Admin.

- **Keycloak**: Keycloak is an OAuth 2.0 compliant Identity Access Management (IAM) system used to manage the access to Rancher for cluster controls.

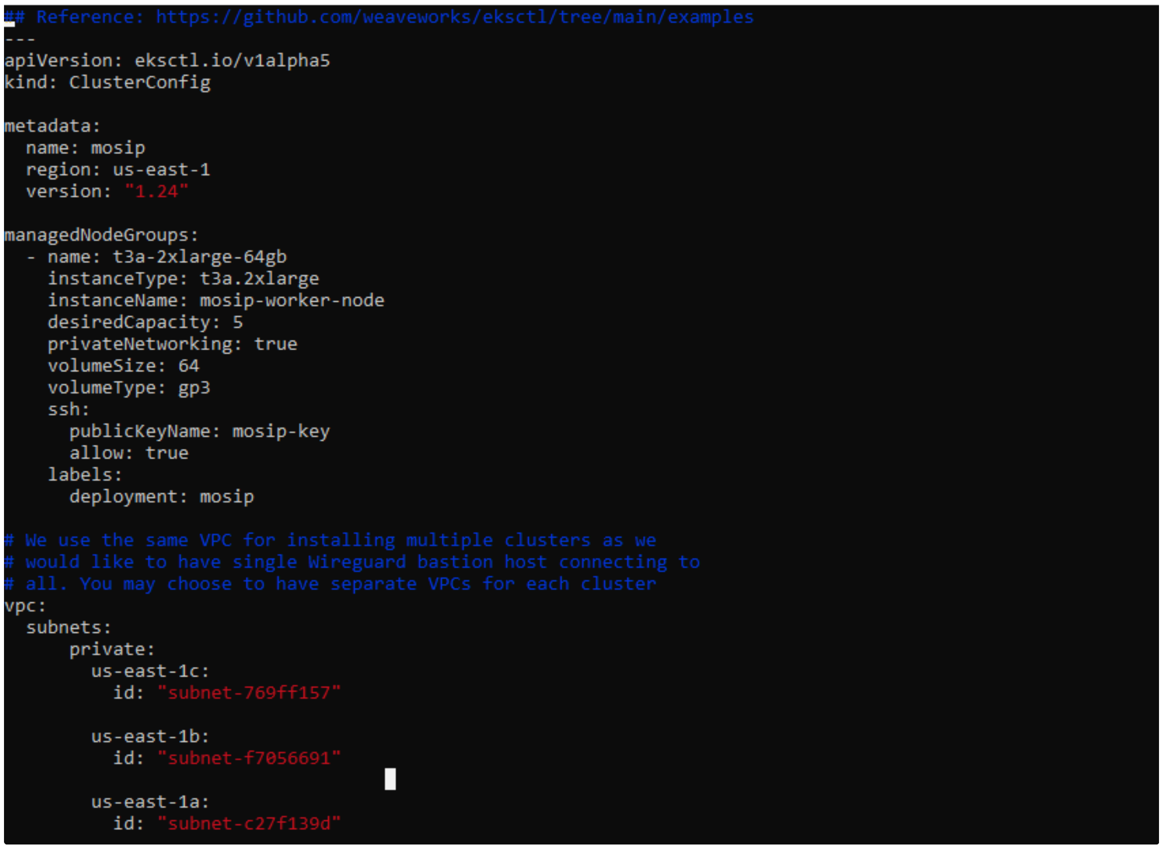
  -
    ```
    cd $K8_ROOT/rancher/keycloak
    ./install.sh <iam.host.name>

    ```
  - `keycloak_client.json`: Used to create SAML client on Keycloak for Rancher integration.
- **Keycloak - Rancher Integration**
  - Login as `admin` user in Keycloak and make sure an email id, and first name field is populated for `admin` user. This is important for Rancher authentication as given below.
  - Enable authentication with Keycloak using the steps given here.
  - In Keycloak add another Mapper for the rancher client (in Master realm) with following fields:
    - Protocol: saml
    - Name: username
    - Mapper Type: User Property
    - Property: username
    - Friendly Name: username
    - SAML Attribute Name: username
    - SAML Attribute NameFormat: Basic
  - Specify the following mappings in Rancher's Authentication Keycloak form:
    - Display Name Field: givenName
    - User Name Field: email
    - UID Field: username
    - Entity ID Field: https://your-rancher-domain/v1-saml/keycloak/saml/metadata
    - Rancher API Host: https://your-rancher-domain
    - Groups Field: member

AWS EKS Cluster Setup for Mosip

Overview

The instructions here install an EKS cluster on AWS along with Network Loadbalancer and Istio. We have chosen cloud's Network Load Balancer (Layer 4) over Application Load Balancer (Layer 7) as we have application load balancing done by Istio Ingress running inside the cluster.

- prerequisites are the same as you above installed for rancher cluster creation.
- Hardware, network, certificate requirements. Compute Node requirements are already configured in `cluster.config.sample`. But make sure this config is as per your requirement.
- Install
  - Copy `cluster.config.sample` to `mosip.cluster.config`.
  - Review the parameters of `mosip.cluster.config` carefully. Please find the below image for the config script. And make sure before running script your rancher config should not present in `.kube` folder with the name `config`.



```
## Reference: https://github.com/weaveworks/eksctl/tree/main/examples
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: mosip
  region: us-east-1
  version: "1.24"

managedNodeGroups:
  - name: t3a-2xlarge-64gb
    instanceType: t3a.2xlarge
    instanceName: mosip-worker-node
    desiredCapacity: 5
    privateNetworking: true
    volumeSize: 64
    volumeType: gp3
    ssh:
      publicKeyName: mosip-key
      allow: true
    labels:
      deployment: mosip

# We use the same VPC for installing multiple clusters as we
# would like to have single Wireguard bastion host connecting to
# all. You may choose to have separate VPCs for each cluster
vpc:
  subnets:
      private:
        us-east-1c:
          id: "subnet-769ff157"

        us-east-1b:
          id: "subnet-f7056691"

        us-east-1a:
          id: "subnet-c27f139d"
```

  - Install

```
eksctl create cluster -f mosip.cluster.config
```

  - It takes around 30 minutes to create (or delete a cluster).
  - The last line of output is similar to the following example line.
    - `[✓] EKS cluster "my-cluster" in "region-code" region is ready`
  - In case the cluster failed due to some error delete the cluster with the below command and recreate it.
    - `eksctl delete cluster --region=(your region) --name=rancher`
  - The config file for the new cluster will be created on `~/.kube/config`
  - After creating the cluster make a backup copy of the `config` with a suitable name in `~/.kube/` folder, eg. `mosip_config` because if you create the cluster again using `eksctl` it will override the existing `~/.kube/config`. Set file permission to `chmod 400 ~/.kube/rancher_config` to avoid any accidental changes or deletions.

MOSIP K8 Cluster Global configmap, Ingress, and Storage Class setup

- **Global configmap**: Global configmap contains the list of necessary details to be used throughout the namespaces of the cluster for common details.
  - `cd $K8_ROOT/mosip`
  - Copy `global_configmap.yaml.sample` to `global_configmap.yaml.`
  - Update the domain names in `global_configmap.yaml` and run.
    - `kubectl apply -f global_configmap.yaml`

Import Mosip Cluster into Rancher UI

- Login as admin in the Rancher console
- Select `Import Existing` for cluster addition.
- Select `Generic` as the cluster type to add.
- Fill the `Cluster Name` field with a unique cluster name and select `Create`.
- You will get the kubectl commands to be executed in the Kubernetes cluster. Copy the command and execute from your PC. (make sure your kube-config file is correctly set to Mosip cluster)

```
eg.
kubectl apply -f https://rancher.eks.sandbox.net/v3/import/pdmkx6b4xxtpcd699gzwdtt5bckwf4ctdgr7xkmmtwg8dfjk4hmbpk

```

- Wait for few seconds after executing the command for the cluster to get verified.
- Your cluster is now added to the rancher management server.

Storage Class setup

GP2

- Default storage class on EKS is `gp2` which by is in "Delete" mode which means if PV is deleted, the underlying storage is also deleted.
- Add the `gp2` storage class config to add the following in YAML: `allowVolumeExpansion: true`.
- Create storage class `gp2-retain` by running `sc.yaml` for PV in Retain mode. Set the storage class as gp2-retain in case you want to retain PV. Learn more on persistence.

```
kubectl apply -f sc.yaml
```

- If the PV gets deleted (say cluster was retarted), then you will have to define a PV connecting to this instance of storage (you will need volume ID etc). TODO: how to do this?
- You can go with the default gp2 EBS volume attached to all the nodes or else you can install LongHorn also but longhorn will not provide you the multipath option for PVC. So go with default gp2 one as default keep longhorn as optional storage class. And to mount this gp2 PVCs you need to install `aws-ebs-csi-driver-on-aws-eks-for-persistent-storage` follow this document here .
- After installing make sure storage class should be default one. To check run this command `kubectl get sc` it will show all the storage classes and if your storage class is not set as default one go with this document to make it default one.
- You can Install LongHorn also.

Volume expansion

If a particular PVC is running short of storage, follow these steps for EKS (AWS) storage.

1. Find out the name of PV corresponding to the PVC from Rancher.
2. On AWS console --> Volumes search for this volume.
3. Increase the storage of this volume to the desired capacity. *(Storage can only be increased)*.
4. On Rancher, edit PV's YAML config to match the above capacity.

5. On Rancher, edit PVC's YAML to increase the requested storage to the same capacity.

The capacity of the PVC should have increased.

Ingress and load balancer (LB)

Ingress is not installed by default on EKS. We use Istio ingress gateway controller to allow traffic in the cluster. Two channels are created - public and internal. See architecture.

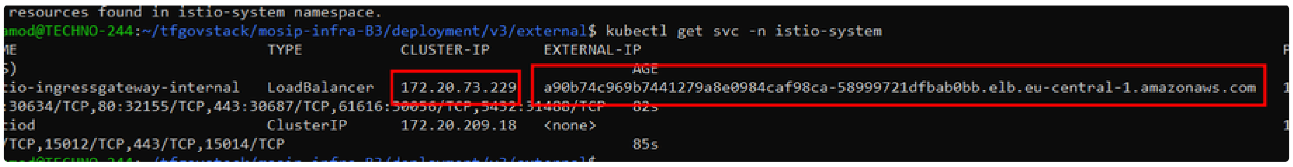
- Install Istioctl as given here
- Install ingresses as given here:

```
cd istio
./install.sh

```

Load Balancers

The above steps will spin-off a load balancer on AWS. You may view them on AWS console. These may be also seen with this command and make sure `CLUSTER-IP` and `EXTERNAL-IP` should be present. Then only your able to map domains to load balancer.

```
kubectl  get svc -n istio-system
```

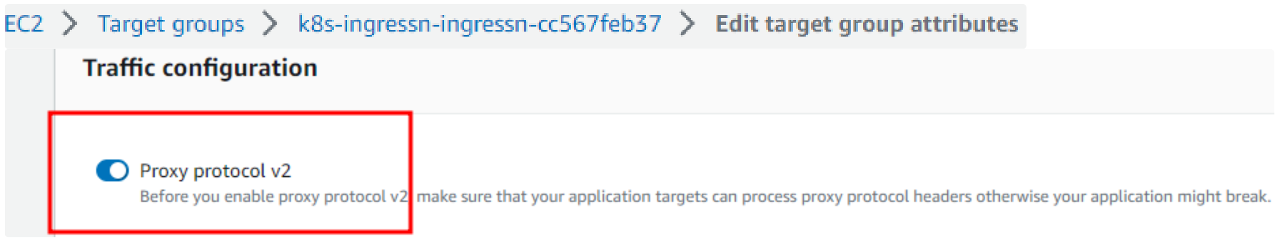


```
 resources found in istio-system namespace.
amod@TECHNO-244:~/tfgovstack/mosip-infra-B3/deployment/v3/external$ kubectl get svc -n istio-system
ME                          TYPE           CLUSTER-IP      EXTERNAL-IP                                                                 P
5)                                                                     AGE
io-ingressgateway-internal   LoadBalancer   172.20.73.229   a90b74c969b7441279a8e0984caf98ca-58999721dfbab0bb.elb.eu-central-1.amazonaws.com   1
:30634/TCP,80:32155/TCP,443:30687/TCP,61616:30058/TCP,5432:31408/TCP   82s
iod                         ClusterIP      172.20.209.18   <none>                                                                      1
/TCP,15012/TCP,443/TCP,15014/TCP                                       85s
```

- Make sure in EKs deployment we are using only one `ingress-gateway-internal` LB for both public and private routing we are not using `ingress-gateway-public`.
- TLS termination is supposed to be on LB. So all our traffic coming to ingress controller shall be HTTP.
- Obtain AWS TLS certificate as given here. Ignore if you have already created one.
- Add the certificates and 443 access to the LB listener.
  - Update listener TCP->443 to **TLS->443** and point to the certificate of domain name that belongs to your cluster. Same as you have done for rancher LB.
  - Forward TLS->443 listner traffic to target group that corresponds to listner on port 80 of respective Loadbalancers. This is because after TLS termination the protocol is HTTP so we must point LB to HTTP port of ingress controller.
- Update health check ports of LB target groups to node port corresponding to port 15021. You can see the node ports with

```
kubectl -n istio-system get svc
```

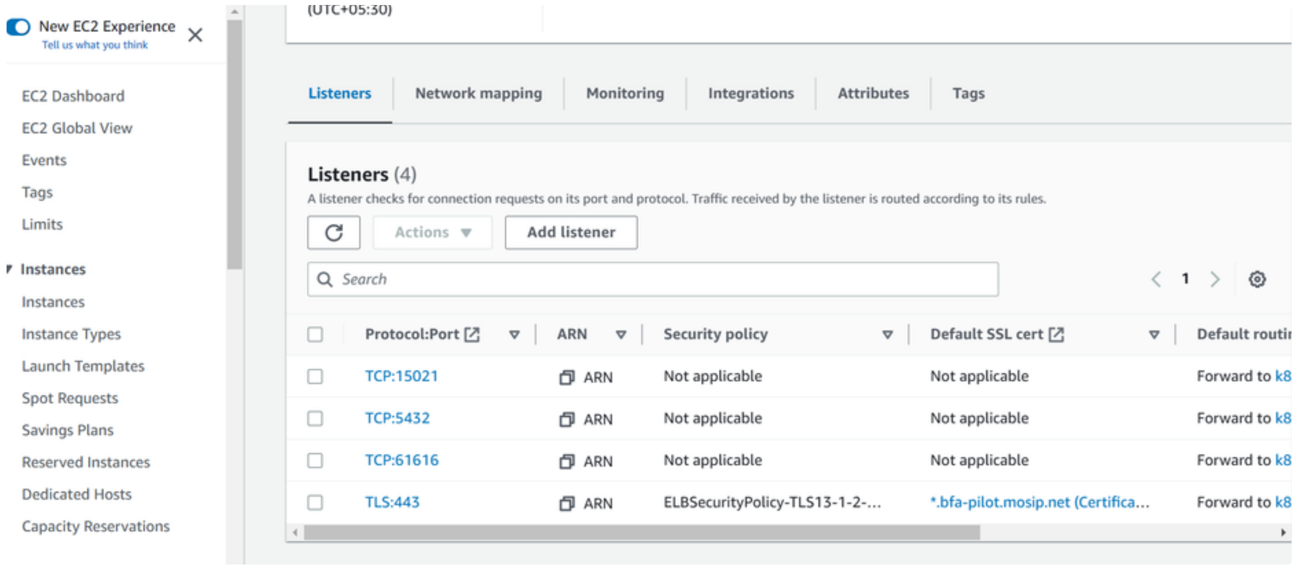
- Enable Proxy Protocol v2 on all target groups.


EC2 > Target groups > k8s-ingressn-ingressn-cc567feb37 > Edit target group attributes
Traffic configuration

Proxy protocol v2

Before you enable proxy protocol v2 make sure that your application targets can process proxy protocol headers otherwise your application might break.

- Make sure all subnets are included in Availability Zones for the LB. Description --> Availability Zones --> Edit Subnets
- Make sure to delete the listenrs for port 80 and 15021 from each of the loadbalancers as we restrict unsecured port 80 access over HTTP. Find the below image for the same changes.



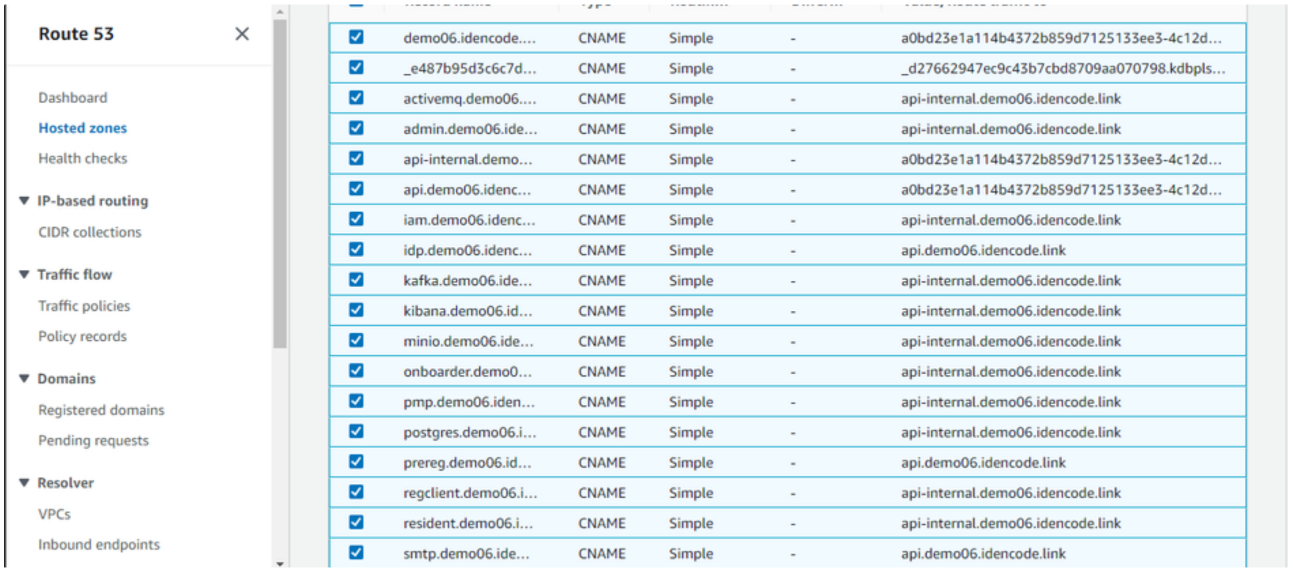
Listeners (4)

A listener checks for connection requests on its port and protocol. Traffic received by the listener is routed according to its rules.

	Protocol:Port	ARN	Security policy	Default SSL cert	Default routi
☐	TCP:15021	ARN	Not applicable	Not applicable	Forward to k8
☐	TCP:5432	ARN	Not applicable	Not applicable	Forward to k8
☐	TCP:61616	ARN	Not applicable	Not applicable	Forward to k8
☐	TLS:443	ARN	ELBSecurityPolicy-TLS13-1-2-...	*.bfa-pilot.mosip.net (Certifica...	Forward to k8

DNS Requirements for mosip cluster:

- Initially, all the services will be accessible only over the internal channel.
- Point all your domain names to internal LoadBalancers DNS/IP initially till testing is done.
- On AWS this may be done on Route 53 console.
- After the Go-live decision enables public access.
- Find the below image for routing on Route53.
- The below image shows an example placeholder for hostnames, the actual name itself varies from organization to organization.
- Only proceed to DNS mapping after the ingressgateways are installed and the nginx reverse proxy is setup.



✓	demo06.idencode....	CNAME	Simple	-	a0bd23e1a114b4372b859d7125133ee3-4c12d...
✓	_e487b95d3c6c7d...	CNAME	Simple	-	_d27662947ec9c43b7cbd8709aa070798.kdbpls...
✓	activemq.demo06....	CNAME	Simple	-	api-internal.demo06.idencode.link
✓	admin.demo06.ide...	CNAME	Simple	-	api-internal.demo06.idencode.link
✓	api-internal.demo...	CNAME	Simple	-	a0bd23e1a114b4372b859d7125133ee3-4c12d...
✓	api.demo06.idenc...	CNAME	Simple	-	a0bd23e1a114b4372b859d7125133ee3-4c12d...
✓	iam.demo06.idenc...	CNAME	Simple	-	api-internal.demo06.idencode.link
✓	idp.demo06.idenc...	CNAME	Simple	-	api.demo06.idencode.link
✓	kafka.demo06.ide...	CNAME	Simple	-	api-internal.demo06.idencode.link
✓	kibana.demo06.id...	CNAME	Simple	-	api-internal.demo06.idencode.link
✓	minio.demo06.ide...	CNAME	Simple	-	api-internal.demo06.idencode.link
✓	onboarder.demo0...	CNAME	Simple	-	api-internal.demo06.idencode.link
✓	pmp.demo06.iden...	CNAME	Simple	-	api-internal.demo06.idencode.link
✓	postgres.demo06.i...	CNAME	Simple	-	api-internal.demo06.idencode.link
✓	prereg.demo06.id...	CNAME	Simple	-	api.demo06.idencode.link
✓	regclient.demo06.i...	CNAME	Simple	-	api-internal.demo06.idencode.link
✓	resident.demo06.i...	CNAME	Simple	-	api-internal.demo06.idencode.link
✓	smtp.demo06.ide...	CNAME	Simple	-	api.demo06.idencode.link

Check Overall if nginx and istio wiring is set correctly

- Install `httpbin` for testing the wiring as per httpbin check.

Monitoring Logging Module deployment

- Install monitoring and logging from here

Mosip External Dependencies setup

- External Dependencies are a set of external requirements needed for functioning of MOSIP's core services like DB, object store, hsm etc.

  -
    ```
    cd $INFRA_ROOT/deployment/v3/external/all
    ./install-all.sh

    ```
- Check detailed installation instruction of all the external componets

MOSIP Modules Deployment

- Now that all the Kubernetes cluster and external dependencies are already installed, will continue with MOSIP service deployment.

  -
    ```
    cd $INFRA_ROOT/deployment/v3/mosip/all
    ./install-all.sh

    ```
- Check detailed MOSIP Modules Deployment MOSIP Modular installation steps.

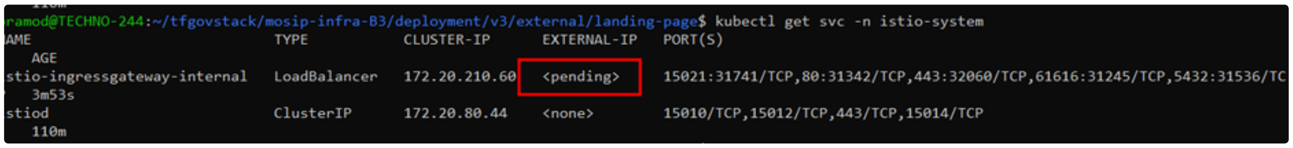
Api Testrig

- MOSIP's sucessfull deployment can be verified by comparing the results of api testrig with testrig benchmark.

  -
    ```
    cd $INFRA_ROOT/deployment/v3/apitestrig
    ./install.sh


    ```
    - When prompted input the hour of the day to execute the api-testrig.
    - Daily api testrig cron jon will be executed at the very opted hour of the day.

Troubleshooting

- If you are facing any issues while accessing the domain names that could be because `proxy-protocol` is not enabled in the target groups. or routing is not done properly and LB listners configurations are not done properly so check everything once again.
- When accessing istio-system from terminal it should show DNS name of load balancer in `EXTERNAL-IP` section or else not able to access endpoints. It causes because of multiple `security-groups` attached to your nodes. Make sure only one security-group attached to each node.



```
ramod@TECHNO-244:~/tfgovstack/mosip-infra-B3/deployment/v3/external/landing-page$ kubectl get svc -n istio-system
AME                          TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)
   AGE
stio-ingressgateway-internal  LoadBalancer   172.20.210.60   <pending>     15021:31741/TCP,80:31342/TCP,443:32060/TCP,61616:31245/TCP,5432:31536/TC
   3m53s
stiod                         ClusterIP      172.20.80.44    <none>        15010/TCP,15012/TCP,443/TCP,15014/TCP
   110m
```

- Sometimes while creating clusters the cluster will create successfully but it will not generate a cluster-config file in `.kube` folder then run this command to generate the config file after cluster creation.
  `eksctl utils write-kubeconfig -n <clustername>`
- Sometimes you will face PVC mount volumes issue for all the stateful sets, and completely env will go down because of restarting nodes or scaling down the nodes. So please avoid this in case of EKS or else you can use EFS as a default storage class to avoid such issues..
- If you want to decrease/increase the no of nodes, you can do that on EKS section NodeGroup group in AWS.
- If you want to delete the whole EKS setup follow this link here