Mosip Environment Deployment on On-prem (AWS,TF-hardware)

INSTALL WINDOWS SUBSYSTEM FOR LINUX

{if not installing enable WSL in windows}

How to Enable Windows Subsystem for Linux.

Open Windows 10 Settings-->

Select Apps.

-Click Programs and Features under the Related settings section on the right.

-Under the Programs and Features page, click Turn Windows features on or off on the left panel.

-Scroll down and enable Windows Subsystem for Linux.

-Click OK to save your changes.

-Hit Restart now to finish the process.

-Similar search: enable windows subsystem for linux

-Reference: How to Enable Windows Subsystem for Linux

INSTALL UBUNTU WSL FROM MICROSOFT STORE

ON WSL :

COPY PEM KEY FROM LOCAL TO WSL

```
sudo chmod 400 <pemkey name> --> only single permission to file
do passwordless authentication between the hosts
```

Install Ansible

```
sudo apt update
sudo apt upgrade
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install -y ansible
```

Install rkev1.3.10

Kubernetes cluster provisioning with Rancher's RKE command - DevOpsSchool.com or Release Release v1.3.10 · rancher/rke

```
sudo apt install wget -y
wget https://github.com/rancher/rke/releases/download/v1.3.10/rke_linux-amd64
chmod 755 rke_linux-amd64
mv rke_linux-amd64 rke
echo $PATH
sudo mv rke /usr/sbin
```

Install helm

```
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3
sudo chmod 700 get_helm.sh
sudo ./get_helm.sh
helm version --client
```

Install kubectl

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
curl -LO "https://dl.k8s.io/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"
```

```
echo "$(cat kubectl.sha256)  kubectl" | sha256sum --check --------this should give output as kubectl: OK
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
chmod +x kubectl
mkdir -p ~/.local/bin
mv ./kubectl ~/.local/bin/kubectl
kubectl version --client
```

Install ISTIO on WSL

refer: Release Istio 1.15.0 · istio/istio or Getting started with Istio on Kubernetes

```
wget https://github.com/istio/istio/releases/download/1.15.0/istio-1.15.0-linux-amd64.tar.gz
tar -xvzf istio-1.15.0-linux-amd64.tar.gz
cd istio-1.15.0
export PATH=$PWD/bin:$PATH
istioctl install --set profile=demo
```

This will install the Istio 1.15.0 demo profile with ["Istio core" "Istiod" "Ingress gateways" "Egress gateways"] components into the cluster. Proceed? (y/N): y

kubectl get pods --all-namespaces --> verify isto pods are up and running

CLONE THE REQUIRED 3 REPOSITORIES WITH V1.2.0.1-B3 TAG

git clone -b v1.2.0.1-B3 GitHub - mosip/k8s-infra: Kubernetes infrastructure to deploy MOSIP modules.
git clone -b v1.2.0.1-B3 GitHub - mosip/mosip-infra: MOSIP deployment Infrastructure repo
git clone -b v1.2.0.1-B1 GitHub - mosip/reporting: For MOSIP reporting and analytics

---

HARDWARE REQUIREMENTS FOR RANCHER CLUSTER

Purpose	vCPUs	RAM	Storage (SSD)	N Virtual Machines
Cluster nodes	2	8 GB	32 GB	2
Wireguard bastion host	2	1 GB	8 GB	1
Nginx	2	4GB	16 GB	1

CREATE WIREGAURE VM(instance) --> t2.micro
Copy id_rsa.pub from WSL to WireGuard Node

ON WIREGUARD INSTANCE

Ref: Wireguard Bastion Host

INSTALL DOCKER

```
sudo apt update
sudo apt install apt-transport-https ca-certificates curl software-properties-common -y
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] Index of linux/ubuntu/ focal stable"
sudo apt install docker-ce -y
sudo systemctl status docker
sudo usermod -aG docker $USER
```

INSTALL WIREGUARD

```
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/config:/config \
  -v /lib/modules:/lib/modules \
  --sysctl="net.ipv4.conf.all.src_valid_mark=1" \
  --restart unless-stopped \
```

GitHub

INSTALL WIREGAURD CLIENT ON SYSTEM

Ref: Installation - WireGuard

In above installation command peers means no. of clients

after installation config folder got created
copy the peer details for the particular peer

ON WIREGUARD CLIENT

Add tunnel to it and copy peer details
Add port of wireguard peer in security grup (custom UDP --> anywhere ipv4 , ipv6)
Open the required ports on aws Security group refer ports.yml

EXAMPLE:

```
 Address = 10.13.13.2
 PrivateKey = +NdQUctViIWf+E+hLg8fmn+X70M0Q+S1hZZgiCnZY0M=
 ListenPort = 51820  ---> (add this port in custom UDP as mention above)

 [Peer]
 PublicKey = 0kHwU/O6juzidCk6glh7l3RVgrWgAhRq8Pkh0jKUdQs=
 PresharedKey = 0m3N2FsCGs8Hh8nBLQQseeGP5lgyen/6FispuEmz3c4=
 Endpoint = 54.164.25.127:51820      ----> ip of wiregaurd VM or Instance / if stopped Vms change the Public ips in wireguard client
 AllowedIPs = 172.31.0.0/16         -----> ip of loadbalancer (incase of aws we have VPC cider block)
```

and ACTIVATE WIREGAURD CLIENT

LAUNCH 2 INSTANCES FOR RANCHER NODES --> t2.large
open all the ports in security grup which is mention in ports.yml file (in k8s-infra repo)
k8s-infra/rancher/on-prem/ports.yml

Copy id_rsa.pub from WSL to Both Rancher Nodes

ON WSL

```
 cd k8s-infra/rancher/on-prem
 cp hosts.ini.sample hosts.ini     --->(make a copy of host.ini.sample file)
 Edit Host.ini files ---> with perticular users(root / ubuntu), ips of hosts and path of pemkey

 ansible-playbook -i hosts.ini ports.yaml
```

ansible-playbook -i hosts.ini docker.yaml   ---> if got error for docker installation (open docker.yml and remove wiregaurd host just keep cluster because we already have docker in wiregaurd)

run rke Config for the perticular no. of nodes

rke config

```
[+] Cluster Level SSH Private Key Path [~/.ssh/id_rsa]: /home/shiv/B1.pem
[+] Number of Hosts [1]: 2
[+] SSH Address of host (1) [none]: 172.31.88.97
[+] SSH Port of host (1) [22]:
[+] SSH Private Key Path of host (172.31.88.97) [none]: /home/shiv/B1.pem
[+] SSH User of host (172.31.88.97) [ubuntu]:
[+] Is host (172.31.88.97) a Control Plane host (y/n)? [y]: y
[+] Is host (172.31.88.97) a Worker host (y/n)? [n]: y
[+] Is host (172.31.88.97) an etcd host (y/n)? [n]: y
[+] Override Hostname of host (172.31.88.97) [none]: Node1
[+] Internal IP of host (172.31.88.97) [none]: 172.31.88.97
[+] Docker socket path on host (172.31.88.97) [/var/run/docker.sock]:
[+] SSH Address of host (2) [none]: 172.31.84.72
[+] SSH Port of host (2) [22]:
[+] SSH Private Key Path of host (172.31.84.72) [none]: /home/shiv/B1.pem
[+] SSH User of host (172.31.84.72) [ubuntu]:
[+] Is host (172.31.84.72) a Control Plane host (y/n)? [y]: y
[+] Is host (172.31.84.72) a Worker host (y/n)? [n]: y
[+] Is host (172.31.84.72) an etcd host (y/n)? [n]: y
[+] Override Hostname of host (172.31.84.72) [none]: Node2
[+] Internal IP of host (172.31.84.72) [none]: 172.31.84.72
[+] Docker socket path on host (172.31.84.72) [/var/run/docker.sock]:
[+] Network Plugin Type (flannel, calico, weave, canal, aci) [canal]:
[+] Authentication Strategy [x509]:
[+] Authorization Mode (rbac, none) [rbac]:
[+] Kubernetes Docker image [rancher/hyperkube:v1.22.9-rancher1]:
[+] Cluster domain [cluster.local]:
[+] Service Cluster IP Range [10.43.0.0/16]:
[+] Enable PodSecurityPolicy [n]:
[+] Cluster Network CIDR [10.42.0.0/16]:
[+] Cluster DNS Service IP [10.43.0.10]:
[+] Add addon manifest URLs or YAML files [no]:
```

---

cluster.yml is created

nano cluster.yml Remove the default Ingress install by editing :

```
ingress:
  provider: none
```

and give cluster name "<any name>" -->(B1_Rancher)

rke up -> this will setup cluster

After successfull creation of cluster follow the below steps:

cp kube_config_cluster.yml $HOME/.kube/onprem_Rancher_config   <give file name as per your choice for eg : onprem_Rancher_config>
chmod 400 $HOME/.kube/onprem_Rancher_config
sudo cp $HOME/.kube/onprem_Rancher_config $HOME/.kube/config
export KUBECONFIG="$HOME/.kube/onprem_Rancher_config"

kubectl get nodes

INSTALL INGRESS CONTROLLER (WSL)

helm repo add ingress-nginx Welcome - Ingress-Nginx Controller
helm repo update

```
helm install ingress-nginx ingress-nginx/ingress-nginx \
  --namespace ingress-nginx \
  --version 4.0.18 \
  --create-namespace  \
  -f ingress-nginx.values.yaml
```

SETUP NGINX REVERSE PROXY

CREATE 1 INSTANCE FOR RANCHER NGINX --> t2.medium

ON RANCHER NGINX INSTANCE

(open the https http port in security group)

INSTALL PYTHON-3

```
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
sudo apt install python3.8 -y
python3 --version
```

SSL CERTIFICATES WITH LETSENCRYPT

```
sudo apt install letsencrypt -y
sudo apt install certbot python3-certbot-nginx -y
```

ON AWS CONSOLE

ROUTE 53 SETTINGS

Route53--> hosted zones --> idencode.link --> create record

name: onprem1.idencode.link
value: private ip of Nginx instance
CREATE RECORD

name: rancher.onprem1.idencode.link
value: private ip of Nginx instance
CREATE RECORD

name: keycloak.onprem1.idencode.link
value: private ip of Nginx instance
CREATE RECORD

ON RANCHER NGINX INSTANCE

Generate certificates for your domain name {EXAMPLE: onprem1.idencode.link}

    sudo certbot certonly --agree-tos --manual --preferred-challenges=dns -d *.sandbox.xyz.net -d sandbox.xyz.net

modify the above command with your domain name (as shown below)

    sudo certbot certonly --agree-tos --manual --preferred-challenges=dns -d *.onprem1.idencode.link -d onprem1.idencode.link

run command provide (email id , yes, yes)

create TXT RECORD --> {ON ROUTE 53 AWS CONSOLE}
_acme-challenge.onprem1.idencode.link
value: as output shown on cmd

GO TO URL --> DNS Propagation Checker - Global DNS Checker Tool
copy _acme-challenge.techno.idencode.link record name and paste in above link --> select TXT and click on search all sholud be green ticked

(wait fro few minutes and then click to continue)

Press Enter to Continue

Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/techno.idencode.link/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/techno.idencode.link/privkey.pem
  Your cert will expire on 2023-04-24. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot
  again. To non-interactively renew *all* of your certificates, run
  "certbot renew"
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt: Donate
  Donating to EFF: Support EFF's Work on Let's Encrypt

WE WILL SEE THE SUCCESS MESSAGE AS ABOVE

host -t TXT _acme-challenge.onprem1.idencode.link   <edit domain name accordingly>

NGINX INSTALLATION ON RANCHAR NGINX INSTANCE

git clone -b v1.2.0.1-B3 GitHub - mosip/k8s-infra: Kubernetes infrastructure to deploy MOSIP modules.
cd k8s-infra/rancher/on-prem/nginx/
sudo ./install.sh

=====>
The following internal ip will have to be DNS-mapped to rancher.xyz.net and iam.xyz.net.

Give the internal interface ip of this node here. Run `ip a` to get all the interface addresses (without any whitespaces) : 172.31.15.120 --> ip of nginx server (rancher)
=====>
Give path for SSL Certificate for rancher.xyz.net (without any whitespaces) : /etc/letsencrypt/live/onprem1.technoforte.co.in/fullchain.pem
=====>
Give path for SSL Certificate Key for rancher.xyz.net (without any whitespaces) : /etc/letsencrypt/live/onprem1.technoforte.co.in/privkey.pem
=====>
Give list of ips of all nodes in the rancher cluster (without any whitespaces, comma seperated) : 172.31.92.246,172.31.95.138
=====>
Give nodeport of the ingresscontroller of rancher cluster (without any whitespaces) (default is 30080) :
Reading package lists... Done
Building dependency tree
Reading state information... Done
nginx is already the newest version (1.18.0-0ubuntu1.4).
0 upgraded, 0 newly installed, 0 to remove and 13 not upgraded.
Nginx Installation succesful

sudo systemctl status nginx (verify nginx status)

INSTALL RANCHER ON WSL USING HELM

cd k8s-infra/rancher/rancher-ui
sudo nano rancher-values.yml
update hostname from Route 53 records (example: rancher.onprem1.idencode.link)

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

kubectl get all -n cattle-system (verify the pods and replicaset got generated)

GO TO RANCHER URL: https://rancher.onprem1.idencode.link
login as admin admin
keep copy of rancher demo password for further logins: xo6VfB7dX7RYYxjQ

INSTALL PREREQUISITES FOR LONGHORN ON WSL

cd k8s-infra/mosip/longhorn
./pre_install.sh

GO TO RANCHER URL: https://rancher.onprem1.technoforte.co.in
--> apps --> Longhorn (101.1.0+up1.3.2) --> Install
--> next --> under edit options --> Longhorn storage class setting set replicas as (2)
--> under edit YAML option -->

line no. 31 : guaranteedEngineManagerCPU: 5
line no. 32 : guaranteedReplicaManagerCPU: 5

INSTALL

KEYCLOAK SETUP

cd /k8s-infra/rancher/keycloak
sudo nano values.yaml

edit hostname : keycloak.onprem1.idencode.link

./install.sh keycloak.onprem1.technoforte.co.in --> run file with keycloak host name

```
    YOU WILL GET THIS OUTPUT :
    ** Please be patient while the chart is being deployed **

        Keycloak can be accessed through the following DNS name from within your cluster:

            keycloak.keycloak.svc.cluster.local (port 80)

        To access Keycloak from outside the cluster execute the following commands:

        1. Get the Keycloak URL and associate its hostname to your cluster external IP:

          export CLUSTER_IP=$(minikube ip) # On Minikube. Use: `kubectl cluster-info` on others K8s clusters
          echo "Keycloak URL: <http://keycloak.techno.idencode.link/auth">
          echo "$CLUSTER_IP  keycloak.techno.idencode.link" | sudo tee -a /etc/hosts

        2. Access Keycloak using the obtained URL.
        3. Access the Administration Console using the following credentials:

         echo Username: admin
         echo Password: $(kubectl get secret --namespace keycloak keycloak -o jsonpath="{.data.admin-password}" | base64 --decode)
```

RUN THE COMMANDS FROM OUTPUT CAREFULLY IN SEQUENCE AS SHOWN BELOW

```
kubectl cluster-info
export CLUSTER_IP=$10.2.1.52    → Ip of nginx node
echo "Keycloak URL: http://keycloak.onprem1.idencode.link/auth"
echo "$CLUSTER_IP  keycloak.onprem1.idencode.link" | sudo tee -a /etc/hosts
```

sudo nano /etc/hosts --> cross check the domain is mapped to ip or not and crosscheck the IP

kubectl get all -n keycloak

add the postgres SQL port 5432 to rancher node security group --> now we will able to see the volume on Longhorn UI

access https://keycloak.onprem1.idencode.link/auth --> on browser
administration console--> user: admin
password: go to rancher-->secrates-->keycloak and copy password and login
password : UPx4jICy2L

RANCHER AND KEYCLOAK INTEGRATION

Ref: Configure Keycloak (SAML) | Rancher
Ref: Rancher v2.X KeyCloak Authentication Backend Configuration

KEYCLOAK CONFIGURATION

Create a new client

Client ID: https://rancher.onprem1.idencode.link/v1-saml/keycloak/saml/metadata -->{replace your host}
Client Protocol: saml
Root URL: Leave empty
SAVE

Name: rancher
Enabled: ON
Login Theme: keycloak
Sign Documents: ON
Sign Assertions: ON
Encrypt Assertions: OFF
Client Signature Reguired: OFF
Force Post Binding: OFF
Front Channel Logout: OFF
Force Name ID Format: OFF
Name ID Format: username
Valid Redirect URLs: https://rancher.onprem1.idencode.link/v1-saml/keycloak/saml/acs -->{replace your host}
IDP Initiated SSO URL Name: IdPSSOName

SAVE

CLICK ON MAPPERS-->CREATE

<u>MAPPERS FOR RANCHER CLIENT</u>

1.

Protocol: saml
Name: username
Mapper Type: User Property
Property: username
Friendly Name: username
SAML Attribute Name: username
SAML Attribute NameFormat: Basic
SAVE

2.

Protocol: saml
Name: groups
Mapper Type: Group list
Group attribute name: member
Friendly Name: Leave empty
SAML Attribute Name: Basic
Single Group Attribute: ON
Full group path: OFF
SAVE

ADD BUILTIN --> select all --> ADD SELECTED

SAML Descriptior XML file
https://keycloak.onprem1.idencode.link/auth/realms/master/protocol/saml/descriptor --> {replace your keycloak host in link}

GENERATE OPEN SSL CERTIFICATE
run below commad in local system (through gitbash ,cmd etc)

```
openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout myservice.key -out myservice.cert
```

GO TO RANCHER:

Ref: https://github.com/mosip/k8s-infra/tree/main/rancher - Connect your Github account

user and autherntication--> auth provider--> keycloak --> follow the document

Display Name Field: givenName
User Name Field: uid or email
UID Field: username
Groups Field: member
Entity ID Field:
Rancher API Host: https://rancher.onprem1.idencode.link

PROVIDE PRIVATE KEY GENERATED IN OPEN SSL CERTIFICATE
PROVIDE CERTIFICATE GENERATED IN OPEN SSL CERTIFICATE
PROVIDE SAML DESCRIPTIOR XML FILE OUTPUT

ENABLE

After successfull Indegration of Keycloak and Rancher you should be able to login to keycloak through Rancher URL

MOSIP CLUSTER CREATION

HARDWARE REQUIREMENTS FOR MOSIP CLUSTER

Purpose	vCPUs	RAM	Storage (SSD)	Number of VMs*
Cluster nodes	12	32 GB	128 GB	6
Wireguard bastion host	2	4 GB	8 GB	1
Nginx	2	4GB	16 GB	1

NOTE: NO NEED TO CREATE NEW WIREGUARD USE THE PREVIOUSLY CREATED

MOSIP CLUSTER CREATION

CREATE 6 INSTANCES FOR MOSIP CLUSTER NODES --> t2.2xlarge with volume 128 GB
CREATE 1 INSTANCE FOR MOSIP NGINX --> t2.medium add ports (61616 ,5432, http https)

Copy id_rsa.pub from WSL to all Mosip Nodes and Nginx
and open ports from ports.yml for Mosip Nodes

cd k8s-infra/mosip/on-prem

cp hosts.ini.sample hosts.ini
Edit hosts.ini file in mosip with particular ips

ansible-playbook -i hosts.ini env-check.yaml   ---> To Perform Environment check on all the cluster nodes

ansible-playbook -i hosts.ini ports.yaml   ---> to open ports in firewall settings

nano docker.yaml ---> remove wireguard from hosts
ansible-playbook -i hosts.ini docker.yaml   --->Install docker on all nodes.

CREATE CLUSTER CONFIG FILE:

controlplane, etcd, worker: Specify controlplane, etc on at least three nodes. All nodes may be worker.
[+] Is host (172.31.84.72) a Control Plane host (y/n)? [y]: y
[+] Is host (172.31.84.72) a Control Plane host (y/n)? [y]: y
[+] Is host (172.31.84.72) a Control Plane host (y/n)? [y]: y
[+] Is host (172.31.84.72) a Control Plane host (y/n)? [y]: n

```
[+] Is host (172.31.84.72) a Control Plane host (y/n)? [y]: n
[+] Is host (172.31.84.72) a Control Plane host (y/n)? [y]: n
```

AS MENTION ABOVE MENTION y for three nodes and n for three nodes of mosip cluster

rke config

Remove the default Ingress install by editing cluster.yaml:

```
ingress:
  provider: none
```

ADD CLUSTER NAME IN MOSIP'S cluster.yml:

cluster_name: sandbox-name   --> name as per choice

Bring up the cluster:

rke up

```
cp kube_config_cluster.yml $HOME/.kube/onprem2_mosip_config
chmod 400 $HOME/.kube/onprem2_mosip_config
sudo cp  $HOME/.kube/onprem2_mosip_config  $HOME/.kube/config
export KUBECONFIG="$HOME/.kube/onprem2_mosip_config"
```

kubectl get nodes

GLOBAL CONFIGMAPS SETUPS

refer for the domain name format:-->
https://github.com/mosip/k8s-infra/blob/main/mosip/on-prem/requirements.md

DNS requirements

The following DNS mappings will be required.

create unique domain name record in route 53 with ip of mosip-nginx instance --> (choose diff domain from RANCHER AND KEYCLOAK created previously)

installation-domain	: onprem1v3.technoforte.co.in	--> private ip of mosip nginx node
mosip-api-host	: api.onprem1v3.technoforte.co.in	--> public ip of mosip nginx mode
mosip-api-internal-host	: api-internal.onprem1v3.technoforte.co.in	--> private ip of mosip nginx node
mosip-prereg-host	: prereg.onprem1v3.technoforte.co.in	--> CNAME record --> value = api.onprem1v3.technoforte.co.in
mosip-activemq-host	: activemq.onprem1v3.technoforte.co.in	--> CNAME record --> value = api-internal.onprem1v3.technoforte.co.in
mosip-kibana-host	: kibana.onprem1v3.technoforte.co.in	--> CNAME record --> value = api-internal.onprem1v3.technoforte.co.in
mosip-regclient-host	: regclient.onprem1v3.technoforte.co.in	--> CNAME record --> value = api-internal.onprem1v3.technoforte.co.in
mosip-admin-host	: admin.onprem1v3.technoforte.co.in	--> CNAME record --> value = api-internal.onprem1v3.technoforte.co.in
mosip-minio-host	: minio.onprem1v3.technoforte.co.in	--> CNAME record --> value = api-internal.onprem1v3.technoforte.co.in
mosip-kafka-host	: kafka.onprem1v3.technoforte.co.in	--> CNAME record --> value = api-internal.onprem1v3.technoforte.co.in
mosip-iam-external-host	: iam.onprem1v3.technoforte.co.in	--> CNAME record --> value = api-internal.onprem1v3.technoforte.co.in
mosip-postgres-host	: postgres.onprem1v3.technoforte.co.in	--> CNAME record --> value = api-internal.onprem1v3.technoforte.co.in
mosip-pmp-host	: pmp.onprem1v3.technoforte.co.in	--> CNAME record --> value = api-internal.onprem1v3.technoforte.co.in
mosip-onboarder-host	: onboarder.onprem1v3.technoforte.co.in	--> CNAME record --> value = api-internal.onprem1v3.technoforte.co.in
mosip-resident-host	: resident.onprem1v3.technoforte.co.in	--> CNAME record --> value = api.onprem1v3.technoforte.co.in
mosip-esignet-host	: esignet.onprem1v3.technoforte.co.in	--> CNAME record --> value = api.onprem1v3.technoforte.co.in
mosip-smtp-host	: smtp.onprem1v3.technoforte.co.in	--> CNAME record --> value = api-internal.onprem1v3.technoforte.co.in
mosip-healthservices-host	: healthservices.onprem1v3.technoforte.co.in	--> CNAME record --> value = api.onprem1v3.technoforte.co.in

```
cd k8s-infra/mosip/
cp global_configmap.yaml.sample global_configmap.yaml
```

```
sudo nano global_configmap.yaml
```

Update the domain names in global_configmap.yaml and run

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: global
  namespace: default
data:
  installation-name: technoforte
  installation-domain: onprem1v3.technoforte.co.in
  mosip-version: v1.2.0.1-B2                ---->(verify version)
  mosip-api-host: api.onprem1v3.technoforte.co.in
  mosip-api-internal-host: apiinternal.onprem1v3.technoforte.co.in
  mosip-prereg-host: prereg.onprem1v3.technoforte.co.in
  mosip-activemq-host: activemq.onprem1v3.technoforte.co.in
  mosip-kibana-host: kibana.onprem1v3.technoforte.co.in
  mosip-admin-host: admin.onprem1v3.technoforte.co.in
  mosip-regclient-host: regclient.onprem1v3.technoforte.co.in
  mosip-minio-host: minio.onprem1v3.technoforte.co.in
  mosip-kafka-host: kafka.onprem1v3.technoforte.co.in
  mosip-iam-external-host: iam.onprem1v3.technoforte.co.in
  mosip-postgres-host: postgres.onprem1v3.technoforte.co.in
  mosip-pmp-host: pmp.onprem1v3.technoforte.co.in
  mosip-onboarder-host: onboarder.onprem1v3.technoforte.co.in
  mosip-resident-host: resident.onprem1v3.technoforte.co.in
  mosip-compliance-host: compliance.onprem1v3.technoforte.co.in
  mosip-esignet-host: esignet.onprem1v3.technoforte.co.in
  mosip-smtp-host: smtp.onprem1v3.technoforte.co.in
  is_glowroot_env: absent
```

```
kubectl apply -f global_configmap.yaml
kubectl get cm global
```

REGISTER MOSIP CLUSTER WITH RANCHER

Login as admin in Rancher console
Select Import Existing for cluster addition.
Select the Generic as cluster type to add.
Fill the Cluster Name field with unique cluster name and select Create.
You will get the kubectl commands to be executed in the Mosip cluster (WSL)

In rancher go to clusters wait for cluster to get into active state

---

INSTALL LONGHORN PERSISTENCE STORAGE :

INSTALL PREREQUISITES FOR LONGHORN (WSL)

```
cd k8s-infra/mosip/longhorn
./pre_install.sh
```

INSTALL LONGHORN FROM RANCHER DASHBOARD:

open rancher dashboard --> Mosip cluster --> cluster tools --> longhorn --> install
--> next --> under edit options --> Longhorn storage class setting set replicas as (2)
--> under edit YAML option edit the below mentioned keys value as 5 -->

```
guaranteedEngineManagerCPU: 5
guaranteedReplicaManagerCPU: 5
```

INSTALL

SETUP ISTIO FOR SERVICE DISCOVERY AND INGRESS {make sure istio is pre installed in WSL [steps are at the start of Document]}

```
cd k8s-infra/mosip/on-prem/istio/chart/istio-addons
nano values.yaml
```

EDIT THE HOSTS AS SHOWN BELOW:

```
internalHost: apiinternal.onprem1v3.technoforte.co.in
publicHost: api.onprem1v3.technoforte.co.in
```

```
cd k8s-infra/mosip/on-prem/istio
./install.sh
kubectl get svc -n istio-system
```

ON MOSIP NGINX NODE:

NGINX REVERSE PROXY SETUP FOR MOSIP CLUSTER

INSTALL ANSIBLE

```
sudo apt update
sudo apt upgrade -y
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install -y ansible
```

INSTALL PYTHON 3

```
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
sudo apt install python3.8
python3 --version
```

CLONE K8S-INFRA REPOSITORY ON MOSIP NGINX SERVER

git clone -b v1.2.0.1-B3 GitHub - mosip/k8s-infra: Kubernetes infrastructure to deploy MOSIP modules.

SSL CERTIFICATES WITH LETSENCRYPT

```
sudo apt install letsencrypt -y
sudo apt install certbot python3-certbot-nginx -y
```

Generate certificates for your domain name {EXAMPLE: technoforte.idencode.link}

```
sudo certbot certonly --agree-tos --manual --preferred-challenges=dns -d *.sandbox.xyz.net -d sandbox.xyz.net
```

modify the above command with your domain name (as shown below)
sudo certbot certonly --agree-tos --manual --preferred-challenges=dns -d *.technoforte.idencode.link -d technoforte.idencode.link

run command provide (email id , yes, yes)

create TXT RECORD --> {ON ROUTE 53 AWS CONSOLE}
_acme-challenge.technoforte.idencode.link
value: as output shown on cmd

GO TO URL --> DNS Propagation Checker - Global DNS Checker Tool
copy _acme-challenge.technoforte.idencode.link record name and paste in above link --> select TXT and click on search all sholud be green ticked

(wait fro few minutes and then click to continue)

Press Enter to Continue

Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/technoforte.idencode.link/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/technoforte.idencode.link/privkey.pem
  Your cert will expire on 2023-03-15. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot
  again. To non-interactively renew *all* of your certificates, run
  "certbot renew"
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt: Donate
  Donating to EFF: Support EFF's Work on Let's Encrypt

WE WILL SEE THE SUCCESS MESSAGE AS ABOVE

host -t TXT _acme-challenge.technoforte.idencode.link <edit domain name accordingly>

NGINX INSTALLATION ON MOSIP NGINX INSTANCE

cd k8s-infra/rancher/on-prem/nginx/
sudo ./install.sh

=====>
The following internal ip will have to be DNS-mapped to all internal domains from you global_configmap.yaml. Ex: api-internal.sandbox.xyz.net, iam.sandbox.xyz.net, etc.
Give the internal interface ip of this node here. Run `ip a` to get all the interface addresses (without any whitespaces) : 10.2.1.90 (private IP of ngnx)
=====>
This nginx's public ip will have to be DNS-mapped to all public domains from you global_configmap.yaml. Ex: api.sandbox.xyz.net, prereg.sandbox.xyz.net, etc.
The above mentioned nginx's public ip might be different from this nginx machine's public interface ip, if you have provisioned public ip seperately that might be forwarding traffic to this interface ip.
Give the public interface ip of this node here. Run `ip a` to get all the interfaces, In case not exposing api's to public give private ip only. : 10.2.1.90 (private IP of ngnx)

=====>
Give list of (comma seperated) publicly exposing domain names (without any whitespaces). Ex: api.sandbox.xyx.net, prereg.sandbox.xyz.net, resident.sandbox.xyz.net, idp.sandbox.xyz.net etc : api.onpremdev.idencode.link, prereg.onpremdev.idencode.link, resident.onpremdev.idencode.link, esignet.onpremdev.idencode.link
=====>
Give path for SSL Certificate (fullchain.pem) for sandbox.xyz.net (without any whitespaces) : Ex: /etc//letsencrypt/live/sandbox.xyz.net/fullchain.pem/etc/letsencrypt/live/onpremdev.idencode.link/fullchain.pem
=====>
Give path for SSL Certificate Key (privkey.pem) for sandbox.xyz.net (without any whitespaces): Ex: /etc/letsencrypt/live/sandbox.xyz.net/privkey.pem : /etc/letsencrypt/live/onpremdev.idencode.link/privkey.pem
=====>
Give list of (comma seperated) ips of all nodes in the mosip cluster (without any whitespaces) :
10.2.1.84,10.2.1.85,10.2.1.86,10.2.1.87,10.2.1.88,10.2.1.89
=====>
Give nodeport of http port of the mosip cluster public ingressgateway (without any whitespaces) (default is 30080) :
=====>
Give nodeport of http port of the mosip cluster internal ingressgateway (without any whitespaces) (default is 31080) :
=====>
Give nodeport of postgres port of the mosip cluster internal ingressgateway (without any whitespaces) (default is 31432) :
=====>
Give nodeport of activemq port of the mosip cluster internal ingressgateway (without any whitespaces) (default is 31616) :
Reading package lists... Done
Building dependency tree
Reading state information... Done
nginx is already the newest version (1.18.0-0ubuntu1.4).
nginx set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Nginx installed succesfully.

sudo systemctl status nginx   (verify nginx status)

INSTALL HTTPBIN (WSL)

cd k8s-infra/utils/httpbin
./install.sh

curl https://api.onprem1v3.technoforte.co.in/httpbin/get?show_env=true   ---{CHANGE COMMAND AS PER YOUR DOMAIN}
curl https://apiinternal.onprem1v3.technoforte.co.in/httpbin/get?show_env=true ---{CHANGE COMMAND AS PER YOUR DOMAIN}

INSTALL MONITORING THROUGH RANCHER DASHBOARD ON MOSIP CLUSTER

Rancher --> Mosip cluster --> Apps --> Monitoring --> Install --> Next (with default settings) --> Edit yaml (line No. 492) -->INGRESS-NGINX ENABLED (make it): false --> Install
VERIFY THE MONITORING is present on rancher dashboard after installation complete

---

LOGGING

cd k8s-infra/logging/chart/istio-addons   ---> edit kibana domin as per mention in configmaps in values.yaml file

INSTALL ELASTICSEARCH, KIBANA AND ISTIO ADDONS
  cd k8s-infra/logging
  ./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

INSTALL RANCHER FLUENTd SYSTEM

Install Logging from Apps and marketplace within the Rancher UI

Select Chart Version 101.1.3+up3.17.7 from Rancher console -> Apps & Marketplaces.

ADD INDEX LIFECYCLE POLICY AND INDEX TEMPLATE TO ELASTICSEARCH

./elasticsearch-ilm-script.sh

CONFIGURE RANCHER FLUENTD

Use the following command to create elasticsearch ClusterOutput.

kubectl apply -f clusteroutput-elasticsearch.yaml

Use the following command to create mosip-logs ClusterFlow.

kubectl apply -f clusterflow-elasticsearch.yaml

DASHBOARDS SETUP

Run the following to load all dashboards in the ./dashboards folder to Kibana.

./load_kibana_dashboards.sh <path for dashboard folder> <path for cluster-kube-config-file> ---{Example : ./load_kibana_dashboards.sh /home/shiv/Onprem/reporting/dashboards /home/shiv/.kube/onprem_mosip_config}

Give Kibana Host Name (Example: "kibana.sandbox.mosip.net" or "box.mosip.net/kibana"): (default: kibana.b01.idencode.link) --> Enter

Give the installation name (Use "_" instead of "-". And no capitals/symbols.): (default: b01) --> Enter

VIEW DASHBOARD: Kibana (open kibana domain https://kibana.technoforte.idencode.link) --> Menu (on top left) --> Dashboard --> Select the dashboard to see loaded dashboards

---

keep copy of path

config file path : /home/shiv/.kube/e2e_mosip_config

pem key path: /home/shiv/Onprem/id_rsa

INSTALLATION OF EXTERNAL SERVICES (WSL)

Install all the external services as per sequence mention in Install-all.sh file present on path : mosip-infra/deployment/v3/external/all

1.POSTGRES:

cd mosip-infra/deployment/v3/external/postgres/chart/istio-addons --> edit postgres domin as per mention in configmaps in values.yaml file

cd mosip-infra/deployment/v3/external/postgres

./install.sh <config file path of mosip cluster> --> {example: /home/shiv/.kube/onprem_mosip_config}

INITIALIZE DB:

./init_db.sh <config file path of mosip cluster>

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

2.IAM:

cd mosip-infra/deployment/v3/external/iam/chart/istio-addons --> edit keycloakExternalHost domin as per mention in configmaps in values.yaml file

cd mosip-infra/deployment/v3/external/iam

./install.sh <path of kubeconfig file for mosip cluster> --> {example: /home/shiv/.kube/onprem_mosip_config}

KEYCLOAK INIT:

./keycloak_init.sh <config file path of mosip cluster>

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

3. SOFTHSM:

cd mosip-infra/deployment/v3/external/hsm/softhsm/

./install.sh <config file path of mosip cluster>   --> {example: /home/shiv/.kube/onprem_mosip_config}

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

4. MINIO:

cd mosip-infra/deployment/v3/external/object-store/minio/chart/istio-addons --> edit minio domin as per mention in configmaps in values.yaml file

cd mosip-infra/deployment/v3/external/object-store/minio

./install.sh <config file path of mosip cluster>   --> {example: /home/shiv/.kube/onprem_mosip_config}

CREATE SECRETS FOR CONFIG SERVER:

cd mosip-infra/deployment/v3/external/object-store/

./cred.sh <config file path of mosip cluster>

Plesae select the type of object-store to be used:
1: for minio native using helm charts
2: for s3 object store
Please choose the correct option as mentioned above(1/2) 1
Please provide pretext value : <Empty>

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

5. CLAMAV:

cd mosip-infra/deployment/v3/external/antivirus/clamav/

./install.sh <config file path of mosip cluster>   --> {example: /home/shiv/.kube/onprem_mosip_config}

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

6. ACTIVEMQ:

cd mosip-infra/deployment/v3/external/activemq

./install.sh <config file path of mosip cluster>   --> {example: /home/shiv/.kube/onprem_mosip_config}

> NOTE: 1 pod will be in Running state. 1/2
> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

7. KAFKA:

cd mosip-infra/deployment/v3/external/kafka/chart/istio-addons/   --> edit kafka domin as per mention in configmaps in values.yaml file

cd mosip-infra/deployment/v3/external/kafka

./install.sh <config file path of mosip cluster>   --> {example: /home/shiv/.kube/onprem_mosip_config}

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

8. MSG-GATEWAY:

```
cd mosip-infra/deployment/v3/external/msg-gateway
./install.sh <config file path of mosip cluster>          --> {example: /home/shiv/.kube/onprem_mosip_config}
```

Would you like to use mock-smtp (Y/N) [Default: Y] : Y   --> Enter Y

9. LANDING PAGE:

```
cd mosip-infra/deployment/v3/external/landing-page
./install.sh
```

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

10. CAPTCHA:

Pre-Requisites

Create a google recaptcha v2 ("I am not a Robot") from Google Recaptcha Admin.

GO TO URL: <https://www.google.com/recaptcha/about/>

v3adminconsole -->
lable: prereg.onpremb3.idencode.link
reCAPTCHA type: reCAPTCHA v2 --> iam not robot
Domains : prereg.onpremb3.idencode.link
SUBMIT

keep copy of secrete key and site key after generation of Recaptcha

```
site key    : 6Le-jswkAAAAAPbNtnLDp4WD1VhiZfc-b-XvigUB
secrate key : 6Le-jswkAAAAAKEbnyWSQgYMM6I19EF4jCJLzHy4
```

v3adminconsole -->
lable: resident.onpremb3.idencode.link
reCAPTCHA type: reCAPTCHA v2 --> iam not robot
Domains : resident.onpremb3.idencode.link
SUBMIT

keep copy of secrete key and site key after generation of Recaptcha

```
site key    : 6Le-jswkAAAAAPbNtnLDp4WD1VhiZfc-b-XvigUB
secrate key : 6Le-jswkAAAAAKEbnyWSQgYMM6I19EF4jCJLzHy4
```

v3adminconsole -->
lable: esignet.onpremb3.idencode.link
reCAPTCHA type: reCAPTCHA v2 --> iam not robot
Domains : esignet.onpremb3.idencode.link
SUBMIT

keep copy of secrete key and site key after generation of Recaptcha

```
site key    : 6Le-jswkAAAAAPbNtnLDp4WD1VhiZfc-b-XvigUB
secrate key : 6Le-jswkAAAAAKEbnyWSQgYMM6I19EF4jCJLzHy4
```

cd mosip-infra/deployment/v3/mosip/captcha
    ./install.sh [kubeconfig]

Please enter the recaptcha admin site key for domain prereg: <enter carefully as per created keys for particular DOMAIN in above step>
Please enter the recaptcha admin secret key for domain prereg: <enter carefully as per created keys for particular DOMAIN in above step>

Please enter the recaptcha admin site key for domain resident: <enter carefully as per created keys for particular DOMAIN in above step>
Please enter the recaptcha admin secret key for domain resident: <enter carefully as per created keys for particular DOMAIN in above step>

Please enter the recaptcha admin site key for domain esignet: <enter carefully as per created keys for particular DOMAIN in above step>
Please enter the recaptcha admin secret key for domain esignet: <enter carefully as per created keys for particular DOMAIN in above step>

INSTALLATION OF MOSIP SERVICES (WSL)

Install all the mosip services as per sequence mention in Install-all.sh file present on path : mosip-infra/deployment/v3/mosip/all   {Make sure you install Partner-onboarder at the last}

1. CONF-SECRETS:

cd mosip-infra/deployment/v3/mosip/conf-secrets

./install.sh [kubeconfig]

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

MODIFY THE BELOW MENTIONED PROPERTIES BEFORE INSTALLING CONFIG-SERVER IN PARTICULAR CONFIG BRANCH FOR ENVIRONMENT

id-authentication-default.properties :
authrequest.received-time-allowed.seconds=120
authrequest.received-time-adjustment.seconds=120

id-authentication-internal-default.properties :
authrequest.received-time-adjustment.seconds=120

---

2. CONFIG-SERVER:

cd mosip-infra/deployment/v3/mosip/config-server
nano values.yml   {make chnages in git repo}

---> IN VALUES.YML   edit uri : GitHub - technoforte/mosip-config: This repository contains MOSIP configuration templates
                                 edit version: e2e1-B3 in values.yaml

./install.sh

Is conf-secrets module installed?(Y/n) Y
Is values.yaml for config-server chart set correctly as part of Pre-requisites?(Y/n) Y

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

3. ARTIFACTORY:

cd mosip-infra/deployment/v3/mosip/artifactory

./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

4. KEYMANAGER:

cd mosip-infra/deployment/v3/mosip/keymanager

./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

5. WEBSUB:

cd mosip-infra/deployment/v3/mosip/websub

./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

6. KERNEL:

cd mosip-infra/deployment/v3/mosip/kernel

./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

7. MASTERDATA-LOADER:

cd mosip-infra/deployment/v3/mosip/masterdata-loader

./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

8. BIOSDK:

cd mosip-infra/deployment/v3/mosip/biosdk

./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

9. PACKETMANAGER:

cd mosip-infra/deployment/v3/mosip/packetmanager

./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

10. DATASHARE:

cd mosip-infra/deployment/v3/mosip/datashare

./install.sh

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

11. PREREG:

cd mosip-infra/deployment/v3/mosip/prereg
./install.sh

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

12. IDREPO:

cd mosip-infra/deployment/v3/mosip/idrepo
./install.sh

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

13. PMS:

cd mosip-infra/deployment/v3/mosip/pms
./install.sh

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

14. MOCK-ABIS:

cd mosip-infra/deployment/v3/mosip/mock-abis
./install.sh

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

15. MOCK-MV:

cd mosip-infra/deployment/v3/mosip/mock-mv
bash install.sh

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

16. REGPROC:

cd mosip-infra/deployment/v3/mosip/regproc
./install.sh

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

17. ADMIN:

cd mosip-infra/deployment/v3/mosip/admin
./install.sh

> NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

18. IDA:

cd mosip-infra/deployment/v3/mosip/ida
./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

19. PRINT:

cd mosip-infra/deployment/v3/mosip/print
./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

20. RESIDENT:

cd mosip-infra/deployment/v3/mosip/resident
./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

21. REGCLIENT:

cd mosip-infra/deployment/v3/mosip/regclient
./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

22. MOCK-SMTP:

cd mosip-infra/deployment/v3/mosip/mock-smtp
./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

23. MOSIP-FILE-SERVER:

cd mosip-infra/deployment/v3/mosip/mosip-file-server
./install.sh

while deploying mosip-file server need to provide link :

Please Enter MOBILE APP Link publicly accessible APK: <https://api.tf1.idencode.link/.well-known> -->{Edit Domain as per your ENV}
Please Enter MOBILE APP Link privately accessible APK: <https://api.tf1.idencode.link/files/mobileapp> -->{Edit Domain as per your ENV}

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

24. PARTNER-ONBOARDER:

cd mosip-infra/deployment/v3/mosip/partner-onboarder
./install.sh

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

REPORTING

INSTALLATION OF DATA PIPELINE

```
cd reporting/scripts
./install.sh [kube-config-file]
```

Give the installation name (Use "_" instead of "-". And no capitals/symbols.): (default: b01) --> Enter
Give the path to debez sample connector file: (default: ../kafka-connect/debez-sample-conn.api) --> Enter
Give the path to folder containing es connectors: (default: ../kafka-connect/ref_connector_api_calls)   --> Enter

UPLOAD KIBANA DASHBOARDS

```
./load_kibana_dashboards.sh <path for dashboard folder> <path for cluster-kube-config-file>
```

NOTE : AFTER COMPLETE INSTALLATION OF SERVICE VERIFY THE PODS ARE UP AND RUNNING IN RANCHER UNDER PARTICULAR NAMESPACE.

SANITY CHECK

VERIFY IN RANCHER ALL THE PODS ARE UP AND RUNNING UNDER MOSIP CLUSTER

CHECK THE LOGS FOR:
-POSTGRES
-MASTERDATA-LOADER
-ONBOARDER

GO TO MAIN DOMAIN / LANDING-PAGE DOMAIN EXAMPLE: http://technoforte.idencode.link   ----> One by one click on every service present

Below there is host link to access POSTGRES --> Password for the POSTGRES is present in RANCHER --> POSTGRES(Namespace) --> STORAGE --> SECRETS --> postgres-postgres

---

CREATE USER IN ADMIN KEYCLOAK FOR MOSIP REGISTRATION AUTHENTICATION IN POSTMAN:

KEYCLOAK --> ADMINISTRATION USER --> USERS -->ADD USER

username : 110123 (provide as per choice) --> SAVE
credentials : Techno@123 --> Temperory --> OFF --> SAVE

ROLE MAPPING:
{AUTH, AUTH_PARTNER, CENTRAL_ADMIN, Default, default-roles-mosip,
DEVICE_PROVIDER, FTM_PROVIDER, GLOBAL_ADMIN, KEY_MAKER, MASTERDATA_ADMIN
MISP, offline_access, ONLINE_VERIFICATION_PARTNER, PARTNER, PARTNER_ADMIN
PARTNERMANAGER, PMS_ADMIN, PMS_USER, POLICYMANAGER, REGISTRATION_ADMIN
REGISTRATION_OFFICER, REGISTRATION_OPERATOR, REGISTRATION_PROCESSOR
REGISTRATION_SUPERVISOR, uma_authorization, ZONAL_ADMIN}

KEYCLOAK --> ADMINISTRATION USER --> USERS -->ADD USER

username : globaladmin (firstname: globaladmin, last name: globaladmin)--> SAVE credentials : Techno@123 --> Temperory --> OFF --> SAVE

ROLE MAPPING:

[AUTH, AUTH_PARTNER, CENTRAL_ADMIN, Default,
GLOBAL_ADMIN, KEY_MAKER, MASTERDATA_ADMIN,
offline_access, ONLINE_VERIFICATION_PARTNER,
REGISTRATION_ADMIN, REGISTRATION_OFFICER,
REGISTRATION_OPERATOR, REGISTRATION_PROCESSOR,
REGISTRATION_SUPERVISOR, uma_authorization, ZONAL_ADMIN]

MACHINE ADDITION:

GO TO ENV LANDING PAGE URL :{example: technoforte.idencode.link}

OPEN ADMIN--> {globaladmin   Techno@123}---> resources--> machines--> create machine--> fill the details from TPM-->

MACHINE NAME: {from tpm detils}
SERIAL NUMBER: 012345
MAC ADDRESS: {192.168.122.90   any dummy}
IP ADDRESS: {192.168.122.84   any dummy}
MACHINE SPECIFIC ID: {Resident virtual machine}
PUBLIC KEY: {from tpm detils}
SIGN KEY: {from tpm detils}
ADMINISTRATION ZONE: {North}
CENTER NAME: {rural muncipal mansara}

DEVICE CERTIFICATE GENERATION:

CA_CERT_UTILITY.zip
15 May 2023, 11:32 AM

EXTRACT CA_CERT_UTILITY

delete intermediate certificate , root certificate , partner certificate.

RUN create-certs -->

country name (IN)   state (KAR)   city (BLR)   organization name (CA) organization unit name (CA) common name (CA)

country name (IN)   state (KAR)   city (BLR)   organization name (SUBCA)   organization unit name (SUBCA) common name (SUBCA)

-----------for partner------------
country name (IN)   state (KAR)   city (BLR)   organization name (SG1111)   organization unit name (SG1111) common name (SG1111)

[intermediate certificate , root certificate , partner certificate got generated]

REGISTER DEVICE ON MOSIP

IMORT FILE TO POSTMAN   {mosip_registration}

V3-MOSIP_Regi...on.json
15 May 2023, 11:36 AM

STEP 1--> LOGIN TO MOSIP --> {verify url as your domain} send

STEP 2--> PARTNER SELF REGISTRATION

change emailid [SG1111@gmail.com]

change organization name [SG1111]

partner id [SG1111] ----> send

STEP 3--> UPLOAD ROOT_CA {verify url as your domain}

open rootCA cetrificate in notepad++

find and replace \n --> \\n

find and replace \r --> empty

copy the certificate with above modification and paste in UPLOAD ROOT_CA in postman -----> send

STEP 4--> UPLOAD INTERMEDIATE_CA {verify url as your domain}

open intermediate CA cetrificate in notepad++

find and replace \n --> \\n

find and replace \r --> empty

copy the certificate with above modification and paste in UPLOAD INTERMEDIATE_CA in postman -----> send

STEP 5--> UPLOAD_PARTNER {verify url as your domain}

open partner cetrificate in notepad++

find and replace \n --> \\n

find and replace \r --> empty

copy the certificate with above modification and paste in UPLOAD_PARTNER in postman partnerid : SG1111 -----> send

STEP 6--> copy any certificate created in CA_CERT_UTILITY folder and paste outside of it RENAME it as mosip-signed
open mosip-signed in notepad++ remove content from it.
copy the signed certificate from postman Form UPLOAD_PARTNER and paste it in mosip-signed

find and replace \\n --> \n SAVE IT MOVE mosip-signed to CA_CERT_UTILITY folder

STEP 7--> open [create-device-keystore] with notepad++
copy the file path of CA_CERT_UTILITY folder and paste it in create-device-keystore in commented command SAVE IT
RUN create-device-keystore
country name [IN] state [KAR] locality name [BLR] organization name [FACE] organization common name [FACE] common name [FACE]

will get SIGNED-DEVICE certificate

FOR CREATING DEVICE CERTIFICATE

open command promt at the location : c/program files/git/usr/bin

run the below command :-- openssl pkcs12 -export -in D:\MOSIP\CA_CERT_UTILITY\CA_CERT_UTILITY\CA_CERT_UTILITY\signed-Device.crt -inkey D:\MOSIP\CA_CERT_UTILITY\CA_CERT_UTILITY\CA_CERT_UTILITY\Device.key -out D:\MOSIP\CA_CERT_UTILITY\CA_CERT_UTILITY\CA_CERT_UTILITY\Device.p12 -name "Device"

password: mosip

verify password: mosip

Device certificate is generated in CA_CERT_UTILITY folder rename it as **Device_mosip**

---

run the below command again at same location :--   openssl pkcs12 -export -in D:\MOSIP\CA_CERT_UTILITY\CA_CERT_UTILITY\CA_CERT_UTILITY\signed-Device.crt -inkey D:\MOSIP\CA_CERT_UTILITY\CA_CERT_UTILITY\CA_CERT_UTILITY\Device.key -out D:\MOSIP\CA_CERT_UTILITY\CA_CERT_UTILITY\CA_CERT_UTILITY\Device.p12 -name "Device"

password: mosipface

verify password: mosipface

DOWNLOAD MOCKMDS

open URL : GitHub - mosip/mosip-mock-services at develop

CLICK ON CODE --> DOWNLOAD ZIP

mosip-mock-se… lop.zip
15 May 2023, 11:59 AM

EXTRACT mosip-mock-services-develop FILE

OPEN file mosip-mock-services-develop --> MockMDS

EDIT POM file in Notepad++

REMOVE <gpgArguments> FROM line no. 129 to 132 as the below mention :

```
<gpgArguments>
    <arg>--pinentry-mode</arg>
    <arg>loopback</arg>
</gpgArguments>
```

ADD after line 128 i.e., inside <configuration> below line need to be added:

```
<skip>true</skip>
```

OPEN CMD in above MockMDS folder

In CMD run below mentioned command

mvn clean install

NOTE: This will generate Target folder in the MockMds folder

OPEN TARGET FOLDER and EDIT APPLICATION FILE WITH NOTEPAD++

EDIT THE BELOW KEYS VALUES AS (Device and mosipface) As Mentioned below

```
mosip.mock.sbi.file.face.keys.keystorefilename=/Biometric Devices/Face/Keys/Device.p12
mosip.mock.sbi.file.face.keys.keyalias=Device
```

mosip.mock.sbi.file.face.keys.keystorepwd=mosipface

mosip.mock.sbi.file.face.keys.keystorefilename.ftm=/Biometric Devices/Face/Keys/Device.p12
mosip.mock.sbi.file.face.keys.keyalias.ftm=Device
mosip.mock.sbi.file.face.keys.keystorepwd.ftm=mosipface

mosip.mock.sbi.file.finger.slap.keys.keystorefilename=/Biometric Devices/Finger/Slap/Keys/Device.p12
mosip.mock.sbi.file.finger.slap.keys.keyalias=Device
mosip.mock.sbi.file.finger.slap.keys.keystorepwd=mosipface

mosip.mock.sbi.file.finger.slap.keys.keystorefilename.ftm=/Biometric Devices/Finger/Slap/Keys/Device.p12
mosip.mock.sbi.file.finger.slap.keys.keyalias.ftm=Device
mosip.mock.sbi.file.finger.slap.keys.keystorepwd.ftm=mosipface

mosip.mock.sbi.file.finger.single.keys.keystorefilename=/Biometric Devices/Finger/Single/Keys/Device.p12
mosip.mock.sbi.file.finger.single.keys.keyalias=Device
mosip.mock.sbi.file.finger.single.keys.keystorepwd=mosipface

mosip.mock.sbi.file.finger.single.keys.keystorefilename.ftm=/Biometric Devices/Finger/Single/Keys/Device.p12
mosip.mock.sbi.file.finger.single.keys.keyalias.ftm=Device
mosip.mock.sbi.file.finger.single.keys.keystorepwd.ftm=mosipface

mosip.mock.sbi.file.iris.double.keys.keystorefilename=/Biometric Devices/Iris/Double/Keys/Device.p12
mosip.mock.sbi.file.iris.double.keys.keyalias=Device
mosip.mock.sbi.file.iris.double.keys.keystorepwd=mosipface

mosip.mock.sbi.file.iris.double.keys.keystorefilename.ftm=/Biometric Devices/Iris/Double/Keys/Device.p12
mosip.mock.sbi.file.iris.double.keys.keyalias.ftm=Device
mosip.mock.sbi.file.iris.double.keys.keystorepwd.ftm=mosipface

mosip.mock.sbi.file.iris.single.keys.keystorefilename=/Biometric Devices/Iris/Single/Keys/Device.p12
mosip.mock.sbi.file.iris.single.keys.keyalias=Device
mosip.mock.sbi.file.iris.single.keys.keystorepwd=mosipface

mosip.mock.sbi.file.iris.single.keys.keystorefilename.ftm=/Biometric Devices/Iris/Single/Keys/Device.p12
mosip.mock.sbi.file.iris.single.keys.keyalias.ftm=Device
mosip.mock.sbi.file.iris.single.keys.keystorepwd.ftm=mosipface

---

EDIT AS PER ENV/DOMAINS

mosip.auth.server.url=https://api-internal.onpremb3.idencode.link/v1/authmanager/authenticate/clientidsecretkey

mosip.auth.clientid=mosip-regproc-client
mosip.auth.secretkey=BAurcdoUzDbNFyby

mosip.ida.server.url=https://api-internal.onpremb3.idencode.link/idauthentication/v1/internal/getCertificate?applicationId=IDA&referenceId=IDA-FIR

COPY DEVICE CERTIFICATE from CA_CERT_UTILITY folder

inside MOCKMDS --> TARGET -->Biometrics devices

-->face -->Keys ---> paste Device certificate
-->Finger -->Single --> Keys ---> paste Device certificate
-->Finger -->Slab --> Keys ---> paste Device certificate

-->Iris --> Single -->Keys ---> paste Device certificate
-->Iris --> Double -->Keys ---> paste Device certificate

in MOCKMDS --> TARGET   run_reg file

PRE-REGISTRATION:

go to url : {verify url as your domain}

-->pre registartion
-->Email: any
-->Otp : 111111

fill all demographic details --> upload documents --> book appointment --> conformation and print the form

---

LAUNCH REG-CLIENT

PUT PRE-REGISTRATION ID --> FETCH DATA
complete all upadtes--> authenticate and save
click on home ----> Pending Approval--> approve---> authenticate
click on home ----> application upload   ----> save to device / upload

---

VIEW PACKET STATUS:

go to admin url ----> packet status ---> pre registration id --> search

---

CHECK STATUS IN DB:

Databases:
mosip_regprc --> schemas --> regprc --> Tables --> registration_transaction

CREDENTAIL_STORAGE STATUS:

Databases:
mosip_credential --> schemas --> credentails --> Tables --> credential_transaction