MOSIP Deployment Guide

For Private Infrastructure

The purpose of this document is to guide the reader through setting up the MOSIP application on private infrastructure. The guide first explains how to set up the hardware, then configure the hardware, and finally install the application. This guide is based on the official deployment instructions. If you are using AWS, you should instead follow the corresponding deployment guide.

We recommend skimming the official deployment instructions to gain context before following this deployment guide.

Setup Hardware

NOTE: If you're using a Cloudlab/GENI datacenter, follow the instructions in appendix A instead.

1. Setup 13 machines. We have had good experiences with running the console machine as a baremetal node, running 10 xen-vms as the MZ Cluster, and running 2 xen-vms as the DMZ cluster. We provision each physical/virtual machine with 4 cores, 8gb of RAM, and a 120gb SSD. We use *CentOS 7.8* as the operating system on each machine.

Setting up the machine environments for MOSIP

Create a new user on the console machine

2. Connect to the shell of the console machine.

3. Create the mosipuser account
   sudo useradd mosipuser
   sudo passwd mosipuser

4. Add mosipuser to the sudoers
   sudo usermod -aG wheel mosipuser

5. Open the sudoers file using
   sudo visudo
   And append these lines to it to give mosipuser unlimited access and prevent applications to prompt for a password

```
mosipuser ALL=(ALL) ALL
%mosipuser ALL=(ALL) NOPASSWD:ALL
```

Give the user ssh permissions as root to all other machines

Keep running these commands on the console machines:

1. Switch to the mosipuser account using the password you created for it
   su - mosipuser

2. Generate the shh keys (just tap return three times)
   ssh-keygen -t rsa
   And copy the ssh public key to clipboard manually (ctrl+c; **copying using mouse or browser causes issues later on when pasting, so don't do it!**)
   cat .ssh/id_rsa.pub

3. Store the ssh keys in the authorized keys of the console VM and the root of all other VMs:
   a. Run this on the console machine and add the copied ssh public key to the file
      nano .ssh/authorized_keys
      Then change the permissions
      chmod 644 .ssh/authorized_keys

   b. Run this on all other machines and add the copied ssh public key to the file
      sudo nano /root/.ssh/authorized_keys

4. Test if the ssh keys were shared correctly by running the below commands
   ssh mosipuser@console
   ssh root@[all other hosts]

Disable the firewall on all other machines

1. Run the below commands on all other machines to disable their firewall
   sudo systemctl stop firewalld
   sudo systemctl disable firewalld

Installing dependencies and downloading the MOSIP repo

Follow these instructions on the Console VM:

1. Install ansible
   sudo yum install ansible

2. Clone the mosip-infra repo
   git clone https://github.com/mosip/mosip-infra.git

3. Change ownership of the cloned repo (if not the owner)
   sudo chown -R mosipuser mosip-infra/

Configuring and Installing MOSIP

Follow these instructions on the Console VM:

1. Print and copy the hostname using:
   hostname

2. Open group_vars/all.yml using
   nano mosip-infra/deployment/sandbox-v2/group_vars/all.yml
   And replace the value of sandbox_domain_name found in the file with your hostname.

3. Add the user nfsnobody which is required later in the installation
   sudo useradd nfsnobody

4. Open both the files below
   nano mosip-infra/deployment/sandbox-v2/group_vars/mzcluster.yml
   nano mosip-infra/deployment/sandbox-v2/group_vars/dmzcluster.yml
   And replace the value of network_interface found in both files with "eth1".

5. Run the ansible scripts that will install MOSIP
   cd mosip-infra/deployment/sandbox-v2/
   ansible-playbook -i hosts.ini site.yml

6. The username if asked for some dashboards is "admin". Tokens for the dashboards can be found at: /tmp/mosip

7. To edit anything related to the MOSIP infrastructure after installation, modify variables and files at /srv/nfs/mosip/config_repo

8. To access the PSQL database use the following command
   psql -h mzworker0 -p 30090 -U postgres
   The password to access the database is "postgres"
   You can also use pgadmin to display the database

9. To access the Pre-Registration UI use the below link:

[Your console hostname]/pre-registration-ui
While testing:

a. You can connect to it using OTP and the static OTP value: 111111
b. You can use this fake valid postal code: 14022

Appendix A: Cloudlab

Cloudlab is NSF infrastructure widely used by the researchers in America. In this section, we include some Cloudlab-specific setup instructions.T o run MOSIP on Cloudlab, we must first request the necessary physical/virtual machines. We do this by starting an experiment:

1. Create a Profile on CloudLab preferably the one provided in the "MOSIP Profile genlib Script" text file (found in "cloudlab_specific" subdirectory)
   Note: If you're creating your own profile based on the MOSIP recommendations, make sure not to use their naming schema that uses ".sb" at the end of the machine hosts. If the machine is called "console.sb" for instance, use "console" instead. (CloudLab does not support the dot in hosts)

2. Start an Experiment with the above generated Profile. (Don't forget to expand the Experiment duration if needed for a longer amount of time)

3. Connect to the shell of the "console" machine and enable SELinux. To do so follow this blog - https://phoenixnap.com/kb/enable-selinux-centos. (If you face any issues with the machine taking a long time to go back online, wait for it or create a new experiment)

4. For the next steps, you can either run the commands specified in this document **or** use the script provided in the "MOSIP Installation Script" text file (found in "cloudlab_specific" subdirectory). **Make sure to execute the steps mentioned in the file before executing the script!**

Appendix B: Common Problems

- **Issue #1: Hosts not found / Could not connect to host**
  Deployment failed due to .sb suffix in files

  Solution:
  Note: This should be done after running the installation script. The script removed .sb suffixes from other file paths.

Remove the .sb suffix in roles/nginx/templates/nginx_https_conf.yml

- **Issue #2: Could not connect to kubernetes cluster on xxx machine**
  Need to change the default network interface from eth0 to another interface.

  Solution:
  Change the interface in group_vars/k8s.yml file.

- **Issue #3: IP address in kubernetes cluster and script are different (e.g. public IP in Kubernetes cluster vs private IP in the script); or different Ethernet IPs between k8s cluster and script**
  Script changes not reflected in the Kubernetes cluster.
  Sometimes the changes are not reflected in the cluster, if you have modified the yaml files.

  Solution:
  Run the "reset" script provided by MOSIP, or recreate the underlying infrastructure (hard reboot).

- **(Cloudlab only) Issue #4: After successfully running Ansible script, you are unable to curl or access the dashboards from within the deployment.**
  Some cloudlab datacenters block accessing websites created locally.

  Solution:
  Call CloudLab and ask them to give your VMs access to the dashboard

Appendix C: Debugging Tips

If the ansible script fails, re-run the ansible script with -vvv for verbose debug logging to figure out the issue.