## Setup Details

### Machine Details

| Cloud VM name | Cloud | Public IP | Private IP | Spec | username |
| --- | --- | --- | --- | --- | --- |
| ap-south-1-test-smpc-server-1 | AWS | **13.200.145.195**  (old 13.232.226.177) | 10.2.1.25 | **2vcpu, 4GiB, 30 GiB SSD** | ubuntu |
| ap-south-1-test-smpc-server-helpernode | AWS | 15.206.80.191 | 10.2.1.171 | 1vcpu, 0.5GiB, 30 GiB SSD | ubuntu |
| smpc-server0 | Azure | 20.197.27.117 | 10.139.0.4 | **2vcpu, 4 GiB, 30 GiB SSD** | ubuntu |

### Reverse Tunnelling

#### Setup

* created a user called “sshtunuser” with no shell login access in the following machine, i.e. can only do reverse tunnelling

| Cloud VM name | Cloud | Public IP | Private IP | Spec | username |
| --- | --- | --- | --- | --- | --- |
| analytics-ui | Azure | 52.172.238.250 | 10.139.0.11 | 1vcpu, 1GiB, 30 GiB HDD | sshtunuser (use this to give access to ssh reverse tunnel only)  ubuntu(sudo) |

### 

* configured sshd config as described in the article to allow reverse tunnel . see [setting up environment](https://cnly.github.io/2018/08/16/setting-up-autossh-to-maintain-a-reverse-tunnel-ssh-server-having-a-dynamic-ip-address.html)

#### Add user for reverse tunnelling

* You can add users by following way
  + Ask the user to generate public-private key using following instruction

| 1. ssh-keygen -b 4096 -f ~/.ssh/iudx\_cloud 2. passphrase refers to encrypting a private key using a passphrase based key. For convenience, can be ignored and press enter, i.e. without passphrase and hence key would be unencrypted 3. send the public key to add access to VMs, the file with ‘.pub’ extension, i.e the file at ~/.ssh/iudx\_cloud.pub 4. **DO NOT SEND the private key** |
| --- |

* + Login as ubuntu (sudo) user to reverse tunnelling machine

| > ssh ubuntu@52.172.238.250 |
| --- |

* + Add the public key in /home/sshtunuser/.ssh/authorized\_keys

| > sudo vim /home/sshtunuser/.ssh/authorized\_keys |
| --- |

#### Remove User for reverse tunnelling

* Login as ubuntu (sudo) user to reverse tunnelling machine

| > ssh ubuntu@52.172.238.250 |
| --- |

* Removal of users by following way, whenever someone leaves the organisation
  + Remove that user public key in the file /home/sstunuser/.ssh/authorized\_keys

| > sudo vim /home/sstunuser/.ssh/authorized\_keys |
| --- |

## Maintenance

### Issues

Any issue related to machines please document here.

### Upgrade

#### OS and packages upgrade

* Do OS and packages upgrade and auto removal of packages monthly for all machines, Preferably on the first tuesday of the month.
* Purpose: packages, os updates . This would include security fixes and frequent upgrade would the keep the machines secure
* Before proceeding with the upgrade, get a list of what is getting updated.
  + Specifically make sure newer docker-ce has no breaking changes
* Upgrade using following commands

| > sudo apt update  > sudo apt upgrade  > sudo apt autoremove # if restart required after upgrade > sudo reboot |
| --- |

### Users Management

#### Cloud VMs

##### Add user

* You can add users by following way
  + Ask the user to generate public-private key using following instruction

| 1. ssh-keygen -b 4096 -f ~/.ssh/iudx\_cloud 2. passphrase refers to encrypting a private key using a passphrase based key. For convenience, can be ignored and press enter, i.e. without passphrase and hence key would be unencrypted 3. send the public key to add access to VMs, the file with ‘.pub’ extension, i.e the file at ~/.ssh/iudx\_cloud.pub 4. **DO NOT SEND the private key** |
| --- |

* + Login as ubuntu (sudo) user to the machine you want to add access to

| > ssh ubuntu@<public-ip> |
| --- |

* + Add the public key in /home/<username>/.ssh/authorized\_keys

| > sudo vim /home/ubuntu/.ssh/authorized\_keys |
| --- |

* Note: all the new users are added with ubuntu username in all vms except reverse tunnel machine. There the user is added as sshtunuser

##### Remove User

* Login as ubuntu (sudo) user to machines, you want to remove a user access to

| > ssh ubuntu@<public-ip> |
| --- |

* Removal of users by following way, whenever someone leaves the organisation
  + Remove that user public key in /home/<username>/.ssh/authorized\_keys

| > sudo vim /home/<username>/.ssh/authorized\_keys |
| --- |

* Note: all the new users are added with ubuntu username in all vms except reverse tunnel machine. There the user is added as sshtunuser

#### Github

##### Add collaborator to github repo for write access

* By default, since its public repo, everyone has read access. To add write access, follow instructions
* Goto motion2nx settings page -> collaborators and teams: <https://github.com/datakaveri/iudx-MOTION2NX/settings/access>
* In manage access section
  + Invite people with their github username
  + Add role as write

##### Remove collaborator to github repo

* Goto motion2nx settings page -> collaborators and teams: <https://github.com/datakaveri/iudx-MOTION2NX/settings/access>
* In manage access section
  + Remove that particular user

##### Add user to push docker image to registry

* Goto this Motion2nx package settings <https://github.com/orgs/datakaveri/packages/container/motion2nx/settings>
* In manage access section
  + Invite people with their github username
  + Add role as write → to be able to push to docker registry
* Send following instructions to be able to login ghcr docker registry and push docker image . ref: <https://docs.github.com/en/packages/working-with-a-github-packages-registry/working-with-the-container-registry#authenticating-with-a-personal-access-token-classic>

##### Remove user from ghcr registry

* Removal of users by following way, whenever someone leaves the organisation
* Goto this Motion2nx package settings <https://github.com/orgs/datakaveri/packages/container/motion2nx/settings>
* In manage access section
  + Remove that particular user