**A Guide to Deploying ElfStore on Reigel using VIoLET**

Credits:

<https://github.com/shriramrseee>

<https://github.com/Sheshadri-1992>

NOTE:

\* Run all the python scripts as sudo.

\* The username “dreamlab” is hard coded.

**PART 1: Setting up Violet, the VMs and building the ElfStore image.**

1. One VM should be Admin VM and others will be Container VMs

2. Generate ssh key in Admin VM using ssh-keygen and copy it to all VMs using **ssh-copy-id** command (VIoLET connects with VMs only using ssh keys)

2.1 After copying the keys, sudo ssh into all the vms from the admin vm. This should add the entries in the known hosts file. The future ssh logins (even via the root user) should be password less.

3. Add nopasswd access to dreamlab on all VMs using visudo command (<https://phpraxis.wordpress.com/2016/09/27/enable-sudo-without-password-in-ubuntudebian/>)

4. Change hostname in all VMs to vm-1, vm-2, vm-3 etc.

4.1 For the vms deployed in reigel ‘preserve\_hostname’ is set to false by default. Set it to true in /etc*/*cloud/cloud.cfg so that after reboot the hostname is retained.

4.2 Use **hostnamectl set-hostname <new\_name>** (should be unique) (might need to be scripted since hostname changes after VM restart)

5. Clone VIoLET git repository on the admin VM in /home/dreamlab (<https://github.com/dream-lab/VIoLET>)

6. Install Docker CE on all vms by executing the docker\_install\_ubuntu.sh script. (remove stop docker)

7. Clone the ElfstoreDocker repo in /home/dreamlab. (https://github.com/ishanSharma07/ElfStoreDocker)

8. Build the ”elfstorenojson” image, on the admin vm. (you can change the tag if you like)

8.1 Build the ”centoselfstorebase” image.

cd /home/dreamlab/ElfStoreDocker/Images/centoselfstorebase

sudo docker build -t centoselfstorebase .

8.2 Build the elfstorenojson image.

cd /home/dreamlab/ElfstoreDocker/

sudo docker build -t elfstorenojson .

9. Save the image.

sudo docker save -o /hdd/elfsImage.tar elfstorenojson &

10. scp the image and load it in all the container vms. (refer C3)

11. Install metis on Admin VM (<http://glaros.dtc.umn.edu/gkhome/metis/metis/download>)

\* The cmake (version 3.x, via apt install cmake) works fine.

12. Install and run Coremark on AdminVM. (Coremark for 6 core VM is 68352) (command is available in VIoLET README) (optional)

13. Edit the files specific to VIoLET.

13.1 Edit vm\_types.json (config for reigel included)

13.2 Edit vm\_config.json

13.3 Edit device\_types.json (image for the containers is specified here)

13.4 Edit infra\_gen.json file (Specify correct no. and type of devices)

14. Run infra\_gen.py

15. Run metis partitioning script (Refer VIoLET README in GitHub for clear instructions)

\* Note if metis\_check.py fails for your vm setup, then try rerunning gpmetis again. If metis\_check fails again then consider downgrading to less resource intensive device types (say, use Pi2B instead of Pi3B).

16. Make changes to the docker run command.

16.1 Remove --storage-opt from docker command to run container in infra\_setup.py (line 176) (because only xfs is supported)

16.2 Add an additional volume bind argument. (-v /home/dreamlab/Logs:/edgefs/logs/)

\* The above folder only contains the edge client logs. The fog and edge server logs are fetched from the containers separately using scripts. (elaborated later)

17. Enable cgroup swap limit capabilities [https://docs.docker.com/install/linux/linux-postinstall/#your-kernel-does-not-support-cgroup-swap-limit-capabilities](https://docs.docker.com/install/linux/linux-postinstall/" \l "your-kernel-does-not-support-cgroup-swap-limit-capabilities)

18. Run start\_docker.py

\* check tail nohup.out for logs. (wait for some time (10 secs) for the container vms to register themselves to the consul service, before executing the next command.)

\* if the error encountered is related to port 2375 is already in use, then, execute the C1 on admin vm. (The ip address list should not include the admin vm ip in the following commands.)

\* If C1 does not work then restart all the container vms. (in most of the cases you will need to restart the vms in order to resolve the above error)

\* If the error encountered is that docker is still running, then execute C2.

19. If you are redeploying VioLET then run delete\_infra.py. Just in case.

20. Run infra\_setup.py

21. Run python sanity\_network.py pvt violet\_private\_1 & python sanity\_network.py pvt violet\_private\_2 & python sanity\_network.py pub violet\_public\_1 (Edit command based on the networks that are created)

22. Before a redeployment, manually stop and delete the Consul container in the admin Vm and restart all the Vms

**PART 2: Deploying Fog and Edge Servers.**

1. Copy VioLET/dump/infra/deployment\_output.json to ElfStoreDocker/deployment/

2. Create the cluster.conf file manually. Example for a D20 config is present in ElfStoreDocker/deployment/.

\* For the fogs use the IP specified in “public\_global\_network” only. This is necessary so that the edges can communicate with any of the available fogs.

3. Run fog\_deployment.py

4. Run edge\_deployment.py

\* Note: If the container name is ‘Edge-1.4’ then the corresponding edge id is 14. This is done only for easy identification of the corresponding edge config file. Consider editing this in case of larger deployments where conflicts may occur. Like ‘Edge-12.4’ and ‘Edge-1.24’ would have the same edge id.

**PART 3: Fetching Logs**

1. Run get\_client\_logs.py (Stored in ~/EdgeClientLogs)

2. Run get\_server\_logs.py (Stored in ~/FogServerLogs and ~/EdgeServerLogs)

**Command Index**

C1

for node in 69 51 52 53 54 55

do

sudo ssh -i ~/.ssh/id\_rsa dreamlab@192.168.0.$node sudo lsof -ti:2375 | xargs -I {} kill {} -9

sudo ssh -i ~/.ssh/id\_rsa dreamlab@192.168.0.$node sudo ps axf | grep docker | grep -v grep | awk '{print "kill -9 " $1}' | sudo sh

done

C2

for node in 69 51 52 53 54 55

do

sudo ssh -i ~/.ssh/id\_rsa dreamlab@192.168.0.$node sudo systemctl stop docker

sudo ssh -i ~/.ssh/id\_rsa dreamlab@192.168.0.$node sudo rm /var/run/docker.pid

done

C3

*Part 1*

for node in 69 51 52 53 54 55

do

sudo scp -i /home/dreamlab/.ssh/id\_rsa /hdd/elfsImage.tar

dreamlab@192.168.0.$node:/home/dreamlab/ &

done

*Part 2 (the following has to be executed manually on each container vm. It doesn't seem to finish execution via bash scripting)*

sudo docker load -i elfsImage.tar

*Part 3 (remove the tar file)*

for node in 69 51 52 53 54 55

do

sudo ssh -i ~/.ssh/id\_rsa dreamlab@192.168.0.$node sudo rm elfsImage.tar &

done