

## gTAF Test Automation Setup (Ubuntu Platform)

### ➤ System dependencies:

- python2.7 or python3.4+
- (<https://linuxize.com/post/how-to-install-python-3-7-on-ubuntu-18-04/>)
- python-setuptools (<https://www.saltycrane.com/blog/2010/02/how-install-pip-ubuntu/>)
- Android adb drivers and set the adb path to PATH variable
- JDK (<https://www.linode.com/docs/development/java/install-java-on-ubuntu-18-04/>)
- Download **sdk-tools-linux.zip** (<https://developer.android.com/studio>)
- Unzip **sdk-tools-linux.zip** and navigate to tools/bin
- Execute below command from tools/bin/
  1. `touch ~/.android/repositories.cfg`
  2. `./sdkmanager --update`
  3. `./sdkmanager --licenses`
- `sudo pip install PyYAML==3.13`
- `sudo pip install opencv-python`
- `sudo apt-get install gradle`
- `sudo pip install opencv-python`
- `sudo apt install tesseract-ocr`
- `sudo pip install pytesseract`
- `sudo pip install tesseract`
- `sudo pip install pillow`
- `sudo pip install mysqlclient`
- `sudo apt-get install python-mysqldb`
- `sudo pip install pymysql`
- `sudo pip install psutil`
- Gmail id should be configured in test device before execution

Once above installation completed, follow below steps to setup and execute gTAF test automation

- If 'ex4\_uiautomator-debug.apk' file is available already then install it to device and skip section (A).

### (A) Build apk from UiAutomator code snippet and install to device:

- ✓ Download gTAF, extract gTAF and give permission '`sudo chmod 777 -R gTAF`'
- ✓ Go to mobly-snippet-lib/
- ✓ Update sdk.dir value with "`../../sdk-tools-linux`" in local.properties file
- ✓ Execute below command to build the apk,

```
./gradlew examples:ex4_uiautomator:assembleDebug
```

- ✓ Connect android device and check using 'adb devices' command, device id should be display
- ✓ Execute to push the apk file to android devices-

```
adb install -r ./examples/ex4_uiautomator/build/outputs/apk/debug/ex4_uiautomator-debug.apk
```

Step (A) can be skip if only want to execute existing test cases by installing “ex4\_uiautomator-debug.apk” file to device direct with below command-

```
adb install -r ex4_uiautomator-debug.apk
```

### (B) Execute gTAF test case on single device:

- ✓ Go to ~gTAF/mobly/
- ✓ sudo python setup.py install
- ✓ edit **gtaf\_config.yml** file and update android device's serial number as given below.

```
TestBeds:
- Name: GTAF_TEST_BED_CONTACT
  Controllers:
    AndroidDevice:
      - serial: FA6BA0300749
        phone_number: 123456
```

- ✓ Edit **gTAF\_config.py** file and update dev\_type ‘**mobile**’ and path as given below.

```
dev_type = "mobile"
gTAF_home = "/home/edat/Desktop/mobly_setup/mobly_test/mobly/"

execution_summary_csv_file = "/gtaflog/execution_summary.csv"

html_report_file = "/gtaflog/gTAF_report.html"
log_path = "/gtaflog"
sql_db_host = 'localhost'
sql_db_name = 'gtafreports'
sql_db_table_name = 'GTAFREPORTS'
map_test_device_id = "FA6BA0300749"
YT_TEST_DEVICE_SERIAL = "FA6BA0300749"
YT_BT_TARGET_SERIAL = "FA6A40300536"
YT_BT_DISCOVERER_SERIAL = "FA6BA0300749"
#CONTACT_APP_TEST_DEVICE_SERIAL = "FA6BA0300749"
GMAP_AUDIO_PLAYER_DEVICE_SERIAL = "FA6A40300536"
GMAP_SEARCH_AUDIO_FILE = "map_search_HCL.wav"
```

### (C) gTAF execution for multiple devices in parallel :

- ✓ Go to ~gTAF/mobly/
- ✓ sudo python setup.py install
- ✓ edit **gtaf\_config.yml** file and update android device's serial numbers against both test beds as mentioned below.

```
TestBeds:
- Name: GTAF_TEST_BED_CONTACT_device1
  Controllers:
    AndroidDevice:
      - serial: serial1
        phone_number: 123456
- Name: GTAF_TEST_BED_CONTACT_device2
  Controllers:
    AndroidDevice:
      - serial: serial2
        phone_number: 123456
```

- ✓ edit **gTAF\_config.py** file and update dev\_type 'mobile' and path as given below.

```
dev_type = "mobile"
gTAF_home = "/home/edat/Desktop/gTAF/mobly/"

execution_summary_csv_file = "/gtaflog/execution_summary.csv"

html_report_file = "/gtaflog/gTAF_report.html"
log_path = "/gtaflog"
sql_db_host = 'localhost'
sql_db_name = 'gtafreports'
sql_db_table_name = 'GTAFREPORTS'
map_test_device_id = "FA6BA0300749"
YT_TEST_DEVICE_SERIAL = "FA6BA0300749"
YT_BT_TARGET_SERIAL = "FA6A40300536"
YT_BT_DISCOVERER_SERIAL = "FA6BA0300749"
#CONTACT_APP_TEST_DEVICE_SERIAL = "FA6BA0300749"
GMAP_AUDIO_PLAYER_DEVICE_SERIAL = "FA6A40300536"
GMAP_SEARCH_AUDIO_FILE = "map_search_HCL.wav"
```

- Execute below commands from two separate terminals to execute same test case on both devices

**Command-1:**

```
sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_merge_duplicates
--test_bed GTAF_TEST_BED_CONTACT_device1
```

**Command-2:**

```
sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_merge_duplicates
--test_bed GTAF_TEST_BED_CONTACT_device2
```

**Jenkins commands to run contact's test suite on above both devices:**

- Create two separate jobs on jenkins and copy below commands to execute suite

**For Job1 [Name: test\_contacts\_suite\_1], jenkins's shell commands are given below:**

[To create Job on jenkins and jenkins setup, please refer section '**Jenkins CI Setup on Ubuntu**']

```
set +e
sudo chmod 777 -R /tmp/logs/mobly
sudo chmod 777 -R /gtaflog
cd /home/edat/Desktop/gTAF/mobly
sudo rm -rf jenkins_logs/*
sudo rm /gtaflog/execution_summary.csv
set +e
sudo python add_csv_header.py

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_merge_duplicates --test_bed GTAF_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAF_TEST_BED_CONTACT_device1 test_merge_duplicates

sleep 5
set +e
```

```
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_favourite_in_all_contact --test_bed GTAF_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_favourite_in_all_contact

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_google_play_service_permission_revoke --test_bed
GTAf_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_google_play_service_permission_revoke

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_share_contacts --test_bed GTAF_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_share_contacts

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_create_contact_name --test_bed GTAF_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_create_contact_name

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_create_contact_all_fields --test_bed GTAF_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_create_contact_all_fields

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_remove_favorite_contact --test_bed GTAF_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_remove_favorite_contact

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_contact_permission_revoke_from_gmail --test_bed
GTAf_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_contact_permission_revoke_from_gmail

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_delete_contact --test_bed GTAF_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_delete_contact

sleep 5
```

```

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_contact_permission_grant_from_gmail --test_bed
GTAf_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_contact_permission_grant_from_gmail

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_create_label --test_bed GTAf_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_create_label

sleep 5

sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_suggestion_welcome_card --test_bed GTAf_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_suggestion_welcome_card

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_contact_help_feedback --test_bed GTAf_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_contact_help_feedback

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_search_favourite_on_top --test_bed GTAf_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_search_favourite_on_top

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_strong_match_for_duplicates --test_bed GTAf_TEST_BED_CONTACT_device1
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device1 test_strong_match_for_duplicates
sudo python send_email.py
#sudo python summary.py
#sudo python send_report_email.py ${BUILD_NUMBER}

```

**For Job2 [Name: test\_contacts\_suite\_2], jenkins's shell commands are given below:**

```

set +e
sudo chmod 777 -R /tmp/logs/mobly
sudo chmod 777 -R /gtaflog
cd /home/edat/Desktop/gTAF/mobly
sudo rm -rf jenkins_logs/*
sudo rm /gtaflog/execution_summary.csv
set +e
sudo python add_csv_header.py

sleep 5

```

```
set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_merge_duplicates --test_bed GTAf_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_merge_duplicates

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_favourite_in_all_contact --test_bed GTAf_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_favourite_in_all_contact

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_google_play_service_permission_revoke --test_bed
GTAf_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_google_play_service_permission_revoke

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_share_contacts --test_bed GTAf_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_share_contacts

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_create_contact_name --test_bed GTAf_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_create_contact_name

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_create_contact_all_fields --test_bed GTAf_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_create_contact_all_fields

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_remove_favorite_contact --test_bed GTAf_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_remove_favorite_contact

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_contact_permission_revoke_from_gmail --test_bed
GTAf_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_contact_permission_revoke_from_gmail

sleep 5
```

```

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_delete_contact --test_bed GTAF_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_delete_contact

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_contact_permission_grant_from_gmail --test_bed
GTAf_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_contact_permission_grant_from_gmail

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_create_label --test_bed GTAF_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_create_label

sleep 5

sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_suggestion_welcome_card --test_bed GTAF_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_suggestion_welcome_card

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_contact_help_feedback --test_bed GTAF_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_contact_help_feedback

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_search_favourite_on_top --test_bed GTAF_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_search_favourite_on_top

sleep 5

set +e
sudo python test_contacts_app.py -c gtaf_config.yml --test_case
test_strong_match_for_duplicates --test_bed GTAF_TEST_BED_CONTACT_device2
sudo python test_report.py ${BUILD_NUMBER} ${JOB_NAME}
GTAf_TEST_BED_CONTACT_device2 test_strong_match_for_duplicates
sudo python send_email.py
#sudo python summary.py
#sudo python send_report_email.py ${BUILD_NUMBER}

```

Change 'cd /home/edat/Desktop/gTAF/mobly' command in above section with correct path.

**(D) Execute below commands to run respective Google map test case:**

- **Accessibility test run:**

```
python test_google_map.py -c gtaf_config.yml --test_case test_accessibility_talkback_settings --test_bed GTAF_TEST_BED_GMAP
```

- **Location services disabled:**

```
python test_google_map.py -c gtaf_config.yml --test_case test_gmap_location_services_disabled --test_bed GTAF_TEST_BED_GMAP
```

- **Multiway point route test run:**

```
python test_google_map.py -c gtaf_config.yml --test_case test_multiway_point_route --test_bed GTAF_TEST_BED_GMAP
```

- **Search bar test run:**

```
python test_google_map.py -c gtaf_config.yml --test_case test_search_bar_autocomplete --test_bed GTAF_TEST_BED_GMAP
```

- **Tap on current location test run:**

```
python gmap_test.py -c gtaf_config.yml --test_case test_tapping_current_place_button --test_bed GTAF_TEST_BED_GMAP
```

### **Execute below commands to run Youtube app test case:**

- ✓ edit **gtaf\_config.yml** file and update android device's serial number against 'GTAf\_TEST\_BED\_YOUTUBE' test bed.

```
python test_youtube_app.py -c gtaf_config.yml --test_case test_caption --test_bed GTAF_TEST_BED_YOUTUBE
```

```
python test_youtube_app.py -c gtaf_config.yml --test_case test_Settings_About_Items --test_bed GTAF_TEST_BED_YOUTUBE
```

```
python test_youtube_app.py -c gtaf_config.yml --test_case test_Video_Not_Available --test_bed GTAF_TEST_BED_YOUTUBE
```

```
python test_youtube_app.py -c gtaf_config.yml --test_case test_YouTube_Play_Pause --test_bed GTAF_TEST_BED_YOUTUBE
```

### **Execute below commands to run Contacts app test case:**

#### **→ Run below shell script to run all contacts test cases-**

Copy 'gtaf\_contacts\_suite.sh' to mobly/ directory

Give permission 'sudo chmod 777 gtaf\_contacts\_suite.sh'

Execute by 'sudo ./gtaf\_contacts\_suite.sh | tee out.log'

#### **→ Below commands are to execute individual test case-**

- `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_merge_duplicates --test_bed GTAF_TEST_BED_CONTACT`
- `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_favourite_in_all_contact --test_bed GTAF_TEST_BED_CONTACT`
- `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_google_play_service_permission_revoke --test_bed GTAF_TEST_BED_CONTACT`
- `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_share_contacts --test_bed GTAF_TEST_BED_CONTACT`
- `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_create_contact_name --test_bed GTAF_TEST_BED_CONTACT`



- `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_create_contact_all_fields --test_bed GTAF_TEST_BED_CONTACT`
  - `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_remove_favorite_contact --test_bed GTAF_TEST_BED_CONTACT`
  - `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_contact_permission_revoke_from_gmail --test_bed GTAF_TEST_BED_CONTACT`
  - `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_delete_contact --test_bed GTAF_TEST_BED_CONTACT`
  - `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_contact_permission_grant_from_gmail --test_bed GTAF_TEST_BED_CONTACT`
  - `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_create_label --test_bed GTAF_TEST_BED_CONTACT`
  - `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_suggestion_welcome_card --test_bed GTAF_TEST_BED_CONTACT`
  - `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_contact_help_feedback --test_bed GTAF_TEST_BED_CONTACT`
  - `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_search_favourite_on_top --test_bed GTAF_TEST_BED_CONTACT`
  - `sudo python test_contacts_app.py -c gtaf_config.yml --test_case test_strong_match_for_duplicates --test_bed GTAF_TEST_BED_CONTACT`
- Above test cases have been executed successfully on below devices with Contacts(v.3.6.8.256022923)]
1. Pixel : PQ1A.190105.004), Android-v9
  2. Pixel 2 : Android-v10
  3. Pixel 3 : Android-v9
  4. Samsung Galaxy S7[SM-G930F]: R16NW.G930FXXS5ESF3, Android-v8.0.0

### **Report Server Stop/Start Process:**

# Stop Report Service

`sudo docker container stop visualization3`

`sudo docker container stop gtafmysql`

# Start Report Service

`sudo docker container start gtafmysql`

`sudo docker container start visualization3`

# Execute below step whenever you are start or restart "visualization3"

`cd /home/hclauto/Downloads/`

`sudo ./elkupdate.sh`

# Validate Report Services

# `sudo docker container ls -a`

CONTAINER ID PORTS	IMAGE	COMMAND	CREATED	STATUS NAMES
1651283f4587 minutes 0.0.0.0:80->80/tcp, 0.0.0.0:5044->5044/tcp, 0.0.0.0:5601->5601/tcp, 0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp	mobly/visualization:3.0	"/usr/local/bin/star..."	1 minutes ago	Up 1
5fafa67a744b 0.0.0.0:3306->3306/tcp, 33060/tcp	mysql	"docker-entrypoint.s..." gtafmysql	2 minutes ago	Up 2 minutes

# Make sure below URL are working before using report dashboard (if not please validate Report Services)

# http://localhost:9200

# http://localhost/hello.html

Report URL

# http://localhost:5601/app/kibana

# Go to dashboard > gtafreportsdashboard

### **Execute below commands to run web test case:**

python test\_selenium.py

### **Jenkins CI Setup on Ubuntu:**

Execute below command on terminal-

1. `wget -q -O - https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo apt-key add -`
2. `echo deb https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list`
3. `sudo apt-get update`
4. `sudo apt-get install jenkins`
5. `sudo systemctl start jenkins`
6. `sudo systemctl status jenkins`

Output :

jenkins.service - LSB: Start Jenkins at boot time

Loaded: loaded (/etc/init.d/jenkins; bad; vendor preset: enabled)

Active:active (exited) since Thu 2017-04-20 16:51:13 UTC; 2min 7s ago

Docs: man:systemd-sysv-generator(8)

7. `sudo ufw allow 8080`

8. `sudo ufw status`

Output:

Status: active

To

Action

From

--	-----	----
OpenSSH	ALLOW	Anywhere
8080	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
8080 (v6)	ALLOW	Anywhere (v6)

Note: If the firewall is inactive, the following commands will make sure that OpenSSH is allowed and then enable it.

9. `sudo ufw allow OpenSSH`

10. `sudo ufw enable`

copy password using command : `sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

- ➔ Open `http://localhost:8080` in browser
- ➔ copy the 32-character alphanumeric password from the terminal and paste it into the "Administrator password" field.
- ➔ Click "Continue". The next screen present the option of installing suggested plugins
- ➔ Once jenkins is up, click on New Item to create a new Job
- ➔ Enter name and select Freestyle project, click ok
- ➔ In Build section, choose Add build step to 'Execute shell' and write below commands in Command section:
- ➔ Apply and save
- ➔ Click on 'Build Now' option, it will trigger execution from Jenkins
- ➔ Execution summary will be created at :
  - `/gtaflog/execution_summary.csv`
  - `/gtaflog/gTAF_report.html`
- ➔ Set ANDROID\_HOME system variable in jenkins-
  - Click on Manage Jenkins
  - Configure System
  - Go to Global properties section
  - Select Environment variables
  - Name : ANDROID\_HOME
  - Value : `../../sdk-tools-linux/platform-tools/`
  - Apply & save
- ➔ Add jenkins as root users:
  - add below line in `/etc/sudoers` file,
  - jenkins ALL=(ALL) NOPASSWD: ALL**
- ➔ copy adb binary to `/usr/local/sbin` and execute '`sudo chmod 777 /usr/local/sbin/adb`'

- ➔ copy and paste commands [In jenkins\_commands.txt file] on 'Execute shell' section in Jenkins.