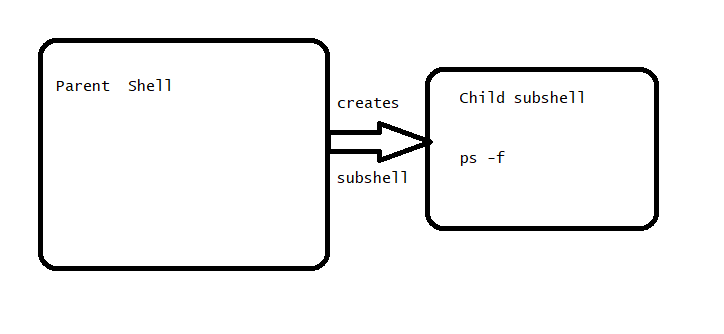
**Error Checking and Handling**

* The following topics
  + Error checking
  + Error handling
  + Error Prevention
* The exit status (exit codes or return codes) is the way Bash communicates the successful or unsuccessful termination of a process to its parent.
* Linux command execution process 

Let’s use mktemp command

mktemp --help

* Lets write a script with exit codes for success & failure

#!/bin/bash

# Run a command with will always work

mktemp

# storing the exit code of mktemp

mktemp\_ec=$?

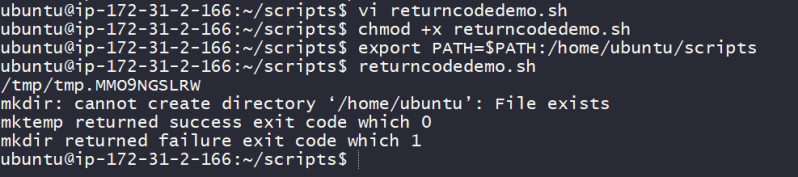
# Run a command which always fails

mkdir /home/ubuntu

mkdir\_ec=$?

echo "mktemp returned success exit code which ${mktemp\_ec}"

echo "mkdir returned failure exit code which ${mkdir\_ec}"

* now execute this script 
* Return code of 0 is success any other return code is failure

**Test Shorthand**

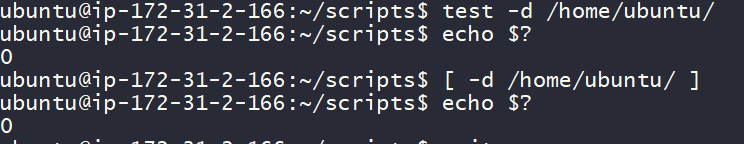
* We use *test* command in lot of scripts. Lets explore test
* Directory checking

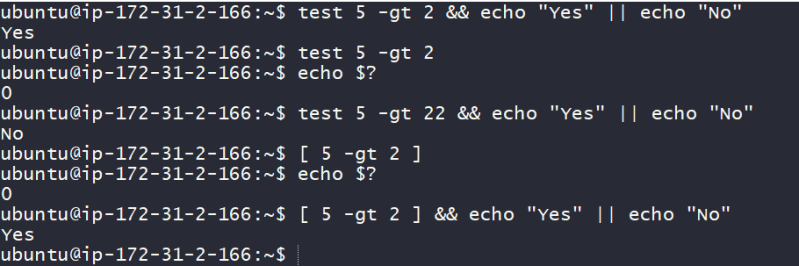
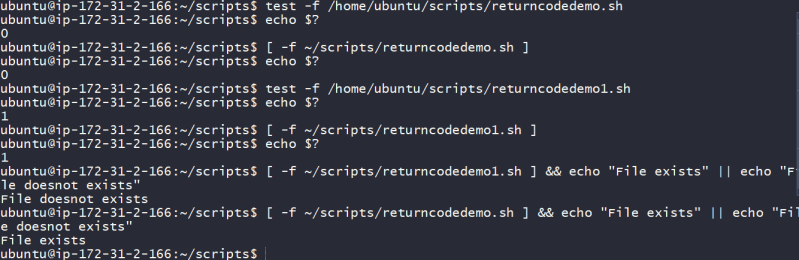
# full command

test -d <dirpath>

# short hand

[ -d <dirpath> ]



* Using test we can do comparisons using *-gt, -ne, -eq, -lt* 
* To check the file existence use *test -f* 
* So far we looked at how to check for directories, files and conditions which can act as error checks. if we combine this check with conditional statements we can avoid errors.
* For handling errors also we need a conditional statements

**if-then-exit**

* The basic idea is to test for a condition (IF), and if that condition is true, we do something (THEN) and if condition is false we return failure exit codes (exit)
* We want to write a script which prints content of file
* lets write a basic if condition

if <condition> then

statements

fi

* Using this lets write the script as shown below

#!/bin/bash

FILE=/tmp/random.txt

# Check if the file exits

if [[ -f ${FILE} ]]; then

cat ${FILE}

exit 0

fi

exit 1

* Now execute the script for negative test (/tmp/random.txt doesnot exist) 