Conditional Logic

- Lets understand Conditional-Logic

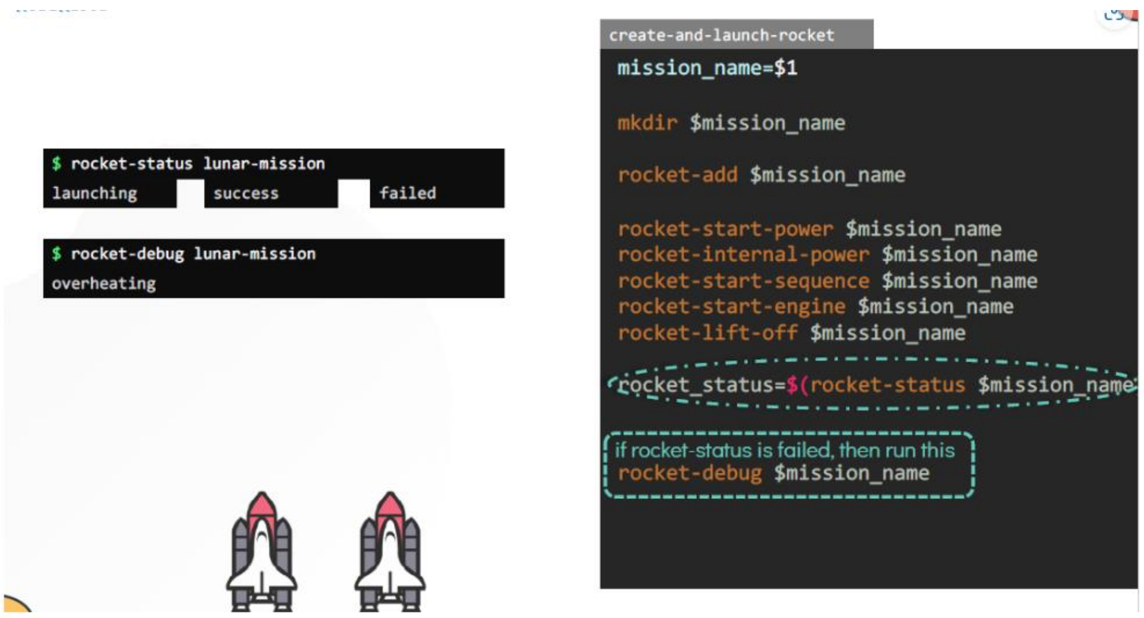
  Create and Launch Rocket

  - To Check the status of the rocket use below command:

    ```
    $ rocket-status lunar-mission
    ```

  - To debug the status of the rocket.

    ```
    $ rocket-debug lunar-mission
    ```



```
$ rocket-status lunar-mission
launching          success          failed
```

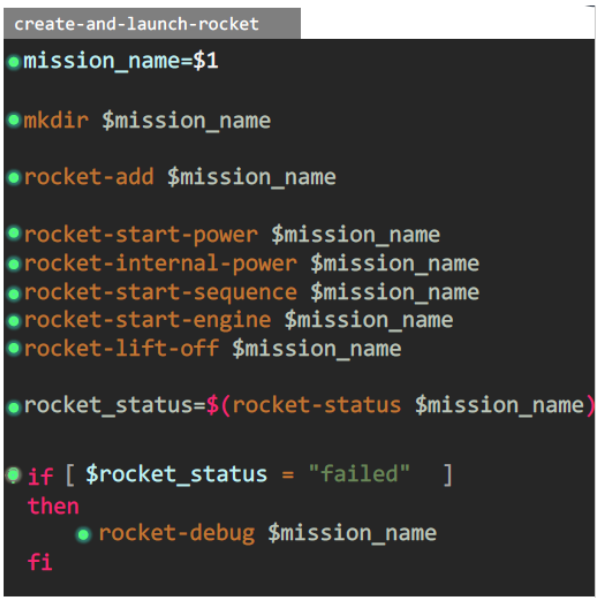
```
$ rocket-debug lunar-mission
overheating
```

```
create-and-launch-rocket
mission_name=$1

mkdir $mission_name

rocket-add $mission_name

rocket-start-power $mission_name
rocket-internal-power $mission_name
rocket-start-sequence $mission_name
rocket-start-engine $mission_name
rocket-lift-off $mission_name

rocket_status=$(rocket-status $mission_name

if rocket-status is failed, then run this
rocket-debug $mission_name
```

Conditional Statement

- `if` is defined as

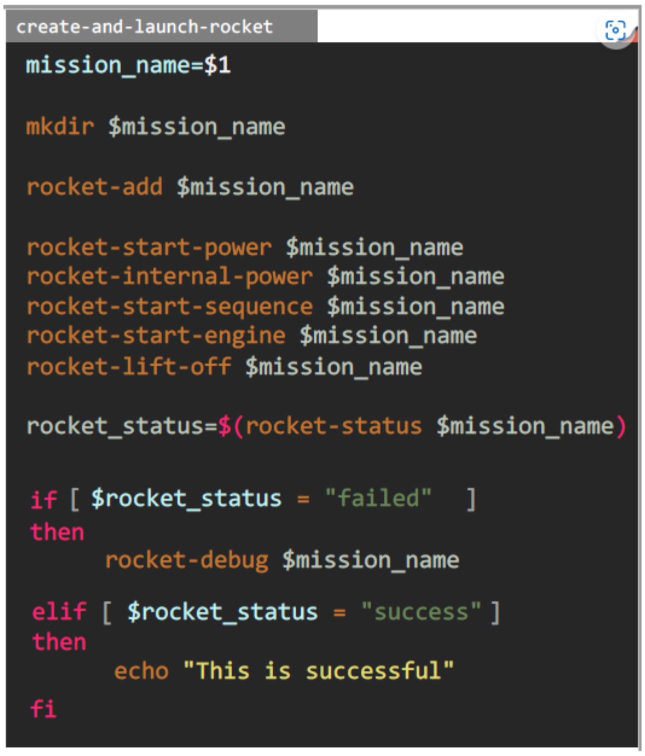
  ```
  if [ $rocket_status = "failed" ]
  then
    rocket-debug $mission_name
  fi
  ```



```
create-and-launch-rocket
mission_name=$1

mkdir $mission_name

rocket-add $mission_name

rocket-start-power $mission_name
rocket-internal-power $mission_name
rocket-start-sequence $mission_name
rocket-start-engine $mission_name
rocket-lift-off $mission_name

rocket_status=$(rocket-status $mission_name)

if [ $rocket_status = "failed"  ]
then
     rocket-debug $mission_name
fi
```

- `elif` condition is defined as

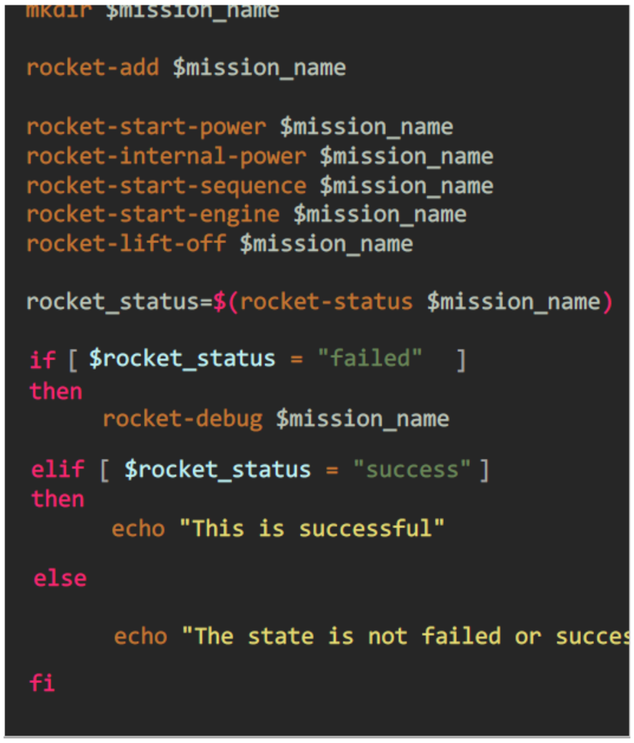
```
if [ $rocket_status = "failed" ]
then
  rocket-debug $mission_name
elif [ $rocket_status = "success" ]
then
  echo "This is successful"
fi
```



```
create-and-launch-rocket
mission_name=$1

mkdir $mission_name

rocket-add $mission_name

rocket-start-power $mission_name
rocket-internal-power $mission_name
rocket-start-sequence $mission_name
rocket-start-engine $mission_name
rocket-lift-off $mission_name

rocket_status=$(rocket-status $mission_name)


if [ $rocket_status = "failed"  ]
then
      rocket-debug $mission_name

elif [ $rocket_status = "success" ]
then
       echo "This is successful"
fi
```

- `else` is written as

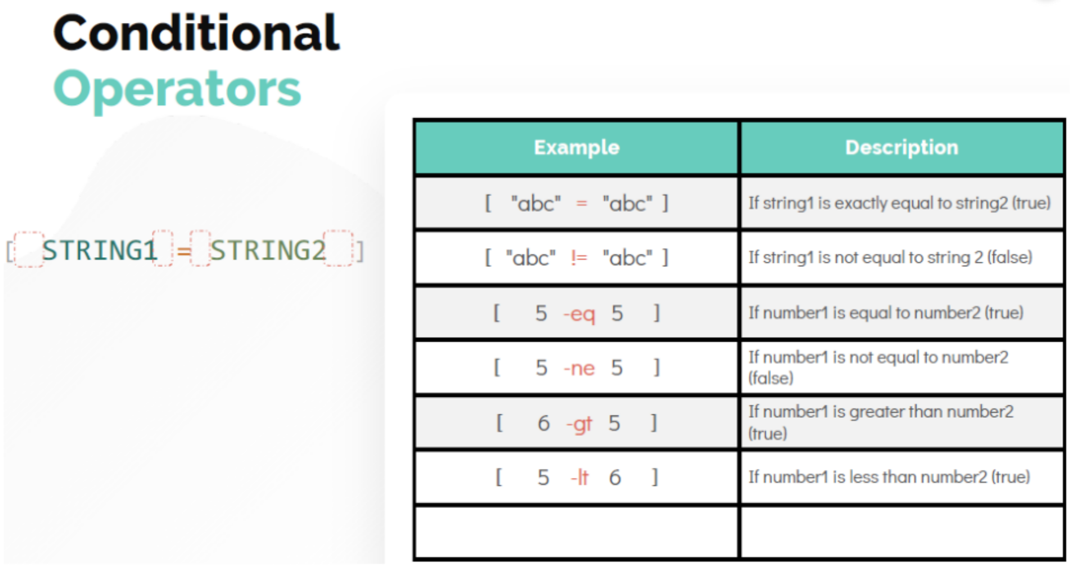
```
if [ -d "/home/bob/caleston" ]
then
  echo "Directory exists"
else
  echo "Directory not found"
```



```
mkdir $mission_name

rocket-add $mission_name

rocket-start-power $mission_name
rocket-internal-power $mission_name
rocket-start-sequence $mission_name
rocket-start-engine $mission_name
rocket-lift-off $mission_name

rocket_status=$(rocket-status $mission_name)

 if [ $rocket_status = "failed"   ]
 then
       rocket-debug $mission_name

 elif [ $rocket_status = "success" ]
 then
        echo "This is successful"

 else

        echo "The state is not failed or succes

 fi
```

Conditional Operators

- Comparing statement can be used as:

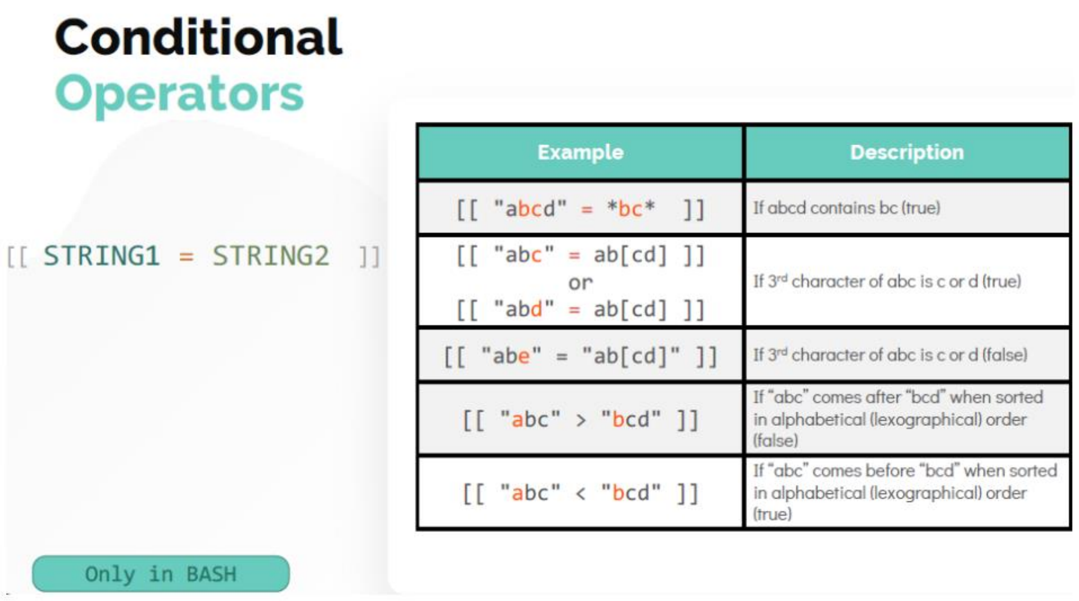


Conditional Operators

```
[ STRING1 = STRING2 ]
```

Example	Description
["abc" = "abc"]	If string1 is exactly equal to string2 (true)
["abc" != "abc"]	If string1 is not equal to string 2 (false)
[5 -eq 5]	If number1 is equal to number2 (true)
[5 -ne 5]	If number1 is not equal to number2 (false)
[6 -gt 5]	If number1 is greater than number2 (true)
[5 -lt 6]	If number1 is less than number2 (true)

- Conditional Operators that works in `bash`



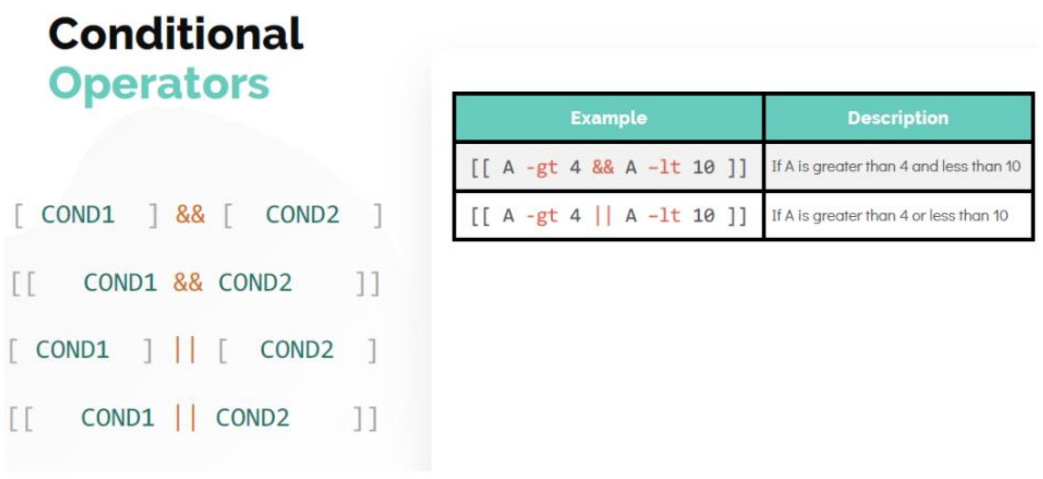
Conditional Operators

```
[[ STRING1 = STRING2 ]]
```

Example	Description
`[[ "abcd" = *bc* ]]`	If abcd contains bc (true)
`[[ "abc" = ab[cd] ]]` or `[[ "abd" = ab[cd] ]]`	If 3rd character of abc is c or d (true)
`[[ "abe" = "ab[cd]" ]]`	If 3rd character of abc is c or d (false)
`[[ "abc" > "bcd" ]]`	If "abc" comes after "bcd" when sorted in alphabetical (lexographical) order (false)
`[[ "abc" < "bcd" ]]`	If "abc" comes before "bcd" when sorted in alphabetical (lexographical) order (true)

Only in BASH

- `AND` and `OR` Operators

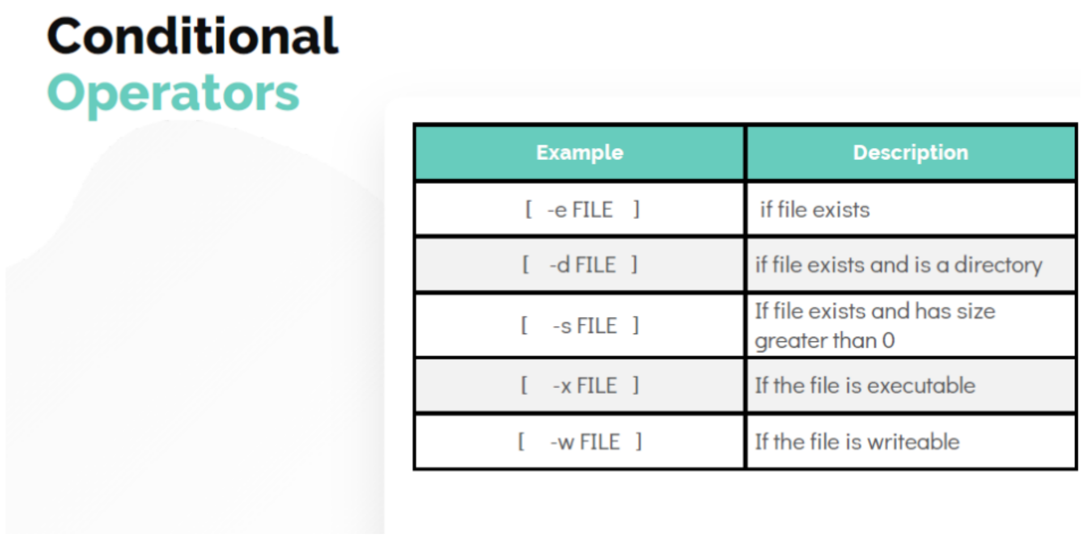


Conditional Operators

```
[ COND1 ] && [ COND2 ]
[[ COND1 && COND2 ]]
[ COND1 ] || [ COND2 ]
[[ COND1 || COND2 ]]
```

Example	Description
`[[ A -gt 4 && A -lt 10 ]]`	If A is greater than 4 and less than 10
`[[ A -gt 4 || A -lt 10 ]]`	If A is greater than 4 or less than 10

- Conditional operation description



Conditional Operators

Example	Description
[-e FILE]	if file exists
[-d FILE]	if file exists and is a directory
[-s FILE]	If file exists and has size greater than 0
[-x FILE]	If the file is executable
[-w FILE]	If the file is writeable