

Netact

mardi 15 mai 2018 10:18

Iempere

https://10.9.128.12/dana-na/auth/url_5/welcome.cgi pour ouvrir le tunnel

login → niane@nsn-intra

MP Windows

<https://srn4was.netact4.srntre.nsn-rdnet.net/authentication/authentication/login.jsp>

login : radiofr1 / Qwer_1234

Local

<https://netact2.net2.csvlab.fr/authentication/Login> omc/Nokia_123

C:\Windows\System32\drivers\etc

tu edites le fichier hosts

10.5.67.217 netact2 netact2.net2.csvlab.fr

10.5.67.200 vm16 vm16.net2.csvlab.fr

10.5.67.201 vm17 vm17.net2.csvlab.fr

Documentation

Rechercher : Andrew and amina to provide infos to connect to was

Fmcli pour fault manager

raccli pour les operaton CM

Swmcli pour les software manager

Sent to network to modify plan without import plan in netact

For alarm upload omc user need a certificate

UltraCaller

vendredi 27 juillet 2018 10:43

support.ultracaller@nokia.com

UltraCaller :

Server UC : Installé sous CentOS et placé dans la zone des PC automation - IP : 155.132.146.52

- User/pwd : crtuser/crtuser, root/aikon321
- Access au serveur uc via web : <http://155.132.146.52/uc/index.php>
User/pwd : admin/aikon321
- Pour l'install, j'ai principalement suivi l'aide en ligne :
<http://155.132.146.52/uc/help/index.html>

Proxy UC : installé sous CentOS et placé dans la cage de Faraday - IP : 155.132.146.53

- 2 cartes USB 5 ports sont installées. Il faudra certainement compléter par la suite.. Aussi possibilité de contrôler plusieurs PC Proxy via le même serveur.
- Même user/pwd que le serveur
- L'appli proxy (/home/crtuser/proxy) doit être lancée en su.

Installation d'un nouveau téléphone :

- Le package adb a été copié sur le proxy (/home/crtuser/ultracaller.apk)
- L'install se fait via cette commande pour chaque mobile connecté (voir détails en attach avec le support uc)
adb -s NB1GAD1782306814 install ./ultracaller.apk

Commandes Linux pour Proxy UltraCaller :

- Lancer le serveur proxy :
/home/crtuser/uc_proxy/proxy
- Redémarrer le serveur adb :
adb start-server
- Arrêter le serveur adb :
adb kill-server
- Lister les téléphones connectés :
adb devices -l
- Installer UltraCaller sur un téléphone :
adb -s 1015fab5a6ba2605 install ./ultracaller.apk
- Lister les processus UltraCaller
ps -eaf | grep -i adb
- Se connecter sur le téléphone :
adb -s 1015fab5a6ba2605 shell

Seveur UltraCaller :

- Start ntp:
service ntpd stop
- Verifier serveur ntp:
ntpq -pn
- Afficher les processus docker
docker ps
- Stop/start/kill docker
docker restart 38d330e04081
docker stop 38d330e04081
docker kill 38d330e04081
- Rep log docker:
ls /var/log/

Library python

def setup_ue : met l'IMEI dans un tableau pour empêcher de lancer des tests simultanés sur le

même mobile

def get_imei (alias): récupère l'alias de l'imei au début de chaque test case

def send_ssh_command: envoie, attends la fin de l'exécution et lit la sortie output ligne par ligne

def send_ssh_command_without_waiting: lance la commande et récupère le PID pour le stopper plus tard

def def send_to_api : envoie un nouveau test case à l'url d'UltraCaller, vérifie que l'url est joignable (response.ok) et renvoie la réponse d'UltraCaller qui contient le "testID" du test case lancé

def get_test_result: envoie et une requête à UC avec l'imei du mobile et le "testID" pour contre le statut du test

Et UC répond répond comme ci-dessous si la requête est bien reçue

```
<- [ftp_download] rcvd: {u'test': [{u'imei': u'357749065499775', u'desc': u'', u'testId':  
u'DCPGwaBUmx_k25-10150', u'ret': u'OK', u'name': u'taf_FTP_DCPGwaBUmx'}]}
```

Lancer soak_test

Sur autom7 ou automcat

nohup /nfs/Automation/Automat/tools/motor_soaktest >/tmp/motor_soaktest 2>&1 &

Robot Framework

vendredi 27 juillet 2018 15:39

Liens à consulter :

<http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html>

<https://deussyss.developpez.com/tutoriels/Python/Robotframework/>

<http://robotframework.org/robotframework/latest/libraries/BuiltIn.html>

<http://doc.ute.nsn-rdnet.net/>

<https://www.programcreek.com/python/example/100026/selenium.webdriver.FirefoxProfile>

<https://requests.readthedocs.io/en/master/>

```
crtadmin@automVM6:~$ sudo -i
```

```
[sudo] password for crtadmin:
```

```
MP : CRT_adm!
```

VM6 pour lancer scripts automatisé :

- (155.132.146.85) avec user/passwd : crtuser/P4crt!
- Rep Scripts :
/nfs/Automation/Automat/scripts/
- Nom du script:
SBTS17_OAMsanity.robot
- Nom script .sh pour lancer le Robot
OAMsanityV17.sh

Test Non reg :

- Résultat test block/unblock cells
2G Cells. OnAir=21 Enabled=21 Disabled=0 ----> Nombre Transmetteur TRX des cellules
3G Cells. OnAir=1 Enabled=1 Disabled=11 ----> Nombre de cells blocked/unblocked
4G Cells. OnAir=8 Enabled=8 Disabled=1 ----> Nombre de cells blocked/unblocked

Lancer un robot

- Help
Robot -h
 - Lancement robot sans le .sh
/usr/local/bin/robot --dryrun -v BTS_ADDRESS:10.5.67.29 -t CheckOnAir -l logModou.html /nfs/Automation/Automat/TestCases/SBTS_V17/SBTS17_OAMsanity.robot
 - Arguments commande précédente:
--dryrun : vérifie pas d'erreur dans la syntaxe de la commande
-l : pour renommer le fichier /tmp/logModou.html
-v : pour forcer la variable "BTS_ADDRESS" du script robot
-t : pour choisir le test case à exécuter

-o : pour définir le fichier output
-r : pour définir le fichier report
-d : pour spécifier le répertoire où les résultats sont envoyés
- | | | |
|---|--------|--|
| -i sanity //pour lancer les tests avec [Tags]= sanity défini au début du test case : [Tags] | sanity | |
| -e sanity // pour exclure les test cases avec [Tags] sanity | | |

Script pour lancer plusieurs fois un test :

/nfs/Automation/Automat/scripts/modou_5_iterations.sh

Répertoire du NAS :

\\155.132.146.66\Automation

Scheduler le script à exécuter :

- crtuser@automVM6:~\$ crontab -l (Pour lister les taches)
- crtuser@automVM6:~\$ crontab -e (Pour éditer la liste de taches)

Exemple de taches Scheduler à 6h tous les jours :

```
00 06 * * * /nfs/Automation/Automat/scripts/modou_5_iterations.sh  
> /nfs/Automation/Automat/scripts/Modou_sanity-20180630.log 2>&1
```

Code case : 586

A FAIRE

- Modifier les variables pour Reset/Block/Unblock specific RF Module dans les robots FL18 et FL19 du nouveau NFS
- Change {ALIAS} -->Done
- Test Setup et Test Teardown au début et à la fin de chaque TC --> Done

En FL18, les test cases suivants ont le bouton stop robot dans le code du test case

CheckOnAir

Backup scf
Block/unblock/reset all rf modules
Rollback

Modifier le nom des tempate FL19 des upgrade paths sur le nouveau NFS

Creation d'un fichier profile selenium

ute@debdev:~/tests_robot\$ cat [profile_firefox.py](#)

```
from selenium import webdriver
```

```
def create_profile(path):  
    fp = webdriver.FirefoxProfile()  
    fp.set_preference("network.proxy.type",0)  
    # fp.set_preference("network.proxy.autoconfig_url", 'http://proxyconf.glb.nsn-net.net/proxy.pac')  
    # fp.set_preference("browser.download.dir", path) //download de fichiers dans ce rép  
    #fp.set_preference("browser.helperApps.neverAsk.saveToDisk",'application/csv') //download fichier sans interaction humaine  
    fp.update_preferences()  
    return fp.path
```

Dans robot:

Library	profile_firefox.py
---------	------------------------------------

```
${profile_firefox}= create_profile /tmp/profile_firefox  
SeleniumLibrary.Open Browser  ${URL_WebEM}  ${BROWSER}  ff_profile_dir=${profile_firefox}
```

Atténuateur

lundi 30 juillet 2018 12:14

Connexion atténuateur :

PC pour accéder à l'atténuateur avec remote desk

@IP: 155.132.146.111

Login: crtuserd

MP: Pass4CRT!d

Interface :

<http://10.98.1.45>

Interface atténuateur Abdallah

<http://10.28.4.218>

Les cellules contrôlées par l'atténuateur :

- GNCEL-2 --> ATT01
- LNCEL-22 --> ATT02

Id machine + URL

vendredi 9 novembre 2018 11:24

Comment installer ADB

<https://confluence.int.net.nokia.com/display/LTEVERTA/How+to+install+Google+Android+driver>

Téléchargement nouveau load

- https://wft.int.net.nokia.com/SBTS/CI_ENB/builds/SBTS17A_ENB_1000_000652_000000
- Aller dans build
- Choisir SBTS, download

Serveur UltraCAller

<http://155.132.146.52/uc/index.php>

Serveur MCP

<http://mcpnlv.ih.lucent.com:8080/Mcp2/mcp.html>

PC recup trace téléphone

Login: ECS-PV

MP: Alcatel123

PC pour acceder à l'atténuateur avec remote desk

@IP: 155.132.146.111

Login: crtuserd

MP: Pass4CRT!d

Interface web atténuateur : 10.98.1.44

Serveur FTP tempéré

@IP: 10.9.146.30

@IP: 10.8.17.63

Login: iperf_user

MP: \$iperf

Serveur FTP Local

Id root = root/alcatel

10.19.1.66

10.19.1.67

10.19.1.68

10.19.1.69

Login: ftpuser

MP:ftpuser

Direction Download:

Rep= /home/ftpuser/DL

Nom des fichiers:

10Mo

1Go_1, 1Go_2, 1Go_3, 1Go_4, 1Go_5, 1Go_6

5G_1, 5G_2, 5G_3, 5G_4

8GO.blank

Direction Upload:

Rep= /home/ftpuser/UL

iperf

Lancer sur le serveur: iperf3 -s -i 1

Lancer sur le mobile : iperf -c 10.19.1.68 -u -t 120 -b 100M -i 1
iperf -c 10.19.1.68 -u -b 20m

RDC sur machine connectée à la BTS FL 4881

@IP 155.132.146.14

Login: PL-13902\lte

MP: @Lcatell

NSP 18.10

- URL : <https://10.5.67.1> ; Login/MP : admin/NokiaNsp1!
- Connexion ssh : 10.5.67.1 ; login/MP : nsp/vision
- Connexion swicl: cd /opt/nsp/Swicl.17.6.Linux ; ./swicl.bsh -p nspcsv
- BTS : <https://10.28.8.1:3600/#/bts-status/site-view> ; ID = 21322

Ultracaller server

IP: 155.132.146.52 ; login/MP: crtadmin/CRT_adm!

Connection Root : sudo -i

User toor4nsn BTS :

toor4nsn / oZPSOPOrRieRtu

Chine

Automcat : 10.108.147.235

Machine pour VMs : 10.108.147.234

VM20 : 10.108.147.236

ByPass authentication CAT:

Dans "profil.php" mettre :

```
<?php
/**
session_start();
if (!isset($_SESSION['profil']))
{
    header('Location: index.php');
    exit();
}
**/
?>
```

```
<a style="color: #124191" id="username">
    <i class="fas fa-user"></i>
    <!-- //<?php echo $_SESSION['userinfo'] ?> -->
    demo
</a>
```

Pointer <http://10.108.147.235/sequencer/profil.php> dans nokia-->index.php

small cell Chappe

10.5.72.69

Rooted mobile

vendredi 23 novembre 2018 16:32

Tools

- Emma: flashtool for sony devices
- Xperia companion: to backup conf for sony devices : mot de passe = sony
- Odin: flashtool for samsung

Samsung

- Télécharger "twrp" pour le modèle du téléphone :
<https://twrp.me/Devices/>
- Télécharger une archive zip "superSU"
- Pousser l'archive zip "superSU" sur le téléphone
Adb push superSu.zip /sdcard/Download
- Installer et lancer odin3
- Choisir l'option "AP"
- Désactiver la garantie
aller dans developer option --> activer OEM
- Mettre le téléphone en mode "Download" :
Boutton "power + home + volume down"
- Eteindre le téléphone
- Redémarrer le téléphone en mode "recovery" :
Boutton "power + home + volume up"
- Installer superSU
- Rebooter le téléphone

Sony XZ:

- Télécharger les drivers usb sony xz
 - Installer les drivers USB sony XZ
<https://developer.android.com/studio/run/oem-usb#InstallingDriver>
1. Connect your Android device to your computer's USB port.
 2. From Windows Explorer, open **Computer Management**.
 3. In the **Computer Management** left pane, select **Device Manager**.
 4. In the **Device Manager** right pane, locate and expand **Portable Devices** or **Other Devices**, depending on which one you see.
 5. Right-click the name of the device you connected, and then select **Update Driver Software**.
 6. In the **Hardware Update wizard**, select **Browse my computer for driver software** and click **Next**.
 7. Click **Browse** and then locate the USB driver folder. For example, the Google USB Driver is located in C:\Users\niane\Desktop\Root_Devices\sony_8331\Xperia_XZ_driver
 8. Click **Next** to install the driver.
 - Installer ADB et fastboot
 - Vérifier que vous pouvez déverrouiller le bootloader sur votre device
 9. In your device, open the dialer and enter *****7378423***** to access the service menu.
 10. Tap **Service info > Configuration > Rooting Status**. If Bootloader unlock allowed says **Yes**, then you can continue with the next step. If it says **No**, or if the status is missing, your device cannot be unlocked.

- Débloquer le bootloader
- 1. Aller dans <https://developer.sony.com/develop/open-devices/get-started/unlock-bootloader/#unlock-code>
- 2. Choisir la version du mobile --> récupérer le code unlock
- 3. Eteindre votre téléphone
- 4. Appuyer sur volume UP et brancher le (fastboot mode)
- 5. Sur le device taper les commandes suivantes:
`fastboot devices` (if no devices are printed, open the Devices Manager, click on the fastboot driver and press Update ---> Browse and point towards the location of the so0118adb.inf)
`fastboot -i 0x0fce oem unlock 0x+code No spaces between code and 0x<`
- Installer TWRP Recovery (mettre telephone en fastboot mode)
`fastboot flash recovery twrp.img`
- Clear Cache data avec TWRP
- Taper les commandes suivante dans le répertoire qui contient les fichiers boot.img et system.img (mettre telephone en fastboot mode)
`fastboot flash system system.img`
`fastboot flash boot boot.img`

<https://www.getdroidtips.com/download-android-8-0-oreo-aosp-sony-xperia-xz-custom-rom/>

- Reboot device
- Copier l'archive superSU sur le device
Adb push supersu.zip /sdcard/Download
- Démarrer le téléphone en mode recovery
Boutton power + volume down
- Installer supersu
Intall --> aller dans /sdcard/Download

<https://confluence.int.net.nokia.com/display/LTEVERTA/How+to+install+Google+Android+driver>
<https://confluence.int.net.nokia.com/display/LTEVERTA/How+to+prepare+Linux+UE+PC+to+work+with+UE+Manager+and+ta+ue>

<http://ue-community.china.nsn-net.net/>
<https://nokia.sharepoint.com/sites/csvcnucommunity>
https://www.yammer.com/nokia.com/#/threads/inGroup?type=in_group&feedId=12751349

Hi, Niane, I just root samsung phone. I found a root guide about Nokia 6.1 here <https://forum.xda-developers.com/nokia-6-2018/how-to/root-nokia-6-1-devices-t3831682> you can have a check. I didn't find any suitable guide about Sony phone, maybe you can try CF-Auto-Root method.

Reporting compteurs BTS

- Via WebEM : Aller dans compteurs --> choisir num cellule > LTE Radio Bearer > date
- Via Netact : Reporting > performance Manager > reporting tools > construire son rapport (Tech: LTE, Hiérarchie : Lncel, LTE ressource: select compteur, raw ...)

QCI:

1,2,3,4 = Garanted Bit Rate (GBR) ---> fournis par le PCRF
5...9 = Maximum Bit Rate (MBR)

QCI VoLTE

- QCI 9: default bearer
- QCI 5: Signalisation SIP (demandé numéro appelé, établissement du call, sonnerie mobile...)
- QCI 1 : dedicated bearer pour acheminer la voix (Fournie à l'IMS par le PCRF)

Décodage trace wireshark BTS pendant l'établissement d'un call VoLTE pour trouver les 3 QCI

- Filtre protocolaire wireshark : s1ap
- Attach request (active data) : InitialUEMessage, Service request
- Request capacity UE: InitialContextSetupRequest, UECapabilityInformation (dépiler pour trouver le default QCI > clic droit sur QCI > add as a colonne)
- Check du message attach : s1ap.procedureCode==5 (filtre wireshark)

Serveur vocal :

2G, 3G : 00104

4G: 0104

Activiter le MME Local à partir du scf

- Chercher "MME" dans le scf
- Unlock le MME 10.11.1.2

Trouver MCC et MNC dans le scf

- Chercher "plmnId" dans le scf
- MCC 208 = France
- MNC 1 = France

Trouver informations transport dans le SCF

- Chercher "IPIF-1" ou "IPIF-2" ou "localIpAddr"

Freq et cells :

- Un module radio offre 3 secteurs
- 1 secteur équivaut à 2G + 3G + 4G
- En 2G la BW = 200 KHz
- 3G et 4G réutilisation fréquence entre cellules

Identification des cells :

- 2G : BCCH (Broadcast Control Channel)
- 3G : PSC (Primary Scrambling Code)
- 4G : PCI

Param 3G :

- RF info--> Rx power : Puissance du signal en dbm (user defined->sumary info->wcdma)
- PSC : Pour identifier la cellule
- MCC : Mobile Country Code

- MNC : Mobile Network Code
- LAC : Local Area Code

Param 4G :

- RSRP : Puissance signal reçu
- EARFCN : Permet de retrouver la fréquence sur laquelle le signal est émis

Adresses IP serveurs FTP :

10.9.146.30 login/MP iperf_user2/\$iperf
10.8.17.63 login/MP iperf_user2/\$iperf

Debug iperf:

Tcpdump -nn -i "interface" port 5001

XCAL :

- Contact : Frank
- Support.emea@accuver.com
- Pour ADB sur le téléphone : *#0808#

PC XCAL :

- Login : ECS-PV
- Mp: Alcatel123

Cours LTE Amine

SIB 1 --> via BCCH PLMN

- Cell Selection Criteria : niveau signal minimum d'émission du mobile pour tenter l'attach sur une cellule
- Scheduling information for SIB2 and other SIBs

SIB 2 --> BCCH

- RACH Configuration
- Common Channel Configuration
- Cell Barring Configuration

Les infos SIB 1/2 permettent de vérifier les préconditions du RACH (premier message que le UE envoie à la BTS)

Après les messages SIB 1/2 : le mobile envoie un message RRC request à la BTS, la BTS répond en donnant les critères d'attach, l'UE envoie un message au MME pour s'attacher sur une cellule.

Canal PDCCH --> diffuse des infos commun à tous les UE. Le DCI est décodé par le mobile pour savoir les données qui le concernent (Quel PRB décodé dans la trame reçu, sur PRB envoyé les données à la BTS).

Canal PDSCH --> diffuse les données d'application à décoder par le mobile

Dans RRC connexion Request le mobile envoie son IMSI temporaire à la BTS, la BTS répond par un message rrc connection setup et les infos nécessaires pour la signalisation, après rrc setup completed le mobile envoie un message NAS à destination du MME (BTS n'interprète pas les messages NAS)

CQI indique le niveau du signal

<http://www.sharetechnote.com/>

Notes

vendredi 27 juillet 2018 16:51

Inventaire des téléphones :

- Android : 30, 31, 32, 33, 34, 36, 37, 49
- iPhone : 11, 12, 13, 14, 15, 16, 17
- Tel dans le Frigo : 30, 31, 32, 33, 34, 36, 37, 49

- Ubuntu:

Login/MP = crtuser/P4crt! Root/P4crt!

MySQL login/MP = root/P4crt!

Liste modules:

MySQL : /etc/php/7.0/mods-available/

Php : /etc/php/7.0/apache2/php.ini

Config proxy pour apt-get

Aller > param système > reseau > serveur mandataire

AirScale :

ASIA --> offre l'accès à l'interface web de la base, la conf

ABIA --> pour les modules radio, les cellules, les param

FL18 CLI : Command Line Interface

<http://cebula.emea.nsn-net.net:8888/spec/admin-spec/wikis/remote-api-doc>

Formation :

Cette session est réservée aux « new comers » n'ayant aucune connaissance en réseaux 4G/5G et sera uniquement en présentiel (pas de Webex).

Pour ceux qui ne seraient pas sur le site mercredi, un enregistrement de la session précédente est disponible sur le Nokia Learning Portal (suivre le lien

Video<<https://nokialearn.csod.com/DeepLink/ProcessRedirect.aspx?module=lodetails&lo=8d94f70c-6c3c-4180-b833-c8b547c09437>>)

Vous trouverez aussi tous les détails du programme sur la page Confluence des matins du CoDe :

<https://confluence.int.net.nokia.com/pages/viewpage.action?spaceKey=5GFTPARIS&title=On+boarding+-+Les+Matins+du+CoDe>

Jquery

<http://malsup.com/jquery/block/#dialog>

Turn on phone, enable Developer Options by tapping several times on "**build number**" (Settings -> About Phone) and enable "USB Debugging"

Code erreur FL18

/opt/Nokia/Managers/BTS Site/BTS Site Manager/tools/CLI/README.txt

All tools use exit codes to roughly indicate any problems during execution.
Common exit codes are

- 0 No problems
- 1 Miscellaneous problems
- 2 Invalid operation
- 3 Invalid parameters
- 4 Site unreachable
- 5 Site reserved
- 6 Login failed
- 7 Unexpected error
- 8 Timeout

JSON request for 2 variables

https://135.238.178.173/inc/get_saved_sequence.php?username=%27niane%27&seqName=%27test%27

Mettre à jour UTE ADMIN

Se connecter avec user crtadmin sur une vm, se mettre en sudo et taper la commande:

```
pip install -U ute_admin --upgrade
```

MySQL

lundi 21 janvier 2019 11:51

MAJ champ "state" de "automVM5"

```
UPDATE vm SET state = 'LOCKED' WHERE name = 'automVM5';
```

```
UPDATE software SET name = 'FL19_ENB_0000_000172_000016' WHERE id = '16';
```

MAJ champ "state" de toutes les lignes qui commence par "automVM"

```
UPDATE vm SET state = 'UNLOCKED' WHERE name LIKE 'automVM%';
```

Affiche les lignes qui commence par 'automVM' dont le state est différent de "LOCKED"

```
SELECT * FROM vm WHERE state != 'LOCKED' AND name LIKE 'automVM%';
```

Afficher Max ID d'une table

```
select MAX(id) FROM tcresult;
```

Backup database (commandes à exécuter sur la console du PC et pas sur celle Mysql)

```
mysqldump -u root -p --databases lab > /home/crtuser/lab_db-backup.sql
```

```
mysqldump -u root -p --all-databases > /home/crtuser/full_db-backup.sql
```

```
mysqldump -ucrtuser -pcrtuser lab TestStatus > /tmp/TestStatus.sql
```

```
mysql -ucrtuser -pcrtuser lab < /tmp/TestStatus.sql
```

Supprimer une ligne

```
DELETE FROM `nom_table` WHERE `nom_colonne`=" "
```

Ajouter une colonne à une table

```
ALTER TABLE [table_to_update] ADD COLUMN [table_to_add] VARCHAR(15) AFTER name;
```

Connexion sur une base MySQL distante

```
mysql -ucrtuser -D lab -h 135.238.178.190 -p
```

Configuration alias

```
vi .bash_aliases
```

Augmenter le nombre caractère table

```
ALTER TABLE software MODIFY name VARCHAR(130);
```

CREATE TABLE

```
create table servers (ip VARCHAR(20), file VARCHAR(250), timeout VARCHAR(250));
```

```
create table params_seq (divName VARCHAR(50), tcName VARCHAR(150), args VARCHAR(250),  
seqName VARCHAR(50), user VARCHAR(30));
```

INSERT DATA TO TABLE

```
INSERT INTO ftp_servers (ip, file, timeout) VALUES('10.19.1.66', '10Mo 1Go_1 1Go_2 1Go_3 1Go_4  
1Go_5 1Go_6 5G_1 5G_2 5G_3  
5G_4 8GO.blank', '1m 2m 3m 4m 5m 10m 15m 20m 25m 30m 35m 40m 45m 50m 55m 60m');
```

DELETE COLUMN

```
ALTER TABLE ftp_servers DROP COLUMN timeout;
```

RENAME COLUMN

```
ALTER TABLE table_name CHANGE COLUMN old_name new_name varchar(150);
```

Duplicate table

```
CREATE TABLE new_table LIKE old_table;  
INSERT new_table SELECT * FROM old_table;
```

Install lib robot for database

```
pip install PyMySQL  
pip install robotframework-databaselibrary
```

5G add cells inventory

```
ALTER TABLE bts ADD COLUMN nrcell VARCHAR(1000) AFTER lcell;
```

Modification des fichiers:

inc/bts_processing.php

Bts.html

Linux

mardi 29 janvier 2019 11:59

Stdin= clavier =0

Stdout= ecran = 1

Stderr = 2 =/dev/null = sortie error comme "file does not existe"

2>&1 = stderr et stdout sont redirigés au même endroit

Date

date '+%Y%m%d'

Return PID:

Commande; echo \$!

NTP

Show synchro: ntpq -pn

Start ntp : service ntpd stop/status/start

* = synchroniser

Synchroniser avec la date du peer : ntpdate -v 155.132.146.66

#service ntp status

service ntp stop

service ntp status

ntpq -pn

ntpdate -v 155.132.146.66

apt install ntpdate

ntpdate -v 155.132.146.66

service ntp start

ntpq -pn

date

Process X = gestionnaire interface graphique

Affiche les process liés à l'interface graphique :#psg X

Copier fichier avec les mêmes droits

#cp -p

Fichier de log

/var/log/messages

/var/log/syslog

Reboot propre

shutdown -r

Info Model Search string

robot -v BTS_ADDRESS:[fd6b:b951:3aa5:622::2:1] -v PORT:3600 -t dumpAll

/nfs/Automation/Automat/tools/example_infomodel.robot > /tmp/21378.infomodel

#grep MME 21367.txt

#grep MME 21367.txt| awk '{print \$1}'

grep

'^/MRBTS-1/RAT-1/BTS_L-1/BTS_CONF-1/NP-1/SCF-1/MRBTS_A-21367/LNBTS_A-21367/LNMME_A-[0-9] ' 21367.txt

Lister les montage disques/fichiers

cat /etc/fstab

Remote copy

```
# scp -r source @IP_dest:/rep_dest
```

Synchroniser 2 répertoires sur 2 machines

Freefilesync

Nmap:

```
# nmap -p 443 -Pn 10.28.4.117
```

```
# nmap -p 443 -6 -Pn fd6b:b951:3aa5:622::1:31
```

Afficher le PID du terminal en cours

```
# echo $$
```

Replace all string in text file

```
:%s/nfs\Automation\Automat\TestCases\nfs\robot_scripts\TestCases
```

```
:%s/nfs\robot_scripts\TestCases\nfs\Automation\Automat\TestCases
```

Raccourcir Adresse IPV6

```
echo fd6b:b951:3aa5:0622::1:0059 | sed -e 's/:0*/:/g'
```

```
fd6b:b951:3aa5:622::1:59
```

Tcpdump iperf

```
# tcpdump -nn -i "interface" port 5001
```

Remote copy

```
# scp fl18.php crtadmin@155.132.146.21:/var/www/html/nokia/
```

Création lien symbolique

```
ln -s software_update_and_activate_6 oam_flsuua45
```

Ajouter un alias

```
vi ~/.bashrc
```

Pour copier les templates avec les liens symbolinks

```
tar czvf TestCases.tar.gz TestCases ### pour zipper
```

```
scp TestCases.tar.gz csvuser@10.108.147.235:/nfs/robot_scripts/
```

```
tar xzvf TestCases.tar.gz ### pour unzipper
```

Send POST command to api

```
curl -XPOST https://www.hackthebox.eu/api/invite/generate
```

NSP

lundi 18 février 2019 10:01

BTS21327

fd6b:b951:3aa5:622::1:69

Swicl profiles

NSP 19.4 profile = nsp2csv

NSP 19.7 profile = nsp3csv

Répertoire Swicl server

/opt/nsp/

Connexion swicl avec profil nsp2csv

-bash-4.2\$./swicl.bsh -p nsp2csv

Installer une nouvelle release NSP

1. Aller dans « C:\nsp » et supprimer le dossier « client »
2. Lancer la page <https://10.5.67.2/> login/MP = admin/NokiaNsp1!
3. Installer "Network Functions Manager-Packet "

ID connexion nsp1:

web: <https://10.5.67.1/>; login/MP = admin/NokiaNsp1!

ssh: 10.5.67.1 ; login/MP/Profile = nsp/vision/nspcsv;

ID connexion nsp2:

web: <https://10.5.67.1/>;

ssh: 10.5.67.1 ; login/MP/Profile = nsp/vision/nsp2csv;

Debug:

getenv : pour voir le fichier de log

debug all true : pour activer le debug

show alarm on MRBTS

fm dis basic MRBTS-21322 -semi

attribute alarmName: 1472=cell shutdown

fm dis faultManager:network@MRBTS-21322@MRBTS-21322@LNBTS-21322@LNCEL-12

|alarm-1472-3-698 -a alarmName -semi

Add snapshot to NE

snapshot addNE snap-mgr:8 MRBTS-21322

Parametre change Name for cell:

list cell : dis racfl LNCEL -like network:MRBTS-21322:% -stab

FDN : network:MRBTS-21322:MRBTS-21322:LNBTS-21322:LNCEL-1

commande to change attribute a2TimeToTriggerActGERANMeas to 512ms : setOnline racfl

LNCEL network:MRBTS-21322:MRBTS-21322:LNBTS-21322:LNCEL-1

a2TimeToTriggerActGERANMeas 512ms

Get MRBTS Name:

get racfl MRBTS network:MRBTS-21322:MRBTS-21322 btsName

Cm Resync From NSP

cm resync network:MRBTS-21322

launch script on swicl
execScript ./script/lockallcell.tcl

Lister les param MRBTS TL
dis ractl MRBTS

Lister les param MRBTS FL
dis racfl MRBTS

Lister les MRBTS
ls racfl MRBTS -like network:MRBTS%

Lister les objets connectés au NSP
ls netw NetworkElement

Log reference

<http://135.238.178.146/traces/MRBTS-21322/Minh-Tien/26-02-2019-16-20-59/logMinh-Tien.html>

Répertoire Mapping PLMN

X:\Regression_Campaign\Test_case_descriptions\Test Area Procedures\Test_Area_X_procedure
PLMN_Mapping_Table_Change

[\\155.132.146.66\crt_verizon\Regression_Campaign\Test_case_descriptions\Test Area Procedures
\Test_Area_X_procedure](http://155.132.146.66/crt_verizon/Regression_Campaign/Test_case_descriptions/Test_Area_Procedures/Test_Area_X_procedure)

NSP 19.4 repertoire backup = /opt/nsp/nfmp/nebackup/NE3S/MRBTS-ID/

NSP 18 repertoire backup = /opt/nsp/nfmp/FLEXIMRBTS/NE3S/MRBTS-ID/

Inventory

mercredi 20 mars 2019 17:09

Mot de passe crtadmin = CRT_adm!

Faire le backup des scf (save dans SCF_FILES sur le NFS)

Sur automVM1 (password=P4crt:)

```
crtuser@automVM1:/nfs/Automation/Automat/backup_auto$ vi newbackup_auto_ip6.sh
```

```
crtuser@automVM1:/nfs/Automation/Automat/backup_auto$ ./newbackup_auto_ip6.sh
```

Insert SCF data to DB

Sur automVM1 (Fabrique inventory: fichier "sbts_LATEST.csv" dans /nfs/Automation/Automat/INVENTORY)

```
crtuser@automVM1:/nfs/Automation/Automat/backup_auto$ ./invent_from_scf.sh
```

Insertion des Test cases dans la database sur autom7

```
cd /nfs/Automation/Automat/tools
```

```
more tc2db.sh
```

```
./tc2db.sh /nfs/Automation/Automat/TestCases/cat/FL18cat.robot
```

Single inventory

Robot "inventory2" crée les fichiers pour populer les tables suivantes de la database:

Cells

Rmods

PCI

BTS

Global inventory

Script scheduler dans crontab et qui lance les autres scripts =

/nfs/robot_admin/backup_auto/newbackup_auto.sh

```
# contab -l
```

Infomodel BTS

```
robot -t dumpAll -v BTS_ADDRESS:10.5.79.53 -L DEBUG -d /tmp infomodel.robot > /nfs/robot_scripts/dump_infomodel/21113.txt
```

Faire un infomodel pour FL18:

```
robot -v BTS_ADDRESS:10.5.72.25 -v PORT:3600 -t
```

```
dumpAll /nfs/Automation/Automat/tools/infomodel.robot > /tmp/4881.infomodel
```

Sinon pour des bts normales :

```
robot -v BTS_ADDRESS:10.28.7.57 -t dumpAll /nfs/Automation/Automat/tools/infomodel.robot > 21372-infomodol
```

Single inventory

```
crtuser@automVM1:/nfs/robot_admin$ ./single_inventory.sh @IP_BTS
```

Installation AutomCAT

lundi 25 mars 2019 11:43

Configuration <documentRoot>

Editer fichier /etc/apache2/sites-available/automatization_csv.conf
Autoriser l'accès au <document>: chmod 777 /var/www/html/nokia/

Enable/Disable web site

```
# a2ensite automatization_csv.conf
# a2dissite automatization_csv.conf
# service apache2 reload
```

Configure alias

Edit file : /etc/apache2/mods-available/alias.conf

```
Alias /traces "/nfs/Automation/Automat/traces/"
<Directory "/nfs/Automation/Automat/traces">
    Options Indexes Multiviews FollowSymlinks
    AllowOverride All
    Require all granted
</Directory>
```

start/stop/status Apache

/etc/init.d/apache2 status

Backup database on 155.132.146.21

```
mysqldump -u root -p --databases lab > /home/crtuser/lab_db-backup.sql
mysqldump -u root -p --all-databases > /home/crtuser/full_db-backup.sql
```

Importer database lab de autom7 à automCAT

Sur autom7

```
scp lab_db-backup.sql crtadmin@135.238.178.190:/home/crtadmin/
```

Sur automCAT

Login/MP Mysql= root/P4root!

```
crtadmin@automCAT:~$ mysql -u root -p
mysql> create database lab;
mysql> exit;
crtadmin@automCAT:~$ mysql -u root -p lab < /home/crtadmin/lab_db-backup.sql
crtadmin@automCAT:~$ mysql -u root -p
mysql> use lab;
mysql> CREATE USER 'crtuser'@'localhost' IDENTIFIED BY 'crtuser';
mysql> CREATE USER 'crtuser'@'%' IDENTIFIED BY 'crtuser';
mysql> GRANT ALL PRIVILEGES ON *.* TO 'crtuser'@'localhost' WITH GRANT OPTION;
mysql> GRANT ALL ON lab.* TO 'crtuser'@'%';
mysql> FLUSH PRIVILEGES;
root@automCAT:~# vi /etc/mysql/mysql.conf.d/mysqld.cnf
and comment #bind-address = 127.0.0.1
root@automCAT:~# service mysql restart
root@automCAT:~# service mysql status
```

Software upload

Put in /etc/php/7.0/apache2/php.ini file

session.upload_progress.enabled = ON

file_uploads = On

upload_max_filesize = 60000M

```
post_max_size = 6000M
```

Install php_ssh2

```
sudo apt-get install php7.0-cli  
sudo apt-get install libssh2-1 php-ssh2
```

Testing:

```
php -a  
php > $result = ssh2_connect('localhost');  
php > var_dump($result);  
resource(2) of type (SSH2 Session)
```

stop/start Apache

```
/etc/init.d/apache2 status
```

Check log apache2/PHP

```
/var/log/apache2/error.log
```

How To Create a Self-Signed SSL Certificate for Apache in Ubuntu 16.04

Step 1: Create the SSL Certificate

TLS/SSL works by using a combination of a public certificate and a private key. The SSL key is kept secret on the server. It is used to encrypt content sent to clients. The SSL certificate is publicly shared with anyone requesting the content. It can be used to decrypt the content signed by the associated SSL key.

We can create a self-signed key and certificate pair with OpenSSL in a single command:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

Both of the files you created will be placed in the appropriate subdirectories of the /etc/ssl directory:

```
private key will be placed in /etc/ssl/private  
public certificate in /etc/ssl/certs
```

While we are using OpenSSL, we should also create a strong Diffie-Hellman group, which is used in negotiating Perfect Forward Secrecy with clients.

We can do this by typing:

```
sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```

Step 2: Configure Apache to Use SSL

1. We will create a configuration snippet to specify strong default SSL settings.
2. We will modify the included SSL Apache Virtual Host file to point to our generated SSL certificates.
3. (Recommended) We will modify the unencrypted Virtual Host file to automatically redirect requests to the encrypted Virtual Host.

Create a new snippet in the /etc/apache2/conf-available directory. We will name the file ssl-params.conf to make its purpose clear:

```
sudo nano /etc/apache2/conf-available/ssl-params.conf
```

Copy the provided settings in /etc/apache2/conf-available/ssl-params.conf file:

```
# from https://cipherli.st/  
# and https://raymii.org/s/tutorials/Strong\_SSL\_Security\_On\_Apache2.html  
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH  
SSLProtocol All -SSLv2 -SSLv3
```

```

SSLHonorCipherOrder On
# Disable preloading HSTS for now. You can use the commented out header line that includes
# the "preload" directive if you understand the implications.
#Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains; preload"
Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains"
#Header always set X-Frame-Options DENY
Header always set X-Content-Type-Options nosniff
# Requires Apache >= 2.4
SSLCompression off
SSLSessionTickets Off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
SSLOpenSSLConfCmd DHParameters "/etc/ssl/certs/dhparam.pem"

```

Save and close the file when you are finished.

Modify the Default Apache SSL Virtual Host File

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

Change /etc/apache2/sites-available/default-ssl.conf file like that:

```

<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin your_email@example.com
        ServerName server_domain_or_IP

        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLEngine on

        SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
        SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

        <FilesMatch "\.(cgi|shtml|phtml|php)$">
            SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
            SSLOptions +StdEnvVars
        </Directory>

        BrowserMatch "MSIE [2-6]" \
            nokeepalive ssl-unclean-shutdown \
            downgrade-1.0 force-response-1.0

    </VirtualHost>
</IfModule>

```

(Recommended) Modify the Unencrypted Virtual Host File to Redirect to HTTPS

As it stands now, the server will provide both unencrypted HTTP and encrypted HTTPS traffic. For better security, it is recommended in most cases to redirect HTTP to HTTPS automatically.

To adjust the unencrypted Virtual Host file to redirect all traffic to be SSL encrypted, we can open the /etc/apache2/sites-available/000-default.conf file:


```
<VirtualHost *:80>
```

```
    Redirect "/" "https://your_domain_or_IP/"
```

```
</VirtualHost>
```

Save and close the file when you are finished.

Step 3: Adjust the Firewall

If you have the ufw firewall enabled, as recommended by the prerequisite guides, might need to adjust the settings to allow for SSL traffic. Luckily, Apache registers a few profiles with ufw upon installation.

We can see the available profiles by typing:

```
sudo ufw app list
```

You can see the current setting by typing:

```
sudo ufw status
```

If you allowed only regular HTTP traffic earlier, your output might look like this:

Output

Status: active

To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere
Apache	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
Apache (v6)	ALLOW	Anywhere (v6)

To additionally let in HTTPS traffic, we can allow the "Apache Full" profile and then delete the redundant "Apache" profile allowance

```
sudo ufw allow 'Apache Full'
sudo ufw delete allow 'Apache'
```

Your status should look like this now:

```
sudo ufw status
```

Output

Status: active

To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere
Apache Full	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
Apache Full (v6)	ALLOW	Anywhere (v6)

Step 4: Enable the Changes in Apache

Now we can enable the SSL and headers modules in Apache, enable our SSL-ready

Virtual Host, and restart Apache.

We can enable `mod_ssl`, the Apache SSL module, and `mod_headers`, needed by some of the settings in our SSL snippet, with the `a2enmod` command:

```
sudo a2enmod ssl
```

```
sudo a2enmod headers
```

Next, we can enable our SSL Virtual Host with the `a2ensite` command:

```
sudo a2ensite default-ssl
```

We will also need to enable our `ssl-params.conf` file, to read in the values we set:

```
sudo a2enconf ssl-params
```

At this point, our site and the necessary modules are enabled. We should check to make sure that there are no syntax errors in our files. We can do this by typing:

```
sudo apache2ctl configtest
```

If everything is successful, you will get a result that looks like this:

Output

```
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using
127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
```

If your output has Syntax OK in it, your configuration file has no syntax errors. We can safely restart Apache to implement our changes:

```
/etc/init.d/apache2 restart
```

Migration sequenceur sur automCAT

```
Modifier /var/www/html/site/in/save_tc.php
/var/www/html/nokia/sequencer/inc/poubelle/
```

```
retrieveSeq.php
automVM2%
```

Mise en place du LDAP

```
apt-get install php7.0-ldap
a2enmod authnz_ldap
/etc/init.d/apache2 restart
ldapsearch -H ldaps://ed-p-gl.emea.nsn-net.net -b "ou=internal,ou=People,o=nsn" -o nettimeout=5 -
LLL -x "(uid=niame)"
```

PHP disabled notice and warning

```
Edit file : /etc/php/7.0/apache2/php.ini
;error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT
error_reporting = E_COMPILE_ERROR|E_RECOVERABLE_ERROR|E_ERROR|E_CORE_ERROR
```

Git

vendredi 19 avril 2019 11:28

La commande `sudo apt install git-all` permet d'installer git avec les interfaces graphiques
\$ `sudo apt install git-all` (interface graphique git GUI)

La commande `git config` permet de configurer l'outil git en donnant mon nom et adresse email afin d'insérer automatiquement ces informations lors de mes modifications de code source

```
$ git config --global user.name "niane modou"
$ git config --global user.email "nianemodou@gmail.com"
$ git config --global --list
```

la commande `git add` permet d'ajouter les modifications de l'espace de travail dans la zone d'index (modification d'un fichier, création d'un nouveau fichier...)

```
$ git add [nom_du_fichier]
$ git add . (pour toutes les mods)
```

la commande `git commit` permet de pousser le contenu de la zone index dans l'historique du projet (dépot git local)

```
$ git commit -m"ajout d'une balise paragraph"
```

la commande `git diff` permet de comparer le contenu du dépôt git local et celui de l'espace de travail

```
$ git diff
```

la commande `git diff --cached` permet de comparer le contenu du dépôt git local et celui de la zone d'index

```
$ git diff --cached
```

la commande `git reset` permet de supprimer les mods d'un fichier qui est dans la zone index

```
$ git reset [nom_du_fichier]
```

la commande `git status` permet de lister les changements de l'espace de travail par rapport au dépôt git local

```
$ git status
```

La commande `git log` permet d'afficher la liste des commit

```
$ git log
$ git log -n 2 (pour afficher les 2 derniers commit)
```

La commande `git show SHA-1` permet d'afficher le contenu de ce commit

```
$ git show e5478f0820243ec78022fc54db03913d7f8cd166
```

La commande `git show master` permet d'afficher le contenu du commit pointer par master

```
$ git show master
```

La commande `git checkout SHA-1` permet de se séplacer dans un commit (Head pointe sur ce commit)

```
$ git checkout e5478f0820243ec78022fc54db03913d7f8cd166
```

La commande `git checkout master` permet de revenir à la branche master

```
$ git checkout master
```

La commande `git tag` permet de renommer un SHA-1 sous forme d'un nom plus explicite (faire d'abord un checkout SHA-1 pour pointer sur le commit)

```
$ git tag Mon_1er_SITE -m"la premiere version de mon site"
```

La commande `git --delete` permet de supprimer un tag

```
$ git tag --delete Mon_1er_SITE
```

La commande git tag permet de lister l'ensemble des tags

```
$ git tag
```

La commande git clone permet de dupliquer un dépôt distant en local (remote_url --> cliquer sur le nom du projet à cloner dans GitHub et copier l'URL HTTPS ou SSH, clone_mon_site_web --> nom du dossier dans lequel on va cloner le dépôt distant)

```
$ git clone [remote_url] [local_folder_name]
```

```
$ git clone https://github.com/mod72/mon\_site\_web.git clone_mon_site_web
```

La commande git config permet de configurer l'outil git mais il faut d'abord se déplacer dans dossier où on a cloné le dépôt distant

```
$ git config user.name "niane modou"
```

```
$ git config user.email "nianemodou@gmail.com"
```

La commande git remote -v permet de lister les pointeurs de notre dépôt local (le dépôt origin sert d'origine à notre dépôt local)

```
$ git remote -v
```

La commande git remote show origin permet d'avoir des informations sur un remote

```
$ git remote show origin
```

La commande git remote add origin [remote_url] permet d'ajouter un pointeur à notre dépôt local

```
$ git remote add origin https://github.com/mod72/mon\_site\_web.git
```

La commande git push -u origin master permet de pousser l'ensemble des modifications que nous avons effectuées au niveau des commit sur le dépôt local vers notre dépôt hébergé par GitHub (local --> remote)

```
$ git push -u origin master
```

La commande git push origin [nom_du_tag] permet de pousser le tag sur le dépôt local vers le remote

```
$ git push origin [nom_du_tag]
```

```
$ git push --tags (pour pousser tous les tags)
```

La commande git fetch permet de déplacer les modifications de notre dépôt distant vers le dépôt local (remote --> local)

```
$ git fetch
```

La commande git pull permet de déplacer le tag master du dépôt local vers le tag origin master qui correspond à la dernière version récupérée sur le dépôt distant

```
$ git pull (réalise automatiquement un git fetch si cela est nécessaire)
```

La commande git blame [nom_du_fichier] permet d'avoir ligne par ligne les modifications qui ont eu lieu dans le fichier et leurs auteurs

```
$ git blame [nom_du_fichier]
```

```
$ git blame -L 10,20 [nom_du_fichier] (pour afficher les modifications entre la ligne 10 et 20)
```

```
$ git blame -L 10,+4 [nom_du_fichier] (pour afficher à partir de la ligne 10 les 4 prochaines lignes)
```

La commande git stash save "message" permet de mettre de côté des modifications présent dans notre espace de travail

```
$ git stash save "modification de gitignore"
```

```
$ git stash list (pour lister les stash)
```

```
$ git stash show
```

La commande git stash pop 0 permet de réinjecter dans l'espace de travail des modifications qui étaient mis de côté

\$ git stash pop 0

La commande git pull --rebase permet d'avoir un historique de merge beaucoup plus clair (1.commit modifications dépôt local, 2.git pull --rebase, 3.Ajouter un tag, 4.git push, 5.git push --tags)

\$ git pull --rebase

La commande git branch [nom_branch] permet de créer une branche à partir du commit sur lequel on a pointé

\$ git branch [nom_branch]

La commande git checkout -b [new_branch] permet de créer une nouvelle branche et de se déplacer dessus

\$ git checkout -b [new_branch]

La commande git push --set-upstream origin [nom_branch] permet de lier une branche locale à une branche distante

\$ git push --set-upstream origin [nom_branch]

La commande git branch permet d'afficher la liste des branches

\$ git branch

La commande git branch -a permet de voir la liste des branches qui sont sur le dépôt distant

\$ git branch -a

La commande git cherry-pick SHA-1 permet de copier sur la branche master un commit qui a été réalisé sur une autre branche (se positionner d'abord sur la branche master)

\$ git cherry-pick 6018d4733f32fd821912ca840755ca1c4cee0149

La commande git merge [nom_branch] permet de faire un merge de la branche [nom_branch] et la branche master (se positionner d'abord sur la branche master)

\$ git merge [nom_branch]

La commande git rebase master permet de ramener la branche courante sur la branche master (se positionner sur la branche qu'on souhaite ramener sur une autre; le premier commit de cette branche sera ramener sur le dernier commit du master)

\$ git rebase master (si git rencontre un conflit, aller dans le fichier qui cause problème et supprimer les lignes ajoutées par git, puis faire "git add [nom_fichier_modifié] , "git rebase --continue" , "git checkout master", "git merge [nom_branch_ranemer]")

La commande git branch -d [nom_branch] permet de supprimer une branche (ne pas se positionner sur la branche que l'on souhaite supprimer)

\$ git branch -d [nom_branch] (suppression en local)

\$ git push origin --delete [nom_branch] (suppression sur le dépôt distant)

#Un commit est constitué des infos suivantes:

-identifiant SHA-1 unique de 40 caractères

-Un ensemble de mods

-commentaire décrivant le commit

-les infos sur l'auteur

-une date de création

-une liste (SHA-1) de son ou de ses parents

#Le fichier .gitignore contient la liste des fichiers dans notre dépôt local que l'on souhaite ignorer lors des commit

AWS

lundi 9 septembre 2019 09:34

9:30		[users] Cloud Computing Services – Room Tera	
•	State of "compute" services: Data Centers, edge, VMs, containers, serverless, automation, hardware acceleration		
•	RGPD / where are my data, what security?		
•	1h presentation, 15' discussion, 15' break		