**AWS – Amazon Web Service**

Contents

[ **Bastion Server** 2](#_Toc507611375)

1. [Our Bastion IP 2](#_Toc507611376)

[ **CIT - Server details** 2](#_Toc507611377)

1. [Bastion IP 2](#_Toc507611378)

[ **UAT - Server details** 3](#_Toc507611379)

1. [Bastion IP 3](#_Toc507611380)

[ **To run any Scripts:** 3](#_Toc507611381)

[ **To know the permission of any files:** 3](#_Toc507611382)

[ **To give permission to any file** 3](#_Toc507611383)

[ **To Debug the file** 4](#_Toc507611384)

[ **To do Sudo su** 4](#_Toc507611385)

[ **To Exit** 4](#_Toc507611386)

[ **To Kill the Port:** 5](#_Toc507611387)

[ **To move one folder** 5](#_Toc507611388)

[ **To run the mock service in QA sever:** 5](#_Toc507611389)

[ **To list the Databases:** 5](#_Toc507611390)

[ **To make the schema as default schema:** 6](#_Toc507611391)

[ **To list the Tables:** 6](#_Toc507611392)

[ **To Execute the Stored Procedure:** 6](#_Toc507611393)

[ **SQL QUERIES** 6](#_Toc507611394)

[ **Secure Shell** (SSH) 7](#_Toc507611395)

[ **Secure copy Protocol or SCP** 9](#_Toc507611396)

[ **How to create id\_rsa file for any one?** 9](#_Toc507611397)

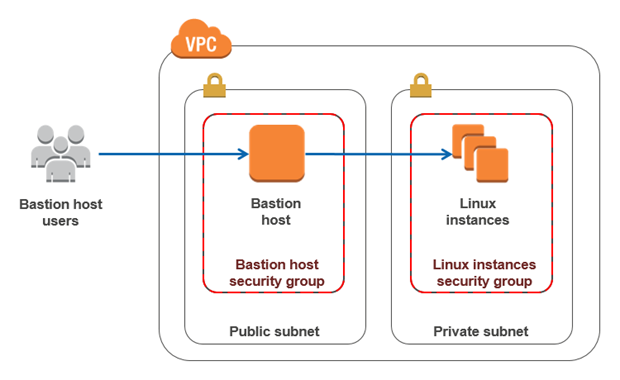
[ What is WinSCP? 10](#_Toc507611398)

[ Open VPN IP: 11](#_Toc507611399)

* **Bastion Server** (Landing server)

Bastion hosts are instances that sit within your public subnet and are typically accessed using SSH or RDP. Once remote connectivity has been established with the bastion host, it then acts as a **‘jump’ server,** allowing you to use SSH or RDP to log in to other instances (within private subnets) deeper within your VPC. When properly configured through the use of security groups and Network ACLs (NACLs), the bastion essentially acts as a bridge to your private instances via the internet.

Our Bastion IP = 13.228.127.243



Expansions:

SSH - *Secure Shell*

RDP - Remote Desktop *Protocol*

VPC - Virtual Private Cloud (Amazon VPC)

ACL - Access control list

NACL - Network ACL

# **CIT - Server details**

Bastion IP = 13.228.127.243

Login

Username: balamurugan.m

Passphrase: xxxxxxxxxx

Application server URL:

ssh balamurugan.m@cit-ap-app-01.sg.bimamobile.internal

Db server URL:

ssh balamurugan.m@cit-ap-app-01.sg.bimamobile.internal

mysql -h cit-ap-rds-01.sg.bimamobile.internal -P 3306 -u mipadmin -pTG47g7NyeabzzyXnXDAw

ESB Server URL:

ssh balamurugan.m@cit-ap-esb-01.sg.bimamobile.internal

QA server URL:

ssh balamurugan.m@ap-qa-01.sg.bimamobile.internal

```````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````

* **UAT - Server details:**

Bastion IP: 13.228.127.243

Login

Username: balamurugan.m

Passphrase: xxxxxxxxxx

Application server

ssh balamurugan.m@uat-ap-app-01.sg.bimamobile.internal

Db server

ssh balamurugan.m@uat-ap-app-01.sg.bimamobile.internal

mysql -h uat-ap-rds-01.sg.bimamobile.internal -P 3306 -u mipadmin -pTG47g7NyeabzzyXnXDAw

ESB Server

ssh balamurugan.m@uat-ap-esb-01.sg.bimamobile.internal

````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````

# **To run any Scripts:**

* login to ESB server
* type command “**sudo su – bimaapp**” (bimaapp is the user which has more access permission)
* The Scripts will be present in the below mentioned path:

cd /opt/scripts

* Command to run the Scripts:

**./<shell script.sh>**

**````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

# **To know the permission of any files:**

* Command **= ls -l**

**````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

# **To give permission to any file**

* Command**:**

**chmod 750 <filename>.sh**

*(Note: Login to bimaapp user before giving permission)*

* 4 read (r)
* 2 write (w)
* 1 execute (x)

7 = 4+2+1 (read/write/execute)

6 = 4+2 (read/write)

5 = 4+1 (read/execute)

4 = 4 (read)

3 = 2+1 (write/execute)

2 = 2 (write)

1 = 1 (execute)

**````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

# **To Debug the file**

* Command

**sh -x <scriptname>.sh**

**````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

# **To do Sudo su**

Steps:

* Login to specific server with your credentials
* Type "**sudo su - bimaapp**"
* (Now u will login to bimaapp login , not your login)
* Now execute what u want, with more access permissions

Example:

balamurugan.m@fiji-esb-01:~$ **sudo su - bimaapp**

**bimaapp@fiji-esb-01:~$** cd /opt/scripts

bimaapp@fiji-esb-01:/opt/scripts$ xxxxxxxxxxxxx

**````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

# **To Exit**

* Command**:**

**Ctrl + C**

**````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

# **To Kill the Port:**

* Command

**lsof -i:(portnumber)**

Example

lsof -i:8088

You will get the information of the port:

Example:

balamurugan.m@ap-qa-01:/opt/SoapUI/bin$ lsof -i:8088

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

java **13546** balamurugan.m 113u IPv6 2157916 0t0 TCP \*:omniorb (LISTEN)

Now we need to kill the PID

* Command

**Kill (PID)**

Example:

Kill 13546

**````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

# **To move one folder**

* Command

**mv <file name> <folder name>**

Example:

Filename = id\_rsa

Folder name = .ssh

Command: **mv id\_rsa .ssh**

**````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

# **To run the mock service in QA sever:**

Steps:

Go to the folder where **mockservicerunner** application was deployed

*Here the mockservicerunner is deployed in QA server in the path “* ***/opt/SoapUI/bin****”*

Command**:** **./mockservicerunner.sh (path where your mock service is present)**

Example

./mockservicerunner.sh /home/balamurugan.m/Soap\_project/FJ\_WIN\_BACK\_CR/FIJI-prepost0-chargingGateway-soapui-project.xml

**````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

**Data Base AWS Information**

# **To list the Databases:**

show databases;

# **To make the schema as default schema:**

Use <databases>

# **To list the Tables:**

Show Tables;

# **To Execute the Stored Procedure:**

Step 1: Edit the stored procedure given by dev team.

*Add*

DELIMITER $$

DROP PROCEDURE IF EXISTS <your procedure name>$$

*Remove local host , user name from the content (proceedure content)*

END$$

(Note : already END statement will be there just add $$ at the end )

Step2: Transfer to cit\_ap\_app (Using WInscp)

Check in the term server whether the the file is there or not by logging in the term server and type “ls” command

Step 3: Run the below command in Landing server to move the file to CIt serevr

scp ASSIGN\_OFFER.sql balamurugan.m@cit-ap-app-01.sg.bimamobile.internal:/home/bimaapp/SQL\_Files/

Step 4 : Run the below command in the CIT server

mysql -h cit-ap-rds-01.sg.bimamobile.internal -P 3306 -u mipadmin -pTG47g7NyeabzzyXnXDAw Your Schema name < /home/bimaapp/Automatoin\_Test/Your stored procedure.sql

*Example:*

mysql -h cit-ap-rds-01.sg.bimamobile.internal -P 3306 -u mipadmin -pTG47g7NyeabzzyXnXDAw mipfj < /home/bimaapp/Automatoin\_Test/ASSIGN\_OFFER.sql

````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````

# **SQL QUERIES**

select \* from customer\_details order by cust\_id desc limit 5;

select \* from customer\_details where cust\_id = 78820;

select \* from customer\_details where msisdn = 7077777;

select \* from offer\_subscription order by offer\_sn\_id desc limit 5;

select \* from offer\_subscription where cust\_id = 78820;

Update offer\_subscription set is\_confirmed = 1, confirmation\_channel\_id = 2, confirmed\_date = '2018-01-23 12:28:36' where offer\_sn\_id =156370;

Update offer\_subscription set is\_confirmed = 1, confirmation\_channel\_id = 2, confirmed\_date = '2018-01-23 12:28:36' where cust\_id = 78820;

Update offer\_subscription set deducted\_offer\_amount = 10 where cust\_id = 78804;

Update offer\_subscription set offer\_cover = 100, hospital\_cover = 100, offer\_charges =1000, is\_deduction\_completed = 1, is\_prev\_deduction\_successful = 1, next\_deduction\_amount = 2, amount\_deducted\_today = 10, cycle\_id = 2018012578804, is\_prior\_month\_purchase\_successful = 1, rolled\_out\_amount = 5, pooled\_date = '2018-01-23 12:28:36' where cust\_id = 78804;

select \* from insured\_relative\_details order by ins\_id desc limit 5;

select \* from insured\_relative\_details where cust\_id = 78820;

select \* from sms\_in\_queue order by sms\_queue\_id desc limit 5;

select \* from sms\_in\_queue where sms\_msisdn = 7077777;

select \* from bima\_cancellations order by bc\_id desc limit 5;

select \* from bima\_cancellations where cust\_id = 78820;

select \* from offer\_cancellations order by oc\_id desc limit 5;

select \* from offer\_cancellations where cust\_id = 78820;

`````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````

# [**Secure Shell**](https://en.wikipedia.org/wiki/Secure_Shell) (SSH)

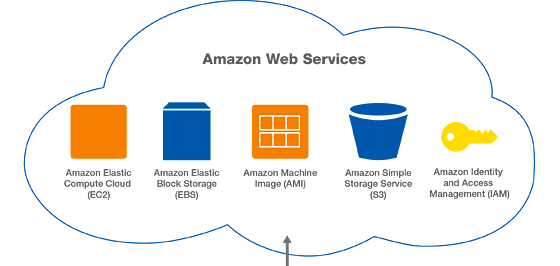
SSH, also known as Secure Socket Shell, is a network protocol that provides administrators with a secure way to access a remote computer.

SSH also refers to the suite of utilities that implement the protocol. Secure Shell provides strong authentication and secure encrypted data communications between two computers connecting over an insecure network such as the Internet.

SSH is widely used by network administrators for managing systems and applications remotely, allowing them to log in to another computer over a network, execute commands and move files from one computer to another.

Reference: <http://searchsecurity.techtarget.com/definition/Secure-Shell>

**`````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**



**```````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

**Amazon VPC:**

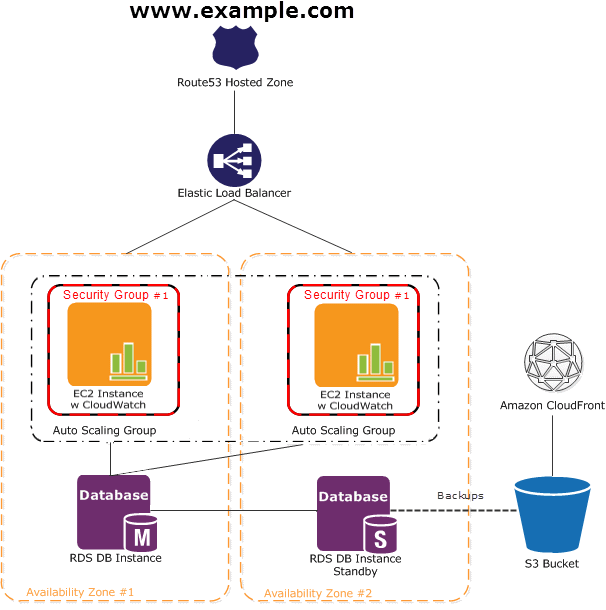
<https://www.youtube.com/watch?v=HgBADK-0oF4>

**````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

**AWS Route 53:**

<https://www.youtube.com/watch?v=JAYNGI3H6Ak>

**````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**



**```````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

# **Secure copy Protocol or SCP**

is a means of securely transferring [computer files](https://en.wikipedia.org/wiki/Computer_file) between a local host and a remote [host](https://en.wikipedia.org/wiki/Server_(computing)) or between two remote hosts. It is based on the [Secure Shell](https://en.wikipedia.org/wiki/Secure_Shell) (SSH) protocol.

"SCP" commonly refers to both the Secure Copy Protocol and the program itself

`````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````

# **How to create id\_rsa file for any one?**

If this file is not kept in the .ssh folder in landing server (term) of each person, then he/she can’t able to move from term server to app/DB/ESB server.

Steps:

Step 1: click on "PUTTYGEN.EXE" file.

Step 2: Select "Load" button

Step 3: Select the already created and save Private Key of the user whom we want to create the id\_rsa file.

Step 4: Enter the Passphrase of the user, in the popup.

Step 5: Select "conversions" menu, which is present in the top.

Step 6: Select "Export OpenSSH key".( second option in the drop down)

Step 7: Save the file with the name "id\_rsa" (the file name should be only **id\_rsa** )

Thats is you have successfully created the id\_rsa file...

Now we have to put this id\_rsa file in to the ".ssh" folder in our landing server.

we can do this with the help of "Winscp" software.

Once you logged in with the Winscp , you will be landing into the root folder.

Move the "id\_rsa" file from your desktop/anyfolder to the root folder (simply drag from left to right).

Now login to landing server(term) and do "cd " command.. you could see your id\_rsa file over there.

Now we need to move this file to the ".ssh" folder.

Use the command "mv id\_rsa .ssh" this will move the file to the ".ssh" folder.

Next, we have to give only Read permission to this file.

So, for that use the command "chmod 400 id\_rsa".

So now the "id\_rsa" file is successfully moved to the .ssh folder, also the read permission is also given.

Check the file by giving "cd .ssh" command.. you could see the file now in .ssh folder..

````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````

# What is WinSCP?

WinSCP is a free and open-source SFTP, FTP, WebDAV, S3 and SCP client for Microsoft Windows. Its main function is secure file transfer between a local and a remote computer.

<https://winscp.net/eng/docs/start>

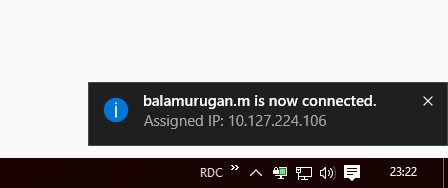
```````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````

LS : command:

<https://www.tecmint.com/15-basic-ls-command-examples-in-linux/>

`````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````

# Open VPN IP:



``````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````