ORACLE Certified Professional

ORACLE E-Business Suite 11*i* Certified Professional

Author – *A.Kishore/Sachin*
http://appsdba.info

Password-less SSH Setup

The term password-less means that SSH authentication is carried out by using public and private keys. Using public/private key authentication with SSH enables SSH logins without requiring passwords interactively and this is known as SSH key-authentication.

There are many reasons why we would want to use password-less SSH service on our Linux systems.Lets for an example, if you are a system administrator and responsible for managing a lot of Linux systems then you probably know the difficulty to remember and provide login information for each different system. Also some of the services on our Linux box (such as back up scripts, cron jobs etc.) may require automatic logins to other systems in order to perform their tasks non-interactively. Password-less SSH configuration can help us with such situations.

Here in this demonstration ,the user “oracle” needs a secure password less access to another user “root” in a server “ebs.com”.

Getting Started:

Step 1: If we want to start fresh with each machine, then remove the ".ssh" directory that resides in the home directory for each machine."cd" to the home directory and remove the ssh directory

```
cd /home/oracle/
rm -r .ssh
```

Step 2 : Generate the public key private key pair by running the below command as user “oracle”

```
$ ssh-keygen -t rsa
```

Please Note : We can run the command ssh-keygen from any directory but the id files will be generated in .ssh dir of user’s home directory.

Step 3 : Change directory to .ssh directory and run the below command and check the key files.

```
$ ls –la
```

We will get the below private and public keys.
id_rsa
id_rsa.pub
known_hosts

Step 4 : Copy the rsa public key to the remote server.

scp id_rsa.pub root@ebs.com:/home/root/.ssh

Please Note : If .ssh directory is not present in remote server then we can create the directory and use scp to copy the public key to remote server.

Step 5 : Login to the remote host with the password.

Once file is copied over, login to the remote host using ssh with password and go to .ssh directory under user home directory.

$ ssh root@ebs.com
$ cd .ssh

Step 6 : Rename the public key file to authorized_keys.

If the authorized_keys file already exists then append the new keys to the existing file as shown below.
cat id_rsa.pub >> authorized_keys

Step 7 : Change the key file and directory permissions.

$ chmod 600 authorized_keys
$ cd ..
$ chmod 700 .ssh
$ logout

Step 8 : Try the ssh connection to server "ebs.com" from local machine as oracle user.

$ ssh root@ebs.com
$ pwd
/home/root

It will not ask for any password anymore for "root" user.