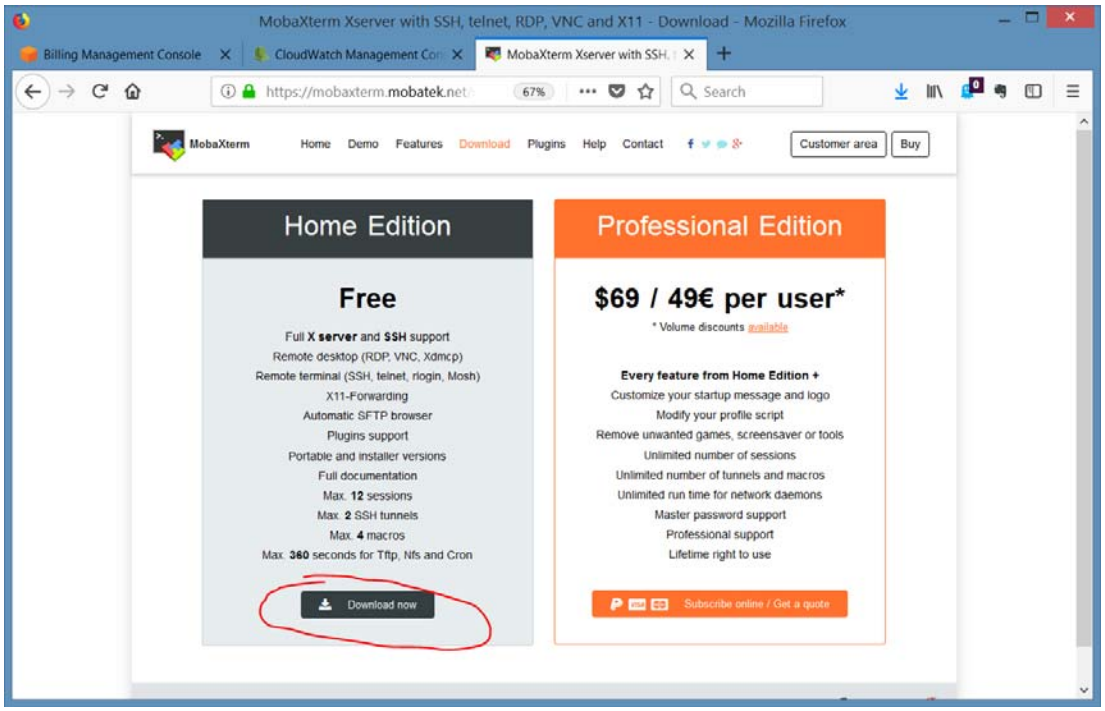
# Connecting to the Linux server using MobaXterm and preparing the server for Oracle installation

**Updated:** 8/24/2019 3:55 PM
**Author:** Peter Wolcott

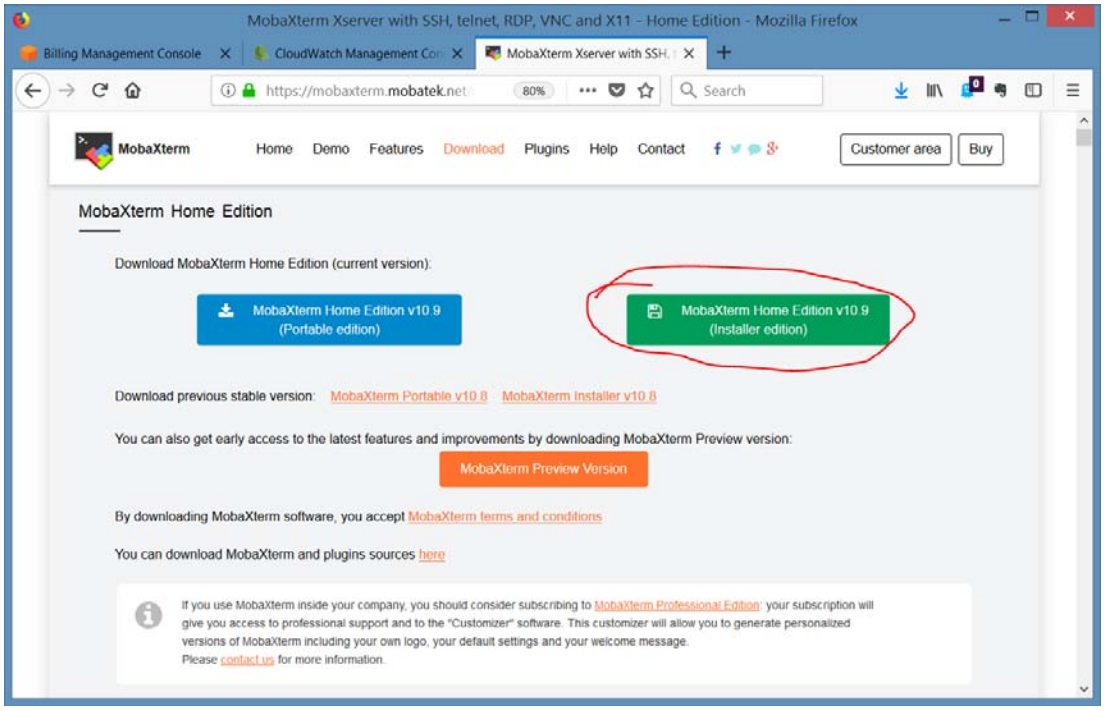
Acknowledgments: Thanks to Arthur Dayton for providing the instructions to carry out these tasks.

Previous: Provisioning an Oracle Server on Amazon AWS

Downloading and installing MobaXterm
http://mobaxterm.mobatek.net/download.html

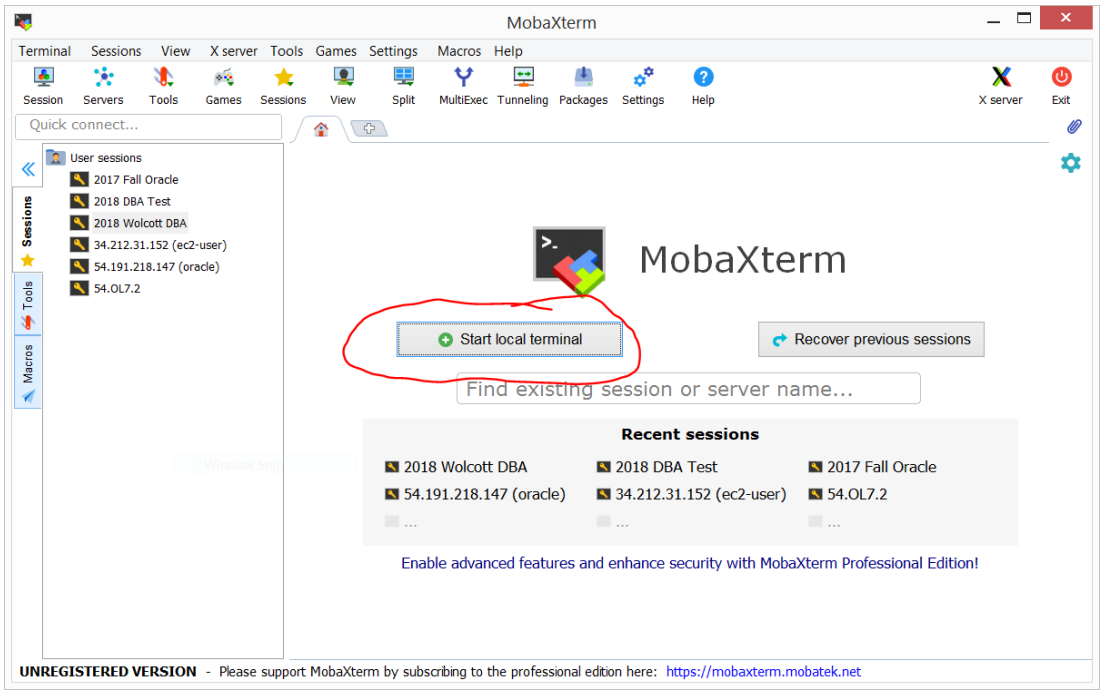



Click on 'Download now'

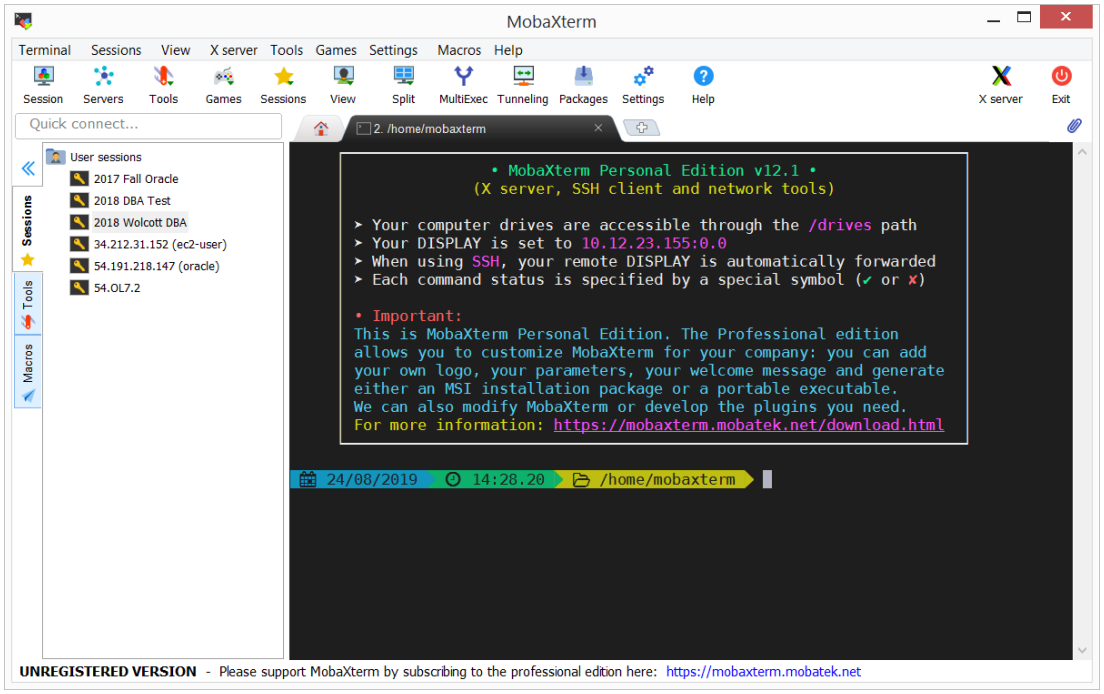



I'm going to choose the installer edition.  Extract all of the files, click on the .msi file to install.

Start MobaXterm




I have some previous sessions, which you won't have.  Click on 'Start local terminal'




We are going to do two things: (1) Prepare the server environment so that it can accept an Oracle installation, and (2) set up a desktop on the server to make it usable.

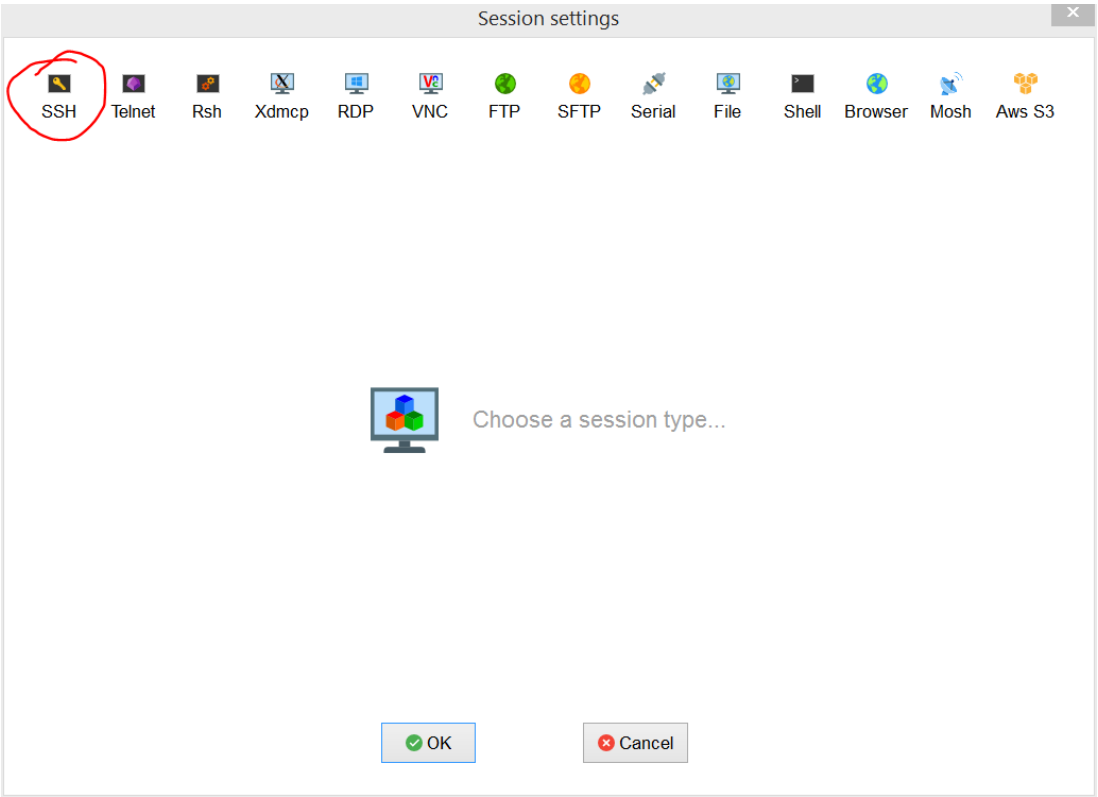
**PREPARE THE SERVER ENVIRONMENT**

To prepare the server environment, you will carry out the following steps:

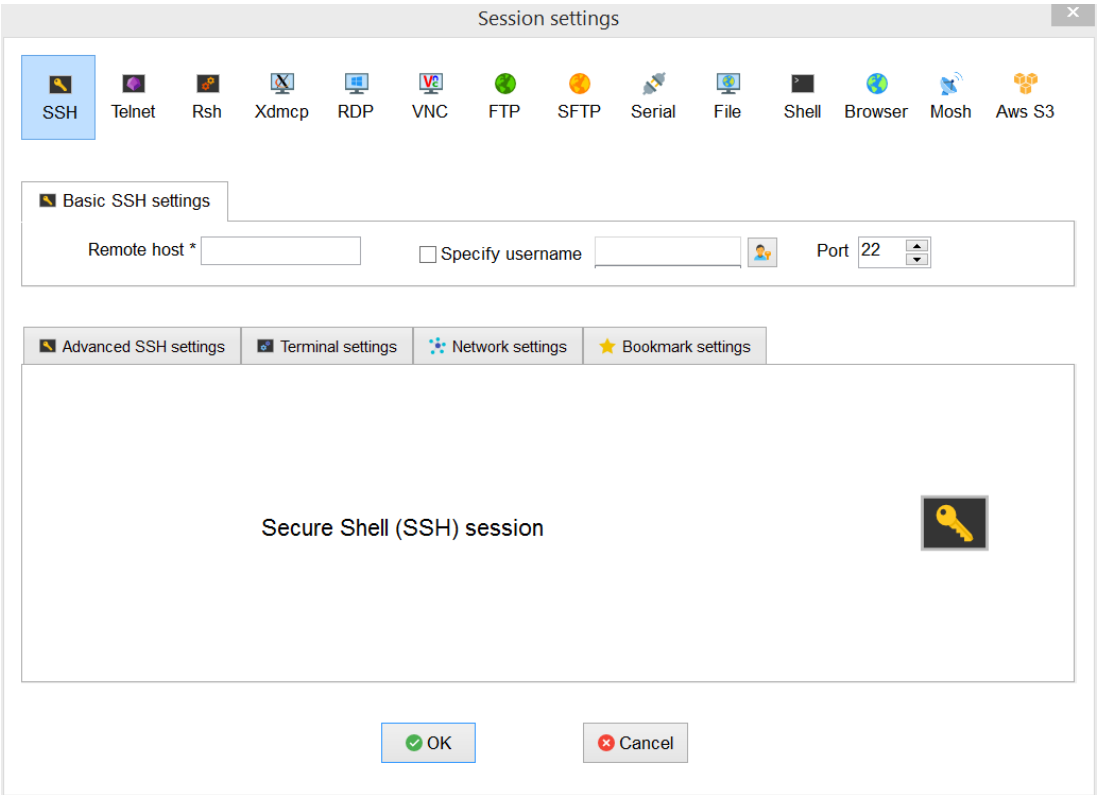
Step #1: Create a session to the server
Step #2: Change passwords
Step #3: Check for updates
Step #4: Install some packages to make life easier
Step #5: Create a user to be used for the Oracle installation
Step #6: Attach storage to the machine
Step #7: Change the ownership of some directories

**Step #1: Create a session to the server**

From your MobaXterm window, click on the Session icon in the upper-left corner 

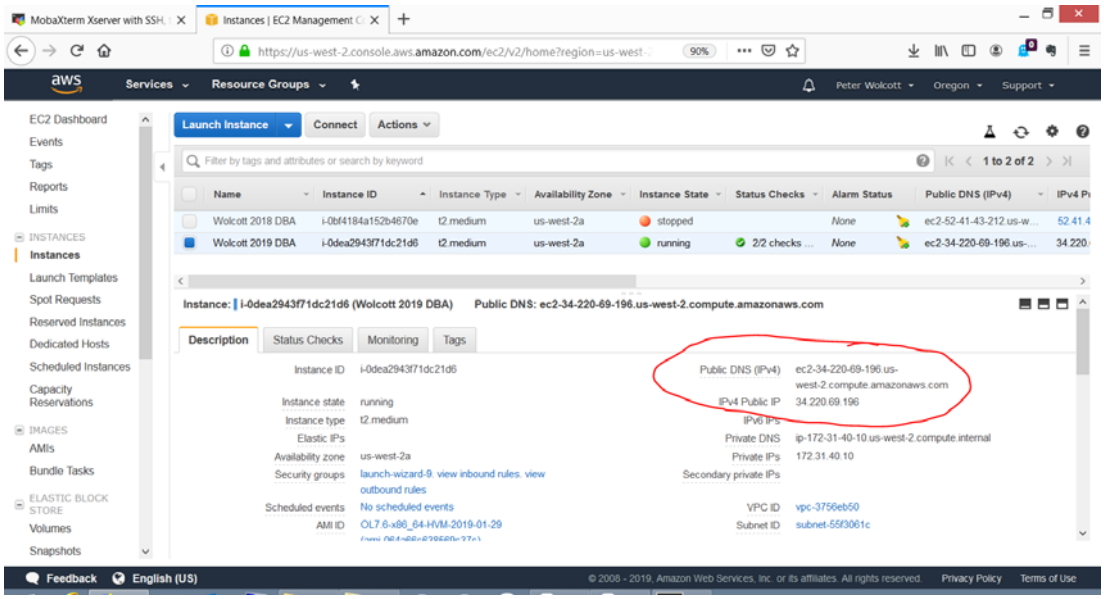



Click on 'SSH'. SSH stands for 'Secure SHell' and is a program from SSH Communications Security Ltd. that enables a login to another computer in a manner that protects the session from IP spoofing, IP source routing, and DNS spoofing. SSH encrypts an entire login session, including transmission of passwords.

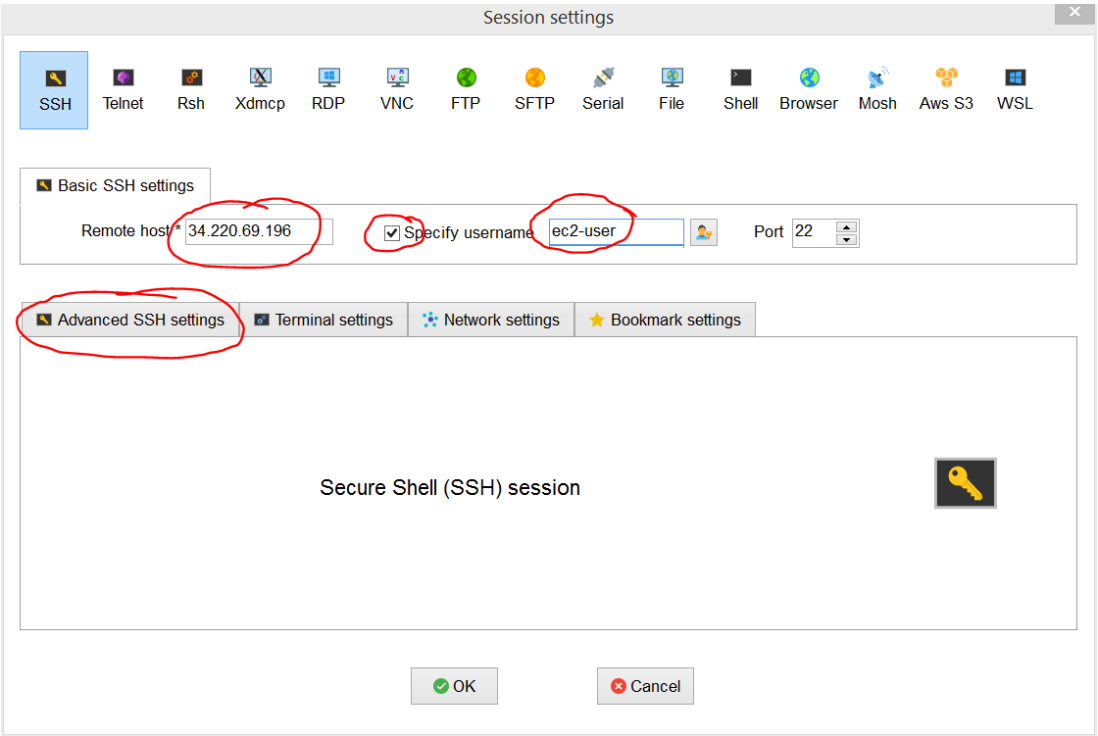



Now, specify the information needed to connect to your server. The *Remote host* information is the Public DNS or Public IP information found on your Instance information page in Amazon AWS. Copy and paste either of the two pieces of information into *Remote host.* Here I am

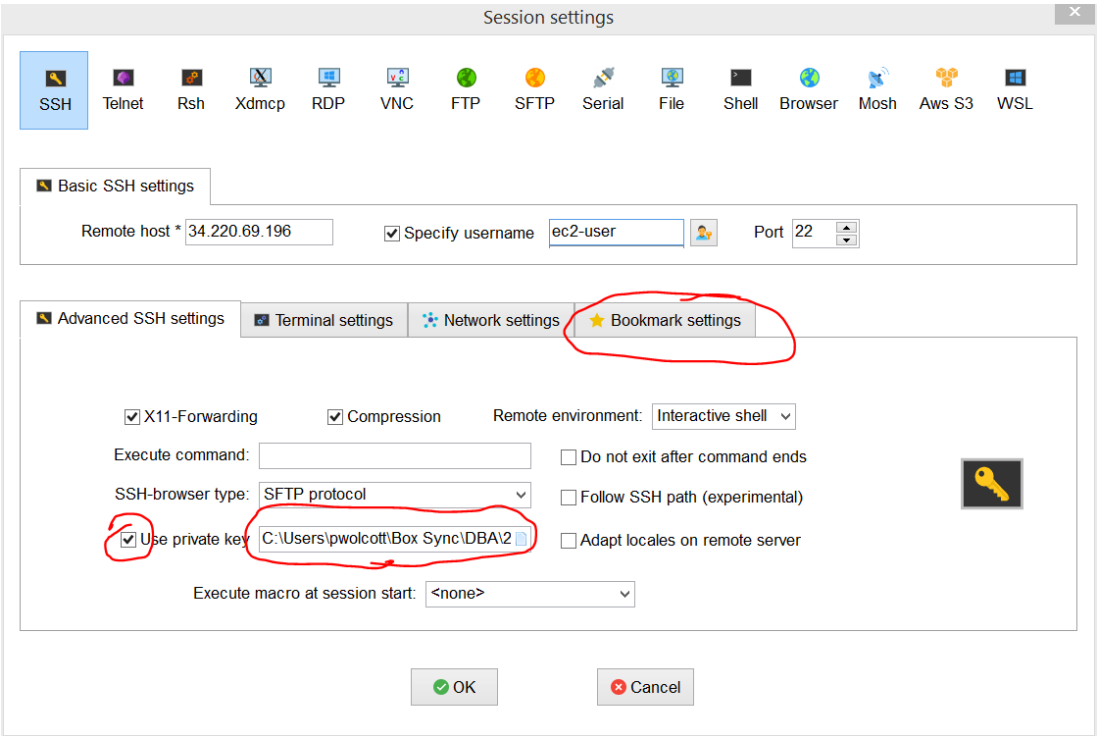
showing the information for MY server. You will have to put in the information for YOUR server, of course.



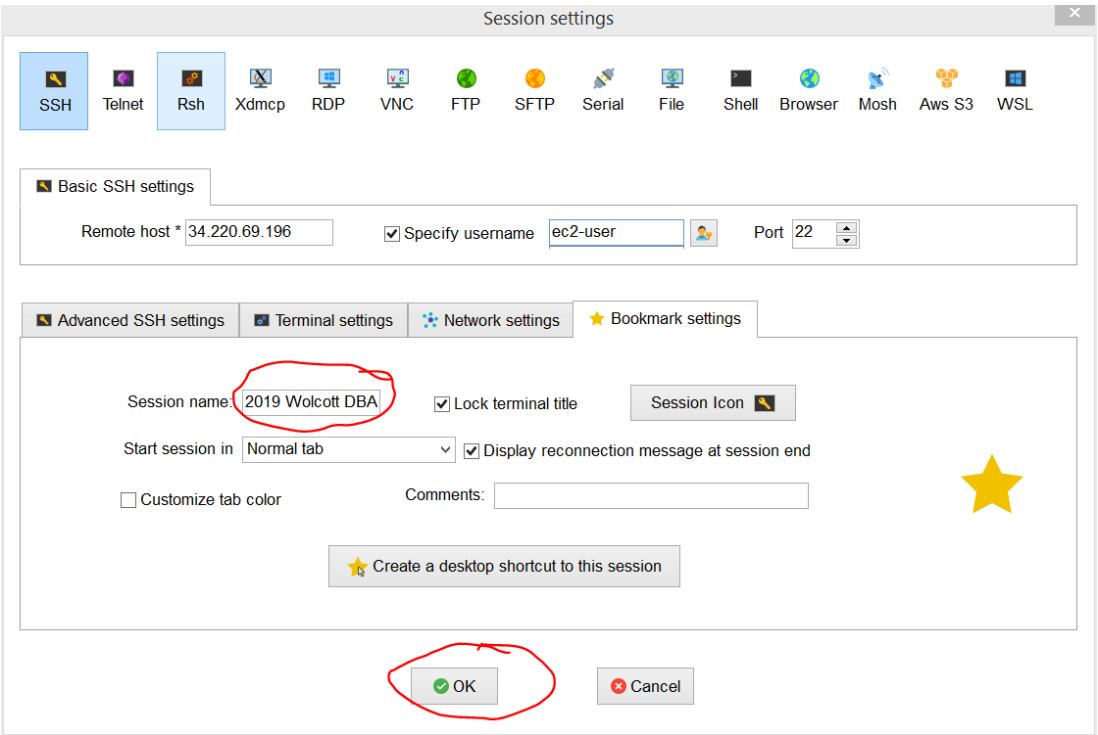

Enter in the Remote host information, and specify the username. The default username for the instance is *ec2-user*



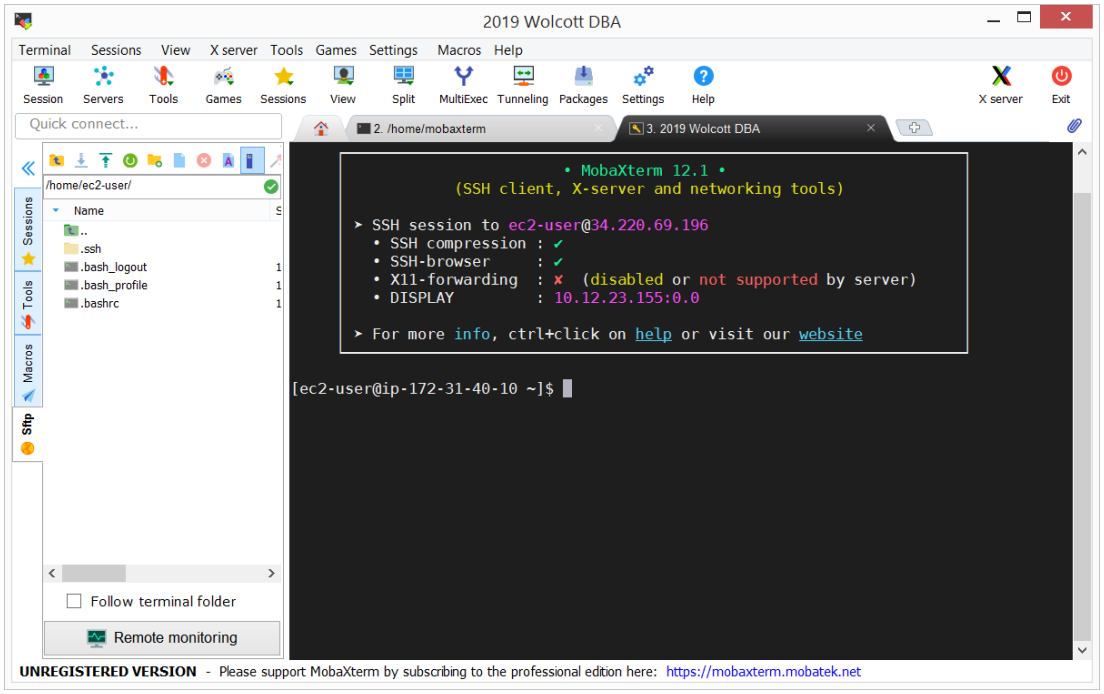

Next, click on 'Advanced SSH settings'. Here we will specify the private key that was generated earlier. Click the check box by 'Use private key' and enter in the path name to the .pem file you saved earlier.




Finally, click on 'Bookmark settings' so that you can give the session a name you can easily recognize it by. Enter in something sensible in the 'Session name:' field.




Click on 'OK'. This will initiate the creation of a session to your server.




**Step #2: Change passwords**

The ec2-user is the super (root) user that allows us to perform all of the necessary operations on the server. We'll change this password first, using the command
$ sudo passwd <username>

sudo is a command to run another command as the superuser.

```
[ec2-user@ip-172-31-40-10 ~]$ sudo passwd ec2-user
Changing password for user ec2-user.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-40-10 ~]$
```

**Step #3: Check for updates**

Use the utility *yum* to check for updates. *yum,* which stands for 'Yellowdog Updater, Modified' is an open source tool for automatically managing updates of software using the RPM (RPM Package Manager) package management system. RPM verifies the authorship and integrity of software.

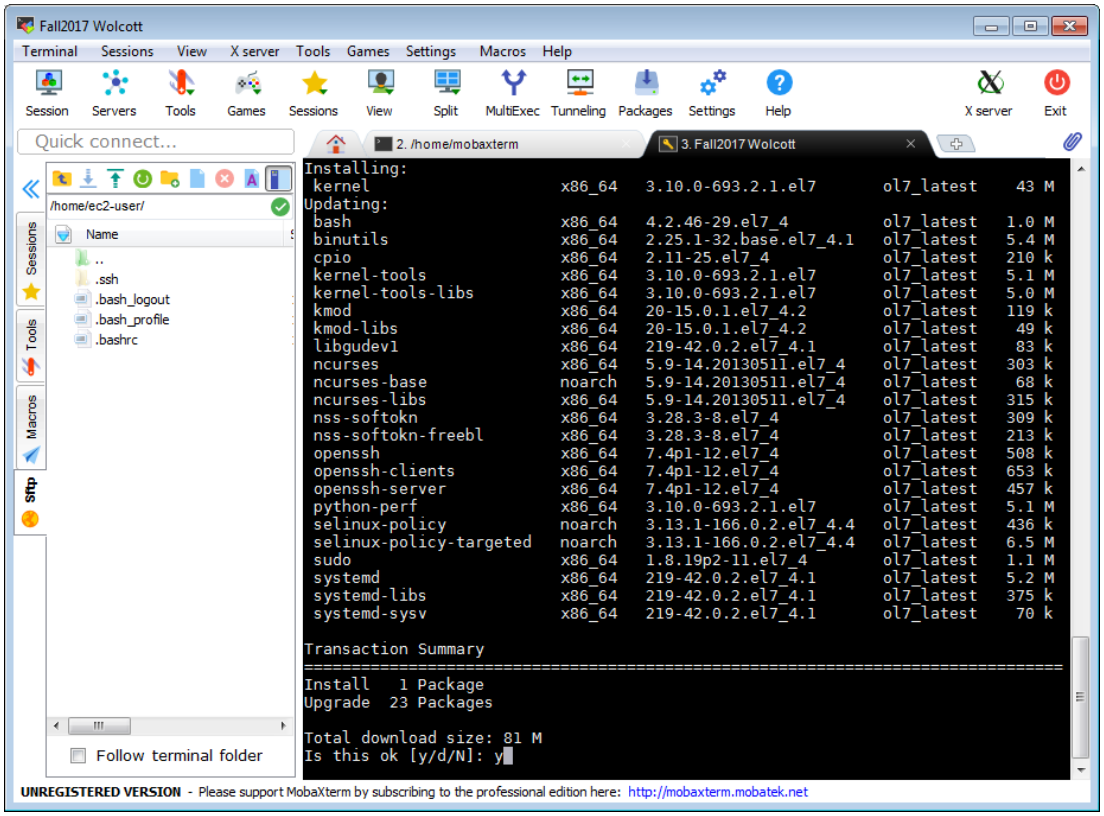
Use the command: `sudo yum update`

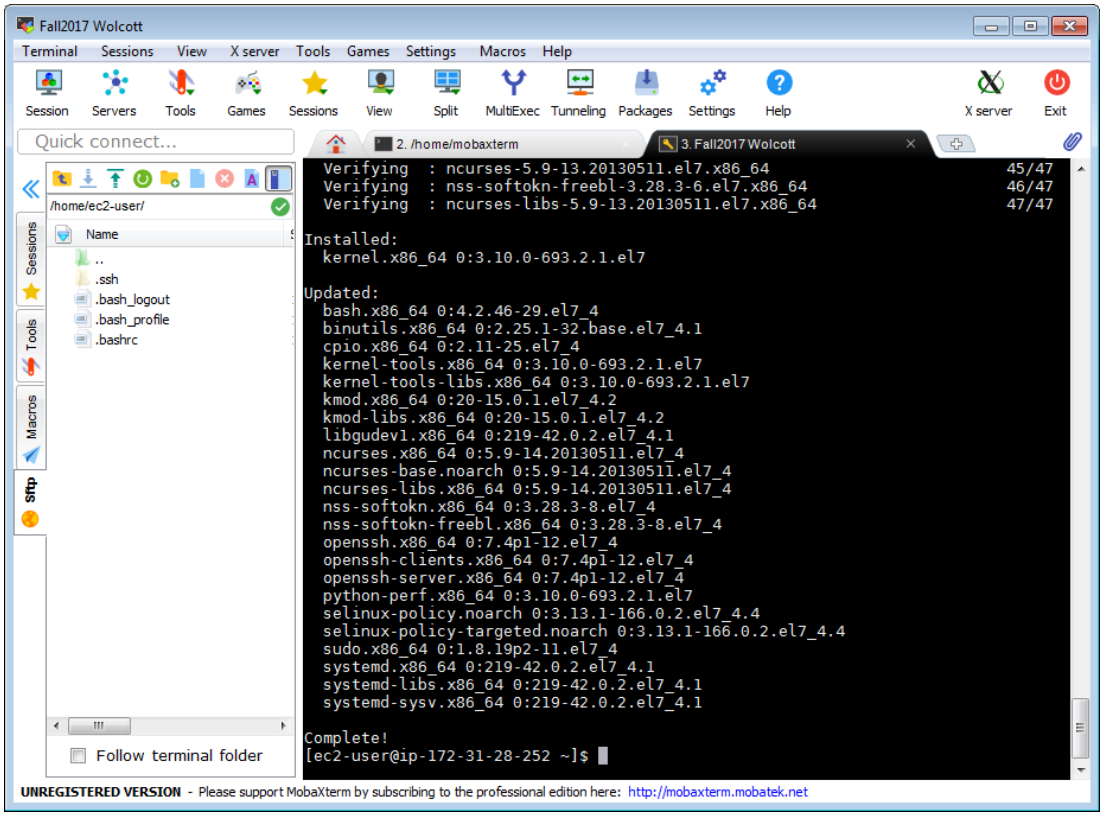
It is possible that your server is already updated, in which case no updates will be found.

If there are updates, you may see something like this:

```
[ec2-user@ip-172-31-40-10 ~]$ sudo yum update
Loaded plugins: ulninfo
ol7_UEKR5                                                | 2.5 kB  00:00:00
ol7_addons                                               | 2.5 kB  00:00:00
ol7_latest                                               | 2.7 kB  00:00:00
(1/2): ol7_latest/x86_64/updateinfo                      | 1.1 MB  00:00:00
(2/2): ol7_latest/x86_64/primary_db                      |  19 MB  00:00:00
No packages marked for update
[ec2-user@ip-172-31-40-10 ~]$
```

Or in past years, I've experienced this:

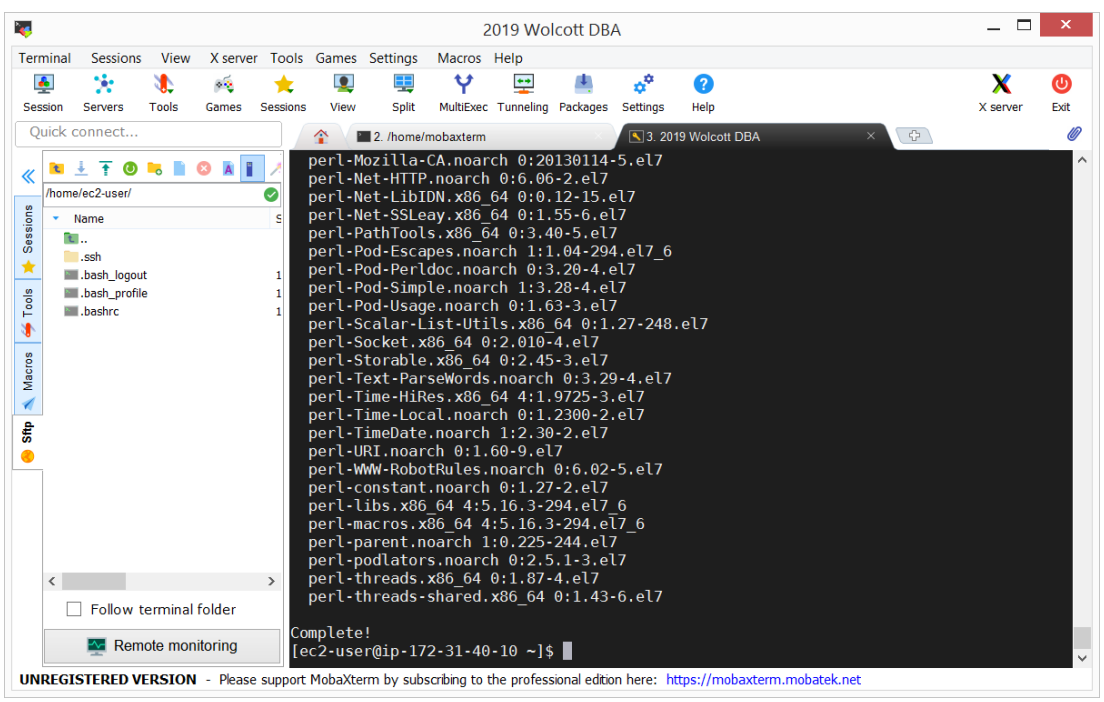





## Step #4: Install some packages to make life easier

`sudo yum install wget zip unzip -y` -- wget is a utility for communicating via HTTP

`sudo yum install perl-libwww-perl.noarch -y` -- perl will be needed by some of the other utilities we use later




## Step #5: Create a user to be used for the Oracle installation

Oracle likes to have a particular user and group with particular permissions to do the installation. We will create these next. While these can be created manually, Oracle has provided scripts that can do this, making the process much easier.

First, let's take a look at the users and groups that already exist.

We'll use a Linux command *getent* that is able to get entries from particular Linux text files. Two of these, passwd and group, store details on users and groups. We'll also use the command *grep* (which stands for **g**lobally search a **r**egular **e**xpression and **p**rint) to read through the entries and search for those that satisfy an expression. In this case, we'll be search for those entries in the password file that have a /home directory. We'll also be searching the group file for entries that have a four-digit group number. The pipeline operator ('|') takes the output of one command and sends it as the input to another command.

`getent passwd | grep "/home"` : List the users in the passwd file and use grep to find the ones that have a '/home' directory (the non-system users)

```
[ec2-user@ip-172-31-40-10 ~]$ getent passwd | grep "/home"
ec2-user:x:1000:1000:Cloud User:/home/ec2-user:/bin/bash
[ec2-user@ip-172-31-40-10 ~]$
```

Here, there is only the ec2-user

`getent group | grep [1-9][0-9][0-9][0-9]`

"[1-9][0-9][0-9][0-9]" is a regular expression that specifies a four digit number whose leading digit is not zero (1-9)

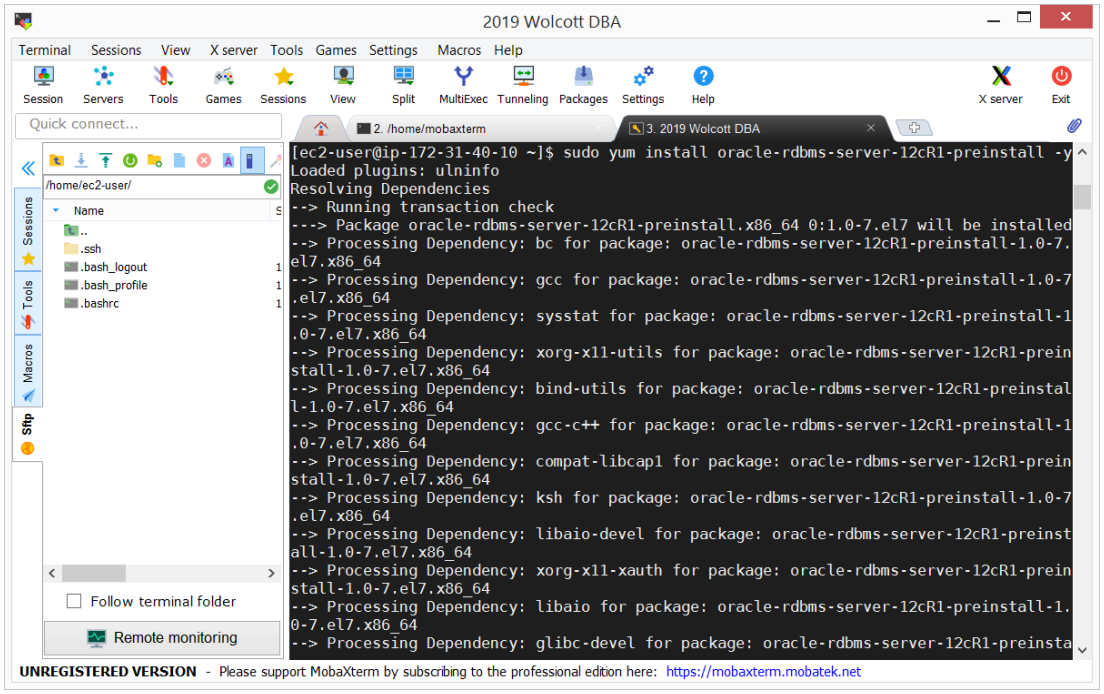
```
[ec2-user@ip-172-31-40-10 ~]$ getent group | grep [1-9][0-9][0-9][0-9]
ec2-user:x:1000:
[ec2-user@ip-172-31-40-10 ~]$
```

Here, there is only one group with a four-digit group identifier (one non-system group)

How, we will create the user and groups we need for Oracle.

We will again use *yum* to find and install the packages that we'll use to create the users and groups needed for the Oracle 12cR1 DBMS. (No package is yet available for Oracle 12cR2....Perhaps later).

`sudo yum install oracle-rdbms-server-12cR1-preinstall -y`




This preinstall script installs a *lot* of software necessary for your Oracle installation....

We can now see that we have an Oracle user.

```
[ec2-user@ip-172-31-40-10 ~]$ getent passwd | grep "/home"
ec2-user:x:1000:1000:Cloud User:/home/ec2-user:/bin/bash
oracle:x:54321:54321::/home/oracle:/bin/bash
[ec2-user@ip-172-31-40-10 ~]$
```

We can also see that three new groups have been created.

```
[ec2-user@ip-172-31-40-10 ~]$ getent group | grep [1-9][0-9][0-9][0-9]
ec2-user:x:1000:
nfsnobody:x:65534:
oinstall:x:54321:
dba:x:54322:oracle
[ec2-user@ip-172-31-40-10 ~]$
```

Now, we need to change the password on the Oracle user

```
sudo passwd oracle
```

```
[ec2-user@ip-172-31-40-10 ~]$ sudo passwd oracle
Changing password for user oracle.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-40-10 ~]$
```

Now, you can switch users, using the *su* (switch user) command.
Go to the home directory using the *cd* (change directory) command. The '~' takes you to the home.
The *pwd* (present working directory) command shows you which directory you are in.

```
[ec2-user@ip-172-31-40-10 ~]$ su oracle
Password:
[oracle@ip-172-31-40-10 ec2-user]$ cd ~
[oracle@ip-172-31-40-10 ~]$ pwd
/home/oracle
[oracle@ip-172-31-40-10 ~]$
```

To go back to the ec2-user, you can use the *exit* command

```
[oracle@ip-172-31-40-10 ~]$ exit
exit
[ec2-user@ip-172-31-40-10 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-40-10 ~]$
```

We now need to enable the oracle user to connect to the the system remotely. We need to provide a key file to associate with our private key. To do this we

1. switch to the oracle user and go into the oracle home directory (important) (su command)
2. create a location for the public key file needed to connect (mkdir - make directory)
3. modify the permissions of the directory (chmod - change mode, used to change the access permissions)
4. create an authorized key file in the directory (touch - a utility to change timestamps, but can create an empty file)
5. set permissions on the file (chmod)
6. use an AWS utility to copy the public key to our authorized_keys file (GET from an AWS service and use the '>' operator to put the results in our file; the '0' (zero) in the command indicates public key 0, which is my-public-key. openssh-key says to return the key in the OpenSSH key format. )

```
GET http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key>.ssh/authorized_keys
```

```
[ec2-user@ip-172-31-40-10 ~]$ su oracle
Password:
[oracle@ip-172-31-40-10 ec2-user]$ cd ~
[oracle@ip-172-31-40-10 ~]$ pwd
/home/oracle
[oracle@ip-172-31-40-10 ~]$ mkdir .ssh
[oracle@ip-172-31-40-10 ~]$ chmod 700 .ssh
[oracle@ip-172-31-40-10 ~]$ touch .ssh/authorized_keys
[oracle@ip-172-31-40-10 ~]$ GET http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key>.ssh/authorized_keys
[oracle@ip-172-31-40-10 ~]$
```

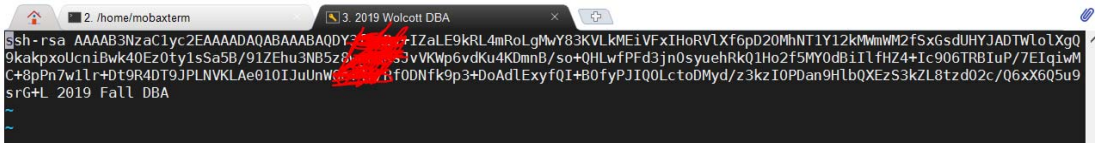
In order to verify what we have, use the *ls* (list directories and files) and *ls -a* (list all directories and files, including hidden ones, like .ssh).

```
[oracle@ip-172-31-40-10 ~]$ ls
[oracle@ip-172-31-40-10 ~]$ ls -a
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc  .kshrc  .ssh
[oracle@ip-172-31-40-10 ~]$ cd .ssh
[oracle@ip-172-31-40-10 .ssh]$ ls
authorized_keys
[oracle@ip-172-31-40-10 .ssh]$
```

Open the authorized_keys file using the *vi* editor. vi is a very basic, common command line editor (See, for example, http://www.yolinux.com/TUTORIALS/LinuxTutorialAdvanced_vi.html).

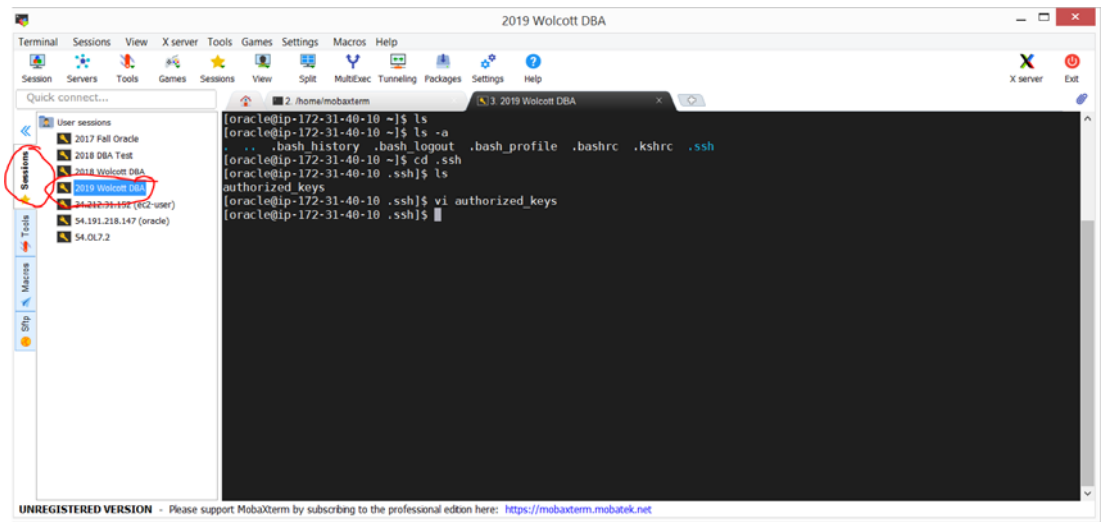
```
[oracle@ip-172-31-40-10 ~]$ ls
[oracle@ip-172-31-40-10 ~]$ ls -a
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc  .kshrc  .ssh
[oracle@ip-172-31-40-10 ~]$ cd .ssh
[oracle@ip-172-31-40-10 .ssh]$ ls
authorized_keys
[oracle@ip-172-31-40-10 .ssh]$ vi authorized_keys
[oracle@ip-172-31-40-10 .ssh]$
```

vi will open and you can see the key value within the file.

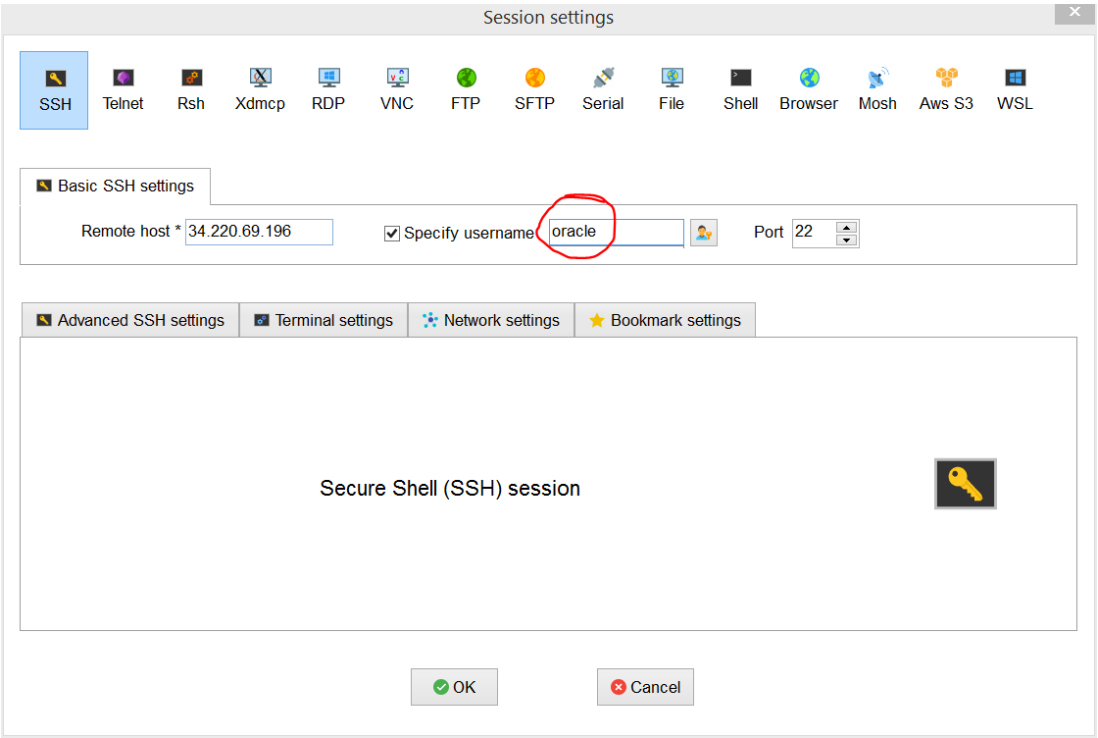



Type ':' to return to the command line, and type 'q' to quit.

Click on the Sessions tab on the left of your window and right-click on the session. Choose 'Edit session'

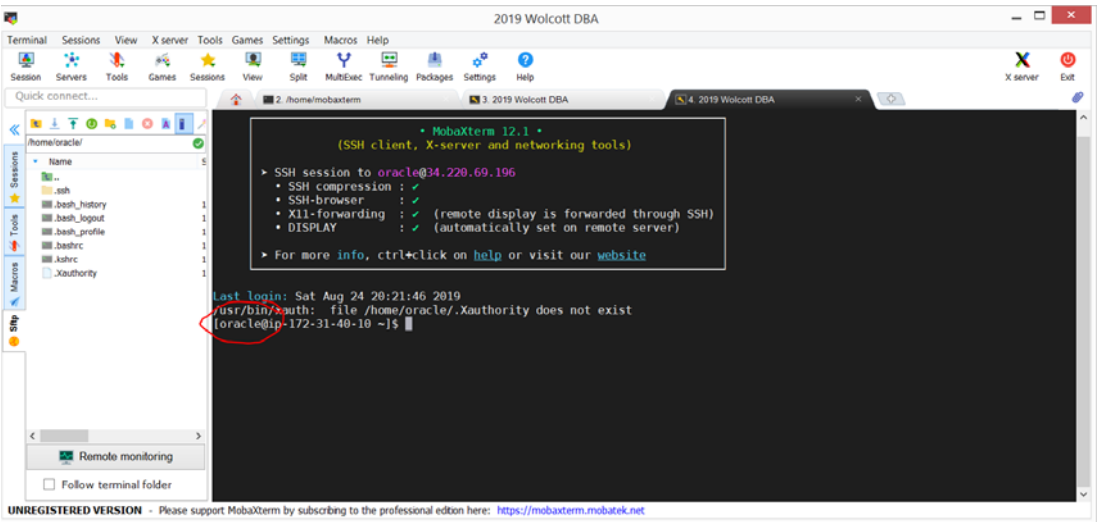



We'll now change the user from ec2-user to oracle




Click ok, and double-click on the session name

You should now see another window open and a connection made with the oracle user.




**Step #6: Attach storage to the machine**

Go back to the tab with your earlier session and type *exit* to get back to the ec2-user.

The command *lsblk* shows the storage that is available to this system.

```
[oracle@ip-172-31-40-10 ~]$ ls
[oracle@ip-172-31-40-10 ~]$ ls -a
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc  .kshrc  .ssh
[oracle@ip-172-31-40-10 ~]$ cd .ssh
[oracle@ip-172-31-40-10 .ssh]$ ls
authorized_keys
[oracle@ip-172-31-40-10 .ssh]$ vi authorized_keys
[oracle@ip-172-31-40-10 .ssh]$ exit
exit
[ec2-user@ip-172-31-40-10 ~]$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdc    202:32   0  10G  0 disk
xvda    202:0    0  15G  0 disk
└─xvda1 202:1    0  15G  0 part /
xvdd    202:48   0  30G  0 disk
xvdb    202:16   0   5G  0 disk
[ec2-user@ip-172-31-40-10 ~]$
```

Notice that none of the blocks has a mountpoint (except xvda1). This means that they are not attached to the machine, and there's no way to use them. They are also raw storage, with no formatting.

The first thing to do is to create the swap drive on xvdb.

`sudo mkswap /dev/xvdb` creates the swap drive.

```
[ec2-user@ip-172-31-40-10 ~]$ sudo mkswap /dev/xvdb
Setting up swapspace version 1, size = 5242876 KiB
no label, UUID=a4274072-596d-4220-857c-5f5122e5e05a
[ec2-user@ip-172-31-40-10 ~]$ free -m
              total        used        free      shared  buff/cache   available
Mem:           3674          81        3080           8         512        3359
Swap:             0           0           0
[ec2-user@ip-172-31-40-10 ~]$
```

The free -m command shows that it hasn't been turned on. There's no free space on it.

use the *swapon* command to turn it on.

`sudo swapon /dev/xvdb`

```
[ec2-user@ip-172-31-40-10 ~]$ sudo swapon /dev/xvdb
[ec2-user@ip-172-31-40-10 ~]$ free -m
              total        used        free      shared  buff/cache   available
Mem:           3674          85        3076           8         512        3355
Swap:          5119           0        5119
[ec2-user@ip-172-31-40-10 ~]$
```

We now format the other two drives as normal file systems

```
sudo mkfs -t ext4 /dev/xvdc
sudo mkfs -t ext4 /dev/xvdd
```

```
[ec2-user@ip-172-31-40-10 ~]$ sudo mkfs -t ext4 /dev/xvdc
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
655360 inodes, 2621440 blocks
131072 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2151677952
80 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[ec2-user@ip-172-31-40-10 ~]$
```

```
[ec2-user@ip-172-31-40-10 ~]$ sudo mkfs -t ext4 /dev/xvdd
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
1966080 inodes, 7864320 blocks
393216 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2155872256
240 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[ec2-user@ip-172-31-40-10 ~]$
```

The storage is now formatted, but it's still not connected to anything.

```
[ec2-user@ip-172-31-40-10 ~]$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdc    202:32   0  10G  0 disk
xvda    202:0    0  15G  0 disk
└─xvda1 202:1    0  15G  0 part /
xvdd    202:48   0  30G  0 disk
xvdb    202:16   0   5G  0 disk [SWAP]
[ec2-user@ip-172-31-40-10 ~]$
```

To make the storage accessible, we are going to "mount" the drives, or associate them with particular directories.

Go back to the highest level directory for the ec2-user (*cd /*) and use the *ls* command to see the directories. Create one more ('inv', for inventory), using the mkdir command.

```
sudo mkdir inv
```

```
[ec2-user@ip-172-31-40-10 ~]$ cd /
[ec2-user@ip-172-31-40-10 /]$ ls
bin  boot  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  u01  usr  var
[ec2-user@ip-172-31-40-10 /]$ sudo mkdir inv
[ec2-user@ip-172-31-40-10 /]$ ls
bin  boot  dev  etc  home  inv  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  u01  usr  var
[ec2-user@ip-172-31-40-10 /]$
```

Here there are two directories, 'u01' and 'inv' that we will attach our drives to.

```
sudo mount /dev/xvdc inv
sudo mount /dev/xvdd u01
```

```
[ec2-user@ip-172-31-40-10 /]$ sudo mount /dev/xvdc inv
[ec2-user@ip-172-31-40-10 /]$ sudo mount /dev/xvdd u01
[ec2-user@ip-172-31-40-10 /]$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdc    202:32   0  10G  0 disk /inv
xvda    202:0    0  15G  0 disk
└xvda1 202:1    0  15G  0 part /
xvdd    202:48   0  30G  0 disk /u01
xvdb    202:16   0   5G  0 disk [SWAP]
[ec2-user@ip-172-31-40-10 /]$
```

Now they are available to use. Files that created in these directories are stored on the associated drives.

Finally, we want to make sure that these drives are mounted automatically when the system starts so that we don't have to manually mount them each time. We will modify a file called fstab which is use to associate storage with directories. When your machine starts up, the system goes there to find out what to mount.

Let's unmount the drives associated with inv and u01 (use the *umount* command - note: it's umount, not unmount), so that we can see more clearly what is going on.

```
sudo umount /dev/xvdc
sudo umount /dev/xvdd
```

```
[ec2-user@ip-172-31-40-10 /]$ sudo umount /dev/xvdc
[ec2-user@ip-172-31-40-10 /]$ sudo umount /dev/xvdd
[ec2-user@ip-172-31-40-10 /]$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdc    202:32   0  10G  0 disk
xvda    202:0    0  15G  0 disk
└xvda1 202:1    0  15G  0 part /
xvdd    202:48   0  30G  0 disk
xvdb    202:16   0   5G  0 disk [SWAP]
[ec2-user@ip-172-31-40-10 /]$
```

Open the file /etc/fstab using *vi* .

```
sudo vi /etc/fstab
```

```
[ec2-user@ip-172-31-40-10 /]$ sudo vi /etc/fstab
```

Type 'i' (insert) to edit.  Position the cursor after the UUID line (use arrows & Enter)

```
#
# /etc/fstab
# Created by anaconda on Tue Nov 27 11:40:36 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=d5388b06-35e3-435e-8eaf-09a08de83883 /                       ext4    defaults        0 0
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

type in the three lines shown, just below the UUID line

```
#
# /etc/fstab
# Created by anaconda on Tue Nov 27 11:40:36 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=d5388b06-35e3-435e-8eaf-09a08de83883 /                       ext4    defaults        0 0
/dev/xvdb none swap defaults 0 0
/dev/xvdc /inv ext4 defaults 0 0
/dev/xvdd /u01 ext4 defaults 0 0
~
~
~
~
~
~
~
~
```

Now press the ESC key (to turn off insert) and :wq to write and quit.

We don't yet see that the drives are mounted.

```
[ec2-user@ip-172-31-40-10 /]$ sudo vi /etc/fstab
[ec2-user@ip-172-31-40-10 /]$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdc    202:32   0  10G  0 disk
xvda    202:0    0  15G  0 disk
└xvda1 202:1    0  15G  0 part /
xvdd    202:48   0  30G  0 disk
xvdb    202:16   0   5G  0 disk [SWAP]
[ec2-user@ip-172-31-40-10 /]$
```

We get them mounted through the fstab file by using the `sudo mount -a` command

```
[ec2-user@ip-172-31-40-10 /]$ sudo mount -a
[ec2-user@ip-172-31-40-10 /]$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdc    202:32   0  10G  0 disk /inv
xvda    202:0    0  15G  0 disk
└xvda1 202:1    0  15G  0 part /
xvdd    202:48   0  30G  0 disk /u01
xvdb    202:16   0   5G  0 disk [SWAP]
[ec2-user@ip-172-31-40-10 /]$
```

We can now see that the drives are mounted correctly.

**Step #7: Change the ownership of some directories**

The last thing to do is to change the ownership of the `u01` and `inv` directories to the oracle user. Oracle requires that the owner of the directories where the software will reside is the oracle user.

If we list the files and directories at present, we can see that the inv and u01 directories are owned by the root user.

```
[ec2-user@ip-172-31-40-10 /]$ ls -l
total 68
lrwxrwxrwx.   1 root root     7 Nov 27  2018 bin -> usr/bin
drwxr-xr-x.   4 root root  4096 Aug 22 21:00 boot
drwxr-xr-x.  18 root root  2980 Aug 24 19:32 dev
drwxr-xr-x.  74 root root  4096 Aug 24 20:50 etc
drwxr-xr-x.   4 root root  4096 Aug 24 20:13 home
drwxr-xr-x.   3 root root  4096 Aug 24 20:37 inv
lrwxrwxrwx.   1 root root     7 Nov 27  2018 lib -> usr/lib
lrwxrwxrwx.   1 root root     9 Nov 27  2018 lib64 -> usr/lib64
drwx------.   2 root root 16384 Jan 29  2019 lost+found
drwxr-xr-x.   2 root root  4096 Apr 11  2018 media
drwxr-xr-x.   2 root root  4096 Apr 11  2018 mnt
drwxr-xr-x.   2 root root  4096 Apr 11  2018 opt
dr-xr-xr-x. 108 root root     0 Aug 24 19:32 proc
dr-xr-x---.   4 root root  4096 Aug 22 20:58 root
drwxr-xr-x.  22 root root   640 Aug 24 20:13 run
lrwxrwxrwx.   1 root root     8 Nov 27  2018 sbin -> usr/sbin
drwxr-xr-x.   2 root root  4096 Apr 11  2018 srv
dr-xr-xr-x.  13 root root     0 Aug 24 19:32 sys
drwxrwxrwt.   8 root root  4096 Aug 24 20:48 tmp
drwxr-xr-x.   3 root root  4096 Aug 24 20:38 u01
drwxr-xr-x.  13 root root  4096 Nov 27  2018 usr
drwxr-xr-x.  18 root root  4096 Nov 27  2018 var
[ec2-user@ip-172-31-40-10 /]$
```

The *chown* command can change the ownership. We want to change the ownership (chown) recursively (R) to the oracle user oinstall group (oracle.oinstall) for the /inv directory. We do the same for the /u01 directory.

```
sudo chown -R oracle.oinstall /inv
sudo chown -R oracle.oinstall /u01
```

```
[ec2-user@ip-172-31-40-10 /]$ sudo chown -R oracle.oinstall /inv
[ec2-user@ip-172-31-40-10 /]$ sudo chown -R oracle.oinstall /u01
[ec2-user@ip-172-31-40-10 /]$ ls -l
total 68
lrwxrwxrwx.   1 root   root         7 Nov 27  2018 bin -> usr/bin
drwxr-xr-x.   4 root   root      4096 Aug 22 21:00 boot
drwxr-xr-x.  18 root   root      2980 Aug 24 19:32 dev
drwxr-xr-x.  74 root   root      4096 Aug 24 20:50 etc
drwxr-xr-x.   4 root   root      4096 Aug 24 20:13 home
drwxr-xr-x.   3 oracle oinstall  4096 Aug 24 20:37 inv
lrwxrwxrwx.   1 root   root         7 Nov 27  2018 lib -> usr/lib
lrwxrwxrwx.   1 root   root         9 Nov 27  2018 lib64 -> usr/lib64
drwx------.   2 root   root     16384 Jan 29  2019 lost+found
drwxr-xr-x.   2 root   root      4096 Apr 11  2018 media
drwxr-xr-x.   2 root   root      4096 Apr 11  2018 mnt
drwxr-xr-x.   2 root   root      4096 Apr 11  2018 opt
dr-xr-xr-x. 108 root   root         0 Aug 24 19:32 proc
dr-xr-x---.   4 root   root      4096 Aug 22 20:58 root
drwxr-xr-x.  22 root   root       640 Aug 24 20:13 run
lrwxrwxrwx.   1 root   root         8 Nov 27  2018 sbin -> usr/sbin
drwxr-xr-x.   2 root   root      4096 Apr 11  2018 srv
dr-xr-xr-x.  13 root   root         0 Aug 24 19:32 sys
drwxrwxrwt.   8 root   root      4096 Aug 24 20:54 tmp
drwxr-xr-x.   3 oracle oinstall  4096 Aug 24 20:38 u01
drwxr-xr-x.  13 root   root      4096 Nov 27  2018 usr
drwxr-xr-x.  18 root   root      4096 Nov 27  2018 var
[ec2-user@ip-172-31-40-10 /]$
```

Next: Installing a desktop and VNC server on our Linux server.