**ORACLE MANAGEMENT CLOUD: DATABASE AGENT DEPLOYMENT / MONITORING:**

**Details:**

In this lab, you will create a database, connect to it, modify it, and prepare it to be monitored using *Oracle Management Cloud*. To do this, we will configure a *Cloud Agent* to the host machine supporting the instance of *Oracle Database*, and then relate the agent back to an instance of *Oracle Management Cloud* to do monitoring. It is critical to carefully follow the steps outlined in this guide to accomplish this. With that, let’s get started!

**Prerequisites:**

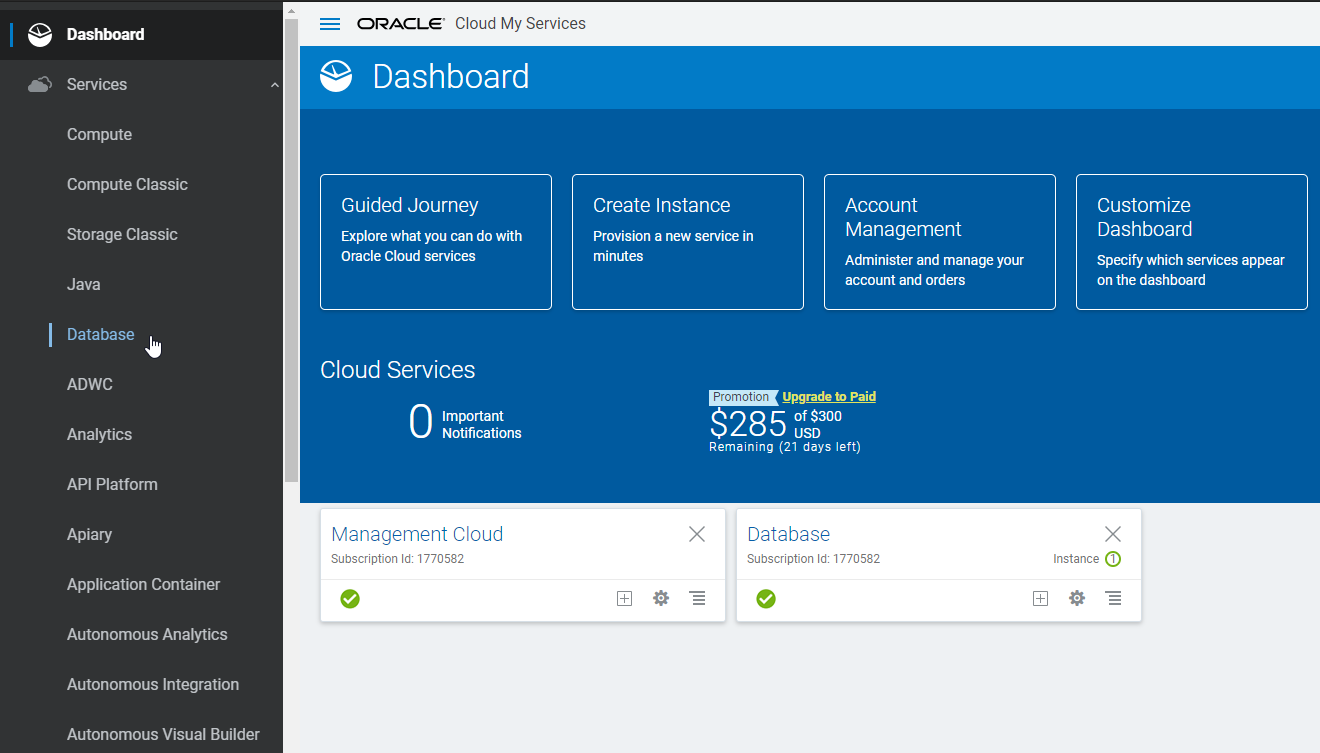
There are some key components you’ll need to have throughout the course of this lab. I’m going to go through this lab assuming you’re using a Windows machine, but I will explain the simple changes to make if you’re using Linux. Since we’re going to be moving files to remote hosts, we’re going to need an SCP client (such as WinSCP or PSCP). We’re also going to an SSH client (my client of choice is PuTTy). And of course, you’ll need an Oracle Cloud Account.

**Pre-Lab Exercise:**

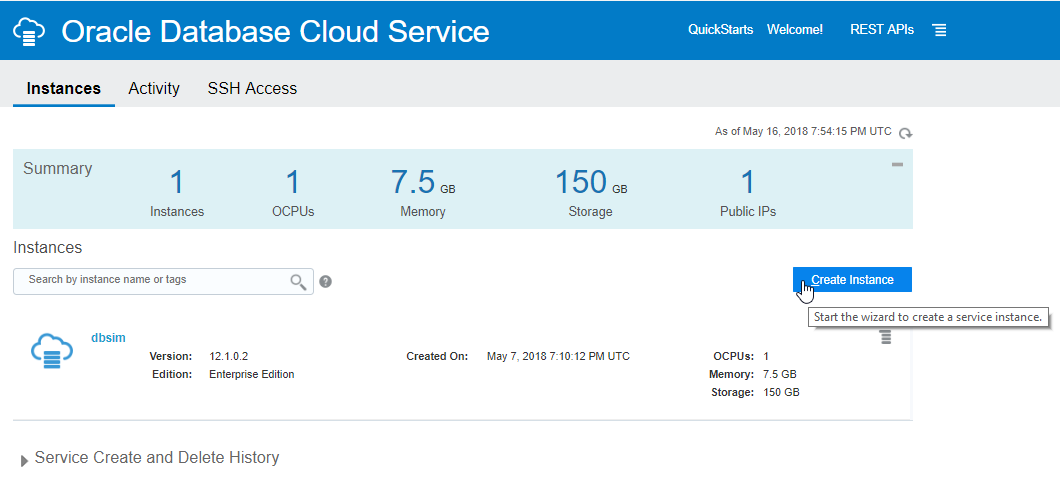
Download: PuTTy, PuTTy Gen, PSCP. (You can download these [here](https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html))Make SSH keys with PuTTy Gen. (You can find a tutorial [here](https://www.youtube.com/watch?v=-92wEg68SKQ)). If you have any trouble with creating the SSH Keys, reach out to [zach.hamilton@oracle.com](mailto:zach.hamilton@oracle.com) for help.

**Creating an instance of Oracle Database Cloud Service:**

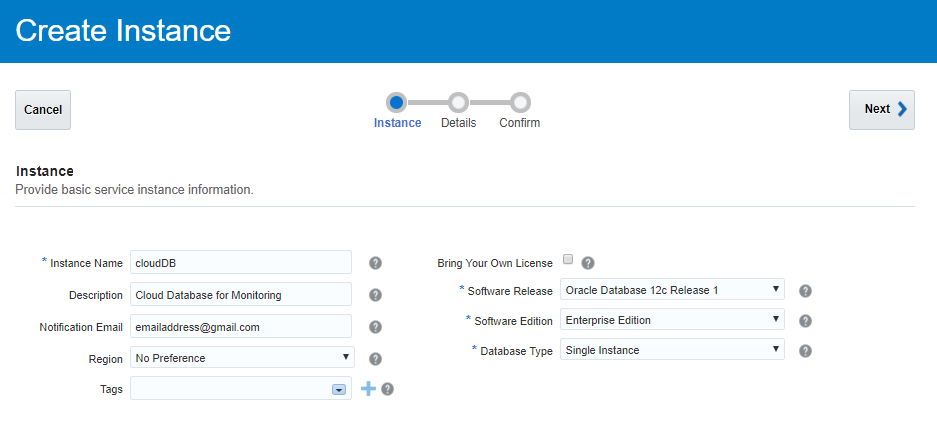
Step 1: From the My Services Dashboard, select the “Action Menu” at the top-left of the page to pop-out the window. Click on the “Services” dropdown and find and click “Database”.



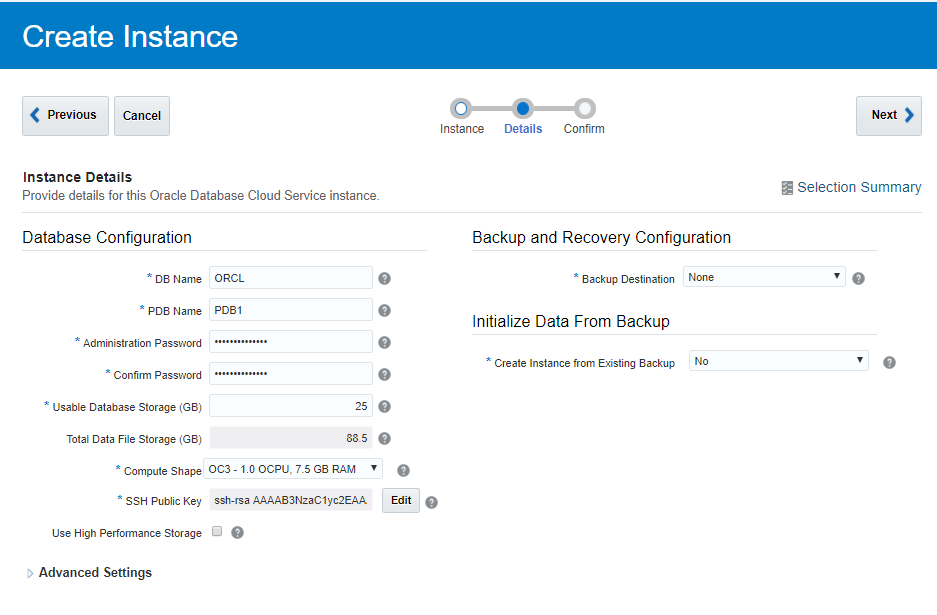
Step 2: On the Database Cloud Service console, select “Create Instance” to create a new instance of Oracle Database Cloud Service.



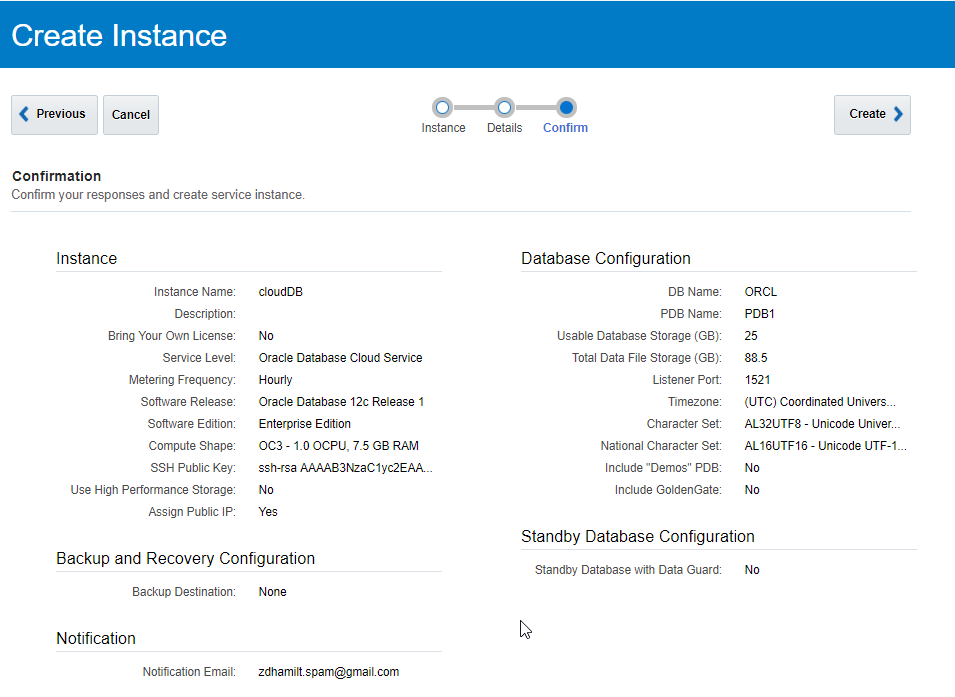
Step 3: Configure the correct options for the database instance you’re going to create. This page only *requires* you to provide a database name. For simplicity and consistency throughout this lab, use “cloudDB”. **Leave the other options unchanged.** When the options are added, click “Next”. (Note: provide a valid email address to get notifications about service maintenance, availability, and more by email about your instance).



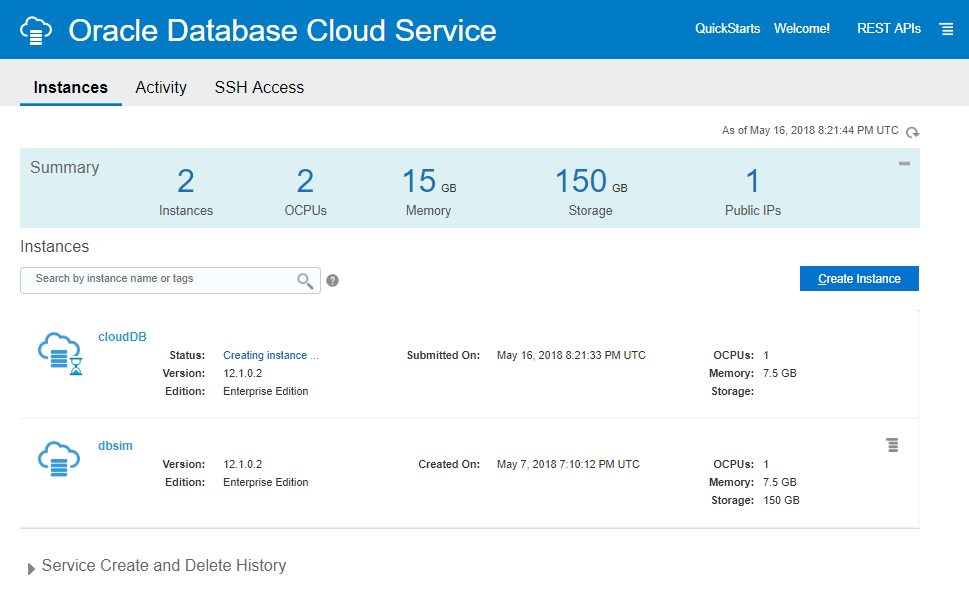
Step 4: On this page you’ll need to configure just a few options. The first is the administrator password. **Take note of this password because you will need to use it later.** After you’ve entered an administrator password, configure the *Public Key* of the SSH keypair you made in the *Pre-Lab Setup*. To do this, click “Edit”, then the radio button “Key Value”, then paste the contents of your *Public Key* into the text box, then click “Enter”. Next, select the “Backup Destination” dropdown and select “None”. When you’ve done that, click “Next” to proceed.



Step 5: On the next page, you will have a summary of all the information from the database you’re about to create. It should look like such:



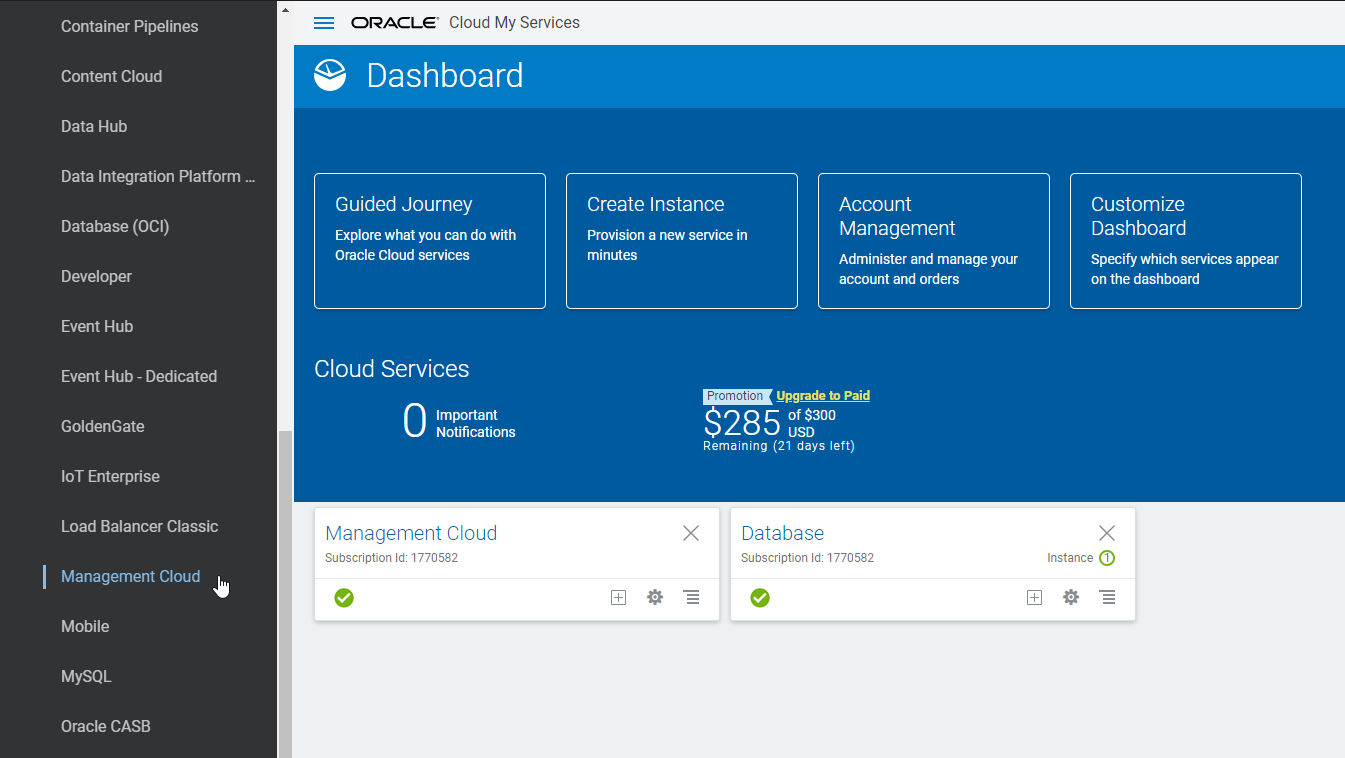
Once you see this page, you’re ready to create the instance. Click “Create” to proceed. You should be brought back to the Database Cloud Service summary page and see the new instance being created.



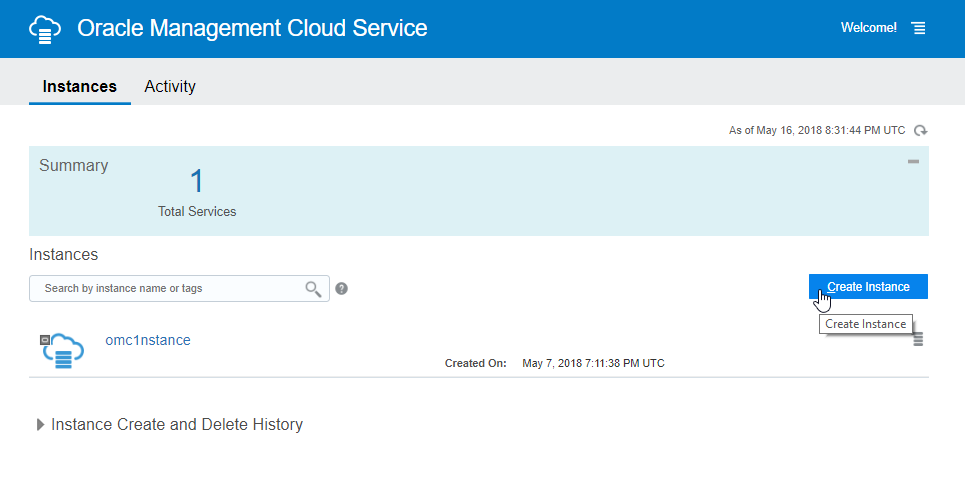
Once you see this page, you’re done with Database Cloud Service for a while. Remember how you got to this console because we will be coming back later in the lab. While the database in provisioning, we’ll create an instance of Oracle Management Cloud.

**Creating an instance of Oracle Management Cloud:**

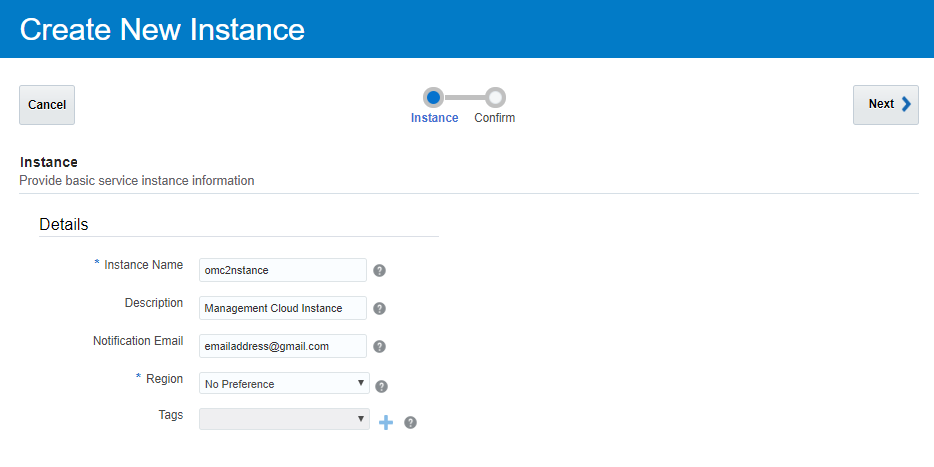
Step 1: From the My Services Dashboard, once again, click the “Action Menu” at the top-left of the screen to pop-out the window. After the window is popped out, click the “Services” dropdown. When the services dropdown is open, scroll until you find “Management Cloud” and select it.



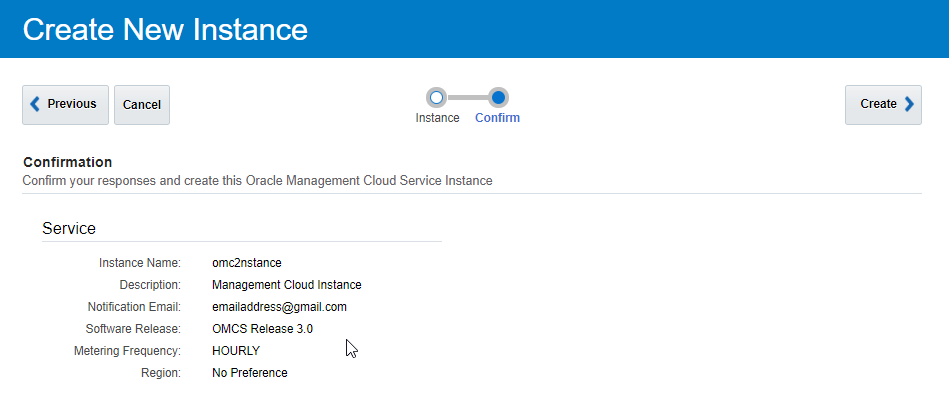
Step 2: Once on the Management Cloud Service console, find and select the “Create Instance” button and select it.



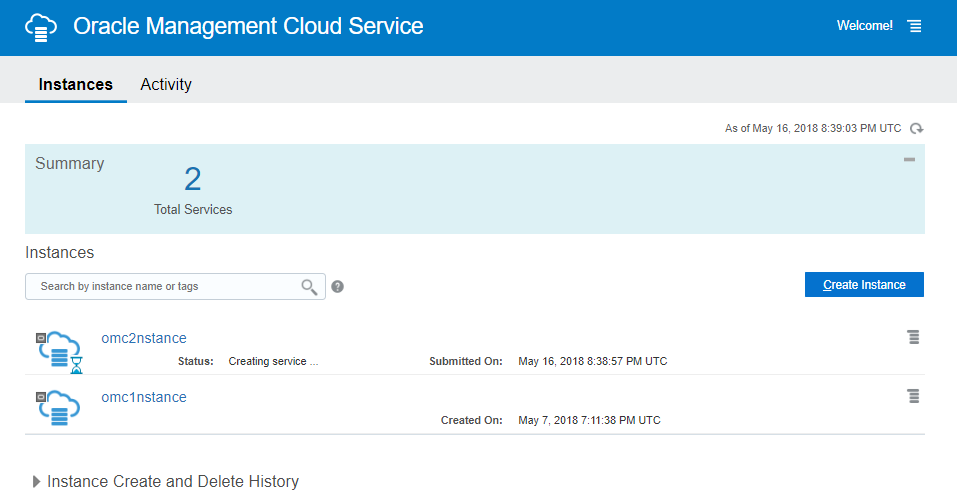
Step 3: On the Create New Instance page, name the instance you are creating. I named my instance “omc2nstance”. You can name this instance anything you would like. Once you have named your instance, click “Next”.



Step 4: Next you should see the summary page for the instance you are about to create. Review the details and then click “Create”.



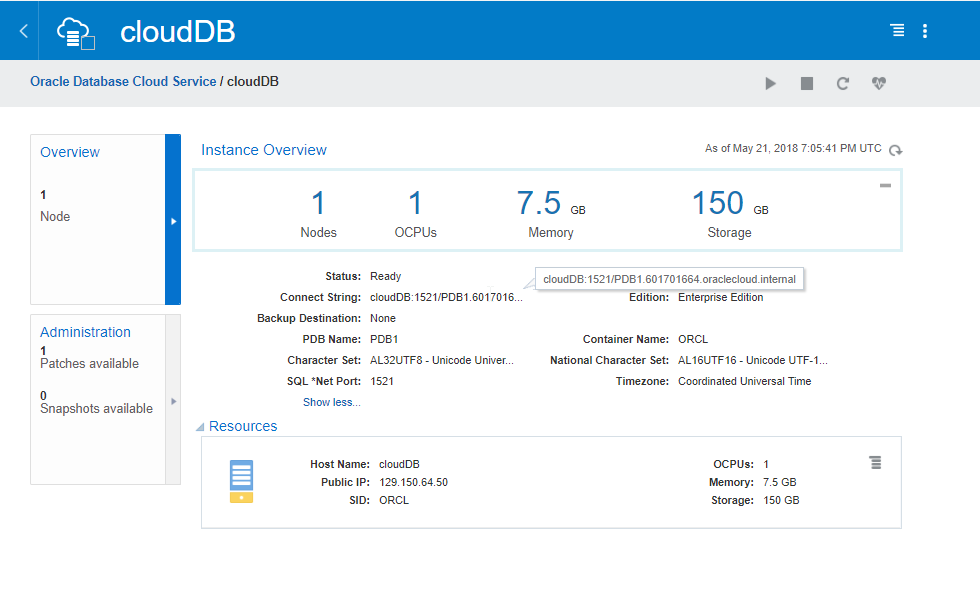
You should be brought back to the Management Cloud Service console page and see the new instance you just created being provisioned.



We’re going to need the instance to be live before we can use it. You’re ready to move to the next step.

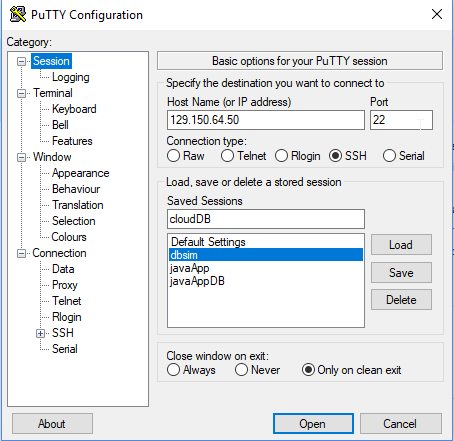
**Connecting to an instance of Database Cloud Service:**

Step 1: Navigate to Database Cloud Service console (from the My Services Dashboard, click the pop-out menu on the left and then select Database under Services). Once you are at the console, click on the database instance you created (“cloudDB”). You want to get the instance details page, which looks like this:

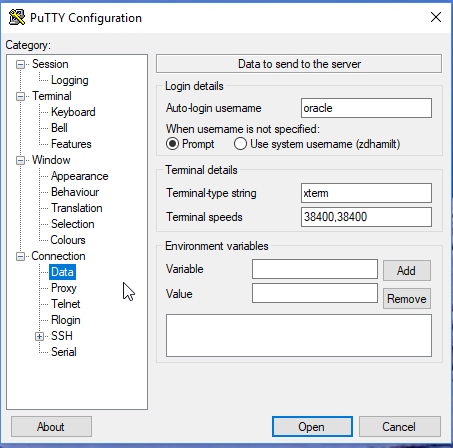


Step 2: Take some very important information down from this page. We’re going to want the Public IP, SID, Host Name, and Connect String (see above). I usually keep open a window with Notepad and store the information there.

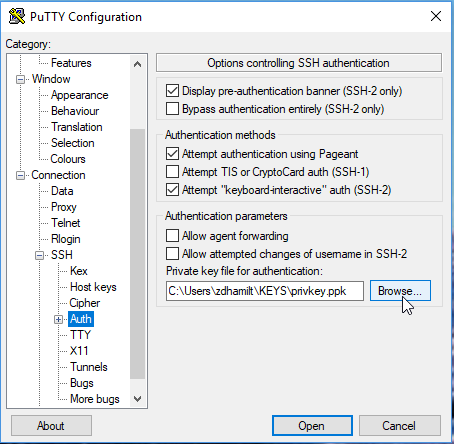
Step 3: Open a new PuTTy session. We’re going to add three pieces of important information (Public IP, Auto-login username, and the private encryption key). The first thing we will do is name the settings we’re configuring in the “Session > Saved Sessions” field. As you can see, I’m naming mine “cloudDB”. Next, add the public IP address in the “Session > Host Name (or IP address)” field. Both can be seen below:



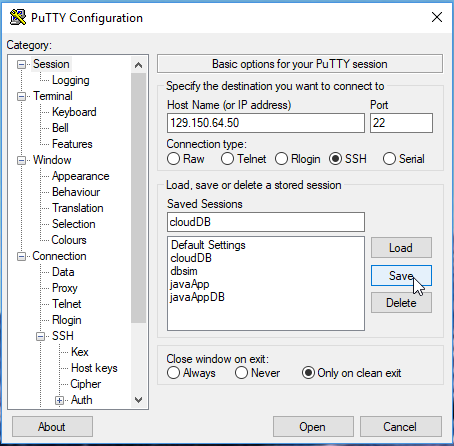
Next, we’re going to go to the “Connection > Data > Auto-login username” field and enter “oracle”. See below:

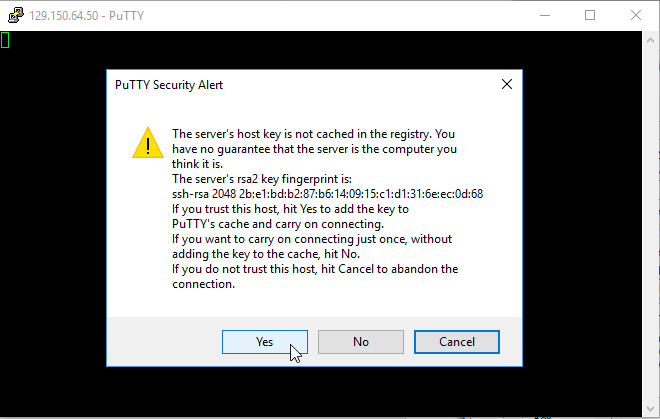


Once you’ve done that, navigate to “Connection > Expand ‘SSH’ > Auth > Private key file for authentication”. Select the “Browse” button next to the field and find and select your private key file. See below:

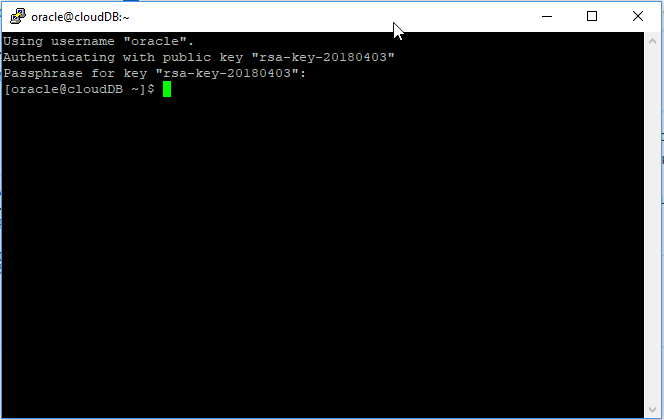


Now that you have all the settings for the connection configured, go back to the “Session” tab. We want to save all these settings so we can load them anytime we come back and want to create this same session. Click on “Save” as seen below.



Once you have saved the settings, select “Open”. You should get a warning (see below) that the RSA key fingerprint is not cached. Select “Yes” to continue. You can expect to not see this warning again after accepting. 

Once you select “Yes” and proceed, you should see something like the following:



You’re now connected to the host machine where your database instance is being hosted.

Step 4: Now we’re going to create a database user (with the name of your choice) both to have and to demonstrate using the SQL\*NET utilities that have been pre-configured to this machine. In the command line of the PuTTy session you have open to the host machine, execute the statement:

$ sqlplus / as sysdba

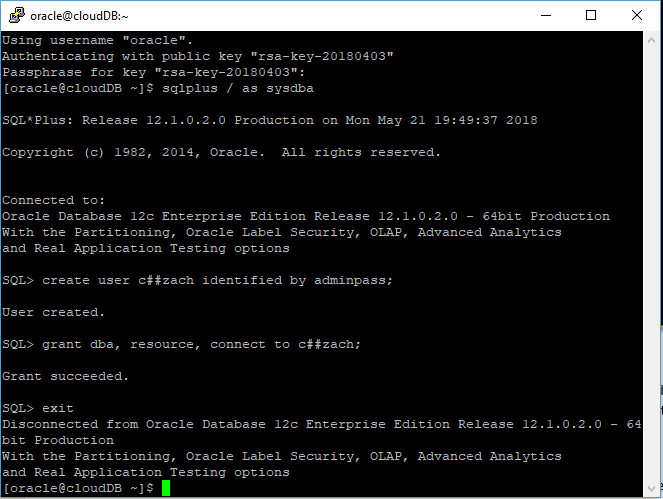
This will open-up the SQL\*Plus utility that we can use to create a new user in our database instance. By default, the DBCS instance we created is a multitenant database (if you want more information, click [here](https://www.oracle.com/database/technologies/multitenant.html)). So, we’re going to make a privileged user that exists in both the container database, and the pluggable database within our instance. We will do this by typing the following command from the SQL\*Plus command line:

SQL> create user c##zach identified by adminpass;

This will create a new user “c##zach” (Zach is my first name, you can concatenate anything you desire after the prefix “c##”) with the password “adminpass” (similarly, you can define the password to be whatever you chose). Next, we’re going to create our new user “c##zach” some privileges (DBA, resource, connect). To grant the user privileges, execute the following statement from the SQL\*Plus command line:

SQL> grant dba, resource, connect to c##zach;

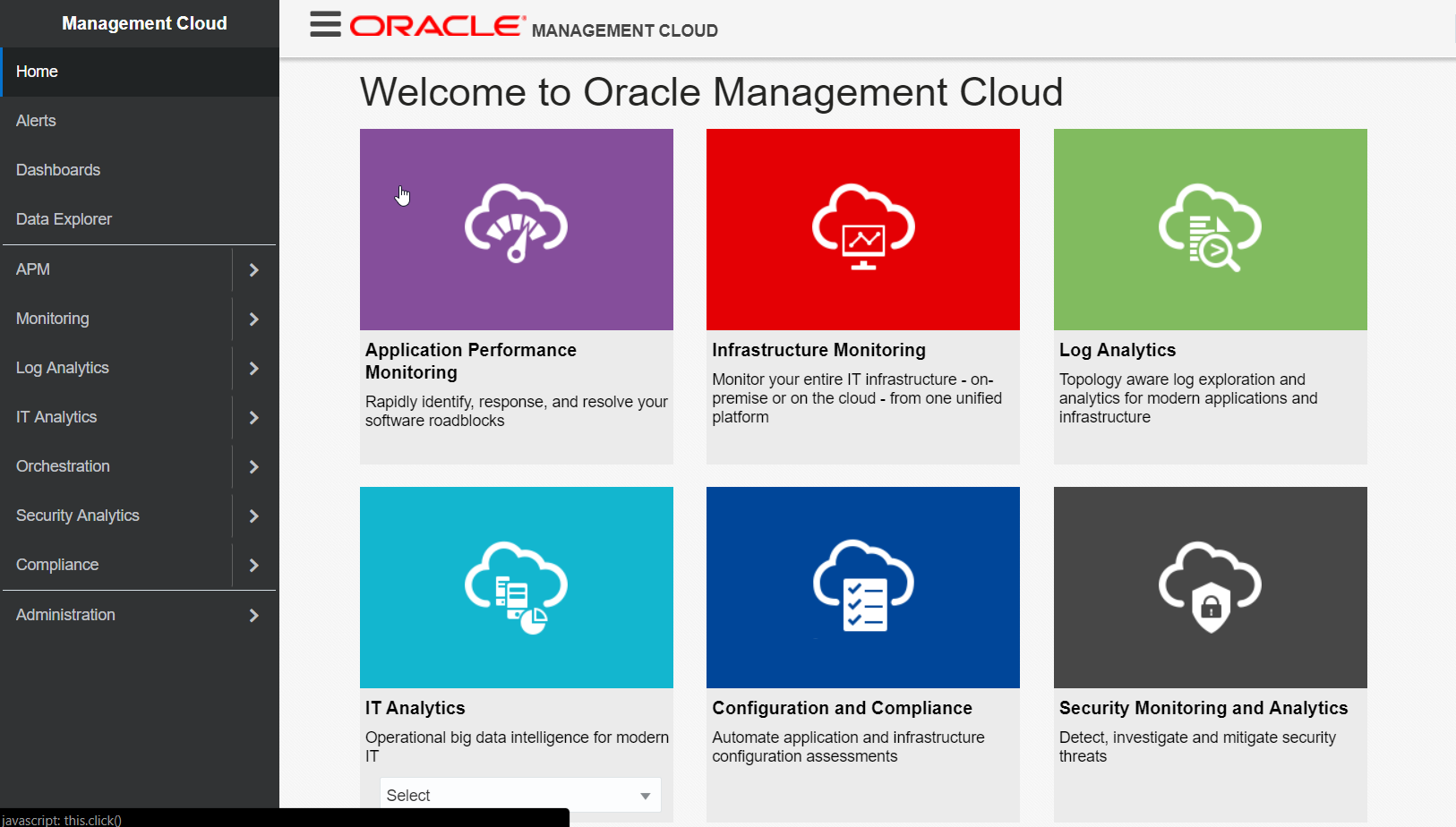
You now have yourself a new privileged user in your database. We won’t need this SQL\*Plus or PuTTy session for a while, so go ahead and exit both by simply typing and executing “exit” at the command lines. (See below for a whole recap).



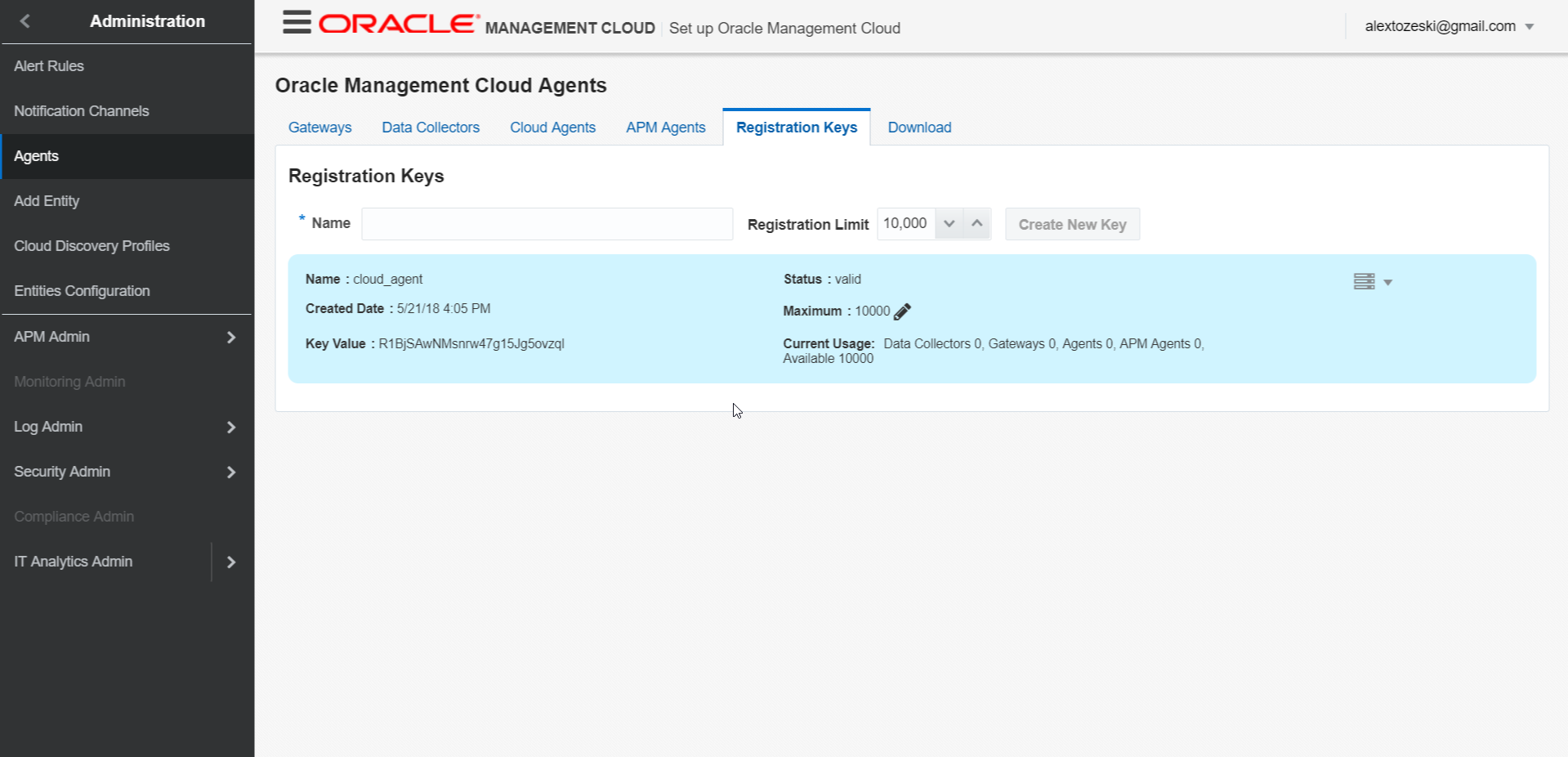
Keep PuTTy close by, we will need to use in again soon to validate some file transfers we’re going to do.

**Downloading Agent software from OMC instance:**

Step 1: Navigate to your OMC instance from the “My Services” Dashboard. You can do this by clicking the action menu at the top left of the dashboard, and selecting “Management” from the “Services” dropdown. Once you are in the OMC service console, click the action menu to the left of your instance name and select “OMC URL” to navigate to the OMC dashboard. Once there, you should see something like this:

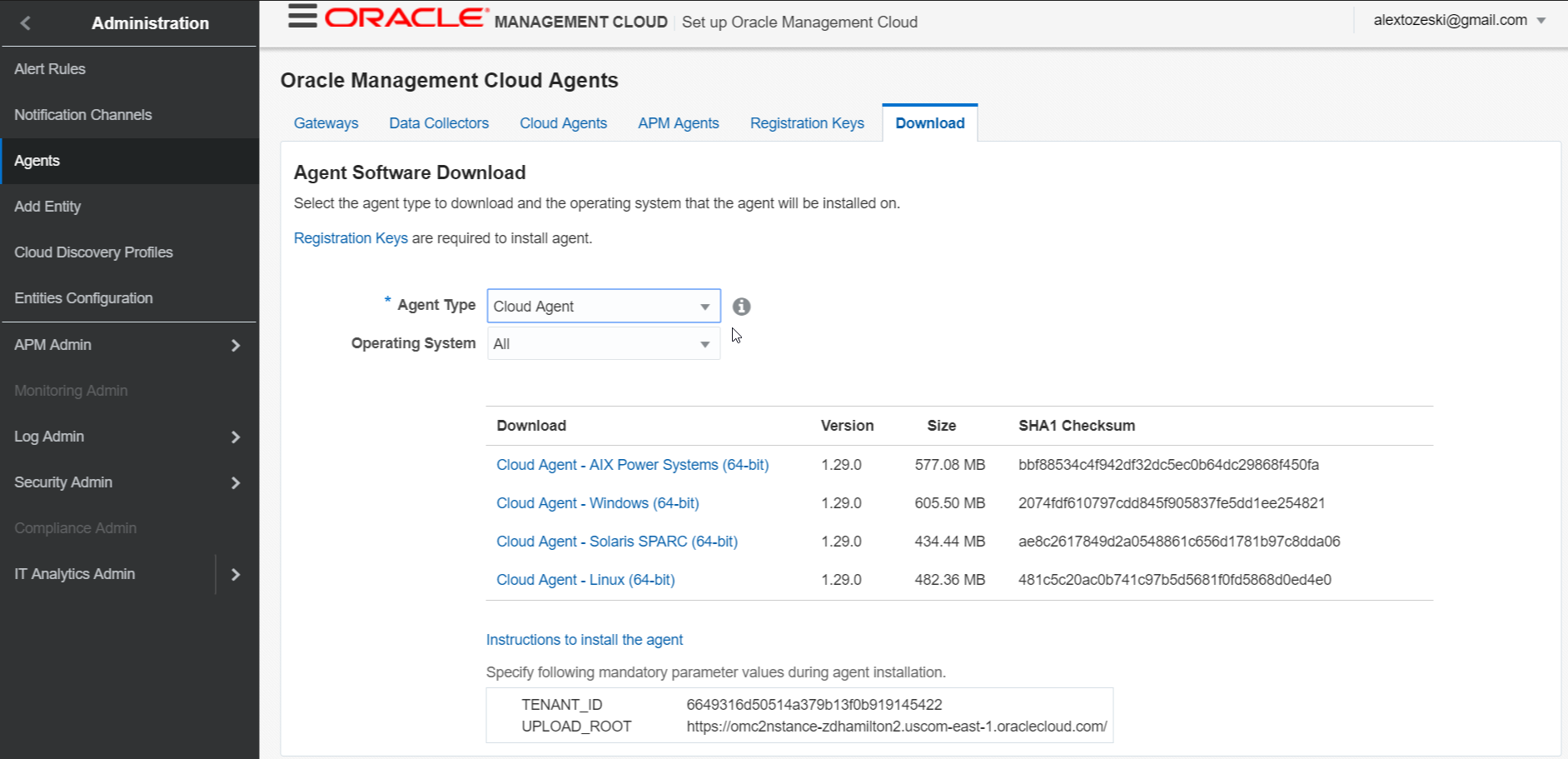


Step 2: Navigate to the Agent software download page by selecting “Administration > Agents” from the pop-out menu on the left. Once you’ve selected the “Agents” page, select the “Registration Keys” tab. Now, create a new registration key by naming the key, and selecting “Create New Key”. This should result in something looking like this:



Once you’ve create the registration key, save the key value somewhere (I always save it to the save Notepad document as my Pubic IP, SID, Host Name, and Connect String).

Step 3: Now we’re going to download the cloud agent software. To do this, navigate to the next tab on the “Agents” page titled “Downloads”. Select “Cloud Agent” from the “Agent Type” drop down menu. You can leave the Operating system drop down as “All”. The page should look like the following:



Before you download the agent software, you should take note of both the TENANT\_ID and the UPLOAD\_ROOT that you see at the bottom of the page (once again, I would save this with the other information you’ve been saving about your instances). Now, you are going to download the “Cloud Agent – Linux (64-bit)” package. To simplify the process of doing the file transfers later, make a dedicated folder for this lab to save the cloud agent software there. If not, just keep note of the file paths of your encryption keys, the cloud agent software, etc.

**Transferring the Agent Software to the Cloud Host:**

Since we’re assuming you’re using a Windows machine, you will need an SCP client to do the file transfer to the cloud instance. In the pre-requisites section of this lab, you should have already downloaded PSCP along with PuTTy and PuTTyGen. Unlike PuTTy and PuTTyGen, PSCP doesn’t have a user interface. We will be using it directly from the Command Prompt of your machine. You are going to either need to configure the executable (PSCP.exe) to your system PATH or move PSCP.exe to the same directory as the downloaded cloud agent .zip file. I have configured PSCP to my system PATH, so my statements will look slightly different than if you’re doing it from the same directory as the cloud agent software.

Step 1: If you configured PSCP to the system PATH, the setup of the command will look like the following:

> pscp -i <private key> <file to move> oracle@<public ip>:/home/oracle/

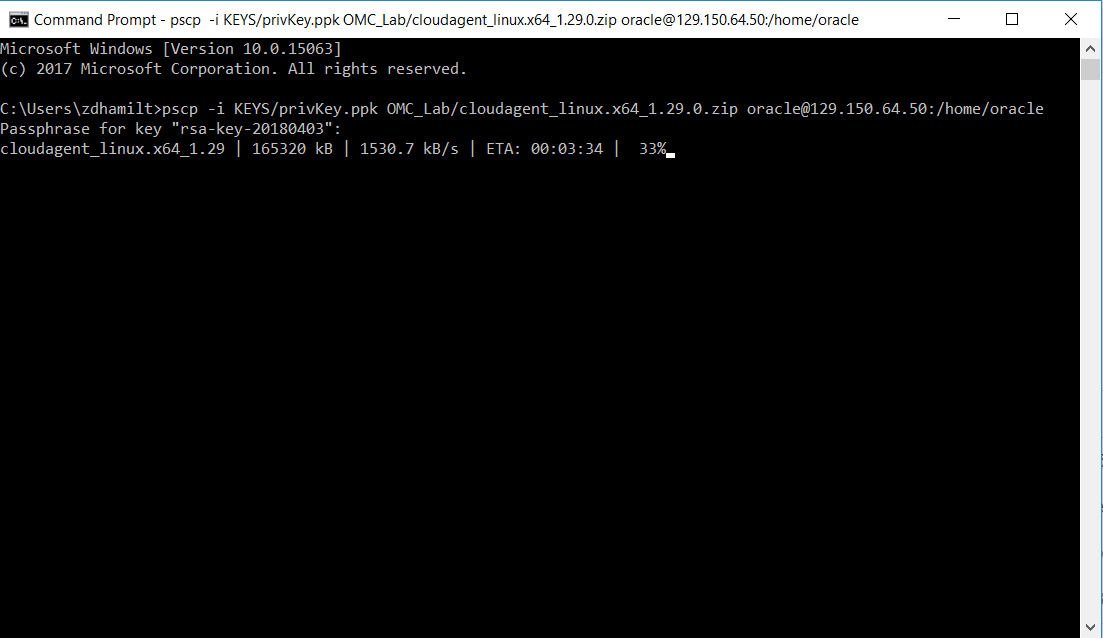
> pscp -i path/to/privateKey.ppk path/to/cloudagent\_linux.x64\_1.29.0.zip oracle@<public ip>:/home/oracle/

If you just moved PSCP.exe to the same directory as cloudagent\_linux.x64\_1.29.0.zip (let’s say they’re in ‘C:/OMC\_Lab/’), then executing the statements from the PWD ‘C:/OMC\_Lab/’ will look like:

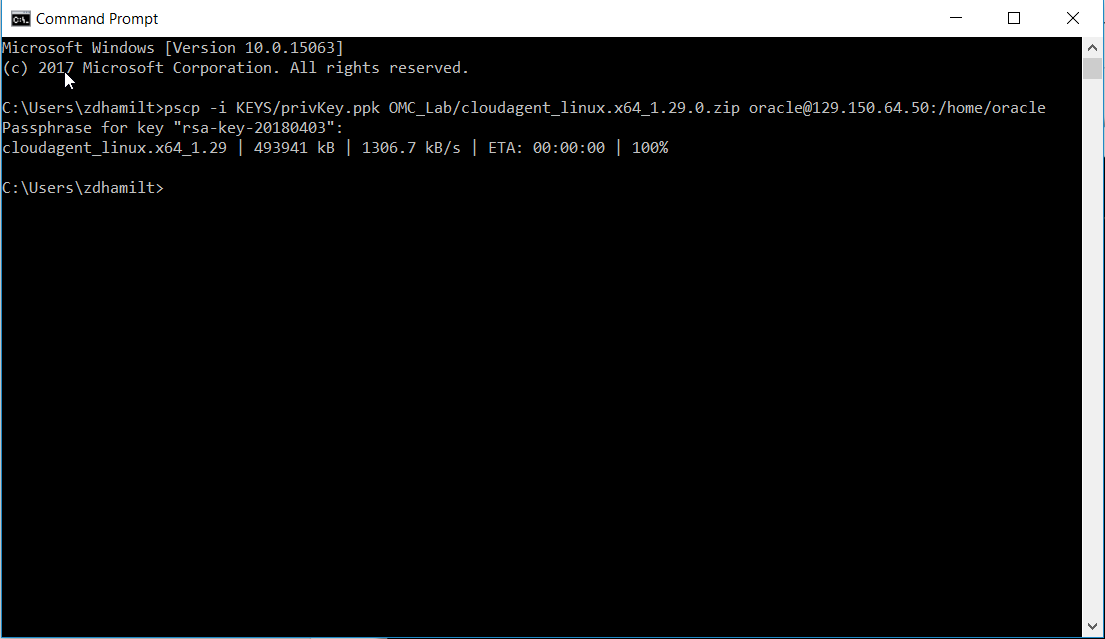
> pscp.exe -i <private key> <file to move> oracle@<public ip>:/home/oracle/

> pscp.exe -i path/to/privateKey.ppk cloudagent\_linux.x64\_1.29.0.zip oracle@<public ip>:/home/oracle/

If you’ve executed those statements correctly, you should see something like this:



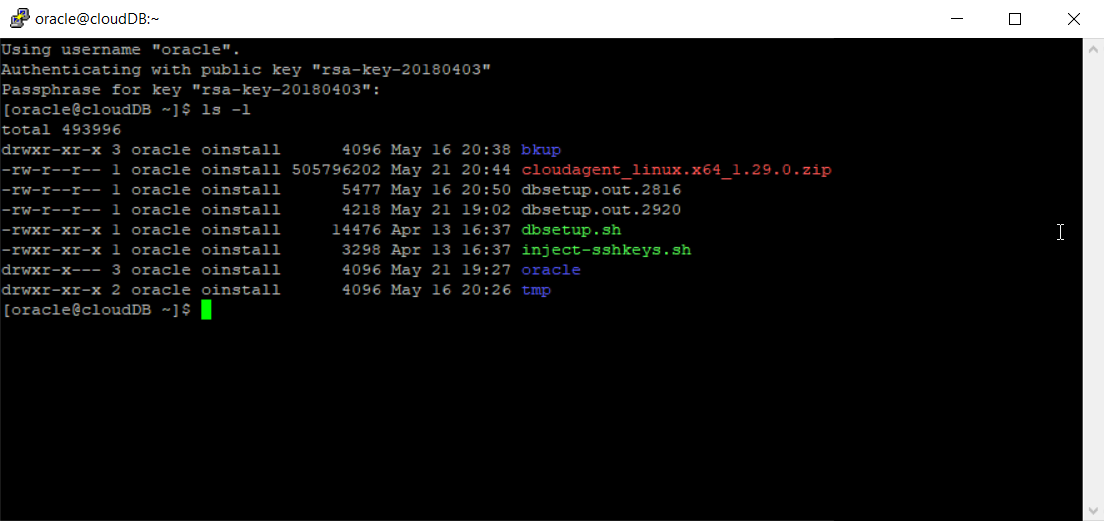
Wait for the file to completely transfer to the remote machine that your database is hosted on. Once the transfer is complete, you should see something like this:



Once you’ve completed the transfer, we should create another PuTTy session with our remote host to double check that the file is now at “/home/oracle/” on our host machine. Open a PuTTy session, select the saved settings by selecting “cloudDB” (or whatever you named it), and click “Load”. Once the settings are loaded, select “Open” to start a new terminal session with the host. Once connected, execute the following statement:

$ ls -l

If the file transfer was successful, you should see the .zip file listed in the terminal session. It should look something like the following:



The file has been successfully transferred to the remote machine!

**Installing the Agent software on the remote machine:**

Step 1: Now we’re going to create a temporary directory to unzip the contents of the .zip file into. From the same PuTTy session, execute the following commands:

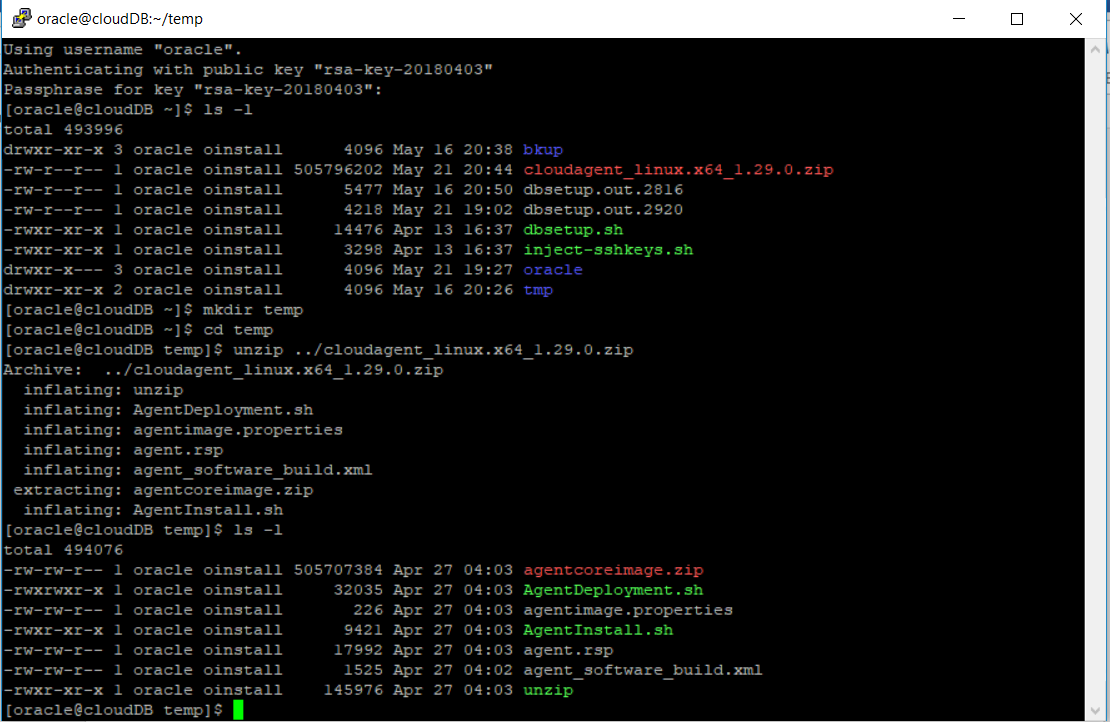
$ mkdir temp

$ cd temp

$ unzip ../cloudagent\_linux.x64\_1.29.0.zip

$ ls -l

If you did it all right, it should look like the following:



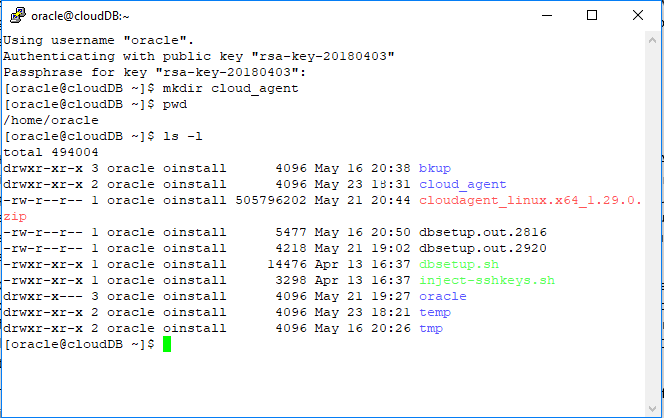
Step 2: Now we need to install the agent software from the “/temp” directory that we just created. To do this, we’re going to edit the response file “agent.rsp”. Setting up the response file to be prepared for installation is going to require some information about our OMC instance. The key pieces of information you’re going to need are:

1. TENANT\_ID
2. UPLOAD\_ROOT
3. AGENT\_REGISTRATION\_KEY
4. AGENT\_BASE\_DIRECTORY

Hopefully you’ve saved this information to a Notepad document! If not, you can find the TENANT\_ID and UPLOAD\_ROOT in the OMC console by navigating to: (From left-hand pop-out) Administration > Agents > Download (tab on agent’s page). The information about the UPLAOD\_ROOT and TENANT\_ID should be at the bottom of the page once you select something for the “Agent Type” dropdown. To get the registration key (you should have made this already), you should select the “Registration” tab next to the “Downloads” tab. If you have already created a registration key, it should be shown. If not, create one quickly and then note it.

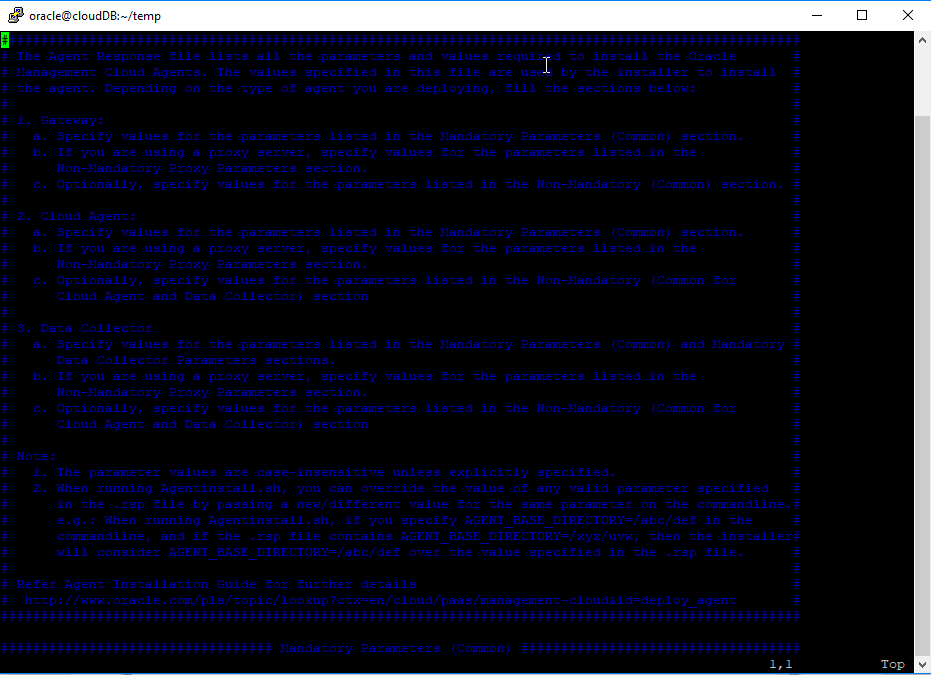
Once you have those pieces of information we’re going to edit the file on the remote machine. This is one of the more difficult to explain steps in the process, so follow the instructions carefully. If you have experience using Linux solely from the command line, you’re most likely familiar with “vi”. Vi is a text editor that works directly from the command like using only keyboard keystrokes. We’re going to use Vi to add the details about our TENANT\_ID, UPLOAD\_ROOT, etc. to the “agent.rsp” file.

The first thing you’ll want to do is create a new directory that we will define as the “AGENT\_BASE\_DIRECTORY” for the installation in the “agent.rsp” file. I’m going to title mine “cloud\_agent” and I’m going to create it at “/home/oracle/cloud\_agent”. You should have something that looks like mine:

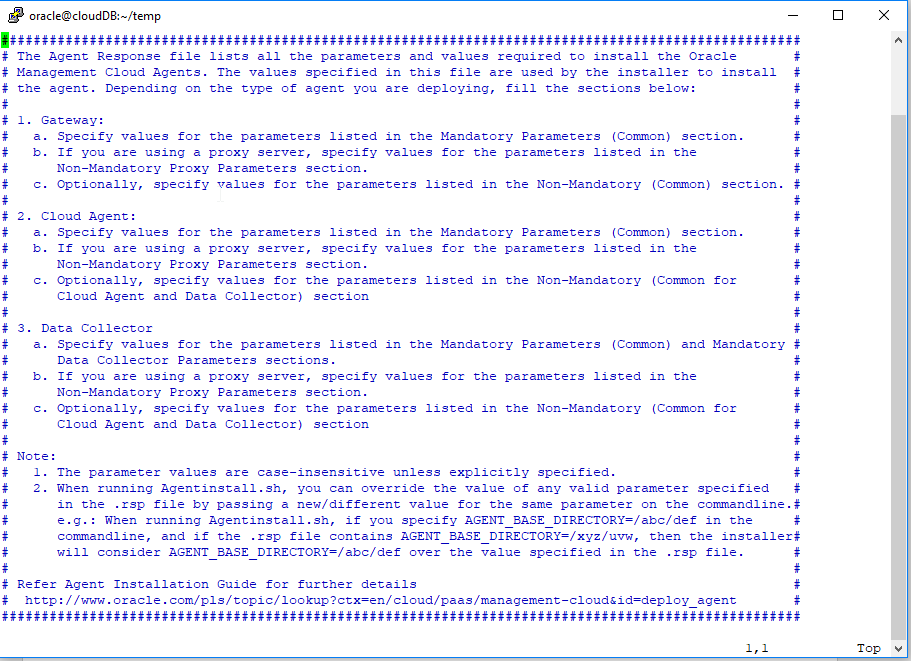


This is going to be the AGENT\_BASE\_DIRECTORY”. Also, you may have noticed that the color scheme of my PuTTy session has changed. This can be a very useful thing to do when using Vi because the default color settings make it difficult to read some text within the Vi editor. For instance, compare the next two images.

With PuTTy default color settings:



With the settings changed:



The difference is clear. If you are interested in changing the color settings like I have you’ll need to go to (in PuTTy) “Window > Colors > Select a color to adjust”. Change the “Default Background” RGB values to (255, 255, 255) and then change the “Default Foreground” RGB values to (0, 0, 0).

Step 3: If you’ve changed the settings or not, it’s time to take the information from our Notepad document and add it to “agent.rsp” by leveraging Vi. It’s time to follow these steps very closely!

Start by creating a new PuTTy session and connecting to your remote machine. Once you’ve connected, move to the folder where you unzipped the packaged agent software:

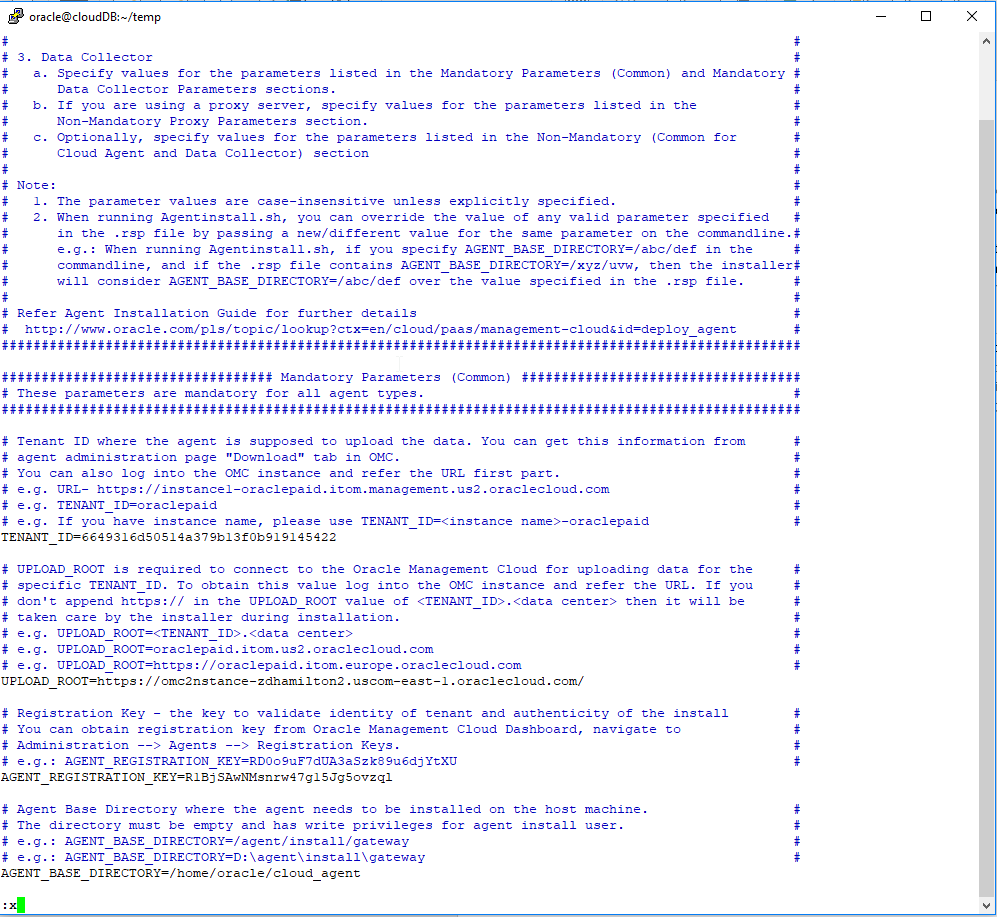
$ cd temp

Within the “temp” directory, execute the following command:

$ vi agent.rsp

You should be in the Vi editor now. There’s some important things to know. You won’t need your mouse for pretty much anything. Use the arrow keys to move the cursor to the “=” after “TENANT\_ID”. Now start typing. You should start adding text to the document. Now since the TENTANT\_ID isn’t a simple string, we’ll ideally copy-paste it into this document. You won’t be able to do so with Ctrl + V. Go to your Notepad document and copy the TENANT\_ID string. Now, regain the focus of the PuTTy session. With your cursor still after the “=”, simply “Right-click” to paste the string. Don’t leave a space between “TENANT\_ID=” and the string. Now, use the arrow keys again to get to the next variable and do the same thing.

Once you are done, press “Escape” once, then type “:x” followed by enter. Once you click “Escape”, anything you type should come up at the bottom, very last, line of the PuTTy session window. The “:x” command will save your changes and close the Vi session. Alternatively, you can also use “:q” to close the session without saving. This is may be confusing if you’ve never used Vi. If you need a guide to it, you can find that [here](https://www.cs.colostate.edu/helpdocs/vi.html). Everything (right before executing “:x” should look like the following if you’ve done it correctly:



Now that everything in the “agent.rsp” file is configured properly, we can install the agent! To do this, change to the “/home/oracle/temp” directory where the unzipped software package is found. Once there, you’ll just need to run the install script. This can be done by executing the following commands:

$ cd /home/oracle/temp

$ ./AgentInstall.sh

This script will install all the agent software into the “/home/oracle/cloud\_agent” directory we defined in the “agent.rsp” file. Once you’ve done that, it should look like this from the command line:



The agent is now monitoring the remote machine! It should start sending information to OMC about the host. To configure the agent to monitor the database, you’ll need to run some other scripts in the installation directory (“/cloud\_agent”). Move to the directory with all the scripts for the agent by doing the following:

$ cd cloud\_agent/plugins/oracle.em.sgfm.zip/1.29.0/scripts/

Then try to run the script by doing:

$ ./grantPrivilegesMonSvc.sh

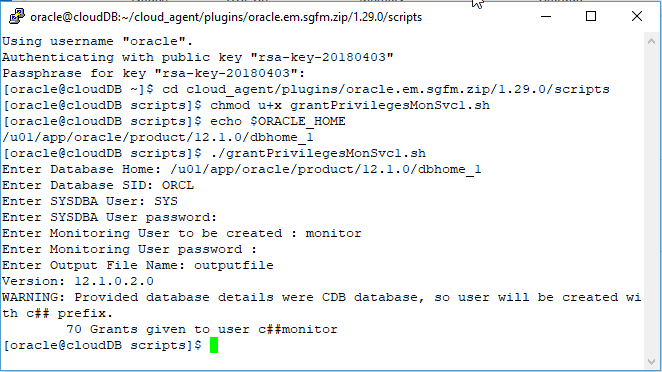
If you get a permission error like I did, give yourself execute permission:

$ chmod u+x grantPrivilegesMonSvc.sh

You can now execute the script. It’s going to ask you the Database home, to get this information, execute:

$ echo $ORACLE\_HOME

Now you should have what you need to run the script. Run the script with the same command as above. The prompt will ask you for the “Database Home” (this is the result of $ORACLE\_HOME), the “SID” (this should be ORCL by default unless you specified otherwise when creating the instance), a SYSDBA user (this should just be SYS), the SYSDBA user password (this is the password you create when you created the instance earlier, and should have written down), the new user to create (this can be anything, its creating a new user in the database to monitor it, so I call my “monitor”), and a password for the new user (you can use anything you want). You should see something like the following if you’ve done that all correctly:



Everything *should* be set up on the remote machine to now configure a database entity in OMC. Based on desired monitoring, the same process can be taken to other privilege granting scripts in the same directory. Let’s go to the OMC console and add an entity.

**Adding an Oracle Database Entity:**

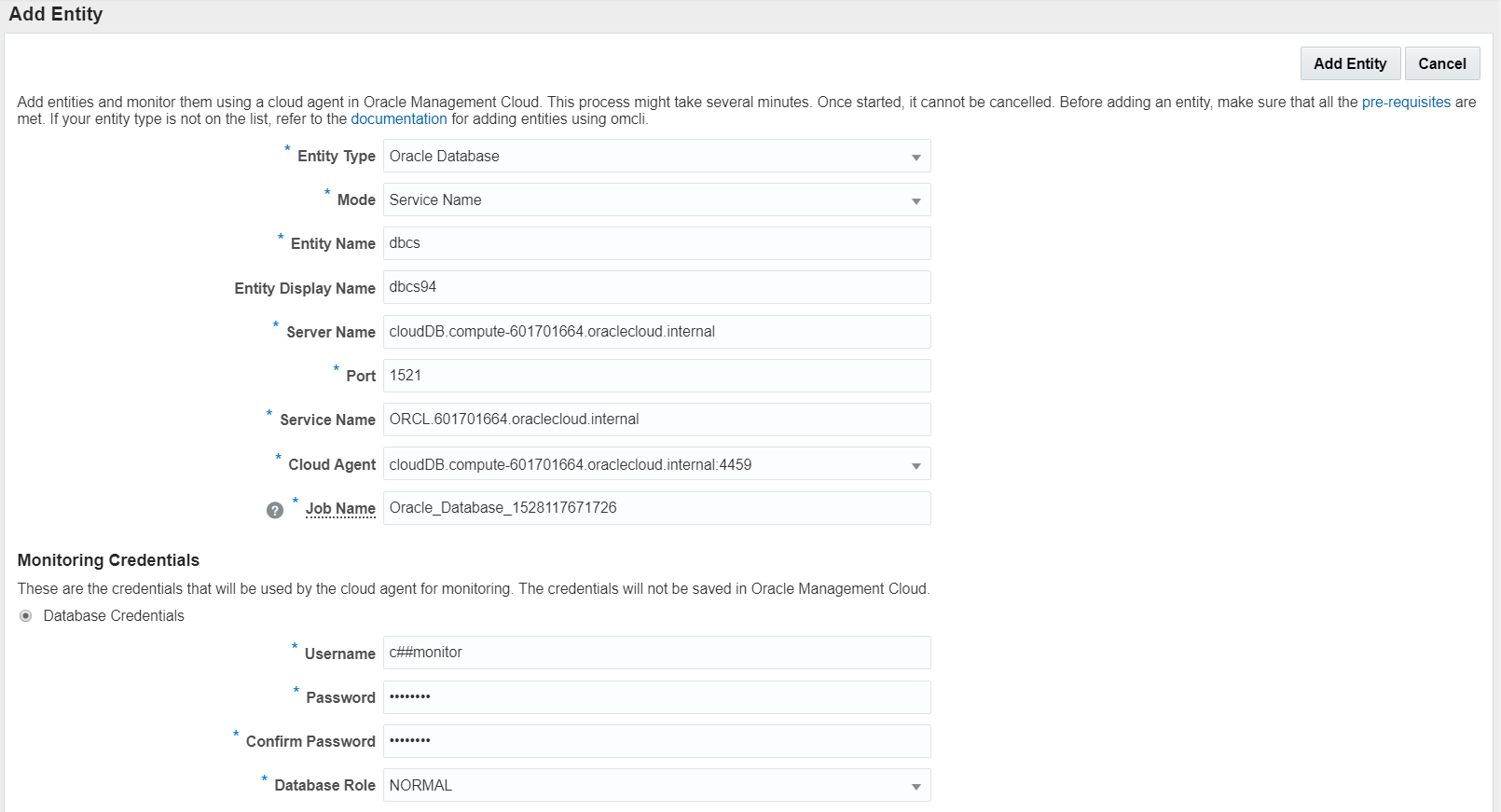
To configure the agent to the Oracle Database entity we’re going to need a few pieces of information first. The “Add Entity” process is going to ask us some information about the database instance we have running on our remote machine. First thing to do—PuTTy back into your remote machine. Once you have a new session, enter the following command and execute it:

$ lsnrctl status

This should give us the information we need about the instance to add the entity. Hopefully you have something like what I have here:



We’re going to take the value of “HOST” in the “Listening Endpoints Summary…” and the value in quotation marks in the “Services Summary…”. Take a close look at the values I have above and write them down somewhere. Hopefully you remember how to get back to the Management Cloud console. Once you’re there, navigate to “Administration > Add Entity”. If you haven’t added anything yet the page should be blank. Select “Add Entity”. Once you are on the page for adding entities, configure the settings the way I have below. You should use the information that you wrote down in the step above. If everything looks the way I have it below, click “Add Entity”.



Now you have configured an Oracle Management Cloud agent to communicate with a remote Linux machine as well as an instance of Oracle Database! You can now do to the dashboards page and start seeing the ways that information about the machine and the database are being pulled. I strongly suggest restoring a database backup to the database instance in order to generate some log files within the database that you can then use for discovery.

That’s all for now! More will come later.