

Oracle Machine Learning for R

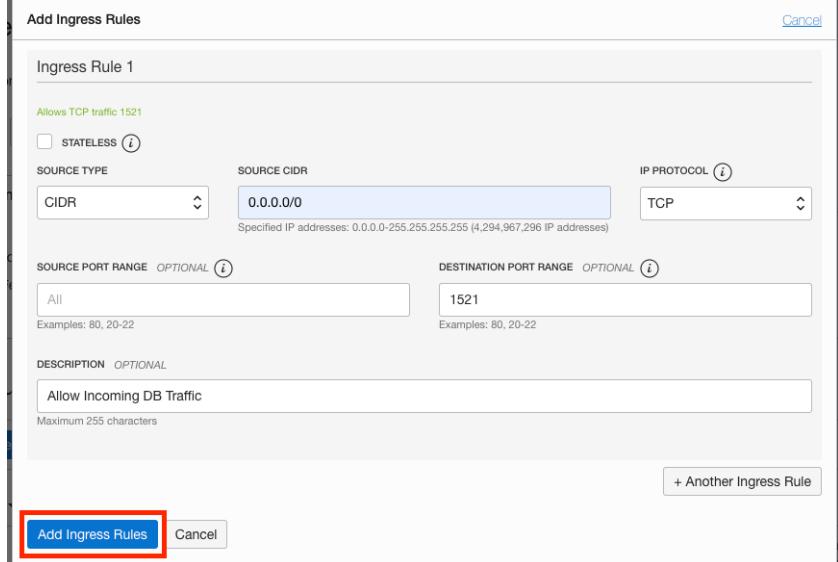
Server & Client-Side Installation & Setup: Oracle Database Cloud 19c

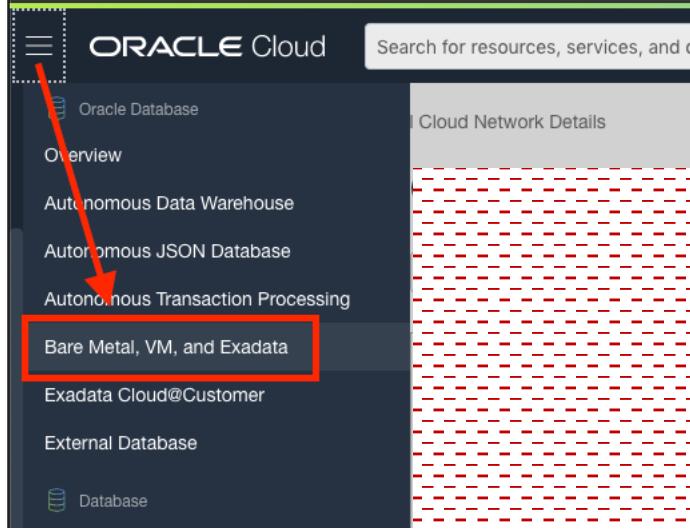
Step	Screenshot
<p>Let's start off by creating a basic network configuration to host our Oracle Database (Server-Side) and Oracle Compute Instance (Client-Side)</p> <p>From the OCI Console navigate to Menu > Networking > Virtual Cloud Network</p>	<p>The screenshot shows the Oracle Cloud navigation bar with the following menu structure:</p> <ul style="list-style-type: none"> Core Infrastructure Compute Block Storage Object Storage File Storage Networking (highlighted with a red box) Oracle Database Overview Autonomous Data Warehouse Dashboard Actions Overview Virtual Cloud Networks (highlighted with a red box) Dynamic Routing Gateways Customer-Premises Equipment
<p>For simplicity let's use the VCN Wizard to create a basic network configuration.</p> <p>Click 'Start VCN Wizard'</p>	<p>The screenshot shows the 'Virtual Cloud Networks' page with the following interface:</p> <ul style="list-style-type: none"> Left sidebar: Networking, Overview, Virtual Cloud Networks (highlighted), Dynamic Routing Gateways, Customer-Premises Equipment, VPN Connections, Load Balancers, FastConnect, IP Management. Right panel: Virtual Cloud Networks are virtual, private networks. Action buttons: Create VCN (blue), Start VCN Wizard (highlighted with a red box). Form fields: Name (placeholder: vcn1).
<p>Select 'VCN with Internet Connectivity'</p> <p>Click 'Start VCN Wizard'</p>	<p>The screenshot shows the 'Start VCN Wizard' dialog with the following options:</p> <ul style="list-style-type: none"> Radio buttons: VCN with Internet Connectivity (selected, highlighted with a red box) and VCN with Internet Connectivity and Site-to-Site VPN Connect. Diagram: A network diagram showing a VCN with Internet Connectivity, including a public subnet, private subnet, Internet gateway (IG), NAT gateway (NAT), and service gateway (SG). Description: Creates a VCN with a public subnet that can be reached from the internet. Also creates a private subnet that can connect to the internet through a NAT gateway, and also privately connect to the Oracle Services Network. Includes: VCN, public subnet, private subnet, Internet gateway (IG), NAT gateway (NAT), service gateway (SG). Action buttons: Start VCN Wizard (highlighted with a red box) and Cancel.

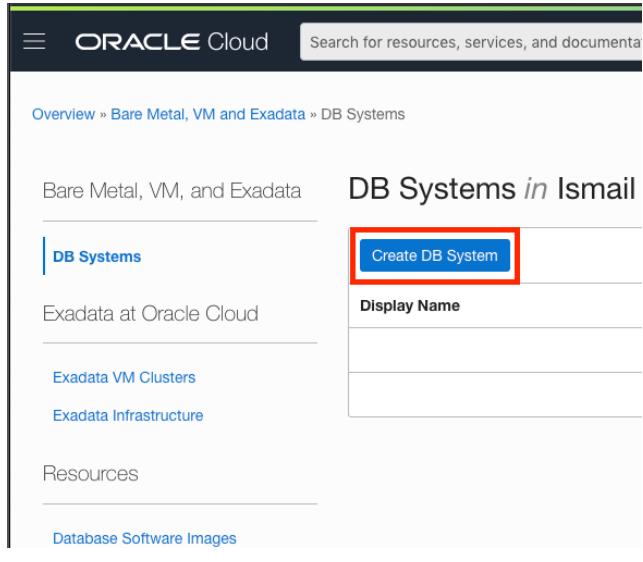
<p>Enter the following information:</p> <p>VCN Name Compartment VCN CIDR Block Public Subnet CIDR Private Subnet CIDR Select Use DNS Hostnames</p> <p>Click 'Next'</p>	<p>Create a VCN with Internet Connectivity</p>
<p>Review the networking configuration that will be deployed and Click 'Create'.</p> <p>Once created Click 'View Virtual Cloud Network'</p>	



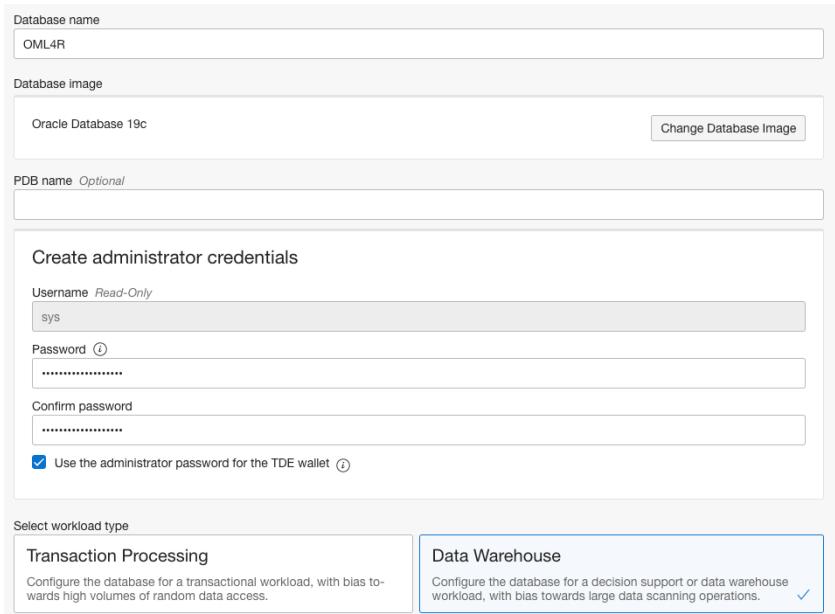
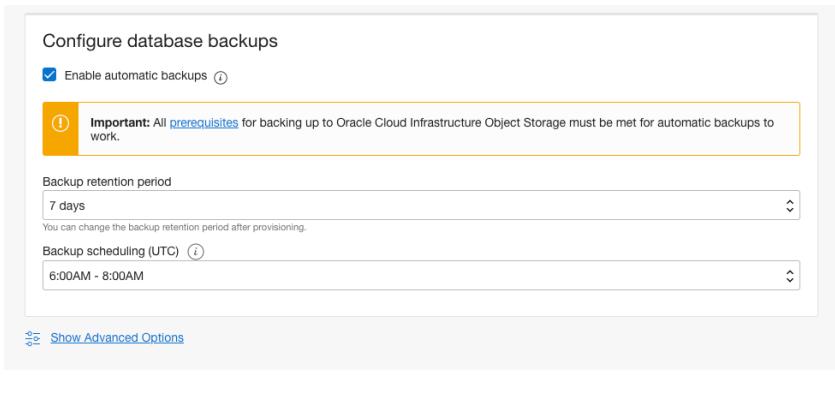
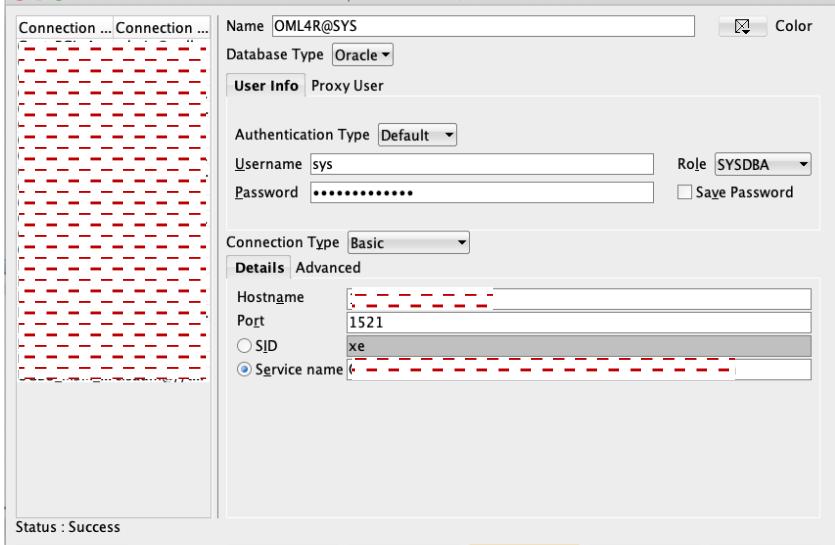
<p>We then want to edit the Security List of the Public Subnet to allow traffic into the Destination Port 1521 to allow access to the DB System.</p> <p>Click 'Security Lists'</p>	<p>Resources</p> <ul style="list-style-type: none"> Subnets (2) CIDR Blocks (1) Route Tables (2) Internet Gateways (1) Dynamic Routing Gateways (0) Network Security Groups (0) Security Lists (2) DHCP Options (1) Local Peering Gateways (0) 	<p>Subnets in Ismail</p> 																														
<p>Click 'Default Security List'.</p>	<p>Security Lists in Ismail Compartment</p> <table border="1"> <thead> <tr> <th>Name</th> <th>State</th> </tr> </thead> <tbody> <tr> <td>Security List for Private Subnet</td> <td>Available</td> </tr> <tr> <td>Default Security List for</td> <td>Available</td> </tr> </tbody> </table>	Name	State	Security List for Private Subnet	Available	Default Security List for	Available																									
Name	State																															
Security List for Private Subnet	Available																															
Default Security List for	Available																															
<p>Click 'Add Ingress Rule'</p>	<p>Ingress Rules</p> <table border="1"> <thead> <tr> <th colspan="5">Add Ingress Rules</th> </tr> <tr> <th><input type="checkbox"/></th> <th>Stateless ▾</th> <th>Source</th> <th>IP Protocol</th> <th>Source Port Range</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>No</td> <td>0.0.0.0/0</td> <td>TCP</td> <td>All</td> </tr> <tr> <td><input type="checkbox"/></td> <td>No</td> <td>0.0.0.0/0</td> <td colspan="2">ICMP</td> </tr> <tr> <td><input type="checkbox"/></td> <td>No</td> <td>0.0.0.0/0</td> <td colspan="2">ICMP</td> </tr> <tr> <td colspan="5">0 Selected</td> </tr> </tbody> </table>	Add Ingress Rules					<input type="checkbox"/>	Stateless ▾	Source	IP Protocol	Source Port Range	<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	<input type="checkbox"/>	No	0.0.0.0/0	ICMP		<input type="checkbox"/>	No	0.0.0.0/0	ICMP		0 Selected					
Add Ingress Rules																																
<input type="checkbox"/>	Stateless ▾	Source	IP Protocol	Source Port Range																												
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All																												
<input type="checkbox"/>	No	0.0.0.0/0	ICMP																													
<input type="checkbox"/>	No	0.0.0.0/0	ICMP																													
0 Selected																																

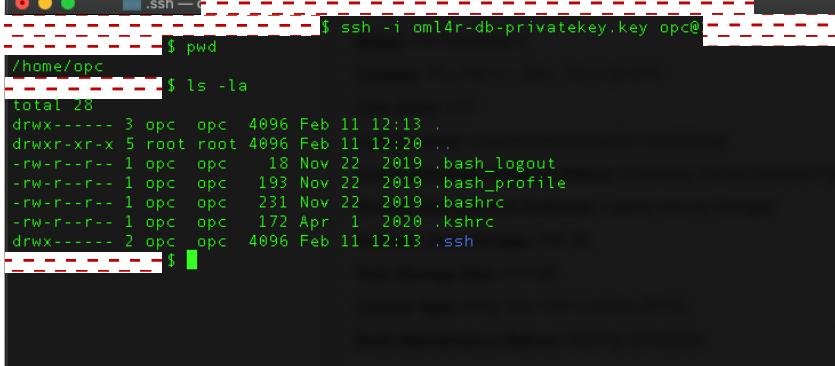
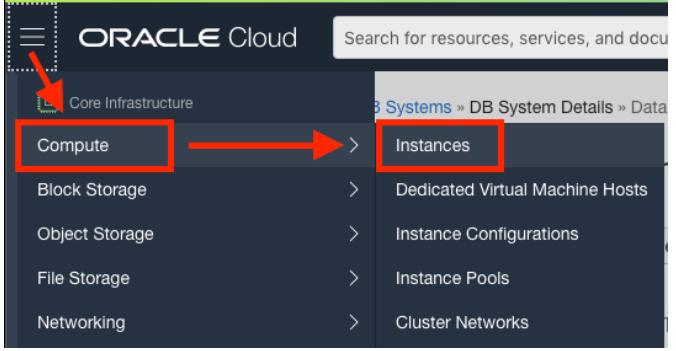
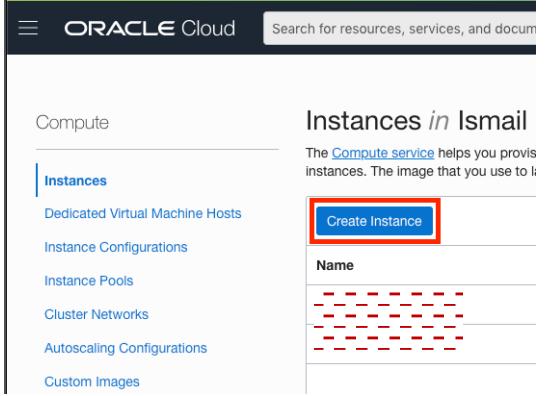
<p>Enter the following:</p> <p>CIDR – 0.0.0.0/0 (All IP) IP Protocol – TCP Destination Port – 1521</p> <p>Click ‘Add Ingress Rule’</p>	
--	--

<p>Now we can create and deploy our Oracle Database Cloud within the Public Subnet that has been created.</p> <p>Click on the Menu and navigate to ‘Bare Metal, VM, and Exadata’.</p>	
--	---

<p>Click on ‘Create DB System’.</p>	
--	--

<p><i>Fill in the following information:</i></p> <p>Compartment DB System Name Availability Domain Shape Type – VM VM Shape – ‘VM.Standard2.2’ Node Count – 1 DB Edition – Extreme Performance</p>	
<p><i>Select the following information:</i></p> <p>Storage – Logical Volume Manager Storage – 256GB Generate SSH Keys or Upload your Own</p>	
<p><i>Select the following information:</i></p> <p>License VCN – Pre-Created Subnet – Pre-Created Hostname Prefix</p> <p><i>Click ‘Next’</i></p>	

<p>Select the following information:</p> <p>DB Name DB Image – 19c Sys Password Workload Type – Data Warehouse</p>	 <p>Database name: OML4R</p> <p>Database image: Oracle Database 19c</p> <p>PDB name: <i>Optional</i></p> <p>Create administrator credentials:</p> <p>Username: Read-Only sys</p> <p>Password: <i>*****</i></p> <p>Confirm password: <i>*****</i></p> <p><input checked="" type="checkbox"/> Use the administrator password for the TDE wallet</p> <p>Select workload type:</p> <p>Data Warehouse (selected)</p>
<p>Select 'Enable Automatic Backups'</p> <p>Choose Retention Period</p> <p>Choose Back Scheduling Time</p> <p>Click 'Create DB System'.</p>	 <p>Configure database backups</p> <p><input checked="" type="checkbox"/> Enable automatic backups</p> <p>Important: All prerequisites for backing up to Oracle Cloud Infrastructure Object Storage must be met for automatic backups to work.</p> <p>Backup retention period: 7 days</p> <p>Backup scheduling (UTC): 6:00AM - 8:00AM</p> <p>Show Advanced Options</p>
<p>Once the DB has been provisioned, we can test to see if we can connect via SQL Developer and also SSH into the DB VM.</p> <p>Open up SQL Developer and Create a New Connection to our DB using the connection details on the DB Instance Page.</p> <p>Execute a test SQL Statement.</p>	 <p>New / Select Database Connection</p> <p>Name: OML4R@SYS</p> <p>Database Type: Oracle</p> <p>User Info: Proxy User</p> <p>Authentication Type: Default</p> <p>Username: sys</p> <p>Password: <i>*****</i></p> <p>Role: SYSDBA</p> <p>Save Password: <input type="checkbox"/></p> <p>Connection Type: Basic</p> <p>Details: Advanced</p> <p>Hostname: <i>*****</i></p> <p>Port: 1521</p> <p>SID: xe</p> <p>Service name: <db_unique_name>.host_domain_name></p> <p>Status: Success</p> <p>Help Save Clear Test Connect Cancel</p> <p>Service Name = <db_unique_name>.host_domain_name></p>

<p>Create a new user Ismail that we can work with.</p>	<pre>-- View PDBs SELECT PDB_ID, PDB_NAME, STATUS FROM DBA_PDBS; -- Switch to pluggable DB ALTER SESSION SET CONTAINER = OML4R_PDB1; -- Create User CREATE USER ismail IDENTIFIED BY [REDACTED] -- Grant DBA Role to User GRANT SYSDBA TO ismail; -- Set Default Tablespace ALTER USER ismail DEFAULT TABLESPACE USERS; -- GGrant Unlimited Quota on USERS Tablespace ALTER USER ismail QUOTA UNLIMITED ON USERS; -- View Default Table Space select username, default_tablespace from dba_users where username = 'ISMAIL';</pre>
<p>Now open up a terminal window (mac) and test if we can ssh into the DBCS VM using the SSH Keys we generated at DB creation time.</p> <p>chmod 600 <private_key></p> <p>ssh -i <private_key> opc@<ip_address></p>	 <p>The terminal window shows a successful SSH session to the Oracle Cloud VM. The command entered was \$ ssh -i oml4r-db-privatekey.key opc@<ip_address>. The output shows the user's home directory (~/.ssh) and its contents, including the private key file (oml4r-db-privatekey.key).</p>
<p>Now let's create our compute instance which will act as our Client-Side Machine to interact with the Server.</p> <p>Navigate to Menu > Compute > Instances.</p>	 <p>The screenshot shows the Oracle Cloud Core Infrastructure menu. The 'Compute' option is selected, and the 'Instances' sub-option is highlighted. Red arrows point from the 'Compute' and 'Instances' labels to their respective menu items.</p>
<p>Click on 'Create Instance'.</p>	 <p>The screenshot shows the 'Instances in Ismail' creation page. The 'Create Instance' button is highlighted with a red box. The page includes fields for 'Name' and other configuration options.</p>

<p>Enter the following:</p> <p>Name Compartiment Availability Domain OS Image – Oracle Linux Shape – VM.Standard2.1</p>	<p>Name OML4R-Client</p> <p>Create in compartment Ismail</p> <p>Configure placement and hardware</p> <p>The availability domain helps determine which shapes are available. A shape is a template that determines the number of CPUs, amount of memory, allocated to an instance. The image is the operating system that runs on top of the shape.</p> <p>Availability domain AD 1 AD 2 AD 3</p> <p><input type="checkbox"/> Choose a fault domain for this instance If you don't select a fault domain, Oracle will choose the best placement for you. Learn more</p> <p>Image</p> <p>ORACLE Linux Oracle Linux 7.9 Image build: 2021.01.12-0</p> <p>Shape</p> <p>intel VM.Standard2.1 Virtual Machine, 1 core OCPU, 15 GB memory, 1 Gbps network bandwidth</p>
<p>Select our VCN</p> <p>Select our Public Subnet</p> <p>Assign a Public IP Address</p>	<p>Configure networking</p> <p>Networking is how your instance connects to the internet and other resources in the Console. To make sure you can connect to your instance, assign a public IP address to the instance.</p> <p>Network <input checked="" type="radio"/> Select existing virtual cloud network <input type="radio"/> Create new virtual cloud network <input type="radio"/> Enter subnet OCID</p> <p>Virtual cloud network in Ismail (Change Compartment)</p> <p>Subnet <input checked="" type="radio"/> Select existing subnet <input type="radio"/> Create new public subnet</p> <p>Subnet in Ismail (Change Compartment)</p> <p>Public Subnet+</p> <p><input type="checkbox"/> Use network security groups to control traffic</p> <p>Public IP Address <input checked="" type="radio"/> Assign a public IPv4 address <input type="radio"/> Do not assign a public IPv4 address</p> <p>Assigning a public IP address makes this instance accessible from the internet. If you're not sure whether you need a public IP address, you can always assign one later.</p>
<p>For simplicity select 'choose public key file' and upload our DB public key file that we generated earlier. Therefore, we can use the same SSH Private Key to access the Client.</p> <p>Set Boot Volume size to 100GB</p> <p>Click 'Create'</p>	<p>Add SSH keys</p> <p>Linux-based instances use an SSH key pair instead of a password to authenticate remote users. Generate a key pair or upload your own connect to the instance, you will provide the associated private key.</p> <p><input type="radio"/> Generate SSH key pair <input checked="" type="radio"/> Choose public key files <input type="radio"/> Paste public keys <input type="radio"/> No SSH keys</p> <p>SSH public keys</p> <p>Drop files here. Or browse to a location.</p> <p>oml4r-db-publickey.pub x Must be in OpenSSH format. View example</p> <p>Configure boot volume</p> <p>Your boot volume is a detachable device that contains the image used to boot your compute instance.</p> <p><input checked="" type="checkbox"/> Specify a custom boot volume size Volume performance varies with volume size. Default boot volume size: 46.6 GB</p> <p>Boot volume size (GB) 100 Integer between 50 GB and 32,768 GB (32 TB). Must be larger than the default boot volume size for the selected image.</p> <p><input type="checkbox"/> Use in-transit encryption Encrypts data in transit between the instance, the boot volume, and the block volumes.</p>

Now that our Server (Database VM) and Client (Compute) is configured and set up the first thing we need to do is install R on both which is a dependency for OML4R.

We will use Oracles distribution of R which contains a package called ROracle which enables interaction between R and the DB, and it also simplifies the installation.

We will install Oracle R Distribution using Linux Yum as it automatically resolves the RPM dependencies.

SSH into our Client Compute Instance.

`ssh -i <private_key> opc@<ip_address>`

```
$ ssh -i oml4r-db-privatekey.key opc@  
The authenticity of host '...' is unknown.  
ECDSA key fingerprint is ...  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '...' to the list of known hosts.  
$ pwd  
/home/opc  
$
```

Change to the `/etc/yum.repos.d` directory.

```
$ cd /etc/yum.repos.d  
$ ls  
ksplie-ol7.repo      oracle-epel-ol7.repo      uek-ol7.repo  
ksplie-uptrack.repo  oraclelinux-developer-ol7.repo  virt-ol7.repo  
mysql-ol7.repo       oracle-linux-ol7.repo  
oci-included-ol7.repo oracle-softwarecollection-ol7.repo  
$
```

<p>As there is no <i>public-yum-ol7</i> file present we need to download Oracle Public Yum using (ol7 stands for Oracle Linux 7)</p> <p><i>sudo wget https://public-yum.oracle.com/public-yum-ol7.repo</i></p>	<pre>\$ sudo wget https://public-yum.oracle.com/public-yum-ol7.repo --2021-02-11 15:36:08-- https://public-yum.oracle.com/public-yum-ol7.repo Resolving public-yum.oracle.com (public-yum.oracle.com)... 104.84.57.106 Connecting to public-yum.oracle.com (public-yum.oracle.com) 104.84.57.106 :443... connected. HTTP request sent, awaiting response... 200 OK Length: 16402 (16K) [text/plain] Saving to: 'public-yum-ol7.repo' 100%[=====] 16,402 --.-K/s in 0s 2021-02-11 15:36:08 (138 MB/s) - 'public-yum-ol7.repo' saved [16402/16402]</pre>
<p>Open the <i>public-yum-ol7.repo</i> in a text editor and specify <i>enabled=1</i> for the <i>latest</i> and <i>addons</i> and <i>optional_latest</i> sections. These are where the R packages and dependencies are located.</p> <p><i>Sudo vi public-yum-ol7.repo</i></p> <p>Update the required values.</p>	<pre>\$ sudo vi public-yum-ol7.repo [ol7_latest] name=Oracle Linux \$releasever Latest (\$basearch) baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/latest/\$basearch/ gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle gpgcheck=1 enabled=1 [ol7_optional_latest] name=Oracle Linux \$releasever Optional Latest (\$basearch) baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/optional/latest/\$basearch/ gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle gpgcheck=1 enabled=1 [ol7_addons] name=Oracle Linux \$releasever Add ons (\$basearch) baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/addons/\$basearch/ gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle gpgcheck=1 enabled=1</pre> <p>For this to work on the DB server I had to also manually add a yum entry for [ol7_u9_base] and disable [ol7_latest] as the dependencies required have moved.</p>
<p>Execute the yum install command to install Oracle R Distribution.</p> <p><i>sudo yum install R-3.6.1</i></p> <p>Use <i>R -version</i> to check everything has installed.</p>	<pre>\$ sudo yum install R-3.6.1 Dependency Updated: libblkid.x86_64 0:2.23.2-65.0.1.el7_9.1 libmount.x86_64 0:2.23.2-65.0.1.el7_9.1 libsmartcols.x86_64 0:2.23.2-65.0.1.el7_9.1 libuuid.x86_64 0:2.23.2-65.0.1.el7_9.1 util-linux.x86_64 0:2.23.2-65.0.1.el7_9.1 zlib.x86_64 0:1.2.7-19.el7_9 Complete! [opc@oml4r-client yum.repos.d]\$ R --version Oracle Distribution of R version 3.6.1 (--) -- "Action of the Toes" Copyright (C) The R Foundation for Statistical Computing Platform: x86_64-pc-linux-gnu (64-bit) R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under the terms of the GNU General Public License versions 2 or 3. For more information about these matters see https://www.gnu.org/licenses/.</pre>

<p>Now let's repeat the exact same steps but on the DB Server.</p> <p>SSH into our Server DB Instance.</p> <pre>ssh -i <private_key> opc@<ip_address></pre>	
<p>Change to the <i>/etc/yum.repos.d</i> directory.</p>	<pre>\$ cd /etc/yum.repos.d</pre>
<p>As there is no public-yum-ol7 file present we need to download Oracle Public Yum using (ol7 stands for Oracle Linux 7)</p> <pre>sudo wget https://public-yum.oracle.com/public-yum-ol7.repo</pre>	
<p>Open the <i>public-yum-ol7.repo</i> in a text editor and specify <i>enabled=1</i> for the <i>latest</i> and <i>addons</i> and <i>optional_latest</i> sections. These are where the R packages and dependencies are located.</p> <p>Sudo vi public-yum-ol7.repo</p> <p>Update the required values.</p>	<pre>[ol7_latest] name=Oracle Linux \$releasever Latest (\$basearch) baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/latest/\$basearch/ gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle gpgcheck=1 enabled=1 [ol7_optional_latest] name=Oracle Linux \$releasever Optional Latest (\$basearch) baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/optional/latest/\$basearch/ gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle gpgcheck=1 enabled=1 [ol7_addons] name=Oracle Linux \$releasever Add ons (\$basearch) baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/addons/\$basearch/ gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle gpgcheck=1 enabled=1</pre>
<p>Execute the yum install command to install Oracle R Distribution.</p>	<pre>\$ sudo yum install R-3.6.1</pre>

<p><i>sudo yum install R-3.6.1</i></p> <p>Use R --version to check everything has installed.</p>	<pre>Complete! [REDACTED] \$ R --version Oracle Distribution of R version 3.6.1 (-- -- "Action of the Toes" Copyright (C) The R Foundation for Statistical Computing Platform: x86_64-pc-linux-gnu (64-bit) R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under the terms of the GNU General Public License versions 2 or 3. For more information about these matters see https://www.gnu.org/licenses/.</pre>
<p>As of Oracle Database 18c, the rqcfg.sql installation script for OML4R Server is part of the database.</p> <p>The script is in the \$ORACLE_HOME/R/server/ directory.</p> <p>The rqcfg.sql script enables the OML4R Server components that are part of the database, configures some aspects of the server, and installs some OML4R database objects.</p>	<pre>[REDACTED] \$ ssh -i oml4r-db-privatekey.key [REDACTED] Last login: Thu Feb 11 17:55:06 2021 from [REDACTED]</pre>
<p>SSH into our DB Instance.</p> <p>ssh -i <private_key> opc@<ip_address></p> <p>Switch to the Oracle users as they have the dba group which is needed.</p> <p>sudo su - oracle</p>	<p>The OML4R Server packages are in the \$ORACLE_HOME/R/library/ directory. For Oracle Database 18c and 19c, the server packages in Oracle Database are built under R-3.3.0 and are compatible with R-3.3.0. If you are using R-3.3.0, you can install OML4R Server by running the rqcfg.sql installation script.</p> <p>For use with R-3.6.1 or later, the OML4R Server packages are built under R-3.6.1. To use OML4R with R-3.6.1, you must download and install those OML4R Server packages before running the rqcfg.sql installation script.</p> <pre>[REDACTED] \$ sudo su - oracle Last login: Fri Feb 12 11:17:23 UTC 2021 on pts/0</pre>

In Oracle Database 18c and 19c, the OML4R Server packages are in the \$ORACLE_HOME/R/library directory. For use with R-3.6.1, you can either rename the R-3.6.1 OML4R packages in that directory or you can delete them. We will rename them.

Cd \$ORACLE_HOME/R/library

mv ORE ORE.orig

Do this for all exiting packages.

```
$ cd $ORACLE_HOME/R/library
$ ls
ORE      OREcommon    OREdplyr    OREembedded    OREmodels    OREserver    ORExml
OREbase  OREdm        OREeda      OREgraphics    OREpredict   OREstats
$ mv ORE ORE.orig
$ mv OREbase OREbase.orig
$ mv OREcommon OREcommon.orig
$ mv OREdm OREdm.orig
$ mv OREdplyr OREdplyr.orig
$ mv OREeda OREeda.orig
$ mv OREembedded OREembedded.orig
$ mv OREgraphics OREgraphics.orig
$ mv OREmodels OREmodels.orig
$ mv OREPredict OREPredict.orig
$ mv OREserver OREserver.orig
$ mv OREstats OREstats.orig
$ mv ORExml ORExml.orig
$ ls
ORE.orig      OREdm.orig      OREembedded.orig    OREPredict.orig    ORExml.orig
OREbase.orig  OREdplyr.orig  OREgraphics.orig  OREserver.orig
OREcommon.orig OREeda.orig   OREmodels.orig   OREstats.orig
```

Visit <https://www.oracle.com/database/technologies/r-enterprise-downloads.html>

And download the R-3.6.1 compatible OML4R Server Packages to an installation directory, such as /oml4rserver_install_dir/

Oracle Machine Learning for R Documentation

Note: Starting with Oracle Database 18c, to install Oracle Machine Learning for R (formerly Oracle R Enterprise) see Oracle Machine Learning for R Installation (OML4R) and Administration Guide for instructions.

Platform	OML4R 1.5.1 (R-3.6.1) Documentation	System Requirements
Windows		Client (6M) Supporting (4M) Server (14M)
Linux 64-bit		Client (5M) Supporting (4M) Server (17M)
Linux 8 64-bit		

Switch back to the opc user: **exit**

Create the new directory: **sudo mkdir /oml4rserver_install_dir/**

```
$ exit
logout
$ sudo mkdir /oml4rserver_install_dir
```

<p>Once the download is complete on your local machine. Use SCP to copy the .zip file into the DB Server Home Directory</p>	<pre>\$ scp -i oml4r-db-privatekey.key/Downloads/ore-server-linux-x86-64-1.5.1.zip opc@130.61.160.88:/home/opc ore-server-linux-x86-64-1.5.1.zip 100% 17MB 4.3MB/s 00:04</pre> <p><code>scp -i oml4r-db-privatekey.key/Downloads/ore-server-linux-x86-64-1.5.1.zip opc@<db-ip-address>:/home/opc</code></p>
<p>Move the file for our DB Server Home Directory into the Installation Directory.</p> <p><code>sudo mv ore-server-linux-x86-64-1.5.1.zip /omlrserver_install_dir/</code></p>	<pre>\$ sudo mv ore-server-linux-x86-64-1.5.1.zip /omlrserver_install_dir/</pre>
<p>Navigate to the installation directory.</p> <p><code>cd /oml4rserver_install_dir</code></p>	<pre>\$ cd /omlrserver_install_dir/ \$ ls ore-server-linux-x86-64-1.5.1.zip \$</pre>
<p>Unzip the Downloaded File.</p> <p><code>sudo unzip ore-server-linux-x86-64-1.5.1.zip</code></p>	<pre>\$ sudo unzip ore-server-linux-x86-64-1.5.1.zip Archive: ore-server-linux-x86-64-1.5.1.zip creating: server/ inflating: server/OREbase_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: server/OREgraphics_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: server/OREserver_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: server/ORE_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: server/OREdpolyr_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: server/OREembed_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: server/OREstats_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: server/OREpredict_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: server/OREcommon_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: server/OREmodels_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: server/OREdm_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: server/ORExml_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz \$</pre>
<p>Switch users to oracle.</p> <p><code>sudo su - oracle</code></p> <p>Navigate into the <i>server</i> directory and execute all the executables.</p>	

```
cd
/oml4rserver_install_dir/server
```

ORE CMD INSTALL ORE_.....-gnu.tar.gz

For all Packages. This will install OML4R Packages for R-3.6.1.

```
$ ORE CMD INSTALL OREserver_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz
* installing to library '/u01/app/oracle/product/19.0.0/dbhome_1/R/library'
* installing *binary* package 'OREserver' ...
* DONE (OREserver)

$ ORE CMD INSTALL OREstats_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz
installing to library '/u01/app/oracle/product/19.0.0/dbhome_1/R/library'
* installing *binary* package 'OREstats' ...
* DONE (OREstats)

$ ORE CMD INSTALL ORExml_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz
* installing to library '/u01/app/oracle/product/19.0.0/dbhome_1/R/library'
* installing *binary* package 'ORExml' ...
* DONE (ORExml)
```

We can now install OML4R Server by running the rqcfg.sql script.

Before we do that make note of the following:

- Permanent tablespace for the RQSYS schema
- Temporary tablespace
- Oracle_Home
- R_Home

You can use the following sql command to identify your tablespaces available.

```
SQL> select tablespace_name, contents from user_tablespaces;
TABLESPACE_NAME          CONTENTS
-----  -----
SYSTEM                  PERMANENT
SYSAUX                 PERMANENT
UNDOTBS1                UNDO
TEMP                   TEMPORARY
USERS                  PERMANENT
```

Query Oracle home from cmd using: **echo \$ORACLE_HOME**

Query from cmd **R RHOME** to get R Home - /usr/lib64/R

Lets set the R_HOME environment variable using:

R_HOME=/usr/lib64/R

```
$ R_HOME=/usr/lib64/R
$ echo $R_HOME
/usr/lib64/R
$
```

Permanent – SYSAUX

Temporary – TEMP

ORACLE_HOME - /u01/app/oracle/product/19.0.0/dbhome_1

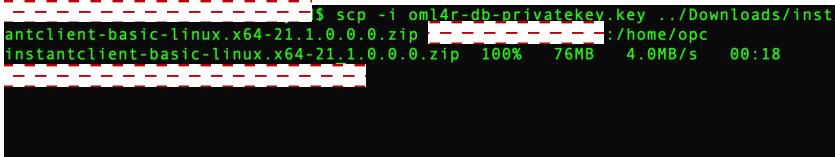
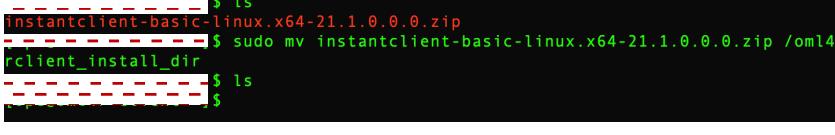
R_HOME - /usr/lib64/R (this should be the one under ORACLE_HOME)



<p>Now we have the needed information let's log into the DB using sql plus so we can execute the rqcfg.sql file.</p> <p><i>cd ~</i></p> <p><i>sqlplus / as sysdba</i></p>	<pre>\$ cd ~ \$ sqlplus / as sysdba SQL*Plus: Release 19.0.0.0.0 - Production on Fri Feb 12 12:19:37 2021 Version 19.9.0.0.0 Copyright (c) 1982, 2020, Oracle. All rights reserved. Connected to: Oracle Database 19c EE Extreme Perf Release 19.0.0.0.0 - Production Version 19.9.0.0.0 SQL> █</pre>
<p>To capture the log, spool the installation steps to an external file.</p> <p><i>spool install.txt</i></p>	<pre>SQL> spool install.txt SQL> █</pre>
<p>Alter our session to set our container DB as our pluggable DB.</p> <p><i>alter session set container=oml4r_pdb1;</i></p>	<pre>SQL> alter session set container=oml4r_pdb1; Session altered. SQL></pre>
<p>Alter the default limit for password.</p> <p><i>alter profile default limit password_verify_function null;</i></p>	<pre>SQL> alter profile default limit password_verify_function null; Profile altered. SQL> █</pre>
<p>Run the rqcfg.sql file.</p> <p><i>@\$ORACLE_HOME/R/server/rqcfg.sql</i></p> <p>Enter our parameters that we defined above.</p>	<pre>SQL> @\$ORACLE_HOME/R/server/rqcfg.sql Session altered. Enter value for 1: SYSAUX Enter value for 2: TEMP Enter value for 3: /u01/app/oracle/product/19.0.0/dbhome_1 Enter value for 4: /usr/lib64/R</pre>



<p>Once finished running exit the sql prompt. Exit</p> <p>You can cat the install.txt file to check for any errors.</p> <p>cat install.txt</p>	<pre>-----\$ cat install.txt SQL> alter session set container=oml4r_pdb1; Session altered. SQL> alter profile default limit password_verify_function null; Profile altered. SQL> @\$ORACLE_HOME/R/server/rqcfg.sql Session altered. Enter value for 1: SYSAUX Enter value for 2: TEMP Enter value for 3: /u01/app/oracle/product/19.0.0/dbhome_1 Enter value for 4: /usr/lib64/R PL/SQL procedure successfully completed.</pre>
<p>Navigate to the ORACLE_HOME/bin directory and grant read and execute permissions to all users to the ORE Directory.</p> <p>cd \$ORACLE_HOME/bin</p> <p>chmod 755 ORE</p>	<pre>\$ cd \$ORACLE_HOME/bin \$ chmod 755 ORE</pre>
<p>In order for the Oracle Client to talk to the DB Server we need to Install Oracle Database Instance Client to the Client.</p> <p>SSH into our Client Compute Instance.</p> <p>ssh -i <private_key> opc@<ip_address></p> <p>Create an Installation directory.</p> <p>sudo mkdir /oml4rclient_install_dir</p>	<pre>[user@host ~]\$ cd .ssh [user@host .ssh]\$ ssh -i oml4r-db-privatekey.key opc@192.168.1.11 Last login: Thu Feb 11 15:22:01 2021 from 192.168.1.11 [opc@host ~]\$ sudo mkdir /oml4rclient_install_dir [opc@host ~]\$</pre>

<p>Visit the Oracle Instant Client Download Page (https://www.oracle.com/database/technologies/instant-client.html) and download the Instant Client Basic Package.</p>	<h2>Oracle Instant Client Downloads</h2> <p>Instant Client for Microsoft Windows</p> <ul style="list-style-type: none"> Instant Client for Microsoft Windows (x64) Instant Client for Microsoft Windows (32-bit) Instant Client for Microsoft Windows 64-bit Itanium <p>Instant Client for Linux</p> <ul style="list-style-type: none"> Instant Client for Linux x86-64 (highlighted) Instant Client for Linux x86 Instant Client for Linux AMD64 (32-bit and 64-bit) Instant Client for Linux on Power Big Endian (32-bit) Instant Client for Linux on Power Big Endian (64-bit) Instant Client for Linux on Power Little Endian (64-bit) <p>Version 21.1.0.0.0 (Requires glibc 2.14)</p> <p>Base - one of these packages is required</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding-bottom: 5px;">Name</th><th style="text-align: left; padding-bottom: 5px;">Download</th></tr> </thead> <tbody> <tr> <td style="padding-top: 5px;">Basic Package (ZIP)</td><td style="padding-top: 5px;"> instantclient-basic-linux.x64-21.1.0.0.0.zip (highlighted)</td></tr> <tr> <td style="padding-top: 15px;">Basic Package (RPM)</td><td style="padding-top: 15px;"> oracle-instantclient-basic-21.1.0.0.0-1.x86_64.rpm</td></tr> <tr> <td style="padding-top: 15px;">Basic Light Package (ZIP)</td><td style="padding-top: 15px;"> instantclient-basiclite-linux.x64-21.1.0.0.0.zip</td></tr> </tbody> </table>	Name	Download	Basic Package (ZIP)	 instantclient-basic-linux.x64-21.1.0.0.0.zip (highlighted)	Basic Package (RPM)	 oracle-instantclient-basic-21.1.0.0.0-1.x86_64.rpm	Basic Light Package (ZIP)	 instantclient-basiclite-linux.x64-21.1.0.0.0.zip
Name	Download								
Basic Package (ZIP)	 instantclient-basic-linux.x64-21.1.0.0.0.zip (highlighted)								
Basic Package (RPM)	 oracle-instantclient-basic-21.1.0.0.0-1.x86_64.rpm								
Basic Light Package (ZIP)	 instantclient-basiclite-linux.x64-21.1.0.0.0.zip								
<p>Once downloaded locally, Secure Copy the file into the Client Instance Home Directory</p> <pre>scp -i oml4r-db-privatekey.key/Downloads/instantclient-basic-linux.x64-21.1.0.0.0.zip opc@<client-ip-address>:/home/opc</pre>									
<p>SSH into the Client and move the file into the Installation Directory</p> <pre>sudo mv instantclient-basic-linux.x64-21.1.0.0.0.zip /oml4rclient_install_dir</pre>									



<p>Navigate to your Installation Directory <code>cd /oml4rclient_install_dir</code></p> <p>Unzip the File <code>sudo unzip instantclient-basic-linux.x64-21.1.0.0.0.zip</code></p>	<pre>\$ cd /oml4rclient_install_dir/ instantclient-basic-linux.x64-21.1.0.0.0.zip \$ ls \$ sudo unzip instantclient-basic-linux.x64-21.1.0.0.0.zip Archive: instantclient-basic-linux.x64-21.1.0.0.0.zip inflating: instantclient_21_1/adrci inflating: instantclient_21_1/BASIC_LICENSE inflating: instantclient_21_1/BASIC_README inflating: instantclient_21_1/genz1 linking: instantclient_21_1/libclntshcore.so -> libclntshcore.so.21.1 linking: instantclient_21_1/libclntshcore.so.12.1 -> libclntshcore.so.21.1 linking: instantclient_21_1/libclntshcore.so.18.1 -> libclntshcore.so.21.1</pre>								
<p>Revisit the Oracle Instant Client Download Page (https://www.oracle.com/database/technologies/instant-client.html) and download the Instant Client Package SDK.</p>	<p>Development and Runtime - optional packages</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Download</th> </tr> </thead> <tbody> <tr> <td>SDK Package (ZIP)</td> <td> instantclient-sdk-linux.x64-21.1.0.0.0.zip</td> </tr> <tr> <td>SDK Package (RPM)</td> <td> oracle-instantclient-devel-21.1.0.0.0-1.x86_64.rpm</td> </tr> <tr> <td>JDBC Supplement Package (ZIP)</td> <td> instantclient-jdbc-linux.x64-21.1.0.0.0.zip</td> </tr> </tbody> </table>	Name	Download	SDK Package (ZIP)	 instantclient-sdk-linux.x64-21.1.0.0.0.zip	SDK Package (RPM)	 oracle-instantclient-devel-21.1.0.0.0-1.x86_64.rpm	JDBC Supplement Package (ZIP)	 instantclient-jdbc-linux.x64-21.1.0.0.0.zip
Name	Download								
SDK Package (ZIP)	 instantclient-sdk-linux.x64-21.1.0.0.0.zip								
SDK Package (RPM)	 oracle-instantclient-devel-21.1.0.0.0-1.x86_64.rpm								
JDBC Supplement Package (ZIP)	 instantclient-jdbc-linux.x64-21.1.0.0.0.zip								
<p>Once downloaded locally, Secure Copy the file into the Client Instance Home Directory</p> <p><code>scp -i oml4r-db-privatekey.key ./Downloads/instantclient-sdk-linux.x64-21.1.0.0.0.zip opc@<client-ip-address>:/home/opc</code></p>	<pre>\$ cd .ssh \$ scp -i oml4r-db-privatekey.key ./Downloads/instantclient-sdk-linux.x64-21.1.0.0.0.zip :/home/opc instantclient-sdk-linux.x64-21.1.0.0.0.zip 100% 975KB 1.6MB/s 00:00</pre>								
<p>Move the file into the Installation Directory</p> <p><code>sudo mv instantclient-sdk-linux.x64-21.1.0.0.0.zip /oml4rclient_install_dir</code></p>	<pre>\$ ls instantclient-sdk-linux.x64-21.1.0.0.0.zip \$ sudo mv instantclient-sdk-linux.x64-21.1.0.0.0.zip /oml4rclient_install_dir</pre>								

<p>Navigate to your Installation Directory</p> <pre>cd /oml4rclient_install_dir</pre> <p>Unzip the File</p> <pre>sudo unzip instantclient-sdk-linux.x64-21.1.0.0.0.zip</pre>	<pre>\$ cd /oml4rclient_install_dir/ \$ ls instantclient_21_1 instantclient-basic-linux.x64-21.1.0.0.0.zip instantclient-sdk-linux.x64-21.1.0.0.0.zip \$ sudo unzip instantclient-sdk-linux.x64-21.1.0.0.0.zip Archive: instantclient-sdk-linux.x64-21.1.0.0.0.zip inflating: instantclient_21_1/SDK_README creating: instantclient_21_1/sdk/ creating: instantclient_21_1/sdk/demo/ inflating: instantclient_21_1/sdk/demo/occidemod.sql inflating: instantclient_21_1/sdk/demo/setuporamysql.sh inflating: instantclient_21_1/sdk/demo/oraaccess.xml inflating: instantclient_21_1/sdk/demo/occidemo.sql inflating: instantclient_21_1/sdk/demo/demo.mk inflating: instantclient_21_1/sdk/demo/occobjtyp.</pre>						
<p>We now need to install the OML4R packages on the Client Instance.</p> <p>Visit https://www.oracle.com/database/technologies/r-enterprise-downloads.html and download the corresponding Client OML4R Packages, ensure it is the same version as on the Server.</p>	<h2>Oracle Machine Learning for R Documentation</h2> <p>Note: Starting with Oracle Database 18c, to install Oracle Machine Learning for R (formerly Oracle R Machine Learning for R) see Oracle Machine Learning for R Installation (OML4R) and Administration Guide for instructions.</p> <table border="1"> <thead> <tr> <th>Platform</th><th>OML4R 1.5.1 (R-3.6.1) Documentation System Requirements</th></tr> </thead> <tbody> <tr> <td>Windows</td><td> <ul style="list-style-type: none"> Client (6M) Supporting (4M) Server (14M) </td></tr> <tr> <td>Linux 64-bit</td><td> <ul style="list-style-type: none"> Client (5M) Supporting (4M) Server (17M) </td></tr> </tbody> </table>	Platform	OML4R 1.5.1 (R-3.6.1) Documentation System Requirements	Windows	<ul style="list-style-type: none"> Client (6M) Supporting (4M) Server (14M) 	Linux 64-bit	<ul style="list-style-type: none"> Client (5M) Supporting (4M) Server (17M)
Platform	OML4R 1.5.1 (R-3.6.1) Documentation System Requirements						
Windows	<ul style="list-style-type: none"> Client (6M) Supporting (4M) Server (14M) 						
Linux 64-bit	<ul style="list-style-type: none"> Client (5M) Supporting (4M) Server (17M) 						
<p>Once downloaded locally, Secure Copy the file into the Client Instance Home Directory</p> <pre>scp -i oml4r-db-privatekey.key ./Downloads/ore-client-linux-x86-64-1.5.1.zip opc@<client-ip-address>:/home/opc</pre>	<pre>Last login: Fri Feb 12 13:59:12 on ttys001 \$ cd .ssh \$ scp -i oml4r-db-privatekey.key ./Downloads/ore-client-linux-x86-64-1.5.1.zip opc@<client-ip-address>:/home/opc ore-client-linux-x86-64-1.5.1.zip 100% 4912KB 2.5MB/s 00:01</pre>						

<p>Move the file into the Installation Directory</p> <pre>sudo mv ore-client-linux-x86-64-1.5.1.zip /oml4rclient_install_dir</pre>	<pre>\$ cd ~ \$ ls ore-client-linux-x86-64-1.5.1.zip \$ sudo mv ore-client-linux-x86-64-1.5.1.zip /oml4rclient_install_dir</pre>
<p>Navigate to the Installation Directory</p> <pre>cd /oml4rclient_install_dir</pre> <p>Unzip the Client File</p> <pre>sudo unzip ore-client-linux-x86-64-1.5.1.zip</pre>	<pre>\$ cd /oml4rclient_install_dir/ \$ ls instantclient_21_1 instantclient-basic-linux.x64-21.1.0.0.0.zip instantclient-sdk-linux.x64-21.1.0.0.0.zip ore-client-linux-x86-64-1.5.1.zip \$ sudo unzip ore-client-linux-x86-64-1.5.1.zip Archive: ore-client-linux-x86-64-1.5.1.zip inflating: client/ORE_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: client/OREbase_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: client/OREcommon_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: client/OREdm_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: client/OREdplyr_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: client/OREeda_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: client/OREembed_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: client/OREgraphics_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: client/OREmodels_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: client/OREpredict_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: client/OREstats_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz inflating: client/ORExml_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz</pre>
<p>Navigate to the client directory.</p> <pre>cd client</pre> <p>Now Install the OML4R Packages.</p> <pre>sudo R CMD INSTALL ORE....-gnu.tar.gz</pre> <p>For all files within the directory.</p>	<pre>\$ cd client/ \$ sudo R CMD INSTALL ORE_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz * installing to library '/usr/lib64/R/library' * installing *binary* package 'ORE' ... * DONE (ORE) Making 'packages.html' ... done \$ sudo R CMD INSTALL OREbase_1.5.1_R_x86_64-unknown-linux-gnu.tar.gz * installing to library '/usr/lib64/R/library' * installing *binary* package 'OREbase' ... * DONE (OREbase) Making 'packages.html' ... done</pre>

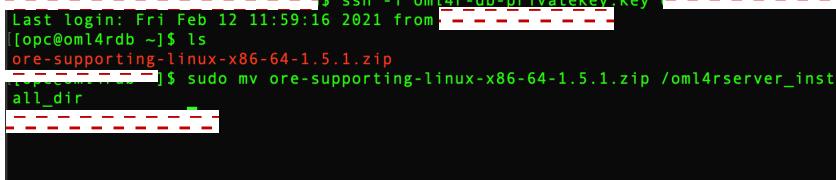
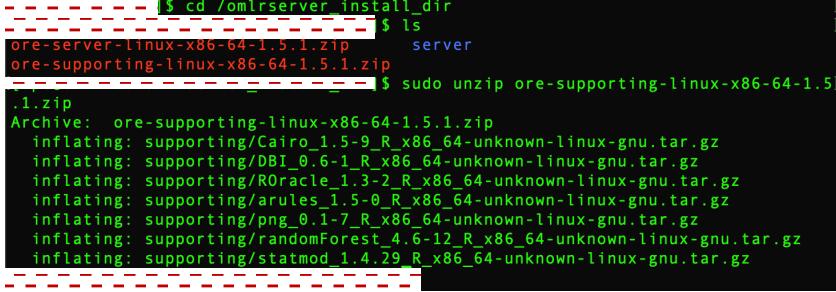
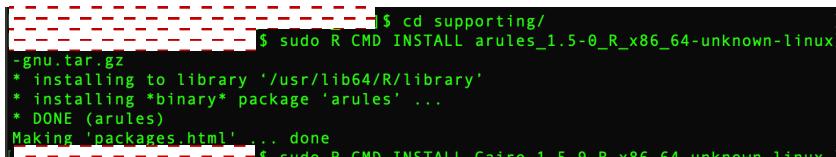
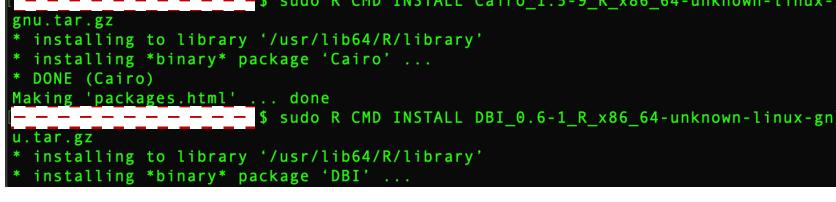
<p>We will now install the OML4R supporting packages on each client computer and on the server that hosts the OML4R Server.</p> <p>Let's first install this onto the Client Instance.</p> <p>Visit (https://www.oracle.com/database/technologies/r-enterprise-downloads.html) and download the corresponding Supporting OML4R Packages, ensure it is the same version as on the Server.</p>	<h2>Oracle Machine Learning for R Documentation</h2> <p>Note: Starting with Oracle Database 18c, to install Oracle Machine Learning for R (formerly Oracle R Oracle Machine Learning for R Installation (OML4R) and Administration Guide for instructions.</p> <table border="1"> <thead> <tr> <th>Platform</th><th>OML4R 1.5.1 (R-3.6.1) Documentation System Requirements</th></tr> </thead> <tbody> <tr> <td>Windows</td><td> Client (6M) Supporting (4M) Server (14M) </td></tr> <tr> <td>Linux 64-bit</td><td> Client (5M) Supporting (4M) Server (17M) </td></tr> </tbody> </table>	Platform	OML4R 1.5.1 (R-3.6.1) Documentation System Requirements	Windows	Client (6M) Supporting (4M) Server (14M)	Linux 64-bit	Client (5M) Supporting (4M) Server (17M)
Platform	OML4R 1.5.1 (R-3.6.1) Documentation System Requirements						
Windows	Client (6M) Supporting (4M) Server (14M)						
Linux 64-bit	Client (5M) Supporting (4M) Server (17M)						
<p>Once downloaded locally, Secure Copy the file into the Client Instance Home Directory</p> <pre>scp -i oml4r-db-privatekey.key ./Downloads/ore-supporting-linux-x86-64-1.5.1.zip opc@<client-ip-address>:/home/opc</pre>							
<p>Move the file into the Installation Directory</p> <pre>sudo mv ore-supporting-linux-x86-64-1.5.1.zip /oml4rclient_install_dir</pre>							

<p>Navigate to the Installation Directory</p> <pre>cd /oml4rclient_install_dir</pre> <p>Unzip the Client File</p> <pre>sudo unzip ore-supporting-linux-x86-64-1.5.1.zip</pre>	<pre>\$ cd /oml4rclient_install_dir \$ ls client instantclient_21_1 instantclient-basic-linux.x64-21.1.0.0.0.zip instantclient-sdk-linux.x64-21.1.0.0.0.zip ore-client-linux-x86-64-1.5.1.zip ore-supporting-linux-x86-64-1.5.1.zip \$ sudo unzip ore-supporting-linux-x86-64-1.5.1.zip Archive: ore-supporting-linux-x86-64-1.5.1.zip inflating: supporting/Cairo_1.5-9_R_x86_64-unknown-linux-gnu.tar.gz inflating: supporting/DBI_0.6-1_R_x86_64-unknown-linux-gnu.tar.gz inflating: supporting/ROracle_1.3-2_R_x86_64-unknown-linux-gnu.tar.gz inflating: supporting/arules_1.5-0_R_x86_64-unknown-linux-gnu.tar.gz inflating: supporting/png_0.1-7_R_x86_64-unknown-linux-gnu.tar.gz inflating: supporting/randomForest_4.6-12_R_x86_64-unknown-linux-gnu.tar.gz inflating: supporting/statmod_1.4.29_R_x86_64-unknown-linux-gnu.tar.gz</pre>
<p>Navigate to the supporting directory.</p> <pre>cd supporting</pre> <p>Now Install the OML4R Packages.</p> <pre>sudo R CMD INSTALL arules...gnu.tar.gz</pre> <p>For all files within the directory.</p>	<pre>\$ cd supporting/ \$ sudo R CMD INSTALL arules_1.5-0_R_x86_64-unknown-linux-gnu.tar.gz * installing to library '/usr/lib64/R/library' * installing *binary* package 'arules' ... * DONE (arules) Making 'packages.html' ... done \$ sudo R CMD INSTALL Cairo_1.5-9_R_x86_64-unknown-linux-gnu.tar.gz * installing to library '/usr/lib64/R/library' * installing *binary* package 'Cairo' ... * DONE (Cairo) Making 'packages.html' ... done \$ sudo R CMD INSTALL DBI_0.6-1_R_x86_64-unknown-linux-gnu.tar.gz * installing to library '/usr/lib64/R/library' * installing *binary* package 'DBI' ... * DONE (DBI)</pre>
<p>When using Linux the Cairo and png packages require the presence of two OS dependencies:</p> <ul style="list-style-type: none"> • Cairo-devel • Libpng-devel <p>To verify we they are installed we can run:</p> <pre>rpm -qa libpng-devel rpm -qa cairo-devel</pre> <p>If they are installed, then the name of the RPM is returned. If not, then run the below:</p> <pre>yum install cairo-devel yum install libpng-devel</pre>	<pre>\$ sudo rpm -qa cairo-devel cairo-devel-1.15.12-4.el7.x86_64 \$ sudo rpm -qa libpng-devel libpng-devel-1.5.13-8.el7.x86_64</pre>



<p>The final thing we need to do is set the LD_LIBRARY_PATH environment variable.</p> <p>Navigate to: <i>cd /etc/profile.d</i></p> <p>Create an .sh file which sets the variable system wide.</p> <p><i>sudo vi ore_ld_library_path.sh</i></p> <p>Enter the following in the file: <i>export LD_LIBRARY_PATH=/oml4rclient_install_dir/instantclient_18_5/</i></p> <p>We can now exit the Client Instance.</p>	<pre>\$ cd /etc/profile.d \$ sudo vi ore_ld_library_path.sh \$ cat ore_ld_library_path.sh export LD_LIBRARY_PATH=/oml4rclient_install_dir/instantclient_18_5/ \$</pre> <p>If you set this variable under <code>~/.bashrc</code> it is not recognised system wide and hence will cause issues later when trying to load in the ORE library.</p> <p>You can query <i>printenv</i> from the <i>cmd</i> to check that it is set.</p>
<p>Now let's repeat the above and install the Supporting packages within the DB Server.</p> <p>As we have already downloaded the Supporting packages, Secure Copy the file into the DB Server Home Directory</p> <p><u><i>scp -i oml4r-db-privatekey.key/Downloads/ore-supporting-linux-x86-64-1.5.1.zip opc@<db-ip-address>:/home/opc</i></u></p>	<pre>\$ scp -i oml4r-db-privatekey.key/Downloads/ore-supporting-linux-x86-64-1.5.1.zip opc@<db-ip-address>:/home/opc ore-supporting-linux-x86-64-1.5.1.zip 100% 4080KB 4.0MB/s 00:01</pre>



<p>SSH into our DB Server Instance.</p> <pre>ssh -i <private_key> opc@<db_ip_address></pre> <p>Move the file into the Installation Directory</p> <pre>sudo mv ore-supporting-linux-x86-64-1.5.1.zip /oml4rserver_install_dir</pre>	
<p>Navigate to the Installation Directory</p> <pre>cd /oml4rserver_install_dir</pre> <p>Unzip the Client File</p> <pre>sudo unzip ore-supporting-linux-x86-64-1.5.1.zip</pre>	
<p>Switch user to oracle.</p> <pre>sudo su - oracle</pre> <p>Navigate to the supporting directory.</p> <pre>cd supporting</pre> <p>Now Install the OML4R Packages.</p> <pre>sudo R CMD INSTALL arules...gnu.tar.gz</pre> <p>For all files within the directory.</p>	  

<p>When using Linux the Cairo and png packages require the presence of two OS dependencies:</p> <ul style="list-style-type: none"> • Cairo-devel • Libpng-devel <p>To verify we they are installed we can run: rpm -qa libpng-devel rpm -qa cairo-devel</p> <p>If they are installed, then the name of the RPM is returned. If not, then run the below: yum install cairo-devel yum install libpng-devel</p>	<pre>\$ rpm -qa libpng-devel libpng-devel-1.5.13-8.el7.x86_64 \$ rpm -qa cairo-devel cairo-devel-1.15.12-4.el7.x86_64 \$</pre>
<p>Now let's test that all the OML4R components are working on the DB OML4R Server.</p> <p>SSH into our DB Server.</p> <p>ssh -i <private_key> opc@<db_ip_address></p> <p>Switch to the oracle user: sudo su - oracle</p> <p>Start the ORE terminal: ORE</p>	<pre>\$ ssh -i oml4r-db-privatekey.key Last login: Sat Feb 13 17:59:24 2021 from [REDACTED] [REDACTED] \$ sudo su - oracle Last login: Sat Feb 13 18:00:24 UTC 2021 [REDACTED] \$ ORE</pre>
<p>Load in the ORE library from the command prompt:</p> <p>library(ORE)</p>	<pre>You are using Oracle's distribution of R. Please contact Oracle Support for any problems you encounter with this distribution. > library(ORE)</pre>

Let's connect to the Database via the ORE Library using the ISMAIL user we created earlier.

```
ore.connect(user="ISMAIL",
            host="localhost",
            password="",
            service_name="",
            port=1521, all=TRUE)
```

```
> ore.connect(user="ISMAIL", host="localhost", password="XXXXXXXXXX", service_
> _name="XXXXXXXXXX", port=1521, all=TRUE)
> [1]
```

Execute the following R Code:

List tables within DB:
ore.ls()

Retrieve Table from DB into an R Data Frame Proxy Object:
mytable_proxy <- ore.pull(MYTABLE)

Display the class of the proxy object:
class(mytable_proxy)

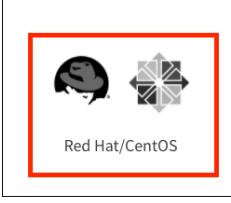
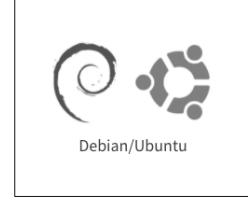
Display the Head of the Data Frame:
head(mytable_proxy)

Now we have tested the OML4R components work, we can quit the console.

q()

```
> ore.ls()
[1] "MYTABLE"
>
> mytable_proxy <- ore.pull(MYTABLE)
Warning message:
ORE object has no unique key - using random order
>
> class(mytable_proxy)
[1] "data.frame"
>
> head(mytable_proxy)
  PERSON AGE
1 ismail  23
2 ismail  21
> [1]
```

<p>Now let's test our connection to see if we can connect an OML4R Client to an OML4R Server.</p> <p>SSH into our Client Compute Instance.</p> <p>ssh -i <private_key> opc@<ip_address></p> <p>Set the R Home</p> <p>Start R</p> <p>R</p>	<pre>Oracle Distribution of R version 3.6.1 (--) -- "Action of the Toes" Copyright (C) The R Foundation for Statistical Computing Platform: x86_64-pc-linux-gnu (64-bit) R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under certain conditions. Type 'license()' or 'licence()' for distribution details. Natural language support but running in an English locale R is a collaborative project with many contributors. Type 'contributors()' for more information and 'citation()' on how to cite R or R packages in publications. Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help. Type 'q()' to quit R. You are using Oracle's distribution of R. Please contact Oracle Support for any problems you encounter with this distribution. > </pre>
<p>Load in the ORE Script</p> <p>library(ORE)</p>	<pre>You are using Oracle's distribution of R. Please contact Oracle Support for any problems you encounter with this distribution. > library(ORE) </pre>
<p>Let's connect to the Database via the ORE Library using the ISMAIL user we created earlier.</p> <p>ore.connect(user="ISMAIL", host="", password="", service_name="", port=1521, all=TRUE)</p>	<pre>> ore.connect(user="ISMAIL", host="████████", password="████████#", servl > e_name="████████", port=1521, all=TRUE) > > </pre>

<p>Execute the following R Code:</p> <p>List tables within DB: <i>ore.ls()</i></p> <p>Retrieve Table from DB into an R Data Frame Proxy Object: <i>mytable_proxy <- ore.pull(MYTABLE)</i></p> <p>Display the class of the proxy object: <i>class(mytable_proxy)</i></p> <p>Display the Head of the Data Frame: <i>head(mytable_proxy)</i></p> <p>Now we have tested the OML4R components work, we can quit the console. <i>q()</i></p>	<pre>> ore.ls() [1] "MYTABLE" > > mytable_proxy <- ore.pull(MYTABLE) Warning message: ORE object has no unique key - using random order > > class(mytable_proxy) [1] "data.frame" > > head(mytable_proxy) PERSON AGE 1 ismail 23 2 ismail 21 > </pre>
<p>We can now install R Studio Server on to our Client to allow multiple users to access R Studio via the Browser.</p> <p>Visit the Download Page and select Red Hat/CentOS download for instructions. Oracle Linux Source Code is that of Red Hat.</p>	<h3>Download RStudio Server v1.4.1103</h3> <p>RStudio Server enables you to provide a browser based interface to a version of R running on a remote Linux server, bringing the power and productivity of the RStudio IDE to server-based deployments of R.</p> <p>Do you need support or a commercial license? Compare our commercial and open source products.</p> <p>Managing Packages</p> <p>To install packages on Linux faster and easier, consider RStudio Package Manager.</p> <p>Choose your Linux Platform:</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Red Hat/CentOS</p> </div> <div style="text-align: center;">  <p>Debian/Ubuntu</p> </div> <div style="text-align: center;">  <p>SLES/openSUSE</p> </div> </div> <p>“</p>

Navigate to the /oml4rclient_install_dir

```
cd
/oml4rclient_install_dir
```

```
$ cd /oml4rclient_install_dir/
$ ls
client
instantclient_18_5
instantclient_21_1
instantclient-basic-linux.x64-18.5.0.0.0dbru.zip
instantclient-basic-linux.x64-21.1.0.0.0.zip
instantclient-sdk-linux.x64-18.5.0.0.0dbru.zip
instantclient-sdk-linux.x64-21.1.0.0.0.zip
ore-client-linux-x86-64-1.5.1.zip
ore-supporting-linux-x86-64-1.5.1.zip
supporting
$
```

Based on the instructions on the R Studio download page, execute the following to Download the R Studio RPM.

```
sudo wget
https://download2.rstudio.org/server/centos7/x86_64/rstudio-server-rhel-1.4.1103-x86_64.rpm
```

```
$ sudo wget https://download2.rstudio.org/server/centos7/x86_64/rstudio-server-rhel-1.4.1103-x86_64.rpm
--2021-02-15 16:19:24--  https://download2.rstudio.org/server/centos7/x86_64/rstudio-server-rhel-1.4.1103-x86_64.rpm
Resolving download2.rstudio.org (download2.rstudio.org)... 13.35.15.98, 13.35.15.122, 13.35.15.36, ...
Connecting to download2.rstudio.org (download2.rstudio.org)|13.35.15.98|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 65355644 (62M) [application/x-redhat-package-manager]
Saving to: 'rstudio-server-rhel-1.4.1103-x86_64.rpm'

100%[=====] 65,355,644 308KB/s in 3m 12s
2021-02-15 16:22:37 (332 KB/s) - 'rstudio-server-rhel-1.4.1103-x86_64.rpm' saved
[65355644/65355644]
$
```

Once the R Studio Server RPM has downloaded, we can then install it.

```
sudo yum install
rstudio-server-rhel-1.4.1103-x86_64.rpm
```

```
Main PID: 10955 (rserver)
Memory: 14.0M
CGroup: /system.slice/rstudio-server.service
└─10955 /usr/lib/rstudio-server/bin/rserver

Feb 15 16:30:40 oml4r-client systemd[1]: Starting RStudio Server...
Feb 15 16:30:40 oml4r-client systemd[1]: Started RStudio Server.
   Verifying : psmisc-22.20-17.el7.x86_64                                     1/3
   Verifying : rstudio-server-1.4.1103-1.x86_64                               2/3
   Verifying : postgresql-libs-9.2.24-4.el7_8.x86_64                         3/3

Installed:
  rstudio-server.x86_64 0:1.4.1103-1

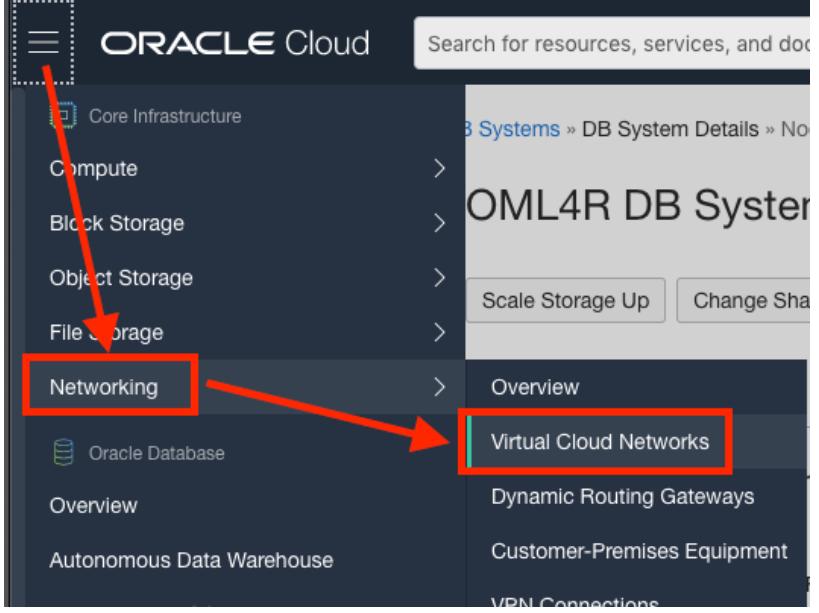
Dependency Installed:
  postgresql-libs.x86_64 0:9.2.24-4.el7_8      psmisc.x86_64 0:22.20-17.el7

Complete!
```



<p>Once installed we then need to create the /etc/rstudio/rserver.conf file and set the R_HOME and ORACLE_HOME</p> <pre>sudo vi /etc/rstudio/rserver.conf</pre> <p>rsession-ld-library-path=/usr/lib64/R/lib:/oml4rclient_install_dir/instantclient_18_5</p>	<pre>\$ cd ~ [sudo vi /etc/rstudio/rserver.conf] [oracle@mlr-client ~]\$ sudo cat /etc/rstudio/rserver.conf # Server Configuration File rsession-ld-library-path=/usr/lib64/R/lib:/oml4rclient_install_dir/instantclient_18_5</pre>
<p>Now create the configuration file /usr/lib64/R/etc/Renviron.site and supply the values for ORACLE_HOME, ORACLE_HOSTNAME, ORACLE_SID</p> <pre>cd /usr/lib64/R/etc sudo vi Renviron.site</pre>	<pre>\$ cd /usr/lib64/R/etc \$ ls javaconf ldpaths Makeconf Renviron repositories \$ sudo vi Renviron.site ORACLE_HOME=/oml4rclient_install_dir/instantclient_18_5 ORACLE_HOSTNAME=[REDACTED] ORACLE_SID=OML4R</pre> <p>To get your SID you can query your DB using - select sys_context('userenv','instance_name') from dual;</p> <p>ORACLE_HOME=/oml4rclient_install_dir/instantclient_18_5 ORACLE_HOSTNAME=DB IP ADDRESS ORACLE_SID=OML4R</p>
<p>Now we can restart the R Studio Server.</p> <pre>sudo rstudio-server restart</pre>	<pre>\$ sudo rstudio-server restart</pre>



<p>R Studio Server runs on Port 8787 so we must now edit our Client Compute Instance Subnet to allow traffic into this Port.</p> <p>From the OCI Console navigate to Menu > Networking > Virtual Cloud Networks.</p>																
<p>Click on our VCN.</p>	<p>Virtual Cloud Networks in Ismail C</p> <p>Virtual Cloud Networks are virtual, private networks that you set up in your region.</p> <table border="1"> <thead> <tr> <th colspan="2">Create VCN</th> <th>Start VCN Wizard</th> </tr> <tr> <th>Name</th> <th>State</th> <th></th> </tr> </thead> <tbody> <tr> <td>[REDACTED]</td> <td>Available</td> <td></td> </tr> <tr> <td>[REDACTED]</td> <td>Available</td> <td></td> </tr> <tr> <td>[REDACTED]</td> <td>Available</td> <td></td> </tr> </tbody> </table>	Create VCN		Start VCN Wizard	Name	State		[REDACTED]	Available		[REDACTED]	Available		[REDACTED]	Available	
Create VCN		Start VCN Wizard														
Name	State															
[REDACTED]	Available															
[REDACTED]	Available															
[REDACTED]	Available															
<p>Select 'Security Lists'</p>	<p>Resources</p> <ul style="list-style-type: none"> Subnets (2) CIDR Blocks (1) Route Tables (2) Internet Gateways (1) Dynamic Routing Gateways (0) Network Security Groups (0) Security Lists (2) DHCP Options (1) <p>Subnets in Ismail C</p> <table border="1"> <thead> <tr> <th colspan="2">Create Subnet</th> </tr> <tr> <th>Name</th> <th></th> </tr> </thead> <tbody> <tr> <td>Private Subnet</td> <td></td> </tr> <tr> <td>Public Subnet</td> <td></td> </tr> </tbody> </table>	Create Subnet		Name		Private Subnet		Public Subnet								
Create Subnet																
Name																
Private Subnet																
Public Subnet																



	<h3>Security Lists in Ismail Compartment</h3> <p>Create Security List</p> <p>Name</p> <p>Security List for Private Subnet </p> <p>Default Security List for </p>												
	<h3>Ingress Rules</h3> <p>Add Ingress Rules Edit Remove</p> <table border="1"> <thead> <tr> <th></th> <th>Stateless ▾</th> <th>Source</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>No</td> <td>0.0.0.0/0</td> </tr> <tr> <td><input type="checkbox"/></td> <td>No</td> <td>0.0.0.0/0</td> </tr> <tr> <td><input type="checkbox"/></td> <td>No</td> <td> </td> </tr> </tbody> </table>		Stateless ▾	Source	<input type="checkbox"/>	No	0.0.0.0/0	<input type="checkbox"/>	No	0.0.0.0/0	<input type="checkbox"/>	No	
	Stateless ▾	Source											
<input type="checkbox"/>	No	0.0.0.0/0											
<input type="checkbox"/>	No	0.0.0.0/0											
<input type="checkbox"/>	No	 											
	<p>Enter the following information:</p> <p>Source Type: CIDR Source CIDR: 0.0.0.0/0 IP Protocol: TCP Destination Port: 8787</p> <p>Click 'Add Ingress Rule'.</p> <p>Ingress Rule 1</p> <p>Allows TCP traffic 8787</p> <p><input type="checkbox"/> STATELESS <small>(i)</small></p> <p>SOURCE TYPE: CIDR <small>(i)</small></p> <p>SOURCE CIDR: 0.0.0.0/0 <small>(i)</small> <small>Specified IP addresses: 0.0.0.255,255.255.255 (4,294,967,296 IP addresses)</small></p> <p>IP PROTOCOL: TCP <small>(i)</small></p> <p>SOURCE PORT RANGE: OPTIONAL <small>(i)</small> <input type="text" value="All"/> <small>Examples: 80, 20-22</small></p> <p>DESTINATION PORT RANGE: OPTIONAL <small>(i)</small> <input type="text" value="8787"/> <small>Examples: 80, 20-22</small></p> <p>DESCRIPTION: OPTIONAL <input type="text" value="Allow Incoming RStudio Server Traffic"/> <small>Maximum 255 characters</small></p> <p>+ Another Ingress</p> <p>Add Ingress Rules Cancel</p>												

We now need to enable Port Forwarding on our local laptop to be able to connect to our RStudio Server running on our Client Instance.

Running on a Mac:
Navigate to `~/ssh` folder.

`cd ~/ssh`

Create or edit the config file

`vi config`

Enter the port forward details:

```
Host rstudio-server
  HostName XXX.XXX.XXX.XX
  User opc
  IdentityFile /Users/isyed/.ssh/oml4r-db-privatekey.key
  LocalForward 8787 localhost:8787
```

```
$ cd ~/.ssh
$ vi config
$ cat config
Host bastion
  HostName [REDACTED]
  User opc
  IdentityFile /Users/isyed/.ssh/[REDACTED].key

Host rdg-server
  HostName [REDACTED]
  User opc
  IdentityFile /Users/isyed/.ssh/[REDACTED].key
  ProxyJump opc@bastion
  LocalForward 8080 localhost:8080

Host rstudio-server
  HostName [REDACTED]
  User opc
  IdentityFile /Users/isyed/.ssh/[REDACTED]-privatekey.key
  LocalForward 8787 localhost:8787
```

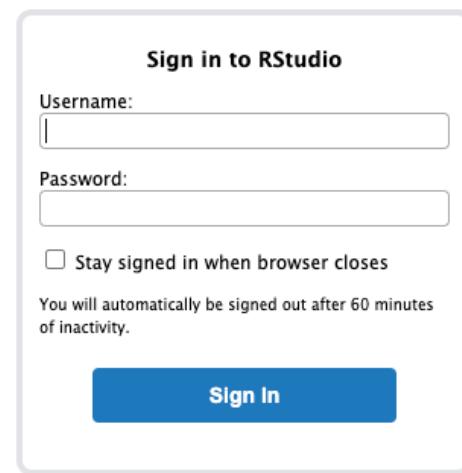
Now let's open up the SSH tunnel.

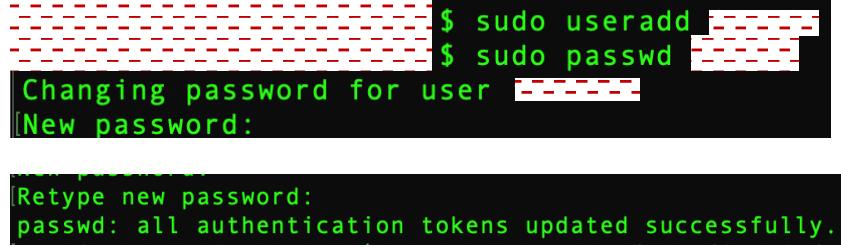
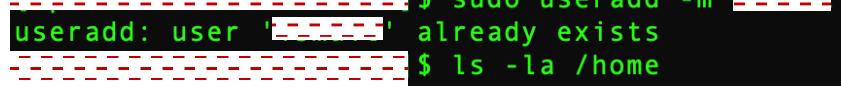
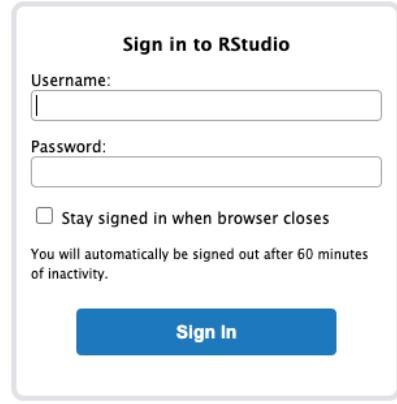
`ssh rstudio-server`

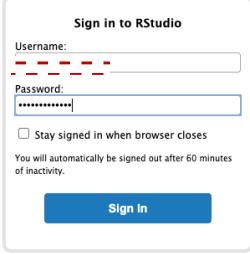
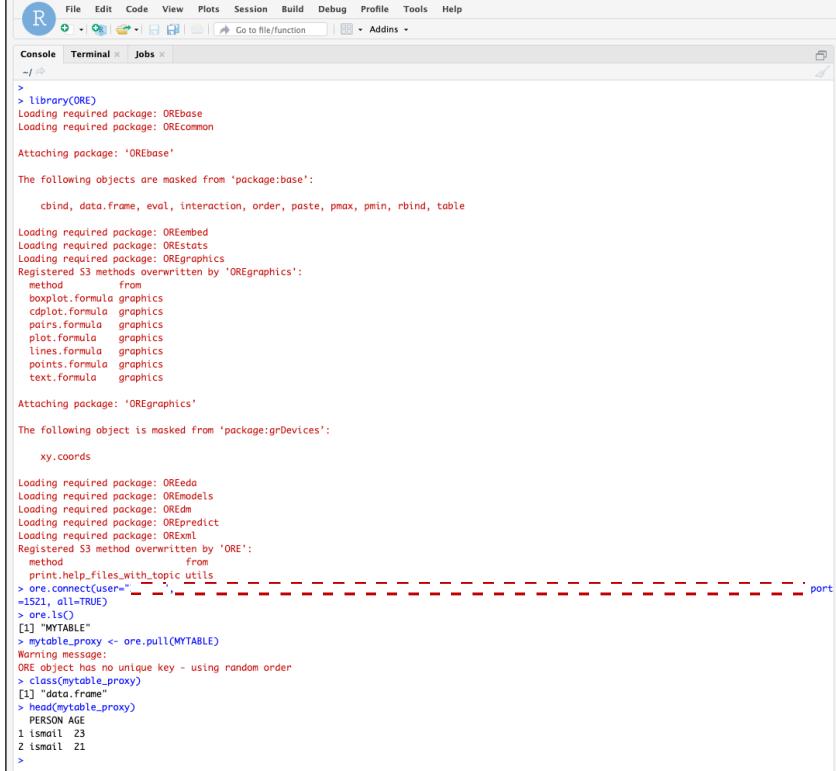
On your local laptop open up your browser and visit:

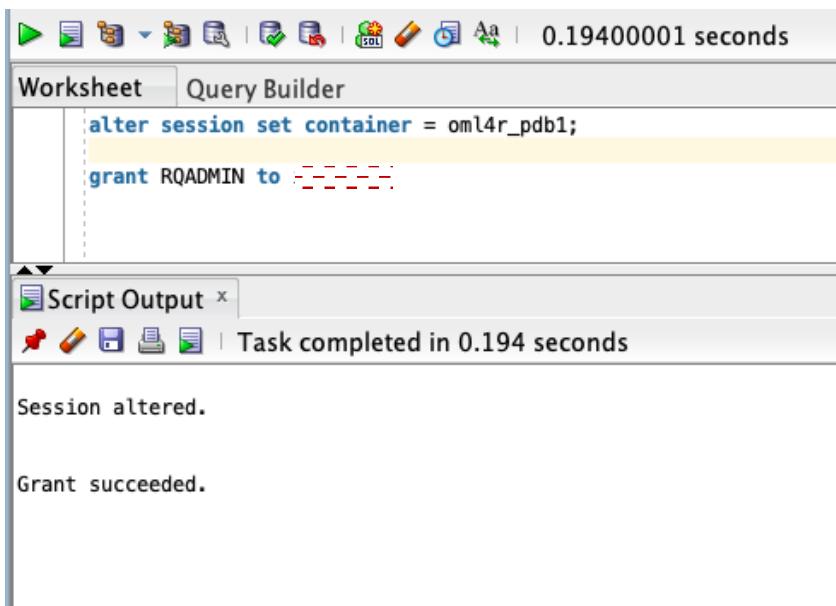
<http://localhost:8787>

You should see the RStudio Server Login Page.



<p>To login to R Studio Server, we need to create Users on the Client Compute Instance.</p> <p>SSH into our Client Compute Instance.</p> <p>ssh -i <private_key> opc@<ip_address></p>	 <pre>\$ ssh -i oml4r-db-privatekey.key im Last login: Mon Feb 15 18:08:00 2021 from im.net</pre>
<p>Create a new User.</p> <p>sudo useradd <user></p> <p>Assign user a password.</p> <p>sudo passwd <password></p> <p>Enter Password Confirm Password</p>	 <pre>\$ sudo useradd \$ sudo passwd Changing password for user >New password: [New password] Retype new password: passwd: all authentication tokens updated successfully.</pre>
<p>Create the Home directory for your user</p> <p>sudo useradd -m <username></p>	 <pre>\$ sudo useradd -m useradd: user 'im' already exists \$ ls -la /home</pre>
<p>Exit from your Client Shell and create your SSH Tunnel.</p> <p>ssh rstudio-server</p> <p>On your local laptop open up your browser and visit: http://localhost:8787</p> <p>You should see the RStudio Server Login Page.</p>	 <p>The screenshot shows the "Sign in to RStudio" interface. It has fields for "Username" and "Password", a checkbox for "Stay signed in when browser closes" (unchecked), and a note stating "You will automatically be signed out after 60 minutes of inactivity." A blue "Sign In" button is at the bottom.</p>

<p>Enter in your newly created Username and Password.</p> <p>Click Sign In</p>	
<p>Now you can test the connectivity to the OML4R Server DB.</p> <p><i>library(ORE)</i></p> <p>Let's connect to the Database via the ORE Library using the ISMAIL user we created earlier.</p> <pre>ore.connect(user=" ", host="", password=" ", service_name=" ", port=1521, all=TRUE)</pre> <p>List tables within DB: <i>ore.ls()</i></p> <p>Retrieve Table from DB into an R Data Frame Proxy Object: <i>mytable_proxy <- ore.pull(MYTABLE)</i></p> <p>Display the class of the proxy object: <i>class(mytable_proxy)</i></p> <p>Display the Head of the Data Frame: <i>head(mytable_proxy)</i></p>	 <pre> > library(ORE) Loading required package: OREbase Loading required package: OREcommon Attaching package: 'OREbase' The following objects are masked from 'package:base': cbind, data.frame, eval, interaction, order, paste, pmax, pmin, rbind, table Loading required package: OREembed Loading required package: OREstats Loading required package: OREgraphics Registered S3 methods overwritten by 'OREgraphics': method from boxplot.formula graphics cdplot.formula graphics pairs.formula graphics plot.formula graphics lines.formula graphics points.formula graphics text.formula graphics Attaching package: 'OREgraphics' The following object is masked from 'package:grDevices': xy.coords Loading required package: OREeda Loading required package: OREmodels Loading required package: OREdm Loading required package: OREPredict Loading required package: ORExml Registered S3 method overwritten by 'ORE': method from print.help_files_with_topic.utils > ore.connect(user=" ", host=" ", password=" ", service_name=" ", port=1521, all=TRUE) > ore.ls() [1] "MYTABLE" > mytable_proxy <- ore.pull(MYTABLE) Warning message: ORE object has no unique key - using random order > class(mytable_proxy) [1] "data.frame" > head(mytable_proxy) PERSON AGE 1 ismail 23 2 ismail 21 > </pre>

<p>The final option thing you might want to do is to grant your DB user the RQADMIN role to allow the DB user to be responsible for creating and managing the R Script Repository.</p> <p>Login to SQLDeveloper as SYS</p> <p>Alter Session to Pluggable DB: <i>alter session set container = oml4r_pdb1;</i></p> <p>Grant Role to User: <i>grant RQADMIN to <username>;</i></p>	 <p>The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, two SQL statements are visible:</p> <pre>alter session set container = oml4r_pdb1; grant RQADMIN to <username>;</pre> <p>Below the worksheet is the Script Output window, which displays the results of the execution:</p> <pre>Session altered. Grant succeeded.</pre>
Your Installation of OML4R on 19c Oracle Database Cloud with Client Machine is Complete.	SUCCESSFUL