

SQL Server vNext on Linux Ubuntu - Part 1

Built on SQL Server 2016, SQL Server vNext represents a major step towards making SQL Server a cross operating systems and bringing the power of SQL Server to Linux, in part 1 of this article we will go through the processes to install SQL Server vNext and related tools on Linux Ubuntu the world's most popular cloud operating system, Ubuntu is more than twice as popular on the Amazon cloud as all other operating systems combined.

To install SQL Server vNext on Ubuntu follow these steps

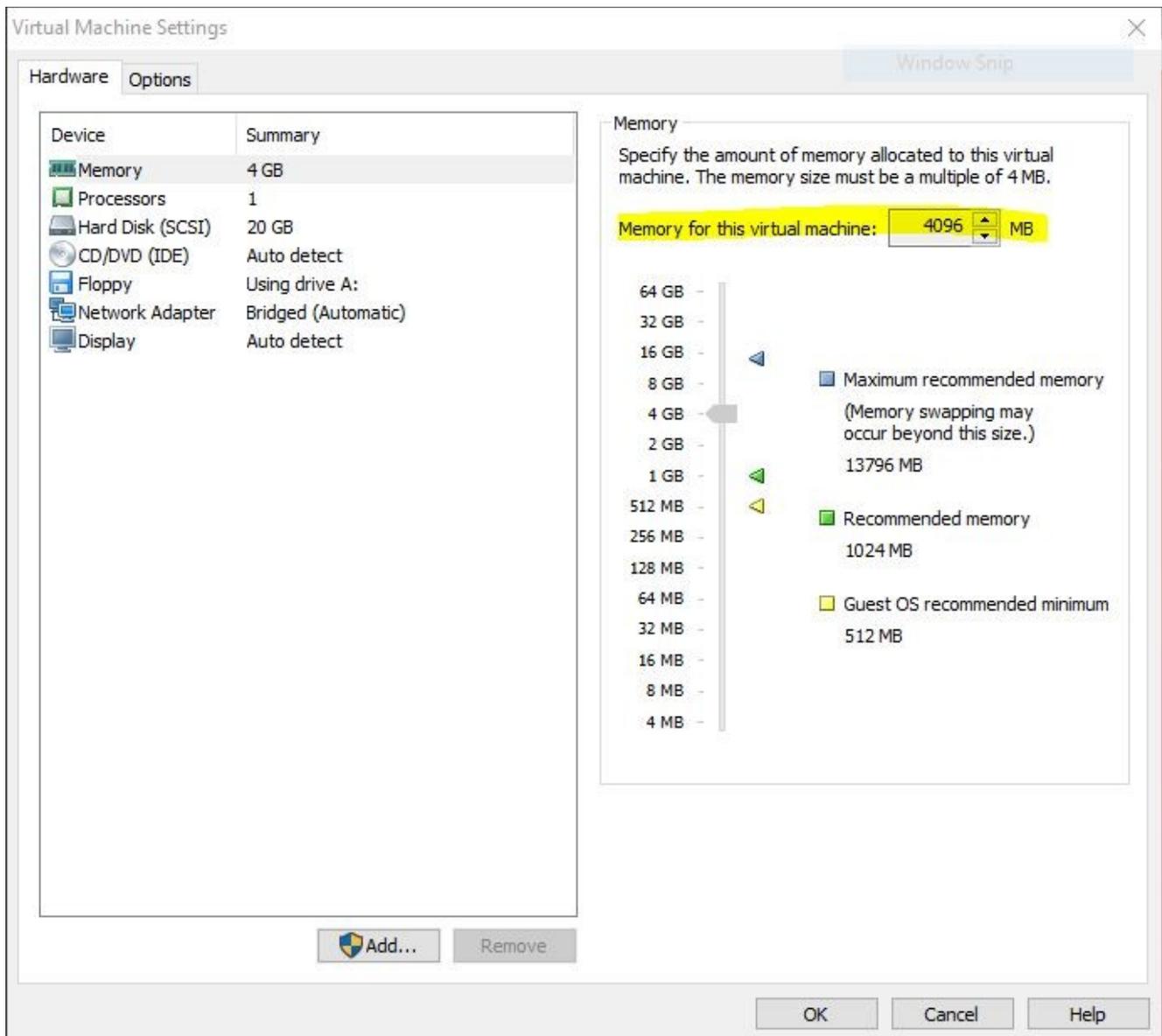
1. Download Ubuntu 16.04 from <http://ubuntu-cd.mirror.iweb.ca/16.04/>

In my case, I selected the desktop image as this will also install the graphical user interface.

2. From your virtualization software, creates a new VM using .iso file downloaded in the previous step

- Your VM should have at least 3.25 GB, to ensure optimal performance of SQL Server set your VM at 4 GB or more
- Make sure your VM is connected to internet, this is necessary to download packages from Microsoft site

In my case I have used VMware Workstation to create my Ubuntu VM, you can use free virtualization tools such VirtualBox, Xen, Hyper-V or VMware Workstation Player



3. Start your VM and launch Ubuntu terminal console

We will use Ubuntu terminal, but later I will show you how to execute Linux commands from Visual Studio Code and Putty, to use Linux terminal in Ubuntu

- Keyboard Shortcut: Ctrl + Alt + T

- Launcher + Search and enter terminal word (to create a shortcut drag the icon to the Launcher bar)



4. Ensure your system is fully updated by running the following command from Ubuntu Terminal console

```
$ sudo apt-get update
```

```
mbouarroudj@Ubuntu1604:~$ sudo apt-get update
[sudo] password for mbouarroudj:
Hit:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:2 http://ca.archive.ubuntu.com/ubuntu xenial InRelease
Hit:3 http://ca.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:4 http://ca.archive.ubuntu.com/ubuntu xenial-backports InRelease
Hit:5 https://packages.microsoft.com/ubuntu/16.04/mssql-server xenial InRelease
Reading package lists... Done
```

5. Switch to root user to run commands with root privileges

```
$ sudo su
```

6. Add the repository information for SQL Server to `/etc/apt/sources.list.d/`

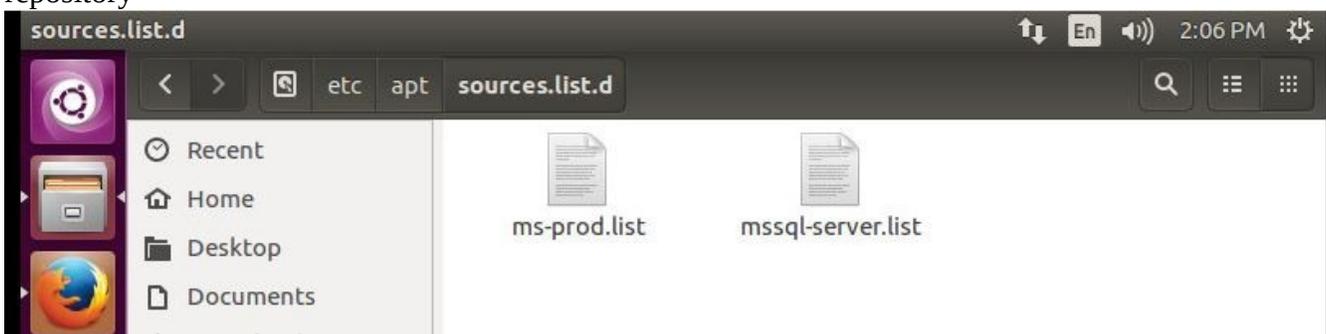
```
# curl https://packages.microsoft.com/config/ubuntu/16.04/mssql-server.list >
/etc/apt/sources.list.d/mssql-server.list
# curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
# exit
```

curl is a command line tool for transferring data from or to a server, if you have used the indicated link for Ubuntu iso file the tool is already installed, otherwise, you should install it using:

```
sudo apt-get install curl
```

```
mbouarroudj@Ubuntu1604:~$ sudo su
root@Ubuntu1604:/home/mbouarroudj# curl https://packages.microsoft.com/config/ubuntu/16.04/mssql-server.list > /etc/apt/sources.list.d/mssql-server.list
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100    87    100    87    0    0    55    0  0:00:01  0:00:01  --:--:--   55
root@Ubuntu1604:/home/mbouarroudj# curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100   983    100   983    0    0  1195    0  --:--:--  --:--:--  --:--:--  1194
OK
```

when you perform the indicated commands, mssql-server.list is copied to etc/apt/sources.list.d/ repository



7. Run the following commands to prepare the repository data and install SQL Server vNext on Linux Ubuntu

```
$ sudo apt-get update
$ sudo apt-get install mssql-server
```

sudo: will perform the following command with root (super-user) capabilities

apt-get: is a command line tool used by Ubuntu to manage software packages

update: is an option to update packages lists to ensure that apt-get will install the most recent version of the package

install: the option to install mssql-server as it is currently listed in local software list

```

(Reading database ... 237295 files and directories currently installed.)
Preparing to unpack .../mssql-server_14.0.100.187-1_amd64.deb ...
Unpacking mssql-server (14.0.100.187-1) ...
Processing triggers for libc-bin (2.23-0ubuntu3) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up libc++1:amd64 (3.7.0-1) ...
Setting up libjemalloc1 (3.6.0-9ubuntu1) ...
Setting up mssql-server (14.0.100.187-1) ...

+-----+
| Please run /opt/mssql/bin/sqlservr-setup to complete the setup of |
|                               Microsoft(R) SQL Server(R).           |
+-----+

Processing triggers for libc-bin (2.23-0ubuntu3) ...
mbouarroudj@Ubuntu1604:~$ █

```

8. Run the configuration script to accept the license agreement and provide the System Administrator (SA) password (SQL Server vNext will run with SQL authentication mode only)

```
$ sudo /opt/mssql/bin/sqlservr-setup
```

```

mbouarroudj@Ubuntu1604:~$ sudo /opt/mssql/bin/sqlservr-setup
Microsoft(R) SQL Server(R) Setup

You can abort setup at anytime by pressing Ctrl-C. Start this program
with the --help option for information about running it in unattended
mode.

Please enter a password for the system administrator (SA) account:
Please confirm the password for the system administrator (SA) account:

Setting system administrator (SA) account password...

Do you wish to start the SQL Server service now? [y/n]: █

```

```

Do you wish to start the SQL Server service now? [y/n]: y
Do you wish to enable SQL Server to start on boot? [y/n]: y
Created symlink from /etc/systemd/system/multi-user.target.wants/mssql-server.service to /lib/systemd/system/mssql-server.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/mssql-server-telemetry.service to /lib/systemd/system/mssql-server-telemetry.service.

Setup completed successfully.

```

You should be prompted to start and enable SQL Server Service, if not run the following commands to enable and start the service

```

$ sudo systemctl enable mssql-server
$ sudo systemctl start mssql-server

```

Please note that SQL Agent is not available yet

9. You can now check and make sure SQL server is listening on 1433 port

```
$ sudo netstat -peant | grep ":1433"
```

```
mbouarroudj@Ubuntu1604:~$ sudo netstat -peant | grep ":1433"
tcp        0      0 0.0.0.0:1433        0.0.0.0:*          LISTEN      997
           36815          4854/sqlservr
```

You can also run the following command to view the status of SQL Server service

```
$ sudo systemctl status mssql-server
```

```
mbouarroudj@Ubuntu1604:~$ sudo systemctl status mssql-server
[sudo] password for mbouarroudj:
● mssql-server.service - Microsoft(R) SQL Server(R) Database Engine
   Loaded: loaded (/lib/systemd/system/mssql-server.service; enabled; vendor pre
   Active: active (running) since Sun 2016-12-25 13:26:31 EST; 42min ago
   Main PID: 4831 (sqlservr)
   CGroup: /system.slice/mssql-server.service
           └─4831 /opt/mssql/bin/sqlservr
             └─4854 /opt/mssql/bin/sqlservr
```

Note

if 1433 port is not opened you need to open it

```
$ sudo firewall-cmd --zone=public --add-port=1433/tcp --permanent
```

```
$ sudo firewall-cmd --reload
```

Install the command-line tools and ODBC drivers

10. Install the command-line tools (SQLCMD and BCP) and also ODBC drivers

Same as we did to install SQL Server vNext, we have to add the repository information

```
$ sudo su
```

```
# curl https://packages.microsoft.com/config/ubuntu/16.04/prod.list > /etc/apt/sources.list.d/ms-  
prod.list
```

```
# exit
```

```
mbouarroudj@Ubuntu1604:~$ sudo su
[sudo] password for mbouarroudj:
root@Ubuntu1604: /home/mbouarroudj# curl https://packages.microsoft.com/config/ubuntu/16
.04/prod.list > /etc/apt/sources.list.d/ms-prod.list
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  79  100    79    0    0    33      0  0:00:02  0:00:02  --:--:--   33
root@Ubuntu1604: /home/mbouarroudj#
```


Migrate a SQL Server databases on Windows to SQL Server vNext on Linux

The recommended Microsoft technique to migrate SQL Server databases on Windows to SQL Server vNext on Linux is to use SQL Server backup and restore;

Let's terminate the first part of this article by migrating a SQL Server database from SQL Server 2014 on windows to SQL vNext on Linux Ubuntu. There are many way to transfer files from Windows to Linux, the easiest way is to use Putty a free SSH and telnet client for Windows

To restore a database backup on Linux Ubuntu machine follow these steps

1. In your windows machine download Putty.zip and unzip the file
<http://www.putty.org/>

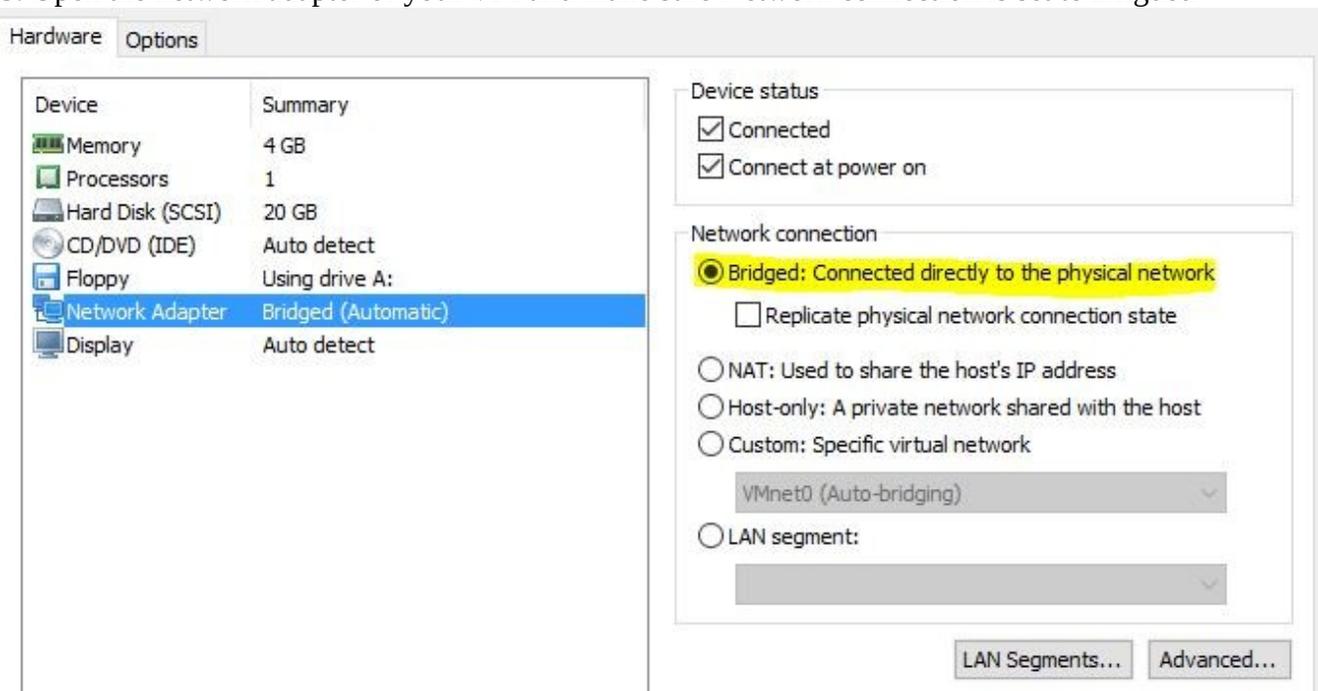
Please note that pscp.exe is the console version when putty.exe is the shell version

2. Setting up SSH and open a port 22 by running the following commands in your Ubutu Terminal console

```
$ sudo apt-get install openssh-server
```

```
$ sudo ufw allow 22
```

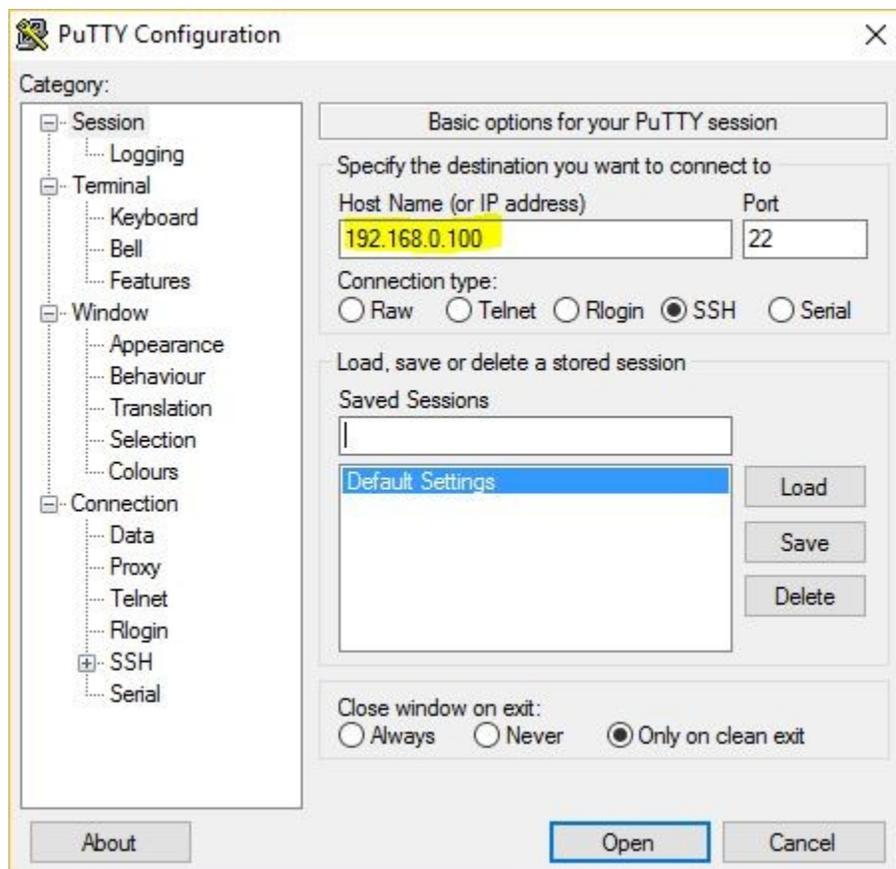
3. Open the network adapter of your VM and make sure Network connection is set to Bridged



4. Get the ip adr of your VM by running
\$ ifconfig

```
mbouarroudj@Ubuntu1604:~$ ifconfig
ens160  Link encap:Ethernet  HWaddr 00:0c:29:16:a7:c8
        inet addr:192.168.0.100  Bcast:192.168.0.255  Mask:255.255.255.0
        inet6 addr: fe80::cc8f:60ae:8c0b:547f/64  Scope:Link
        inet6 addr: fe80::fa3d:6d37:8753:68a5/64  Scope:Link
        inet6 addr: fe80::3031:472d:b09e:b99f/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:3085  errors:0  dropped:0  overruns:0  frame:0
```

5. Start putty.exe and enter the IP adr of your Ubutu VM to test the connection (if SSH is not installed the connection will fail)



You can use Putty console to create a directory for your destination backup by running mkdir command

6. From windows machine run pscp.exe tool to transfer your backup to the folder you created in the previous step (make sure you respect the case for Linux paths)

Here is an example of command to transfer SQL Server backup from windows path:

```
C:\SQLBackups\AdventureWorks2014.bak
```

To Ubuntu desktop:

```
home/mbouarroudj/Desktop/
```

```
pscp C:\SQLBackups\AdventureWorks2014.bak
```

```
mbouarroudj@192.168.0.100:/home/mbouarroudj/Desktop/AdventureWorks2014.bak
```

When prompted, enter your password for the indicated user

```
C:\Tools\Putty>pscp C:\SQL2016Backups\AdventureWorks2014.bak mbouarroudj@192.168.0.100:/home/mbouarroudj/Desktop/AdventureWorks2014.bak
mbouarroudj@192.168.0.100's password:
AdventureWorks2014.bak | 45664 kB | 11416.0 kB/s | ETA: 00:00:00 | 100%
```

7. Restore AdventureWorks2014.bak in SQL Server vNext

- On Linux machine connect to SQL Server instance with SQLCmd and run the following query:

```
select physical_name from sys.database_files
```

By default the path for .mdf and .ldf files is 'c:\var\opt\mssql\data\' , because 'C:\' path maps to the root directory (/) we will restore our database in '\var\opt\mssql\data\' path

- Create RestoreScript.sql file with this code (in our case we have created the file in the Ubuntu Desktop)

```
RESTORE DATABASE AdventureWorks2014
FROM DISK = '/home/mbouarroudj/Desktop/AdventureWorks2014.bak'
WITH
MOVE 'AdventureWorks2014_Data' TO '/var/opt/mssql/data/AdventureWorks2014_Data.mdf',
MOVE 'AdventureWorks2014_Log' TO '/var/opt/mssql/data/AdventureWorks2014_Log.ldf'
GO
```

In Ubuntu terminal console (or Putty console) run the following command

```
$ sqlcmd -H localhost -U SA -P YourSaPassword -i /home/mbouarroudj/Desktop/RestoreScript.sql
```

```
mbouarroudj@Ubuntu1604: ~  
mbouarroudj@Ubuntu1604:~$ sqlcmd -H localhost -U SA -P a4t18288- -i /home/mbouarroudj/Desktop/RestoreScript.sql  
Processed 24248 pages for database 'AdventureWorks2014', file 'AdventureWorks2014_Data' on file 1.  
Processed 4 pages for database 'AdventureWorks2014', file 'AdventureWorks2014_Log' on file 1.  
Converting database 'AdventureWorks2014' from version 782 to the current version 860.  
Database 'AdventureWorks2014' running the upgrade step from version 782 to version 801.  
Database 'AdventureWorks2014' running the upgrade step from version 801 to version 802.  
Database 'AdventureWorks2014' running the upgrade step from version 802 to version 803.  
Database 'AdventureWorks2014' running the upgrade step from version 803 to version 804.  
Database 'AdventureWorks2014' running the upgrade step from version 804 to version 805.  
Database 'AdventureWorks2014' running the upgrade step from version 805 to version 806.  
Database 'AdventureWorks2014' running the upgrade step from version 806 to version 807.  
Database 'AdventureWorks2014' running the upgrade step from version 807 to version 808.
```

Connect to AdventureWorks2014 using SqlCmd and run some test queries

```
mbouarroudj@Ubuntu1604: ~  
1> use adventureworks2014  
2> go  
Changed database context to 'AdventureWorks2014'.  
1> select top 5 * from Person.Person  
2> go  
BusinessEntityID PersonType NameStyle Title FirstName LastName  
MiddleName Suffix EmailPromotion AdditionalContactInfo  
Demographics  
rowguid ModifiedDate  
-----  
-----  
-----
```

Notes

- Database compression is supported and you can restore a compressed database without any issue

- You can restore from SQL Server 2014 and 2016
- Based on my tests, FILESTREAM is not [yet] supported in SQL Server vNext CTP 1.1, you cannot restore a database with FILESTREAM files.

Limitations of SQL Server vNext CTP 1.1 (December 2016)

At the time of writing this article here are some unsupported features and services

SQL Server Agent

SQL Server Browser

Full-text Search

Replication

Always On Availability Groups

Multiple instances, only single instance installation is supported, to have more than one instance on a given host we should use VMs.

For an exhaustive list check Microsoft releases notes:

<https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-release-notes>

Conclusion

In this part we have seen how easy to install SQL Server vNext on Linux Ubuntu a Debian-based distributions, also we have seen how to install Sql Cmd, BCP tools, and also ODBC drivers.

We were able to run basic queries against SQL Server vNext, create our own new databases and restore AdventureWorks2014 from SQL Server 2014.

In Part 2 of this article we will see how to:

- Install Visual Studio Code on Linux Ubuntu VM
- Install MSSQL extension for Visual Studio Code
- Connect to SQL Server vNext using VS code and run T-SQL codes against a new database
- Install .NET core and C# extension for VS Code, and also build a sample project in C# to connect and retrieve data from SQL Server vNext
- Connect to SQL Server vNext from windows server using SQL Server Management Studio

About me

My name is Mohamed Bouarroudj and I am SQL Server DBA working at IBM Canada and certified MCSA - SQL Server 2012/2014, I have started working with SQL Server since 6.5 version, my favourite subjects are High Availability and Performances. I am also the author of SQLDBDiff a freeware and shareware tool that compares the structure and the data of SQL Server databases and generates the sync scripts, link: www.sqldbtools.com

You can contact me on mbouarroudj@sqldbtools.com

Montreal December 2016