

Virtualization of Multi Server Environment Results in Enhanced Communication and Fortification of Transport Layer Security Protocol

Kuljeet Kaur ,Dr. G.Geetha
School of Computer Applications,
Lovely Professional University (Phagwara –[PB]. India)

Abstract— Transport layer security protocol could be fortified if virtualization is done in the Multi Server Environment of an Organization. Paper explicates the process of generating multi server environment with the help of SQL Server 2008 R2. Complete implementation process of, how one instance of SQL Server can communicate with another instance of SQL Server in the multi server environment, is explained in the paper. Steps which are involved for process of virtualization of instances with the help of Linked Servers in SQL Server are elucidated in details in the paper. Linked Servers will enhance the security at the transport layer when instances will start communicating with each other with virtualization. Conclusion is drawn that virtualization done in the multi server environment of an organization will result in enhanced communication of instances and fortification of the transport layer security protocol.

Keywords: Multi Server Environment, Virtualization, Fortification, Transport Layer, Linked Servers

I. INTRODUCTION

Whenever there is communication at the transport layer and resources are to be accessed from remote systems then generally for accessing remote system should have different memory units, like Physical Disks or Hard Disks. If same operating system and the same machine are to be used for accessing or communication then concept of virtualization could be used. Virtualization is the creation of a virtual (rather than actual) version of something, such as an operating system, a server, a storage device or network resources [1]. So paper makes use of the concept of virtualization for generating a multi server environment which would not be real but processing or accessing or communication of instances would be real.

The structure of the remainder of the paper is as follows. In Section II process of communication from one instance to second instance in SQL Server is mentioned, in Section III elucidation of process of communication from second instance to first instance is done, in Section IV linked servers used for setting connections for communication in the multi server environment are deployed, in Section V conclusion is drawn that virtualization will result in enhancement of communication and fortification of transport layer security protocol and in Section VI specifies the references.

II. PROCESS OF COMMUNICATION FROM ONE INSTANCE TO SECOND INSTANCE IN SQL SERVER

Two instances of SQL Server can communicate with each other only when there is a strong connection between them. For establishing a connection need of two SQL Server installed on same operating system is required [2]. Different or same hard drive depending upon need of the organization could be used for installation. Below mentioned are the steps along with diagrams of LIVE implementation for establishing a connection from one instance in SQL Server to second instance in SQL Server:

1. Establish a connection with the Server after opening First SQL SERVER Instance [3].

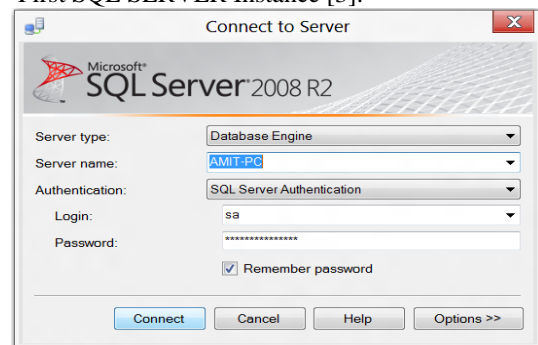


Figure 1: Connection of First Instance of SQL Server to Server

2. Establish a connection with the Server after opening Second SQL SERVER Instance.

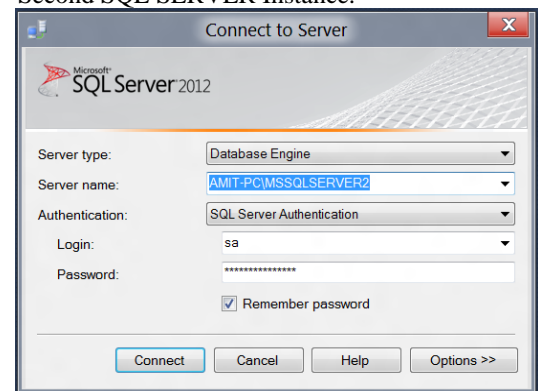


Figure 2: Connection of Second Instance of SQL Server to Server

3. Connect First Instance of SQL Server with Second Instance of SQL Server with the following steps [5]:
 - Click Connect in Object Explorer
 - Server Name:- Second Instance Server Name(Example-AMIT-PC/MSSQLSERVER2)
 - Authentication:- SQL SERVER Authentication
 - Login:- Second Server Instance Login Name(Example:-sa)
 - Password:- Second Server Instance Password(Example:-gaurav)
 - Now click on Connect.



Figure 3: Connection of First Instance of SQL Server to Second Instance of SQL Server

4. Check either connection is established between the Instances or not [4]. (Example:- AMIT-PC/MSSQLSERVER2) in First Instance of SQL SERVER(Ex:- AMIT-PC)
- 5.

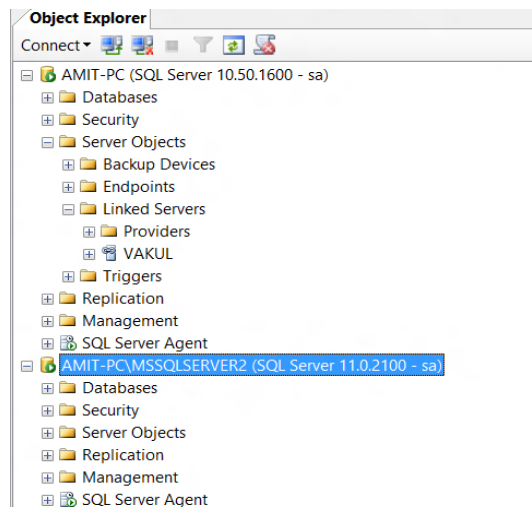


Figure 4: Check either the connection is established

6. Create Linked Server in First Instance to Access the Databases of Second Instance
 - i) In object explorer of First Instance->Open Server Objects->Right Click On Linked Server->Select New Linked Server.

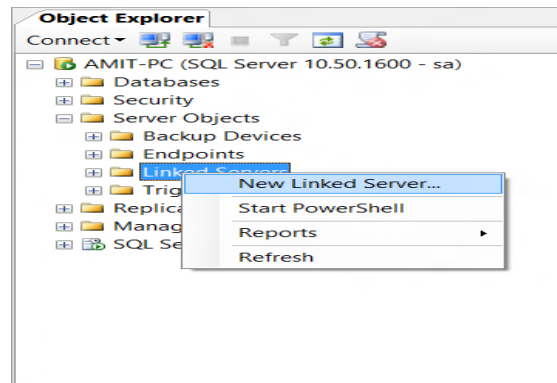


Figure 5: Create Linked Server in First Instance of SQL Server

- ii) Following Window would appear when click on New Linked Server is done:

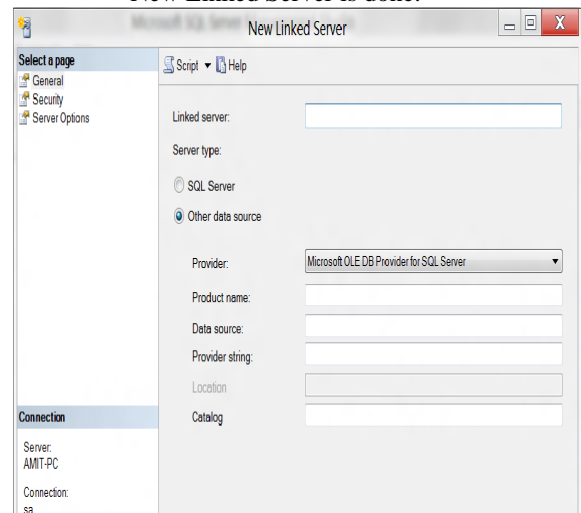


Figure 6: Link of New Linked Servers

- iii) In the General properties following specifications to be written [6]:
 - Linked Server Name:-Any name(VAKUL)
 - Provider:- Select Microsoft OLEDB Provider for SQL SERVE
 - Product Name:- Any name (But Different From Linked Server Name) (Example:-vakul1)
 - Data Source:- Second Instance SERVER Name (Example-AMIT-PC\MSSQLSERVER2)
 - Provider String:- source=second instance SERVER name(Example-AMIT-PC\MSSQLSERVER2);database=any database of Second Instance(Example-amrit); userid=Login name of Second Instance(Example-sa);password=second instance Server Password(Example-gaurav)

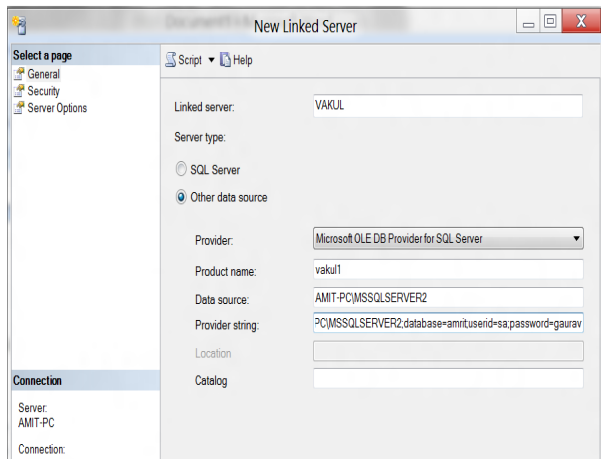


Figure 7: General Properties of New Linked Servers

iv) Select the next property of Security.

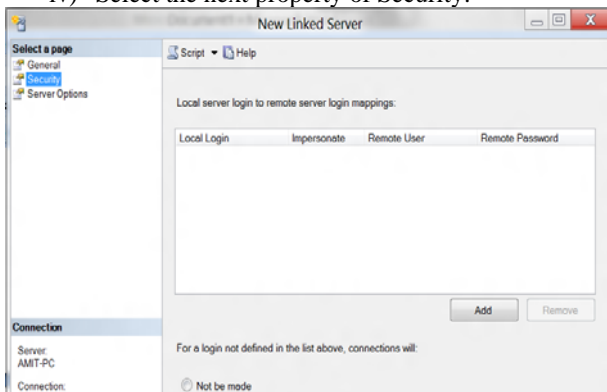


Figure 8: Security Property of New Linked Servers

v) After selecting Security option from the left-> Now click on ADD button

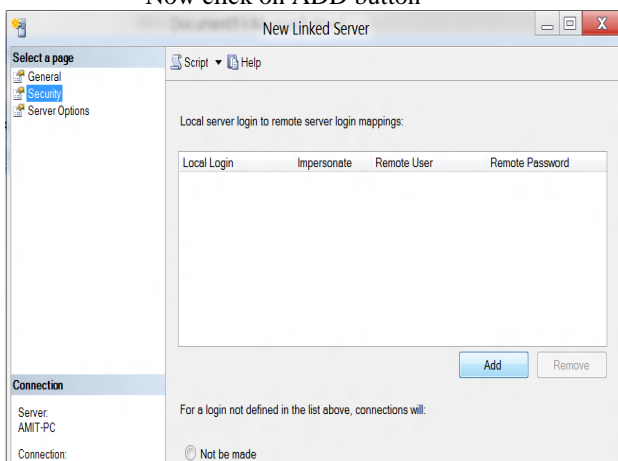


Figure 9: Adding Details to the Security Property of New Linked Servers

vi) After clicked on ADD button you have to insert the

- Local login:- First instance login name(example-sa)

- Remote User:- Second Instance Login Name(Example-sa)
- Remote Password:- Second Instance Server Password (Example-gaurav)
- Impersonate:- Do not Check the Impersonate checkbox.

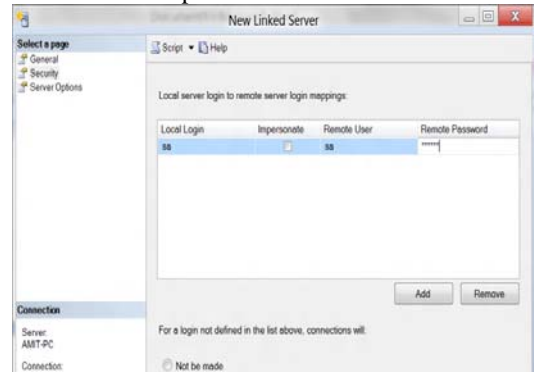


Figure 10: Adding Login Details to the New Linked Servers

vii) Now select option Server Option from the left of the Current Window

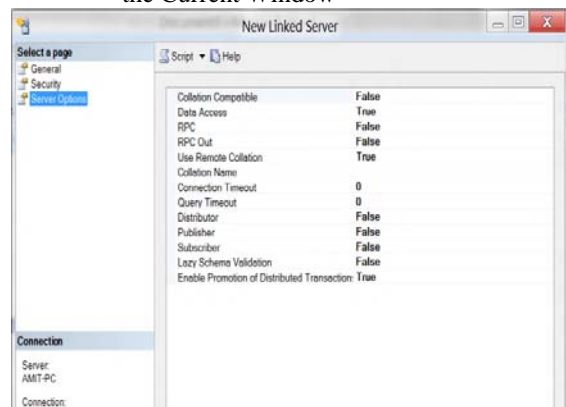


Figure 11: Server Properties of New Linked Servers

viii) After selecting Server Option two things to do:-

- RPC:- True
- RPC OUT:- True

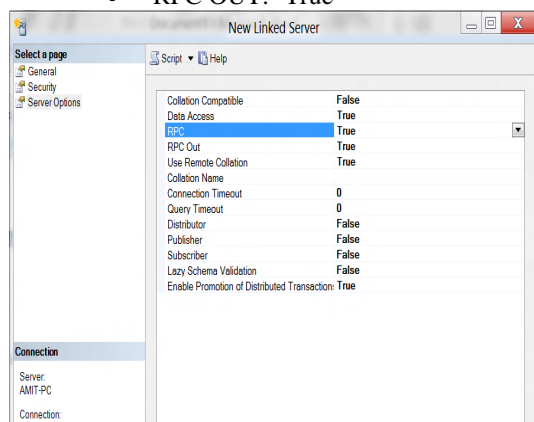


Figure 12: Server properties of New Linked Servers

7. Press ok to save all the details.
8. Linked Server has been created in First Instance of SQL SERVER. Do check in Server Objects of First SQL SERVER Instance->Open Linked Server there you find your Linked Server name(Example:- VAKUL)

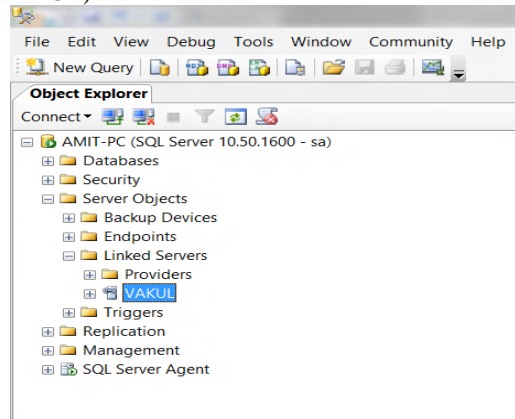


Figure 13: Check the availability of Second Instance through Linked Servers in First Instance

9. Execution of queries could be done in First Instance of SQL SERVER:-
 - Select * from linked-server-name.database-name(Second Instance Database).dbo.table-name(Second Instance Table name belongs to that Database which we given before in this query)
 - Select * from VAKUL.amrit.dbo.emp

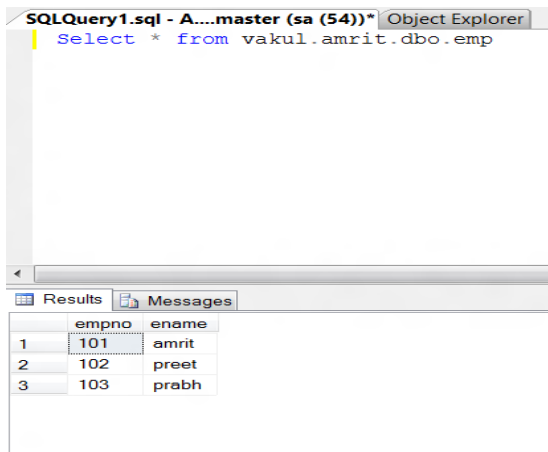


Figure 14: Execution of Query

10. Create Table in Second Instance of SQL SERVER(E.g.:- AMIT-PC\MSSQLSERVER2) using Linked Server VAKUL
 EXEC linked-server-name.database.dbo.sp_executesql N'
 Create table dbo.tablename (new table)(Statements:
)'
 EXEC VAKUL.amrit.dbo.sp_executesql N'
 Create table dbo.stu
 (Id numeric (10), name varchar (10))'

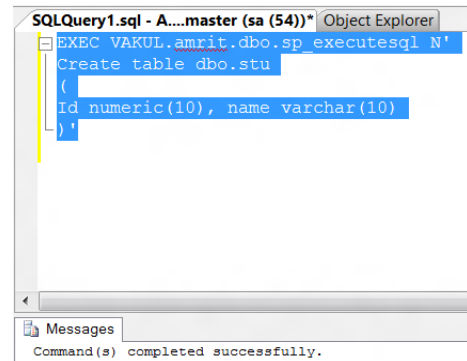


Figure 15: Creation of Table in Second Instance

11. Open Second Instance of SQL Server to Check that STU Table is Created or Not in Second Instance Of SQL SERVER

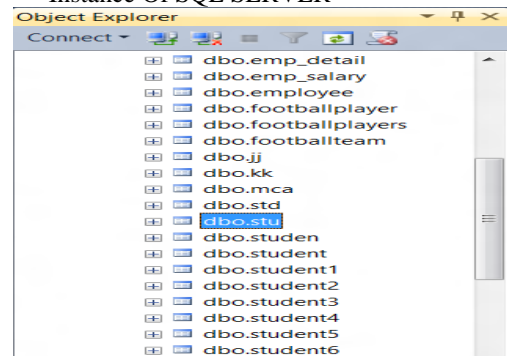


Figure 16: Check Table in the Second Instance of SQL Server

III. PROCESS OF COMMUNICATION FROM SECOND INSTANCE TO FIRST INSTANCE IN SQL SERVER

The process to be followed for connecting second instance to first instance is mentioned below. After two instances are connected then communication becomes quite easy. Below mentioned are the steps along with diagrams of LIVE implementation for establishing a connection from Second instance in SQL Server to First instance in SQL Server:

1. Connect second SQL Server Instance with SERVER.

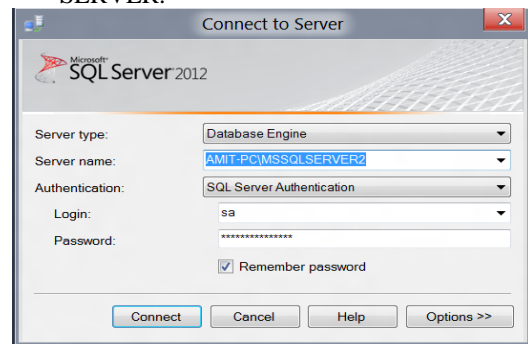


Figure 17: Establishing Connection of Second Instance of SQL Server

2. Open First Instance of SQL SERVER and Connect that with SERVER

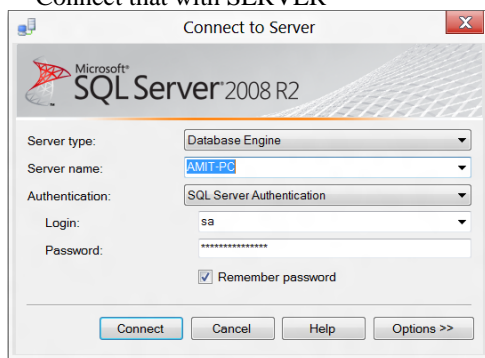


Figure 18: Connection of First Instance with Server

3. Connect Second Instance of SQL SERVER with First Instance of SQL SERVER [7]
 - Click on connect in Object Explorer of Second Instance.
 - Server Name:- First Instance Server Name(Example-AMIT-PC)
 - Authentication:- SQL SERVER Authentication
 - Login:- First Server Instance Login Name(Example:-sa)
 - Password:- First Server Instance Password(Example:-hemant)
 - Now click on Connect.

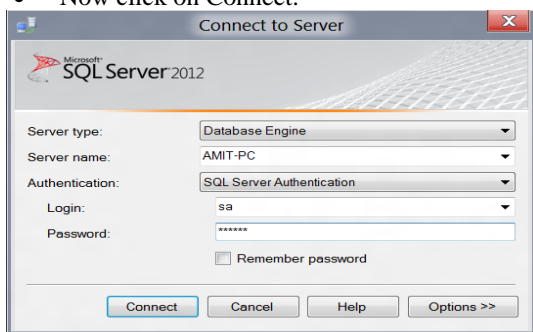


Figure 19: Connection between both Instances

4. Check for the connection with First Server Instance (Example:- AMIT-PC) in Second Instance of SQL SERVER(Example:- AMIT-PC\MSSQLSERVER2)

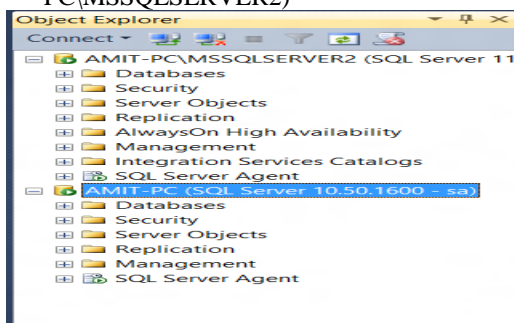


Figure 20: Checking for the Connection between both Instances

5. Create Linked Server in Second Instance to Access the Databases of First Instance

- i) In object explorer of Second Instance of SQL SERVER->Open Server Objects->Right Click on Linked Server->Select New Linked Server.

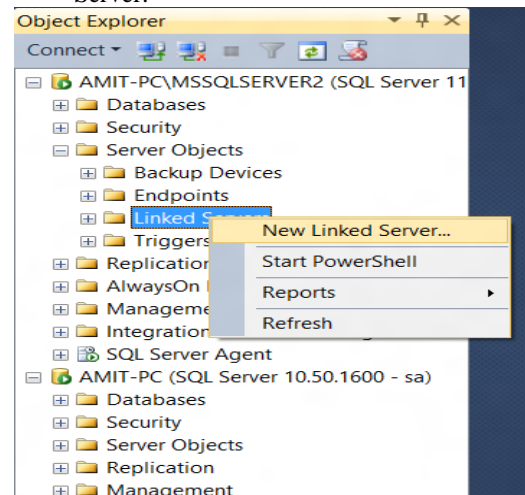


Figure 21: Creating New Linked Server

- ii) General Properties of New Linked Server will appear on the screen:-

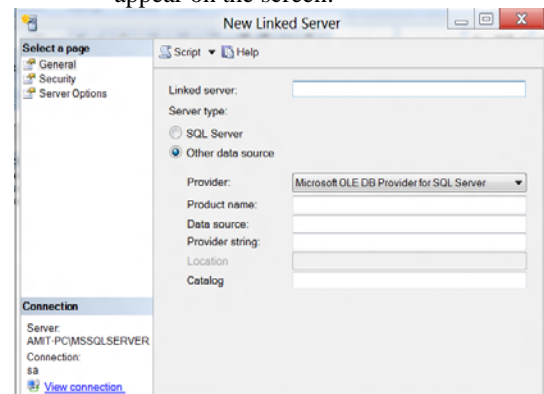


Figure 22: Checking for the Connection between both Instances

- iii) Here By-default general is selected [7]:-
 - Linked Server Name:-Any name(GAURAV)
 - Provider:- Select Microsoft OLEDB Provider for SQL SERVER
 - Product Name:- Any name (But Different From Linked Server Name) (Example-gaurav1)
 - Data Source:- First Instance SERVER Name (Example-AMIT-PC)
 - Provider String:- source=First instance SERVER name(Example-AMIT-PC);database=any database of First Instance(Example-gaurav);userid=Login name of First Instance(Example-sa);password=First instance Server Password(Eg-hemant)

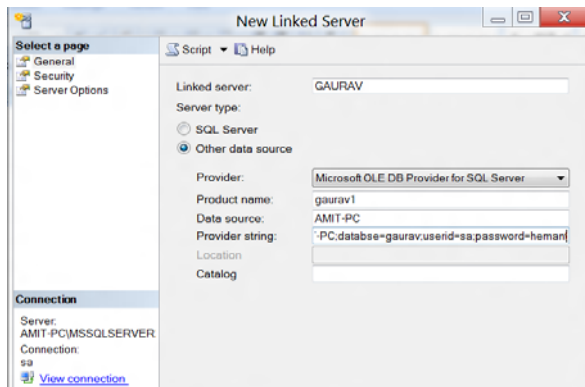


Figure 23: General properties definition

iv) Select the Security Option.

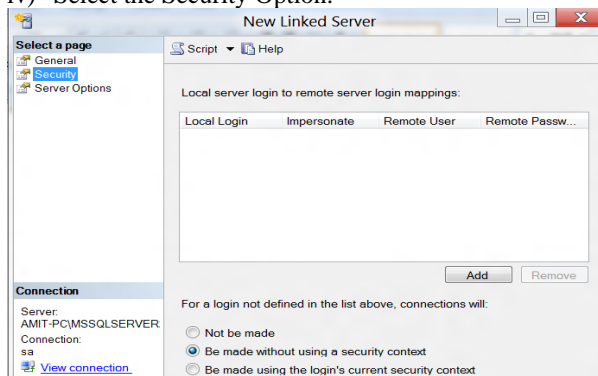


Figure 24: Security Properties of New Linked Server

v) Add Security Options->Click on ADD button

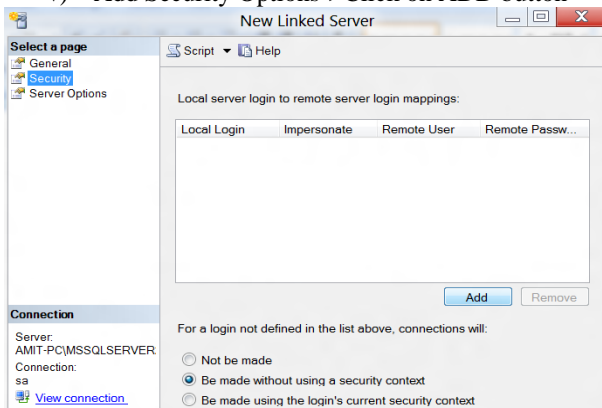


Figure 25: Click Add button in security option

vi) After clicked on ADD button you have to insert the

- Local login:- Second instance login name(example-sa)
- Remote User:- First Instance Login Name(Example-sa)
- Remote Password:- First Instance Server Password (Eg-hemant)
- Impersonate:- Do not check impersonate checkbox.

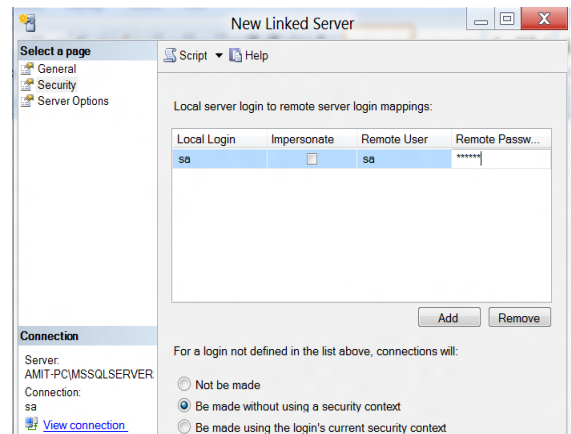


Figure 26: Add Login Details of New Linked Server

vii) Select Server Options from the left of the Current Window

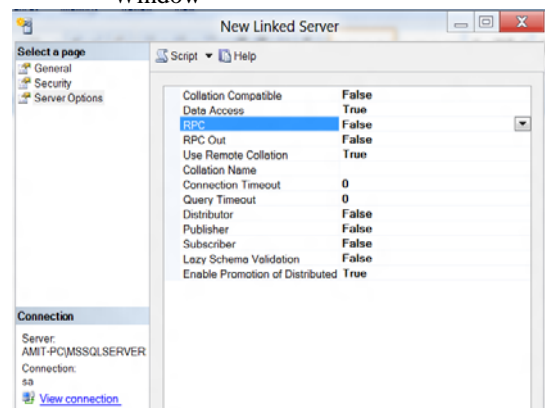


Figure 27: Select Server Options of New Linked Server

viii) After selecting Server Option two things to do:-

- RPC:- SET True (By-Default False)
- RPC OUT:- SET True (By-Default False)
-

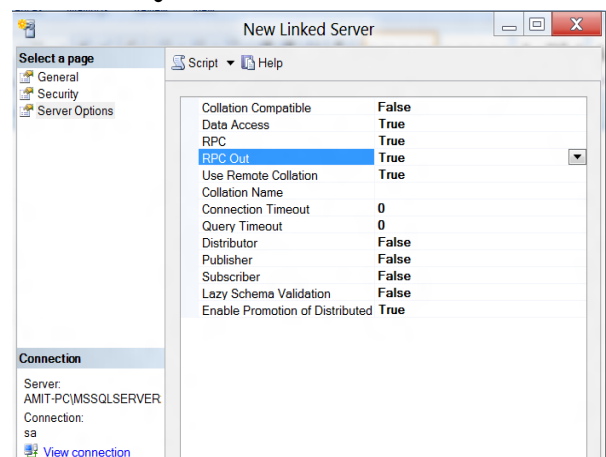


Figure 28: Select Server Options of New Linked Server of RPC and RPC OUT

6. After clicking OK Linked Server is added.
7. Linked Server has been created in Second Instance of SQL SERVER. Do check in Second Instance Server Objects->Open Linked Server there you find your Linked Server name(Ex:-GAURAV)

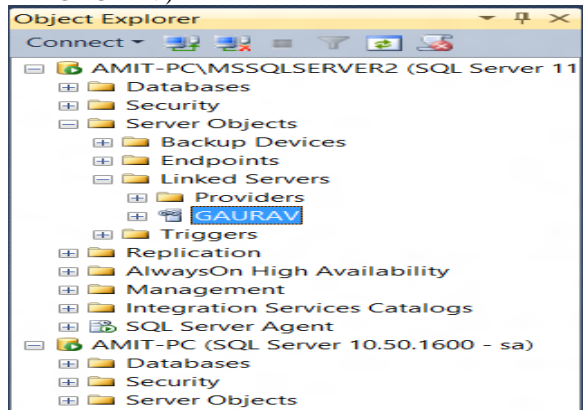


Figure 29: Checking the existence of New Linked Server

8. Start Query in Second Instance of SQL Server To Access the Database and Table of First Instance of SQL SERVER:-

- Select * from linked-server-name.database-name(First Instance Database name).dbo.table-name(First Instance table name which belongs to that database which is given before in this query)
- Select * from GAURAV.gaurav.dbo.dep

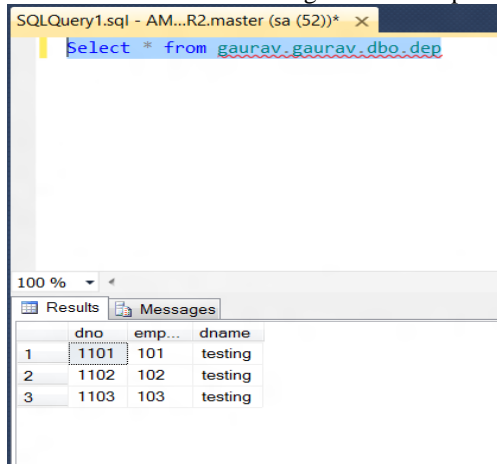


Figure 30: Execution of Query

9. Create Table in First Instance of SQL SERVER(Example:- AMIT-PC) using Linked Server GAURAV

```
EXEC linked-server-name.database.dbo.sp_executesql N'
Create table dbo.tablename(new table)(Statements:)'
EXEC GAURAV.gaurav.dbo.sp_executesql N'
Create table dbo.dept(
Id numeric(10), name varchar(10))'
```

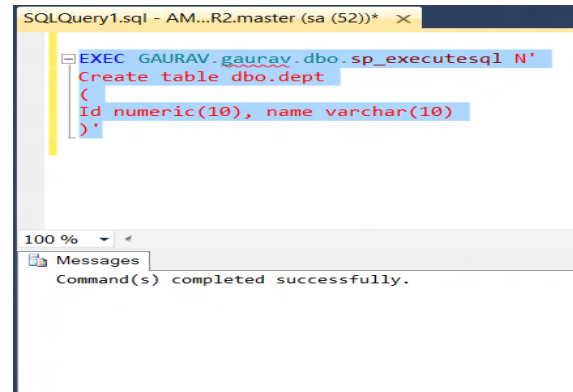


Figure 31: Creation of Table in the Second Instance of SQL Server

10. Open First Instance of SQL Server (Example:- AMIT-PC) to Check that DEPT Table is Created or Not in First Instance of SQL SERVER

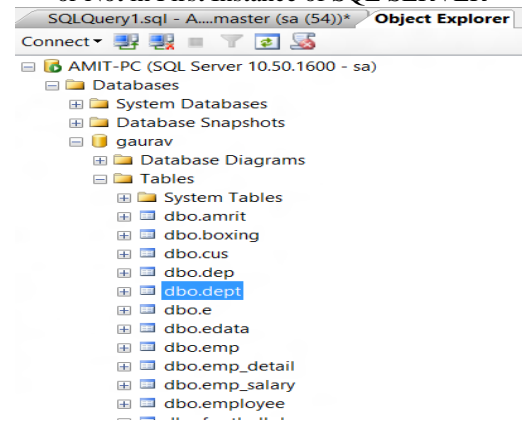


Figure 32: Checking of existence of Instances

IV. LINKED SERVERS USED FOR SETTING CONNECTIONS FOR COMMUNICATION IN THE MULTI SERVER ENVIRONMENT

Configuration of linked server is required to enable the SQL Server Database Engine to execute commands. This execution would be against OLE DB data sources outside of the instance of SQL Server.

Typically linked servers are configured to enable the Database Engine to execute a Transact-SQL statement that includes tables in another instance of SQL Server, or another database product such as Oracle [8]. Many types OLE DB data sources can be configured as linked servers, including Microsoft Access and Excel.

Linked servers offer the following advantages [9]:

- The ability to access data from outside of SQL Server.
- The ability to issue distributed queries, updates, commands, and transactions on heterogeneous data sources across the enterprise.
- The ability to address diverse data sources similarly.

It is required in the multi server environment. Two servers would be created through virtualization as mentioned in Section II and III. Two instances would communicate with each other easily and this communication would be quiet secure as the concept of Linked Servers is used. Secure communication would enhance the complete transport layer.

V. CONCLUSION OF PAPER THAT VIRTUALIZATION RESULTS IN ENHANCED COMMUNICATION AND FORTIFICATION OF TRANSPORT LAYER SESURITY PROTOCOL

Physically when two servers would be used it would result in

- Increase in the cost
- Space consuming
- Compatibility issue need to be resolved
- Resources required would be more

But with the same memory and virtualization all these issues could be resolved. Virtualization gives virtual environment to the servers and for that two instances of SQL Server are installed in the same system [10]. Same hard disk but different or same hard drives could be used, No compatibility issue would be there because same operating system is being used. Following benefits would be there when virtualization would be used in the multi server environment in an organization:

- No compatibility issue as same operating system is being used by different instances.
- If different operating systems in the same machine are used then also memory location is same.
- Data accessing would be secure.
- Communication between the two instances would be easy with the help of Linked Servers.
- As communication is possible only on the transport layer so it is being enhanced with the

implementation of virtualization in the multi server environment of an organization [11]. So overall when virtualization is implemented in the multi server environment of an organization then transport layer security protocol is fortified.

ACKNOWLEDGEMENT

We want to personally thank one of the students of MCA (Honors) Gaurav Kumar Arora for his special contribution by providing a physical environment at his location for successful implementation of Multi Server Environment.

VI. REFERENCES

- [1] "Virtualization in SQL Server 2008", www.microsoft.com/en-us/server-cloud/...server/hyper-v.aspx
- [2] "Setting Connections in SQL Server 2008", www.connectionstrings.com/sql-server-2008
- [3] "How to connect SQL Server Instances using SQL Server Management Studio", kb.discountasp.net/.../how-to-connect-to-sql-server-2008-using-sql-s...
- [4] "Active Connections in SQL Server 2008", www.jasonholden.com/blog/.../SQL-Server-2008-Active-Connection
- [5] "Linked Server Components", <http://msdn.microsoft.com/en-us/library/ms188279.aspx>
- [6] "Configuration of Linked Servers", <http://msdn.microsoft.com/en-us/library/ms188279.aspx>
- [7] "Adding SQL Server Instances", stackoverflow.com/.../how-do-i-add-a-sql-server-2008-service-istan...
- [8] "Windows Server Virtualization", www.windowsnetworking.com/.../Understanding-Windows-Server-2...
- [9] "Linked Servers in SQL Servers in 2008", www.execsql.com/post/linked-servers-in-sql-server-2005
- [10] "Multiple Instances in SQL Server", itknowledgeexchange.techtarget.com/sql-server//single-instance-vs-m...
- [11] Kuljeet Kaur and G Geetha. Article: Fortification of Transport Layer Security Protocol by using Password and Fingerprint as Identity Authentication Parameters. International Journal of Computer Applications 42(6):36-42, March 2012. Published by FCS,NY,USA. (<http://www.ijcaonline.org/archives/volume42/number6/5700-7751>)