**Workshop on MS SQL Server On Linux Day 3**

**Environment:**

Domain Name: abc.com

Domain server FQDN:Dc.abc.com  
Domain Ip: 10.0.0.50

**Today Agenda:**

**Create certificates for node21 and node22**

**Create endpoints and configure always on.**

**Install pace maker and add ag into pace maker**

**Manually failover to another Node.**

**Step By step:**

1.Disable SELINUX and Firewall

vi /etc/sysconfig/selinux

Change SELINUX=enforcing to Permissive and save the file on both nodes and

setenforce 0;systemctl stop firewalld;systemctl disable firewalld

2.Install Pacemaker software on both nodes

yum install pacemaker pcs fence-agents-all -y

3.Note: Once pacemaker was installed, you will see user created named **hacluster** - we need reset the password for this user on both nodes

To reset the password for hacluster

passwd hacluster

4. Start and Enable PaceMaker Services on both nodes

systemctl start pcsd;systemctl enable pcsd

5. Add membership of the cluster using hacluster username

pcs cluster auth node21 node22 -u hacluster

6. create Cluster named "LinuxSQLCluster" in my case

pcs cluster setup --name LinuxSQLCluster node21 node22

7.view Configuration of the Cluster on both nodes

cat /etc/corosync/corosync.conf

8. start the cluster on any one node.

pcs cluster start --all

9. check the status of the cluster and enable the cluster

pcs status

pcs cluster enable --all

10.Note: To disable the stonith device configuration in cluster

pcs property set stonith-enabled=false

11.Install SQL Server High Availability Package on both nodes

yum install mssql-server-ha

12.Enable Always on Availability Groups and restart SQL Server on both nodes

/opt/mssql/bin/mssql-conf set hadr.hadrenabled 1;systemctl restart mssql-server

13.Open SSMS and create Certificate for each node

# Node Name : node21

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Pass@123';

GO

CREATE CERTIFICATE node21\_Cert

WITH SUBJECT = 'Node21 AG Certificate';

GO

BACKUP CERTIFICATE node21\_Cert

TO FILE = '/var/opt/mssql/data/node21\_Cert.cer';

GO

CREATE ENDPOINT SQLAG

STATE = STARTED

AS TCP (

LISTENER\_PORT = 5022,

LISTENER\_IP = ALL)

FOR DATABASE\_MIRRORING (

AUTHENTICATION = CERTIFICATE node21\_Cert,

ROLE = ALL);

GO

14.Open SSMS and create Certificate for each node

# Node Name : node22

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Pass@123';

GO

CREATE CERTIFICATE node22\_Cert

WITH SUBJECT = 'node22 AG Certificate';

GO

BACKUP CERTIFICATE node22\_Cert

TO FILE = '/var/opt/mssql/data/node22\_Cert.cer';

GO

CREATE ENDPOINT SQLAG

STATE = STARTED

AS TCP (

LISTENER\_PORT = 5022,

LISTENER\_IP = ALL)

FOR DATABASE\_MIRRORING (

AUTHENTICATION = CERTIFICATE node22\_Cert,

ROLE = ALL);

GO

15.Copy Certificate of one node to other using SCP

# on Node21

scp -r /var/opt/mssql/data/node21\_Cert.cer root@node22:/var/opt/mssql/data/node21\_Cert.cer

# On Node 22

scp -r /var/opt/mssql/data/node22\_Cert.cer root@node21:/var/opt/mssql/data/node22\_Cert.cer

16.Change Ownership of certificate to mssql on each node

cd /var/opt/mssql/data/;chown mssql:mssql node21\_Cert.cer;chown mssql:mssql node22\_Cert.cer;ls -ltr

17.

Create instance Level SQL User (AGUser in my case on each node) using SSMS and Open SSMS and create User

USE [master]

GO

CREATE LOGIN [AGUser] WITH PASSWORD=N'123', DEFAULT\_DATABASE=[master], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF

GO

ALTER SERVER ROLE [sysadmin] ADD MEMBER [AGUser]

GO

USE [master]

GO

CREATE USER [AGUser] FOR LOGIN [AGUser]

GO

USE [master]

GO

ALTER ROLE [db\_owner] ADD MEMBER [AGUser]

GO

18.Restore certificate of Other Nodes into the present node using SSMS below: Login to Node21

CREATE CERTIFICATE node22\_Cert

AUTHORIZATION AGUser

FROM FILE = '/var/opt/mssql/data/node22\_Cert.cer';

GO

--Grant permission to connect to the endpoint of node21

GRANT CONNECT ON ENDPOINT::SQLAG TO AGUser;

--Login to Node22

CREATE CERTIFICATE node21\_Cert

AUTHORIZATION AGUser

FROM FILE = '/var/opt/mssql/data/node21\_Cert.cer';

Go

--Grant permission to connec to the endpoint of node22

GRANT CONNECT ON ENDPOINT::SQLAG TO AGUser;

19. Copy the back up from primary node to secondary node

scp -r /var/opt/mssql/data/test.bak root@node22:/var/opt/mssql/data/

chown mssql:mssql /var/opt/mssql/data/test.bak; ls -ltr

20. Create Availability Group using SSMS with Cluster type External

21.Create a new login or use the same login to give Pacemaker permission and provide view server permission, I will give

sysadmin to this user just for this demo

# On all Nodes Edit vi /var/opt/mssql/secrets/passwd

echo 'AGUser' >> ~/testpasswd;echo '123'>> ~/testpasswd

mv ~/testpasswd /var/opt/mssql/secrets/passwd;chmod 400 /var/opt/mssql/secrets/passwd

cat /var/opt/mssql/secrets/passwd

22.Create the AG resource in the Pacemaker cluster

pcs resource create LinuxRG ocf:mssql:ag ag\_name=linuxag meta failure-timeout=30s --master meta notify=true

23.Create IP resource for Listener

sudo pcs resource create ListenerAg ocf:heartbeat:IPaddr2 ip=10.0.0.23 cidr\_netmask=8

24.Create an ordering constraint to ensure that the AG resource is up and running before the IP address. While the colocation

constraint implies an ordering constraint, this enforces it

sudo pcs constraint order promote LinuxRG-master then start ListenerAg

# Let's Test Failover

================================

25.Failover from running node to another node.

Check the resources running on which node by using below command

pcs status

pcs resource move LinuxRG-master node21 --master

# Check Constraints

pcs constraint list --full

26.Removing to location constraint to make successfully automatic failover

pcs constraint remove cli-prefer-LinuxRG-master