CLUSTER OPERATION MANUAL

HAP

(ASCS & ERS)

Version 1.0

# pacemaker CLuster setup (AP) HAP

Contents

[pacemaker CLuster setup (AP) HAP 1](#_Toc92215426)

[1 PURPOSE 2](#_Toc92215427)

[2 SCOPE 2](#_Toc92215428)

[3 Prerequisites 2](#_Toc92215429)

[4 INSTANCE CALCULATION 3](#_Toc92215430)

[5 HA Details 4](#_Toc92215431)

[6 HA Overiew 5](#_Toc92215432)

[7 Load Balancer setup 6](#_Toc92215433)

[7.1 Load Balancer Configuration 6](#_Toc92215434)

[7.2 REFERENCE DOCUMENTS / LINKS 12](#_Toc92215435)

[8 Install & Configure SuSE Pacemaker cluster 13](#_Toc92215436)

[8.1 CRM Configuration 21](#_Toc92215437)

[8.2 REFERENCE DOCUMENTS / LINKS 24](#_Toc92215438)

[9 Cluster patching process 25](#_Toc92215439)

[10 HA TEsting 26](#_Toc92215440)

[11 DOCUMENT REVISION HISTORY 26](#_Toc92215441)

# PURPOSE

The purpose of this document is to configure High Availability on servers using Azure NetApp Files with client technical solution arena for the BHF Project.

# SCOPE

This section brings the scope to configure HA on SUSE Linux servers.

# Prerequisites

Below are the prerequisites for SUSE high availability configuration:

* Make Sure Full server back up in place for both the nodes.
* SAP NetWeaver requires shared storage for the transport and profile directory. In our case, we will deploy Azure Net app Files resources.
* The cluster must include a valid STONITH method. In our case STONITH device using Azure fence agent, we need to have the detail of the fencing agent configuration.
* Name resolution of the cluster nodes and the virtual IP address must resolve on all cluster nodes.

# INSTANCE CALCULATION

|  |  |
| --- | --- |
| NN | ASCS Instance Number |
| EE | ERS Instance Number |

* If using Standard Load Balancer, select **HA ports**
* If using Basic Load Balancer, create Load balancing rules for the following ports

For ASCS (NN-01) and ERS (EE-10):

|  |  |
| --- | --- |
| **Instance** | System |
| 32NN | 3201 |
| 36NN | 3601 |
| 39NN | 3901 |
| 81NN | 8101 |
| 5NN13 | 50113 |
| 5NN14 | 50114 |
| 5NN16 | 50116 |
| 32EE | 3210 |
| 33EE | 3310 |
| 5EE13 | 51013 |
| 5EE14 | 51014 |
| 5EE16 | 51016 |
| 620NN, 621EE (Probe) | 62001, 62110 |

# HA Details

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Component** | | | | | | **Hostname Name** | | | | **IP Address** | | | | **SID** | | | | | **Comments** | | | | |
| S/4HANA On-Premise Edition (ASCS ) | | | | | | hapcs01use2pr | | | | 10.213.34.39 | | | | HAP | | | | | VM - Physical IP | | | | |
| S/4HANA On-Premise Edition (ERS ) | | | | | | hapcs02use2pr | | | | 10.213.34.41 | | | | HAP | | | | | VM - Physical IP | | | | |
| S/4HANA ASCS VIP 1 - ASCS | | | | | | saphapcs01pr | | | | 10.213.34.40 | | | | HAP | | | | | Load Balancer Front End IP | | | | |
| S/4HANA ASCS VIP 2 - ERS | | | | | | saphapcs02pr | | | | 10.213.34.42 | | | | HAP | | | | | Load Balancer Front End IP | | | | |
| S/4HANA APP VIP 3 - PAS | | | | | | saphapap01pr | | | | 10.213.34.88 | | | | HAP | | | | | Virtual IP 1 | | | | |
| S/4HANA APP VIP 4 - AAS | | | | | | saphapap02pr | | | | 10.213.34.89 | | | | HAP | | | | | Virtual IP 2 | | | | |
| **Load balancer** | | | |  | | | |  | | | | |  | | | |  | | | | |  | |
| **Name** | | | | **SKU** | | | | **Frontend IP** | | | | | **Backend pool** | | | | **Health probe** | | | | | **Load balancing rule** | |
| sap-ilb-hapapp-use2-pr-01 | | | | Standard | | | | HAP\_ASCS\_FrontendIP01 | | | | | HAP\_app\_Backendpool01 | | | | HAP\_ASCS\_Probe01 | | | | | HAP\_ASCS\_Rule01 | |
| HAP\_ERS\_FronendIP01 | | | | | HAP\_ERS\_Probe01 | | | | | HAP\_ERS\_Rule01 | |
| **Backend pool** | | | | | | |  | | | |  | | | | |  | | | |  | | | |
| **Name** | | | | | | | **IP version** | | | | **VM** | | | | | **NIC** | | | | **IP address** | | | |
| HAP\_app\_Backendpool01 | | | | | | | IPv4 | | | | hapcs01use2pr | | | | | hapcs01use2pr\_nic01 | | | | 10.213.34.39 | | | |
| hapcs02use2pr | | | | | hapcs02use2pr\_nic01 | | | | 10.213.34.41 | | | |
| **Health probe** | | | | | | |  | | | |  | | | | |  | | | |  | | | |
| **Name** | | | | | | | **Protocol** | | | | **Port** | | | | | **Interval** | | | | **Unhealthy threshold** | | | |
| HAP\_ASCS\_Probe01 | | | | | | | TCP | | | | 62001 | | | | | 5 | | | | 2 | | | |
| HAP\_ERS\_Probe01 | | | | | | | TCP | | | | 62110 | | | | | 5 | | | | 2 | | | |
| **Load balancing rule** |  |  |  | |  | | | |  | | |  | | |  | | |  | | |  | |  | |
| **Name** | **IP version** | **Frontend IP** | **Protocol** | | **HA Port** | | | | **Backend Port** | | | **Backend pool** | | | **Health probe** | | | **Session persistence** | | | **Idle timeout** | | **Floating IP** | |
| HAP\_ASCS\_Rule01 | IPv4 | HAP\_ASCS\_FrontendIP01 | TCP | | Selected | | | | 62001 | | | HAP\_app\_Backendpool01 | | | HAP\_ASCS\_Probe01 | | | Client IP and Protocol | | | 30 | | Enable | |
| HAP\_ERS\_Rule01 | IPv4 | HAP\_ERS\_FronendIP01 | TCP | | Selected | | | | 62110 | | | HAP\_app\_Backendpool01 | | | HAP\_ERS\_Probe01 | | | Client IP and Protocol | | | 30 | | Enable | |

# HA Overiew



# Load Balancer setup

## Load Balancer Configuration

The below tasks will be followed to configure Load Balancer for HAP.

| # | Step | Notes |
| --- | --- | --- |
| 1 | Ensure Contributor Privileges | Contributor Privileges on Azure subscription is required to execute tasks. |
| 2 | Load Balancer Configuration | 1.Create load balancer (internal, standard)     1. Create the frontend IP addresses for ASCS      1. Create the frontend IP addresses for ERS |
| 3 | Create the backend pool | 1. Create the backend pool 2. Open the load balancer, select backend pools, and click Add 3. Enter the name of the new backend pool      1. Click Add |
| 4 | Create the health probes | 1. Create the health probes    1. Port 62001 for ASCS       1. Open the load balancer, select health probes, and click Add       2. Enter the name of the new health probe       3. Select TCP as protocol, port 62001, keep Interval 5 and Unhealthy threshold 2       4. Click OK      * 1. Port 62110 for ERS |
| 5 | Load-balancing rules | 1. Load-balancing rules    1. Create a rule for the ASCS       1. Open the load balancer, select Load-balancing rules and click Add       2. Enter the name of the new load balancer rule       3. Select the frontend IP address for ASCS, backend pool, and health probe you created earlier       4. Select HA ports       5. Make sure to enable Floating IP       6. Click OK      * Repeat the steps above to create load balancing rules for ERS |

## REFERENCE DOCUMENTS / LINKS

Below are the links for documents mentioned in this document

|  |  |
| --- | --- |
| **Document Name** | **Microsoft Azure Documentation Link** |
| Deploy Azure Load Balancer manually via Azure portal | <https://docs.microsoft.com/en-us/azure/virtual-machines/workloads/sap/high-availability-guide-suse-netapp-files#deploy-azure-load-balancer-manually-via-azure-portal> |

# Install & Configure SuSE Pacemaker cluster

The below tasks will be followed to install & configure SuSE Pacemaker cluster.

| # | Step | Notes |
| --- | --- | --- |
| 1 | Ensure Admin Privileges | Admin Privileges are required to execute tasks. |
| 2 | Cluster Installation | 1. Update SLES on node1, node2   sudo zypper update     1. Install component, needed for cluster resources on node1, node2   sudo zypper in socat     1. Install azure-lb component, needed for cluster resources on node1, node2    sudo zypper in resource-agents    Note Check the version of package resource-agents and make sure the minimum version requirements are met:   * For SLES 12 SP4/SP5, the version must be at least resource-agents-4.3.018.a7fb5035-3.30.1. * For SLES 15/15 SP1, the version must be at least resource-agents-4.3.0184.6ee15eb2-4.13.1.      1. Configure the operating system on node1, node2   In some cases, Pacemaker creates many processes and thereby exhausts the allowed number of processes. In such a case, a heartbeat between the cluster nodes might fail and lead to failover of your resources. We recommend increasing the maximum allowed processes by setting the following parameter.    # Edit the configuration file  sudo vi /etc/systemd/system.conf    # Change the DefaultTasksMax  #DefaultTasksMax=512  DefaultTasksMax=4096    #and to activate this setting  sudo systemctl daemon-reload    # test if the change was successful  sudo systemctl --no-pager show | grep DefaultTasksMax    Reduce the size of the dirty cache. For more information, see [Low write performance on SLES 11/12 servers with large RAM](https://www.suse.com/support/kb/doc/?id=7010287).    sudo vi /etc/sysctl.conf    # Change/set the following settings  vm.dirty\_bytes = 629145600  vm.dirty\_background\_bytes = 314572800     1. Configure cloud-netconfig-azure for HA Cluster on node1, node2     Note Check the installed version of package **cloud-netconfig-azure** by running **zypper info cloud-netconfig-azure**. If the version in your environment is 1.3 or higher, it is no longer necessary to suppress the management of network interfaces by the cloud network plugin. If the version is lower than 1.3, we suggest to update package **cloud-netconfig-azure** to the latest available version.    Change the configuration file for the network interface as shown below to prevent the cloud network plugin from removing the virtual IP address (Pacemaker must control the VIP assignment). For more information, see [SUSE KB 7023633](https://www.suse.com/support/kb/doc/?id=7023633).    # Edit the configuration file  sudo vi /etc/sysconfig/network/ifcfg-eth0    # Change CLOUD\_NETCONFIG\_MANAGE  # CLOUD\_NETCONFIG\_MANAGE="yes"  CLOUD\_NETCONFIG\_MANAGE="no"     1. Enable ssh access on node1     sudo ssh-keygen    # Enter file in which to save the key (/root/.ssh/id\_rsa): -> Press ENTER  # Enter passphrase (empty for no passphrase): -> Press ENTER  # Enter same passphrase again: -> Press ENTER    # insert the public key you copied in the last step into the authorized keys file on the second server  sudo vi /root/.ssh/authorized\_keys    # copy the public key  sudo cat /root/.ssh/id\_rsa.pub     1. Enable ssh access on node2     sudo ssh-keygen    # Enter file in which to save the key (/root/.ssh/id\_rsa): -> Press ENTER  # Enter passphrase (empty for no passphrase): -> Press ENTER  # Enter same passphrase again: -> Press ENTER    # insert the public key you copied in the last step into the authorized keys file on the second server  sudo vi /root/.ssh/authorized\_keys    # copy the public key  sudo cat /root/.ssh/id\_rsa.pub     1. Enable ssh access on node1   # insert the public key you copied in the last step into the authorized keys file on the first server  sudo vi /root/.ssh/authorized\_keys     1. Install Fence agents on node1, node2   sudo zypper install fence-agents    Important If using Suse Linux Enterprise Server for SAP 15, be aware that you need to activate additional module and install additional component, that is prerequisite for using Azure Fence Agent. To learn more about SUSE modules and extensions see [Modules and Extensions explained](https://www.suse.com/documentation/sles-15/singlehtml/art_modules/art_modules.html). Follow the instructions bellow to install Azure Python SDK.    The following instructions on how to install Azure Python SDK are only applicable for Suse Enterprise Server for SAP **15**.   * If you are using Bring-Your-Own-Subscription, follow these instructions     #Activate module PackageHub/15/x86\_64  sudo SUSEConnect -p PackageHub/15/x86\_64  #Install Azure Python SDK  sudo zypper in python3-azure-sdk     * If you are using Pay-As-You-Go subscription, follow these instructions     #Activate module PackageHub/15/x86\_64  zypper ar <https://download.opensuse.org/repositories/openSUSE:/Backports:/SLE-15/standard/> SLE15-PackageHub  #Install Azure Python SDK  sudo zypper in python3-azure-sdk     1. Setup host name resolution on node1, node2   You can either use a DNS server or modify the /etc/hosts on all nodes. This example shows how to use the /etc/hosts file. Replace the IP address and the hostname in the following commands. The benefit of using /etc/hosts is that your cluster becomes independent of DNS, which could be a single point of failures too.    sudo vi /etc/hosts  Insert the following lines to /etc/hosts. Change the IP address and hostname to match your environment    # IP address of the first cluster node  Ip\_node1 node1  # IP address of the second cluster node  Ip\_node2 node2     1. Install Cluster on node1     sudo ha-cluster-init -u    # ! NTP is not configured to start at system boot.  # Do you want to continue anyway (y/n)? y  # /root/.ssh/id\_rsa already exists - overwrite (y/n)? n  # Address for ring0 [10.0.0.6] Press ENTER  # Port for ring0 [5405] Press ENTER  # SBD is already configured to use /dev/disk/by-id/scsi-36001405639245768818458b930abdf69;/dev/disk/by-id/scsi-36001405afb0ba8d3a3c413b8cc2cca03;/dev/disk/by-id/scsi-36001405f88f30e7c9684678bc87fe7bf - overwrite (y/n)? n  # Do you wish to configure an administration IP (y/n)? n     1. **A**dd node2 to cluster     sudo ha-cluster-join    # ! NTP is not configured to start at system boot.  # Do you want to continue anyway (y/n)? y  # IP address or hostname of existing node (e.g.: 192.168.1.1) [] ip\_node1  # /root/.ssh/id\_rsa already exists - overwrite (y/n)? n     1. Change hacluster password to the same password on node1, node2   sudo passwd hacluster     1. Adjust corosync settings on node1, node2     Add the following bold content to the file if the values are not there or different. Make sure to change the token to 30000 to allow Memory preserving maintenance. For more information, see [this article for Linux](https://docs.microsoft.com/en-us/azure/virtual-machines/maintenance-and-updates#maintenance-that-doesnt-require-a-reboot) or [Windows](https://docs.microsoft.com/en-us/azure/virtual-machines/maintenance-and-updates#maintenance-that-doesnt-require-a-reboot).    **[...]**  **token: 30000**  **token\_retransmits\_before\_loss\_const: 10**  **join: 60**  **consensus: 36000**  **max\_messages: 20**    interface {  [...]  }  transport: udpu  }  nodelist {  node {  ring0\_addr:10.0.0.6  }  node {  ring0\_addr:10.0.0.7  }  }  logging {  [...]  }  quorum {  # Enable and configure quorum subsystem (default: off)  # see also corosync.conf.5 and votequorum.5  provider: corosync\_votequorum  expected\_votes: 2  two\_node: 1  }    Then restart the corosync service    sudo service corosync restart |
|  | Default Pacemaker configuration for Azure fence agent | * 1. Enable the use of a STONITH device and set the fence delay on node1   sudo crm configure property stonith-timeout=144  sudo crm configure property stonith-enabled=true     1. # List the resources to find the name of the SBD device   sudo crm resource list  sudo crm resource stop stonith-sbd  sudo crm configure delete stonith-sbd  sudo crm configure primitive stonith-sbd stonith:external/sbd \  params pcmk\_delay\_max="15" \  op monitor interval="15" timeout="15"   1. Execute following commands, if you are using Azure fence agent as STONITH. After assigning roles to both cluster nodes, you can configure the STONITH devices in the cluster.   sudo crm configure property stonith-enabled=true  crm configure property concurrent-fencing=true  # replace the bold string with your subscription ID, resource group of the VM, tenant ID, service principal application ID and password  sudo crm configure primitive rsc\_st\_azure stonith:fence\_azure\_arm \  params subscriptionId="subscription ID" resourceGroup="resource group" tenantId="tenant ID" login="application ID" passwd="password" \  pcmk\_monitor\_retries=4 pcmk\_action\_limit=3 power\_timeout=240 pcmk\_reboot\_timeout=900 pcmk\_host\_map="prod-cl1-0:prod-cl1-0-vm-name;prod-cl1-1:prod-cl1-1-vm-name" \  op monitor interval=3600 timeout=120  sudo crm configure property stonith-timeout=900 |
|  | Pacemaker configuration for Azure scheduled events | Azure offers [scheduled events](https://docs.microsoft.com/en-us/azure/virtual-machines/linux/scheduled-events). Scheduled events are provided via meta-data service and allow time for the application to prepare for events like VM shutdown, VM redeployment, etc. Resource agent [**azure-events**](https://github.com/ClusterLabs/resource-agents/pull/1161) monitors for scheduled Azure events. If events are detected, the agent will attempt to stop all resources on the impacted VM and move them to another node in the cluster. To achieve that additional Pacemaker resources must be configured.     1. Make sure the package for the **azure-events** agent is already installed on node1, node2 and up to date     sudo zypper info resource-agents    **2.** Configure the resources in Pacemaker on node1    #Place the cluster in maintenance mode  sudo crm configure property maintenance-mode=true    #Create Pacemaker resources for the Azure agent  sudo crm configure primitive rsc\_azure-events ocf:heartbeat:azure-events op monitor interval=10s  sudo crm configure clone cln\_azure-events rsc\_azure-events    #Take the cluster out of maintenance mode  sudo crm configure property maintenance-mode=false    Note  After you configure the Pacemaker resources for azure-events agent, when you place the cluster in or out of maintenance mode, you may get warning messages like:  WARNING: cib-bootstrap-options: unknown attribute 'hostName\_ **hostname**'  WARNING: cib-bootstrap-options: unknown attribute 'azure-events\_globalPullState'  WARNING: cib-bootstrap-options: unknown attribute 'hostName\_ **hostname**'  These warning messages can be ignored. |

## CRM Configuration

|  |  |
| --- | --- |
| **Step** | **Notes** |
| # crm config show | node 1: hapcs01use2pr \  attributes azName=hapcs01use2pr standby=off  node 2: hapcs02use2pr \  attributes standby=off azName=hapcs02use2pr  primitive fs\_HAP\_ASCS01 Filesystem \  params device="10.213.32.103:/ascsers-use2-pr/HAP/ASCS" directory="/usr/sap/HAP/ASCS01" fstype=nfs options="sec=sys,vers=4.1" \  op start timeout=60s interval=0 \  op stop timeout=60s interval=0 \  op monitor interval=20s timeout=40s  primitive fs\_HAP\_ERS10 Filesystem \  params device="10.213.32.103:/ascsers-use2-pr/HAP/ERS" directory="/usr/sap/HAP/ERS10" fstype=nfs options="sec=sys,vers=4.1" \  op start timeout=60s interval=0 \  op stop timeout=60s interval=0 \  op monitor interval=20s timeout=40s  primitive nc\_HAP\_ASCS01 azure-lb \  params port=62001  primitive nc\_HAP\_ERS10 azure-lb \  params port=62110  primitive rsc\_azure-events azure-events \  op monitor interval=10s  primitive rsc\_sap\_HAP\_ASCS01 SAPInstance \  operations $id=rsc\_sap\_HAP\_ASCS01-operations \  op monitor interval=11 timeout=60 on-fail=restart \  params InstanceName=HAP\_ASCS01\_saphapcs01pr START\_PROFILE="/sapmnt/HAP/profile/HAP\_ASCS01\_saphapcs01pr" AUTOMATIC\_RECOVER=false \  meta resource-stickiness=5000 target-role=Started  primitive rsc\_sap\_HAP\_ERS10 SAPInstance \  operations $id=rsc\_sap\_HAP\_ERS10-operations \  op monitor interval=11 timeout=60 on-fail=restart \  params InstanceName=HAP\_ERS10\_saphapcs02pr START\_PROFILE="/sapmnt/HAP/profile/HAP\_ERS10\_saphapcs02pr" AUTOMATIC\_RECOVER=false IS\_ERS=true  primitive rsc\_st\_azure stonith:fence\_azure\_arm \  params subscriptionId=94d6ed7c-2ce4-4292-b7a1-aa60e5aab40a resourceGroup=sap-rg-hap-use2-pr tenantId=2658e698-ac38-4347-a56f-3f96a6bfa8ff login=570ed24c-105d-400b-a83e-6fd8043b19ea passwd="\*\*\*\*\*\*" pcmk\_monitor\_retries=4 pcmk\_action\_limit=3 power\_timeout=240 pcmk\_reboot\_timeout=900 \  op monitor interval=3600 timeout=120  primitive vip\_HAP\_ASCS01 IPaddr2 \  params ip=10.213.34.40 cidr\_netmask=24 nic=eth0 \  op monitor interval=10 timeout=20  primitive vip\_HAP\_ERS10 IPaddr2 \  params ip=10.213.34.42 cidr\_netmask=24 nic=eth0 \  op monitor interval=10 timeout=20  group SAP-HAP\_ASCS01 fs\_HAP\_ASCS01 nc\_HAP\_ASCS01 vip\_HAP\_ASCS01 rsc\_sap\_HAP\_ASCS01 \  meta resource-stickiness=3000  group SAP-HAP\_ERS10 fs\_HAP\_ERS10 nc\_HAP\_ERS10 vip\_HAP\_ERS10 rsc\_sap\_HAP\_ERS10  clone cln\_azure-events rsc\_azure-events  location cli-ban-rsc\_sap\_HAP\_ASCS01-on-hapcs01use2pr rsc\_sap\_HAP\_ASCS01 role=Started \  rule -inf: #uname eq hapcs01use2pr and date lt "2021-12-22 16:25:32Z"  colocation col\_sap\_HAP\_no\_both -5000: SAP-HAP\_ERS10 SAP-HAP\_ASCS01  order ord\_sap\_HAP\_first\_start\_ascs Optional: rsc\_sap\_HAP\_ASCS01:start rsc\_sap\_HAP\_ERS10:stop symmetrical=false  property cib-bootstrap-options: \  have-watchdog=false \  dc-version="1.1.19+20181105.ccd6b5b10-3.28.1-1.1.19+20181105.ccd6b5b10" \  cluster-infrastructure=corosync \  cluster-name=hacluster \  stonith-enabled=true \  maintenance-mode=false \  no-quorum-policy=ignore \  stonith-timeout=150s \  startup-fencing=true \  hostName\_hapcs01use2pr=hapcs01use2pr \  azure-events\_globalPullState=IDLE \  hostName\_hapcs02use2pr=hapcs02use2pr \  last-lrm-refresh=1640193021  rsc\_defaults rsc-options: \  resource-stickiness=1 \  migration-threshold=3  op\_defaults op-options: \  timeout=600 \  record-pending=true |
| # crm stat  On node 1 |  |
| # crm stat  On node 2 |  |

## REFERENCE DOCUMENTS / LINKS

Below are the links for documents mentioned in this document

|  |  |
| --- | --- |
| **Document Name** | **Microsoft Azure Documentation Link** |
| Setting up Pacemaker on SUSE Linux Enterprise Server in Azure | <https://docs.microsoft.com/en-us/azure/virtual-machines/workloads/sap/high-availability-guide-suse-pacemaker#cluster-installation> |

# Cluster patching process

Please follow the below steps to patch the server

zypper update -y (To install the update)

Once patches installed please Reboot the server, you can do it on terminal by using init 6/reboot commands or in Azure portal by using restart option

#reboot

Once patching has been completed successfully and servers is Up. Please take post checks

#zypper lu

#zypper pchk

#rpm -qa --last | more (Please take 1st page screenshot)

**If it’s a cluster server follow the below steps (before rebooting) for OS(Linux) patch installation**

1. Additional application server was stopped. Basis has performed application checks after stopping additional app server which was working fine.
2. IO team will place the Cluster in maintenance mode
3. Login to Secondary Node and validate the cluster roles are running on the primary node.
4. Install OS patches on the Secondary Node and reboot the Secondary Node.
5. Once the Secondary Node is up and running fine, perform server sanity health checks.
6. Login to primary Node and switch the cluster role to secondary node.
7. Install OS patches on the primary Node and reboot the primary Node.
8. Once the primary Node is up and running fine, perform server sanity health checks.
9. Login to Secondary Node and switch the cluster role to primary node. Login to primary and validate cluster roles are running file.
10. Inform Application team and Database Team to perform application availability\functionality test.
11. IO team will remove the cluster from maintenance mode post application team confirmation.

# HA TEsting

# DOCUMENT REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Date** | **Author** | **Reason / Change Description** |
| 1.0 |  |  | Initial Document |
|  |  |  |  |