Microsoft Azure and SUSE High Availability

TUT1134 – When Availability Matters

Mark Gonnelly
Senior Consultant

Stephen Mogg
Technical Strategist for SAP and Public Cloud
About This Session

What to Expect:
- HA concepts
- SUSE Cluster Solution
- Implementing HA in Azure
- Best Practices
- Demo
HA Concepts
HA Terms

MTTF

MTTR

RPO
The Goal of HA. Reduce:

MTTR
HA on Azure
Azure resiliency as a platform

HA Sets

To provide redundancy to an application, it is recommended to group two or more virtual machines in an availability set. This configuration ensures that during either a planned or unplanned maintenance event, at least one virtual machine will be available.
Azure resiliency as a platform
Availability Zones

AZ are physically separate locations within an Azure region. Each Availability Zone is made up of one or more datacenters equipped with independent power, cooling, and networking. For each region enabled for AZ, there are three Availability Zones.
Availability Zones

Physical DC / Availability Zones

Subscription 1

Subscription 2
SLAs Using Cloud Native HA Capability

- Single VM: 99.9%
- HA Set: 99.95% (2 VMs)
- Availability Zone: 99.99% (2 VMs)

99.9% Storage SLA (Single Storage account)

If your business needs a higher SLA – you need something more..
SUSE High Availability Extension
SUSE HAE Cluster Components

- corosync (cluster membership)
- pacemaker (crm)
- Resource Agents (RAs)
- Fencing (stonith)
- Kernel

Storage (SBD)
Corosync

Group communication system with additional features for implementing HA for applications
• Messaging and membership layer
• Communicates over multicast or unicast (Azure Unicast only)
• Performs cluster heartbeat
• SUSE Linux Enterprise Server 12/15 it is a separate systemd service

Synchronization, heartbeating etc.
• /etc/corosync/corosync.conf

Shared key for authentication:
• /etc/corosync/authkey
Pacemaker

Pacemaker sits on top of Corosync and manages / monitors / restarts / migrates cluster resources

• CIB (Cluster Information Base) is an XML representing entire cluster state (cibadmin(8))

• Once Pacemaker takes over ownership, nothing else must touch the resource directly, without first putting node / resource in maintenance mode.

• Monitoring with configurable retries and timeouts
Resource Agents

Provides ‘intelligence to Pacemaker’
A script used to start/stop/monitor a resource
• Ideally should be Open Cluster Framework compliant
  • Well defined return values
  • Mandatory operations
• Return value passed back to Pacemaker
• Many providers of RAs
  • Ships with around 140 RA out of the box.
  • Resource Agents for SAP HANA included in SLES for SAP Applications
Why Do We Need Fencing?

To a cluster node, loss of a peer node is indistinguishable from loss of communication with that node.

In the former case, is it safe to failover resources?

And in the latter case?
Split Brain

- When a cluster partitions due to network failure
- Neither side knows if the other is still alive
- Worst case scenario: each side attempts to failover the other's resources
- Better scenario: neither side does anything
  - But then, why do we have a cluster?
- Best scenario: one side is able to guarantee that the other is down
  - Fencing is about moving from an UNKNOWN state to a KNOWN state
SUSE High Availability in Azure
BYOS vs PAYG

**SUSE Linux Enterprise Server**
This is the base OS
Available in Azure

**SUSE Linux Enterprise Server HA Add-on**
This extends the base OS
*BYOS only

**SUSE Linux Enterprise Server for SAP Applications**
Is a **BUNDLE** of the above + special SAP additions + services
Available in Azure
Clustering in the Public Cloud.
The same but different

- Need a shared block device between machines
  - Needed by SBD

- Shared storage (NFS/SMB)
  - Needed by applications

- Control over all network layers
  - Needed by virtual ip failover

_Cluster settings are different from on premises implementations_
Corosync Changes

Increasing timeout (30 Seconds)

[...

token: 30000

token_retransmits_before_loss_const: 10

join: 60

consensus: 36000

max_messages: 20

Fencing of the nodes

- The STONITH device uses a Service Principal to authorize against Microsoft Azure.
- You need to give the Service Principal permissions to start and stop (deallocate) all virtual machines of the cluster.
- The Azure infrastructure is not able to do a kill or force shutdown of a node (only a graceful shutdown).
- Not recommended for anything time critical.

The STONITH device uses a Service Principal to authorize against Microsoft Azure.

You need to give the Service Principal permissions to start and stop (deallocate) all virtual machines of the cluster

```bash
# replace the bold string with your subscription ID, resource group, tenant ID, service principal ID and password
primitive rsc_st_azure stonith:fence_azure_arm
   params subscriptionId="subscription ID" \
   resourceGroup="resource group" \
   tenantId="tenant ID" \
   login="login ID" \
   passwd="password"
```

You need to set a very long stonith-timeout in order to give the agent time to deallocate and restart the machines.

```bash
crm configure property stonith-timeout=900
```
Fencing of the nodes

- As the Azure infrastructure is not able to do a kill or force shutdown of a node (only a graceful shutdown), we stick to the concept of the **SBD device for fencing** with help of an additional very small instance providing a **raw shared disk over iSCSI**.
- From Cluster point of view not different to bare metal

SBD

STONITH Block Device (SBD) fencing is recommended by SUSE
- SBD fencing is highly reliable thanks to hardware watchdog integration

• Independent of management board (firmware, settings, etc.)
• Equal setup in physical and virtual environments, reducing variance in deployments

Integration with Pacemaker & corosync status!
Virtual IP movement between the nodes

- IP movement between the nodes is done by the Azure Loadbalancer with a health probe (*), together with the RA IPAddress2

- It needs an additional rule to the rules in our best practice documents for the probe request.

HA in Azure – IP Address

```bash
sudo crm configure primitive rsc_ip_HN1_HDB03 ocf:heartbeat at:IPaddr2 meta target-role="Started" is-managed="true" operations \$id="rsc_ip_HN1_HDB03-operations" \ op monitor interval="10s" timeout="20s" \ params ip="10.0.0.13"

sudo crm configure primitive rsc_nc_HN1_HDB03 anything \ params binfile="/usr/bin/nc" cmdline_options="-l -k 62503" \ op monitor timeout=20s interval=10 depth=0
```

HA in Azure – NFS (Shared Storage)

Enterprise NFS is coming, until then we need to build an NFS Service

HA NFS Storage with DRBD and Pacemaker

- Use same concepts for IP failover and fencing as mentioned before
- Included in SLES HA
- Documented in standard SUSE HAE documentation

[Diagram showing NFS setup]

DRBD

- Block device that is mirrored with a block device on another computer
- Data is mirrored using the network as transport
- Can be thought of a networked RAID 1
DRBD Configuration

/etc/drbd.conf
main configuration file for DRBD
typically contains only includes statements

/etc/drbd.d/
configuration file include directory

/etc/drbd.d/global_common.conf
file containing the common global configuration directives for DRBD
directives can be overridden by resource specific directives

/etc/drbd.d/*.res
resource (device) definition files
Azure Storage - SMB

- Fully Managed File Shares in the Cloud
- “Lift and shift” legacy apps
- SMB and REST access
- Locally or Geo-Redundant

Virtual machine

Azure Files
\\<account>.file.windows.net\<share>
Microsoft Azure Events Resource Agent

azure-events:
Monitors Azure event metadata, and places node into standby if affected by an upcoming maintenance event. (useful for NFS service?)

Configure primitive resource AzEvents
```
crm configure primitive rsc_AzEvents
ocf:heartbeat:AzEvents op monitor interval=10s
```

Configure clone resource AzEvents
```
crm configure clone cln_AzEvents rsc_AzEvents
```
Conclusion
Setting up Pacemaker on SUSE Linux Enterprise Server in Azure

08/16/2018 • ☑ 15 minutes to read • Contributors ⬤ ⬤ ⬤ ⬤ all

In this article

- SBD fencing
- Cluster installation
- Create Azure Fence agent STONITH device
- Default Pacemaker configuration for SBD

Next steps

There are two options to set up a Pacemaker cluster in Azure. You can either use a fencing agent, which takes care of restarting a failed node via the Azure APIs or you can use an SBD device.
In Conclusion

- Clustering improves reliability, but does not achieve 100%, ever.
- Fail-over clusters reduce service outage, but do not eliminate it.
- High Availability protects data before the service.
- Clusters are more complex than single nodes.
- Clustering broken applications will not fix them.
- Invest in training, processes, knowledge sharing.
- Get expert help for the initial setup, and
- Thoroughly test the cluster regularly.
- Finally – KEEP IT SIMPLE!
Other SUSECON Sessions

- SUSE workloads on Microsoft Azure [CAS1403]
- Fundamentals of managing and securing your SLES workloads on Azure [SPO1454]
- Workshop Install SAP HANA on SLES12 in Azure Cloud [HO1088]
- SLES for SAP HANA On Azure [CAS1086]