VM-Series / Microsoft Azure

VM-Series in Azure

Fast failover for HA cluster

https://github.com/cestebanez91/Azure-VmSeries-Fast-HA

Last Update: April 2020

Table of Contents

1. Presentation of concept

1. Goal of this design

Ensuring VM-Series resiliency in Azure can be achieved in different manners with pros and cons:

- **Deploy an Active / Passive HA cluster of VM-Series**
  This pattern allows session synchronization between primary and secondary VM-Series.
  As there is only one active path at a moment, Source Nat is not needed to avoid asymmetric path return.
  However, this pattern is not horizontally scalable and failover time can take a while, whatever we use secondary IP address move or UDR update methods.

- **Deploy a VM-Series backend pool with Load Balancers**
  This pattern is fully horizontally scalable and failover time is reduced to minimum as all instances in the Load Balancer's backend pool are active.
  However, there is no session synchronization and Source Nat is mandatory to ensure symmetric path return for inbound traffic use case.

The purpose of this new pattern is to provide a setup for VM-Series resiliency in Azure where:

- **Session synchronization is applied**
- **Failover is fast**
- **Source NAT is not needed**

Horizontal scalability is not doable but upgrading capacity for VM-Series is still doable (vertical scaling).

This pattern is based on mutual use of Active / Passive HA cluster and inbound/outbound Load Balancers.

VM-Series are configured in a particular manner to get the correct behavior.

2. How it works

We use an active / passive cluster where only one node is active at a moment, Primary one.

Passive instance does not process Load Balancer health probe so the passive instance is considered as unhealthy then does not receive traffic from Load Balancer, in both direction.

When active node is lost, native PAN-OS HA mechanism performs fast failover between primary and secondary.

Secondary node starts to acknowledge health probe packets so the secondary node becomes healthy.

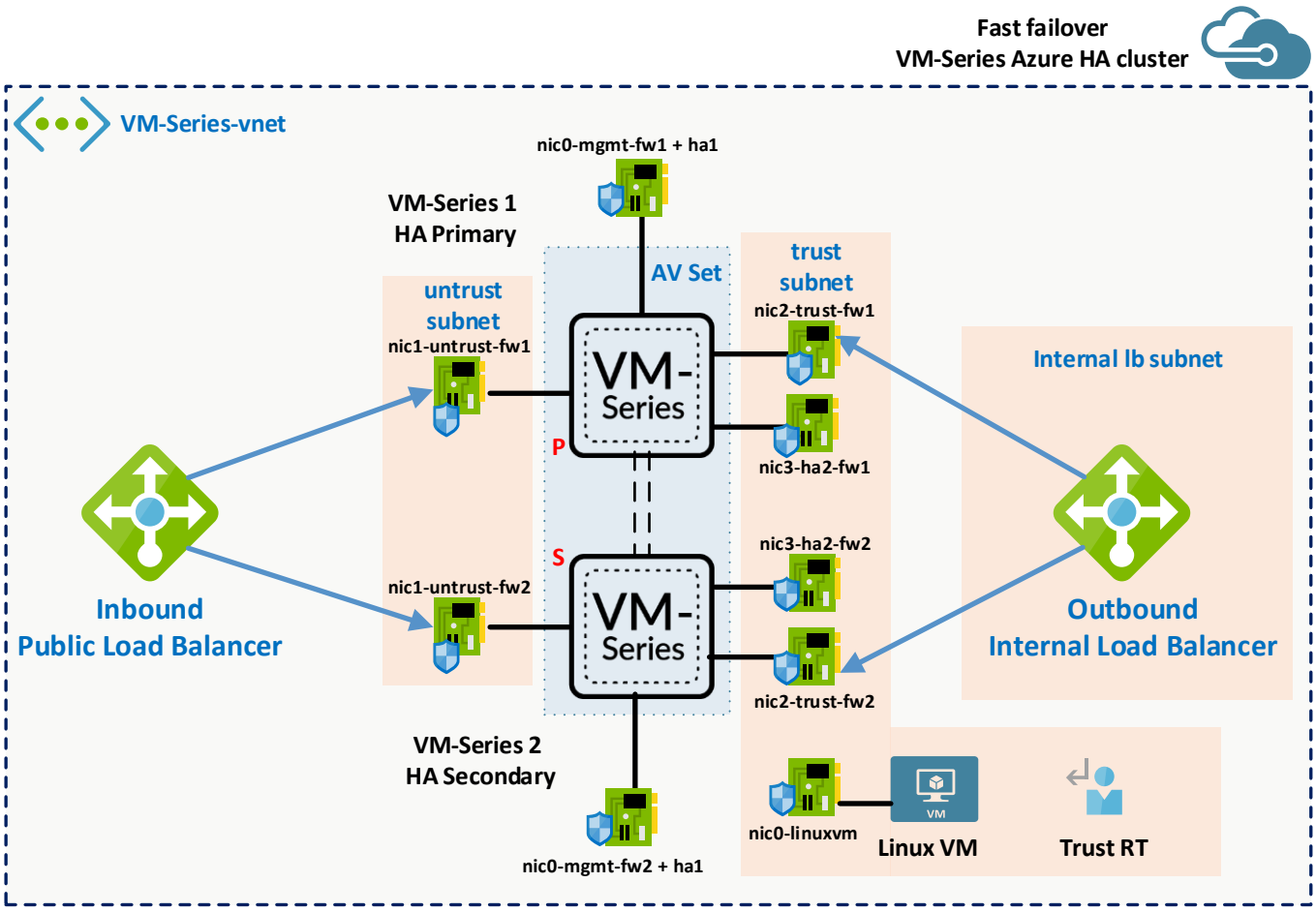
It starts to receive traffic in both directions.

Sessions are synchronized between primary and secondary instances.

Specific VM-Series configuration:

- Session synchronization **is enabled** in high availability configuration.
- Configuration synchronization **is not enabled** in high availability configuration.
    - It allows to have different network interfaces IP addresses configurations on VM1 and VM2 in an active / passive cluster (where both VM are supposed to be mirrored configuration).
    - NAT policies are common for both VM-Series so networks all network interfaces are configured with a primary IP address and a secondary IP address coming from the other node.
    - NAT policies and Security policies are pushed from a Panorama device group in order to overcome non synchronization of configurations between VM1 and VM2.

2. Diagram



For both VM-Series:

- Ethernet 1/0 is used for management interface as usual, in *management subnet*.
- Ethernet 1/1 is used for untrust interface, in *untrust subnet*.
- Ethernet 1/2 is used for trust interface, un *trust subnet*.
- Ethernet 1/3 us used for HA2 interface, in *trust subnet* as well.
- Management interface is used for HA1.

Inbound Public Load Balancer use untrust interfaces as back end pool.

Outbound internal Load Balancer use trust interfaces as back end pool.

Linux test VM is deployed in trust subnet and *Trust RT* route table has a UDR pointing default traffic to outbound internal Load Balancer.

3. Deployment

This deployment is a one click but composed of two steps.

First step is to deploy all Azure components, Virtual Network, subnets, Load Balancers, VM-Series instances, Linux machines and so on.

The second step consists in using bootstrapping process when VM-Series are spin up in order to get the right configuration for HA.

The first step uses an Azure ARM template available on the Github repository.

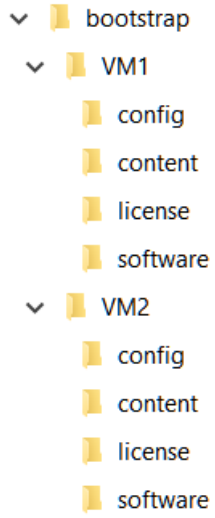
For information this is a modified version based on this one:

https://github.com/wwce/azure-arm/tree/master/Azure-Common-Deployments/v1/2fw_3nic_avset_intlb_extlb
Thanks Matt McLimans for this everyday useful template.

The second step needs you complete your bootstrapping packages for VM1 and VM2 prior to launching your ARM template.

The zip file contains bootstrap structure including one directory for VM1 and another one for VM2 as they have a different configuration.

Below is a typical structure for bootstrapping VM1 and VM2 in your Azure Storage Account.



"bootstrap" is the Azure file shares name

"VM1" and "VM2" are directories in this file share.

"Config", "content", "software" and "license" are subdirectories in "VM1" and "VM2" directories.

Config directories will contain init-cfg.txt file and bootstrap.xml file.

License directories will contain your authcodes if needed for BYOL.

Default username and password for VM1, VM2 and Linux instances are **paloalto/paloalto123!**