stryker

Azure Recommendations

Notice

Last modified: 2023-02-10 06:29

Azure-Development-Docs build 63

Contents

About the Vocera Platform

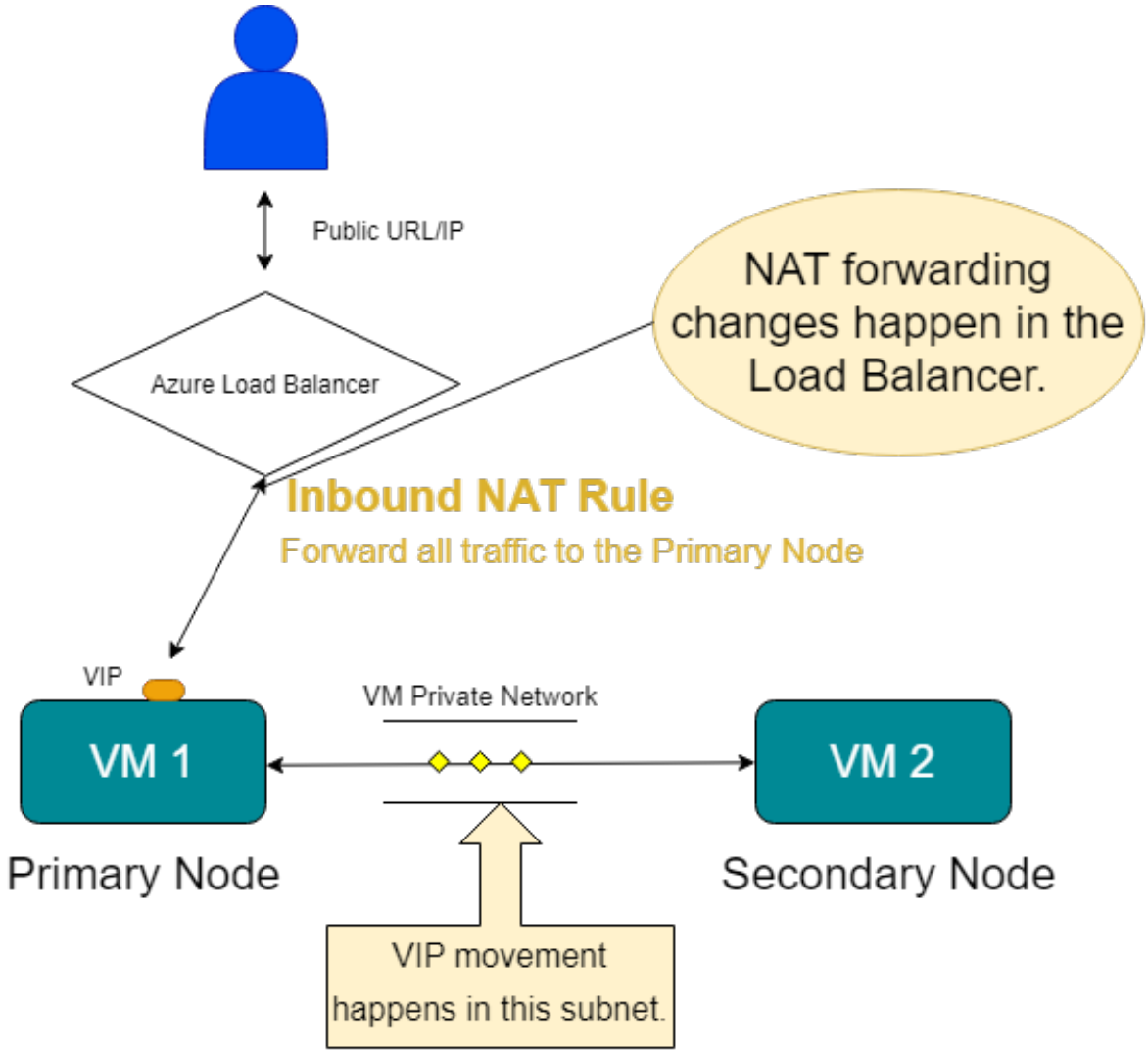
The Vocera Platform optimizes patient safety by helping clinicians make real-time decisions and communicate instantly in critical situations; it is the intelligent ecosystem that connects all the people and information needed to deliver patient care.

With the intelligence of the Vocera Platform, clinicians can quickly determine what to prioritize next and close the loop faster with secure messaging, phone calls, and alert and alarm notifications--all in one place. You can locate people quickly, collaborate productively, and reduce the noise, using the device that fits your workflow--the Vocera Vina smartphone app, the V-Series Smartbadge, or the B-Series Badge. The Vocera Platform enables the flow of meaningful, actionable information between people and systems and allows it to be received when, where, and how it's needed, keeping the patient at the center of care.

In addition, the Vocera Platform now offers simplified deployment, maintenance, and administration, as well as a smaller footprint requiring fewer servers.

Vocera Platform 6X Cluster Setup and VIP movement in the Azure Cloud

This section provides a high level architectural design for Azure and the Virtual Machines (VM) inside of the Azure environment. Additionally, an overview of how Virtual IP (VIP) movement happens in the Azure environment is explained below.



Azure CLI Script: To change the inbound NAT rules on the Load Balancer

How VIP Movement Happens

VIP movement is configured and happens internally inside the Azure Cloud private network between VM 1 and VM 2. The configuration for these VM's occurs in the same way configuration occurs for any VM that is hosted on premises, such as VMware. Once the VM's have been established, all user requests will be filtered through the Azure load balancer.

After establishing the VM's, the client must set up Inbound Network Address Translation (NAT) rules. The Azure CLI script must be used to change and update any NAT rules. Inbound NAT rules are an optional setting in the Azure load balancer, however we are requiring clients to set up these rules. An Inbound NAT rule creates port mappings from frontend to backend, forwarding traffic over a specific port on the frontend to a specific port in the backend. For example, port 389 is required to be opened for LDAP to function appropriately. By writing an Inbound NAT rule, it will ensure all LDAP traffic is always routed through port 389.

Required Minimum Resource Allocation

This section provides information on the minimum required resource allocation for the Azure virtual machine (VM) for planning a Vocera deployment.

This guidance represents the resources required to provide adequate performance for simulated customer environments. These specifications should be considered a starting point. Additional factors such as the number and type of system integrations can impact overall system sizing, and we recommend that you discuss this with your Vocera account team.

The minimum Vocera recommended deployment is two nodes to form a cluster. The resources listed in the table below will need to be multiplied to account for your environment.

All verbiage reflects the Azure environment. This is different than VMWare or other VM providers and hosts.

The following table lists minimum required resources to be allocated to the Azure virtual machine (VM).

Azure VM Sizing, vCPU/RAM, and Drive Requirements

VM Size	Description
Standard D32ds_v4	vCPUs: 32 RAM: 128 GiB

Drive	Description	Number of Units Required
Azure P30 Drives	IOPS: 5000 Throughput: 200MB/s	4
Azure P15 Drives	IOPS: 1100 Throughput: 125MB/s	2

There should be a minimum of 6 drives per installation, four of the Azure P30 drives and two of the Azure P15 drives. These drives account for running the Vocera product, infrastructure, logs, ASL, and a drive for backups.

There is no oversubscription for Azure drives. It is imperative that the customer select the correct drive at the outset because they will not be able to adjust how much space is dedicated to any core. The Azure drives are fixed and noted in the table above.

Azure Recommendations and Requirements

The majority of the Azure configuration will be performed by the IT staff at the facility. It is imperative that the Implementation Engineer or anyone assisting in brining the client up in the Azure environment understand the fundamental processes and what specific information is needed to ensure a smooth and effective installation. This document will walk through the prerequisites of the Azure deployment at a high level and illustrate the main differences and design principles for Azure images.

Prerequisites

In order to set up an Azure deployment, the customer will need access to the following items:

1. Obtain access to Azure through the customer Active Directory login.
2. Log in to Azure with the customer Active Directory account: https://portal.azure.com/
3. Create an Azure Resource Group to collate all the machines. This is an optional step for clients who have multiple machines.
4. Create a Storage Account. This is an optional step.
5. Install the Azure Storage Explorer. This is a required step.
6. Create a partion in Red Hat Enterprise Lincx (RHEL). Make it visible to the RHEL-rootvol volume and resize the disk.

Major Differences and Design Principles for Azure Images

- Azure Virtual Machines (VM) are initially prepared on a Hyper-V platform.
- The OS disk storage will be exported and converted from .VHDX to .VHD format.
  - The .VHD format will be a fixed size and will allocate the entire storage size in a cloud environment.
  - This forces the VM to be smaller in initial storage size, around 10GB. However, it is possible to expand the size post deployment using logical volumes.
- Serial port support and login would be enabled to for debugging, as no monitor screen access exists on the Cloud.
- Azure tools and supporting management packaged are additionally added
  - Azure CLI: az command
  - Azure Cloud services packages: WALinuxAgent cloud-init cloud-utils-growpart gdisk hyperv-daemons

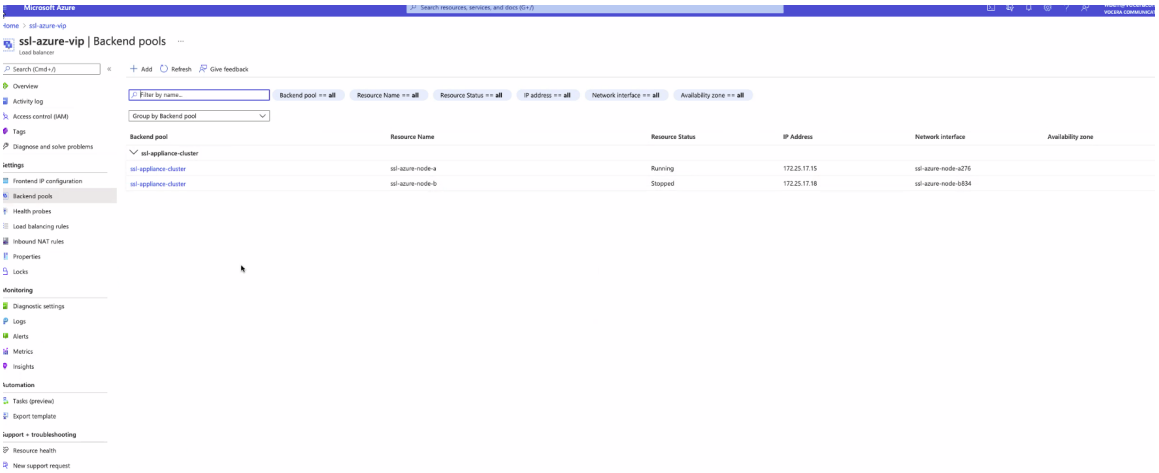
How To Use the Azure Recommendations

The following document describes, at a high level, Backend Pools, the Load Balancer and Rules, and Health Probes. The client will need to set up all three pieces for their Azure integration. This document is not intended to be a full walk through of an Azure deployment; please see Microsoft's Azure documentation for details.

After a client has installed the Azure software, they will need to establish three critical pieces: Backend Pools, Load Balancer Rules, and Health Probes. It is important to remember that the client's IT department will be responsible for setting up Azure as a whole and for creating the pools, rules, and probes. However, it is also important for the Implementation Engineer to be aware of how the client has set these up and be available for guidance questions regarding how Azure interacts with the Vocera Platform.

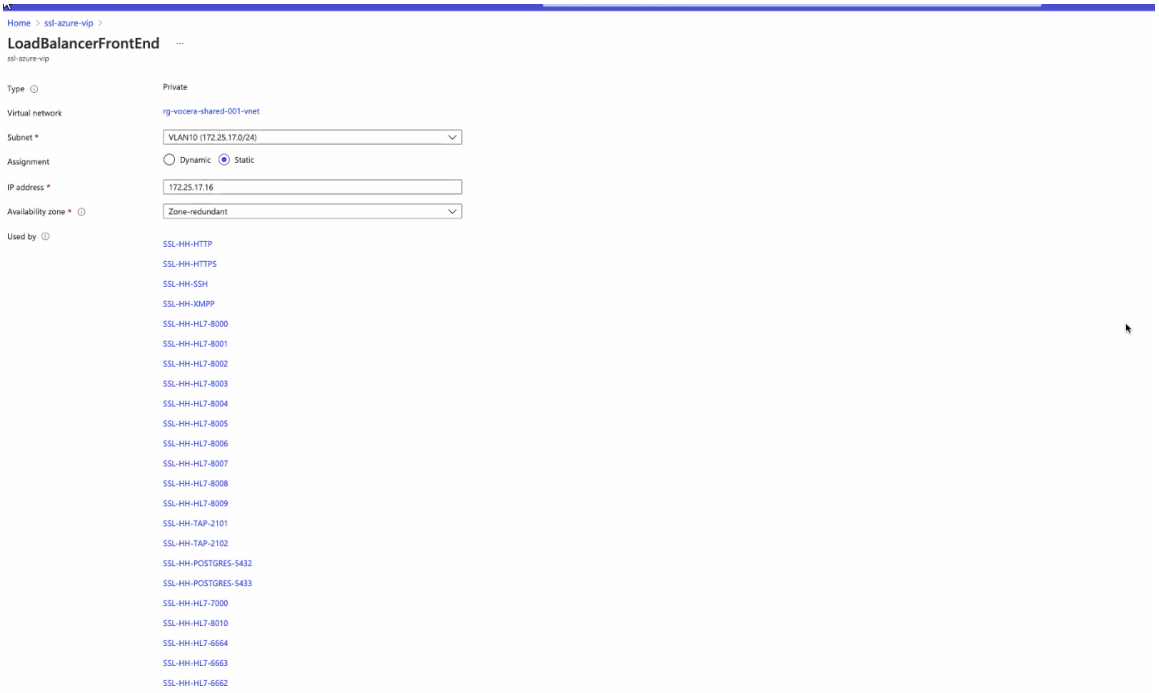
Backend Pools

Backend Pools are defined by Microsoft as: "a group of resources that will serve traffic for a given load-balancing rule." They can also be described as the combination of IP address and a Virtual Network Resource ID. Yet another way to think of the backend pool is to imagine a node. If the client wants to have a Primary Node and a Secondary Node, then they will need two backend pools, one for each node. Below is an example of what the backend pools will look in the Azure dashboard. Notice that the client has set up two pools. They have named one ssl-azure-node-a and the second pool is named ssl-azure-node-b. Under the **Resource Status** heading the first node is running and the second is stopped. We would expect to see one node running and the other in a stopped state awaiting a failover if necessary.



Azure Load Balancer

The Azure Load Balancer is a Layer-4 (TCP and UDP) load balancer that provides high availability by distributing incoming traffic to healthy VMs. Below is a screenshot from the Azure dashboard showing the load balanced front end, as well as all of the Load Balancing Rules that have been created in order to most efficiently direct traffic to healthy VMs.

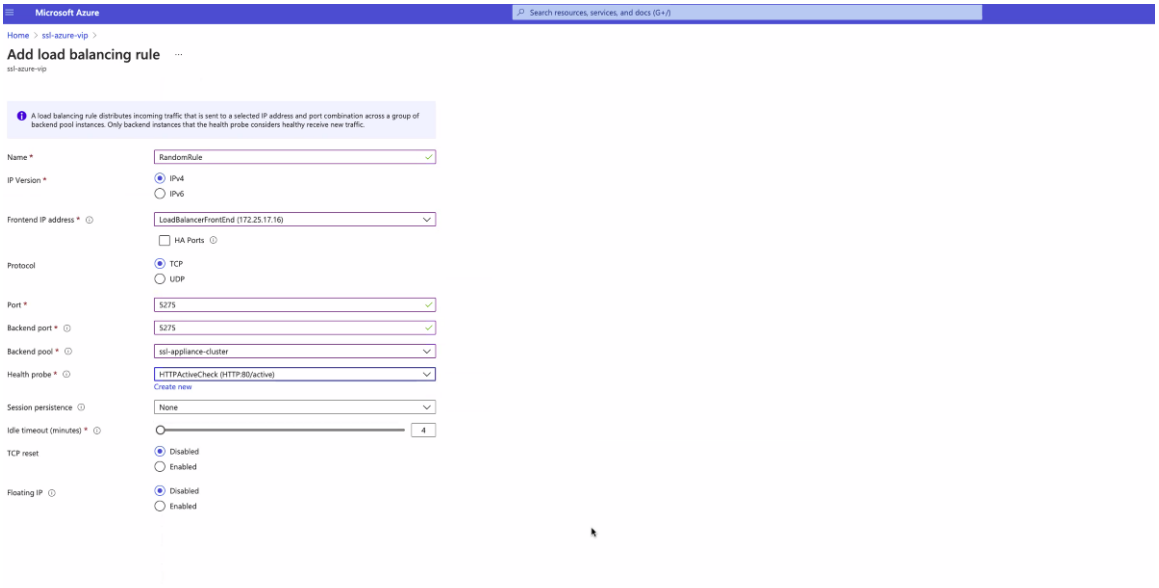


Load Balancing Rules

Now that the Load Balancer has been created, the client must begin to create rules. While a load balancer will evenly distribute incoming network traffic to backend resources, servers, or VMs, rules must be written in order to tell the load balancer where to direct incoming traffic. A load balancing rule distributes incoming traffic that is sent to a selected IP address and port combination across a group of backend pool instances.

There will, at times, be multiple rules for an adapter. For example, for every segment of HL7, a separate load balancing rule must be created. Ensure that the client has a load balancing rule for every port. It is advised to ask the client for a list and check it against their configuration to ensure that all rules have been established.

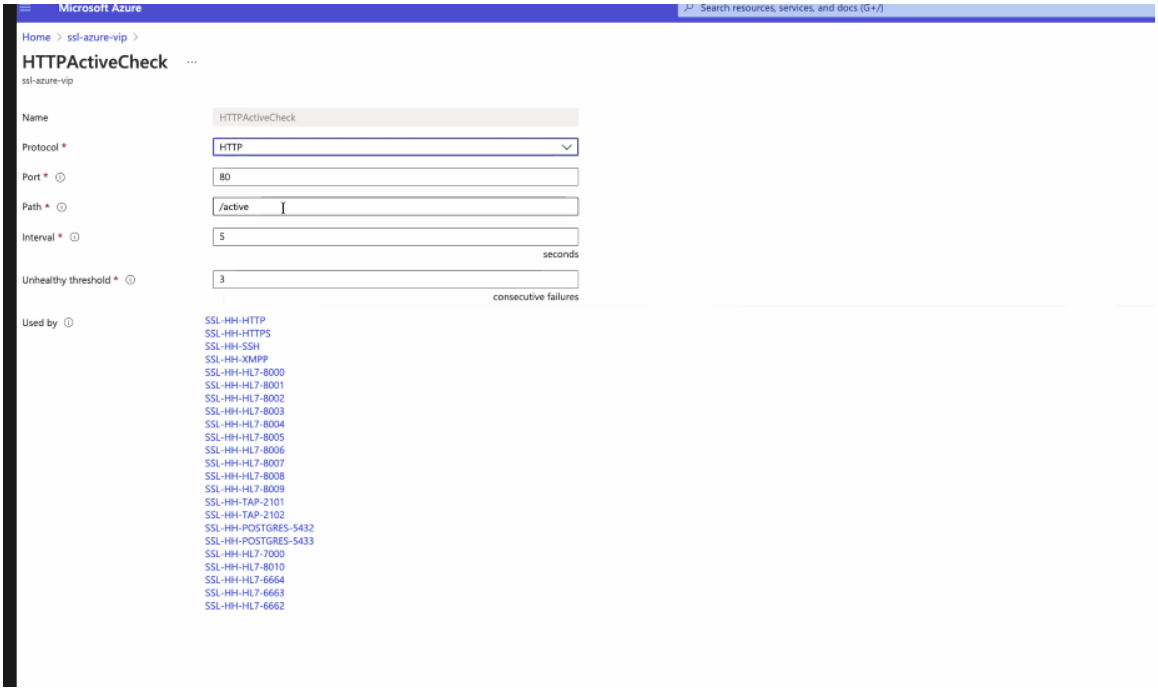
Note: A Health Probe must be attached to every load balancing rule.



Field	Definition
Name	Enter the name of the rule.
IP Version	Select either IPv4 or IPv6, based on client needs.
Frontend IP address	Enter in the IP address that you would like to route the traffic to.
Protocol	Select either TCP or UDP. Note that in the initial iteration of Azure, we recommend selecting TCP only.
Port	Enter in the required port for the rule. For example if the client were writing a rule for LDAP, they would place 389 in this box.
Backend Port	The backend port should match the Port in the previous field.
Backend Pool	The client should select the running backend pool from the drop down list.
Health Probe	The client should select the appropriate Health Probe from the drop down list. If the appropriate Health Probe is not available in the drop down list, use the Create Rule hyperlink.

Health Probes

The final piece that a client must set up is the Health Probe. Once configured, the health probe will run checks to determine if the instance is healthy. If the instance fails its health probe enough times, it will stop receiving traffic until it starts passing health probes again.



Field	Definition
Name	Insert the name of the rule in this field.
Protocol	Select the correct protocol from the drop down box. HTTP or HTTPS
Port	Enter the port number that you would like to run the health probe against. The example selects port 80.
Path	This is the remainder of the full URL for the custom probe. A valid path starts with '/'. In our example, we are checking that the path is active.
Interval (in seconds)	How often the probe runs to check for health. Azure does not recommend setting the interval to lower than 30 seconds.
Unhealthy Threshold	Number of consecutive failed attempts to be considered unhealthy. This can be set to 1 or more.

Finally, you will notice a list containing all of the rules that employ this health probe. While there does need to be a rule for every port, there can be multiple rules attached to the same health probe.