

Linux N-tier application in Azure with Apache Cassandra

11/12/2018 • 10 minutes to read • Contributors 👤 👤 👤 👤

In this article

[Architecture](#)

[Recommendations](#)

[Scalability considerations](#)

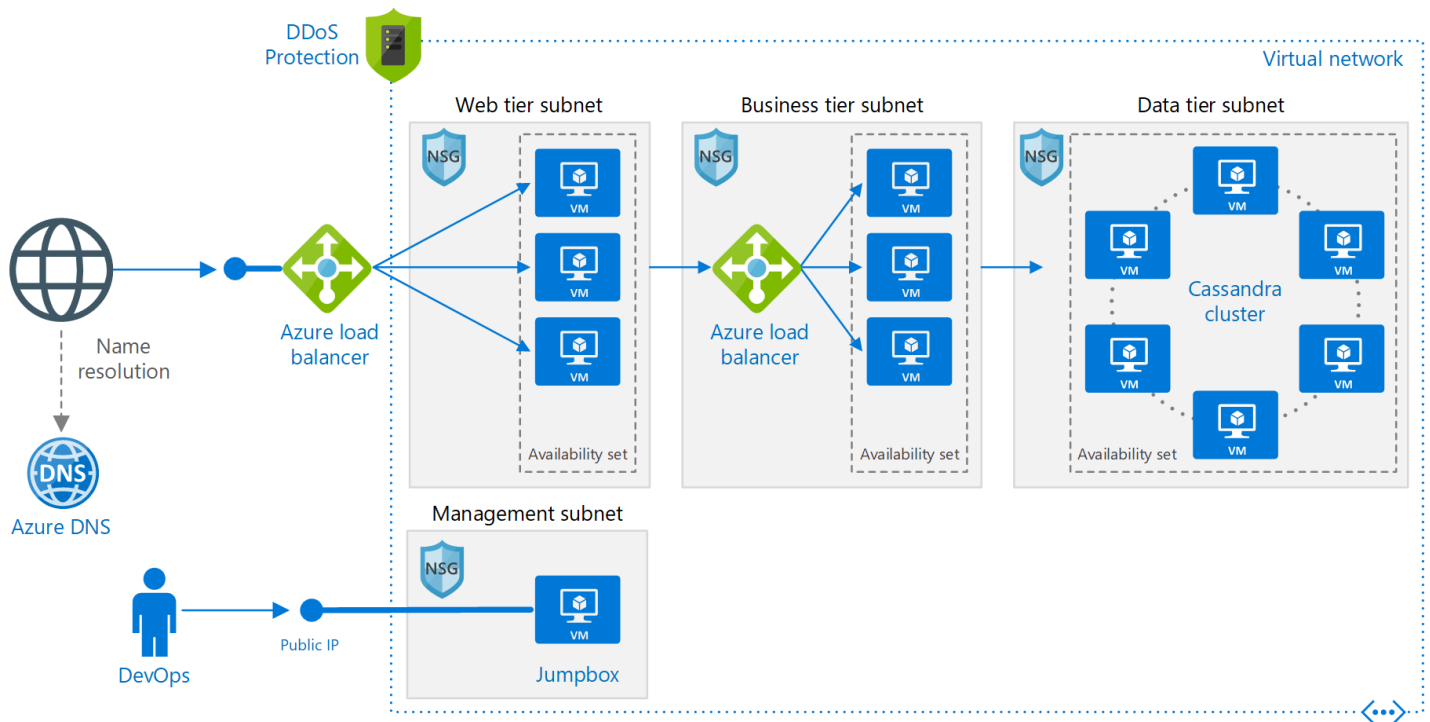
[Availability considerations](#)

[Security considerations](#)

[Deploy the solution](#)

[Next steps](#)

This reference architecture shows how to deploy virtual machines (VMs) and a virtual network configured for an [N-tier](#) application, using Apache Cassandra on Linux for the data tier. [Deploy this solution](#).



Download a [Visio file](#) of this architecture.

Architecture

The architecture has the following components:

- **Resource group.** [Resource groups](#) are used to group resources so they can be managed by lifetime, owner, or other criteria.
- **Virtual network (VNet) and subnets.** Every Azure VM is deployed into a VNet that can be segmented into subnets. Create a separate subnet for each tier.
- **NSGs.** Use [network security groups](#) (NSGs) to restrict network traffic within the VNet. For example, in the three-tier architecture shown here, the database tier accepts traffic from the business tier and the management subnet, but not the web front end.

- **DDoS Protection.** Although the Azure platform provides basic protection against distributed denial of service (DDoS) attacks, we recommend using [DDoS Protection Standard](#), which has enhanced DDoS mitigation features. See [Security considerations](#).
- **Virtual machines.** For recommendations on configuring VMs, see [Run a Windows VM on Azure](#) and [Run a Linux VM on Azure](#).
- **Availability sets.** Create an [availability set](#) for each tier, and provision at least two VMs in each tier, which makes the VMs eligible for a higher [service level agreement \(SLA\)](#).
- **Azure load balancers.** The [load balancers](#) distribute incoming Internet requests to the VM instances. Use a [public load balancer](#) to distribute incoming Internet traffic to the web tier, and an [internal load balancer](#) to distribute network traffic from the web tier to the business tier.
- **Public IP address.** A public IP address is needed for the public load balancer to receive Internet traffic.
- **Jumpbox.** Also called a [bastion host](#). A secure VM on the network that administrators use to connect to the other VMs. The jumpbox has an NSG that allows remote traffic only from public IP addresses on a safe list. The NSG should allow ssh traffic.
- **Apache Cassandra database.** Provides high availability at the data tier, by enabling replication and failover.
- **Azure DNS.** [Azure DNS](#) is a hosting service for DNS domains. It provides name resolution using Microsoft Azure infrastructure. By hosting your domains in Azure, you can manage your DNS records using the same credentials, APIs, tools, and billing as your other Azure services.

Recommendations

Your requirements might differ from the architecture described here. Use these recommendations as a starting point.

VNet / Subnets

When you create the VNet, determine how many IP addresses your resources in each subnet require. Specify a subnet mask and a VNet address range large enough for the required IP addresses, using [CIDR](#) notation. Use an address space that falls within the standard [private IP address blocks](#), which are 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16.

Choose an address range that doesn't overlap with your on-premises network, in case you need to set up a gateway between the VNet and your on-premises network later. Once you create the VNet, you can't change the address range.

Design subnets with functionality and security requirements in mind. All VMs within the same tier or role should go into the same subnet, which can be a security boundary. For more information about designing VNets and subnets, see [Plan and design Azure Virtual Networks](#).

Load balancers

Do not expose the VMs directly to the Internet. Instead, give each VM a private IP address. Clients connect using the IP address of the public load balancer.

Define load balancer rules to direct network traffic to the VMs. For example, to enable HTTP traffic, create a rule that maps port 80 from the front-end configuration to port 80 on the back-end address pool. When a client sends an HTTP request to port 80, the load balancer selects a back-end IP address by using a [hashing algorithm](#) that includes the source IP address. Client requests are distributed across all the VMs.

Network security groups

Use NSG rules to restrict traffic between tiers. For example, in the three-tier architecture shown above, the web tier does not communicate directly with the database tier. To enforce this, the database tier should block incoming traffic from the web tier subnet.

1. Deny all inbound traffic from the VNet. (Use the `VIRTUAL_NETWORK` tag in the rule.)
2. Allow inbound traffic from the business tier subnet.
3. Allow inbound traffic from the database tier subnet itself. This rule allows communication between the database VMs, which is needed for database replication and failover.
4. Allow ssh traffic (port 22) from the jumpbox subnet. This rule lets administrators connect to the database tier from the jumpbox.

Create rules 2 – 4 with higher priority than the first rule, so they override it.

Cassandra

We recommend [DataStax Enterprise](#) for production use, but these recommendations apply to any Cassandra edition. For more information on running DataStax in Azure, see [DataStax Enterprise Deployment Guide for Azure](#).

Put the VMs for a Cassandra cluster in an availability set to ensure that the Cassandra replicas are distributed across multiple fault domains and upgrade domains. For more information about fault domains and upgrade domains, see [Manage the availability of virtual machines](#).

Configure three fault domains (the maximum) per availability set and 18 upgrade domains per availability set. This provides the maximum number of upgrade domains that can still be distributed evenly across the fault domains.

Configure nodes in rack-aware mode. Map fault domains to racks in the `cassandra-rackdc.properties` file.

You don't need a load balancer in front of the cluster. The client connects directly to a node in the cluster.

For high availability, deploy Cassandra in more than one Azure region. Nodes within each region are configured in rack-aware mode with fault and upgrade domains, for resiliency inside the region.

Jumpbox

Don't allow ssh access from the public Internet to the VMs that run the application workload. Instead, all ssh access to these VMs must come through the jumpbox. An administrator logs into the jumpbox, and then logs into the other VM from the jumpbox. The jumpbox allows ssh traffic from the Internet, but only from known, safe IP addresses.

The jumpbox has minimal performance requirements, so select a small VM size. Create a [public IP address](#) for the jumpbox. Place the jumpbox in the same VNet as the other VMs, but in a separate management subnet.

To secure the jumpbox, add an NSG rule that allows ssh connections only from a safe set of public IP addresses. Configure the NSGs for the other subnets to allow ssh traffic from the management subnet.

Scalability considerations

For the web and business tiers, consider using [virtual machine scale sets](#), instead of deploying separate VMs into an availability set. A scale set makes it easy to deploy and manage a set of identical VMs, and autoscale the VMs based on performance metrics. As the load on the VMs increases, additional VMs are automatically added to the load balancer. Consider scale sets if you need to quickly scale out VMs, or need to autoscale.

There are two basic ways to configure VMs deployed in a scale set:

- Use extensions to configure the VM after it's deployed. With this approach, new VM instances may take longer to start up than a VM with no extensions.

- Deploy a [managed disk](#) with a custom disk image. This option may be quicker to deploy. However, it requires you to keep the image up-to-date.

For more information, see [Design considerations for scale sets](#).

Tip

When using any autoscale solution, test it with production-level workloads well in advance.

Each Azure subscription has default limits in place, including a maximum number of VMs per region. You can increase the limit by filing a support request. For more information, see [Azure subscription and service limits, quotas, and constraints](#).

Availability considerations

If you don't use virtual machine scale sets, put VMs for the same tier into an availability set. Create at least two VMs in the availability set to support the [availability SLA for Azure VMs](#). For more information, see [Manage the availability of virtual machines](#). Scale sets automatically use *placement groups*, which act as an implicit availability set.

The load balancer uses [health probes](#) to monitor the availability of VM instances. If a probe can't reach an instance within a timeout period, the load balancer stops sending traffic to that VM. The load balancer will continue to probe, and if the VM becomes available again, the load balancer resumes sending traffic to that VM.

Here are some recommendations on load balancer health probes:

- Probes can test either HTTP or TCP. If your VMs run an HTTP server, create an HTTP probe. Otherwise create a TCP probe.
- For an HTTP probe, specify the path to an HTTP endpoint. The probe checks for an HTTP 200 response from this path. This can be the root path ("/"), or a health-monitoring endpoint that implements some custom logic to check the health of the application. The endpoint must allow anonymous HTTP requests.
- The probe is sent from a [known IP address](#), 168.63.129.16. Make sure you don't block traffic to or from this IP address in any firewall policies or NSG rules.
- Use [health probe logs](#) to view the status of the health probes. Enable logging in the Azure portal for each load balancer. Logs are written to Azure Blob storage. The logs show how many VMs aren't getting network traffic because of failed probe responses.

For the Cassandra cluster, the failover scenarios depend on the consistency levels used by the application and the number of replicas. For consistency levels and usage in Cassandra, see [Configuring data consistency](#) and [Cassandra: How many nodes are talked to with Quorum?](#) Data availability in Cassandra is determined by the consistency level used by the application and the replication mechanism. For replication in Cassandra, see [Data Replication in NoSQL Databases Explained](#).

Security considerations

Virtual networks are a traffic isolation boundary in Azure. VMs in one VNet can't communicate directly with VMs in a different VNet. VMs within the same VNet can communicate, unless you create [network security groups](#) (NSGs) to restrict traffic. For more information, see [Microsoft cloud services and network security](#).

For incoming Internet traffic, the load balancer rules define which traffic can reach the back end. However, load balancer rules don't support IP safe lists, so if you want to add certain public IP addresses to a safe list, add an NSG to the subnet.

DMZ. Consider adding a network virtual appliance (NVA) to create a DMZ between the Internet and the Azure virtual network. NVA is a generic term for a virtual appliance that can perform network-related tasks, such as firewall, packet

inspection, auditing, and custom routing. For more information, see [Implementing a DMZ between Azure and the Internet](#).

Encryption. Encrypt sensitive data at rest and use [Azure Key Vault](#) to manage the database encryption keys. Key Vault can store encryption keys in hardware security modules (HSMs). It's also recommended to store application secrets, such as database connection strings, in Key Vault.


DDoS protection. The Azure platform provides basic DDoS protection by default. This basic protection is targeted at protecting the Azure infrastructure as a whole. Although basic DDoS protection is automatically enabled, we recommend using [DDoS Protection Standard](#). Standard protection uses adaptive tuning, based on your application's network traffic patterns, to detect threats. This allows it to apply mitigations against DDoS attacks that might go unnoticed by the infrastructure-wide DDoS policies. Standard protection also provides alerting, telemetry, and analytics through Azure Monitor. For more information, see [Azure DDoS Protection: Best practices and reference architectures](#).

Deploy the solution


A deployment for this reference architecture is available on [GitHub](#).

Prerequisites

1. Clone, fork, or download the zip file for the [reference architectures](#) GitHub repository.
2. Install [Azure CLI 2.0](#).
3. Install [Node and NPM](#)
4. Install the [Azure building blocks](#) npm package.

bash	 Copy
npm install -g @mspnp/azure-building-blocks	


5. From a command prompt, bash prompt, or PowerShell prompt, sign into your Azure account as follows:

bash	 Copy
az login	

Deploy the solution using azbb

To deploy the Linux VMs for an N-tier application reference architecture, follow these steps:

1. Navigate to the `virtual-machines\n-tier-linux` folder for the repository you cloned in step 1 of the prerequisites above.
2. The parameter file specifies a default administrator user name and password for each VM in the deployment. Change these before you deploy the reference architecture. Open the `n-tier-linux.json` file and replace each **adminUsername** and **adminPassword** field with your new settings. Save the file.
3. Deploy the reference architecture using the **azbb** tool as shown below.

Azure CLI	 Copy
azbb -s <your subscription_id> -g <your resource_group_name> -l <azure region> -p n-tier-linux.json --deploy	

For more information on deploying this sample reference architecture using Azure Building Blocks, visit the [GitHub repository](#).

Next steps

- [Microsoft Learn module: Tour the N-tier architecture style](#)