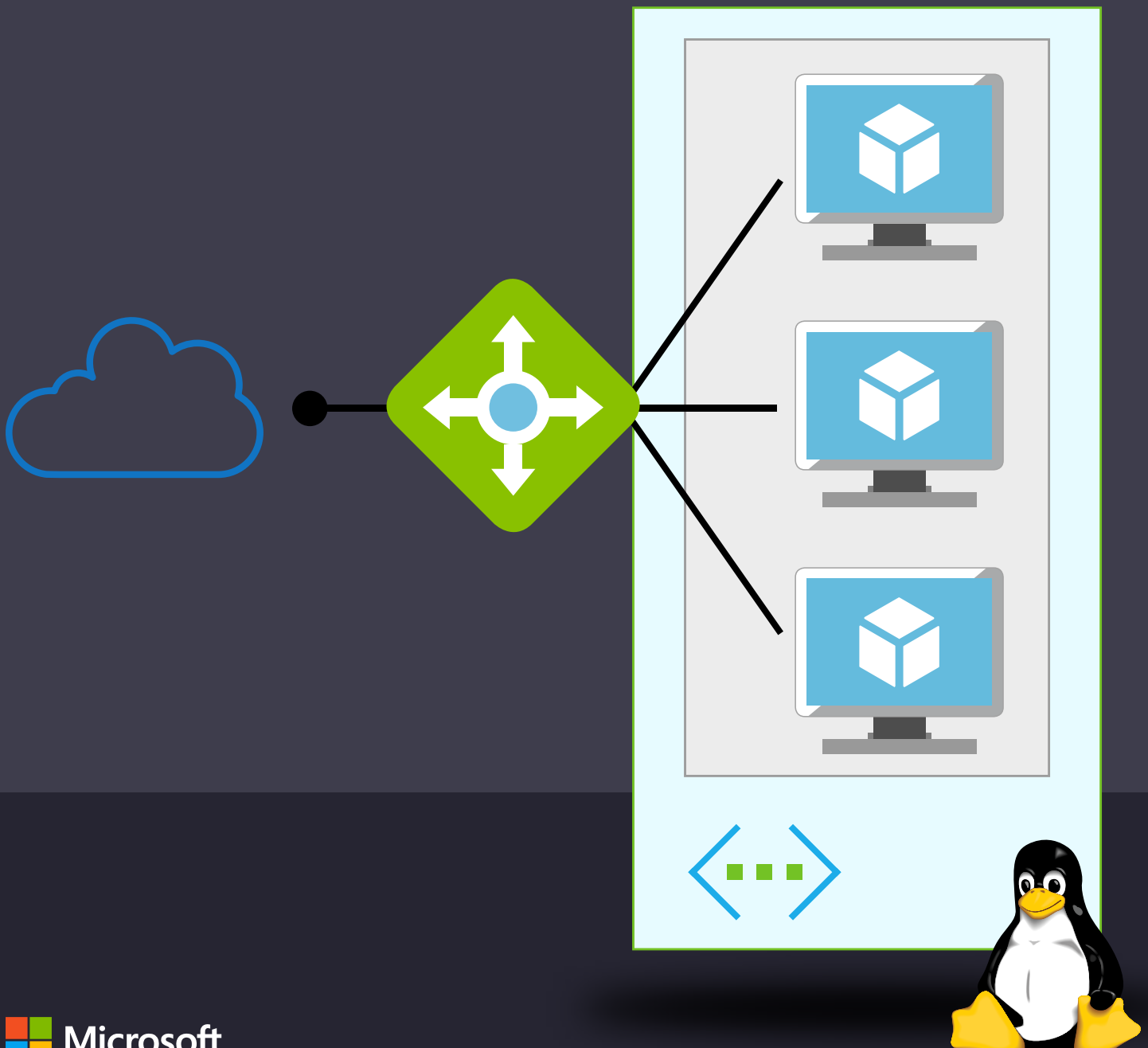
Microsoft Azure

Architectural Blueprints:

How to deploy Linux-based architectures in Azure



Microsoft

Hey IT Professionals, and Cloud Specialists!

Welcome to our guide! We created it to help you deploying Linux-based infrastructure in Microsoft Azure. Azure is supporting many Open Source technologies, incl. long list of Linux operating systems (CentOS, CoreOS, Debian, Oracle Linux, Red Hat Enterprise Linux, SUSE Linux Enterprise, openSUSE, and Ubuntu).

Before you'll start, we strongly suggest to start the free Azure trial. With this free account with 200$ for experiments for first 30-days, you'll be able to follow the instructions provided in this guide and start your first Linux-based Development infrastructure in the Cloud (BTW – with it you can try any Azure services, not only those mentioned later!).

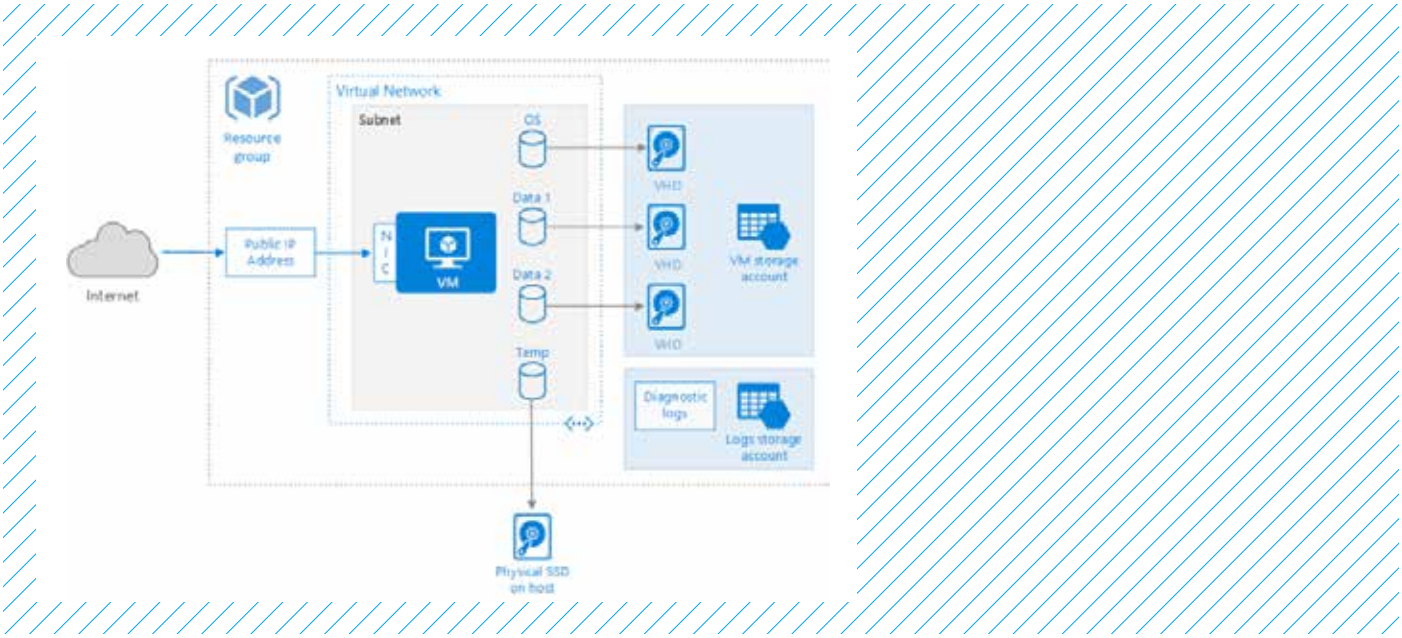
If you're interested in other technical content related to Open Source in the Cloud, feel free to visit our MSDN (with long list of free eBooks and technical webinars), and Channel 9 (with thousands of videos, full of tech-demos by Microsoft Engineers).

Thank you!
Your Azure Team

In this guide

Run a Linux VM on Azure

This reference architecture shows a set of proven practices for running a Linux virtual machine (VM) on Azure. It includes recommendations for provisioning the VM along with networking and storage components. This architecture can be used to run a single instance, and is the basis for more complex architectures such as N-tier applications.



Architecture

Provisioning a VM in Azure involves more moving parts than just the VM itself. There are compute, networking, and storage elements that you need to consider.

- **Resource group.** A resource group is a container that holds related resources. Create a resource group to hold the resources for this VM.
- **VM.** Azure supports running various popular Linux distributions, including CentOS, Debian, Red Hat Enterprise, Ubuntu, and FreeBSD. For more information, see Azure and Linux. You can provision a VM from a list of published images or from a virtual hard disk (VHD) file that you upload to Azure Blob storage.
- **OS disk.** The OS disk is a VHD stored in Azure Storage. That means it persists even if the host machine goes down. The OS disk is `/dev/sda1`.
- **Temporary disk.** The VM is created with a temporary disk. This disk is stored on a physical drive on the host machine. It is not saved in Azure Storage, and might be deleted during reboots and other VM lifecycle events. Use this disk only for temporary data, such as page or swap files. The temporary disk is `/dev/sdb1` and is mounted at `/mnt/resource` lub `/mnt`.
- **Data disks.** A data disk is a persistent VHD used for application data. Data disks are stored in Azure Storage, like the OS disk.
- **Virtual network (VNet) and subnet.** Every VM in Azure is deployed into a VNet that is further divided into subnets.
- **Public IP address.** A public IP address is needed to communicate with the VM—for example over SSH.
- **Network interface (NIC).** The NIC enables the VM to communicate with the virtual network.
- **Network security group (NSG).** The NSG is used to allow/deny network traffic to the subnet. You can associate an NSG with an individual NIC or with a subnet. If you associate it with a subnet, the NSG rules apply to all VMs in that subnet.
- **Diagnostics.** Diagnostic logging is crucial for managing and troubleshooting the VM.

You can download a Visio file of this architecture.

Note

Azure has two different deployment models: Resource Manager and classic. This article uses Resource Manager, which Microsoft recommends for new deployments.

Recommendations

This architecture shows the baseline recommendations for running a Linux VM in Azure. However, we don't recommend using a single VM for mission critical workloads, because it creates a single point of failure. For higher availability, deploy multiple VMs in an availability set. For more information, see Running multiple VMs on Azure.

VM recommendations

Azure offers many different virtual machine sizes, but we recommend the DS- and GS-series because these machine sizes support Premium Storage. Select one of these machine sizes unless you have a specialized workload such as high-performance computing. For details, see virtual machine sizes.

If you are moving an existing workload to Azure, start with the VM size that's the closest match to your on-premises servers. Then measure the performance of your actual workload with respect to CPU, memory, and disk input/output operations per second (IOPS), and adjust the size if needed. If you require multiple NICs for your VM, be aware that the maximum number of NICs is a function of the VM size.

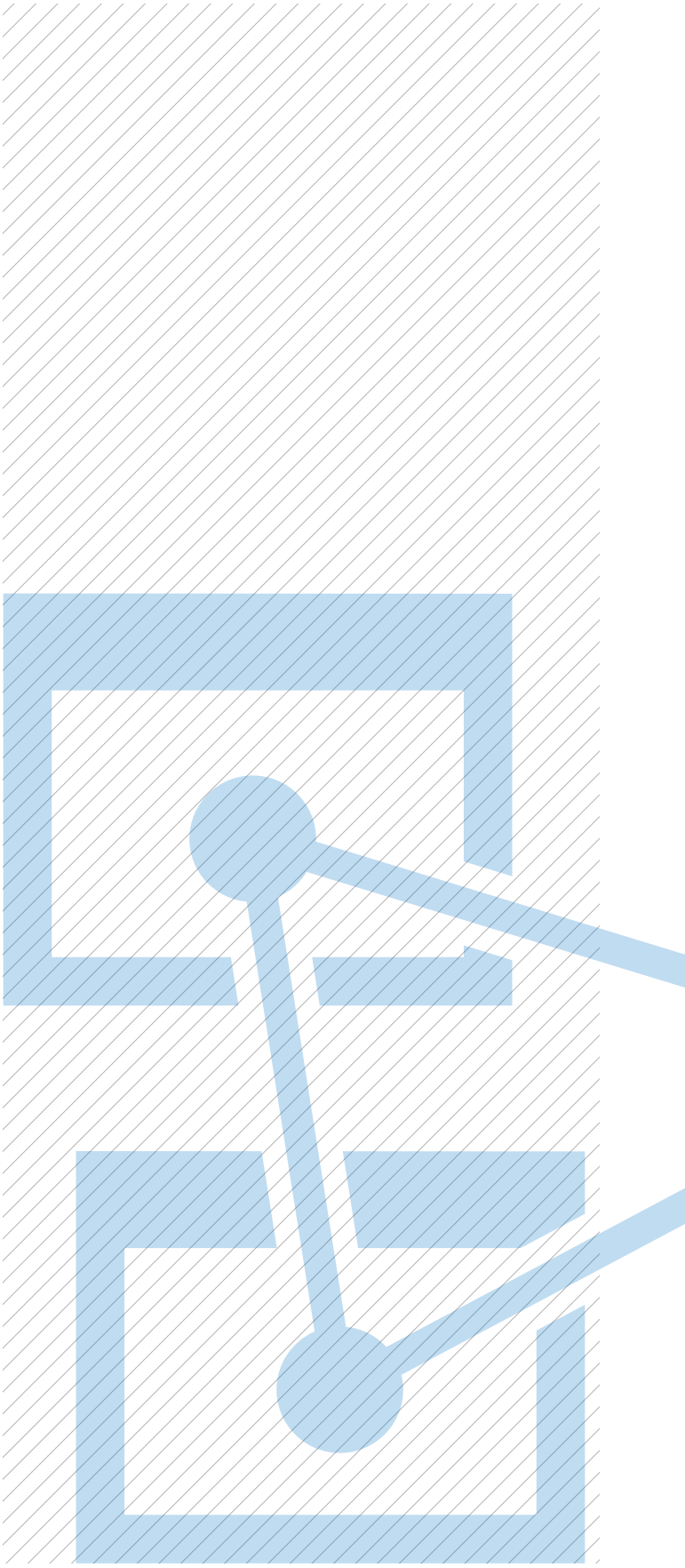
When you provision the VM and other resources, you must specify a region. Generally, choose a region closest to your internal users or customers. However, not all VM sizes may be available in all region. For details, see Services by region. To list the VM sizes available in a given region, run the following Azure command-line interface (CLI) command:

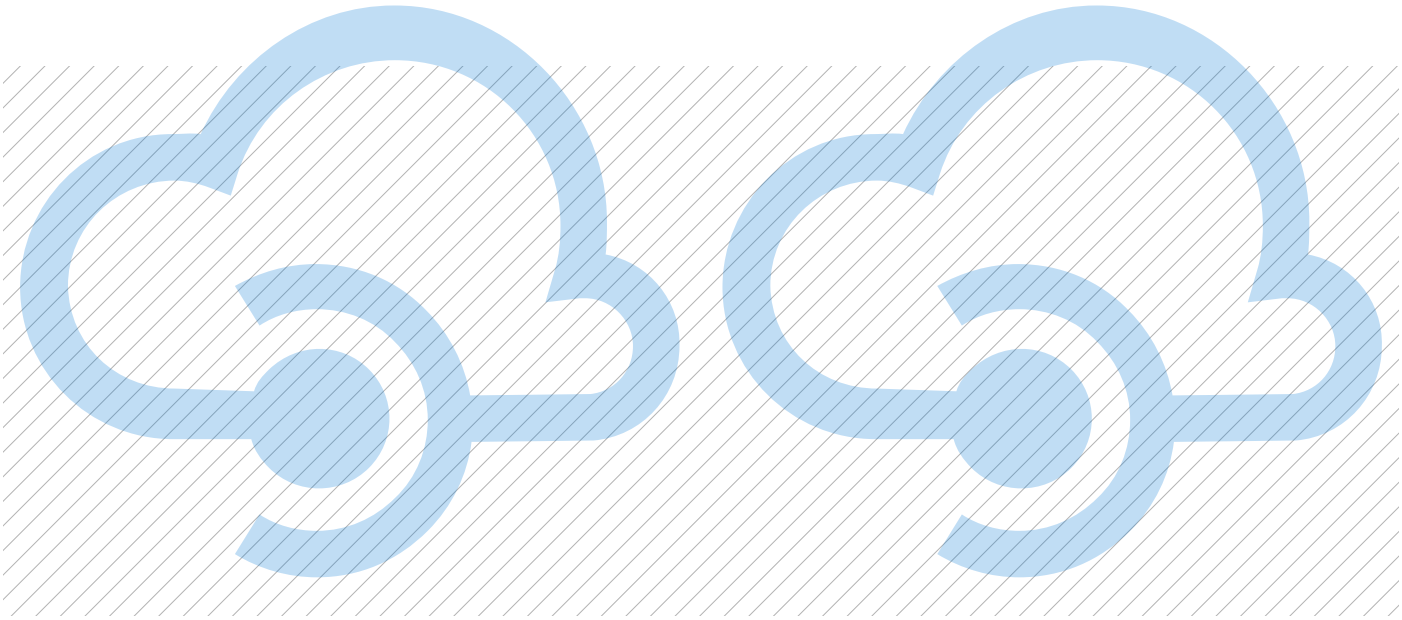
```
azure vm sizes --location <location>
```

For information about choosing a published VM image, see Select Linux VM images with the Azure CLI.

Disk and storage recommendations

For best disk I/O performance, we recommend Premium Storage, which stores data on solid-state drives (SSDs). Cost is based on the size of the provisioned disk. IOPS and throughput (that is, data transfer rate) also depend on disk size, so when you provision a disk, consider all three factors (capacity, IOPS, and throughput).





Important
We recommend the use of managed disks. Managed disks do not require a storage account. You simply specify the size and type of disk and it is deployed in a highly available way. Our reference architectures do not currently deploy managed disks but the template building blocks will be updated to deploy managed disks in version 2.

If you are not using managed disks, create separate Azure storage accounts for each VM to hold the virtual hard disks (VHDs) in order to avoid hitting the IOPS limits for storage accounts.
Add one or more data disks. When you create a VHD, it is unformatted. Log in to the VM to format the disk. In the Linux shell, data disks are displayed as `/dev/sdc`, `/dev/sdd`, and so on. You can run `lsblk` to list the block devices, including the disks. To use a data disk, create a partition and file system, and mount the disk. For example:

bat

```
# Create a partition.
sudo fdisk /dev/sdc    # Enter 'n' to partition, 'w' to write the change.

# Create a file system.
sudo mkfs -t ext3 /dev/sdc1

# Mount the drive.
sudo mkdir /data1
sudo mount /dev/sdc1 /data1
```

If you are not using managed disks and have a large number of data disks, be aware of the total I/O limits of the storage account. For more information, see virtual machine disk limits.
When you add a data disk, a logical unit number (LUN) ID is assigned to the disk. Optionally, you can specify the LUN ID — for example, if you're replacing a disk and want to retain the same LUN ID, or you have an application that looks for a specific LUN ID. However, remember that LUN IDs must be unique for each disk.
You may want to change the I/O scheduler to optimize for performance on SSDs, because the disks for VMs with premium storage accounts are SSDs. A common recommendation is to use the NOOP scheduler for SSDs, but you should use a tool such as iostat to monitor disk I/O performance for your particular workload.
For best performance, create a separate storage account to hold diagnostic logs. A standard locally redundant storage (LRS) account is sufficient for diagnostic logs.

Network recommendations
The public IP address can be dynamic or static. The default is dynamic.

- **Reserve a static IP address** if you need a fixed IP address that won't change — for example, if you need to create an A record in DNS, or need the IP address to be added to a safe list.
- **You can also create a fully qualified domain name (FQDN) for the IP address.** You can then register a CNAME record in DNS that points to the FQDN. For more information, see create a fully qualified domain name in the Azure portal.

All NSGs contain a set of default rules, including a rule that blocks all inbound Internet traffic. The default rules cannot be deleted, but other rules can override them. To enable Internet traffic, create rules that allow inbound traffic to specific ports — for example, port 80 for HTTP. To enable SSH, add a rule to the NSG that allows inbound traffic to TCP port 22.

Scalability considerations

To scale up or down, change the VM size.
To scale out horizontally, put two or more VMs into an availability set behind a load balancer. For details, see running multiple VMs on Azure.

Availability considerations

For higher availability, deploy multiple VMs in an availability set. This also provides a higher service level agreement (SLA).
Your VM may be affected by planned maintenance or unplanned maintenance. You can use VM reboot logs to determine whether a VM reboot was caused by planned maintenance.
VHDs are stored in Azure storage, and Azure storage is replicated for durability and availability.
To protect against accidental data loss during normal operations (for example, because of user error), you should also implement point-in-time backups, using blob snapshots or another tool.

Manageability considerations

Resource groups. Put tightly coupled resources that share the same life cycle into the same resource group. Resource groups allow you to deploy and monitor resources as a group, and roll up billing costs by resource group. You can also delete resources as a set, which is very useful for test deployments. Give resources meaningful names. That makes it easier to locate a specific resource and understand its role. See Recommended Naming Conventions for Azure Resources.

SSH. Before you create a Linux VM, generate a 2048-bit RSA public-private key pair. Use the public key file when you create the VM. For more information, see How to Use SSH with Linux and Mac on Azure.

VM diagnostics. Enable monitoring and diagnostics, including basic health metrics, diagnostics infrastructure logs, and boot diagnostics. Boot diagnostics can help you diagnose boot failure if your VM gets into a nonbootable state. For more information, see Enable monitoring and diagnostics.

The following CLI command enables diagnostics:

```
azure vm enable-diag <resource-group> <vm-name>
```

Stopping a VM. Azure makes a distinction between "stopped" and "deallocated" states. You are charged when the VM status is stopped, but not when the VM is deallocated.

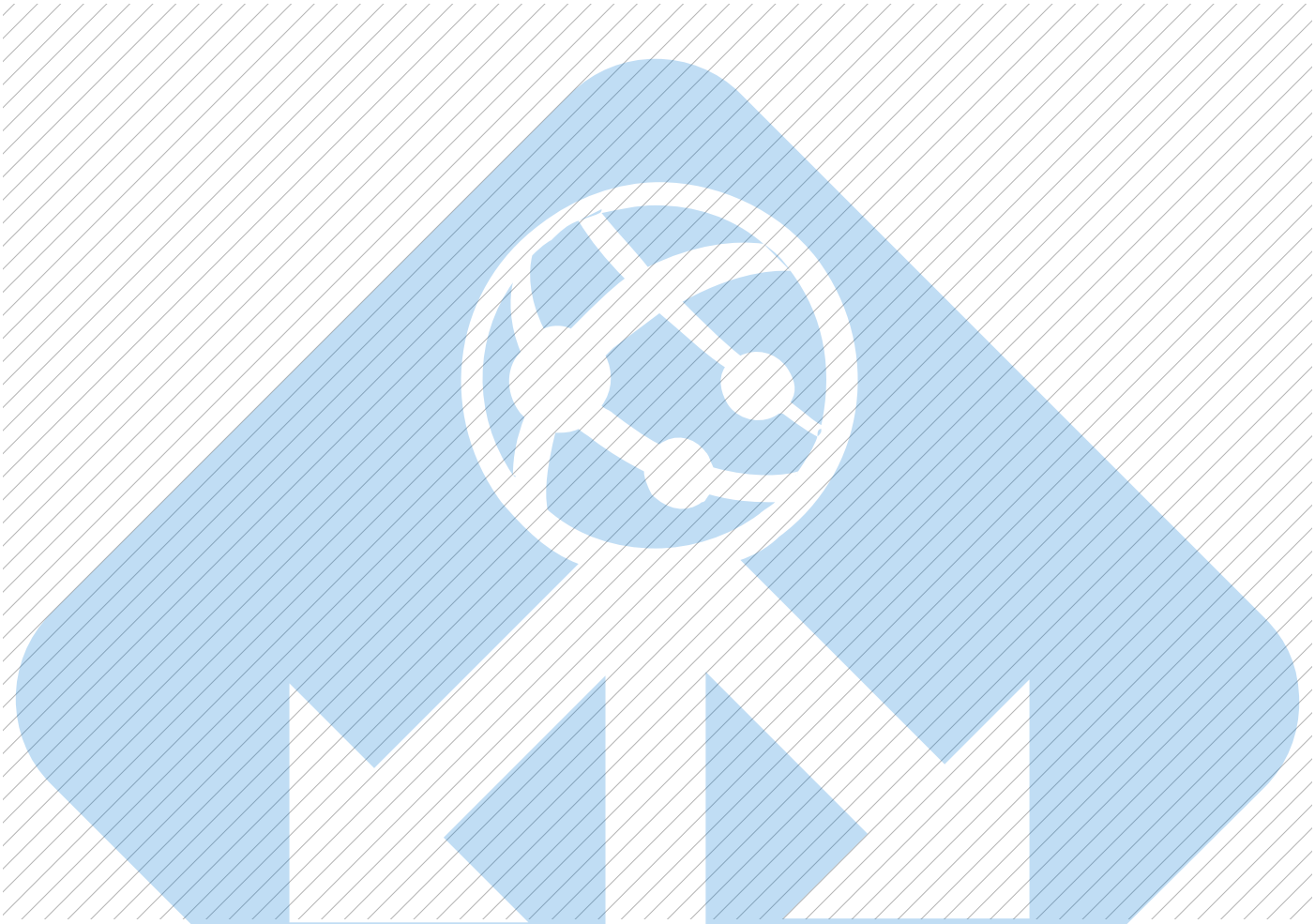
Use the following CLI command to deallocate a VM:

```
azure vm deallocate <resource-group> <vm-name>
```

In the Azure portal, **the Stop** button deallocates the VM. However, if you shut down through the OS while logged in, the VM is stopped but not deallocated, so you will still be charged.

Deleting a VM. If you delete a VM, the VHDs are not deleted. That means you can safely delete the VM without losing data. However, you will still be charged for storage. To delete the VHD, delete the file from Blob storage.

To prevent accidental deletion, use a resource lock to lock the entire resource group or lock individual resources, such as the VM.



Security considerations

Automate OS updates by using the OSPatching VM extension. Install this extension when you provision the VM. You can specify how often to install patches and whether to reboot after patching.

Use role-based access control (RBAC) to control access to the Azure resources that you deploy. RBAC lets you assign authorization roles to members of your DevOps team. For example, the Reader role can view Azure resources but not create, manage, or delete them. Some roles are specific to particular Azure resource types. For example, the Virtual Machine Contributor role can restart or deallocate a VM, reset the administrator password, create a VM, and so forth. Other built-in RBAC roles that might be useful for this architecture include DevTest Labs User and Network Contributor.

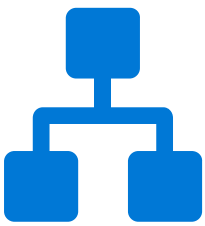
A user can be assigned to multiple roles, and you can create custom roles for even more fine-grained permissions.

Note

RBAC does not limit the actions that a user logged into a VM can perform. Those permissions are determined by the account type on the guest OS.

Use audit logs to see provisioning actions and other VM events.

Consider Azure Disk Encryption if you need to encrypt the OS and data disks.



Deploy the solution

A deployment for this architecture is available on GitHub. It includes a VNet, NSG, and a single VM. To deploy the architecture, follow these steps:

1. Click the button below:

Deploy to Azure

2. Once the link has opened in the Azure portal, you must enter values for some of the settings:

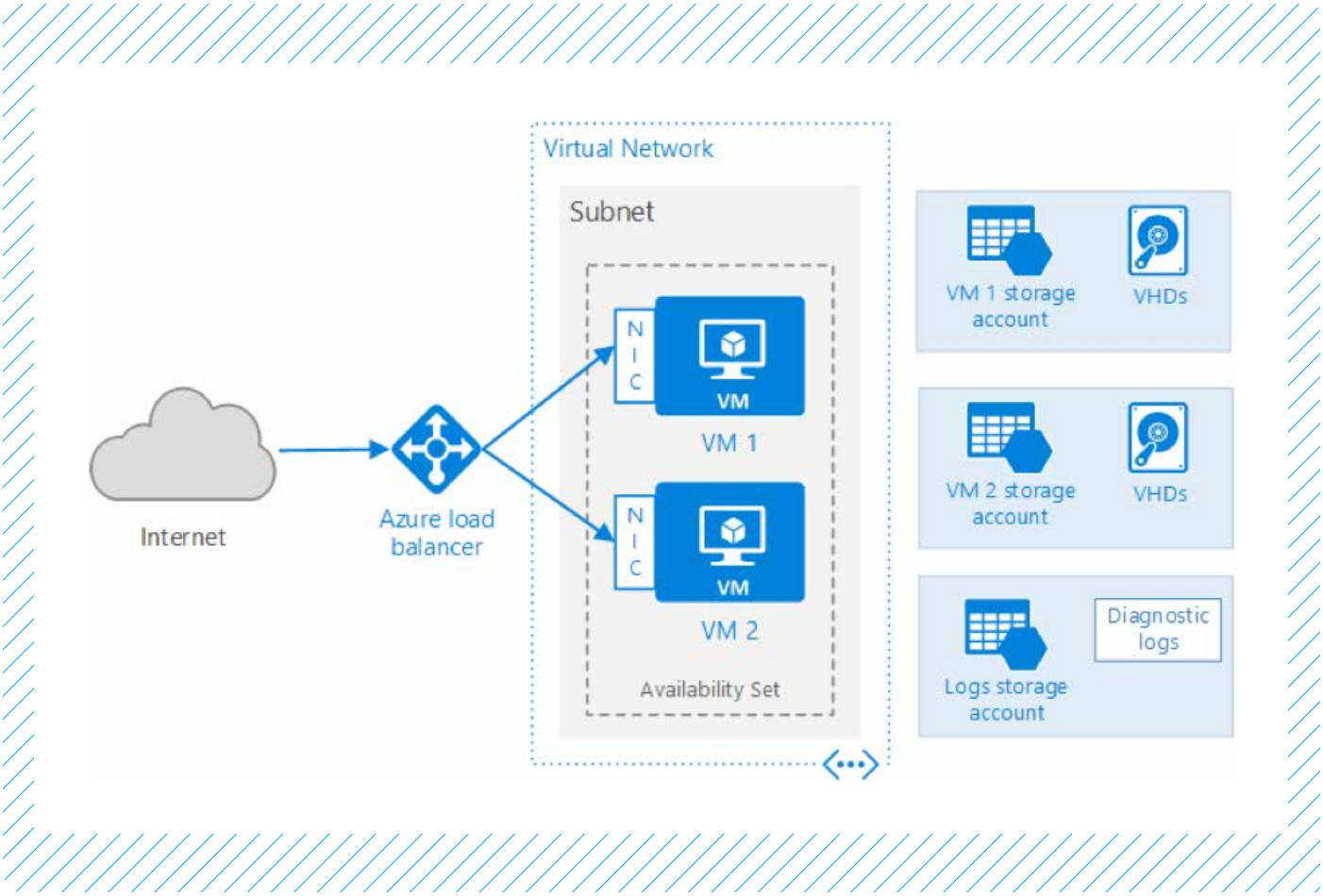
- The **Resource group** name is already defined in the parameter file, so select **Create New** and enter `ra-single-vm-rg` in the text box.
- Select the region from the **Location** drop down box.
- Do not edit the **Template Root Uri** or the **Parameter Root Uri** text boxes.
- Select linux in the **Os Type** drop down box.
- Review the terms and conditions, then click the **I agree to the terms and conditions stated above** checkbox.
- Click on the **Purchase** button.

3. Wait for the deployment to complete.

4. The parameter files include a hard-coded administrator user name and password, and it is strongly recommended that you immediately change both. Click on the VM named `ra-single-vm0` in the Azure portal. Then, click on **Reset password** in the **Support + troubleshooting** section. Select **Reset password** in the **Mode** dropdown box, then select a new **User name** and **Password**. Click the **Update** button to persist the new user name and password.

Run load-balanced VMs for scalability and availability

This reference architecture shows a set of proven practices for running several Linux virtual machines (VMs) behind a load balancer, to improve availability and scalability. This architecture can be used for any stateless workload, such as a web server, and is a building block for deploying N-tier applications.



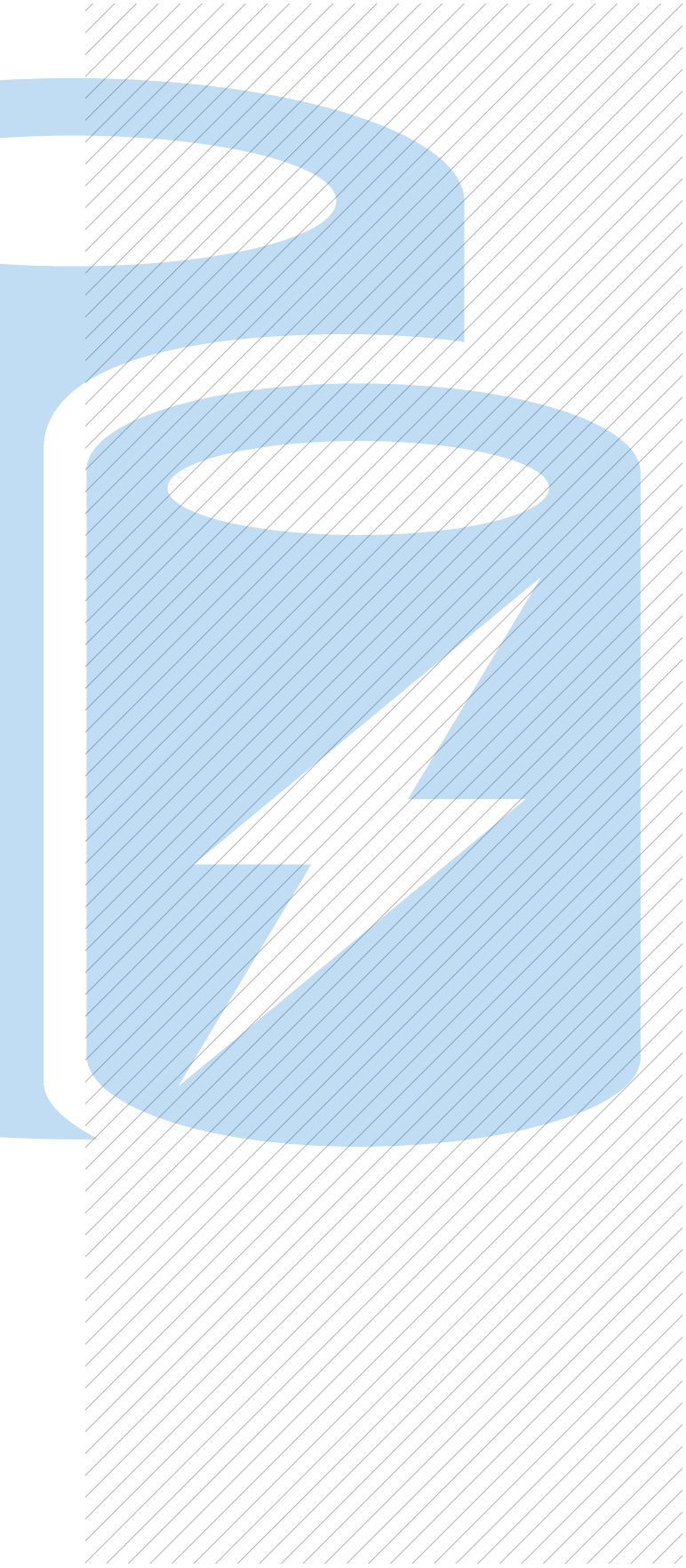
Architecture

This architecture builds on the one shown in Run a Linux VM on Azure. The recommendations there also apply to this architecture.

In this architecture, a workload is distributed across several VM instances. There is a single public IP address, and Internet traffic is distributed to the VMs using a load balancer. This architecture can be used for a single-tier application, such as a stateless web application or storage cluster. It's also a building block for N-tier applications.

The architecture has the following components:

- **Availability set.** The availability set contains the VMs. This makes the VMs eligible for the availability service level agreement (SLA) for Azure VMs. For the SLA to apply, you need a minimum of two VMs in the same availability set.
- **Virtual network (VNet) and subnet.** Every VM in Azure is deployed into a VNet that is further divided into subnets.
- **Azure Load Balancer.** The load balancer distributes incoming Internet requests to the VM instances. The load balancer includes some related resources:
- **Public IP address.** A public IP address is needed for the load balancer to receive Internet traffic.
- **Front-end configuration.** Associates the public IP address with the load balancer.
- **Back-end address pool.** Contains the network interfaces (NICs) for the VMs that will receive the incoming traffic.



- **Load balancer rules.** Used to distribute network traffic among all the VMs in the back-end address pool.
- **Network address translation (NAT) rules.** Used to route traffic to a specific VM. For example, to enable remote desktop protocol (RDP) to the VMs, create a separate NAT rule for each VM.
- **Network interfaces (NICs).** Each VM has a NIC to connect to the network.
- **Storage.** If you are not using managed disks, storage accounts hold the VM images and other file-related resources, such as VM diagnostic data captured by Azure.

You can download a Visio file of this architecture.

Note
Azure has two different deployment models: Resource Manager and classic. This article uses Resource Manager, which Microsoft recommends for new deployments.

Recommendations

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

Availability set recommendations

Create at least two VMs in the availability set, to support the availability SLA for Azure VMs. The Azure load balancer also requires that load-balanced VMs belong to the same availability set.

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.

Network recommendations

Place the VMs within the same subnet. Do not expose the VMs directly to the Internet, but instead give each VM a private IP address. Clients connect using the public IP address of the load balancer.

Load balancer recommendations
Add all VMs in the availability set to the back-end address pool of the 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. In that way, client requests are distributed across all the VMs.
To route traffic to a specific VM, use NAT rules. For example, to enable RDP to the VMs, create a separate NAT rule for each VM. Each rule should map a distinct port number to port 3389, the default port for RDP. For example, use port 50001 for "VM1," port 50002 for "VM2," and so on. Assign the NAT rules to the NICs on the VMs.

Storage account recommendations
Create separate Azure storage accounts for each VM to hold the virtual hard disks (VHDs), in order to avoid hitting the input/output operations per second (IOPS) limits for storage accounts.

Important
We recommend the use of managed disks. Managed disks do not require a storage account. You simply specify the size and type of disk and it is deployed in a highly available way. Our reference architectures do not currently deploy managed disks but the template building blocks will be updated to deploy managed disks in version 2.

Create one storage account for diagnostic logs. This storage account can be shared by all the VMs.

Scalability considerations

To scale out, provision additional VMs and put them in the load balancer's back-end address pool.

Tip
When you add a new VM to an availability set, make sure to create a NIC for the VM, and add the NIC to the back-end address pool on the load balancer. Otherwise, Internet traffic won't be routed to the new VM.

VM scale sets

Another option for scaling is to use a virtual machine scale set. VM scale sets help you to deploy and manage a set of identical VMs. Scale sets support autoscaling 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.

Currently, scale sets do not support data disks. The options for storing data are Azure File storage, the OS drive, the temp drive, or an external store such as Azure Storage. By default, scale sets use "overprovisioning," which means the scale set initially provisions more VMs than you ask for, then deletes the extra VMs. This improves the overall success rate when provisioning the VMs. If you are not using managed disks, we recommend no more than 20 VMs per storage account with overprovisioning enabled, or no more than 40 VMs with overprovisioning disabled.

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

- Use extensions to configure the VM after it is provisioned. 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 additional considerations, see Designing VM Scale Sets For Scale.

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

Availability considerations

The availability set makes your application more resilient to both planned and unplanned maintenance events.

- Planned maintenance occurs when Microsoft updates the underlying platform, sometimes causing VMs to be restarted. Azure makes sure the VMs in an availability set are not all restarted at the same time. At least one is kept running while others are restarting.
- Unplanned maintenance happens if there is a hardware failure. Azure makes sure that VMs in an availability set are provisioned across more than one server rack. This helps to reduce the impact of hardware failures, network outages, power interruptions, and so on.

For more information, see Manage the availability of Linux virtual machines. The following video also has a good overview of availability sets: How Do I Configure an Availability Set to Scale VMs.

Warning
Make sure to configure the availability set when you provision the VM. Currently, there is no way to add a Resource Manager VM to an availability set after the VM is provisioned.

The load balancer uses health probes to monitor the availability of VM instances. If a probe cannot reach an instance within a timeout period, the load balancer stops sending traffic to that VM. However, 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 end-point. 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 in any firewall policies or network security group (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 on the back end are not receiving network traffic due to failed probe responses.

Manageability considerations

With multiple VMs, it is important to automate processes so they are reliable and repeatable. You can use Azure Automation to automate deployment, OS patching, and other tasks. Azure Automation is an automation service based on Windows Powershell that can be used for this. Example automation scripts are available from the Runbook Gallery on TechNet.

Security considerations

Virtual networks are a traffic isolation boundary in Azure. VMs in one VNet cannot communicate directly to 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.

Deploy the solution

A deployment for this architecture is available on GitHub. It includes a VNet, NSG, load balancer, and two VMs. It can be deployed with either Windows or Linux VMs. To deploy the architecture, follow these steps:

1. Click the button below:

Deploy to Azure

2. Once the link has opened in the Azure portal, you must enter values for some of the settings:

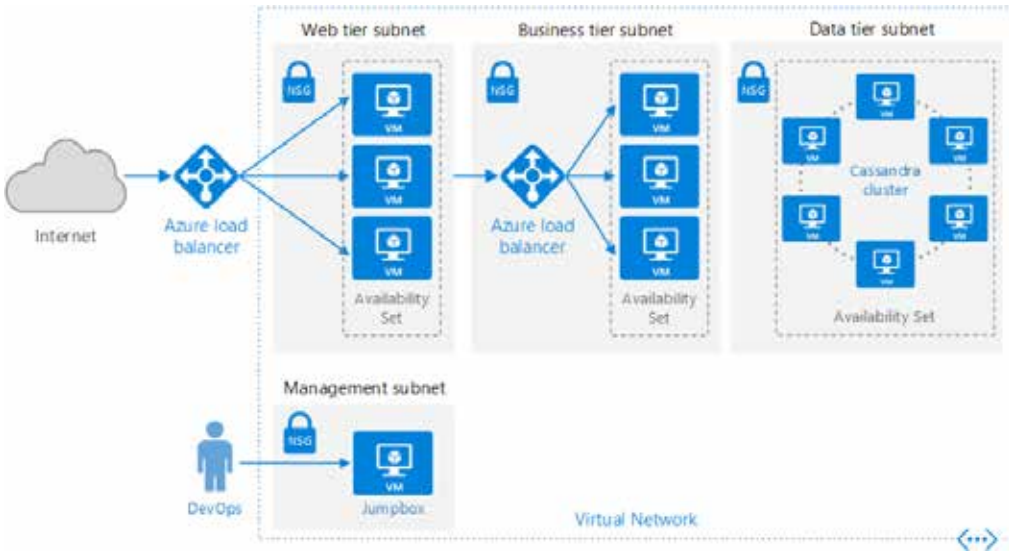
- The **Resource group** name is already defined in the parameter file, so select **Create new** and enter `ra-multi-vm-rg` in the text box.
- Select the region from the **Location** drop down box.
- Do not edit the **Template Root Uri** or the **Parameter Root Uri** text boxes.
- Select either **windows** or **linux** in the **Os Type** drop down box.
- Review the terms and conditions, then click the **I agree to the terms and conditions stated above checkbox**.
- Click the **Purchase** button.

3. Wait for the deployment to complete.

4. The parameter files include a hard-coded administrator user name and password, and it is strongly recommended that you immediately change both. Click the VM named `ra-multi-vm1` in the Azure portal. Then, click **Reset password** in the **Support + troubleshooting** blade. Select Reset **password** in the **Mode** dropdown box, then select a new **User name** and **Password**. Click the **Update** button to save the new user name and password. Repeat for the VM named `ra-multi-vm2`.

Run Linux VMs for an N-tier application

This reference architecture shows a set of proven practices for running Linux virtual machines (VMs) for an N-tier application.



Architecture

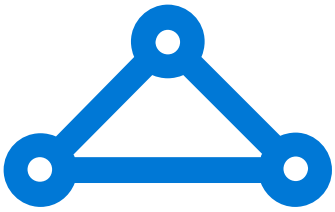
There are many ways to implement an N-tier architecture. The diagram shows a typical 3-tier web application. This architecture builds on Run load-balanced VMs for scalability and availability. The web and business tiers use load-balanced VMs.

- **Availability sets.** Create an availability set for each tier, and provision at least two VMs in each tier. This makes the VMs eligible for a higher service level agreement (SLA) for VMs.
- **Subnets.** Create a separate subnet for each tier. Specify the address range and subnet mask using CIDR notation.
- **Load balancers.** Use an Internet-facing 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.
- **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 permit secure shell (SSH) traffic.
- **Monitoring.** Monitoring software such as Nagios, Zabbix, or Icinga can give you insight into response time, VM uptime, and the overall health of your system. Install the monitoring software on a VM that's placed in a separate management subnet.
- **NSGs.** Use network security groups (NSGs) to restrict network traffic within the VNet. For example, in the 3-tier architecture shown here, the database tier does not accept traffic from the web front end, only from the business tier and the management subnet.
- **Apache Cassandra database.** Provides high availability at the data tier, by enabling replication and failover.

You can download a Visio file of this architecture.

Note

Azure has two different deployment models: Resource Manager and classic. This article uses Resource Manager, which Microsoft recommends for new deployments.



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 does not 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.

For each subnet, specify the address space for the subnet in CIDR notation. For example, '10.0.0.0/24' creates a range of 256 IP addresses. VMs can use 251 of these; five are reserved. Make sure the address ranges don't overlap across subnets. See the Virtual Network FAQ.

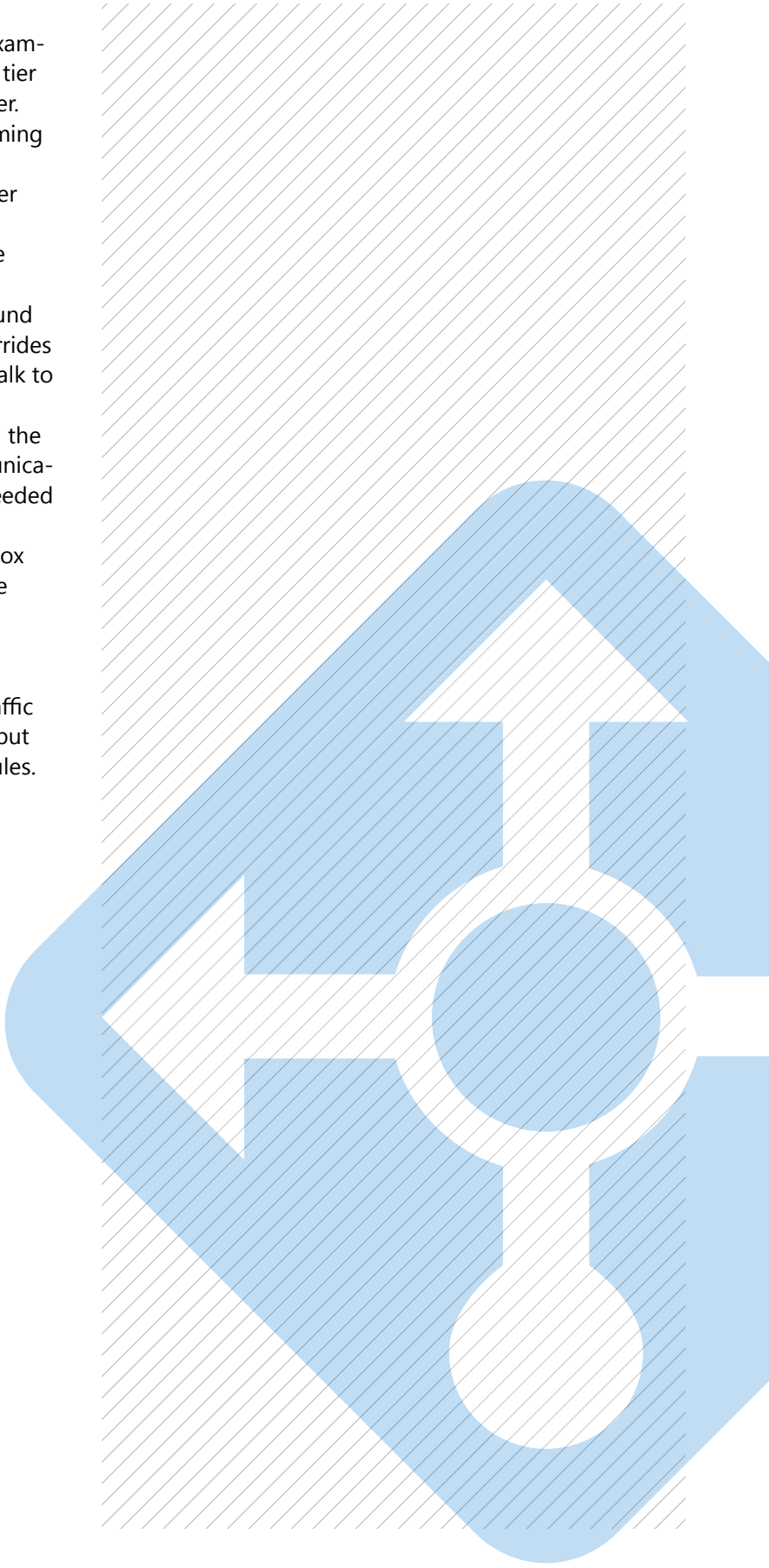
Network security groups

Use NSG rules to restrict traffic between tiers. For example, in the 3-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. Create an NSG and associate it to the database tier subnet.
2. Add a rule that denies all inbound traffic from the VNet. (Use the `VIRTUAL_NETWORK` tag in the rule.)
3. Add a rule with a higher priority that allows inbound traffic from the business tier subnet. This rule overrides the previous rule, and allows the business tier to talk to the database tier.
4. Add a rule that allows inbound traffic from within the database tier subnet itself. This rule allows communication between VMs in the database tier, which is needed for database replication and failover.
5. Add a rule that allows SSH traffic from the jumpbox subnet. This rule lets administrators connect to the database tier from the jumpbox.

Note

An NSG has default rules that allow any inbound traffic from within the VNet. These rules can't be deleted, but you can override them by creating higher-priority rules.



Load balancers
The external load balancer distributes Internet traffic to the web tier. Create a public IP address for this load balancer. See Creating an Internet-facing load balancer.
The internal load balancer distributes network traffic from the web tier to the business tier. To give this load balancer a private IP address, create a frontend IP configuration and associate it with the subnet for the business tier. See Get started creating an Internal load balancer.

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.

Jumpbox
The jumpbox will have minimal performance requirements, so select a small VM size for the jumpbox such as Standard A1.
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.
Do not 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.
To secure the jumpbox, create an NSG and apply it to the jumpbox subnet. Add an NSG rule that allows SSH connections only from a safe set of public IP addresses. The NSG can be attached either to the subnet or to the jumpbox NIC. In this case, we recommend attaching it to the NIC, so SSH traffic is permitted only to the jumpbox, even if you add other VMs to the same subnet.
Configure the NSGs for the other subnets to allow SSH traffic from the management subnet.

Availability considerations

Put each tier or VM role into a separate availability set. At the database tier, having multiple VMs does not automatically translate into a highly available database. For a relational database, you will typically need to use replication and failover to achieve high availability.

If you need higher availability than the Azure SLA for VMs provides, replicate the application across two regions and use Azure Traffic Manager for failover. For more information, see Run Linux VMs in multiple regions for high availability.

Security considerations

Consider adding a network virtual appliance (NVA) to create a DMZ between the public 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.

Scalability considerations

The load balancers distribute network traffic to the web and business tiers. Scale horizontally by adding new VM instances. Note that you can scale the web and business tiers independently, based on load. To reduce possible complications caused by the need to maintain client affinity, the VMs in the web tier should be stateless. The VMs hosting the business logic should also be stateless.

Manageability considerations

Simplify management of the entire system by using centralized administration tools such as Azure Automation, Microsoft Operations Management Suite, Chef, or Puppet. These tools can consolidate diagnostic and health information captured from multiple VMs to provide an overall view of the system.

Deploy the solution

A deployment for this architecture is available on GitHub. The architecture is deployed in three stages. To deploy the architecture, follow these steps:

1. Click the button below:

Deploy to Azure

2. Once the link has opened in the Azure portal, enter the follow values:

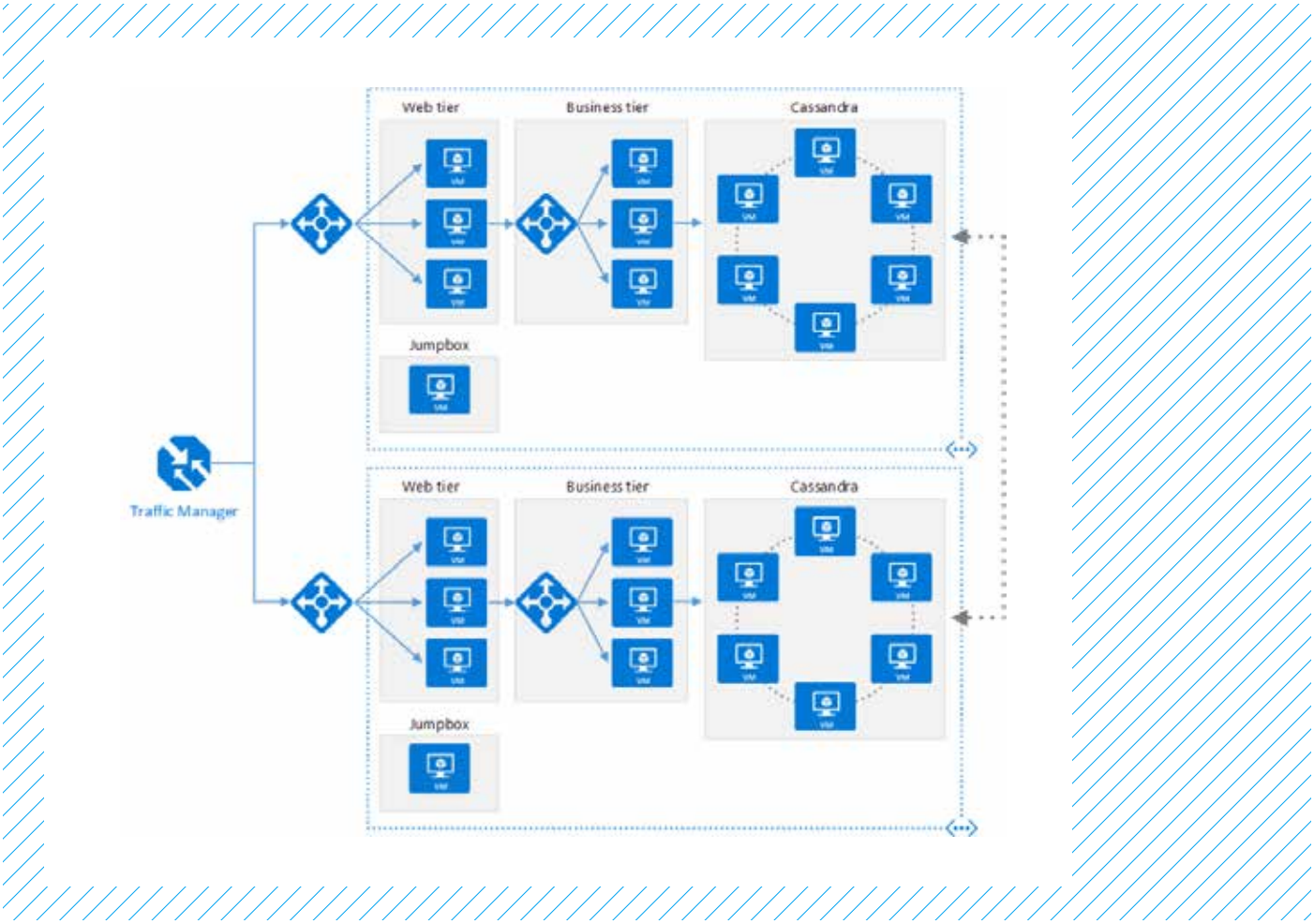
- The **Resource group** name is already defined in the parameter file, so select **Create New** and enter `ra-ntier-cassandra-rg` in the text box.
- Select the region from the **Location** drop down box.
- Do not edit the **Template Root Uri** or the **Parameter Root Uri** text boxes.
- Review the terms and conditions, then click the **I agree to the terms and conditions stated above** checkbox.
- Click on the **Purchase** button.

3. Check Azure portal notification for a message the deployment is complete.

4. The parameter files include a hard-coded administrator user names and passwords, and it is strongly recommended that you immediately change both on all the VMs. Click on each VM in the Azure portal then click on **Reset password** in the **Support + troubleshooting** blade. Select **Reset password** in the **Mode** dropdown box, then select a new **User name** and **Password**. Click the **Update** button to persist the new user name and password.

Run Linux VMs in multiple regions for high availability

This reference architecture shows a set of proven practices for running an N-tier application in multiple Azure regions, in order to achieve availability and a robust disaster recovery infrastructure.



Architecture

This architecture builds on the one shown in Run Linux VMs for an N-tier application.

- **Primary and secondary regions.** Use two regions to achieve higher availability. One is the primary region. The other region is for failover.
- **Azure Traffic Manager.** Traffic Manager routes incoming requests to one of the regions. During normal operations, it routes requests to the primary region. If that region becomes unavailable, Traffic Manager fails over to the secondary region. For more information, see the section Traffic Manager configuration.
- **Resource groups.** Create separate resource groups for the primary region, the secondary region, and for Traffic Manager. This gives you the flexibility to manage each region as a single collection of resources. For example, you could redeploy one region, without taking down the other one. Link the resource groups, so that you can run a query to list all the resources for the application.
- **VNets.** Create a separate VNet for each region. Make sure the address spaces do not overlap.
- **Apache Cassandra.** Deploy Cassandra in data centers across Azure regions for high availability. Within each region, nodes are configured in rack-aware mode with fault and upgrade domains, for resiliency inside the region.

You can download a Visio file of this architecture.

Recommendations

A multi-region architecture can provide higher availability than deploying to a single region. If a regional outage affects the primary region, you can use Traffic Manager to fail over to the secondary region. This architecture can also help if an individual subsystem of the application fails.

There are several general approaches to achieving high availability across regions:

- Active/passive with hot standby. Traffic goes to one region, while the other waits on hot standby. Hot standby means the VMs in the secondary region are allocated and running at all times.
- Active/passive with cold standby. Traffic goes to one region, while the other waits on cold standby. Cold standby means the VMs in the secondary region are not allocated until needed for failover. This approach costs less to run, but will generally take longer to come online during a failure.
- Active/active. Both regions are active, and requests are load balanced between them. If one region becomes unavailable, it is taken out of rotation.

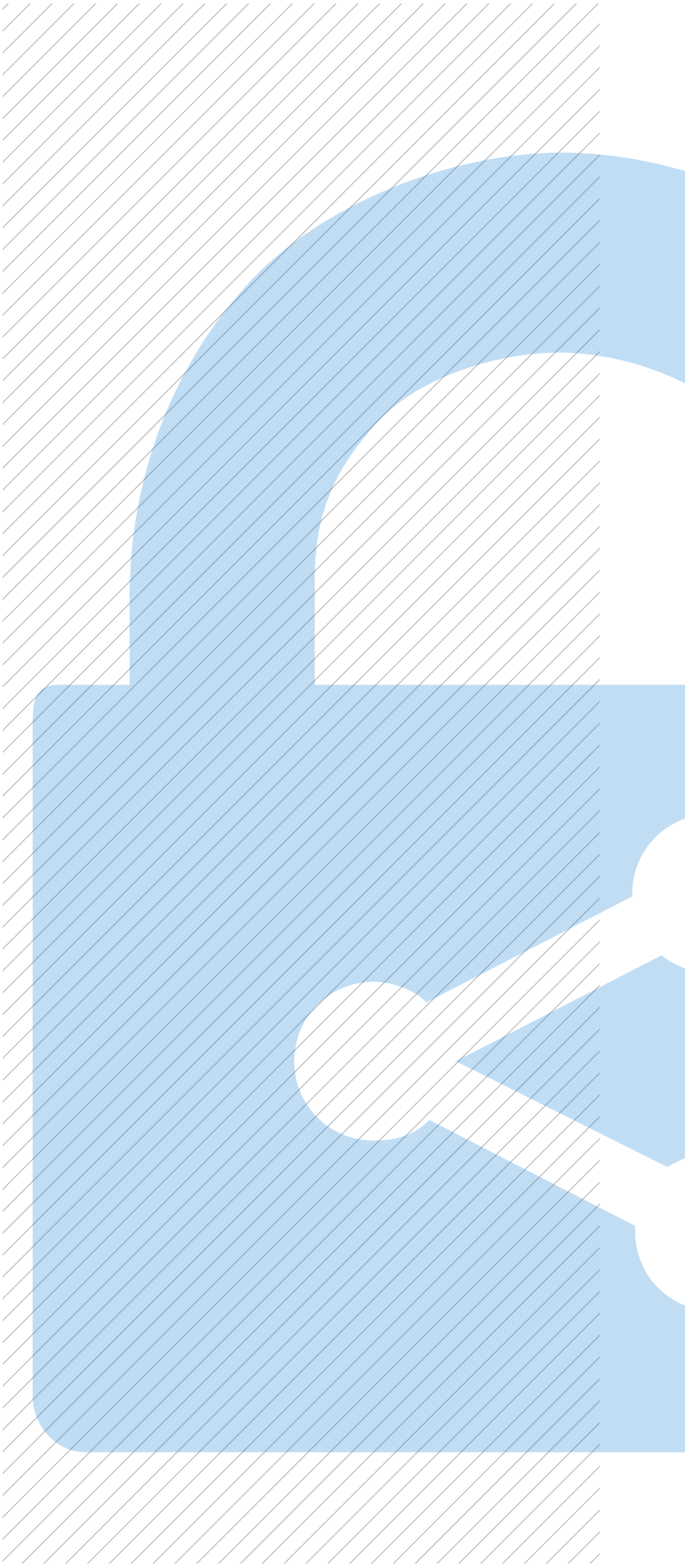
This architecture focuses on active/passive with hot standby, using Traffic Manager for failover. Note that you could deploy a small number of VMs for hot standby and then scale out as needed.

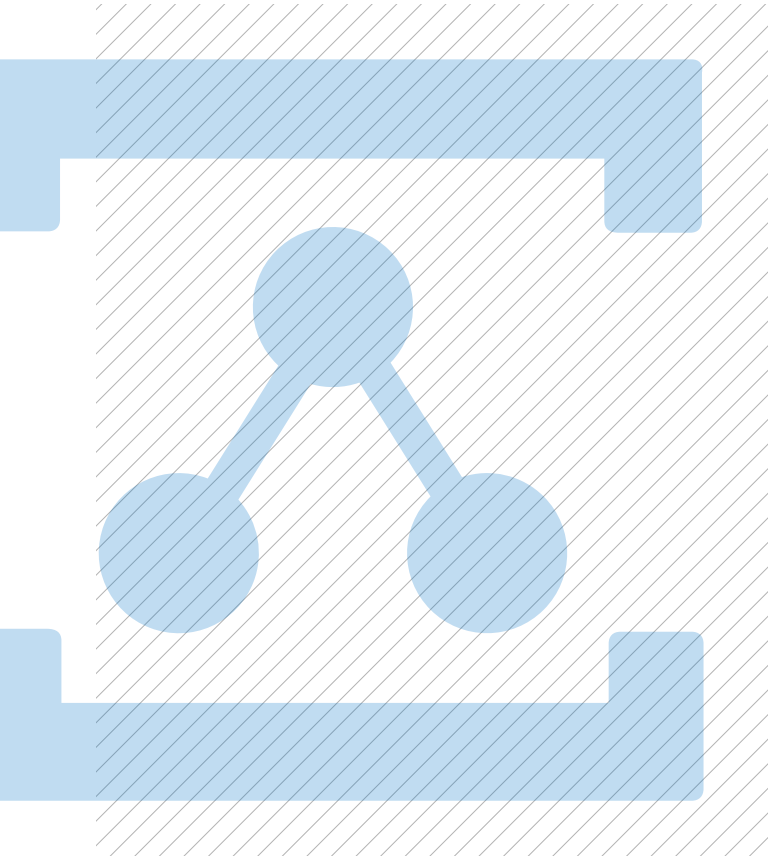
Regional pairing

Each Azure region is paired with another region within the same geography. In general, choose regions from the same regional pair (for example, East US 2 and US Central). Benefits of doing so include:

- If there is a broad outage, recovery of at least one region out of every pair is prioritized.
- Planned Azure system updates are rolled out to paired regions sequentially, to minimize possible downtime.
- Pairs reside within the same geography, to meet data residency requirements.

However, make sure that both regions support all of the Azure services needed for your application (see Services by region). For more information about regional pairs, see Business continuity and disaster recovery (BCDR): Azure Paired Regions.





Traffic Manager configuration

Consider the following points when configuring Traffic Manager:

- **Routing.** Traffic Manager supports several routing algorithms. For the scenario described in this article, use priority routing (formerly called failover routing). With this setting, Traffic Manager sends all requests to the primary region, unless the primary region becomes unreachable. At that point, it automatically fails over to the secondary region. See Configure Failover routing method.
- **Health probe.** Traffic Manager uses an HTTP (or HTTPS) probe to monitor the availability of each region. The probe checks for an HTTP 200 response for a specified URL path. As a best practice, create an endpoint that reports the overall health of the application, and use this endpoint for the health probe. Otherwise, the probe might report a healthy endpoint when critical parts of the application are actually failing. For more information, see Health Endpoint Monitoring Pattern.

When Traffic Manager fails over there is a period of time when clients cannot reach the application. The duration is affected by the following factors:

- The health probe must detect that the primary region has become unreachable.
- DNS servers must update the cached DNS records for the IP address, which depends on the DNS time-to-live (TTL). The default TTL is 300 seconds (5 minutes), but you can configure this value when you create the Traffic Manager profile.

For details, see About Traffic Manager Monitoring.

If Traffic Manager fails over, we recommend performing a manual failback rather than implementing an automatic failback. Otherwise, you can create a situation where the application flips back and forth between regions. Verify that all application subsystems are healthy before failing back.

Note that Traffic Manager automatically fails back by default. To prevent this, manually lower the priority of the primary region after a failover event. For example, suppose the primary region is priority 1 and the secondary is priority 2. After a failover, set the primary region to priority 3, to prevent automatic failback. When you are ready to switch back, update the priority to 1.

The following Azure CLI command updates the priority:

```
batCopy

azure network traffic-manager  endpoint set --r
source-group <resource-group> --profile-name <pr
file>
   --name <traffic-manager-name> --type AzureEn
points --priority 3
```

Another approach is to temporarily disable the endpoint until you are ready to fail back:

```
batCopy

azure network traffic-manager  endpoint set --r
source-group <resource-group> --profile-name <pr
file>
   --name <traffic-manager-name> --type AzureEn
points --status Disabled
```

Depending on the cause of a failover, you might need to redeploy the resources within a region. Before failing back, perform an operational readiness test. The test should verify things like:

- **VMs are configured correctly.** (All required software is installed, IIS is running, and so on.)
- **Application subsystems are healthy.**
- **Functional testing.** (For example, the database tier is reachable from the web tier.)

Cassandra deployment across multiple regions

Cassandra data centers are a group of related data nodes that are configured together within a cluster for replication and workload segregation.

We recommend DataStax Enterprise for production use. For more information on running DataStax in Azure, see DataStax Enterprise Deployment Guide for Azure. The following general recommendations apply to any Cassandra edition:

- Assign a public IP address to each node. This enables the clusters to communicate across regions using the Azure backbone infrastructure, providing high throughput at low cost.
- Secure nodes using the appropriate firewall and network security group (NSG) configurations, allowing traffic only to and from known hosts, including clients and other cluster nodes. Note that Cassandra uses different ports for communication, OpsCenter, Spark, and so forth. For port usage in Cassandra, see Configuring firewall port access.
- Use SSL encryption for all client-to-node and node-to-node communications.
- Within a region, follow the guidelines in Cassandra recommendations.

Availability considerations

With a complex N-tier app, you may not need to replicate the entire application in the secondary region. Instead, you might just replicate a critical subsystem that is needed to support business continuity.

Traffic Manager is a possible failure point in the system. If the Traffic Manager service fails, clients cannot access your application during the downtime. Review the Traffic Manager SLA, and determine whether using Traffic Manager alone meets your business requirements for high availability. If not, consider adding another traffic management solution as a failback. If the Azure Traffic Manager service fails, change your CNAME records in DNS to point to the other traffic management service. (This step must be performed manually, and your application will be unavailable until the DNS changes are propagated.)

For the Cassandra cluster, the failover scenarios to consider depend on the consistency levels used by the application, as well as the number of replicas used. 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.

Manageability considerations

When you update your deployment, update one region at a time to reduce the chance of a global failure from an incorrect configuration or an error in the application.

Test the resiliency of the system to failures. Here are some common failure scenarios to test:

- Shut down VM instances.
- Pressure resources such as CPU and memory.
- Disconnect/delay network.
- Crash processes.
- Expire certificates.
- Simulate hardware faults.
- Shut down the DNS service on the domain controllers.

Measure the recovery times and verify they meet your business requirements. Test combinations of failure modes, as well.

Microsoft Azure

Microsoft