






# Recover from a region-wide service disruption

08/18/2016 • 12 minutes to read • Contributors     

## In this article

[Cloud services](#)

[Virtual machines](#)

[Storage](#)

[Database](#)

[Other Azure platform services](#)

[Checklists for disaster recovery](#)

[Cloud Services checklist](#)

[Virtual Machines checklist](#)

[Storage checklist](#)

[SQL Database checklist](#)

[SQL Server on Virtual Machines checklist](#)

[Service Bus checklist](#)

[App Service checklist](#)

[HDInsight checklist](#)

[SQL Reporting checklist](#)

[Media Services checklist](#)

[Virtual Network checklist](#)

Azure is divided physically and logically into units called regions. A region consists of one or more datacenters in close proximity.

Under rare circumstances, it is possible that facilities in an entire region can become inaccessible, for example due to network failures. Or facilities can be lost entirely, for example due to a natural disaster. This section explains the capabilities of Azure for creating applications that are distributed across regions. Such distribution helps to minimize the possibility that a failure in one region could affect other regions.

## Cloud services

### Resource management

You can distribute compute instances across regions by creating a separate cloud service in each target region, and then publishing the deployment package to each cloud service. However, note that distributing traffic across cloud services in different regions must be implemented by the application developer or with a traffic management service.

Determining the number of spare role instances to deploy in advance for disaster recovery is an important aspect of capacity planning. Having a full-scale secondary deployment ensures that capacity is already available when needed; however, this effectively doubles the cost. A common pattern is to have a small, secondary deployment, just large enough to run critical services. This small secondary deployment is a good idea, both to reserve capacity, and for testing the configuration of the secondary environment.

#### Note

The subscription quota is not a capacity guarantee. The quota is simply a credit limit. To guarantee capacity, the required number of roles must be defined in the service model, and the roles must be deployed.

## Load Balancing

To load balance traffic across regions requires a traffic management solution. Azure provides [Azure Traffic Manager](#). You can also take advantage of third-party services that provide similar traffic management capabilities.

## Strategies

Many alternative strategies are available for implementing distributed compute across regions. These must be tailored to the specific business requirements and circumstances of the application. At a high level, the approaches can be divided into the following categories:

- **Redeploy on disaster:** In this approach the application is redeployed from scratch at the time of disaster. This is appropriate for non-critical applications that don't require a guaranteed recovery time.
- **Warm Spare (Active/Passive):** A secondary hosted service is created in an alternate region, and roles are deployed to guarantee minimal capacity; however, the roles don't receive production traffic. This approach is useful for applications that have not been designed to distribute traffic across regions.
- **Hot Spare (Active/Active):** The application is designed to receive production load in multiple regions. The cloud services in each region might be configured for higher capacity than required for disaster recovery purposes. Alternatively, the cloud services might scale out as necessary at the time of a disaster and failover. This approach requires substantial investment in application design, but it has significant benefits. These include low and guaranteed recovery time, continuous testing of all recovery locations, and efficient usage of capacity.

A complete discussion of distributed design is outside the scope of this document. For further information, see [Disaster Recovery and High Availability for Azure Applications](#).

## Virtual machines

Recovery of infrastructure as a service (IaaS) virtual machines (VMs) is similar to platform as a service (PaaS) compute recovery in many respects. There are important differences, however, due to the fact that an IaaS VM consists of both the VM and the VM disk.

- **Use Azure Backup to create cross region backups that are application consistent.** [Azure Backup](#) enables customers to create application consistent backups across multiple VM disks, and support replication of backups across regions. You can do this by choosing to geo-replicate the backup vault at the time of creation. Note that replication of the backup vault must be configured at the time of creation. It can't be set later. If a region is lost, Microsoft will make the backups available to customers. Customers will be able to restore to any of their configured restore points.
- **Separate the data disk from the operating system disk.** An important consideration for IaaS VMs is that you cannot change the operating system disk without re-creating the VM. This is not a problem if your recovery strategy is to redeploy after disaster. However, it might be a problem if you are using the Warm Spare approach to reserve capacity. To implement this properly, you must have the correct operating system disk deployed to both the primary and secondary locations, and the application data must be stored on a separate drive. If possible, use a standard operating system configuration that can be provided on both locations. After a failover, you must then attach the data drive to your existing IaaS VMs in the secondary DC. Use AzCopy to copy snapshots of the data disk(s) to a remote site.
- **Be aware of potential consistency issues after a geo-failover of multiple VM Disks.** VM Disks are implemented as Azure Storage blobs, and have the same geo-replication characteristic. Unless [Azure Backup](#) is used, there are no guarantees of consistency across disks, because geo-replication is asynchronous and replicates independently. Individual VM disks are guaranteed to be in a crash consistent state after a geo-failover, but not consistent across disks. This could cause problems in some cases (for example, in the case of disk striping).

# Storage

## Recovery by using Geo-Redundant Storage of blob, table, queue and VM disk storage

In Azure, blobs, tables, queues, and VM disks are all geo-replicated by default. This is referred to as Geo-Redundant Storage (GRS). GRS replicates storage data to a paired datacenter hundreds of miles apart within a specific geographic region. GRS is designed to provide additional durability in case there is a major datacenter disaster. Microsoft controls when failover occurs, and failover is limited to major disasters in which the original primary location is deemed unrecoverable in a reasonable amount of time. Under some scenarios, this can be several days. Data is typically replicated within a few minutes, although synchronization interval is not yet covered by a service level agreement.

In the event of a geo-failover, there will be no change to how the account is accessed (the URL and account key will not change). The storage account will, however, be in a different region after failover. This could impact applications that require regional affinity with their storage account. Even for services and applications that do not require a storage account in the same datacenter, the cross-datacenter latency and bandwidth charges might be compelling reasons to move traffic to the failover region temporarily. This could factor into an overall disaster recovery strategy.

In addition to automatic failover provided by GRS, Azure has introduced a service that gives you read access to the copy of your data in the secondary storage location. This is called Read-Access Geo-Redundant Storage (RA-GRS).

For more information about both GRS and RA-GRS storage, see [Azure Storage replication](#).

## Geo-Replication region mappings


It is important to know where your data is geo-replicated, in order to know where to deploy the other instances of your data that require regional affinity with your storage. For more information see [Azure Paired Regions](#).

## Geo-Replication pricing

Geo-replication is included in current pricing for Azure Storage. This is called Geo-Redundant Storage (GRS). If you do not want your data geo-replicated you can disable geo-replication for your account. This is called Locally Redundant Storage, and it is charged at a discounted price compared to GRS.

## Determining if a geo-failover has occurred

If a geo-failover occurs, this will be posted to the [Azure Service Health Dashboard](#). Applications can implement an automated means of detecting this, however, by monitoring the geo-region for their storage account. This can be used to trigger other recovery operations, such as activation of compute resources in the geo-region where their storage moved to. You can perform a query for this from the service management API, by using [Get Storage Account Properties](#). The relevant properties are:

		 Copy
<pre>&lt;GeoPrimaryRegion&gt;primary-region&lt;/GeoPrimaryRegion&gt; &lt;StatusOfPrimary&gt;[Available Unavailable]&lt;/StatusOfPrimary&gt; &lt;LastGeoFailoverTime&gt;DateTime&lt;/LastGeoFailoverTime&gt; &lt;GeoSecondaryRegion&gt;secondary-region&lt;/GeoSecondaryRegion&gt; &lt;StatusOfSecondary&gt;[Available Unavailable]&lt;/StatusOfSecondary&gt;</pre>		

## VM disks and geo-failover

As discussed in the section on VM disks, there are no guarantees for data consistency across VM disks after a failover. To ensure correctness of backups, a backup product such as Data Protection Manager should be used to back up and restore application data.

# Database

## SQL Database

Azure SQL Database provides two types of recovery: Geo-Restore and Active Geo-Replication.

### Geo-Restore

[Geo-Restore](#) is also available with Basic, Standard, and Premium databases. It provides the default recovery option when the database is unavailable because of an incident in the region where your database is hosted. Similar to Point-In-Time Restore, Geo-Restore relies on database backups in geo-redundant Azure storage. It restores from the geo-replicated backup copy, and therefore is resilient to the storage outages in the primary region. For more details, see [Restore an Azure SQL Database or failover to a secondary](#).

### Active Geo-Replication

[Active Geo-Replication](#) is available for all database tiers. It's designed for applications that have more aggressive recovery requirements than Geo-Restore can offer. Using Active Geo-Replication, you can create up to four readable secondaries on servers in different regions. You can initiate failover to any of the secondaries. In addition, Active Geo-Replication can be used to support the application upgrade or relocation scenarios, as well as load balancing for read-only workloads. For details, see [configure Geo-Replication](#) and to [fail over to the secondary database](#). Refer to [Design an application for cloud disaster recovery using Active Geo-Replication in SQL Database](#) and [Managing rolling upgrades of cloud applications using SQL Database Active Geo-Replication](#) for details on how to design and implement applications and applications upgrade without downtime.

## SQL Server on Virtual Machines

A variety of options are available for recovery and high availability for SQL Server 2012 (and later) running in Azure Virtual Machines. For more information, see [High availability and disaster recovery for SQL Server in Azure Virtual Machines](#).

## Other Azure platform services

When attempting to run your cloud service in multiple Azure regions, you must consider the implications for each of your dependencies. In the following sections, the service-specific guidance assumes that you must use the same Azure service in an alternate Azure datacenter. This involves both configuration and data-replication tasks.

### ⓘ Note

In some cases, these steps can help to mitigate a service-specific outage rather than an entire datacenter event. From the application perspective, a service-specific outage might be just as limiting and would require temporarily migrating the service to an alternate Azure region.

## Service Bus

Azure Service Bus uses a unique namespace that does not span Azure regions. So the first requirement is to set up the necessary service bus namespaces in the alternate region. However, there are also considerations for the durability of the queued messages. There are several strategies for replicating messages across Azure regions. For the details on these replication strategies and other disaster recovery strategies, see [Best practices for insulating applications against Service Bus outages and disasters](#).

## App Service

To migrate an Azure App Service application, such as Web Apps or Mobile Apps, to a secondary Azure region, you must have a backup of the website available for publishing. If the outage does not involve the entire Azure datacenter, it might be possible to use FTP to download a recent backup of the site content. Then create a new app in the alternate region, unless you have previously done this to reserve capacity. Publish the site to the new region, and make any necessary configuration changes. These changes could include database connection strings or other region-specific settings. If necessary, add the site's SSL certificate and change the DNS CNAME record so that the custom domain name points to the redeployed Azure Web App URL.

## HDInsight

The data associated with HDInsight is stored by default in Azure Blob Storage. HDInsight requires that a Hadoop cluster processing MapReduce jobs must be co-located in the same region as the storage account that contains the data being analyzed. Provided you use the geo-replication feature available to Azure Storage, you can access your data in the secondary region where the data was replicated if for some reason the primary region is no longer available. You can create a new Hadoop cluster in the region where the data has been replicated and continue processing it.

## SQL Reporting

At this time, recovering from the loss of an Azure region requires multiple SQL Reporting instances in different Azure regions. These SQL Reporting instances should access the same data, and that data should have its own recovery plan in the event of a disaster. You can also maintain external backup copies of the RDL file for each report.

## Media Services

Azure Media Services has a different recovery approach for encoding and streaming. Typically, streaming is more critical during a regional outage. To prepare for this, you should have a Media Services account in two different Azure regions. The encoded content should be located in both regions. During a failure, you can redirect the streaming traffic to the alternate region. Encoding can be performed in any Azure region. If encoding is time-sensitive, for example during live event processing, you must be prepared to submit jobs to an alternate datacenter during failures.

## Virtual network

Configuration files provide the quickest way to set up a virtual network in an alternate Azure region. After configuring the virtual network in the primary Azure region, [export the virtual network settings](#) for the current network to a network configuration file. In the event of an outage in the primary region, [restore the virtual network](#) from the stored configuration file. Then configure other cloud services, virtual machines, or cross-premises settings to work with the new virtual network.

# Checklists for disaster recovery

## Cloud Services checklist

1. Review the Cloud Services section of this document.
2. Create a cross-region disaster recovery strategy.
3. Understand trade-offs in reserving capacity in alternate regions.
4. Use traffic routing tools, such as Azure Traffic Manager.

## Virtual Machines checklist

1. Review the Virtual Machines section of this document.
2. Use [Azure Backup](#) to create application consistent backups across regions.

## Storage checklist

1. Review the Storage section of this document.
2. Do not disable geo-replication of storage resources.
3. Understand alternate region for geo-replication in the event of failover.
4. Create custom backup strategies for user-controlled failover strategies.

## SQL Database checklist

1. Review the SQL Database section of this document.
2. Use [Geo-Restore](#) or [Geo-Replication](#) as appropriate.

## SQL Server on Virtual Machines checklist

1. Review the SQL Server on Virtual Machines section of this document.
2. Use cross-region AlwaysOn Availability Groups or database mirroring.
3. Alternately use backup and restore to blob storage.

## Service Bus checklist

1. Review the Service Bus section of this document.
2. Configure a Service Bus namespace in an alternate region.
3. Consider custom replication strategies for messages across regions.

## App Service checklist

1. Review the App Service section of this document.
2. Maintain site backups outside of the primary region.
3. If outage is partial, attempt to retrieve current site with FTP.
4. Plan to deploy the site to new or existing site in an alternate region.
5. Plan configuration changes for both application and DNS CNAME records.

## HDInsight checklist

1. Review the HDInsight section of this document.
2. Create a new Hadoop cluster in the region with replicated data.

## SQL Reporting checklist

1. Review the SQL Reporting section of this document.
2. Maintain an alternate SQL Reporting instance in a different region.
3. Maintain a separate plan to replicate the target to that region.

## Media Services checklist

1. Review the Media Services section of this document.
2. Create a Media Services account in an alternate region.
3. Encode the same content in both regions to support streaming failover.
4. Submit encoding jobs to an alternate region in the event of a service disruption.

# Virtual Network checklist

1. Review the Virtual Network section of this document.
2. Use exported virtual network settings to re-create it in another region.