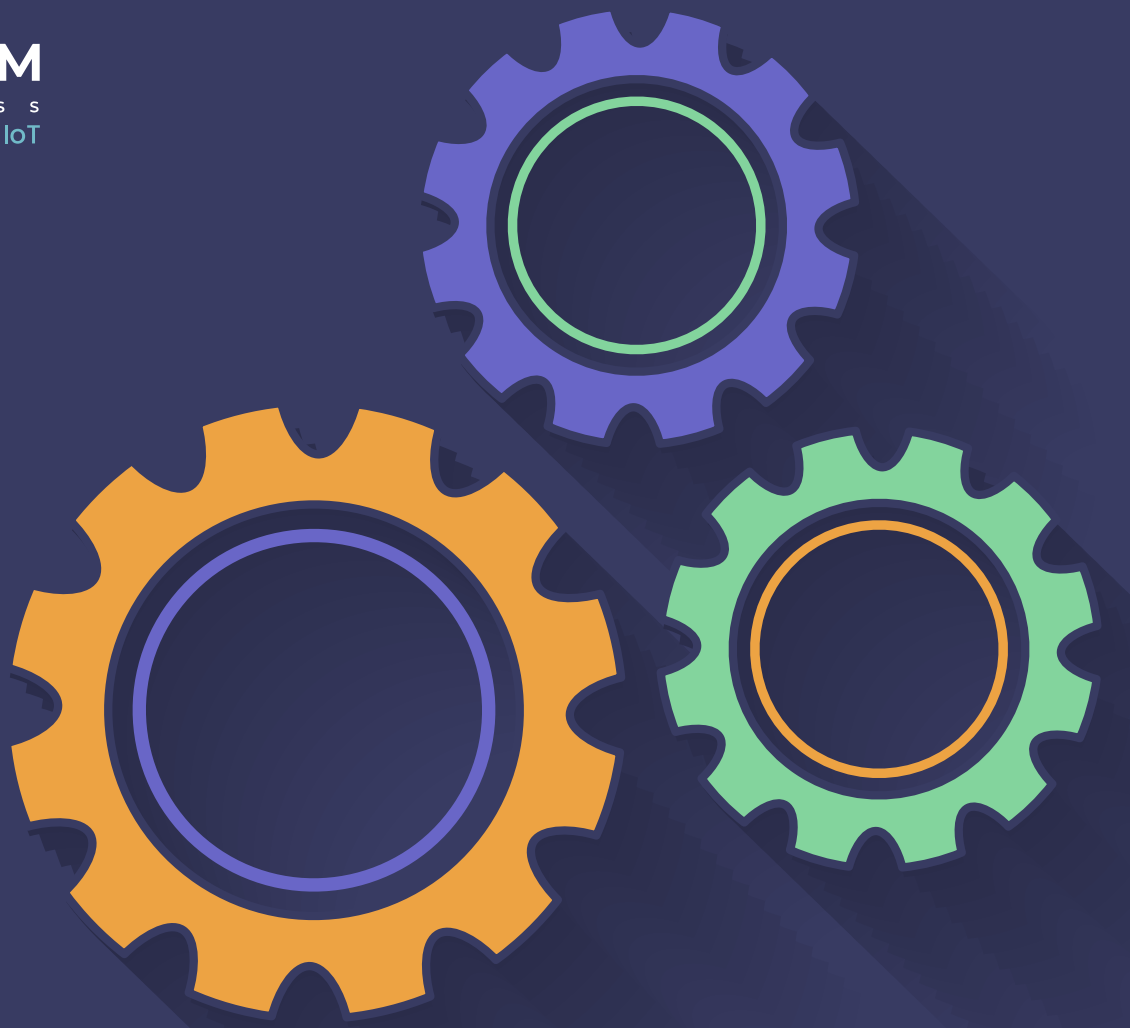
SHIPCOM WIRELESS IoT



Business Continuity High Availability & Disaster Recovery

A Whitepaper by Shipcom Wireless, Inc. | MARCH 2021

When servers are provisioned and line of business applications are deployed on the servers, at some point, these systems fail due to power outage, hard drive failures, network failures or natural disasters. Our solutions must be resilient and designed for failures. Resilience is the ability for a system, a collection of servers, storage and network components, to recover from failure, to avoid or eliminate downtime and data loss, and continue functioning and serving the end users. Resilient applications eliminate single point of failures and return to a fully-functional state after an issue arises. Underneath the umbrella of resiliency are two core concepts - **High Availability and Disaster Recovery.**

- High Availability (HA) is the ability of an application to continue running without any significant downtime, an example of which is a cluster of SQL servers part of a failover group, configured for replication and behind a load balancer, when the master database is unavailable due to a failure, the system fails over to the secondary database in a small timeframe and continues serving users with little to no interruption.

- Disaster Recovery (DR) is the ability to recover from rare, but major incidents, like region outages in Azure due to natural disasters. Applications can be configured for highly available in a single region, especially with the tools and services available in Azure, but when the region goes down, the entire application becomes unavailable to the users. Applications need to be configured to respond to a region outage and bring the application online at another region in the global infrastructure. Disaster Recovery starts when an impact of a fault exceeds the ability of the Highly Available system to recover within the expected SLAs.

Every organization has a different availability requirement, for instance a basic business could be down for the entire weekend and could incur no financial loss, but a retail business that is serving users all over the world could incur a loss of millions of dollars with even 10 minutes of downtime. Every organization has unique requirements, and the applications should be designed to best meet the its needs. Defining a target SLA makes it possible to evaluate whether the architecture meets the business requirements. Some things to consider include:

- What are the availability requirements and how much downtime is acceptable?
- What are the data replication requirements?
- What are the data backup requirements?
- What are the monitoring requirements?
- Does your application have specific latency requirements?
- How much should the business invest in making the application highly available?

Highly available systems guarantee a certain percentage of uptime—for example, a system that has 99.9% uptime will be down only 0.1% of the time —0.365 days or 8.76 hours per year. The number of "nines" is commonly used to indicate the degree of high availability. For example, "five nines" indicates a system that is up 99.999% of the time. Metrics like Recovery Time Objective (RTO), Recovery Point Objective (RPO) and Recovery Level Objective (RLO) help define the target Service Level Agreement (SLA) for each application workload.

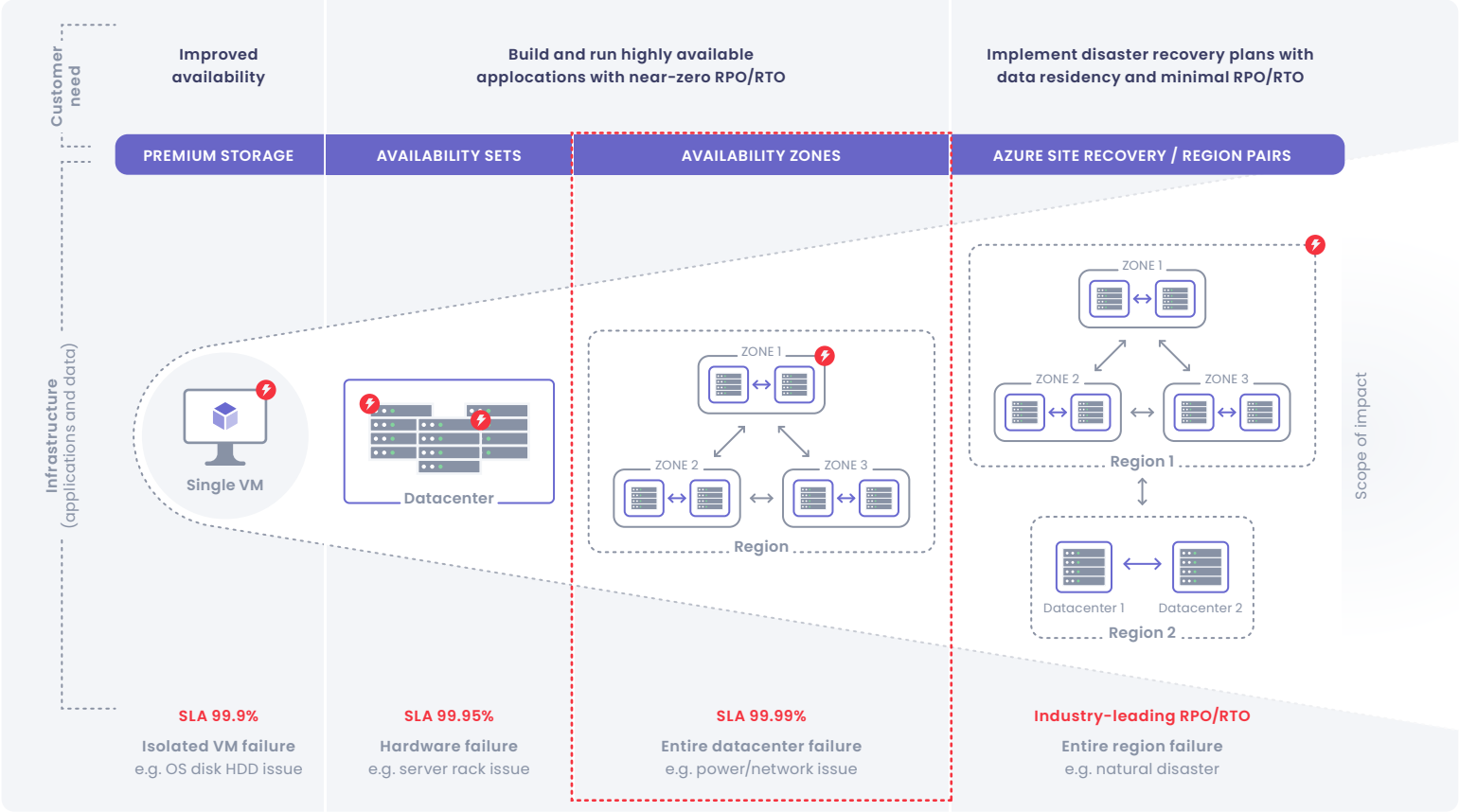
Metrics

RTO stands for recovery time objective. RTO for an application is the maximum acceptable time that an application can be unavailable after some kind of incident. If a customer's business continuity gets affected when the systems are down for more than 60 mins, then RTO for the application will be defined at 60 mins and the application must be back up and running within the defined window. The lower the RTO, the more aggressive the recovery process needs to be.

Consider an application running in the West Coast on the Azure platform, and an earthquake takes out the entire West Cost. With an RTO of 60 mins to restore applications from backup and to get it up in running in a different region would be a difficult task. A warm standby running server, mirroring and replicating data over near real time in the west coast, in an alternate region, where the application starts to fail over could protect against that regional outage scenario and meet the RTO of 60 mins.

RPO stands for recovery point objective. RPO is the maximum duration of data loss that's acceptable during a disaster. For example, a single server with a standalone database and no secondary servers, configured for hourly backups and no data replication is setup for RPO of 60 minutes. If the business requires a lower RPO for this particular dataset, the data would need to be replicated to a secondary database or backups at smaller intervals.

Azure SLAs for the Deployment Topologies



Azure infrastructure is composed of geographies, regions, and Availability Zones, which limit the blast radius of a failure and therefore limit potential impact to customer applications and data.

Availability Sets:

An Availability Set is a logical grouping for isolating VM resources from each other when they are deployed. Azure makes sure that the VMs placed within an Availability Set run across multiple physical servers, compute racks, storage units, and network switches and supports High Availability within a data center.

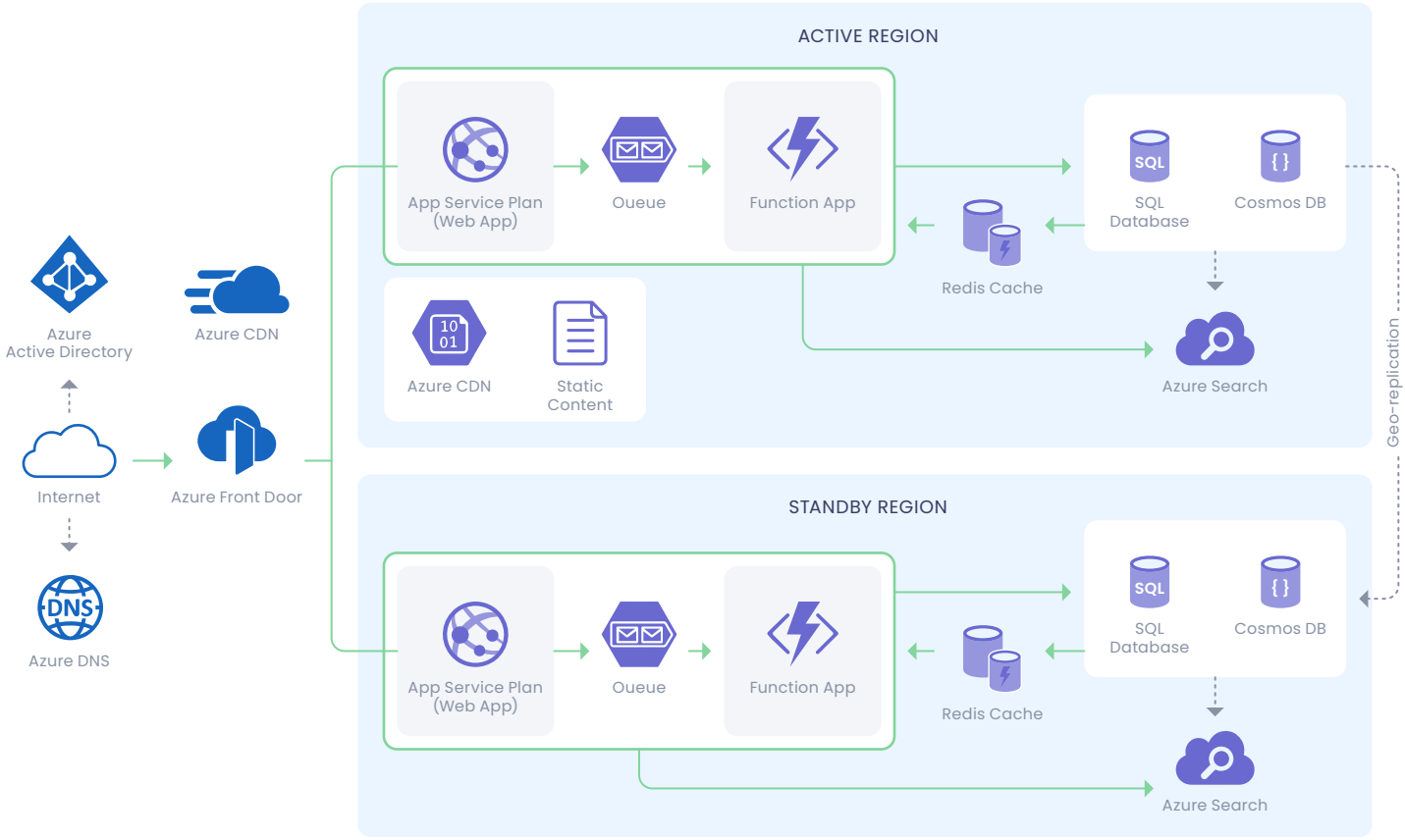
Availability Zones:

Availability Zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters with independent power, cooling, and networking. The physical separation of Availability Zones within a region limits the impact to applications and data from zone failures, such as large-scale flooding, major storms and superstorms, and other events that could disrupt site access, safe passage, extended utilities uptime, and the availability of resources. Availability Zones and their associated datacenters are designed such that if one zone is compromised, the services, capacity, and availability are supported by the other Availability Zones in the region.

Paired Regions:

Azure paired regions are optimized to work together for disaster recovery. There's typically at least 300 miles between the 2 paired regions which provides resiliency for natural disasters. If one region goes down because of a natural disaster, the application fails over to the resources configured in the paired region. For an application spread across both those regions, Microsoft guarantees to have one of those regions recovered with priority. If the application is architected across two regions like that, you've got a better likelihood of getting your application restored faster, and also have sequential updates across paired regions.

Multi Region Deployment Architecture Diagram



The above architecture shows how Azure resources are configured in multiple regions along with Azure Front Door, a load balancer, to achieve high availability. Other Azure load balancers with priority routing like Azure Traffic Manager, Azure Application Gateway can also be configured to route traffic and support high availability.

- **Primary and secondary regions:**
  This architecture uses two regions to support HA and the application components are deployed in paired regions to achieve higher availability. During normal operations, network traffic is routed to the primary region. If the primary region becomes unavailable, traffic is routed to the secondary region.

- **Front Door:** Front Door routes incoming requests to the primary region. If the application running that region becomes unavailable, Front Door fails over to the secondary region.

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.

Front Door Configuration

Routing:

Front Door supports several routing mechanisms like path-based routing, weighted routing, performance -based routing etc. For the approach described in this article, FD is configured for priority routing. With this setting, Front Door sends all requests to the primary region unless the endpoint for that region becomes unreachable. At that point, it automatically fails over to the secondary region. The backend pool is set with different priority values, 1 for the active region and 2 or higher for the standby or passive region.

Health probe:

Front Door uses an HTTP (or HTTPS) probe to monitor the availability of each back end. The probe gives Front Door a pass/fail test for failing over to the secondary region. It works by sending a request to a specified URL path. If it gets a non-200 response within a timeout period, the probe fails. The health probe frequency can be configured based on the number of samples required for evaluation, and the number of successful samples required for the backend to be marked as healthy.
If Front Door marks the backend as degraded, it fails over to the other backend.

As a best practice, the health probe path in the application backend reports the overall health of the application. This health probe checks the critical dependencies such as the App Service apps, Storage Queue, and SQL Database and doesn't monitor the health of lower priority services. Otherwise, the probe might report a healthy backend when critical parts of the application are actually failing.

HS and DR Strategy for the Application Components

AZURE RESOURCES	HIGH AVAILABILITY AND DISASTER RECOVERY STRATEGY	DATA STRATEGY FOR DISASTER RECOVERY
App Services Function Apps	**HA across Availability Sets** Configure Autoscaling on a minimum of 2 instances to load balance the traffic for HA. Enable Health Monitoring on App Services. **DR across Paired Regions** Configure the AppService in Active and Standby Region. Configure Front Door with priority and health probe to load balance traffic on the backend pool to support failover when a disaster occurs. Configure Failure Detection and Monitoring.	Standard and Production Plans for App Services support automated backups and Auto Scaling.
Azure Storage Azure Queues Azure Blobs	**Local Redundant Storage** Data replicates three times within a single datacenter. **Zone Redundant Storage** Data replicates synchronously across zones in a region. Data can be accessed and managed when one of the availability zones becomes unavailable. **Geo Redundant Storage** Data replicates asynchronously across regions, which implies there is a delay when data written to primary is synced with secondary. Account failover can cause data loss. Configure Failure Detection and Monitoring.	Daily Geo Replicated backups of the storage account.
Azure SQL Server	**HA across Availability Zones** Premium Plans allow replication of data across zones. **DR across Paired Regions** Configure Geo Replication that creates readable secondary replicas of data across paired regions. Configure Auto Failover Group with Primary and Secondary DBs configured in paired regions. Configure Failure Detection and Monitoring.	SQL Server configured for full backups every week, differential backups every 12-24 hours, and transaction log backups every 5 to 10 minutes and Backups are Geo Replicated across Paired Regions. Point In Time Restores (PITR) configured for all the backups and can be Geo Restored.

Catamaran NextGen IoT Platform,

a SaaS offering, deployed on the Azure cloud, comprises of a suite of offerings including Cold Chain Logistics, GPS and Fleet Management, Labs Monitoring and Smart Building solutions. Catamaran NextGen IoT Platform implements resilience for all tiers of your application based on the type of application, data requirements, application availability SLAs, failure mode analysis (FMA) and cost versus risk analysis catered to your business needs.

REFERENCES:

https://docs.microsoft.com/ en-us/azure/architecture/reference-architectures/app-service-web-app/multi-region

https://docs.microsoft.com/en-us/azure/architecture/high-availability/building-solutions-for-high-availability

SHIPCOM WIRELESS IoT

www.shipcom.io