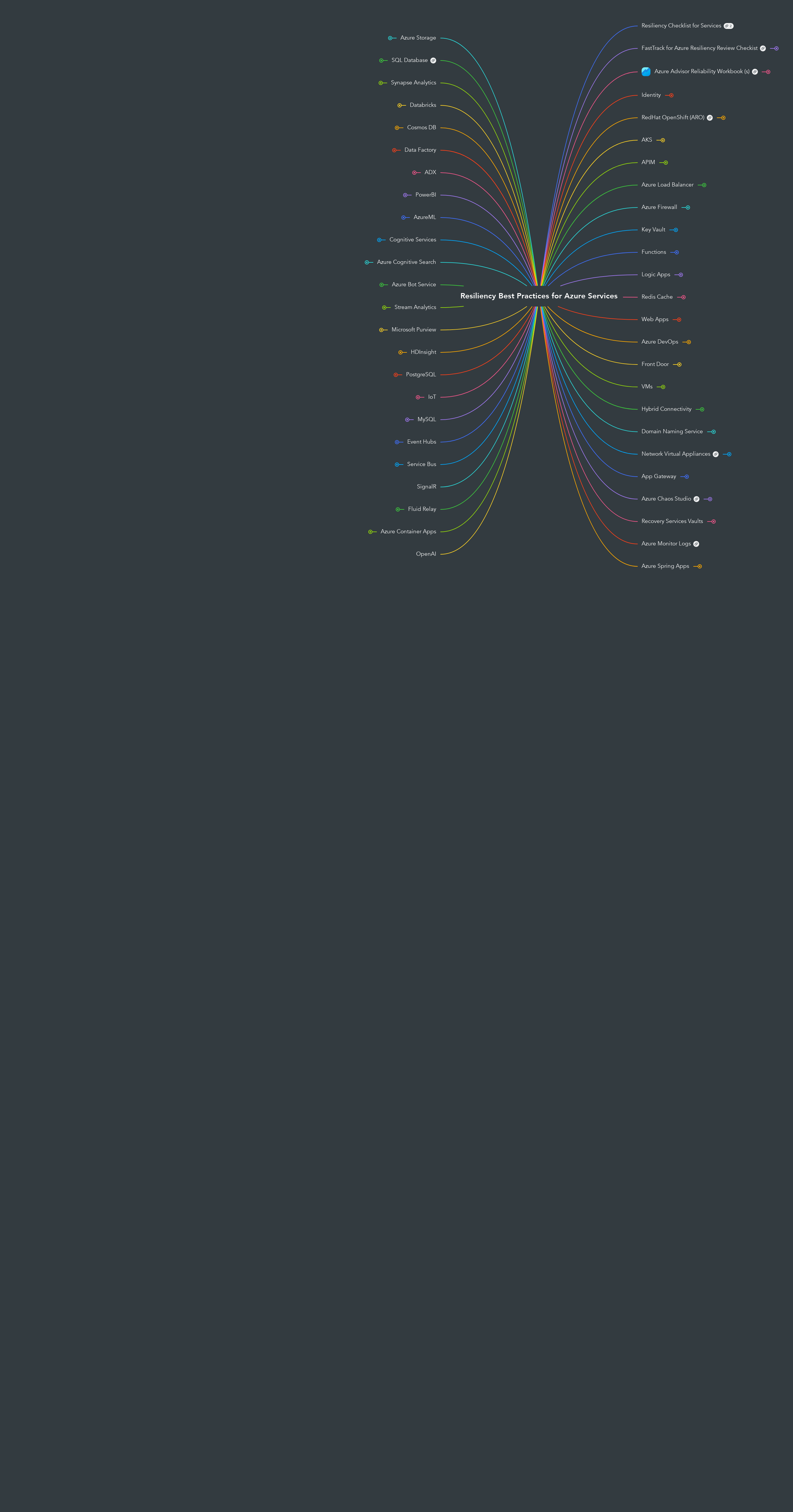


Resiliency Best Practices for Azure Services



Contents

[1 Resiliency Checklist for Services 20](#_Toc147861696)

[2 Key Vault 21](#_Toc147861697)

[2.1 FTA Resiliency Playbook 21](#_Toc147861698)

[2.2 Best Practices 22](#_Toc147861699)

[2.2.1 Key Vault is a managed service and Microsoft will handle the failover if a region goes down. 22](#_Toc147861700)

[2.2.2 Please note: there is no manual-failover. 22](#_Toc147861701)

[2.2.3 When you back up a key vault object, such as a secret, key, or certificate, the backup operation will download the object as an encrypted blob. This blob can't be decrypted outside of Azure. To get usable data from this blob, you must restore the blob ino a key vault within the same Azure subscription and Azure geography. 22](#_Toc147861702)

[2.2.4 Leverage Soft Deletes 22](#_Toc147861703)

[2.2.5 Purge protection 22](#_Toc147861704)

[2.2.6 Backup and restore objects 22](#_Toc147861705)

[2.2.7 Architectural Design Considerations 23](#_Toc147861706)

[3 Web Apps 24](#_Toc147861707)

[3.1 FTA Resiliency Playbook 24](#_Toc147861708)

[3.2 Best Practices 24](#_Toc147861709)

[3.2.1 Multi-tenant service 24](#_Toc147861710)

[3.2.2 App Service Environment v3 24](#_Toc147861711)

[3.2.3 Use Application Insights 25](#_Toc147861712)

[4 APIM 26](#_Toc147861713)

[4.1 Best Practices 26](#_Toc147861714)

[4.1.1 Leverage Availability Zones where regionally applicable 26](#_Toc147861715)

[4.1.2 Use the premium tier for production workloads. 26](#_Toc147861716)

[4.1.3 Run APIM with capacity greater than 1 26](#_Toc147861717)

[4.1.4 For DR, leverage the premium tier with deployments scaled across two or more regions for 99.99% SLA 26](#_Toc147861718)

[4.1.5 Managing the APIM APIs from DevOps 26](#_Toc147861719)

[4.1.6 If using Self-Hosted Gateways 26](#_Toc147861720)

[4.1.7 Be aware of your limits 26](#_Toc147861721)

[4.1.8 Use Policies to add a fail-over backend URL and caching to reduce failing calls. 27](#_Toc147861722)

[4.1.9 Leverage multi-region deployment of API gateways(available only in premium tiers). APIM routes the requests to a regional gateway based on the lowest latency. You could use your own Traffic Manager with custom routing rules . 27](#_Toc147861723)

[4.1.10 In multi-region model, use Policies to route the requests to regional backends based on availability or latency. 27](#_Toc147861724)

[5 VMs 28](#_Toc147861725)

[5.1 Best Practices 28](#_Toc147861726)

[5.1.1 Question(s) you should ask yourself 28](#_Toc147861727)

[5.1.2 Ensure backups are enabled to protect data 28](#_Toc147861728)

[5.1.3 Leverage DevOps (IaC) 29](#_Toc147861729)

[5.1.4 Single Region Resiliency 29](#_Toc147861730)

[5.1.5 Mulit Region Mindset 31](#_Toc147861731)

[5.1.6 Back up Vs. DR - Related scenarios, but not the same 32](#_Toc147861732)

[5.1.7 Plan for Paired Regions 34](#_Toc147861733)

[5.1.8 Plan for Reserved Capacity 35](#_Toc147861734)

[5.2 VMSS 35](#_Toc147861735)

[5.2.1 Best Practices 35](#_Toc147861736)

[6 AKS 37](#_Toc147861737)

[6.1 Best Practices 37](#_Toc147861738)

[6.1.1 Set an Uptime SLA 37](#_Toc147861739)

[6.1.2 Leverage Availability Zones where regionally applicable 37](#_Toc147861740)

[6.1.3 Use ZRS storage with stateful workloads 37](#_Toc147861741)

[6.1.4 Plan for multi-region deployments 38](#_Toc147861742)

[6.1.5 Use System and User node pools 39](#_Toc147861743)

[6.1.6 Kubernetes best practices 39](#_Toc147861744)

[6.1.7 Use Application Insights 40](#_Toc147861745)

[7 Functions 41](#_Toc147861746)

[7.1 FTA Resiliency Playbook 41](#_Toc147861747)

[7.2 Best Practices 41](#_Toc147861748)

[7.2.1 Use an Azure Functions Premium Plan 41](#_Toc147861749)

[7.2.2 Use an App Service Environment v3 41](#_Toc147861750)

[7.2.3 Leverage App Service Plans if you want to control the scale up and scale out yourself. 41](#_Toc147861751)

[7.2.4 Pair a Function App to its own storage account. Try not to re-use storage accounts for Function Apps unless they are tightly coupled. 41](#_Toc147861752)

[7.2.5 Leverage DevOps or GitHub to streamline CI/CD and safeguard your Function App code 41](#_Toc147861753)

[8 Logic Apps 42](#_Toc147861754)

[8.1 FTA Resiliency Playbook 42](#_Toc147861755)

[8.2 Best Practices 43](#_Toc147861756)

[8.2.1 Use Logic App Standard -- Single Tenant 43](#_Toc147861757)

[8.2.2 Use Logic Apps Standard -- App Service Environment v3 (Windows plans only) 43](#_Toc147861758)

[8.2.3 For a resiliency against full-region outages, the recommendation is to setup a logic apps in multiple regions. 43](#_Toc147861759)

[9 Azure Firewall 44](#_Toc147861760)

[9.1 Best Practices 44](#_Toc147861761)

[9.1.1 Leverage Availability Zones where regionally applicable 44](#_Toc147861762)

[10 Front Door 45](#_Toc147861763)

[10.1 FTA Resiliency Playbook 45](#_Toc147861764)

[10.2 Best Practices 46](#_Toc147861765)

[10.2.1 An important point here is that having multiple instances of AFD IS NOT NEEDED FOR ADDITIONAL RESILIENCY unless the customer’s needs exceed the defined limits (number of origin groups, number of origins, number of endpoints, etc.). 46](#_Toc147861766)

[10.2.2 Automatic Failover 46](#_Toc147861767)

[10.2.3 Manual Failover 46](#_Toc147861768)

[10.2.4 Leverage Health Probes 46](#_Toc147861769)

[11 Azure Load Balancer 47](#_Toc147861770)

[11.1 Best Practices 47](#_Toc147861771)

[11.1.1 Leverage the Standard Load Balancer sku 47](#_Toc147861772)

[11.1.2 Leverage Availability Zones where regionally applicable 47](#_Toc147861773)

[12 App Gateway 48](#_Toc147861774)

[12.1 Best Practices 48](#_Toc147861775)

[12.1.1 Leverage Availability Zones where regionally applicable 48](#_Toc147861776)

[12.1.2 Leverage the Standard\_v2 and Waf\_v2 SKUs 48](#_Toc147861777)

[12.1.3 Deploy Application Gateway with at least two instances 48](#_Toc147861778)

[13 Redis Cache 49](#_Toc147861779)

[13.1 Best Practices 49](#_Toc147861780)

[13.1.1 High availability and disaster recovery 49](#_Toc147861781)

[13.1.2 Zone redundancy 49](#_Toc147861782)

[13.1.3 Leverage Redis Persistence 49](#_Toc147861783)

[13.1.4 Use Geo-redundant storage account or zonally redundant where geo is not available 49](#_Toc147861784)

[13.1.5 Passive Geo-replication 49](#_Toc147861785)

[14 Azure Monitor Logs 50](#_Toc147861786)

[15 Azure Chaos Studio 51](#_Toc147861787)

[15.1 Resilience through chaos engineering 51](#_Toc147861788)

[15.1.1 Is a managed service that uses chaos engineering to help you measure, understand, and improve your cloud application and service resilience 51](#_Toc147861789)

[15.1.2 Improve application resilience with chaos testing by deliberately introducing faults that simulate real-world outages. 51](#_Toc147861790)

[15.1.3 Disrupt your apps intentionally to identify gaps and plan mitigations before your customers are impacted by a problem. 51](#_Toc147861791)

[15.2 Common Chaos Engineering Scenarios 51](#_Toc147861792)

[15.2.1 Reproduce an incident that affected your application, to better understand the failure. Ensure that post-incident repairs prevent the incident from recurring. 51](#_Toc147861793)

[15.2.2 Prepare for a major event or season with "game day" load, scale, performance, and resilience validation. 51](#_Toc147861794)

[15.2.3 Do business continuity and disaster recovery (BCDR) drills to ensure that your application can recover quickly and preserve critical data in a disaster. 51](#_Toc147861795)

[15.2.4 Run high availability (HA) drills to test application resilience against region outages, network configuration errors, high stress events, or noisy neighbor issues. 51](#_Toc147861796)

[15.2.5 Plan capacity needs for production environments. 51](#_Toc147861797)

[15.2.6 Run stress tests or load tests. 51](#_Toc147861798)

[15.2.7 Build confidence in services built on cloud-native architectures. 51](#_Toc147861799)

[16 Hybrid Connectivity 52](#_Toc147861800)

[16.1 Best Practices 52](#_Toc147861801)

[16.1.1 VPN Gateway and Express Route DR Best Practices will apply to Hub & Spoke and VWan Architectures 52](#_Toc147861802)

[16.1.2 VPN Gateways 52](#_Toc147861803)

[16.1.3 ExpressRoutes 54](#_Toc147861804)

[16.1.4 Other 59](#_Toc147861805)

[17 FastTrack for Azure Resiliency Review Checkist 60](#_Toc147861806)

[17.1 FastTrack for Azure engineers have compiled a Excel checklist of all the relevant guidance and documentation for resiliency across all supported services. 60](#_Toc147861807)

[18 Azure Advisor Reliability Workbook (s) 61](#_Toc147861808)

[18.1 The objective of this workbook is to provide an overview of the configured state of deployment options that affect the resilience and/or availability SLA for common Azure resource types. 61](#_Toc147861809)

[18.1.1 Building a reliable application in the cloud is different from traditional application development. While historically you may have purchased levels of redundant higher-end hardware to minimize the chance of an entire application platform failing, in thecloud, we acknowledge up front that failures will happen. Instead of trying to prevent failures altogether, the goal is to minimize the effects of a single failing component. 61](#_Toc147861810)

[18.1.2 Reliable applications are 61](#_Toc147861811)

[18.1.3 Understanding how these elements work together — and how they affect cost — is essential to building a reliable application. It can help you determine how much downtime is acceptable, the potential cost to your business, and which functions are necessaryduring a recovery. 61](#_Toc147861812)

[19 Azure DevOps 62](#_Toc147861813)

[19.1 Best Practices 62](#_Toc147861814)

[19.1.1 Azure DevOps 62](#_Toc147861815)

[19.1.2 GitHub 62](#_Toc147861816)

[20 Recovery Services Vaults 63](#_Toc147861817)

[20.1 Azure Site Recovery 63](#_Toc147861818)

[20.1.1 On-premises VM Replication (Hyper-V, VMware or Physical) 63](#_Toc147861819)

[20.1.2 Azure VM Replication 63](#_Toc147861820)

[20.1.3 Define recovery plans to automate the failover sequence for VMs. You can also include automation scripts to reduce manual steps and improve recovery time. 63](#_Toc147861821)

[20.2 Azure Backup 63](#_Toc147861822)

[20.2.1 Enable and LOCK immutability for vaults. This ensures recovery points cannot be deleted before their intended expiry. 63](#_Toc147861823)

[20.2.2 Enable "Always-on soft delete" for vaults protecting critical workloads. 63](#_Toc147861824)

[20.3 When creating Recovery Service Vaults choose the best storage redundancy option for your requirements. Vaults support local, geo and zone redundancy but this setting cannot be changed once the vault is protecting one or more resources. 63](#_Toc147861825)

[21 Network Virtual Appliances 64](#_Toc147861826)

[21.1 Best Practices 64](#_Toc147861827)

[21.1.1 Top questions you should be thinking about 64](#_Toc147861828)

[21.1.2 You want at least two NVAs 64](#_Toc147861829)

[21.1.3 More nodes is more resilient, but more nodes is more money. 64](#_Toc147861830)

[21.1.4 Methods 64](#_Toc147861831)

[21.1.5 HA architectures 64](#_Toc147861832)

[21.1.6 NVAs are often critical resources shared between many different workloads, so their resiliency impacts the resiliency of many different workloads 65](#_Toc147861833)

[22 Identity 66](#_Toc147861834)

[22.1 Azure AD 66](#_Toc147861835)

[22.1.1 Azure AD Connect needs to be planned for resiliency 66](#_Toc147861836)

[22.1.2 This is a global managed service with multiple levels of internal redundancy and automatic recoverability which Microsoft manages 66](#_Toc147861837)

[22.1.3 Use long-live revocable token, cache your token and acquire your silently using Microsoft Identity Library 66](#_Toc147861838)

[22.2 Active Directory B2C 66](#_Toc147861839)

[22.2.1 FTA Resiliency Playbook 66](#_Toc147861840)

[22.2.2 B2C is based on the underlying Azure AD infrastructure. For durability, this infrastructure will replicate any piece of data written to Azure AD at least 4 times and up to 13 datacenters, within a geography, depending on the tenant configuration. The phsical location of the replicas is hidden from the customer and is a detail of the global service. 67](#_Toc147861841)

[22.2.3 This is a global managed service which Microsoft manages. Like AAD, B2C provides automatic failover in the event of an unexpected issue with the service or with the Azure region that is physically running an active replica. 67](#_Toc147861842)

[22.2.4 Reduce auth calls to AAD B2C 67](#_Toc147861843)

[22.2.5 Reduce external API dependencies 67](#_Toc147861844)

[22.2.6 User Sign-In Flow(s) 67](#_Toc147861845)

[22.2.7 Build resilience in your customer identity and access management with Azure Active Directory B2C 67](#_Toc147861846)

[22.2.8 Be aware of how your B2C setup affects your composite SLA 67](#_Toc147861847)

[22.2.9 Custom brand assets should be hosted on a CDN 67](#_Toc147861848)

[22.2.10 Have multiple identiy providers (i.e., login with your microsoft, google, facebook accounts) 67](#_Toc147861849)

[22.3 Windows Server AD 67](#_Toc147861850)

[22.3.1 Follow VM rules for high availability on the VM level (premium disks, two or more in a region, in different availability zones) 67](#_Toc147861851)

[22.3.2 Don't replicate! Replication can create issues with directory synchronization. 67](#_Toc147861852)

[22.3.3 Have active-active for multi-regions 68](#_Toc147861853)

[22.4 Azure AD Domain Services 68](#_Toc147861854)

[22.4.1 Add Azure AD Domain service stamps to additional regions and locations 68](#_Toc147861855)

[22.4.2 Use Replica Sets for DR 68](#_Toc147861856)

[23 Domain Naming Service 69](#_Toc147861857)

[23.1 Azure Private DNS 69](#_Toc147861858)

[23.1.1 Private DNS Zones are multi-region by default 69](#_Toc147861859)

[23.1.2 Verify that Zones are linked to Vnets in multiple regions 69](#_Toc147861860)

[23.1.3 If different Zones are used between regions, verify a plan for making sure that Zones are up to date in a DR failover situation 69](#_Toc147861861)

[23.2 Azure DNS 69](#_Toc147861862)

[23.2.1 Multi-region by default 69](#_Toc147861863)

[23.2.2 Plan for disaster recovery with Azure DNS and Traffic Manager 69](#_Toc147861864)

[23.3 Azure DNS Resolver 69](#_Toc147861865)

[23.3.1 Enable availability zones with Private Resolver 69](#_Toc147861866)

[23.3.2 Plan for failover with Private Resolvers in a Disaster Recovery 69](#_Toc147861867)

[23.4 VM Based DNS Services (AD, Infoblox, etc.) 69](#_Toc147861868)

[23.4.1 Follow VM Guidance 69](#_Toc147861869)

[23.4.2 IF AD based DNS, follow the Identity -> Windows Server AD path 69](#_Toc147861870)

[23.4.3 IF a third party, follow the vendor's specific guidance 69](#_Toc147861871)

[24 Azure Spring Apps 70](#_Toc147861872)

[24.1 Deploy zone redundant ASA instance 70](#_Toc147861873)

[24.1.1 In supported region, Azure spring Apps can be deployed as zone redundant, which means that instances are automatically distributed across availability zones. This feature is only available in Standard and Enterprise tiers. 70](#_Toc147861874)

[24.2 Use Blue-Green deployment pattern for the apps 70](#_Toc147861875)

[24.2.1 Azure Spring Apps permits two deployments for every app, only one of which receives production traffic. You can achieve zero downtime with blue green deployment strategies. Blue green deployment is only available in Standard and Enterprise tiers. You could automate deployment using CI/CD with ADO/GitHub actions 70](#_Toc147861876)

[24.3 Set up alerts and dashboards for monitoring and notification 70](#_Toc147861877)

[24.3.1 Monitor Azure spring apps with logs, metrics and tracing. Integrate ASA with application insights and track failures and create workbooks. 70](#_Toc147861878)

[24.4 Create multi region model for high availability and disaster recovery 70](#_Toc147861879)

[24.4.1 Azure Spring Apps instances could be created in multiple regions for your applications and traffic could be routed by Traffic Manager/Front Door. 70](#_Toc147861880)

[24.5 Enable VMware Spring Cloud Gateway(Enterprise tiers) 70](#_Toc147861881)

[24.5.1 Set up autoscaling in Spring Cloud Gateway 70](#_Toc147861882)

[24.6 Use Enterprise plan for commercial support of spring boot for mission critical apps. With other tiers you get OSS support. 70](#_Toc147861883)

[24.7 Enable autoscale for the apps with Standard consumption & dedicated plan 71](#_Toc147861884)

[25 RedHat OpenShift (ARO) 72](#_Toc147861885)

[25.1 Current ARO SLA - 99.95 72](#_Toc147861886)

[25.2 Leverage Availability Zones where regionally applicable. This is automatic when creating a cluster from the portal if the region supports zones. 72](#_Toc147861887)

[25.2.1 ARO clusters require a minimum of 3 control plane nodes (aka master) and 3 worker nodes (worker nodes expandable). Theese nodes will be spread across zones if applicable. An ARO cluster will minimally have 3 control plan and 3 worker nodes when created. 72](#_Toc147861888)

[25.3 Run user workloads on the worker nodes, not the control plane nodes. 72](#_Toc147861889)

[25.4 Isolate workloads into worker nodes running in individual subnets as needed 72](#_Toc147861890)

[25.4.1 Create infrastructure machine sets to hold infrastructure components. Apply specific Kubernetes labels to these machines and then update the infrastructure components to run on only those machines. 72](#_Toc147861891)

[25.5 Use Kubernetes resource governance (Requests, Limits) for memory and cpu to ensure proper pod scheduling. 72](#_Toc147861892)

[25.5.1 Use the Vertical Pod Autoscaler to help here 72](#_Toc147861893)

[25.6 Use pod disruption budgets and multiple replicas to ensure enough replicas to handle load. 72](#_Toc147861894)

[25.7 Storage 72](#_Toc147861895)

[25.7.1 Backup a cluster state for stateful workload scenarios to a paired region 72](#_Toc147861896)

[25.7.2 If container storage is required, ensure availability across regions if needed: Using RWX storage with inbuilt Azure Files storage class. Using CSI Drivers for storage provisioning. 73](#_Toc147861897)

[25.7.3 Whenever possible, move state out of containers and into external databases that support multi-region replication. Avoid Persistent Volumes. 73](#_Toc147861898)

[25.8 Use a geo-replicated container registry like Azure Container Registry or other 73](#_Toc147861899)

[26 Azure Container Apps 74](#_Toc147861900)

[26.1 Best Practices 74](#_Toc147861901)

[26.1.1 Leverage Availability Zones if regionally applicable 74](#_Toc147861902)

[26.1.2 For cross-region DR, deploy container apps in multiple regions and follow active/active or active/passive application guidance. 74](#_Toc147861903)

[27 IoT 75](#_Toc147861904)

[27.1 IoT Hub 75](#_Toc147861905)

[27.1.1 Leverage Availability Zones if regionally applicable (this is automatically enabled) 75](#_Toc147861906)

[27.1.2 Cross region DR 75](#_Toc147861907)

[27.2 DPS (Device Provisioning Service) 76](#_Toc147861908)

[27.2.1 Best Practices 76](#_Toc147861909)

[27.3 IoT Device Update 77](#_Toc147861910)

[27.3.1 Best Practices 77](#_Toc147861911)

[27.4 Digital Twins 77](#_Toc147861912)

[27.4.1 Best Practices 77](#_Toc147861913)

[28 SignalR 78](#_Toc147861914)

[29 Service Bus 79](#_Toc147861915)

[29.1 FTA Resiliency Playbook 79](#_Toc147861916)

[29.2 Best Practices 80](#_Toc147861917)

[29.2.1 Leverage Availability Zones if regionally applicable (this will be turned on automatically for a new SB namespace created from the portal with the Premium SKUs in a zone-enabled region) 80](#_Toc147861918)

[29.2.2 Geo-Disaster Recovery 80](#_Toc147861919)

[29.2.3 Partitioning 80](#_Toc147861920)

[29.2.4 Retry Logic is baked into the cake and native to the service 80](#_Toc147861921)

[29.2.5 Best practices for insulating applications against Service Bus outages and disasters 80](#_Toc147861922)

[29.2.6 Checklist 80](#_Toc147861923)

[30 MySQL 82](#_Toc147861924)

[30.1 Best Practices 82](#_Toc147861925)

[30.1.1 Leverage Flexible Server 82](#_Toc147861926)

[30.1.2 Leverage Availability Zones where regionally applicable 82](#_Toc147861927)

[30.1.3 Leverage Data-in replication for cross-region DR scenarios 82](#_Toc147861928)

[31 Event Hubs 83](#_Toc147861929)

[31.1 FTA Resiliency Playbook 83](#_Toc147861930)

[31.2 Best Practices 84](#_Toc147861931)

[31.2.1 Leverage Availability Zones if regionally applicable (this will be turned on automatically for a new EH namespace created from the portal with Premium, Dedicated, or Standard SKUs in a zone-enabled region) 84](#_Toc147861932)

[31.2.2 Use the Premium or Dedicated SKUs for predicable performance 84](#_Toc147861933)

[31.2.3 Geo-Disaster Recovery 84](#_Toc147861934)

[31.2.4 Retry Logic is baked into the cake and native to the service 84](#_Toc147861935)

[31.2.5 Design Resilient Event Hubs 84](#_Toc147861936)

[32 HDInsight 85](#_Toc147861937)

[32.1 Best Practices 85](#_Toc147861938)

[32.1.1 Architectures (Azure HDInsight Business Continuity Architectures) 85](#_Toc147861939)

[33 PostgreSQL 86](#_Toc147861940)

[33.1 Best Practices 86](#_Toc147861941)

[33.1.1 Leverage Flexible Server 86](#_Toc147861942)

[33.1.2 Leverage Availability Zones where regionally applicable 86](#_Toc147861943)

[33.1.3 Leverage cross-region read replicas for BCDR 86](#_Toc147861944)

[33.1.4 Use geo-restore for restoring geo-redundant backups. 86](#_Toc147861945)

[34 OpenAI 87](#_Toc147861946)

[35 Azure Bot Service 88](#_Toc147861947)

[35.1 Best Practices 88](#_Toc147861948)

[35.1.1 Reliability in Azure Bot Service 88](#_Toc147861949)

[35.1.2 Regionalization in Azure Bot Service 88](#_Toc147861950)

[35.1.3 Azure Bot Service runs in active-active mode for both global and regional services. When an outage occurs, you don't need to detect errors or manage the service. Azure Bot Service automatically performs autofailover and auto recovery in a multi-region geographical architecture. For the EU bot regional service, Azure Bot Service provides two full regions inside Europe with active/active replication to ensure redundancy. For the global bot service, all available regions/geographies can be served as the global footprint. 88](#_Toc147861951)

[35.1.4 Architectures 88](#_Toc147861952)

[36 Microsoft Purview 90](#_Toc147861953)

[36.1 Data Governance 90](#_Toc147861954)

[36.1.1 FTA Resiliency Playbook 90](#_Toc147861955)

[36.1.2 Best Practices 90](#_Toc147861956)

[36.1.3 Data Catalog 92](#_Toc147861957)

[36.1.4 Data Map 92](#_Toc147861958)

[36.1.5 Data Sharing 92](#_Toc147861959)

[36.1.6 Data Estate 92](#_Toc147861960)

[36.1.7 Data Quality 93](#_Toc147861961)

[36.1.8 Data Policy 93](#_Toc147861962)

[36.2 Compliance 93](#_Toc147861963)

[37 Fluid Relay 94](#_Toc147861964)

[37.1 Best Practices 94](#_Toc147861965)

[37.1.1 Cross Region Replication 94](#_Toc147861966)

[38 AzureML 95](#_Toc147861967)

[38.1 Best Practices 95](#_Toc147861968)

[38.1.1 Your Code 95](#_Toc147861969)

[38.1.2 Your ML Workspace 95](#_Toc147861970)

[38.1.3 Your Registry 95](#_Toc147861971)

[39 Cognitive Services 96](#_Toc147861972)

[39.1 FTA Resiliency Playbook 96](#_Toc147861973)

[39.2 OpenAI 96](#_Toc147861974)

[39.2.1 BackUp your Prompts 96](#_Toc147861975)

[39.2.2 Business Continuity and Disaster Recovery (BCDR) considerations with Azure OpenAI Service 96](#_Toc147861976)

[39.2.3 Backup your ChatGPT conversations 96](#_Toc147861977)

[39.3 Language 96](#_Toc147861978)

[39.3.1 CI/CD for Custom Speech 96](#_Toc147861979)

[39.3.2 QnA Service 96](#_Toc147861980)

[39.4 Vision 96](#_Toc147861981)

[39.5 Decision 96](#_Toc147861982)

[40 Azure Cognitive Search 97](#_Toc147861983)

[40.1 Best Practices 97](#_Toc147861984)

[40.1.1 High availability 97](#_Toc147861985)

[40.1.2 Leverage Availability Zones by enabling read and/or write replicas 97](#_Toc147861986)

[40.1.3 Geo-replication 97](#_Toc147861987)

[40.1.4 Disaster recovery and service outages 97](#_Toc147861988)

[40.1.5 back up and restore alternatives 98](#_Toc147861989)

[40.1.6 Back up and restore an Azure Cognitive Search index 98](#_Toc147861990)

[41 Stream Analytics 99](#_Toc147861991)

[41.1 FTA Resiliency Playbook 99](#_Toc147861992)

[41.2 Best Practices 100](#_Toc147861993)

[41.2.1 High Availability 99% SLA 100](#_Toc147861994)

[41.2.2 Achieve Geo-Redundancy 100](#_Toc147861995)

[41.2.3 Active/Active Approach 100](#_Toc147861996)

[41.2.4 Acitve/Passive Approach 102](#_Toc147861997)

[42 PowerBI 103](#_Toc147861998)

[42.1 Best Practices 103](#_Toc147861999)

[42.1.1 Power BI is fully managed software as a service (SaaS) 103](#_Toc147862000)

[42.1.2 Power BI maintains multiple instances of each component in Azure datacenters (also known as regions) to guarantee business continuity. If there's an outage, or Power BI becomes inaccessible or inoperable in a region, Power BI fails all its components in hat region to a backup instance. The failover restores availability and operability to the Power BI service instance in a new region usually within the same geographic location. 103](#_Toc147862001)

[42.1.3 Source control your PBIT/PBIX files 103](#_Toc147862002)

[42.1.4 Resiliency 103](#_Toc147862003)

[43 ADX 104](#_Toc147862004)

[43.1 Best Practices 104](#_Toc147862005)

[43.1.1 Leverage External Tables and Continuous data export overview to reduce costs 104](#_Toc147862006)

[43.1.2 Leader-follower cluster configuration 104](#_Toc147862007)

[43.1.3 Create Multiple independent clusters 104](#_Toc147862008)

[43.1.4 DR Configurations 105](#_Toc147862009)

[43.1.5 DevOps 109](#_Toc147862010)

[43.1.6 Design, develop, and implement validation routines to ensure all clusters are in-sync from a data perspective. 109](#_Toc147862011)

[43.1.7 Be fully cognizant of what it takes to build a cluster from scratch. 109](#_Toc147862012)

[44 Data Factory 110](#_Toc147862013)

[44.1 FTA Resiliency Playbook 110](#_Toc147862014)

[44.2 Best Practices 110](#_Toc147862015)

[44.2.1 Use zone redundant pipelines in regions that support Availability Zones 110](#_Toc147862016)

[44.2.2 GitHub/Azure DevOps Integration 110](#_Toc147862017)

[44.2.3 Self Hosted Integration Runtimes 111](#_Toc147862018)

[44.2.4 Network 111](#_Toc147862019)

[44.2.5 Key Vault 113](#_Toc147862020)

[45 Cosmos DB 114](#_Toc147862021)

[45.1 FTA Resiliency Playbook 114](#_Toc147862022)

[45.2 Best Practices 114](#_Toc147862023)

[45.2.1 Leverage Availability Zones where regionally applicable, and of course if the service offers it. 114](#_Toc147862024)

[45.2.2 Run multiple replicas of the database (> than 1) in prod 114](#_Toc147862025)

[45.2.3 Leveerage Multi-Region Writes 114](#_Toc147862026)

[45.2.4 Distribute your data globally 114](#_Toc147862027)

[45.2.5 Automatics Backaups 116](#_Toc147862028)

[46 Databricks 118](#_Toc147862029)

[46.1 FTA Resiliency Playbook 118](#_Toc147862030)

[46.2 Best Practices 118](#_Toc147862031)

[46.2.1 For Databricks you have to worry about backing up and recovering 4 items: 118](#_Toc147862032)

[46.2.2 Disaster Recovery Options 119](#_Toc147862033)

[46.2.3 Simplifying DR with Delta Lake 119](#_Toc147862034)

[46.2.4 Disaster Recovery Overview, Strategies, and Assessment 119](#_Toc147862035)

[46.2.5 Tooling 119](#_Toc147862036)

[47 Synapse Analytics 120](#_Toc147862037)

[47.1 Best Practices 120](#_Toc147862038)

[47.1.1 Enable geo-backup. By default, Synapse Analytics takes a full backup of your data every 24 hours for disaster recovery. 120](#_Toc147862039)

[47.1.2 Azure DevOps/Git Integration 120](#_Toc147862040)

[47.1.3 Synapse Pipelines (See ADF Section Above) 120](#_Toc147862041)

[47.1.4 Create Scripts for all DDL Statements and save in GitHub or Azure DevOps Repos 120](#_Toc147862042)

[47.1.5 If you work with Spark Notebooks make sure to integrate with Git or Azure DevOps 120](#_Toc147862043)

[47.1.6 Dedicated Pools 120](#_Toc147862044)

[47.1.7 Serverless Pools 121](#_Toc147862045)

[47.1.8 ARM Templates 121](#_Toc147862046)

[47.1.9 Make sure to re-establish any Synapse Link(s) 121](#_Toc147862047)

[48 SQL Database 122](#_Toc147862048)

[48.1 Best Practices 122](#_Toc147862049)

[48.1.1 Leverage Availability Zones where regionally applicable, and of course if the service offers it. 122](#_Toc147862050)

[48.1.2 Run Business Critical or Premium tiers for applications that require highest SLA 122](#_Toc147862051)

[48.2 Azure SQL DB 122](#_Toc147862052)

[48.2.1 Backups 122](#_Toc147862053)

[48.2.2 BCDR 123](#_Toc147862054)

[48.2.3 Maintenance Windows 123](#_Toc147862055)

[48.3 SQL MI 123](#_Toc147862056)

[48.3.1 Backups 123](#_Toc147862057)

[48.3.2 Maintenance Windows 123](#_Toc147862058)

[48.4 SQL VM 124](#_Toc147862059)

[48.4.1 Azue BackUp Service 124](#_Toc147862060)

[48.4.2 Availability Groups 124](#_Toc147862061)

[48.4.3 SQL IAAS Extension 124](#_Toc147862062)

[48.4.4 Take a Backup of the Database and save it 124](#_Toc147862063)

[48.4.5 Leverage Premium Disks 124](#_Toc147862064)

[49 Azure Storage 125](#_Toc147862065)

[49.1 Best Practices 125](#_Toc147862066)

[49.1.1 Leverage a storagev2 account type for better performance and reliability 125](#_Toc147862067)

[49.1.2 Data Lake Storage Gen2 125](#_Toc147862068)

[49.1.3 Data Protection Options 125](#_Toc147862069)

# Resiliency Checklist for Services

<https://learn.microsoft.com/en-us/azure/architecture/checklist/resiliency-per-service>

# Key Vault

## FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/paas-foundations-playbooks-AKV_v1.docx>

## Best Practices

### Key Vault is a managed service and Microsoft will handle the failover if a region goes down.

#### You don't have to think about it because its a managed service

#### The contents of your key vault are replicated within the region and to a secondary region at least 150 miles away, but within the same geography to maintain high durability of your keys and secrets.

#### During failover, a customer cannot change access policy or firewall configurations and settings. The vault is read-only during this time.

### Please note: there is no manual-failover.

### When you back up a key vault object, such as a secret, key, or certificate, the backup operation will download the object as an encrypted blob. This blob can't be decrypted outside of Azure. To get usable data from this blob, you must restore the blob into a key vault within the same Azure subscription and Azure geography.

### Leverage Soft Deletes

#### Soft-delete retention period

##### Soft-deleted resources are retained for a set period of time, 90 days

### Purge protection

#### Purge protection is an optional Key Vault behavior and is not enabled by default. Purge protection can only be enabled once soft-delete is enabled. It can be turned on via CLI or PowerShell.

### Backup and restore objects

#### Limitations

##### Key Vault does not support the ability to backup more than 500 past versions of a key, secret, or certificate object. Attempting to backup a key, secret, or certificate object may result in an error. It is not possible to delete previous versions of a key, secret, or certificate.

##### Key Vault doesn't currently provide a way to back up an entire key vault in a single operation. Any attempt to use the commands listed in this document to do an automated backup of a key vault may result in errors and won't be supported by Microsoft or the Azure Key Vault team.

### Architectural Design Considerations

#### There is no syncing keys and secrets to another Key Vault

##### So if you want to maintain two key vaults in-sync you will have to use change management to make sure the keys are created in both key vaults.

#### We generally recommend to have different keys between regions, so global outage when single key is corrupted does not happen

# Web Apps

## FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/paas-foundations-playbooks-webapps-functions_v1.docx>

## Best Practices

### Multi-tenant service

<https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/app-service-web-app/zone-redundant?source=recommendations>

#### Use Premium and Standard tiers. These tiers support staging slots and automated backups.

<https://learn.microsoft.com/en-us/azure/app-service/overview-hosting-plans>

#### Leverage Availability Zones where regionally applicable (requires Premium v2 or v3 tier)

<https://learn.microsoft.com/en-us/azure/reliability/migrate-app-service>

#### Implement health checks

<https://learn.microsoft.com/en-us/azure/app-service/monitor-instances-health-check>

#### Backup/restore of webapps

<https://learn.microsoft.com/en-us/azure/app-service/manage-backup>

#### HA checklist

<https://learn.microsoft.com/en-us/azure/architecture/framework/services/compute/azure-app-service/reliability>

#### DR recovery

<https://learn.microsoft.com/en-us/azure/app-service/manage-disaster-recovery#recover-app-content-only>

### App Service Environment v3

<https://learn.microsoft.com/en-us/azure/app-service/environment/overview>

#### Leverage Availability Zones where regionally applicable

<https://learn.microsoft.com/en-us/azure/reliability/migrate-app-service-environment>

#### Ensure "Always On" is enabled

<https://learn.microsoft.com/en-us/azure/azure-functions/dedicated-plan#always-on>

#### Implement health checks

<https://learn.microsoft.com/en-us/azure/app-service/monitor-instances-health-check>

#### HA checklist

<https://learn.microsoft.com/en-us/azure/architecture/framework/services/compute/azure-app-service/reliability>

#### DR recovery

<https://learn.microsoft.com/en-us/azure/app-service/manage-disaster-recovery#recover-app-content-only>

### Use Application Insights

#### Use Availiability Tests & Ping Tests

<https://learn.microsoft.com/en-us/azure/azure-monitor/app/availability-overview>

##### Standard test

##### Custom TrackAvailability test

##### URL ping test

# APIM

## Best Practices

<https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-disaster-recovery-backup-restore>

### Leverage Availability Zones where regionally applicable

<https://learn.microsoft.com/en-us/azure/reliability/migrate-api-mgt>

### Use the premium tier for production workloads.

<https://learn.microsoft.com/en-us/azure/api-management/upgrade-and-scale#upgrade-and-scale>

### Run APIM with capacity greater than 1

<https://learn.microsoft.com/en-us/azure/api-management/api-management-capacity#use-capacity-for-scaling-decisions>

### For DR, leverage the premium tier with deployments scaled across two or more regions for 99.99% SLA

<https://learn.microsoft.com/en-us/azure/api-management/upgrade-and-scale#upgrade-and-scale>

### Managing the APIM APIs from DevOps

<https://learn.microsoft.com/en-us/azure/api-management/devops-api-development-templates>

#### Why you want to do this you don't have to worry about backing and restoring your APIs by leveraging DevOPs or GitHub

### If using Self-Hosted Gateways

<https://learn.microsoft.com/en-us/azure/api-management/self-hosted-gateway-overview>

#### The scalability is connected to how you manage and host your self-hosted gateway on premise

#### Its your responsibility to manage the scalability

### Be aware of your limits

<https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/azure-subscription-service-limits#api-management-limits>

### Use Policies to add a fail-over backend URL and caching to reduce failing calls.

### Leverage multi-region deployment of API gateways(available only in premium tiers). APIM routes the requests to a regional gateway based on the lowest latency. You could use your own Traffic Manager with custom routing rules .

### In multi-region model, use Policies to route the requests to regional backends based on availability or latency.

# VMs

## Best Practices

### Question(s) you should ask yourself

#### 1. How quickly do you need the system to be back up and running in the event of a disaster? (Recovery Time Objective)

##### An aggressive recovery time objective is 30 minutes.

##### A non-aggressive recovery time is 48 hours.

#### 2. How much data are you able to lose in a disaster? (Recovery Point Objective)

##### An aggressive recovery time objective might be measured in seconds to minutes.

##### A non-aggressive recovery point would be measured in hours, up to days. If you are beyond a day, it is more of a backup than a DR recovery.

#### 3. What are you willing to live with and live without during a disaster? (Recovery Level Objective)

#### 4. How much are you willing to pay?

#### 5. How available does your workload need to be? (Availability/update time SLA)

### Ensure backups are enabled to protect data

<https://learn.microsoft.com/en-us/azure/backup/backup-azure-vms-introduction>

### Leverage DevOps (IaC)

#### Use DevOps to deploy infrastructure to be able to rapidly recover

#### Use DevOps to deploy Workloads (apps, etc) to the infrastructure to be able to rapidly recover

### Single Region Resiliency

#### VM Storage Resiliency

##### Use Premium or Ultra disks for prod VMs

<https://learn.microsoft.com/en-us/azure/virtual-machines/disks-types>

##### Ensure Managed Disks are used

<https://learn.microsoft.com/en-us/azure/virtual-machines/managed-disks-overview>

##### Leverage zone-redundant storage (ZRS) for managed disks

##### Use a data disk for application installations.

###### By using a data disk, you can restore the data disk to a new VM, reducing effort to recover and restore data and VM state. You can use IaC to deploy a new VM and app code, and then attach the data disk and be running as opposed to recreating a specific VM state

###### Do not use the Temp disk for anything that is not acceptable to be lost.

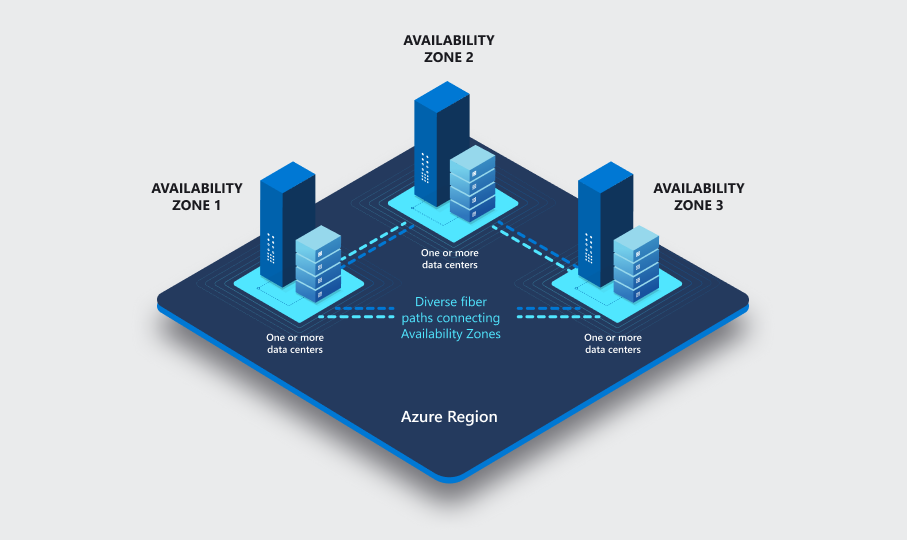
#### VM Compute Resiliency

##### Leverage Availability Zones where regionally applicable

<https://learn.microsoft.com/en-us/azure/virtual-machines/availability>

###### For a given VM role, have at least two VMs in two different availability zones.

###### Availability zones



###### Leverage ZRS storage for VM disks

##### Leverage Availability Sets for redundant VMs if zones not available

<https://learn.microsoft.com/en-us/azure/virtual-machines/availability-set-overview>

###### If you are in a region that does not support Availability Zones, use at least two VMs in Availability Sets to isolate VMs on different fault and update domains.

###### Give each tier its own Availability Set - (Web, App, Data)

#### Single region resiliency has a high availability, but still needs a plan for how resources can be recovered in a disaster.

#### Avoid running a production workload on a single VM.

#### Break VMs into tiers and have redundancy between tiers - do not have a single VM that holds all roles for the application.

### Mulit Region Mindset

#### Costs will go up because you have duplicate environments

#### Leverage Azure Site Recovery for cross-region DR.

<https://learn.microsoft.com/en-us/azure/virtual-machines/availability#azure-site-recovery>

#### Hot/Hot Design - where two active instances of the application are running and have redundant infrastructure. Both can service clients and requests can be loadbalanced between them.

##### This is Front Door / Load Balanced / Always On approach

#### Hot/Cool Design - where there is an active and passive instance of the application in two regions. One handles all client requests, but in a resiliency event, the second can be scaled out and has current data.

##### Front Door / Load Balanced / Active-Passive

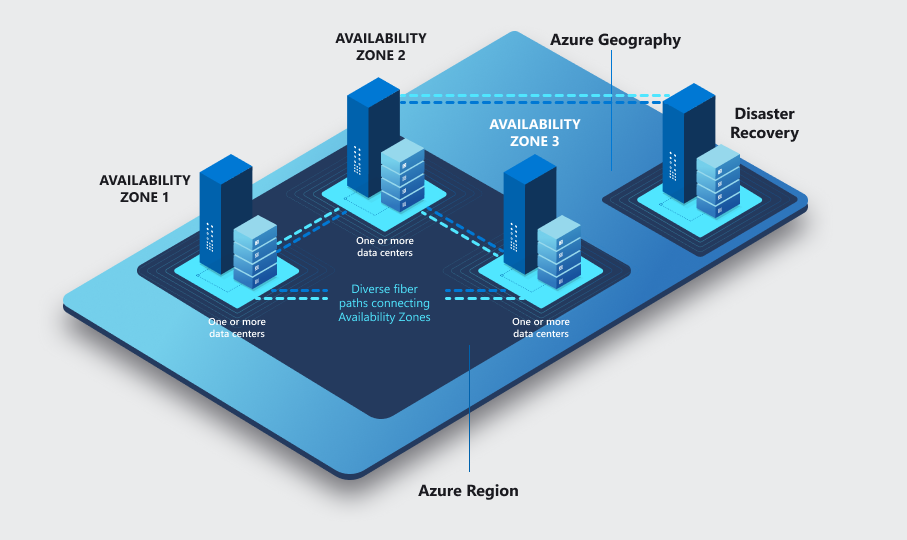
#### Hot/Cold Design - where there is a single instance in the primary region, and replicated instances of VMs in a secondary region. Failover can occur by turning this replicated instances in to active.

##### This is the Azure Site Recovery approach

#### Fast Deploy Design - where you use IaC to rapidly deploy in to another region that already has your data.

##### Requires the data solution to already have replicated data - such as Azure SQL backups or geo-availability, or an existing Always On Availability Group.

### Back up Vs. DR - Related scenarios, but not the same



#### Backup Scenarios are for DATA recovery - information loss protection

##### Malware

##### Accidental Deletion

##### Corruption

##### Regulatory Requirements for data availability

##### Using Azure Backup not to recover a current VM state, but to recover data that is at least a day old, but probably longer.

##### High Recovery Time because the data often has to be restored, sorted through, and then reattached.

##### High Recovery Point Objective because the data is often multiple hours old.

#### Azure Site Recovery are for STATE recovery - restoring both \*compute\* and \*current state\* to an alternative region in the event of a large disaster impacting multiple availability zones (or fault domains)

##### Godzilla Steps on Multiple Azure Data Centers in the same region

##### Multiple power outages lasting longer than on-site power backups can provide

##### Specific Azure Service no longer available in the region due to infrastructure issue

##### Customer specific infrastructure outage (such as third party NVAs deployed to that region)

##### Constant and current replication of VM state, able to be powered on when needed.

##### Low Recovery Time because the VM is already there in a "cold mode" so VM is there.

##### Low Recovery Point Objective because the VM is already there is a "cold mode" so data is current to active replication.

### Plan for Paired Regions

#### Azure regions exist in pairs to provide high availability and resiliency for Azure services.

#### For example, Globally Redundant storage tiers replicate their data to a secondary region where it available in the event of a disaster.

#### These regions are significantly isolated to reduce the chance that both would be down to the same type of large environmental issue (thunderstorms, tornados, etc)

#### Using a Paired Region can help ensure that data is available when you are recovering.

### Plan for Reserved Capacity

#### If there is an outage, others may be trying to deploy to the same region and there may be capacity issues.

#### Reserved Capacity means that you have guaranteed capacity to deploy to.

## VMSS

### Best Practices

#### Leverage Availability Zones if regionally applicable for redundant VMs

<https://learn.microsoft.com/en-us/azure/virtual-machine-scale-sets/overview#why-use-virtual-machine-scale-sets>

#### Ensure Managed Disks are used

<https://learn.microsoft.com/en-us/azure/virtual-machines/managed-disks-overview>

#### Leverage multiple fault domains for regional (non-zonal) VMSS

<https://learn.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-manage-fault-domains>

#### Use Premium or Ultra disks for prod VMs

<https://learn.microsoft.com/en-us/azure/virtual-machines/disks-types>

#### Leverage at least 2 VMs for prod workloads

#### Virtual Machine Scale Sets has the ability to heal

##### VMSS can be set to have a minimum number of functioning nodes

##### If an issue occurs with a node's infrastructure and the node is no longer available, this can trigger the deployment of a new node, making sure that the same number of nodes are accessible

##### Workloads need to be designed with this scaling in mind - application code needs to be able to be deployed, and applications need to be able to operate with nodes being created and destroyed based on availability and performance

#### VMSS are multi-nodes and so Load Balancing is a consideration (see AppGw or LBs)

# AKS

## Best Practices

### Set an Uptime SLA

<https://learn.microsoft.com/en-us/azure/aks/free-standard-pricing-tiers>

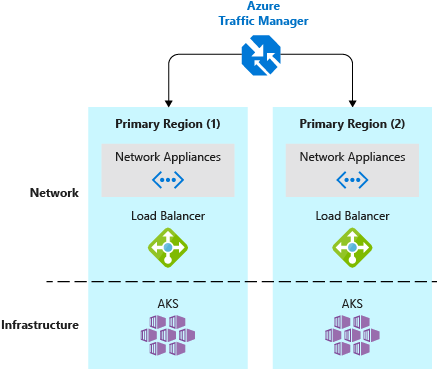
### Leverage Availability Zones where regionally applicable

<https://learn.microsoft.com/en-us/azure/aks/availability-zones>

### Use ZRS storage with stateful workloads

<https://learn.microsoft.com/en-us/azure/aks/availability-zones#azure-disk-availability-zone-support>

### Plan for multi-region deployments



<https://learn.microsoft.com/en-us/azure/aks/operator-best-practices-multi-region#plan-for-multiregion-deployment>

#### Sample Architecture(s)

<https://learn.microsoft.com/en-us/hybrid/app-solutions/pattern-highly-available-kubernetes>

### Use System and User node pools

<https://docs.microsoft.com/en-us/azure/aks/use-multiple-node-pools>

#### Using node pools will split cluster responsibilities and distribute the workload between nodes.

#### Don't run user containers on system node pools

#### Enable cluster auto-scaler

<https://learn.microsoft.com/en-us/azure/aks/cluster-autoscaler>

#### Ensure the capacity of the node pools is at least 5 nodes for prod clusters (3 for System nodepool, 2 or more for User nodepool)

<https://learn.microsoft.com/en-us/azure/aks/use-system-pools?tabs=azure-cli#system-and-user-node-pools>

### Kubernetes best practices

#### Configure Liveness and Readines probes for all deployments

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/>

#### Set pod disruption budgets

<https://docs.microsoft.com/en-us/azure/aks/operator-best-practices-scheduler#plan-for-availability-using-pod-disruption-budgets>

#### Set pod requests and limits

<https://docs.microsoft.com/en-us/azure/aks/developer-best-practices-resource-management#define-pod-resource-requests-and-limits>

#### Set resource quotas on namespaces

<https://docs.microsoft.com/en-us/azure/aks/operator-best-practices-scheduler#enforce-resource-quotas>

#### Leverage Kubernetes' Horizontal Pod Autoscaler (HPA); automated metric-based scaling or vertical scaling by sizing the container instances (cpu/memory).

#### Enable geo-replication for container images

<https://learn.microsoft.com/en-us/azure/container-registry/container-registry-geo-replication>

#### Leverage application routing with Azure Front Door Service

### Use Application Insights

#### Use Availiability Tests & Ping Tests

##### Standard test

##### Custom TrackAvailability test

##### URL ping test

# Functions

## FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/paas-foundations-playbooks-webapps-functions_v1.docx>

## Best Practices

### Use an Azure Functions Premium Plan

<https://learn.microsoft.com/en-us/azure/azure-functions/functions-premium-plan>

#### Leverage Availability Zones where regionally applicable

<https://learn.microsoft.com/en-us/azure/reliability/reliability-functions?toc=%2Fazure%2Fazure-functions%2FTOC.json&tabs=azure-portal#availability-zone-support>

#### Cross-region DR guidance

<https://learn.microsoft.com/en-us/azure/azure-functions/functions-geo-disaster-recovery>

### Use an App Service Environment v3

<https://learn.microsoft.com/en-us/azure/app-service/environment/overview>

#### Ensure "Always On" is enabled

<https://learn.microsoft.com/en-us/azure/azure-functions/dedicated-plan#always-on>

#### Leverage Availability Zones where regionally applicable

<https://learn.microsoft.com/en-us/azure/reliability/migrate-app-service-environment>

### Leverage App Service Plans if you want to control the scale up and scale out yourself.

### Pair a Function App to its own storage account. Try not to re-use storage accounts for Function Apps unless they are tightly coupled.

### Leverage DevOps or GitHub to streamline CI/CD and safeguard your Function App code

<https://learn.microsoft.com/en-us/training/modules/deploy-azure-functions/>

# Logic Apps

## FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/paas-foundations-playbooks-logic_apps_v1.docx>

## Best Practices

### Use Logic App Standard -- Single Tenant

#### Leverage Availability Zones where regionally applicable

#### This tier is a special tuned App Service Plan for Azure Logic Apps that provides a semblance of a consumption model and eliminates cold-starts. This plan offers “Elastic Scale Out” that provides instant scalability without cold starts and is the same runtime offered by the elastic Azure Functions Premium plan.

#### Vnet integration is supported with Private Link for internal-only usage

### Use Logic Apps Standard -- App Service Environment v3 (Windows plans only)

#### Leverage Availability Zones where regionally applicable

#### This is a special “dedicated” plan that can host Azure Web Apps, Azure Functions, and Logic Apps.

#### Fully isolated inside a virtual network and offers the advantage of higher numbers of instances when compared to the Logic Apps Standard – Single Tenant.

### For a resiliency against full-region outages, the recommendation is to setup a logic apps in multiple regions.

#### Deploy multiple Logic Apps in different regions

#### Keep the Logic Apps workflows in sync by deploying to both via a CI/CD process

#### Balance traffic across the Logic Apps in an active/active or active/passive manner, depending on the application

#### If the primary Logic App suffers losses, disruptions, or failures, the secondary Logic App will take on the work.

# Azure Firewall

## Best Practices

### Leverage Availability Zones where regionally applicable

<https://learn.microsoft.com/en-us/azure/firewall/deploy-availability-zone-powershell>

# Front Door

## FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/paas-foundations-playbooks-FrontDoor_v1.docx>

## Best Practices

### An important point here is that having multiple instances of AFD IS NOT NEEDED FOR ADDITIONAL RESILIENCY unless the customer’s needs exceed the defined limits (number of origin groups, number of origins, number of endpoints, etc.).

### Automatic Failover

#### This is a global managed service which Microsoft manages

#### As a best practice, create a health probe path in your application backend that reports the overall health of the application. This health probe should check critical dependencies such as the compute services and storage services running the app (i.e., AKS, SQL Database, Azure Storage, etc.). Otherwise, the probe might report a healthy backend when critical parts of the application are failing.

### Manual Failover

#### If 99.99% SLA isnt good enough and you want to protect against a regional Front Door outage

##### If the Front Door service fails, change your canonical name (CNAME) records in DNS to point to another traffic management service. This step must be performed manually, and your application will be unavailable until the DNS changes are propagated. Azure Traffic Manager is one such option that can be used as a fallback for this situation.

### Leverage Health Probes

#### To determine the health and proximity of each backend for a given Azure Front Door environment, each Front Door environment periodically sends a synthetic HTTP/HTTPS request to each of your configured origins. Azure Front Door then uses these responses from the probe to determine the "best" origin to route your client requests.

#### Determining Health

#### Measuring Latency

# Azure Load Balancer

## Best Practices

### Leverage the Standard Load Balancer sku

<https://learn.microsoft.com/en-us/azure/load-balancer/skus>

### Leverage Availability Zones where regionally applicable

<https://learn.microsoft.com/en-us/azure/load-balancer/load-balancer-standard-availability-zones>

# App Gateway

## Best Practices

### Leverage Availability Zones where regionally applicable

<https://learn.microsoft.com/en-us/azure/application-gateway/application-gateway-autoscaling-zone-redundant>

### Leverage the Standard\_v2 and Waf\_v2 SKUs

<https://learn.microsoft.com/en-us/azure/application-gateway/overview-v2>

### Deploy Application Gateway with at least two instances

# Redis Cache

## Best Practices

### High availability and disaster recovery

### Zone redundancy

#### Azure Cache for Redis supports zone redundant configurations in the Premium and Enterprise tiers. A zone redundant cache can place its nodes across different Azure Availability Zones in the same region. It eliminates data center or AZ outage as a single point of failure and increases the overall availability of your cache. See this article for information on how to set it up.

### Leverage Redis Persistence

#### Because your cache data is stored in memory, a rare and unplanned failure of multiple nodes can cause all the data to be dropped. To avoid losing data completely, Redis persistence allows you to take periodic snapshots of in-memory data, and store it to your storage account.

### Use Geo-redundant storage account or zonally redundant where geo is not available

### Passive Geo-replication

#### Geo-replication is a mechanism for linking two or more Azure Cache for Redis instances, typically spanning two Azure regions. Geo-replication is designed mainly for cross-region disaster recovery. Two Premium tier cache instances are connected through geo-replication in a way that provides reads and writes to your primary cache, and that data is replicated to the secondary cache.

# Azure Monitor Logs

# Azure Chaos Studio

<https://learn.microsoft.com/en-us/shows/azure-friday/an-introduction-to-azure-chaos-studio>

## Resilience through chaos engineering

<https://learn.microsoft.com/en-us/azure/chaos-studio/chaos-studio-overview>

### Is a managed service that uses chaos engineering to help you measure, understand, and improve your cloud application and service resilience

### Improve application resilience with chaos testing by deliberately introducing faults that simulate real-world outages.

### Disrupt your apps intentionally to identify gaps and plan mitigations before your customers are impacted by a problem.

## Common Chaos Engineering Scenarios

### Reproduce an incident that affected your application, to better understand the failure. Ensure that post-incident repairs prevent the incident from recurring.

### Prepare for a major event or season with "game day" load, scale, performance, and resilience validation.

### Do business continuity and disaster recovery (BCDR) drills to ensure that your application can recover quickly and preserve critical data in a disaster.

### Run high availability (HA) drills to test application resilience against region outages, network configuration errors, high stress events, or noisy neighbor issues.

### Plan capacity needs for production environments.

### Run stress tests or load tests.

### Build confidence in services built on cloud-native architectures.

# Hybrid Connectivity

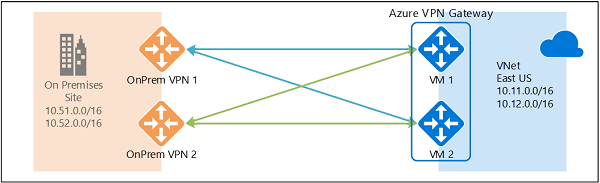
## Best Practices

### VPN Gateway and Express Route DR Best Practices will apply to Hub & Spoke and VWan Architectures

### VPN Gateways

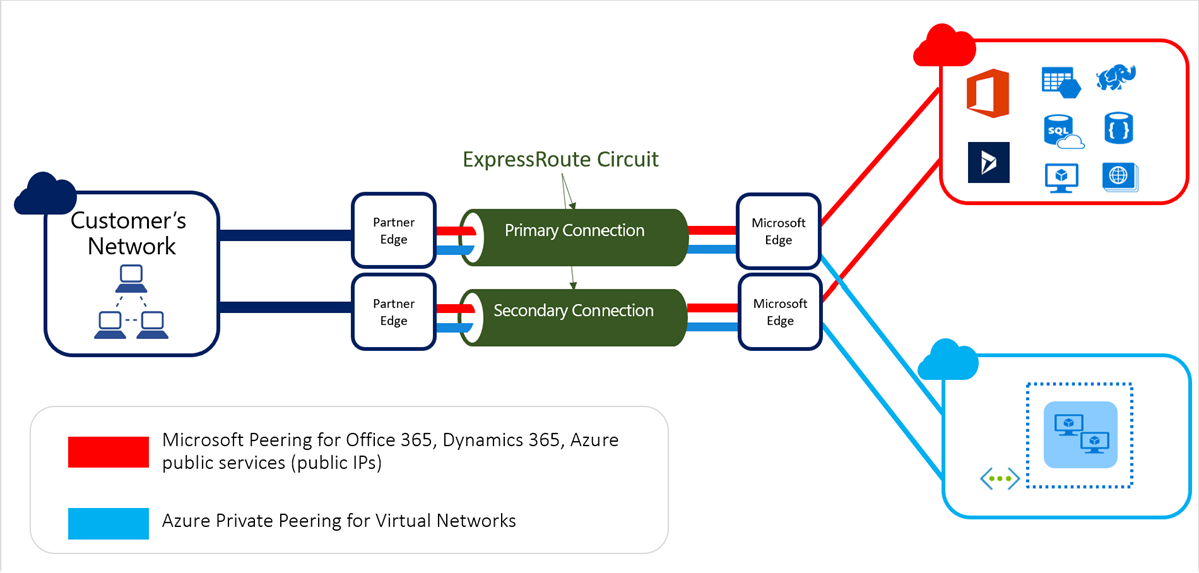
#### Deploy Gateways in to multiple Availability Zones for resiliency

#### Redundancy in Azure is only half of the consideration - the on-prem side of the VPN needs to also be resilient



### ExpressRoutes

#### ExpressRoute Circuits

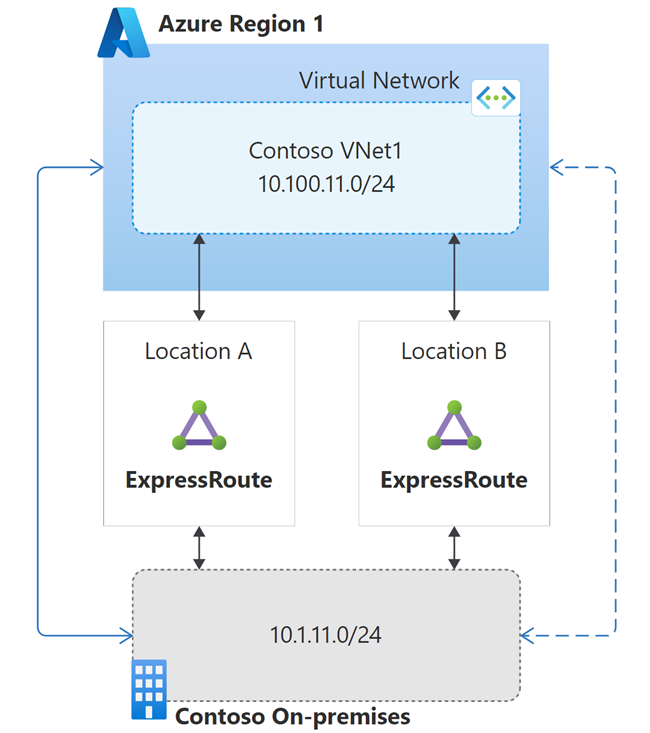


##### Each ExpressRoute Circuit contains two connections

##### Having multiple ExpressRoute Circuits add additional redundant connection pairs

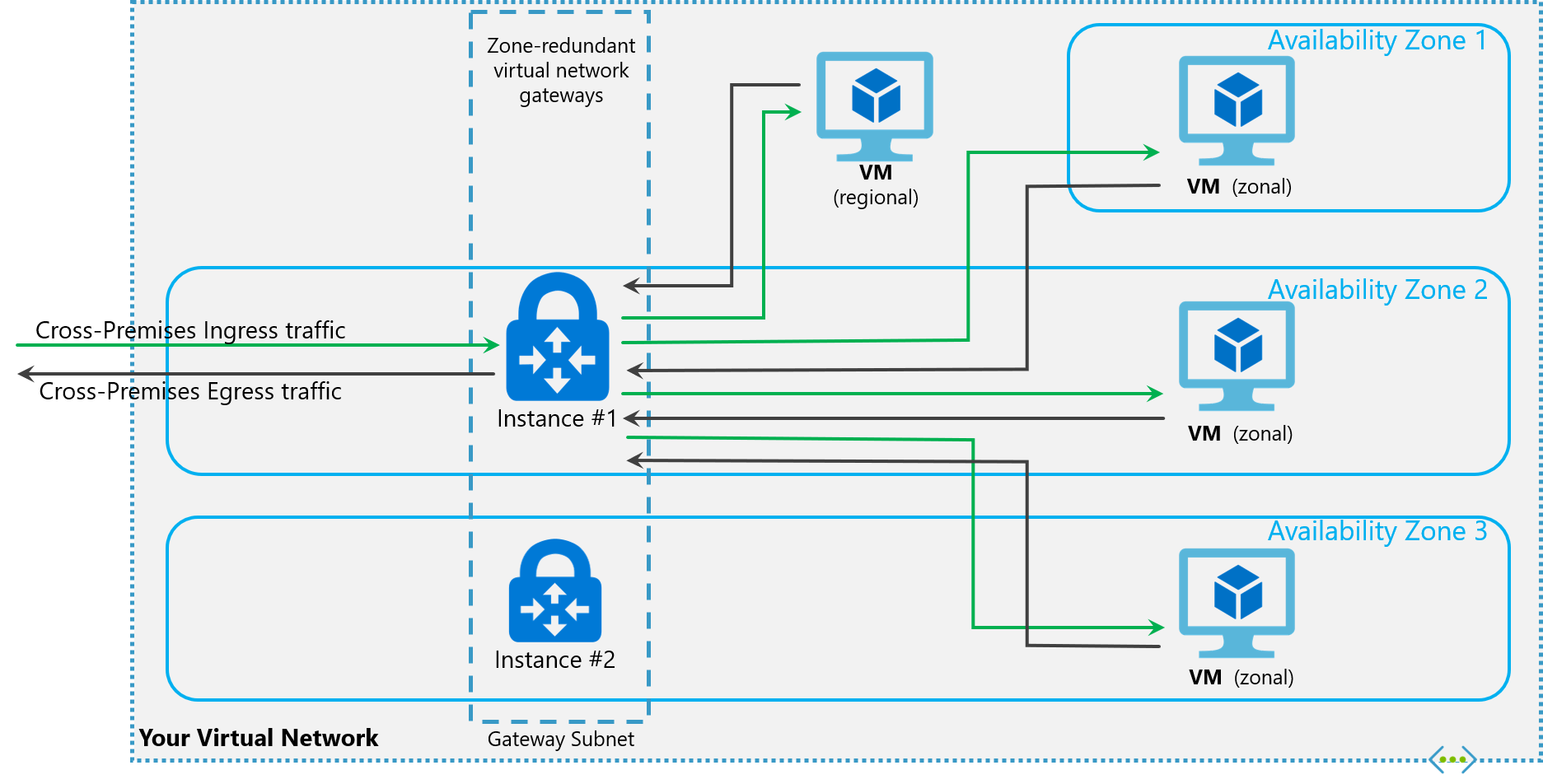
##### You can use VPN as a backup for ExpressRoute.

##### Routing for Multiple Circuits



###### Be aware that you can leverage Connection Weights. You can leverage Connection Weights to ensure their is traffic symmetry.

#### ExpressRoute Gateway



##### Deploy Gateways in to multiple Availability Zones for resiliency

### Other

#### Data Gateway

##### Leverage data gateway cluster to avoid single points of failure and to load balance traffic across gateways in a cluster

# FastTrack for Azure Resiliency Review Checkist

## FastTrack for Azure engineers have compiled a Excel checklist of all the relevant guidance and documentation for resiliency across all supported services.

# Azure Advisor Reliability Workbook (s)

idea_image_84403879_30x30.png

## The objective of this workbook is to provide an overview of the configured state of deployment options that affect the resilience and/or availability SLA for common Azure resource types.

### Building a reliable application in the cloud is different from traditional application development. While historically you may have purchased levels of redundant higher-end hardware to minimize the chance of an entire application platform failing, in the cloud, we acknowledge up front that failures will happen. Instead of trying to prevent failures altogether, the goal is to minimize the effects of a single failing component.

### Reliable applications are

#### Highly available (HA) and run as designed in a healthy state with no significant downtime.

#### Highly available (HA) and run as designed in a healthy state with no significant downtime.

### Understanding how these elements work together — and how they affect cost — is essential to building a reliable application. It can help you determine how much downtime is acceptable, the potential cost to your business, and which functions are necessary during a recovery.

# Azure DevOps

## Best Practices

### Azure DevOps

#### This is a SaaS service so Microsoft takes care of the redundancy

#### Pick your primary region wisely

#### Hosted Agents

##### Use VM scale sets or AKS build agents so that you can build on load rather than just run 1 VM.

###### Why? Because if your 1 VM goes down you won't be able to build

#### Data Protection Agreement

### GitHub

#### GitHub Enterprise Cloud is a SaaS service and redundancy is taken care of by GitHub

#### GitHub Data Protection Agreement

# Recovery Services Vaults

## Azure Site Recovery

### On-premises VM Replication (Hyper-V, VMware or Physical)

#### Capacity planning is required to make sure you have sufficient bandwidth for replication and an estimated number of CPU cores & disk types that will be needed in Azure for failover.

### Azure VM Replication

#### Use Azure Policy to ensure that all critical Azure VMs are protected with ASR.

### Define recovery plans to automate the failover sequence for VMs. You can also include automation scripts to reduce manual steps and improve recovery time.

## Azure Backup

### Enable and LOCK immutability for vaults. This ensures recovery points cannot be deleted before their intended expiry.

### Enable "Always-on soft delete" for vaults protecting critical workloads.

## When creating Recovery Service Vaults choose the best storage redundancy option for your requirements. Vaults support local, geo and zone redundancy but this setting cannot be changed once the vault is protecting one or more resources.

# Network Virtual Appliances

## Best Practices

### Top questions you should be thinking about

#### What is your NVA Hosting?

##### Firewall

##### VPN

##### Load Balancer

##### SD WAN Service

#### What is your expected traffic?

##### East/West Traffic

##### North/South Traffice

###### Internet & Azure

###### OnPrem & Azure

### You want at least two NVAs

### More nodes is more resilient, but more nodes is more money.

### Methods

#### Active-Active

#### Active-Passive

### HA architectures

#### Azure Load Balancer

#### Azure Route Server

#### Gateway Load Balancer

#### Changing PIP/UDR

### NVAs are often critical resources shared between many different workloads, so their resiliency impacts the resiliency of many different workloads

# Identity

## Azure AD

### Azure AD Connect needs to be planned for resiliency

### This is a global managed service with multiple levels of internal redundancy and automatic recoverability which Microsoft manages

#### You don't have to do anything about Azure AD.

#### However, any service or functionality that connects to Azure AD you do have to worry about.

### Use long-live revocable token, cache your token and acquire your silently using Microsoft Identity Library

## Active Directory B2C

### FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/paas-foundations-playbooks-B2C_v1.docx>

### B2C is based on the underlying Azure AD infrastructure. For durability, this infrastructure will replicate any piece of data written to Azure AD at least 4 times and up to 13 datacenters, within a geography, depending on the tenant configuration. The physical location of the replicas is hidden from the customer and is a detail of the global service.

### This is a global managed service which Microsoft manages. Like AAD, B2C provides automatic failover in the event of an unexpected issue with the service or with the Azure region that is physically running an active replica.

#### You don't have to do anything

### Reduce auth calls to AAD B2C

### Reduce external API dependencies

### User Sign-In Flow(s)

#### Make sure that your sign-in user flows are backed up and resilient. Make sure that the code that you use to sign-in your users are backed up and recoverable.

##### Resilient interfaces with external processes

#### Leverage CI/CD and source control for these user flows.

### Build resilience in your customer identity and access management with Azure Active Directory B2C

### Be aware of how your B2C setup affects your composite SLA

#### SLA(s) for multi-region deployments

### Custom brand assets should be hosted on a CDN

### Have multiple identiy providers (i.e., login with your microsoft, google, facebook accounts)

## Windows Server AD

### Follow VM rules for high availability on the VM level (premium disks, two or more in a region, in different availability zones)

### Don't replicate! Replication can create issues with directory synchronization.

### Have active-active for multi-regions

## Azure AD Domain Services

### Add Azure AD Domain service stamps to additional regions and locations

### Use Replica Sets for DR

# Domain Naming Service

## Azure Private DNS

### Private DNS Zones are multi-region by default

### Verify that Zones are linked to Vnets in multiple regions

### If different Zones are used between regions, verify a plan for making sure that Zones are up to date in a DR failover situation

## Azure DNS

### Multi-region by default

### Plan for disaster recovery with Azure DNS and Traffic Manager

## Azure DNS Resolver

### Enable availability zones with Private Resolver

### Plan for failover with Private Resolvers in a Disaster Recovery

## VM Based DNS Services (AD, Infoblox, etc.)

### Follow VM Guidance

### IF AD based DNS, follow the Identity -> Windows Server AD path

### IF a third party, follow the vendor's specific guidance

# Azure Spring Apps

## Deploy zone redundant ASA instance

### In supported region, Azure spring Apps can be deployed as zone redundant, which means that instances are automatically distributed across availability zones. This feature is only available in Standard and Enterprise tiers.

## Use Blue-Green deployment pattern for the apps

### Azure Spring Apps permits two deployments for every app, only one of which receives production traffic. You can achieve zero downtime with blue green deployment strategies. Blue green deployment is only available in Standard and Enterprise tiers. You could automate deployment using CI/CD with ADO/GitHub actions

## Set up alerts and dashboards for monitoring and notification

### Monitor Azure spring apps with logs, metrics and tracing. Integrate ASA with application insights and track failures and create workbooks.

## Create multi region model for high availability and disaster recovery

### Azure Spring Apps instances could be created in multiple regions for your applications and traffic could be routed by Traffic Manager/Front Door.

## Enable VMware Spring Cloud Gateway(Enterprise tiers)

### Set up autoscaling in Spring Cloud Gateway

## Use Enterprise plan for commercial support of spring boot for mission critical apps. With other tiers you get OSS support.

## Enable autoscale for the apps with Standard consumption & dedicated plan

# RedHat OpenShift (ARO)

## Current ARO SLA - 99.95

## Leverage Availability Zones where regionally applicable. This is automatic when creating a cluster from the portal if the region supports zones.

### ARO clusters require a minimum of 3 control plane nodes (aka master) and 3 worker nodes (worker nodes expandable). Theese nodes will be spread across zones if applicable. An ARO cluster will minimally have 3 control plan and 3 worker nodes when created.

## Run user workloads on the worker nodes, not the control plane nodes.

## Isolate workloads into worker nodes running in individual subnets as needed

### Create infrastructure machine sets to hold infrastructure components. Apply specific Kubernetes labels to these machines and then update the infrastructure components to run on only those machines.

## Use Kubernetes resource governance (Requests, Limits) for memory and cpu to ensure proper pod scheduling.

### Use the Vertical Pod Autoscaler to help here

## Use pod disruption budgets and multiple replicas to ensure enough replicas to handle load.

## Storage

### Backup a cluster state for stateful workload scenarios to a paired region

### If container storage is required, ensure availability across regions if needed: Using RWX storage with inbuilt Azure Files storage class. Using CSI Drivers for storage provisioning.

### Whenever possible, move state out of containers and into external databases that support multi-region replication. Avoid Persistent Volumes.

## Use a geo-replicated container registry like Azure Container Registry or other

# Azure Container Apps

## Best Practices

### Leverage Availability Zones if regionally applicable

#### Use more than one replica and enable Zone Redundancy.

#### Note: dedicated tier does not currently support zone redundancy

### For cross-region DR, deploy container apps in multiple regions and follow active/active or active/passive application guidance.

#### Similar guidance as other container solutions using Front Door or Traffic Manager.

# IoT

## IoT Hub

### Leverage Availability Zones if regionally applicable (this is automatically enabled)

<https://learn.microsoft.com/en-us/azure/iot-hub/iot-hub-ha-dr#availability-zones>

### Cross region DR

<https://learn.microsoft.com/en-us/azure/iot-hub/iot-hub-ha-dr#cross-region-dr>

#### Microsoft-initiated failover

<https://learn.microsoft.com/en-us/azure/iot-hub/iot-hub-ha-dr#microsoft-initiated-failover>

##### Microsoft-initiated failover is exercised by Microsoft in rare situations to fail over all the IoT hubs from an affected region to the corresponding geo-paired region.

#### Active/Active

##### Achieve cross region HA

###### You can incorporate a secondary IoT hub and device routing logic to create an Active/Active Scenario

If your IoT Hub service in your primary region is disrupted, devices must start connecting to your secondary region. Given the state-aware nature of most services involved, it's common for solution administrators to trigger the inter-region failover process.

#### Active/Passive

##### Manual failover

<https://learn.microsoft.com/en-us/azure/iot-hub/iot-hub-ha-dr#manual-failover>

###### Step-by-step instructions

<https://learn.microsoft.com/en-us/azure/iot-hub/tutorial-manual-failover>

###### Failback

<https://learn.microsoft.com/en-us/azure/iot-hub/iot-hub-ha-dr#failback>

## DPS (Device Provisioning Service)

### Best Practices

#### Use multiple replicas

##### Azure DPS allows you to provision your devices across multiple replicas for higher availability. Distributing your devices across multiple replicas helps prevent a single point of failure and ensures that provisioning requests can be processed even if one replica experiences issues.

#### Leveage availability zones where regionally applicable (automatic if the region supports zones)

#### Regional redundancy

##### Deploy your Azure DPS instances across multiple Azure regions to achieve geographic redundancy. This helps ensure that your provisioning service remains available even in the event of a regional outage.

#### Enable automatic failover

##### Configure Azure DPS with automatic failover to enable seamless failover to a secondary replica in case of an outage. This ensures minimal disruption to the provisioning process and improves the overall resiliency of your solution.

###### By default, DPS leverages cross-region replication to enable automatic failover. Microsoft-initiated failover is exercised by Microsoft in rare situations when an entire region goes down to fail over all the DPS instances from the affected region to its corresponding secondary region.

#### Monitoring and alerts:

##### Set up monitoring and alerts to receive notifications about the health and performance of your Azure DPS instances. This allows you to proactively address any issues and take necessary actions before they impact your provisioning service.

#### Regular backups

##### It's essential to regularly back up the configuration and state of your Azure DPS instances. This can be achieved by leveraging Azure Backup or other backup mechanisms provided by Azure services. Having backups ensures that you can restore your DPS configuration and device data in case of accidental deletions or data loss.

## IoT Device Update

### Best Practices

#### In cases where an Azure region is unavailable due to an outage, Device Update for IoT Hub supports business continuity and disaster recovery (BCDR) efforts with regional failover pairings. During an outage, data contained in the update files submitted to the Device Update service may be sent to a secondary Azure region.

## Digital Twins

### Best Practices

#### Intra-region HA

##### Built-in redundancy to deliver on uptime of the service

#### Cross region DR

##### Failover to a geo-paired Azure region if there's an unexpected data center failure

# SignalR

# Service Bus

## FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/paas-foundations-playbooks-ASB_v1.docx>

## Best Practices

### Leverage Availability Zones if regionally applicable (this will be turned on automatically for a new SB namespace created from the portal with the Premium SKUs in a zone-enabled region)

#### Both the Service Bus metadata and the messages data are replicated across datacenters in the availability zones configuration

### Geo-Disaster Recovery

#### Metadata replication

##### If enabled, Implements namespace metadata replication to a secondary region. Does not replicate queue/topic message data. Premium sku only.

#### Message replication

##### If an outage cannot be tolerated, do not use the build-in metadata replication option.

##### Leverage a replication pattern to replicate Service Bus messages across two or more sets of cross-region namespaces

### Partitioning

#### Azure Service Bus uses a message broker to handle messages that are sent to a Service Bus queue or topic. By default, all messages that are sent to a queue or topic are handled by the same message broker process. This architecture can place a limitation on the overall throughput of the message queue. However, you can also partition a queue or topic when it is created.

### Retry Logic is baked into the cake and native to the service

### Best practices for insulating applications against Service Bus outages and disasters

### Checklist

#### Evaluate Premier-tier benefits of Azure Service Bus.

#### Ensure that Service Bus Messaging Exceptions are handled properly.

#### Connect to Service Bus with the Advanced Messaging Queue Protocol (AMQP) and use Service Endpoints or Private Endpoints when possible.

#### Review the Best Practices for performance improvements using Service Bus Messaging.

#### Implement geo-replication on the sender and receiver side to protect against outages and disasters.

#### Configure Geo-Disaster.

#### If you need mission-critical messaging with queues and topics, Service Bus Premium is recommended with Geo-Disaster Recovery.

#### Configure Zone Redundancy in the Service Bus namespace (only available with Premium tier).

#### Implement high availability for the Service Bus namespace.

#### Ensure related messages are delivered in guaranteed order.

#### Evaluate different Java Messaging Service (JMS) features through the JMS API.

#### Use .NET Nuget packages to communicate with Service Bus messaging entities.

#### Implement resilience for transient fault handling when sending or receiving messages.

# MySQL

## Best Practices

### Leverage Flexible Server

<https://learn.microsoft.com/en-us/azure/mysql/flexible-server/overview>

### Leverage Availability Zones where regionally applicable

<https://learn.microsoft.com/en-us/azure/mysql/flexible-server/overview#high-availability-within-and-across-availability-zones>

### Leverage Data-in replication for cross-region DR scenarios

<https://learn.microsoft.com/en-us/azure/mysql/flexible-server/overview#setup-hybrid-or-multi-cloud-data-synchronization-with-data-in-replication>

# Event Hubs

## FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/paas-foundations-playbooks-AEH_v1.docx>

## Best Practices

### Leverage Availability Zones if regionally applicable (this will be turned on automatically for a new EH namespace created from the portal with Premium, Dedicated, or Standard SKUs in a zone-enabled region)

#### Both the EH metadata and the event data itself are replicated across zones

### Use the Premium or Dedicated SKUs for predicable performance

### Geo-Disaster Recovery

#### Active/Passive

##### Metadata replication for the Event Hubs. Event data is not replicatedd

##### The built-in geo-disaster recovery feature, when enabled, ensures that the entire configuration of anamespace (Event Hubs, Consumer Groups and settings) is continuously replicated from a primary namespace to a secondary namespace, and it allows a once-only failover move from the primary to the secondary at any time.

##### Feature is designed to make it easier to recover from and abandon a failed Azure region without having to change application configurations

#### Active/Active

##### Enables active/active replication and data merge

##### Should be used for DR configurations where an outage or loss of event data in the downed region cannot be tolerated. For these cases, follow the linked replication guidance and do not use the built-in geo-disaster recovery capability (active/passive).

##### Maintain multiple Event Hubs in different regions and namespaces, and events will be replicated between the hubs

### Retry Logic is baked into the cake and native to the service

### Design Resilient Event Hubs

# HDInsight

## Best Practices

### Architectures (Azure HDInsight Business Continuity Architectures)

#### Apache Hive & Interactive Query

##### Hive active primary with on-demand secondary

##### Hive active primary with standby secondary

#### Apache Spark

##### Spark active primary with on-demand secondary

##### Spark active primary with standby secondary

#### Apache HBase

##### HBase Replication: Leader – Follower model

##### HBase Replication: Leader – Leader model

##### HBase Replication: Multi-Region or Cyclic

#### Apache Kafka

##### Kafka Replication: Active – Passive

##### Kafka Replication: Active – Active

# PostgreSQL

## Best Practices

### Leverage Flexible Server

<https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/overview>

### Leverage Availability Zones where regionally applicable

<https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/overview#architecture-and-high-availability>

### Leverage cross-region read replicas for BCDR

<https://learn.microsoft.com/en-us/azure/postgresql/single-server/concepts-business-continuity#cross-region-read-replicas>

### Use geo-restore for restoring geo-redundant backups.

<https://learn.microsoft.com/en-us/azure/postgresql/single-server/concepts-business-continuity#geo-restore>

# OpenAI

# Azure Bot Service

## Best Practices

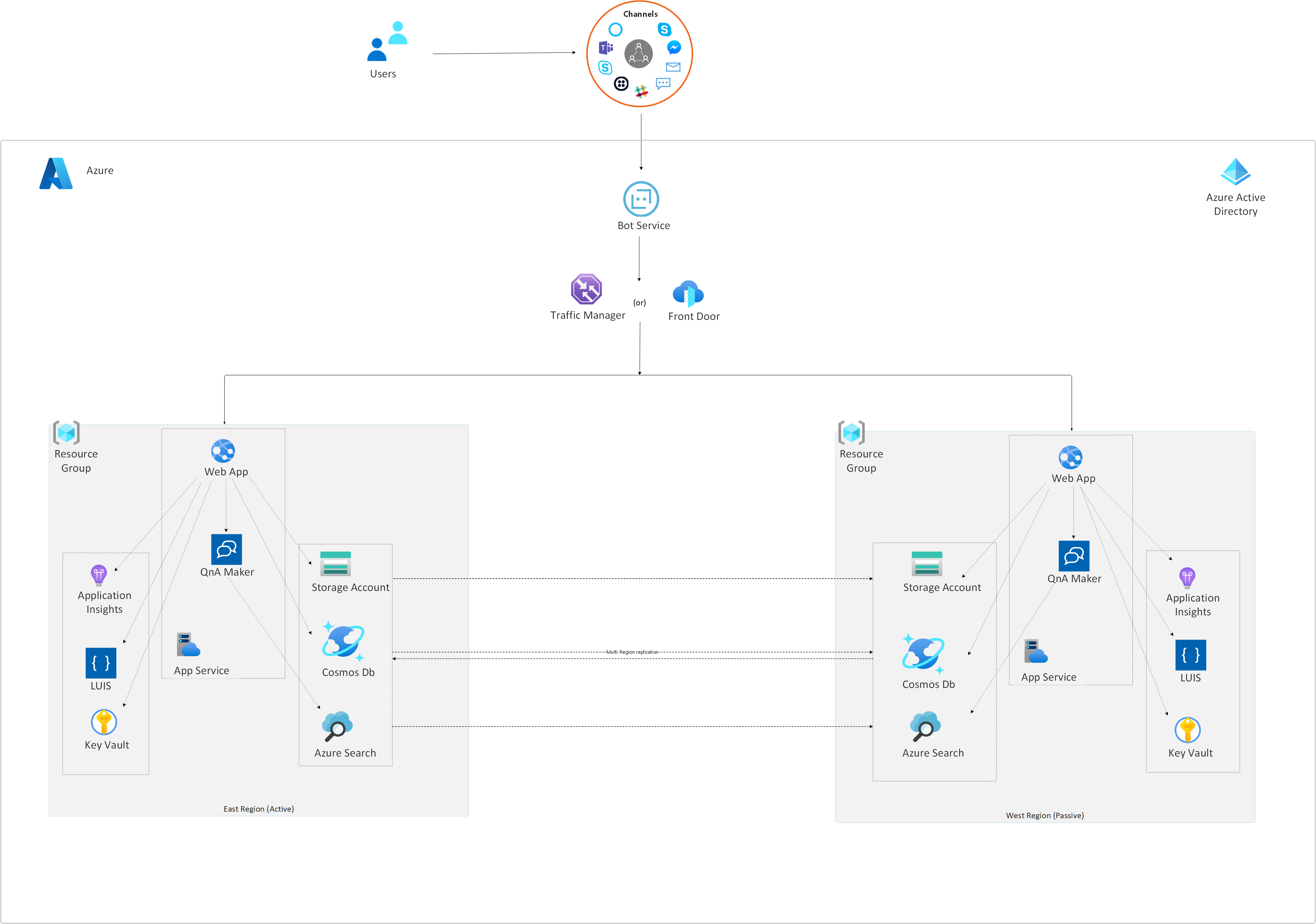
### Reliability in Azure Bot Service

### Regionalization in Azure Bot Service

### Azure Bot Service runs in active-active mode for both global and regional services. When an outage occurs, you don't need to detect errors or manage the service. Azure Bot Service automatically performs autofailover and auto recovery in a multi-region geographical architecture. For the EU bot regional service, Azure Bot Service provides two full regions inside Europe with active/active replication to ensure redundancy. For the global bot service, all available regions/geographies can be served as the global footprint.

### Architectures

#### Disaster recovery for enterprise bots



# Microsoft Purview

## Data Governance

### FTA Resiliency Playbook

### Best Practices

#### High Availability

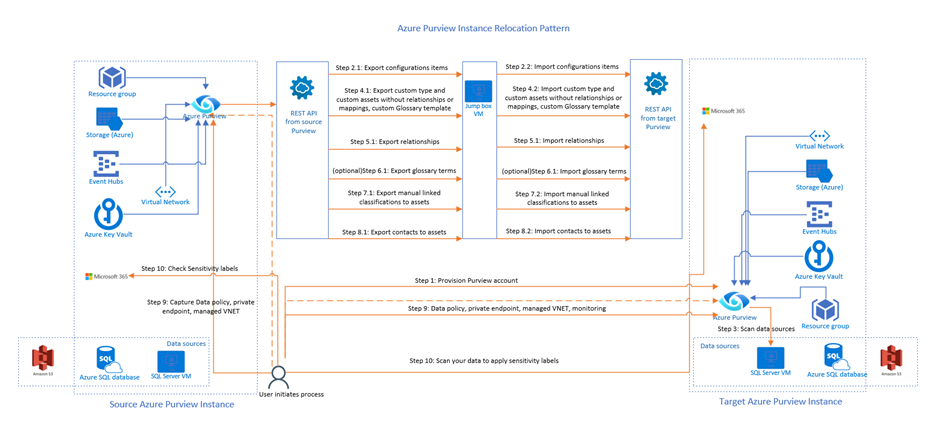
##### Microsoft Purview instances guarantees a monthly uptime percentage of at least 99.9%

#### Disaster recovery

##### Microsoft Purview ensures no data loss but a recovery from outages may require you to restart your workflows such as scans.

##### High-level failover Purview instances: 1. Create the new account 2. Migrate configuration items 3. Run scans 4. Migrate custom typedefs and custom assets 5. Migrate relationships 6. Migrate glossary terms 7. Assign classifications to assets 8. Assign contacts to assets

##### Architecture of disaster recovery



#### Backup and restore

##### You may need to create regular backups from a Microsoft Purview account and use a backup in case a piece of data or configuration is accidentally or deliberately deleted from the Microsoft Purview account by the users.

#### Purview accounts replications

##### Use Microsoft Purview's Event Hubs to subscribe and create entities to another account

### Data Catalog

#### Purview accounts architectures and best practices

#### Collection design

#### Asset lifecycle best practices

#### Automation best practices

#### Backup and migration best practices

#### Business Glossary

#### Workflow

#### Security best practices

### Data Map

#### Lineage best practices

#### Scan registered source

#### Classification best practices

#### Sensitivity label

### Data Sharing

#### In-place ADLS Gen2 & Storage data sharing

### Data Estate

#### Turn on/off Data Estate Insights

#### Data stewardship, Catalog adoption

#### Inventory and ownership

#### Insights for Glossary, classifications, Sensitivity labels

### Data Quality

#### Generate assessment scores

#### Profiling- get summaries of data content

### Data Policy

#### Data Owner access policies

#### Self-service access policies

#### DevOps policies

## Compliance

# Fluid Relay

## Best Practices

### Cross Region Replication

#### To deliver a highly available service, the container data is replicated to another region.

#### Internally, Azure Fluid Relay uses Azure Blob Storage cross-region replication to achieve that. The region where data is replicated is defined by the Azure regional pairs listed on the Cross-region replication in Azure page.

# AzureML

## Best Practices

### Your Code

#### CI/CD

### Your ML Workspace

#### ARM/Bicep/Terraform/PowerShell

### Your Registry

# Cognitive Services

## FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/paas-foundations-playbooks-cog_svcs_v1.docx>

## OpenAI

### BackUp your Prompts

### Business Continuity and Disaster Recovery (BCDR) considerations with Azure OpenAI Service

### Backup your ChatGPT conversations

## Language

### CI/CD for Custom Speech

### QnA Service

#### Move a knowledge base using export-import

## Vision

## Decision

# Azure Cognitive Search

## Best Practices

### High availability

#### Enable 2 replicas to have 99.9% availability for read operations

#### Enable 3 replicas to have 99.9% availability for read/write operations

### Leverage Availability Zones by enabling read and/or write replicas

### Geo-replication

<https://learn.microsoft.com/en-us/azure/search/search-reliability#multiple-services-in-separate-geographic-regions>

#### doesn't provide an automated method of replicating search indexes across geographic regions

#### manually create services and 2 or more regions

#### synchronize data across multiple services

<https://learn.microsoft.com/en-us/azure/search/search-reliability#synchronize-data-across-multiple-services>

##### Option 1: use indexers for updating content on multiple services

##### Option 2: Use REST APIs for pushing content updates on multiple services

<https://learn.microsoft.com/en-us/azure/search/tutorial-optimize-indexing-push-api>

#### Use Azure Traffic Manager to coordinate requests

<https://learn.microsoft.com/en-us/azure/search/search-reliability#use-azure-traffic-manager-to-coordinate-requests>

### Disaster recovery and service outages

<https://learn.microsoft.com/en-us/azure/search/search-reliability#disaster-recovery-and-service-outages>

#### create services in two or more regions

### back up and restore alternatives

<https://learn.microsoft.com/en-us/azure/search/search-reliability#back-up-and-restore-alternatives>

#### use this sample code to back up index definition and snapshot to a series of Json files.

<https://github.com/Azure-Samples/azure-search-dotnet-samples>

### Back up and restore an Azure Cognitive Search index

# Stream Analytics

## FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/paas-foundations-playbooks-stream_analytics_v1.docx>

## Best Practices

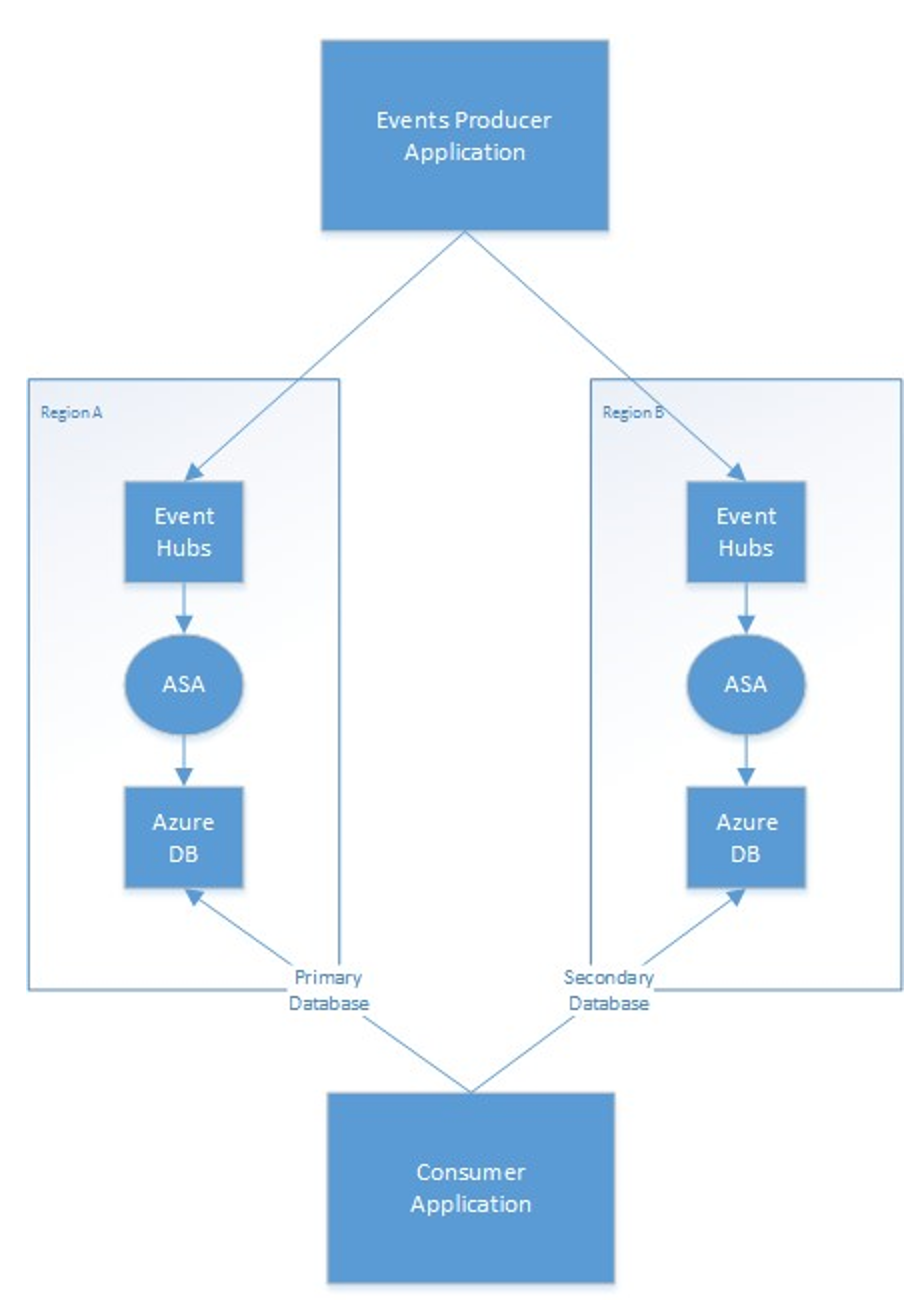
### High Availability 99% SLA

#### Azure Stream Analytics provides high availability (99.9% SLA) for jobs and clusters within a region, the details of which are transparent to the end customer. If failures occur within the service, per the documentation “Azure Stream Analytics guarantees exactly once event processing and at-least-once delivery of events, so events are never lost.”

### Achieve Geo-Redundancy

#### Azure Stream Analytics resources (jobs, clusters, etc.) are regional and do not provide automatic geo-failover. However, you can achieve geo-redundancy by deploying identical Stream Analytics jobs in multiple Azure regions. Each job connects to local input and output sources. It is the responsibility of your application to both send input data into the two regional inputs and reconcile between the two regional outputs.

### Active/Active Approach



### Acitve/Passive Approach

#### As an alternative to this duplicate-processing approach, the ASA job in the secondary region could be passive (stopped), and the output database geo-replicated to the secondary region. On region failure, the database would failover, and the secondary ASA job would be started using the CustomTime start mode to resume processing from before the outage began. While this approach lowers cost, it has the downside of increased recovery time due to database failover.

# PowerBI

## Best Practices

### Power BI is fully managed software as a service (SaaS)

### Power BI maintains multiple instances of each component in Azure datacenters (also known as regions) to guarantee business continuity. If there's an outage, or Power BI becomes inaccessible or inoperable in a region, Power BI fails all its components in that region to a backup instance. The failover restores availability and operability to the Power BI service instance in a new region usually within the same geographic location.

### Source control your PBIT/PBIX files

### Resiliency

# ADX

## Best Practices

<https://learn.microsoft.com/en-us/azure/data-explorer/business-continuity-overview>

### Leverage External Tables and Continuous data export overview to reduce costs

<https://learn.microsoft.com/en-us/azure/data-explorer/kusto/management/data-export/continuous-data-export>

#### Using the correct approach to feed a datalake with cold data and having the Kusto query engine at your disposal at the same time, as in the short-term storage

<https://riccardo-zamana.medium.com/adx-data-explorer-how-to-develop-a-continuous-data-lake-feed-maintaining-a-query-engine-loopback-96bba7c29a4a>

### Leader-follower cluster configuration

<https://learn.microsoft.com/en-us/azure/data-explorer/follower?tabs=csharp>

#### Azure Data Explorer provides an optional follower capability for a leader cluster to be followed by other follower clusters for read-only access to the leader's data and metadata. Changes in the leader, such as create, append, and drop are automatically synchronized to the follower. While the leaders could span Azure regions, the follower clusters should be hosted in the same region(s) as the leader. If the leader cluster is down or databases or tables are accidentally dropped, the follower clusters will lose access until access is recovered in the leader.

### Create Multiple independent clusters

<https://learn.microsoft.com/en-us/azure/data-explorer/business-continuity-create-solution>

#### Azure Data Explorer doesn't support automatic protection against the outage of an entire Azure region. This disruption can happen during a natural disaster, like an earthquake. If you require a solution for a disaster recovery situation, do the following steps to ensure business continuity. In these steps, you'll replicate your clusters, management, and data ingestion in two Azure paired regions.

##### Create two or more independent clusters in two Azure paired regions.

<https://learn.microsoft.com/en-us/azure/data-explorer/business-continuity-create-solution#create-multiple-independent-clusters>

##### Replicate all management activities such as creating new tables or managing user roles on each cluster.

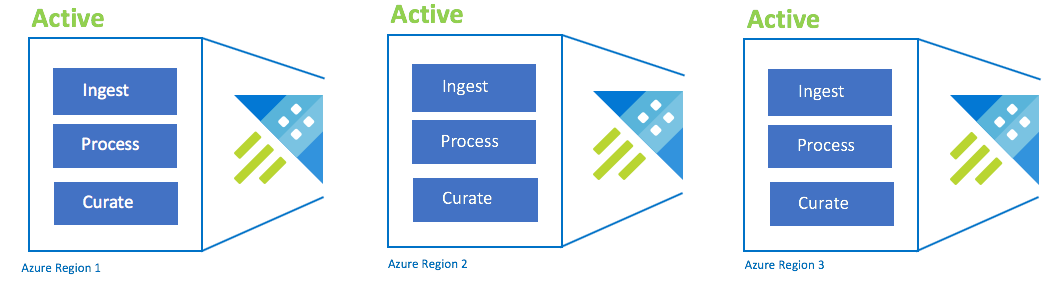
<https://learn.microsoft.com/en-us/azure/data-explorer/business-continuity-create-solution#replicate-management-activities>

##### Ingest data into each cluster in parallel.

### DR Configurations

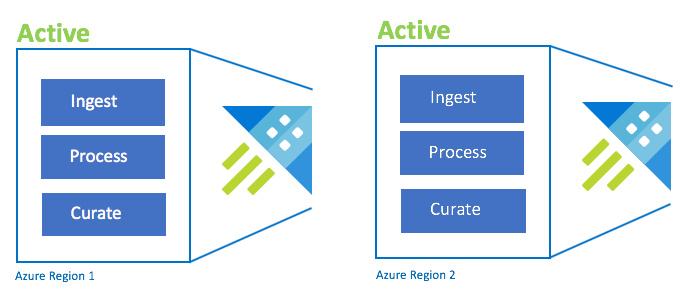
<https://learn.microsoft.com/en-us/azure/data-explorer/business-continuity-overview>

#### Active-Active-Active (always-on) configuration



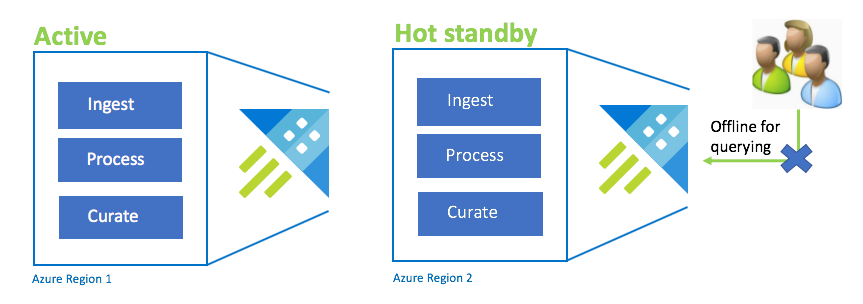
##### This configuration is also called "always-on". For critical application deployments with no tolerance for outages, you should use multiple Azure Data Explorer clusters across Azure paired regions.

#### Active-Active configuration



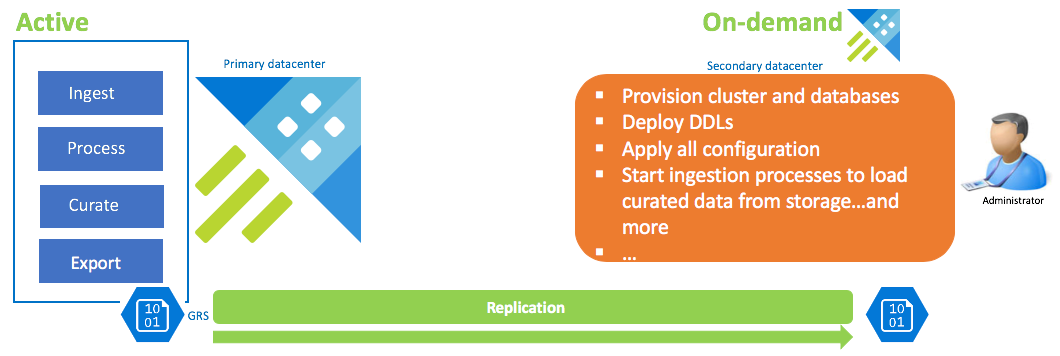
##### This configuration is identical to the active-active-active configuration, but only involves two Azure paired regions. Configure dual ingestion, processing, and curation. Users are routed to the nearest region. The cluster SKU must be the same across regions.

#### Active-Hot standby configuration



##### The Active-Hot configuration is similar to the Active-Active configuration in dual ingest, processing, and curation. While the standby cluster is online for ingestion, process, and curation, it isn't available to query. The standby cluster doesn't need to be in the same SKU as the primary cluster. It can be of a smaller SKU and scale, which may result in it being less performant. In a disaster scenario, users are redirected to the standby cluster, which can optionally be scaled up to increase performance.

#### On-demand data recovery cluster configuration



##### This solution offers the least resiliency (highest RPO and RTO), is the lowest in cost and highest in effort. In this configuration, there's no data recovery cluster. Configure continuous export of curated data (unless raw and intermediate data is also required) to a storage account that is configured GRS (Geo Redundant Storage). A data recovery cluster is spun up if there is a disaster recovery scenario. At that time, DDLs, configuration, policies, and processes are applied. Data is ingested from storage with the ingestion property kustoCreationTime to over-ride the ingestion time that defaults to system time.

### DevOps

#### Wrap DevOps and source control around all your code

##### All database objects, policies, and configurations should be persisted in source control so they can be released to the cluster from your release automation tool.

<https://learn.microsoft.com/en-us/azure/data-explorer/devops>

### Design, develop, and implement validation routines to ensure all clusters are in-sync from a data perspective.

### Be fully cognizant of what it takes to build a cluster from scratch.

# Data Factory

## FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/paas-foundations-playbooks-ADF_v1.docx>

## Best Practices

<https://learn.microsoft.com/en-us/azure/architecture/example-scenario/analytics/pipelines-disaster-recovery>

### Use zone redundant pipelines in regions that support Availability Zones

<https://learn.microsoft.com/en-us/azure/architecture/example-scenario/analytics/pipelines-disaster-recovery>

### GitHub/Azure DevOps Integration

<https://learn.microsoft.com/en-us/azure/data-factory/source-control>

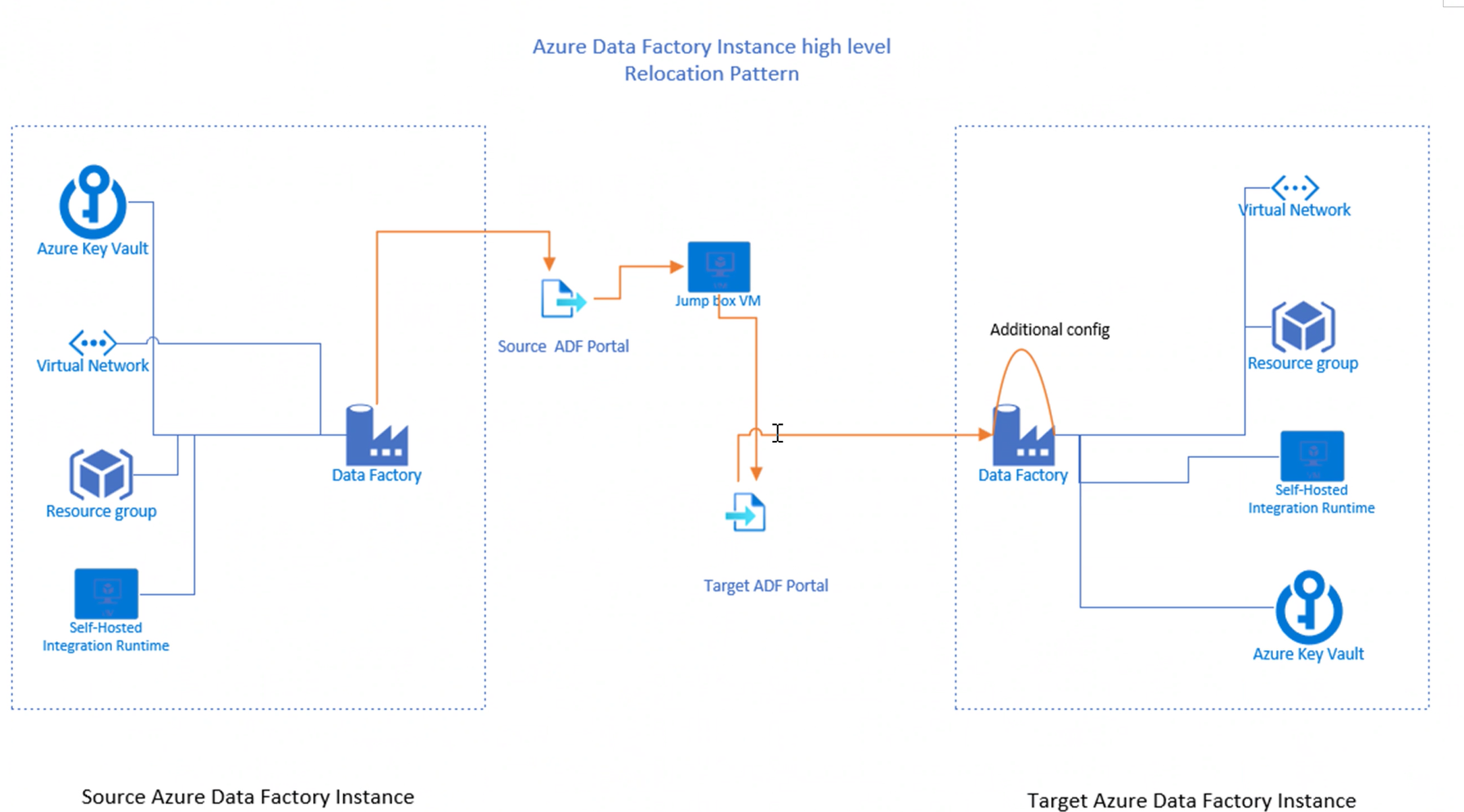
#### Use DevOps to Backup the ARM Templates

### Self Hosted Integration Runtimes

#### Make sure you replicate the SHIR VMs in another region.

### Network

#### Make sure you replicate or duplciate your network in the sister region. You have to make a copy of your VNet in another region



### Key Vault

<https://learn.microsoft.com/en-us/azure/key-vault/general/disaster-recovery-guidance>

#### If your ADF Pipelines use Key Vault you don't have to do anything to replicate Key Vault. Key Vault is a managed service and Microsoft takes care of it for you.

#### In the rare event that an entire Azure region is unavailable, the requests that you make of Azure Key Vault in that region are automatically routed (failed over) to a secondary region (except as noted). When the primary region is available again, requests are routed back (failed back) to the primary region. Again, you don't need to take any action because this happens automatically.

# Cosmos DB

## FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/pass-foundations-playbooks-CosmosDB_v1.docx>

## Best Practices

### Leverage Availability Zones where regionally applicable, and of course if the service offers it.

<https://learn.microsoft.com/en-us/azure/cosmos-db/high-availability#replica-outages>

### Run multiple replicas of the database (> than 1) in prod

<https://learn.microsoft.com/en-us/azure/cosmos-db/high-availability#replica-outages>

### Leveerage Multi-Region Writes

#### Multi-region writes capability allows you to take advantage of the provisioned throughput for your databases and containers across the globe.

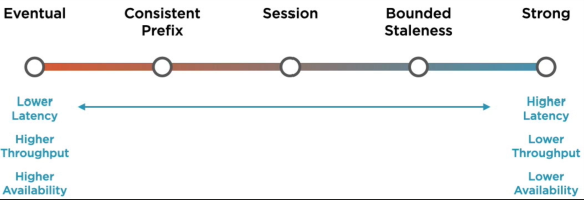
### Distribute your data globally

#### Span Cosmos account across two or more regions with multi-region writes.

<https://learn.microsoft.com/en-us/azure/cosmos-db/high-availability#slas>

#### Choose from several well-defined consistency models

##### Consistency Levels



<https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels>

###### Eventual

Lower Latency

Higher Throughput

Higher Availability

###### Consistent Prefix

###### Session

###### Bounded Staleness

###### Strong

Higher Latency

Lower Throughput

Lower Availability

###### TIP: Be Careful with Strong and Bounded Staleness

Distributed databases relying on replication for high availability, low latency or both, you make the fundamental tradeoff between the read consistency vs. availability, latency, and throughput

When using either the strong or bounded staleness consistency levels, the RU cost of any read operation (point read or query) is doubled.

#### Enable service-managed failover

##### Maintain business continuity during regional outages.

##### Azure Cosmos DB supports service-managed failover during a regional outage. During a regional outage, Azure Cosmos DB continues to maintain its latency, availability, consistency, and throughput SLAs. To help make sure that your entire application is highly available, Azure Cosmos DB offers a manual failover API to simulate a regional outage. By using this API, you can carry out regular business continuity drills.

### Automatics Backaups

#### Azure Cosmos DB automatically takes backups of your data at regular intervals. The automatic backups are taken without affecting the performance or availability of the database operations. All the backups are stored separately in a storage service.

#### Periodic Backups

##### This mode is the default backup mode for all existing accounts. In this mode, backup is taken at a periodic interval and the data is restored by creating a request with the support team. In this mode, you configure a backup interval and retention for your account. The maximum retention period extends to a month. The minimum backup interval can be one hour.

#### Continuous backup with point-in-time restore in Azure Cosmos DB

##### Continous (7 day retention) Backups

##### Continous (30 day retention) Backups

##### Azure Cosmos DB performs data backup in the background without consuming any extra provisioned throughput (RUs) or affecting the performance and availability of your database. Continuous backups are taken in every region where the account exists.

# Databricks

## FTA Resiliency Playbook

<https://github.com/Azure/fta-resiliencyplaybooks/blob/main/pass-foudations-playbooks-ADB_v1.docx>

## Best Practices

### For Databricks you have to worry about backing up and recovering 4 items:

#### Your Databricks Workspace Configuration

##### ARM Templates and DevOps

##### Secret Scopes also need to be backedup

###### Databricks CLI

#### Your Hive Metastore

##### Sharing Metadata Across Different Databricks Workspaces Using Hive External Metastore

<https://techcommunity.microsoft.com/t5/fasttrack-for-azure/sharing-metadata-across-different-databricks-workspaces-using/ba-p/3679757>

##### Disaster Recovery Strategy in Databricks using the Hive External Metastore

<https://techcommunity.microsoft.com/t5/fasttrack-for-azure/disaster-recovery-strategy-in-azure-databricks-using-the-hive/ba-p/3684581>

#### Your Data

##### Deep & Shallow Clones

<https://www.databricks.com/blog/2021/04/20/attack-of-the-delta-clones-against-disaster-recovery-availability-complexity.html>

##### Azure Storage RA-GRS

<https://learn.microsoft.com/en-us/azure/storage/common/storage-redundancy>

###### Download the blob using Secondary Endpoint in RAGRS Storage account

<https://techcommunity.microsoft.com/t5/azure-paas-blog/download-the-blob-using-secondary-endpoint-in-ragrs-storage/ba-p/2403750>

#### Your Code

##### DevOps

<https://learn.microsoft.com/en-us/azure/databricks/dev-tools/index-ci-cd>

### Disaster Recovery Options

<https://learn.microsoft.com/en-us/azure/databricks/administration-guide/disaster-recovery>

#### Active-passive solution strategy

#### Active-active solution strategy

### Simplifying DR with Delta Lake

<https://www.databricks.com/session_na20/simplifying-disaster-recovery-with-delta-lake>

### Disaster Recovery Overview, Strategies, and Assessment

<https://www.databricks.com/blog/2022/04/25/disaster-recovery-overview-strategies-and-assessment.html>

### Tooling

#### Databricks Workspace Migration Tools

<https://github.com/databrickslabs/migrate>

##### This is a migration package to log all Databricks resources for backup and/or migrating to another Databricks workspace. Migration allows a Databricks organization to move resources between Databricks Workspaces, to move between different cloud providers, or to move to different regions / accounts.

#### Databricks Sync (DBSync)

<https://github.com/databrickslabs/databricks-sync>

# Synapse Analytics

## Best Practices

### Enable geo-backup. By default, Synapse Analytics takes a full backup of your data every 24 hours for disaster recovery.

<https://learn.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/backup-and-restore#geo-backups-and-disaster-recovery>

### Azure DevOps/Git Integration

#### Use DevOps to backup and recover your Synapse Resource Configruation

<https://techcommunity.microsoft.com/t5/azure-synapse-analytics-blog/how-to-use-ci-cd-integration-to-automate-the-deploy-of-a-synapse/ba-p/2248060>

##### How to use CI/CD integration to automate the deploy of a Synapse Workspace to multiple environments

### Synapse Pipelines (See ADF Section Above)

<https://learn.microsoft.com/en-us/azure/architecture/example-scenario/analytics/pipelines-disaster-recovery>

#### Use zone redundant pipelines in regions that support Availability Zones

<https://learn.microsoft.com/en-us/azure/architecture/example-scenario/analytics/pipelines-disaster-recovery>

### Create Scripts for all DDL Statements and save in GitHub or Azure DevOps Repos

### If you work with Spark Notebooks make sure to integrate with Git or Azure DevOps

### Dedicated Pools

<https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/migrate/azure-best-practices/analytics/azure-synapse>

#### Database-restore points for Azure Synapse Analytics

##### Azure Synapse Analytics uses database snapshots to provide database-restore points for the warehouse. A data warehouse snapshot creates a restore point that can be used to recover or copy a data warehouse to a previous state.

### Serverless Pools

#### You don't have to worry about Serverless Pools as they are maintained by Microsoft.

### ARM Templates

#### Leverage Terraform or Bicep or PowerShell or Azure CLI or whatever is your goto ARM Template manager.

### Make sure to re-establish any Synapse Link(s)

#### Azure Cosmos DB

<https://learn.microsoft.com/en-us/azure/cosmos-db/synapse-link>

# SQL Database

<https://learn.microsoft.com/en-us/azure/azure-sql/database/business-continuity-high-availability-disaster-recover-hadr-overview?view=azuresql>

## Best Practices

### Leverage Availability Zones where regionally applicable, and of course if the service offers it.

<https://learn.microsoft.com/en-us/azure/azure-sql/database/high-availability-sla?view=azuresql&tabs=azure-powershell#premium-and-business-critical-service-tier-zone-redundant-availability>

### Run Business Critical or Premium tiers for applications that require highest SLA

<https://learn.microsoft.com/en-us/azure/azure-sql/database/high-availability-sla?view=azuresql&tabs=azure-powershell#premium-and-business-critical-service-tier-zone-redundant-availability>

## Azure SQL DB

### Backups

#### Automatics PITR (Point in Time) Backups

##### What type of Storage Reduandancy do you want?

###### LRS

###### GRS

#### Long Term Retention

##### Leverage GRS storage type for the database “Backup Storage Redundancy.”

<https://learn.microsoft.com/en-us/azure/azure-sql/database/automated-backups-overview?view=azuresql#backup-storage-redundancy>

#### Geo-Backups

### BCDR

#### Failover Groups

#### Geo-Replication

##### Use Active Geo-Replication to create a readable secondary in a separate region

<https://learn.microsoft.com/en-us/azure/azure-sql/database/active-geo-replication-overview?source=recommendations&view=azuresql>

### Maintenance Windows

<https://learn.microsoft.com/en-us/azure/azure-sql/database/maintenance-window?view=azuresql>

## SQL MI

### Backups

#### Automatics PITR (Point in Time) Backups

##### What type of Storage Reduandancy do you want?

###### LRS

###### GRS

#### Long Term Retention

##### Leverage GRS storage type for the database “Backup Storage Redundancy.”

<https://learn.microsoft.com/en-us/azure/azure-sql/database/automated-backups-overview?view=azuresql#backup-storage-redundancy>

#### Geo-Backups

##### Use Active Geo-Replication to create a readable secondary in a separate region

<https://learn.microsoft.com/en-us/azure/azure-sql/database/active-geo-replication-overview?source=recommendations&view=azuresql>

### Maintenance Windows

<https://learn.microsoft.com/en-us/azure/azure-sql/database/maintenance-window?view=azuresql>

## SQL VM

### Azue BackUp Service

### Availability Groups

### SQL IAAS Extension

<https://learn.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/sql-server-iaas-agent-extension-automate-management?view=azuresql&tabs=azure-powershell>

### Take a Backup of the Database and save it

### Leverage Premium Disks

# Azure Storage

## Best Practices

### Leverage a storagev2 account type for better performance and reliability

<https://learn.microsoft.com/en-us/azure/storage/common/storage-account-upgrade?tabs=azure-portal>

### Data Lake Storage Gen2

#### Leverage GRS, ZRS or GZRS storage for the highest availability

<https://learn.microsoft.com/en-us/azure/storage/common/storage-redundancy>

##### Redundancy Options

###### Geo-Redundant Storage (GRS)

###### Geo-Zone-Reduandant Storage (GZRS)

###### Read-Access Geo-Redundant Storage (RA-GRS)

###### Read-Access Geo-Zone-Redundant Storage (RA-GXRS)

#### Customer-Managed Failover

<https://learn.microsoft.com/en-us/azure/storage/common/storage-disaster-recovery-guidance>

#### Microsoft-Managed Failover

<https://learn.microsoft.com/en-us/azure/storage/common/storage-disaster-recovery-guidance#microsoft-managed-failover>

##### When storage account is failed over utilizing Microsoft-managed failover, Microsoft handles the entire Failover and Failback process without intervention from the customer.

##### After failback, the storage account is configured to be locally redundant, and the customer has a responsibility to re-enable geo-redundancy for the storage account.

### Data Protection Options

#### Soft Deletes