International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V72I8P120

Volume 72 Issue 8, 134-141, August 2024

Original Article

Preparing for the Unexpected Outages for Your Mission-Critical Cloud Infrastructure with Confidence

Prasad Gandham[1], Ramakanth Damodaram[2], Sunit Randhawa[3]

[1]*Microsoft, Salt Lake City, Utah, USA*
[2]*Amazon Web Services, Dallas, Texas, USA.*
[3]*Amazon Web Services, California, USA.*

[1]*Corresponding Author: pgandham@microsoft.com*

Received: 24 June 2024 Revised: 27 July 2024 Accepted: 12 August 2024 Published: 31 August 2024

Abstract *- Cloud computing is like a virtual computer city, providing the infrastructure to host your mission-critical applications without the burden of managing operational overhead. The reliability of any workload in a cloud environment depends on its architectural design. Here comes the foundational question: "Are you designing your workloads to be resilient?". Resilience stands as a cornerstone for the stability of the workload along with safeguarding access to your sensitive data. The resilience of the infrastructure components is pivotal, especially when entrusted with the vital task of ensuring uninterrupted service. Based on Uptime Institute - the Global Digital Infrastructure Authority 's research in 2022, one in five organizations across the globe have faced 'severe' or 'serious' outages, and 60% of these outages have cost around $100,000 US dollars to the respective organizations. This paper explains the importance of designing and implementing a fault-tolerant design by exploring scenarios ranging from network connectivity to regional outages within data centers and cross-region environments. This research study will empower and prepare the readers for incident readiness by offering methods to reduce blast radius, protect the environment, and keep informed throughout incident lifecycles. It provides a design process flow for applications, databases, and traffic management, which form the nexus of user interaction. Additionally, it shares remediation strategies for resilience failover and guides on defining service level agreements for mission-critical applications. Testing the system's resilience against failures is paramount, whether it is through actual disruptions or simulated scenarios. Exploring the architecture's robustness on how to plan for failures with best practices by using architectural patterns and orchestration techniques plays a pivotal role in fortifying the system against adversities. This research provides a conceptual design guiding practitioners towards safeguarding the integrity and continuity of the mission-critical cloud infrastructure ecosystems from unplanned outages and hardware failures.*

Keywords *- Cloud computing, Resilience, Azure, Amazon web services, Infrastructure, Monitoring, Zones, Regions, Failover, Recovery.*

1. Introduction

In the digital era, ensuring the resilience of IT infrastructure is critical for business continuity and considering the design challenges.[1], [2] for building robust and scalable solutions on the cloud. While significant progress has been made in developing robust and scalable cloud solutions, a gap remains in addressing comprehensive design challenges for truly resilient systems. Existing literature has explored strategies for achieving high availability, fault tolerance, and rapid incident response on cloud platforms. However, these studies often overlook the intricacies of implementing resilience across different zones and regions, particularly when subjected to routine stressors and unexpected failures. This article explores approaches to designing resilient systems using zones, regions, chaos engineering and monitoring the same to keep up to date. Some of the key strategies they employ to ensure high availability, fault tolerance, and rapid incident response. Resilience is about having "the capability to recover when stressed by load (more requests for service), attacks (either accidental through a bug, or deliberate through malicious intention), and failure of any component in the workload's components." where stressors can be classified as routine or one-offs. Routine stressors are the types of failures that are expected to happen quite often. Disks fail, network connections drop and reconnect, and software has bugs; these routine events should not interrupt service, and workloads should recover automatically. Also, Resilience is to sustain failures from zonal, regional outages where organizations can recover with minimal effort. This paper discusses 4 such scenarios which will make the readers think and implement utilizing a cloud adoption framework, well architected

framework. By exploring these four distinct scenarios, this study provides a detailed examination of how organizations can effectively leverage chaos engineering and monitoring strategies to maintain service continuity and recover from failures. The insights derived from this research will empower practitioners to utilize cloud adoption and well-architected frameworks more effectively, ensuring rapid recovery and operational integrity during challenging situations.

2. Related work

2.1. Resiliency Research Based on Current Papers

Based on the research, some of the papers already discuss infrastructure resiliency [3], and this focuses on understanding the reasons for failures in cloud services due to large-scale disruptions and aims to create reliable data services that can tolerate such changes efficiently and robustly as this is still a reactive approach and the design to be resilient which discussed in [4]proposed architecture in creating a resilient cloud computing infrastructure that can withstand and recover from node corruptions. The paper [5] discusses a comprehensive overview of cloud computing concepts, architecture, security, and privacy issues. It emphasizes its importance in modern computing environments but does not provide a scalable way for customers to be proactive in building resilient systems. Regarding [6] does not cover real-world case studies or practical examples to illustrate how organizations have successfully overcome design challenges. Existing frameworks were looked at in this research [7], which discuss Cloud Computing business models but do not delve into specific examples or case studies to illustrate this point further. Researched practical examples for resiliency [8], [9], [10], but these lack the depth of the discussion about zones, regions, and failover testing. Then there is research on monitoring and staying up-to-date on the systems [11], [12], but these lack information on how to stay up-to-date with cloud outages and building the logging. The paper primarily focuses on the concept of resilience in software design, specifically in cloud-based data-handling solutions, but it may lack in-depth discussion on the practical implementation of resiliency strategies in complex software systems.

2.2. The Importance of Resilience

With the exponential growth of cloud computing and the increasing reliance on technology, businesses must be prepared to face unexpected challenges that can disrupt operations. Resilience is not just about having backup and disaster recovery solutions; it involves continuous availability, fault tolerance, and the ability to quickly respond to incidents. Every organization has multiple application tiers that have specific Service Level Agreement (SLA) requirements that need to be met. These SLAs are normally named by a tiering system such as mission-critical, Business-Critical, Production, and Test/Dev. Each SLA will have different availability, recoverability, performance, manageability, and security requirements that need to be. As you read through the doc, you will see more details on the 3 9's and considerations like Disks failing, network connections dropping and reconnecting, and software having bugs; these routine events should not interrupt service, and workloads should recover automatically. These fall under the High Availability (HA) category, and workloads can achieve HA in a single region by making use of multiple zones and sound architecture practices. One-offs are the black swan events when cloud-native services are not available in a region for hours. They fall into the Disaster Recovery (DR) category and require a DR plan to use a different region.

The types of solutions being recommended to provide HA and DR sometimes overlap with the solutions for reducing latency for end users. In most cases, this can be solved for latency reduction more simply by using edge services.

3. Materials and Methods - Key Components of Resilient Architecture

This study focuses on the three methods below:

3.1. Resilient Architecture using Multi Zone and Multi Region - Sample Architecture - What type of Active/Active or Hot Standby

Typically, a company's resiliency goals are tied to its business SLAs promised to customers and partners, which range from 99.9% (3 9's of SLA) to 99.99% (4 9's of SLA). Understanding resiliency patterns and trade-offs to architect efficiently in the cloud is important to keep the cost low when a decision is to be made on whether to do a multi-region or single region deployment.

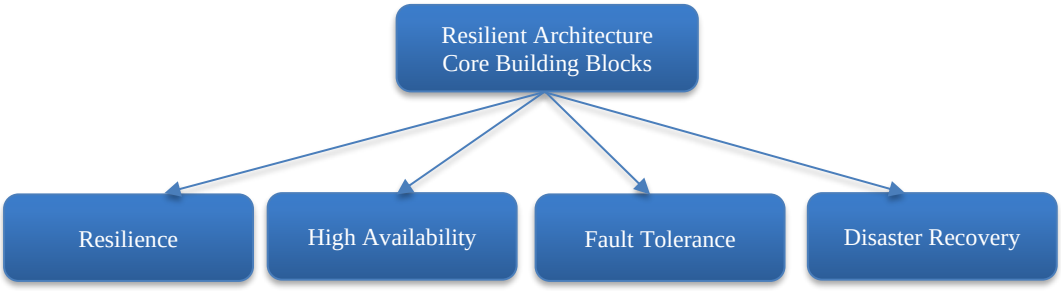

Fig. 1 Components involved in resilient architecture

Typically, multi-region deployment will be more expensive and will give higher SLA with an extra cost coming from design complexity, cost to implement and operational effort. Let us go into detail on these three main core factors before you decide to go multi-region.

The complexity of Design: If a workload is to be run in a multi-region, the system complexity will increase, and emergent behaviors often will become more prevalent. Each individual workload component must be resilient, and it is crucial to eliminate single points of failure across people, processes, and technology. Customers should evaluate their resilience requirements to determine whether increasing system complexity is beneficial or if maintaining simplicity and implementing a Disaster Recovery (DR) plan would be more suitable.

Implementation Costs – Costs often rise significantly when enhancing resilience due to the need for new software and infrastructure components running redundantly. It is essential for these costs to be justified by the potential savings to avoid future losses.

Operational Effort – Implementing and maintaining highly resilient systems demands intricate operational processes and advanced technical expertise. For instance, customers may need to enhance their operational processes using the Operational Readiness Review (ORR) method. Before opting for higher resilience, assess your operational capabilities to ensure you possess the necessary process maturity and skill sets.

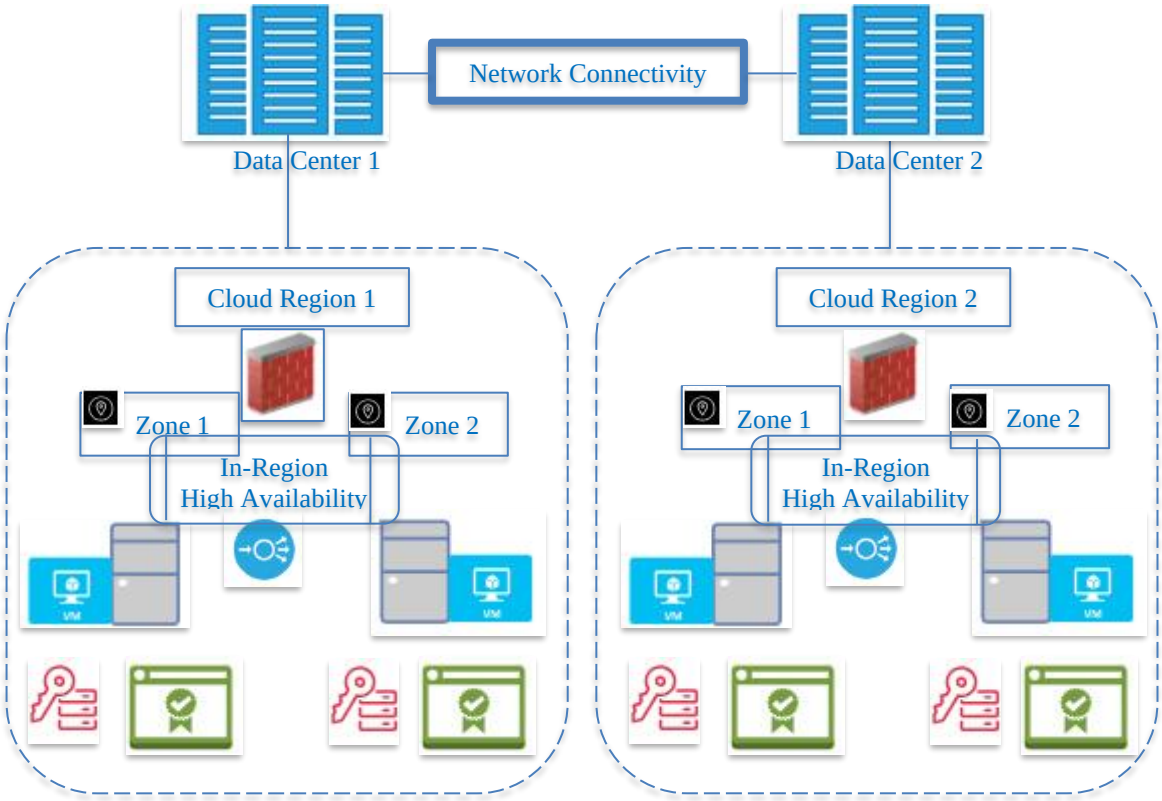

Fig. 2 Sample resilient architecture patterns with multi-region, high availability, resource protection

3.1.1. Multi-Zone Architecture

This pattern is good for low-business-impact applications operating in multiple zones within a single Region. This allows your application to withstand zone-level disruptions. In this pattern, one can leverage auto scaling techniques where if a zone goes down, new compute instances hosting the application (as a Kubernetes container) can be started in different zones. This pattern can have the least design complexity, implementation and operational cost, but it comes at the expense of application recovery. If a zone is down, it will disrupt end users' access to the application while the new resources are being re-provisioned in a new zone.

3.1.2. Multi Zone with Static Stability

Distributed architecture with inherent resilience involves deploying multiple instances across different zones in a region to improve system robustness. This approach leverages consistent performance characteristics to avoid unpredictable behavior under varying conditions. Systems

designed with this principle maintain stable operation regardless of environmental fluctuations. A key benefit of this design is the simplified recovery process during disruptions, as necessary resources are pre-allocated. This means that when issues arise, such as the loss of resources in one zone, the system can continue functioning without relying on control planes for recovery. To implement this strategy, applications must be capable of operating across distributed instances. For compatible applications, deployment across all available zones in a region (typically three or more) is possible. This approach can lead to cost savings by reducing over-provisioning, as capacity can be spread more efficiently across multiple zones compared to a two-zone setup.

3.1.3. Multi-Region DR

Two main strategies exist for multi-region disaster recovery:

1. Minimal Standby: This approach suits applications with Recovery Time and Point Objectives (RTO/RPO) in the range of tens of minutes. It involves active data replication and pre-configured application infrastructure in the backup region. To minimize costs, the application infrastructure remains inactive until a recovery event occurs.
2. Reduced-Capacity Active: This method offers faster recovery times by maintaining a scaled-down version of the application running in the backup region. During a disaster event, the infrastructure can be rapidly scaled up, often through automated processes with minimal manual input. When properly implemented, this approach can achieve RTO/RPO measured in minutes.

Both strategies may result in slight database inconsistencies between regions. To address this, a transaction reconciliation method can be employed.

A recommended approach involves recording each transaction in a distributed event streaming system before committing it to the database. This creates a reliable record of transactions for use during recovery.

In this system, the application logs the transaction record in all regions before database commitment. If a failover occurs, a reconciliation process identifies and applies any transactions present in the log but not yet in the database.

For optimal cost-effectiveness while maintaining high availability (theoretically 99.99% when implemented across two regions), the Minimal Standby approach is often the preferred choice despite its limitations in RTO/RPO.

3.1.4. Multi-Region Active-Active

The multi-region active deployment strategy is best suited for systems demanding immediate(almost real-time) recovery and minimal(almost none) data loss. This approach involves concurrent operation of workloads across several geographical areas, enabling service delivery from multiple locations simultaneously. Such a setup not only safeguards against localized disruptions but also fulfills the most stringent uptime and data preservation requirements. To implement this strategy effectively, it is crucial to incorporate asynchronous data replication mechanisms between the various operational regions.

Below are some of the ways to build resilient architectures because architectural complexity is a factor with multi region Complexity of Design: If a workload is to be run in a multi-region, the system complexity will increase, and emergent behaviors often will become more prevalent.

4. Scenarios to Validate the Resilient Architecture

4.1. Network Connectivity Failure

In case of Networking connectivity failure, it is recommended to plan for multi homing and peering the connectivity between datacenter and cloud to avoid the failure scenario below.

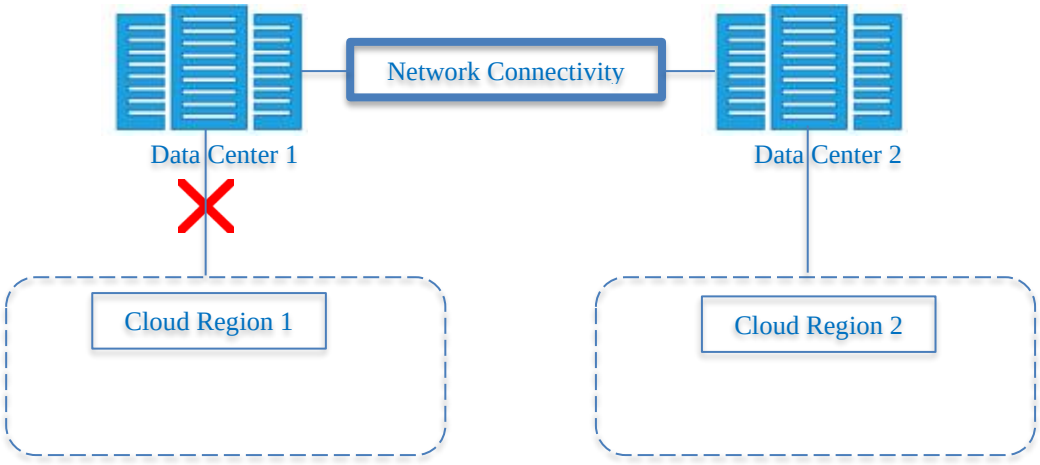

Fig. 3 Sample architecture with network connection failure

To avoid these situations customers could plan for resiliency for the connection. Sharing the documentation on planning for this from Azure [13]so that it can be replicated for other providers.

4.2. No Datacentre or Region Failure, but Data Corruption or Resources not Functional

Each individual workload component must be resilient, and it is crucial to eliminate single points of failure across people, processes, and technology. Customers should evaluate their resilience requirements to the resource level planning or testing each resource failure like below.

In the below architecture, when a virtual machine resource is failed the load balancer can help to route the traffic to another zone. If there is a database failure, the secondary zone will act as the primary. These are some of the resource-level testing/planning for failover that need to be accomplished by the customers as this is a shared responsibility[14].

4.3. Cloud Infrastructure One of the Availability Zones Failures

In case of a region failure cloud providers build a latency-defined perimeter and connect through a dedicated, regional low-latency network. Failure of complete Azure regions is highly unlikely and rare, but to sustain this failure, it is better to automatically distribute Virtual Machines across regions and replicate data across zones, thereby creating an Active/Active DR configuration for protection.

4.4. Cloud Infrastructure Region Failure

It is very unlikely to be a total regional failure, but in this case it is required that one has already planned for RPO and RTO. Consistently performing DR tests and utilizing region replication methods for critical applications.

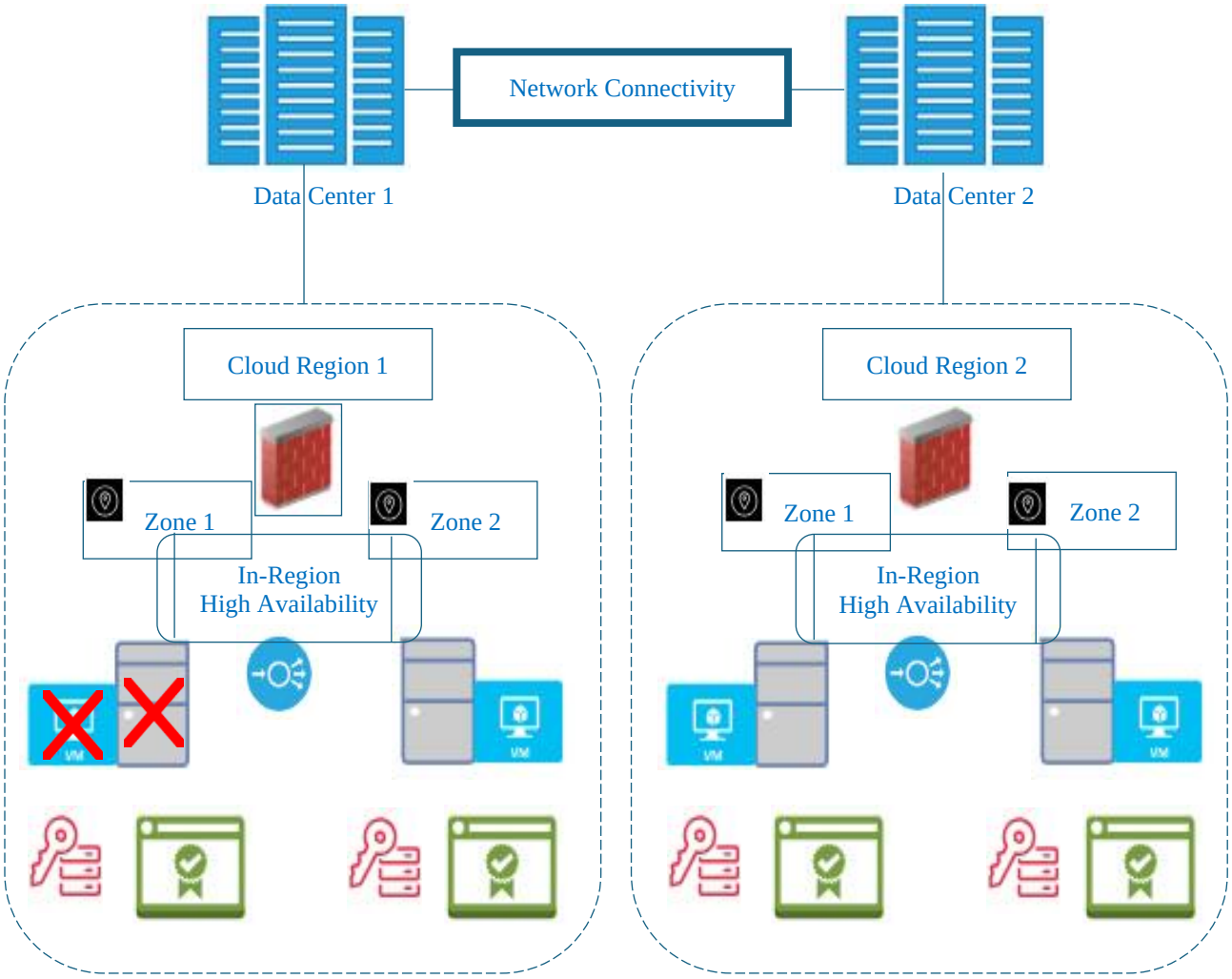

Fig. 4 Sample architecture with resource failures like VM or Database

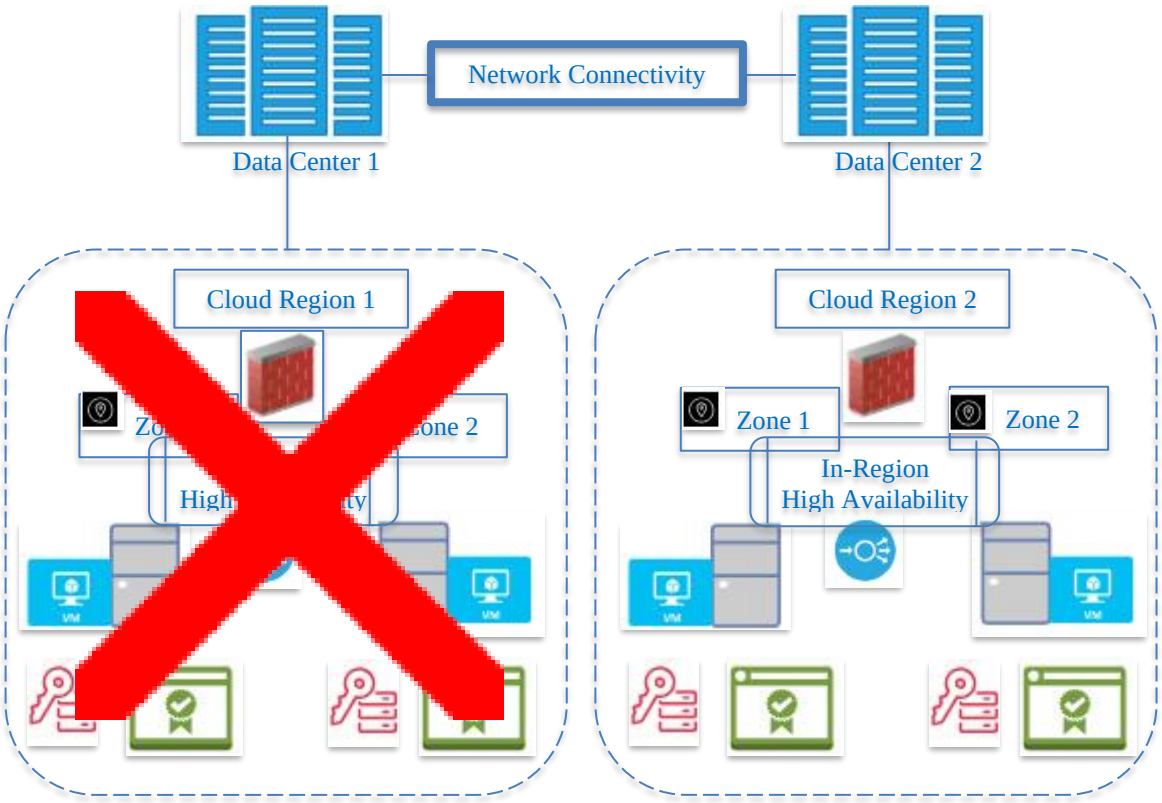

Fig. 5 Sample architecture with region failure

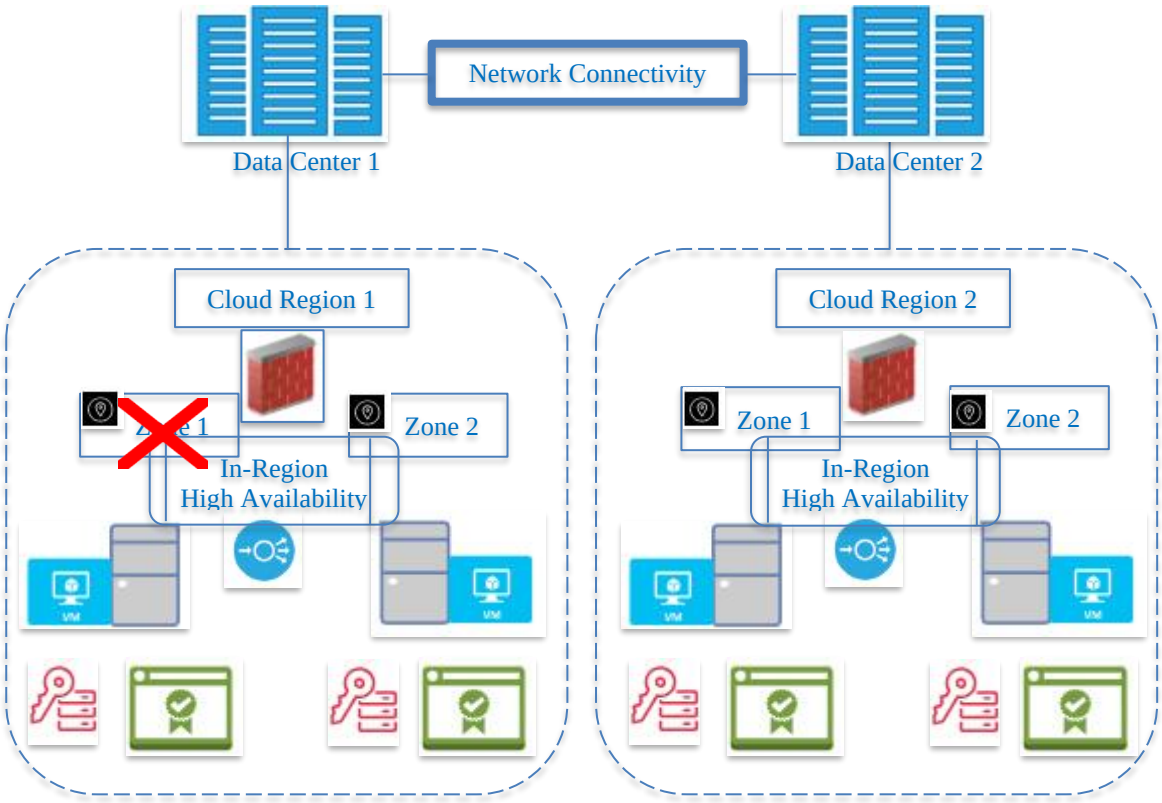

Fig. 6 Sample architecture with zone failure

Table 1. Sample metrics for monitoring

What Resources to be Monitoring	Thresholds/Alerts	How this can be Monitored	High Availability	Region	How Notified
Virtual Machine	Disk Read ad write <= X ms VM Uptime > 999.9%	Cloud native or third-party tools	Cloud tools or third-party tools can be utilized to provide zonal status	VM is replicated or utilizes the latest cloud cross-region VM deployments	Monitoring alerts for metrics, but Zonal/Regional should be part of an ongoing review
Key Vault/Certificates	Expiration	Cloud native or third party	Replicate keys to Zones	Replicate the keys to regions and store the master key	Alerts with enough planning to avoid failures
Network Connection	Connection status	Cloud Native or third party	Zonal setup for Network connectivity	Regional setup for connectivity	

Understanding some of the resiliency patterns from Amazon Web Services is documented here[15]

5. Staying Connected with Life Cycle of Incidents - Monitoring and Alerts and Metrics

Monitoring is essential for aspect critical design areas to be thoroughly planned and considered for any production workloads for business[16] does discuss the importance of cloud applications setting up thresholds and information. However, it suggests future research to identify issues early, maintain performance, ensure availability, plan capacity, and comply with regulatory and security best practices. An ineffective monitoring will not identify performance bottlenecks and impact user experience and productivity.

The [17] shares and discusses the importance of the cloud and testing because security vulnerabilities could remain undetected, exposing the organization to data breaches and compliance violations. Also, an improper monitoring configuration may lead to several issues in their respective environments, risking the stability and reliability of the overall solution.

Additionally, the lack of accurate monitoring data hinders effective capacity planning[18] discusses the importance of capacity planning but lacks how to do this for large scale systems and decision-making, causing resource insufficient scalability.

The following section outlines how to plan for monitoring for security, capacity and proactive approach with simple flow diagrams by discussing what, why and how because Proactive monitoring is essential for early detection and timely response to potential issues.

Sample table to understand planning for thresholds and metrics, but most importantly, monitoring should inform about the deployment is multi region or zonal as well. How will this metric be notified. The table below is only an example of how to think about a plan for it so that you can sustain zonal, regional or hardware failures in the cloud.

It is essential for engineering, operations, Infrastructure and other business teams to stay up to date on what is new. Shared the document, which outlines the following information for Azure[19], but readers can explore it for other providers. The following diagram illustrates how different dashboards provide different views into the system as a whole, as one monitoring dashboard may not fit all; more details in the link [20]

6. Results and Discussion

This document has outlined scenarios that focus on resiliency and prepare your workload to handle unexpected outages. Generally, you should leverage the Cloud Adoption Framework and Well-Architected Framework from the respective cloud providers. Documentation is shared to help one get started. The Cloud Adoption Framework provides best practices to help you digitally transform and accelerate your business outcomes through innovative use of available cloud services from major providers below:

Cloud Adoption Framework -Amazon Web Services [21], [22], [23] and each provider might have their own. Also, it is recommended to a well architected reliability review every 6 to 9 months [24]. Staying connected with cloud provider updates and service outage information is critical for the life cycle to avoid any disruption to the business example of Amazon Web Services [25].

References

[1] Benjamin Kettner, and Frank Geisler, *Achieving Resiliency*, Pro Serverless Data Handling with Microsoft Azure: Architecting ETL and Data-Driven Applications in the Cloud, Apress, Berkeley, CA pp. 195-211, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] P. Chinnasamy et al., "Providing Resilience on Cloud Computing," *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, pp. 1-4, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] K. Tirumala Rao, Sujatha, and N. Leelavathy, "Infrastructure Resiliency in Cloud Computing," *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Singapore, pp. 203-215, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Joshua Baron et al., "An Architecture for a Resilient Cloud Computing Infrastructure," *2013 IEEE International Conference on Technologies for Homeland Security (HST)*, Waltham, MA, USA, pp. 390-395, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[5] Fei Hu et al., "A Review on Cloud Computing: Design Challenges in Architecture and Security," *Journal of Computing and Information Technology*, vol. 19, no. 1, pp. 25-55, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[6] Mouna Jouini, and Latifa Ben Arfa Rabai, *Design Challenges of Cloud Computing*, Enterprise Management Strategies in the Era of Cloud Computing, IGI Global, pp. 1-25, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[7] Mythry Vuyyuru et al., "An Overview of Cloud Computing Technology," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2, no. 3, pp. 244-246, 2012. [Google Scholar] [Publisher Link]

[8] Attila Albini, and Zoltan Rajnai, "General Architecture of Cloud," *Procedia Manufacturing*, vol. 22, pp. 485-490, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[9] Michael Kavis, *Security Design in the Cloud*, Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS), Wiley, pp. 99-118, 2014. [CrossRef] [Publisher Link]

[10] Michael Kavis, *It Starts with Architecture*, Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS), Wiley, pp. 41-54, 2014. [CrossRef] [Publisher Link]

[11] Khalid Alhamazani et al., "An Overview of the Commercial Cloud Monitoring Tools: Research Dimensions, Design Issues, and State-of-the-art," *Computing*, vol. 97, pp. 357-377, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[12] Michael Kavis, *Creating a Centralized Logging Strategy*, Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS), Wiley, pp. 119-124, 2014. [CrossRef] [Publisher Link]

[13] Cynthia Treger, Understanding ExpressRoute Private Peering to Address ExpressRoute Resiliency, Azure Networking Blog, 2024. [Online]. Available: https://techcommunity.microsoft.com/t5/azure-networking-blog/understanding-expressroute-private-peering-to-address/ba-p/4081850

[14] TerryLanfear et al., Shared responsibility in the cloud, Microsoft Azure, 2023. [Online]. Available: https://learn.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility

[15] Haresh Nandwani, Lewis Taylor, and Bonnie McClure, Understand Resiliency Patterns and Trade-Offs to Architect Efficiently in the Cloud, AWS Architecture Blog, 2023. [Online]. Available: https://aws.amazon.com/blogs/architecture/understand-resiliency-patterns-and-trade-offs-to-architect-efficiently-in-the-cloud/

[16] Peschka Steve, "Monitoring and Analysis of Cloud-Based Applications," *U.S. Patent US10972370B1*, 2021. [Google Scholar] [Publisher Link]

[17] Zhongli Na, Wei Liu, and Kai Li, "Implementation of Cloud Component for Security Monitoring and Comprehensive Guarantee of Identifier Resolution System," *2022 3rd Information and Communication Technology Convergence (ICTC)*, Nanjing, China, pp. 167-172, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Carlos Diego Cavalcanti Pereira, "A Functional Paradigm for Capacity Planning of Cloud Computing Workloads," *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, Madrid, ES, pp. 281-283, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Max Melcher, Staying Up to Date with Azure, Azure Architecture Blog, 2020. [Online]. Available: https://techcommunity.microsoft.com/t5/azure-architecture-blog/staying-up-to-date-with-azure/ba-p/1784501

[20] John O'Shea, Building Dashboards for Operational Visibility, The Amazon Builders' Library. [Online]. Available: https://aws.amazon.com/builders-library/building-dashboards-for-operational-visibility/

[21] AWS Cloud Adoption Framework (AWS CAF), AWS Customer Enablement, 2021. [Online]. Available: https://aws.amazon.com/cloud-adoption-framework/

[22] Google Cloud, "*The Google Cloud Adoption Framework*," pp. 1-33. [Publisher Link]

[23] Microsoft Cloud Adoption Framework for Azure, Microsoft Azure. [Online]. Available: https://azure.microsoft.com/en-us/solutions/cloud-enablement/cloud-adoption-framework

[24] Peter Panec, The 5 Steps of the AWS Well-Architected Framework Review Process, Cprime. [Online]. Available: https://www.cprime.com/resources/blog/5-steps-of-aws-well-architected-review-process/

[25] AWS Health Dashboard-Service Health, AWS Health, 2024. [Online]. Available: https://docs.aws.amazon.com/health/latest/ug/aws-health-dashboard-status.html