Reliability on The Cloud

The reliability pillar includes the ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues.

The reliability pillar includes the ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues.

Design Principles: The five design principles for reliability on the cloud:-

Test recovery procedures:

In an on-premises environment, testing is often conducted to prove the system works in a particular scenario. Testing is not typically used to validate recovery strategies. In the cloud, you can test how your system fails, and you can validate your recovery procedures. You can use automation to simulate different failures or to recreate scenarios that led to failures before. This exposes failure pathways that you can test and rectify before a real failure scenario, reducing the risk of components failing that have not been tested before.

Automatically recover from failure:

By monitoring a system for key performance indicators (KPIs), you can trigger automation when a threshold is breached. This allows for automatic notification and tracking of failures, and for automated recovery processes that work around or repair the failure. With more sophisticated automation, it’s possible to anticipate and remediate failures before they occur.

Scale horizontally to increase aggregate system availability:

Replace one large resource with multiple small resources to reduce the impact of a single failure on the overall system. Distribute requests across multiple, smaller resources to ensure that they don’t share a common point of failure.

Stop guessing capacity:

A common cause of failure in on-premises systems is resource saturation, when the demands placed on a system exceed the capacity of that system (this is often the objective of denial of service attacks). In the cloud, you can monitor demand and system utilization, and automate the addition or removal of resources to maintain the optimal level to satisfy demand without over or under-provisioning.

Manage change in automation:

Changes to your infrastructure should be done using automation. The changes that need to be managed are changes to the automation.

Definition

There are three best practice areas for reliability in the cloud:

1. Foundations
2. Change Management
3. Failure Management

To achieve reliability, a system must have a well-planned foundation and monitoring in place, with mechanisms for handling changes in demand or requirements. The system should be designed to detect failure and automatically heal itself.

Best Practices Foundations

Before architecting any system, foundational requirements that influence reliability should be in place. For example, you must have sufficient network bandwidth to your data center. These requirements are sometimes neglected (because they are beyond a single project's scope).

This neglect can have a significant impact on the ability to deliver a reliable system. In an on-premises environment, these requirements can cause long lead times due to dependencies and therefore must be incorporated during initial planning.

The cloud is designed to be essentially limitless, so it is the responsibility of the cloud provider to satisfy the requirement for sufficient networking and compute capacity, while you are free to change resource size and allocation,

such as the size of storage devices, on demand. The following questions focus on foundations considerations for reliability

REL 1: How are you managing service limits for your accounts?

REL 2: How are you planning your network topology ?

Service limits (an upper limit on the number of each resource your team can request) to protect you from accidentally over-provisioning resources. You will need to have governance and processes in place to monitor and change these limits to meet your business needs. As you adopt the cloud, you may need to plan integration with existing on-premises resources (a hybrid approach). A hybrid model enables the gradual transition to an all-in cloud approach over time. Therefore, it's important to have a design for how your cloud and on-premises resources will interact as a network topology.

Change Management

Being aware of how change affects a system allows you to plan proactively, and monitoring allows you to quickly identify trends that could lead to capacity issues or SLA breaches. In traditional environments, change-control processes are often manual and must be carefully coordinated with auditing to effectively control who makes changes and when they are made. You can monitor the behavior of a system and automate the response to KPIs, for example, by adding additional servers as a system gains more users. You can control who has permission to make system changes and audit the history of these changes. The following questions focus on change management considerations for reliability

REL 3: How does your system adapt to changes in demand?

REL 4: How are you monitoring AWS resources?

REL 5: How are you executing change?

When you architect a system to automatically add and remove resources in response to changes in demand, this not only increases reliability but also ensures that business success doesn't become a burden. With monitoring in place, your team will be automatically alerted when KPIs deviate from expected norms.

Automatic logging of changes to your environment allows you to audit and

quickly identify actions that might have impacted reliability. Controls on

change management ensure that you can enforce the rules that deliver the reliability you need.

Failure Management

In any system of reasonable complexity it is expected that failures will occur. It is generally of interest to know how to become aware of these failures, respond to them, and prevent them from happening again. With all providers, you can take advantage of automation to react to monitoring data. For example, when a particular metric crosses a threshold, you can trigger an automated action to remedy the problem.

Also, rather than trying to diagnose and fix a failed resource that is part of your production environment, you can replace it with a new one and carry out the analysis on the failed resource out of band. Since the cloud enables you to stand up temporary versions of a whole system at low cost, you can use automated testing to verify full recovery processes.

The following questions focus on failure management considerations for reliability

REL 6: How are you backing up your data?

REL 7: How does your system withstand component failures?

REL 8: How are you testing your resiliency?

REL 9: How are you planning for disaster recovery?

Regularly back up your data and test your backup files to ensure you can recover from both logical and physical errors. A key to managing failure is the frequent and automated testing of systems to cause failure, and then observe how they recover. Do this on a regular schedule and ensure that such testing is also triggered after significant system changes. Actively track KPIs, such as the recovery time objective (RTO) and recovery point objective (RPO), to assess a system's resiliency (especially under failure-testing scenarios).

Tracking KPIs will help you identify and mitigate single points of failure. The objective is to thoroughly test your system-recovery processes so that you are confident that you can recover all your data and continue to serve your

confident that you can recover all your data and continue to serve your

customers, even in the face of sustained problems. Your recovery processes should be as well exercised as your normal production processes.

Key Services

You would essentially use CloudWatch, which monitors runtime metrics.

The following services and features support the three areas in reliability:

Foundations:

IAM enables you to securely control access to AWS services and resources. Amazon VPC lets you provision a private, isolated section of the AWS Cloud where you can launch AWS resources in a virtual network. AWS Trusted Advisor provides visibility into service limits. AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards web applications running on AWS.

Change Management:

AWS CloudTrail records AWS API calls for your account and delivers log files to you for auditing. AWS Config provides a detailed inventory of your AWS resources and configuration, and continuously records configuration changes. Auto Scaling is a service that will provide an automated demand management for a deployed workload. CloudWatch provides the ability to alert on metrics, including custom metrics.

Failure Management:

AWS CloudFormation provides templates for the creation of AWS resources and provisions them in an orderly and predictable fashion. Amazon S3 provides a highly durable service to keep backups. Amazon Glacier provides highly durable archives. AWS KMS provides a reliable key management system that integrates with many AWS services.