

Executive Overview Section



Thomas Bronack, President

Data Center Assistance Group, LLC

bronackt@dcag.com | bronackt@gmail.com

(917) 673-6992

<https://www.dcag.com>

Executive Briefing

Producing Applications Recovery Certification with AWS

Author: Thomas Bronack, President, Data Center Assistance Group, LLC

Contact: bronackt@dcag.com | bronackt@gmail.com | (917) 673-6992 | www.dcag.com

Purpose

This briefing provides executives with a high-level overview of how AWS enables certifiable application recovery to meet business continuity requirements. It ensures mission-critical systems are tested, resilient, and compliant before entering production.

Key Takeaways

1. Application Recovery Classification

- Continuously Available (CA): Zero downtime – seconds RTO.
- Highly Available (HA): Minimal downtime – minutes RTO.
- Normally Recoverable (NR): Acceptable downtime – hours RTO.

2. Structured Certification Process

- Assessment & Classification → Architecture Design → Build & Configure → Test & Validate → Authorization to Operate → Continuous Monitoring

3. AWS Service Mapping

- CA: Global Accelerator, Route 53, Multi-AZ, DynamoDB Global Tables.
- HA: Elastic Load Balancing, RDS Multi-AZ, Auto Scaling.
- NR: S3, AWS Backup, Glacier, CloudEndure.

Diagram 1: Recovery Classification

Recovery Type:	Recovery Method:	AZ & Regions:	AWS Tool:
CA (Seconds)	Global Accelerator	Multi-AZ	DynamoDB
HA (Minutes)	Load Balancer	RDS Multi-AZ	Auto Scaling
NR (Hours)	S3	AWS Backup	Glacier, CloudEndure

AWS DR Strategy Spectrum

- Backup & Restore → Basic protection (low cost, longer RTO).
- Pilot Light → Minimal infrastructure, scaled up after event.
- → Always-on secondary environment.
- Multi-Site Active/Active → Simultaneous global operations, near-zero downtime.

Diagram 2: DR Strategies vs. RTO

Type of Recovery:	Duration:
Backup/Restore	Hours
Pilot Light	Minutes – Hours
Warm Standby	Minutes
Active/Active	Seconds

Security & Compliance




- AWS GuardDuty, DevOps Guru, Security Hub: Early detection & automated remediation.
- Vulnerability Management (SBOM/CTEM): Ensure apps are free of known CVEs before production.
- Standards Alignment: NIST CSF 2.0, ISO 22301, CIA Triad, FedRAMP.

Executive Value

- **Reduced Risk:** Certainty that workloads can survive disasters.
- **Regulatory Readiness:** Aligns with BC/DR certifications and compliance mandates.
- **Cost Optimization:** Tailored strategies balance resilience with financial impact.
- **Cyber Resilience:** Malware/ransomware recovery built into the recovery plan.

Call to Action

Organizations cannot afford uncertainty when it comes to downtime. Recovery certification ensures systems are:

- Tested 
- Resilient 
- Audit-Ready 

Contact Thomas Bronack to explore how AWS-driven recovery certification can safeguard your enterprise.

Producing Applications Recovery Certification with AWS



Data Center Assistance Group, LLC

Thomas Bronack, President


bronackt@dcag.com | bronacckt@gmail.com

(917) 673-6992

<https://www.dcag.com>

Table of Contents

Contents

Executive Briefing	2
Producing Applications Recovery Certification with AWS	2
Purpose.....	2
Key Takeaways.....	2
 Diagram 1: Recovery Classification.....	2
AWS DR Strategy Spectrum.....	3
Diagram 2: DR Strategies vs. RTO	3
Security & Compliance	3
Executive Value.....	3
Call to Action	3
Executive Summary.....	7
Application Recovery Classification.....	7
Phase 1: Assessment & Classification.....	7
AWS Regions and Availability Zones	7
AWS Resilience Hub.....	8
AWS Well-Architected Six Pillars	9
CSP Cloud Shared Responsibility Model.....	11
Availability Zones and Regions.	11
AWS Disaster Recovery Strategies with Regions and Availability Zones	12
Backup and restore.	12
AWS Backup and Restoration strategies.....	13
Resilience Patterns and Recovery Groups.....	13
Anatomy of a Disaster Event	14
Business Continuity and Disaster Recovery Types.....	14
Evolution of Disaster Recovery	15
AWS Backup and Restoration services with Recovery Points	15
Recovering from Malware and Ransomware	17

Phase 2: AWS Architecture Design	17
AWS GuardDuty – Plans and Descriptions	18
Phase 3: Build & Configure	19
Phase 4: Test & Validate.....	19
Phase 5: Authorization to Operate (ATO).....	19
Phase 6: Continuous Monitoring & Improvement	19
Amazon Guru services as an Application Factory Control Gate	20
AWS Functions for Early Detection & Mitigation	20
Disaster Recovery Plan.....	21
Architecting to withstand failures	21
AWS Elastic DR Services.....	21
AWS Elastic Disaster Recovery to operate disaster recovery scenarios.....	22
AWS Security and Governance features	22
Warm standby	23
AWS Warm Standby approach for disaster recovery operations	23
Multi-site active/active or Hot Standby active/passive.....	23
AWS Active/Active Recovery approach with Multi-Regions	24
AWS services with point-in-time backup.....	25
AWS Cloud Formation.....	26
Detecting a disaster event and enacting a recovery plan	26
Testing the Disaster Recovery Plan	28
AWS Config	29
Conclusion.....	29
Recognizing Disaster Events and Activating Recovery Plans.....	29
Problem/Incident Management and Disaster Recovery Management relationship.....	30
Recovery Process with Recovery Point	30
Call to Action.....	30

Executive Summary

This document provides a structured, AWS-enabled methodology for building, testing, and certifying application business recovery capabilities. It is designed to meet Business Continuity Certification requirements for Continuously Available (CA), Highly Available (HA), and Normally Recoverable (NR) applications. It includes detailed phases, tasks, AWS service mappings, and testing protocols to ensure Recovery Time Objectives (RTOs) and Recovery Groups are met before production deployment.

Application Recovery Classification

Application Type	Description	RTO Target	AWS Services Example
Continuously Available (CA)	Zero downtime tolerance	Seconds	AWS Global Accelerator, Route 53, Multi-AZ, DynamoDB Global Tables
Highly Available (HA)	Minimal downtime tolerance	Minutes	Elastic Load Balancing, RDS Multi-AZ, Auto Scaling
Normally Recoverable (NR)	Acceptable downtime for recovery	Hours	S3, AWS Backup, Glacier, CloudEndure

Phase 1: Assessment & Classification

- Identify business criticality of application.
- Map application dependencies in AWS.
- Classify into CA, HA, or NR categories.

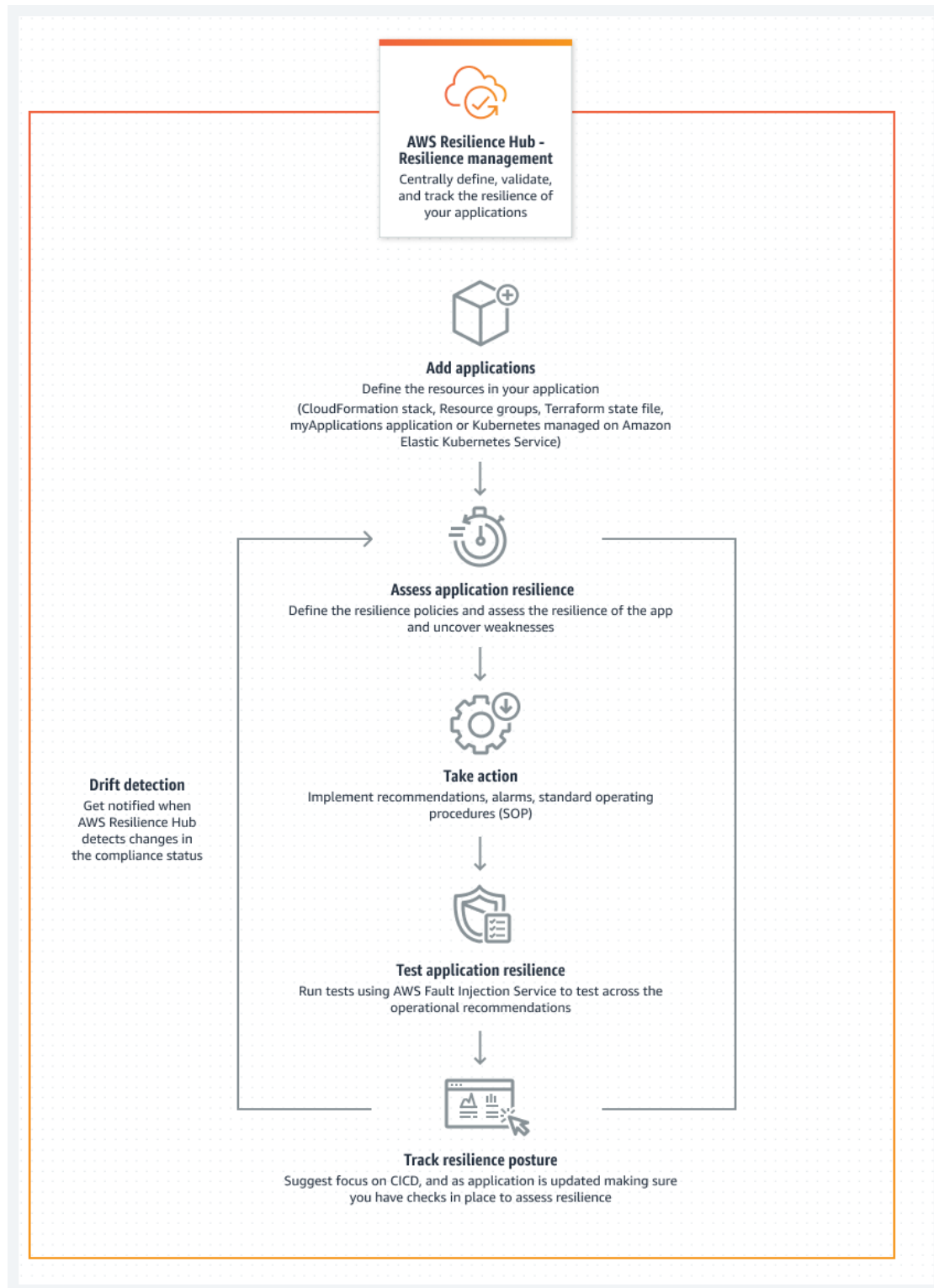
AWS Regions and Availability Zones

Disaster recovery strategies available to you within AWS can be broadly categorized into four approaches, ranging from the low cost and low complexity of making backups to more complex strategies using multiple active Regions. Active/passive strategies use an active site (such as an AWS Region) to host the workload and serve traffic. The passive site (such as a different AWS Region) is used for recovery. The passive site does not actively serve traffic until a failover event is triggered.

It is critical to regularly assess and evaluate your disaster recovery strategy so that you have confidence in invoking it, should it become necessary. Use [AWS Resilience Hub](#) to

continuously validate and track the resilience of your AWS workloads, including whether you are likely to meet your RTO and RPO targets.

AWS Resilience Hub



Link to AWS Disaster [Recovery](#) Strategies

[Link to All AWS Services Documentation](#)

AWS Well-Architected Six Pillars

The framework is based on six pillars:

1. **Operational Excellence:** Operational excellence (OE) is a commitment to build software correctly while consistently delivering great customer experience. The operational excellence pillar contains best practices for organizing your team, designing your workload, operating it at scale, and evolving it over time.

The goal of operational excellence is to get new features and bug fixes into customers' hands quickly and reliably. Organizations that invest in operational excellence consistently delight customers while building new features, making changes, and dealing with failures. Along the way, operational excellence drives towards continuous integration and continuous delivery (CI/CD) by helping developers achieve high quality results consistently.

Utilize Vulnerability Management practices to run SBOMs (Software Bill of Materials) against application to identify programs and open-source modules, then examine public vulnerability databases (i.e., NVD – National Vulnerability Database) to determine if the programs have vulnerabilities (CVE – Common Vulnerability Enumeration / Exploitation) associated with them. If so, use the recommended Update Path (Patch or New Release) to mitigate the vulnerability prior to production acceptance. This will guarantee all components are at current release levels and free of known vulnerabilities. Use Continuous Threat Exploitation Management (CTEM) to identify New Vulnerabilities in production, so that rapid mitigation can be accomplished before hackers take advantage of the new vulnerability.

2. **Security:** The security pillar describes how to take advantage of cloud technologies to protect data, systems, and assets in a way that can improve your security posture. This paper provides in-depth, best-practice guidance for architecting secure workloads on AWS.

Follow the CSF 2.0 Cybersecurity Framework 2.0 guidelines to ensure compliance with security and governance policies. Utilize CIA (Confidentiality, Integrity, and Availability) practices to protect against malware.

3. **Reliability:** The reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it is expected to. This includes the ability to operate and test the workload through its total lifecycle. This paper provides in-depth practice guidance for implementing reliable workloads on AWS.

Utilize an Application Factory concept with Control Gates to validate compliance to required safeguards and standards.

4. **Performance Efficiency:** The performance efficiency pillar includes the ability to use cloud resources efficiently to meet performance requirements, and to maintain that efficiency as demand changes and technologies evolve. Use scalability and load balancing functions to achieve this goal.
5. **Cost Optimization:** Cost optimization is a continual process of refinement and improvement over the span of a workload's lifecycle. The practices in this paper help you build and operate cost-aware workloads that achieve business outcomes while minimizing costs and allowing your organization to maximize its return on investment. Ensure adherence to Service Level Agreements (SLAs) and Error Budgets defined as best practices for supporting clients.
6. **Sustainability:** The discipline of sustainability addresses the long-term environmental, economic, and societal impact of your business activities. The [United Nations World Commission on Environment and Development](#) defines sustainable development as “development that meets the needs of the present without compromising the ability of future generations to meet their own needs.” Your business or organization can have negative environmental impacts like direct or indirect carbon emissions, unrecyclable waste, and damage to shared resources like clean water.

When building cloud workloads, the practice of sustainability is understanding the impacts of the services used, quantifying impacts through the entire workload lifecycle, and applying design principles and best practices to reduce these impacts. This document focuses on environmental impacts, especially energy consumption and efficiency since they are important levers for architects to inform direct action to reduce resource usage.

When focusing on environmental impacts, you should understand how these impacts are typically accounted for and the follow-on impacts to your organization's own emissions accounting. The [Greenhouse Gas Protocol](#) organizes carbon emissions into the following scopes, along with relevant emission examples within each scope for a cloud provider such as AWS:

Scope 1: All direct emissions from the activities of an organization or under their control. For example, fuel combustion by data center backup generators.

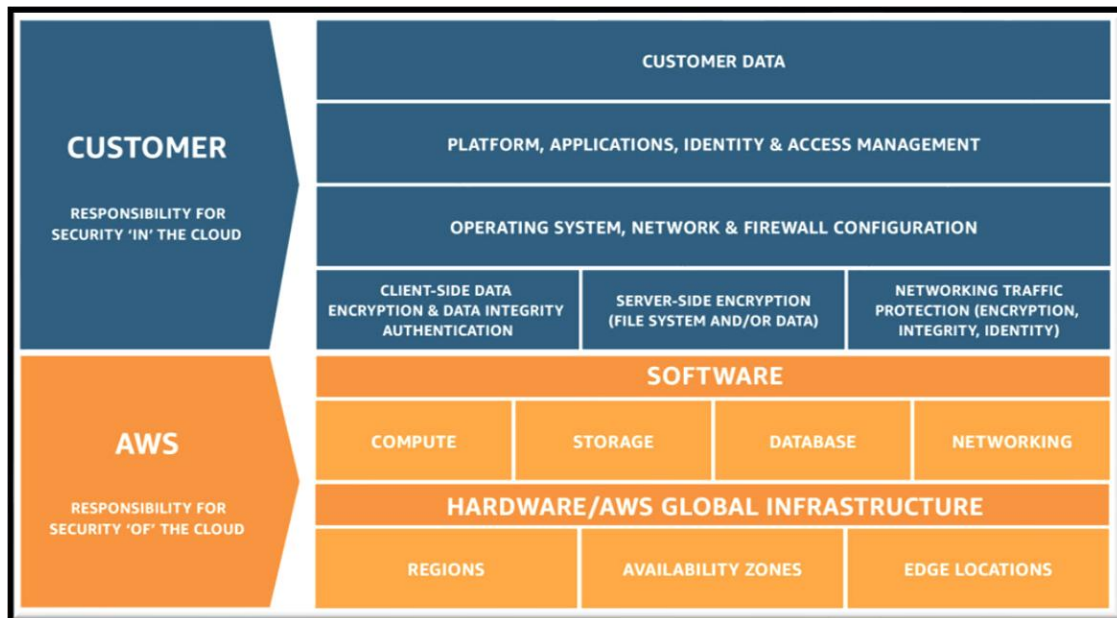
Scope 2: Indirect emissions from electricity purchased and used to power data centers and other facilities. For example, emissions from commercial power generation.

Scope 3: All other indirect emissions from activities of an organization from sources it does not control. AWS examples include emissions related to data center construction, and the manufacture and transportation of IT hardware deployed in data centers.

From an AWS customer perspective, emissions from your workloads running on AWS are accounted for as indirect emissions, and part of your Scope 3 emissions. Each workload deployed generates a fraction of the total AWS emissions from each of the previous scopes. The actual amount varies per workload and depends on factors including the AWS services used, the energy consumed by those services, the carbon intensity of the electric grids serving the AWS data centers where they run, and the AWS procurement of renewable energy.

This document first describes a shared responsibility model for environmental sustainability and then provides architectural best practices so you can minimize the impact of your workloads by reducing the total resources required for them to run in AWS data centers.

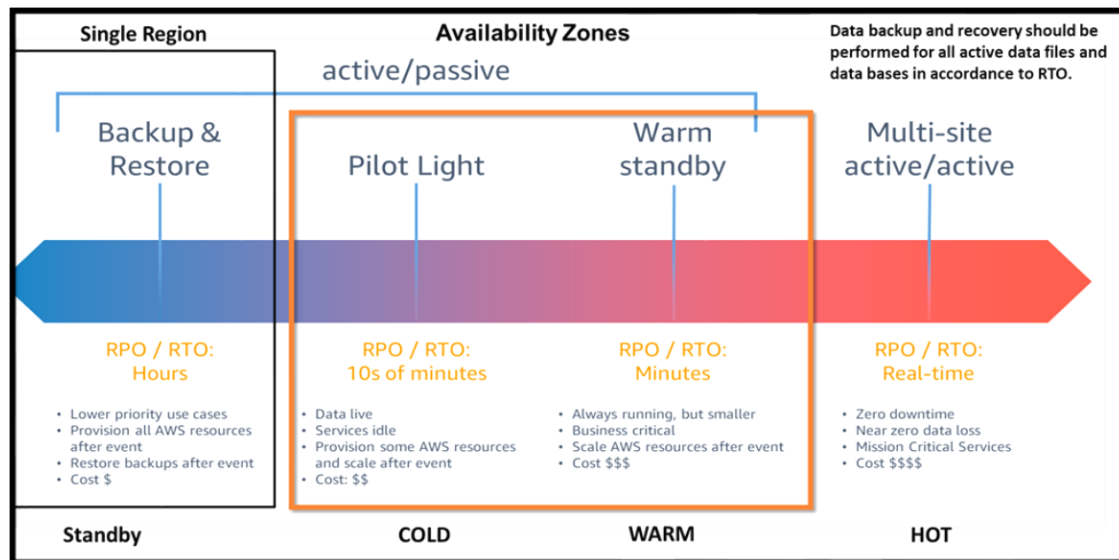
CSP Cloud Shared Responsibility Model



Availability Zones and Regions.

Availability Zones (AZs) are Physical AWS Data Centers, while Regions (Regs) and data centers within the Availability Zone. Continuous Availability and High Availability applications require the use of multiple AZs, while Normal Recovery can be protected within a single AZ using multiple Regions.

AWS Disaster Recovery Strategies with Regions and Availability Zones



For a disaster event based on disruption or loss of one physical data center for a [well-architected](#), highly available workload, you may only require a backup and restore approach to disaster recovery. If your definition of a disaster goes beyond the disruption or loss of a physical data center to that of a Region or if you are subject to regulatory requirements that require it, then you should consider Pilot Light, Warm Standby, or Multi-Site Active/Active.

When choosing your strategy, and the AWS resources to implement it, keep in mind that within AWS, we commonly divide services into the *data plane* and the *control plane*. The data plane is responsible for delivering real-time service while control planes are used to configure the environment. For maximum resiliency, you should use only data plane operations as part of your failover operation. This is because the data planes typically have higher availability design goals than the control planes.

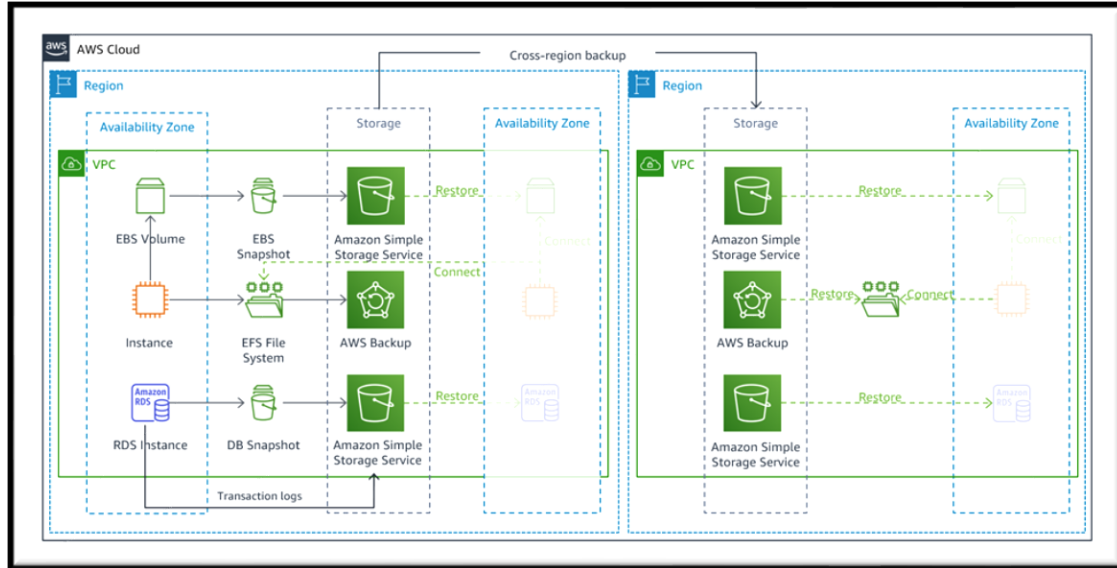
Recover Hardware and Network resources first, then load applications and their data prior to connecting users to the application. If hardware, systems, applications, data, and your stack are in sync within multiple data centers, then an instantaneous recovery can be achieved that is transparent to the end user.

Backup and restore.

Backup and restoration are a suitable approach for mitigating against data loss or corruption. This approach can also be used to mitigate against a regional disaster by replicating data to other AWS Regions, or to mitigate lack of redundancy for workloads deployed to a single Availability Zone. In addition to data, you must redeploy the infrastructure, configuration, and application code in the recovery Region. To enable infrastructure to be redeployed quickly without errors, you should always deploy using infrastructure as code (IaC – JSON files used to define an environment that can be executed to allocate and configure required resources for a recovery event) using services such as [AWS CloudFormation](#) or the [AWS Cloud Development Kit \(AWS CDK\)](#). Without IaC, it may

be complex to restore workloads in the recovery Region, which will lead to increased recovery times and exceed your RTO. In addition to user data, be sure to also back up code and configuration, including [Amazon Machine Images \(AMIs\)](#) you use to create Amazon EC2 instances. You can use [AWS CodePipeline](#) to automate redeployment of application code and configuration.

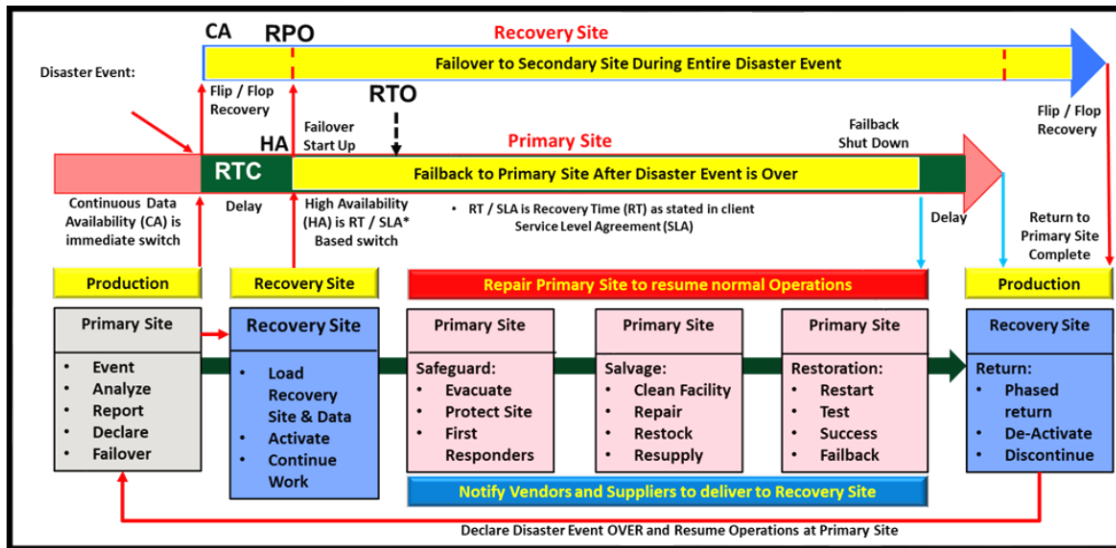
AWS Backup and Restoration strategies



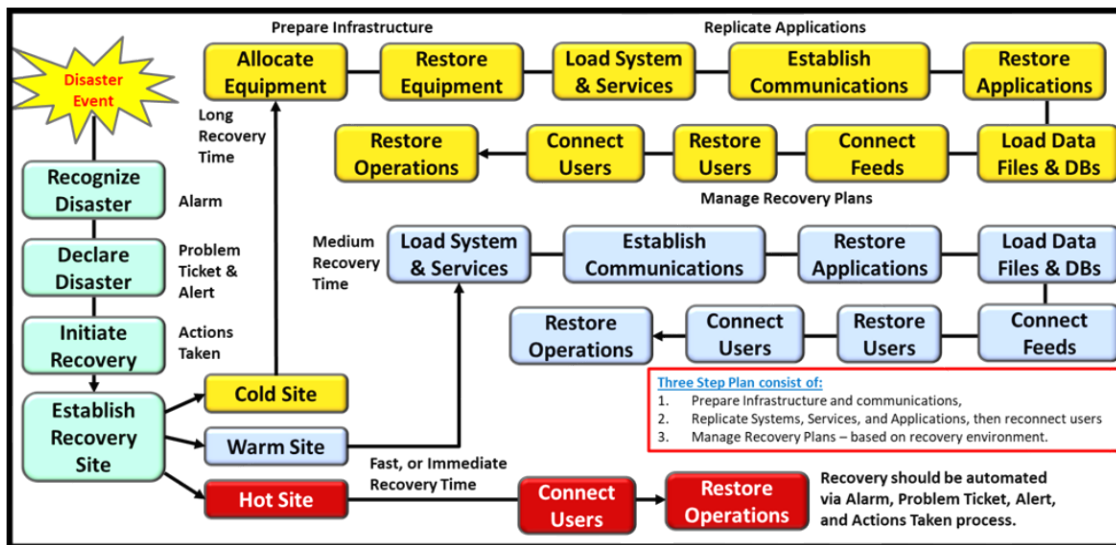
Resiliency Patterns and Recovery Groups

Resiliency Patterns	Single Region	Multiple Regions		
	In-Region	Active Standby (Pilot Light)	Active-Passive (Warm Standby)	Active-Active (Multi-Site)
Pattern Profile	<ol style="list-style-type: none"> 1. TRANSACTIONAL TRAFFIC - handled by primary region only 2. No multi-region INFRASTRUCTURE 3. APPLICATION code only available in single region 4. Multi-region RECOVERY not supported 	<ol style="list-style-type: none"> 1. TRANSACTIONAL TRAFFIC - handled by primary region only 2. INFRASTRUCTURE available on stand-by 3. APPLICATION provisioned, but in shutdown state 	<ol style="list-style-type: none"> 1. TRANSACTIONAL TRAFFIC - handled by primary region only 2. INFRASTRUCTURE available on stand-by 3. Minimal APPLICATION footprint running in 2nd region (all components are spun up and available with min. capacity, where application) 	<ol style="list-style-type: none"> 1. TRANSACTIONAL TRAFFIC - handled by primary region only 2. INFRASTRUCTURE always available in both regions 3. APPLICATION stack running active/active multi-region
Reserve Capacity			Required RESERVE CAPACITY	Required RESERVE CAPACITY
Cross-Region Maintenance	None	<ol style="list-style-type: none"> 1. Maintain PERSISTENT DATA REPLICATION infrastructure 2. APPLICATION CODE maintained for currency in BOTH REGIONS 3. Operate Production from stand-by region periodically 	<ol style="list-style-type: none"> 1. Maintain PERSISTENT DATA REPLICATION infrastructure 2. APPLICATION CODE maintained for currency in BOTH REGIONS 3. Operate Production from stand-by region periodically 	<ol style="list-style-type: none"> 1. Maintain 2-WAY PERSISTENT DATA REPLICATION 2. APPLICATION CODE maintained for currency in BOTH REGIONS 3. Operate Production from stand-by region periodically
Recovery Steps	<ol style="list-style-type: none"> 1. ACQUIRE INFRASTRUCTURE 2. BUILD OUT infrastructure 3. DEPLOY application 4. RECOVER / RECREATE DATA 5. REDIRECT TRAFFIC to region 2 	<ol style="list-style-type: none"> 1. SCALE INFRASTRUCTURE 2. STARTUP application 3. FAILOVER TRAFFIC 	<ol style="list-style-type: none"> 1. AUTO-SCALE INFRASTRUCTURE 2. FAILOVER TRAFFIC 	<ol style="list-style-type: none"> 1. RECOVERY achieved through automated redirect of traffic
Recovery Group (RG)	RG7	RG 4-6	REG 1-3	RG 0
Recovery Time Design (RTD)	Days+	Hours (<8 hrs)	Minutes (<15 mins)	Real-Time (<5mins)
Recovery Point Design (RPCD)	Hours (<8 Hrs)	Minutes (<15 mins)	Minutes (<15 mins)	Real-Time (<0 mins)
Cloud Based Recovery Group Specifications		Preferred Patterns		

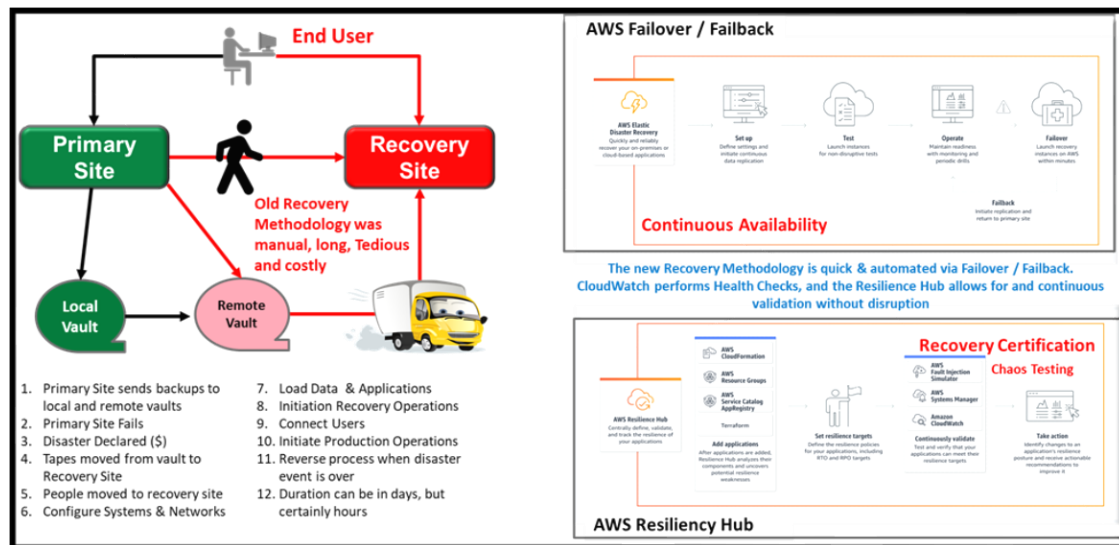
Anatomy of a Disaster Event



Business Continuity and Disaster Recovery Types



Evolution of Disaster Recovery



From the original disaster recovery method of backing systems up on tape, going to a recovery site, building the hardware and communications systems, loading systems, stack support products, applications, connecting users, and recovering applications to the current methods of Continuous Availability and High Availability, we have decreased recovery times from days to seconds with corresponding reductions in costs associated with disaster events.

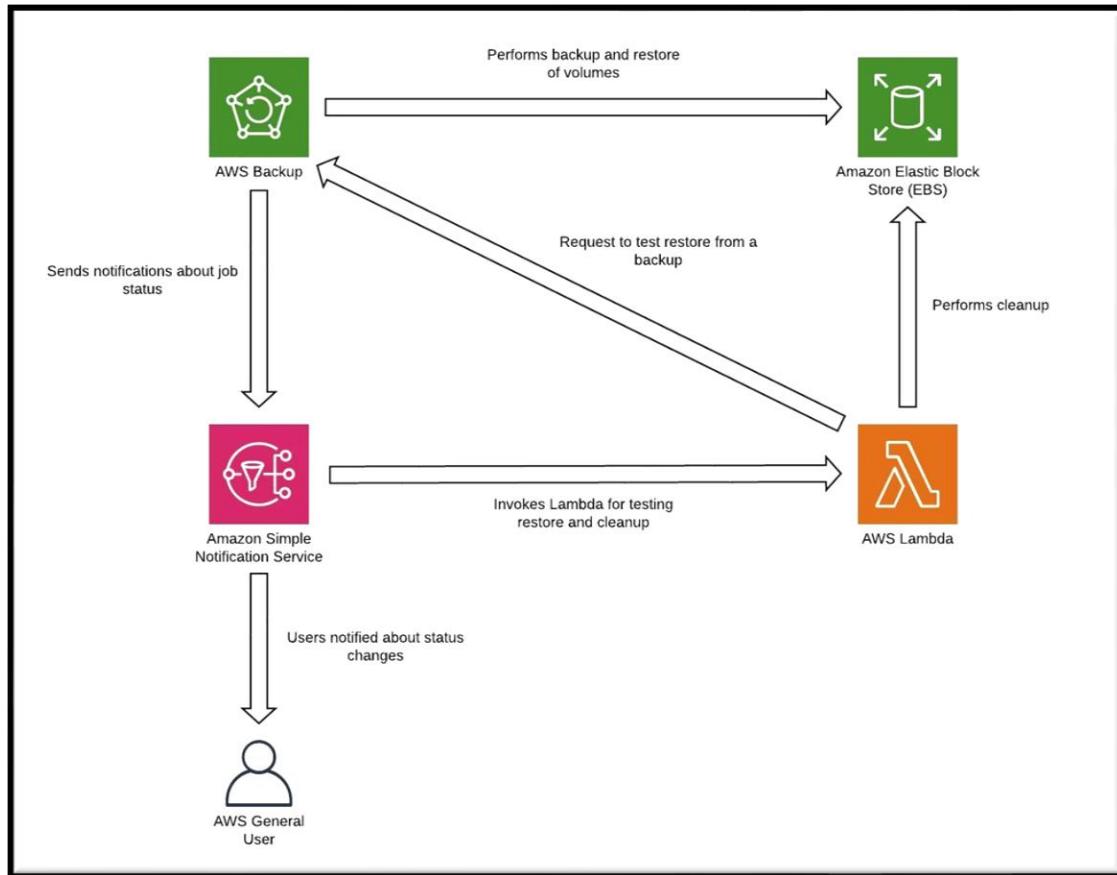
AWS Backup and Restoration services with Recovery Points

Your workload data will require a backup strategy that runs periodically or is continuous. How often you run your backup will determine your achievable recovery point (which should align to meet your RPO). The backup should also offer a way to restore it to the point in time in which it was taken. Backup with point-in-time recovery is available through the following services and resources:

- [Amazon Elastic Block Store \(Amazon EBS\) snapshot](#)
- [Amazon DynamoDB backup](#)
- [Amazon RDS snapshot](#)
- [Amazon Aurora DB snapshot](#)
- [Amazon EFS backup](#) (when using AWS Backup)
- [Amazon Redshift snapshot](#)
- [Amazon Neptune snapshot](#)
- [Amazon DocumentDB](#)
- [Amazon FSx for Windows File Server](#), [Amazon FSx for Lustre](#), [Amazon FSx for NetApp ONTAP](#), and [Amazon FSx for OpenZFS](#)

For Amazon Simple Storage Service (Amazon S3), you can use [Amazon S3 Cross-Region Replication \(CRR\)](#) to asynchronously copy objects to an S3 bucket in the DR region continuously, while providing versioning for the stored objects so that you can choose your

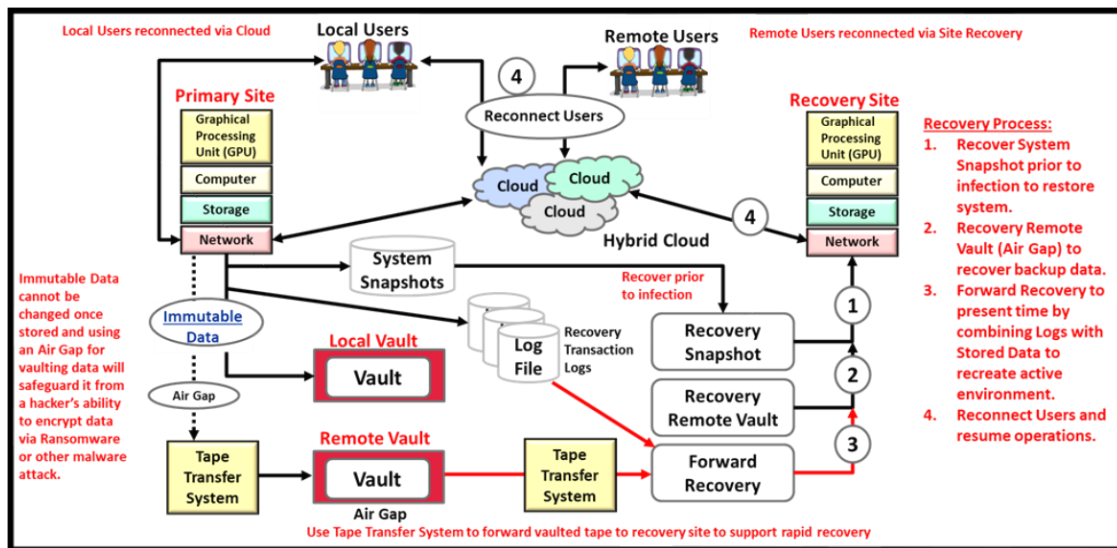
restoration point. Continuous replication of data has the advantage of being the shortest time (near zero) to back up your data but may not protect against disaster events such as data corruption or malicious attack (such as unauthorized data deletion) as well as point-in-time backups. Continuous replication is covered in the [AWS Services for Pilot Light](#) section.



Note

Your backup strategy must include evaluating your backups. See the [Testing Disaster Recovery](#) section for more information. Refer to the [AWS Well-Architected Lab: Testing Backup and Restore of Data](#) for a hands-on demonstration of implementation.

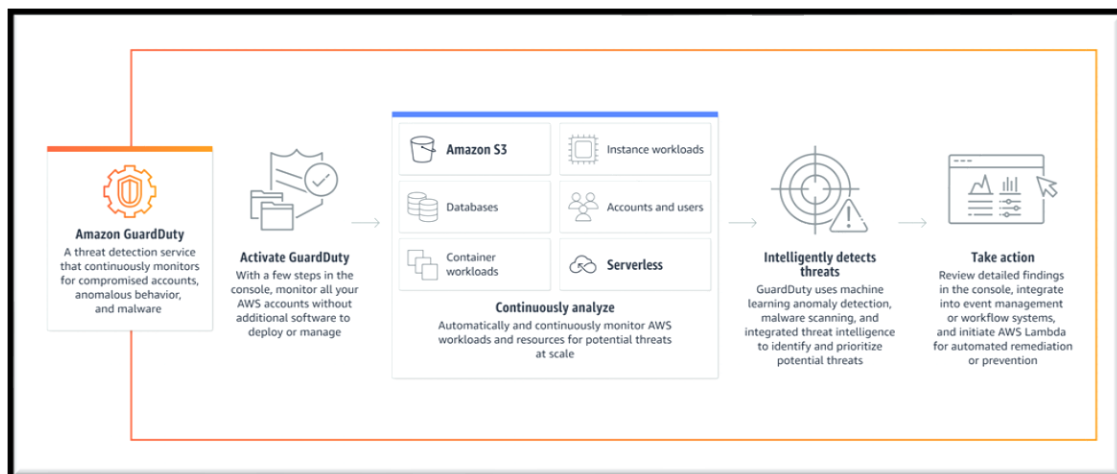
Recovering from Malware and Ransomware



To recover from Malware, viruses, or ransomware, you must select a Recovery Point prior to the infection of data encryption through a Recovery Point. This allows you to start with clean uncorrupted data. Then you must take your log files from the time of infection, clean then erase any malware or infection, then perform a forward recovery until synchronizing data to current operations. Taking data snapshots every 15 minutes will reduce your outage profile and make it easier and less time-consuming to synchronize data to its current state.

Phase 2: AWS Architecture Design

- Design AWS infrastructure for resilience based on classification.
- Select AWS services and redundancy options.
- Incorporate security best practices (IAM, GuardDuty, Config).

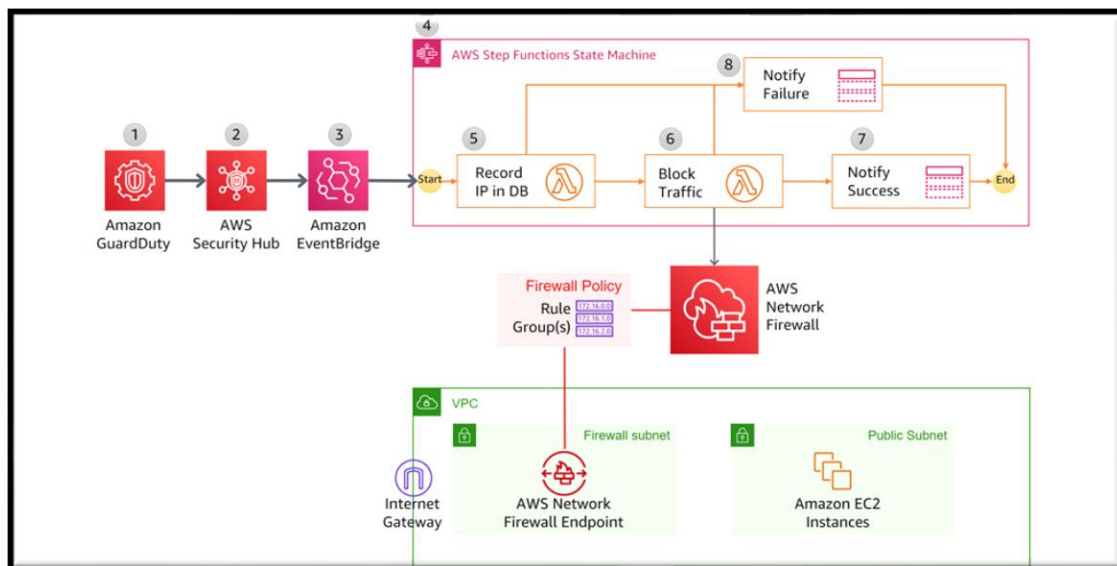


AWS GuardDuty – Plans and Descriptions

Protection plan	Description
S3 Protection	Identifies potential security risks such as data exfiltration and destruction attempts in your Amazon S3 buckets.
EKS Protection	EKS Audit Log Monitoring analyzes Kubernetes audit logs from your Amazon EKS clusters for potentially suspicious and malicious activities.
Runtime Monitoring	Monitors and analyze operating system-level events on your Amazon EKS, Amazon EC2, and Amazon ECS (including AWS Fargate), to detect potential runtime threats.
Malware Protection for EC2	Detects potential presence of malware by scanning the Amazon EBS volumes associated with your Amazon EC2 instances. There is an option to use this feature on-demand.
Malware Protection for S3	Detects potential presence of malware in the newly uploaded objects within your Amazon S3 buckets.
RDS Protection	Analyzes and profiles your RDS login activity for potential access threats to the supported Amazon Aurora and Amazon RDS databases.
Lambda Protection	Monitors Lambda network activity logs, starting with VPC flow logs, to detect threats to your AWS Lambda functions. Examples of these potential threats include crypto mining and communicating with malicious servers.

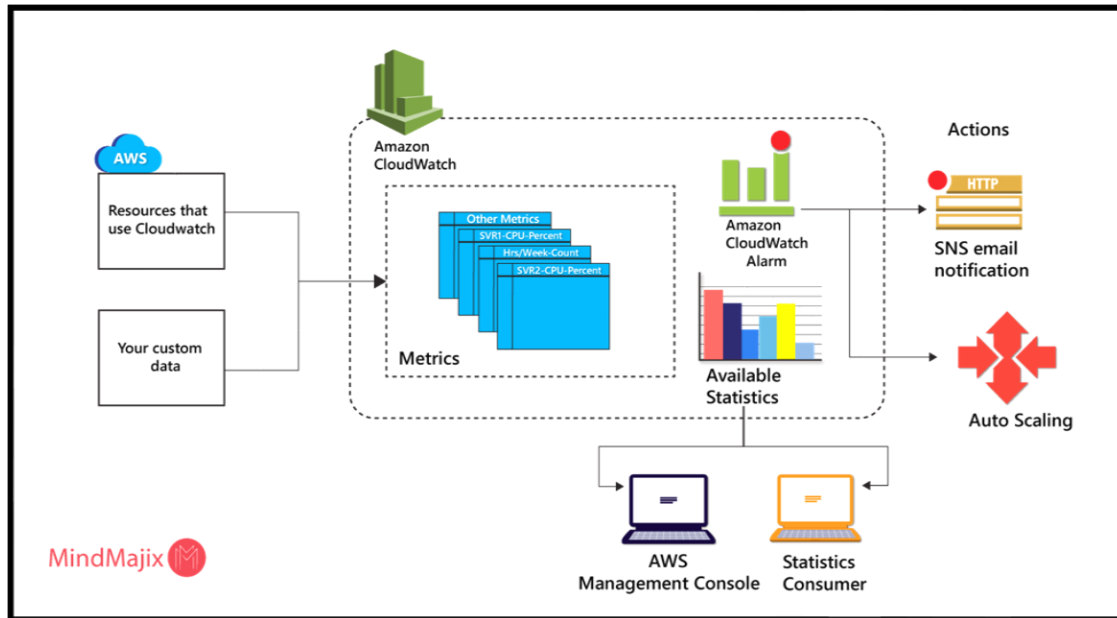
Note: Enable Malware Protection for S3 independently

GuardDuty offers flexibility to use Malware Protection for S3 independently, without enabling the Amazon GuardDuty service. For more information about getting started with only Malware Protection for S3, see [GuardDuty Malware Protection for S3](#). To use all other protection plans, you must enable the GuardDuty service.



Phase 3: Build & Configure

- Provision AWS resources.
- Implement backup and replication strategies.
- Set up monitoring (CloudWatch, X-Ray, CloudTrail).



Phase 4: Test & Validate

- Run recovery simulations.
- Measure actual RTO vs. target RTO.
- Document and resolve gaps.

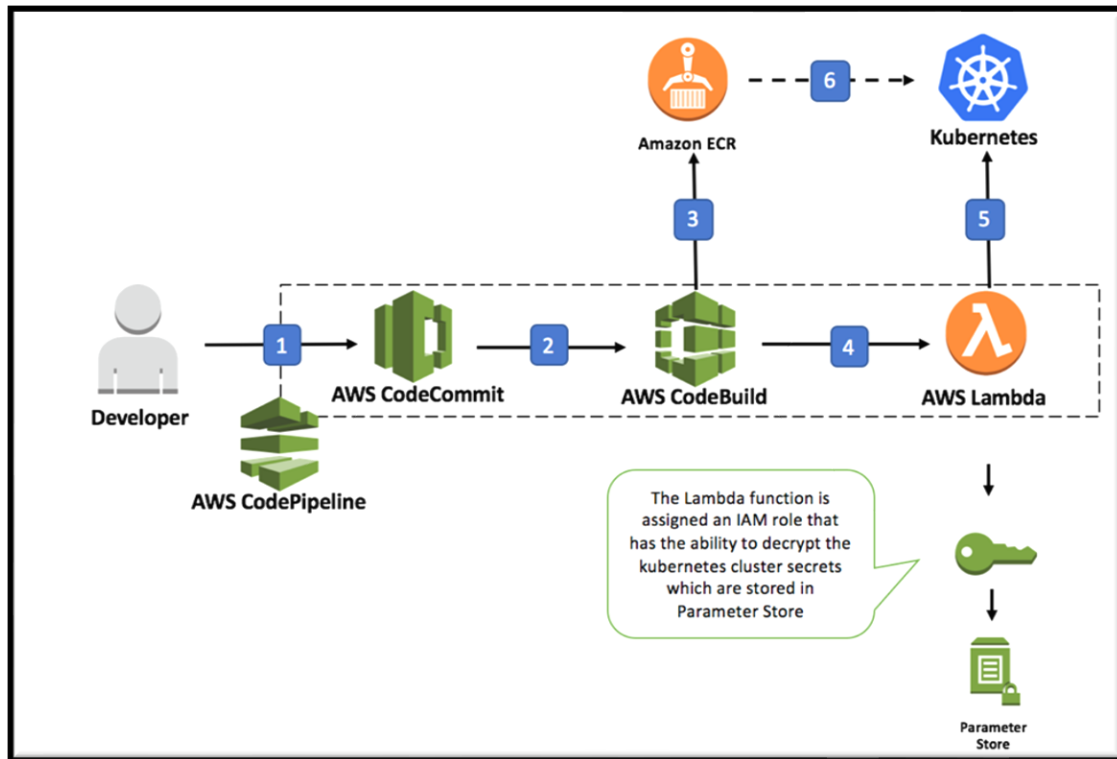
Phase 5: Authorization to Operate (ATO)

- Submit results to compliance team.
- Gain approval for production deployment.

Phase 6: Continuous Monitoring & Improvement

- Enable anomaly detection (DevOps Guru, GuardDuty).
- Run periodic recovery drills.
- Update architecture for new AWS capabilities.
- Guru acts like a Control Gate for developers and a continuous monitoring tool to identify weaknesses that must be addressed prior to production acceptance.

Amazon Guru services as an Application Factory Control Gate



AWS Functions for Early Detection & Mitigation

Once applications are in production, AWS offers a suite of tools for early detection and rapid response to potential disasters:

- **Amazon CloudWatch**: Monitor performance metrics and set alarms for anomaly detection.
- **AWS DevOps Guru**: AI-powered insights for operational issues.
- **AWS GuardDuty**: Threat detection for malicious activity and unauthorized access.
- **AWS Config**: Continuous compliance auditing and remediation.
- **AWS Lambda**: Automated remediation actions triggered by events.
- **Amazon Inspector**: Automated vulnerability scanning.
- **AWS Security Hub**: Centralized view of security alerts and compliance status.

Disaster Recovery Plan

Architecting to withstand failures

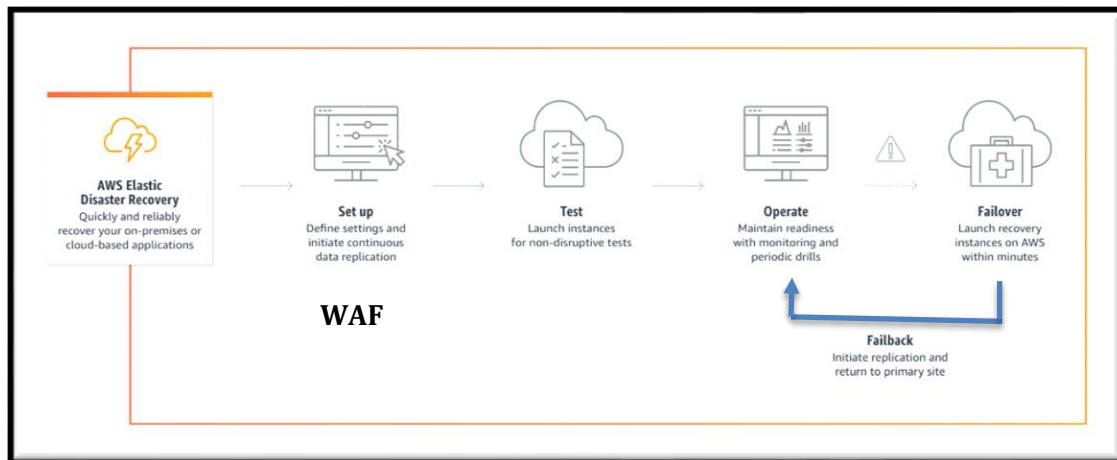
It is recommended to have a Business Continuity Plan detailing which workloads need to resist the fall of an entire region and which one must withstand the fall of an availability zone (most of the applications will be here). Depending on the Recovery Time Objective and the Recovery Point Objective, different techniques will apply.

A common architectural pattern to balance cost, availability, and recovering from different kind of incidents, when the workload requires to withstand the fall of a region, is to have High Availability in multiple Availability Zones, and Disaster Recovery configured to maintain a copy of the data with continuous replication to a different region, and a CloudFormation template to create on demand the infrastructure required to work with the data.

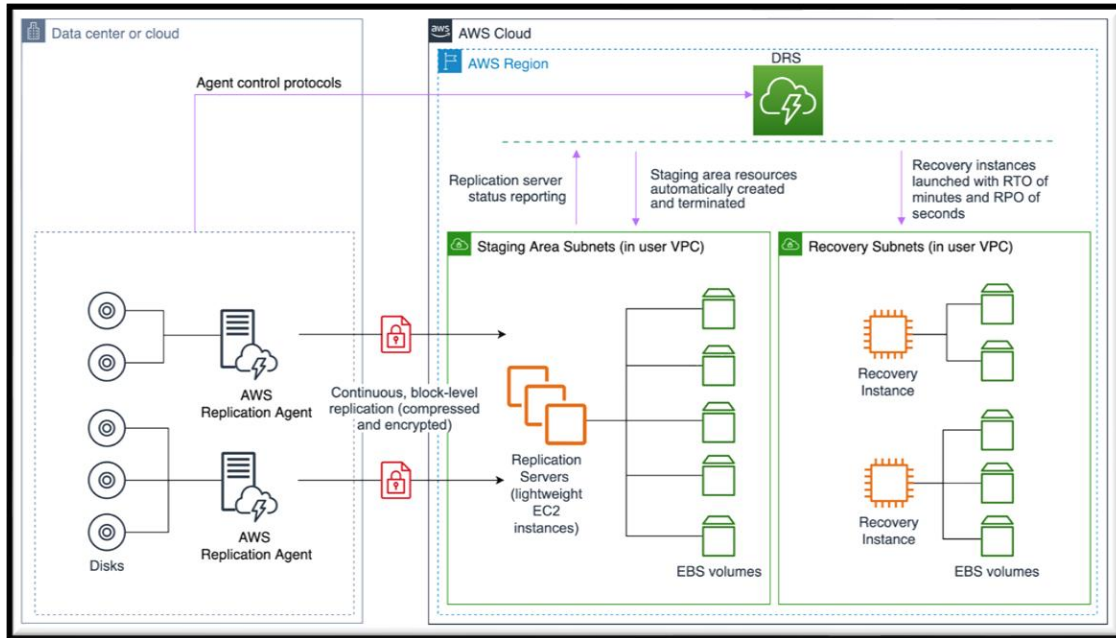
Keep in mind that Disaster Recovery has an advantage over High Availability in the fact that allows to recover to a specific point in time (such as “before the compromise” or “before the ransomware started spreading”)

You can leverage the service [AWS Elastic Disaster Recovery](#) to set up, test and operate with disaster recovery scenarios:

AWS Elastic DR Services



AWS Elastic Disaster Recovery to operate disaster recovery scenarios.



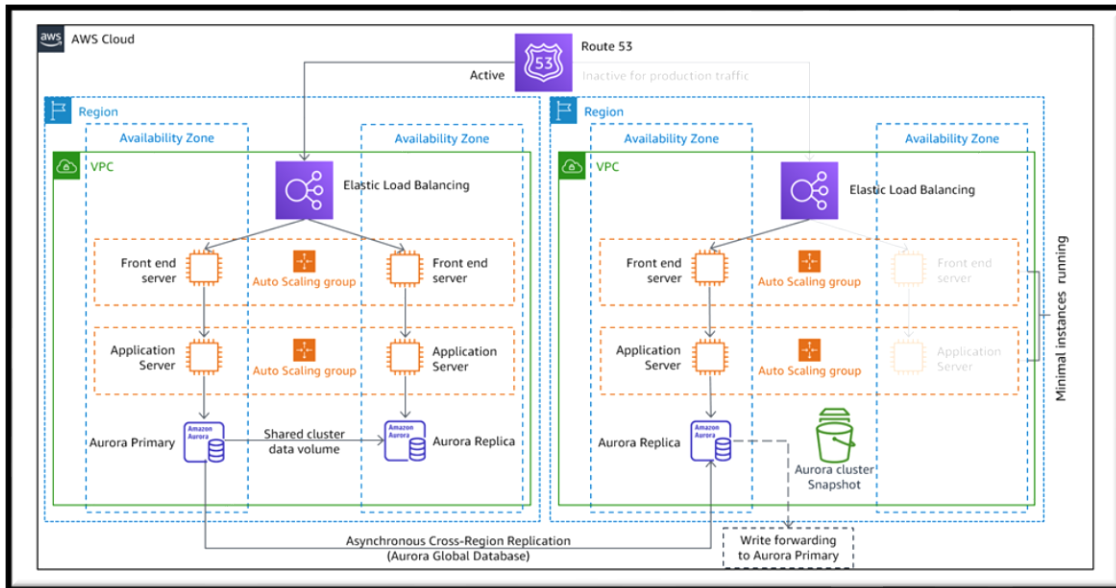
AWS Security and Governance features

Security governance	Assign Security contacts	Select the region(s) to use and block the rest
Security assurance	Evaluate Cloud Security Posture (CSPM)	
Identity and access management	Multi-Factor Authentication	Root Account Protection
	Identity Federation	Cleanup unintended accesses
Threat detection	Detect Common Threats	Audit API calls
		Billing alarms
Vulnerability management		
Infrastructure protection	Cleanup risky open ports	
Data protection	Block Public Access	Analyze data security posture
Application security	WAF with managed rules	
Incident response	Act on Critical Security Findings	
Resiliency	Evaluate Resilience	

Warm standby

The *warm standby* approach involves ensuring that there is a scaled down, but fully functional, copy of your production environment in another Region. This approach extends the pilot light concept and decreases the time to recover because your workload is always-on in another Region. This approach also allows you to perform testing more easily or implement continuous testing to increase confidence in your ability to recover from a disaster.

AWS Warm Standby approach for disaster recovery operations



Note:

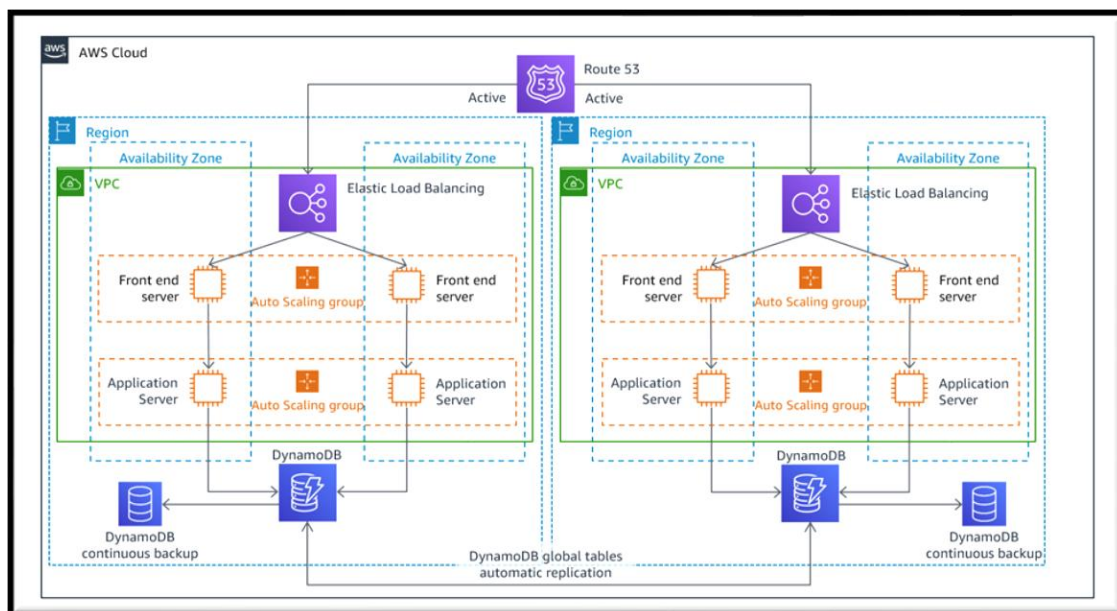
The difference between [pilot light](#) and warm standby can sometimes be difficult to understand. Both include an environment in your DR Region with copies of your primary Region assets. The distinction is that pilot light cannot process requests without additional action taken first, whereas warm standby can manage traffic (at reduced capacity levels) immediately. The pilot light approach requires you to “turn on” servers, deploy additional (non-core) infrastructure, and scale up, whereas warm standby only requires you to scale up (everything is already deployed and running). Use your RTO and RPO needs to help you choose between these approaches.

Multi-site active/active or Hot Standby active/passive

You can run your workload simultaneously in multiple Regions as part of a **multi-site active/active or hot standby active/passive strategy**. Multi-site active/active serves traffic from all regions to which it is deployed, whereas hot standby serves traffic only from a single region, and the other Region(s) are only used for disaster recovery. With a multi-site active/active approach, users can access your workload in any of the Regions in which

it is deployed. This approach is the most complex and costly approach to disaster recovery, but it can reduce your recovery time to near zero for most disasters with the correct choices and implementation (however data corruption may need to rely on backups, which usually results in a non-zero recovery point). Hot standby uses an active/passive configuration where users are only directed to a single region and DR regions do not take traffic. Most customers find that if they are going to stand up a full environment in the second Region, it makes sense to use it active/active. Alternatively, if you do not want to use both Regions to manage user traffic, then Warm Standby offers a more economical and operationally less complex approach.

AWS Active/Active Recovery approach with Multi-Regions



With multi-site active/active, because the workload is running in more than one Region, there is no such thing as failover in this scenario. Disaster recovery testing in this case would focus on how the workload reacts to loss of a Region: Is traffic routed away from the failed Region? Can the other Region(s) manage all the traffic? Testing for a data disaster is also required. Backup and recovery are still required and should be evaluated regularly. It should also be noted that recovery times for a data disaster involving data corruption, deletion, or obfuscation will always be greater than zero and the **recovery point** will always be at some point before the disaster was discovered. If the additional complexity and cost of a multi-site active/active (or hot standby) approach is required to maintain near zero recovery times, then additional efforts should be made to maintain security and to prevent human error to mitigate against human disasters.

AWS services with point-in-time backup

All of the AWS services covered under [backup and restore](#), [pilot light](#), and [warm standby](#) also are used here for **point-in-time data backup**, data replication, active/active traffic routing, and deployment and scaling of infrastructure including EC2 instances.

For the active/passive scenarios discussed earlier (Pilot Light and Warm Standby), both Amazon Route 53 and AWS Global Accelerator can be used for route network traffic to the active region. For the active/active strategy here, both services also enable the definition of policies that determine which users go to which active regional endpoint. With AWS Global Accelerator you set a [traffic dial to control the percentage of traffic](#) that is directed to each application endpoint. Amazon Route 53 supports this percentage approach, and also [multiple other available policies](#) including geo-proximity and latency based ones. [Global Accelerator automatically leverages the extensive network of AWS edge servers](#), to onboard traffic to the AWS network backbone as soon as possible, resulting in lower request latencies.

Asynchronous data replication with this strategy enables near-zero RPO. AWS services like [Amazon Aurora global database](#) use dedicated infrastructure that leaves your databases entirely available to serve your application, and can replicate to up to five secondary Regions with typical latency of under a second. With active/passive strategies, writes occur only to the primary Region. The difference with active/active is designing how data consistency with writes to each active Region are managed. It is common to design user reads to be served from the Region closest to them, known as *read local*. With writes, you have several options:

A **write global strategy** routes all writes to a single Region. In case of failure of that Region, another Region would be promoted to accept writes. [Aurora global database](#) is a good fit for *write global*, as it supports synchronization with read replicas across Regions, and you can promote one of the secondary Regions to take read/write responsibilities in less than one minute. Aurora also supports write forwarding, which lets secondary clusters in Aurora global databases forward SQL statements that perform write operations to the primary cluster.

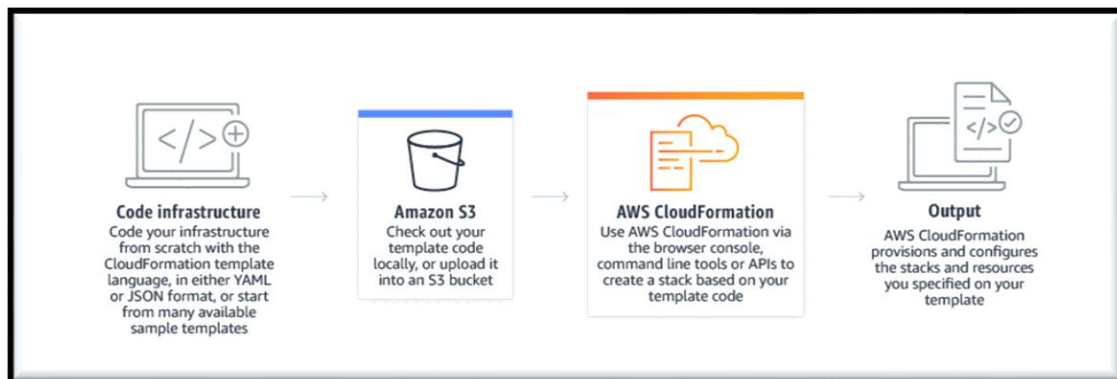
Write local strategy routes writes to the closest Region (just like reads). [Amazon DynamoDB global tables](#) enables such a strategy, allowing read and writes from every region your global table is deployed too. Amazon DynamoDB global tables use a *last writer wins* reconciliation between concurrent updates.

A **write partitioned strategy** assigns writes to a specific Region based on a partition key (like user ID) to avoid write conflicts. Amazon S3 replication [configured bi-directionally](#) can be used for this case, and currently supports replication between two Regions. When implementing this approach, make sure to enable [replica modification sync](#) on both buckets A and B to replicate replica metadata changes like object access control lists (ACLs), object tags, or object locks on the replicated objects. You can also configure whether or not to [replicate delete markers](#) between buckets in your active Regions. In addition to

replication, your strategy must also include point-in-time backups to protect against data corruption or destruction events.

AWS CloudFormation is a powerful tool to enforce consistently deployed infrastructure among AWS accounts in multiple AWS Regions. [AWS CloudFormation StackSets](#) extends this functionality by enabling you to create, update, or delete CloudFormation stacks across multiple accounts and Regions with a single operation. Although AWS CloudFormation uses YAML or JSON to define Infrastructure as Code, [AWS Cloud Development Kit \(AWS CDK\)](#) allows you to define Infrastructure as Code using familiar programming languages. Your code is converted to CloudFormation which is then used to deploy resources in AWS.

AWS Cloud Formation



Detecting a disaster event and enacting a recovery plan

AWS services

All of the AWS services covered under [backup and restore](#), [pilot light](#), and [warm standby](#) also are used here for point-in-time data backup, data replication, active/active traffic routing, and deployment and scaling of infrastructure including EC2 instances.

For the active/passive scenarios discussed earlier (Pilot Light and Warm Standby), both Amazon Route 53 and AWS Global Accelerator can be used to route network traffic to the active region. For the active/active strategy here, both services also enable the definition of policies that determine which users go to which active regional endpoint. With AWS Global Accelerator you set a [traffic dial to control the percentage of traffic](#) that is directed to each application endpoint. Amazon Route 53 supports this percentage approach, and also [multiple other available policies](#) including geo-proximity and latency based ones. [Global Accelerator automatically leverages the extensive network of AWS edge servers](#), to onboard traffic to the AWS network backbone as soon as possible, resulting in lower request latencies.

Asynchronous data replication with this strategy enables near-zero RPO. AWS services like [Amazon Aurora global database](#) use dedicated infrastructure that leaves your databases entirely available to serve your application, and can replicate to up to five secondary Regions with typical latency of under a second. With active/passive strategies, writes occur only to the primary Region. The difference with active/active is designing how data consistency with writes to each active Region are managed. It is common to design user reads to be served from the Region closest to them, known as *read local*. With writes, you have several options:

A **write global strategy** routes all writes to a single Region. In case of failure of that Region, another Region would be promoted to accept writes. [Aurora global database](#) is a good fit for *write global*, as it supports synchronization with read replicas across Regions, and you can promote one of the secondary Regions to take read/write responsibilities in less than one minute. Aurora enables secondary clusters in a global database to forward write SQL statements to the primary cluster.

Write *local* strategy routes writes to the closest Region (just like reads). [Amazon DynamoDB global tables](#) enables such a strategy, allowing read and writes from every region your global table is deployed too. Amazon DynamoDB global tables use a *last writer wins* reconciliation between concurrent updates.

A **write partitioned strategy** assigns writes to a specific Region based on a partition key (like user ID) to avoid write conflicts. Amazon S3 replication [configured bi-directionally](#) can be used for this case, and currently supports replication between two Regions. When implementing this approach, make sure to enable [replica modification sync](#) on both buckets A and B to replicate replica metadata changes like object access control lists (ACLs), object tags, or object locks on the replicated objects. You can also configure whether or not to [replicate delete markers](#) between buckets in your active Regions. In addition to replication, your strategy must also include point-in-time backups to protect against data corruption or destruction events.

AWS CloudFormation is a powerful tool to enforce consistently deployed infrastructure among AWS accounts in multiple AWS Regions. [AWS CloudFormation StackSets](#) extends this functionality by enabling you to create, update, or delete CloudFormation stacks across multiple accounts and Regions with a single operation. Although AWS CloudFormation uses YAML or JSON to define Infrastructure as Code, [AWS Cloud Development Kit \(AWS CDK\)](#) allows you to define Infrastructure as Code using familiar programming languages. Your

code is converted to CloudFormation which is then used to deploy resources in AWS.

Testing the Disaster Recovery Plan

Evaluate disaster recovery implementation to validate the implementation and regularly assess failover to your workload's DR Region to ensure that RTO and RPO are met.

A pattern to avoid is developing recovery paths that are rarely executed. For example, you might have a secondary data store that is used for read-only queries. When you write to a data store and the primary fails, you might want to fail over to the secondary data store. If you do not frequently evaluate this failover, you might find that your assumptions about the capabilities of the secondary data store are incorrect. The capacity of the secondary, which might have been sufficient when you last evaluated, might no longer be able to tolerate the load under this scenario, or service quotas in the secondary Region might not be sufficient.

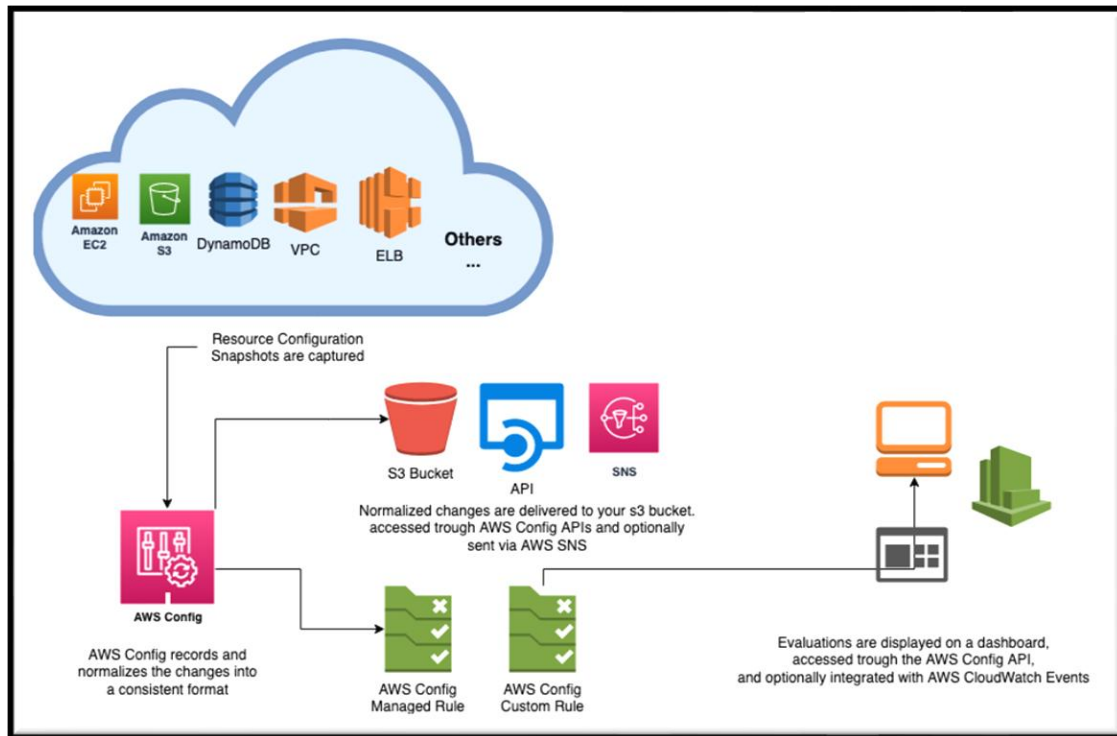
Our experience has shown that the only error recovery that works is the path you assess frequently. This is the reason having frequent recovery paths is best.

You can **establish recovery patterns** and regularly assess them. If you have a complex or critical recovery path, you still need to regularly execute that failure in production to validate that the recovery path works.

Manage configuration drift at the DR Region. Ensure that your infrastructure, data, and configuration are as needed at the DR Region. For example, check that AMIs and service quotas are up to date.

You can utilize [AWS Config](#) to continuously monitor and record your AWS resource configurations. AWS Config can detect drift and trigger [AWS Systems Manager Automation](#) to fix drift and raise alarms. [AWS CloudFormation](#) can additionally detect drift in stacks you have deployed.

AWS Config



Conclusion

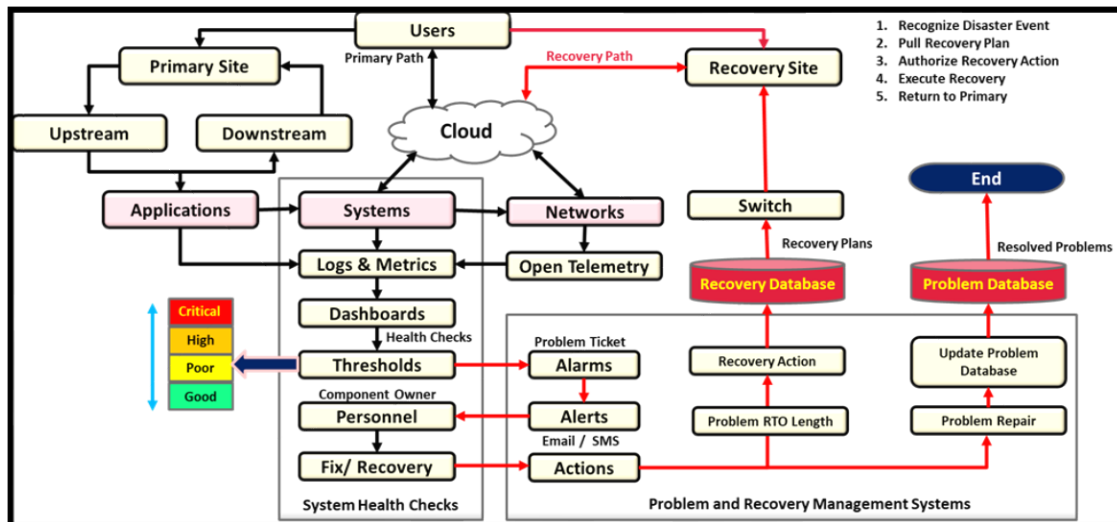
Customers are responsible for the availability of their applications in the cloud. It is important to define what a disaster is and to have a disaster recovery plan that reflects this definition and the impact that it may have on business outcomes. Create Recovery Time Objective (RTO) and Recovery Point Objective (RPO) based on impact analysis and risk assessments and then choose the appropriate architecture to mitigate against disasters. Ensure that detection of disasters is possible and timely — it is vital to know when objectives are at risk. Ensure you have a plan and validate the plan with testing. Disaster recovery plans that have not been validated risk not being implemented due to a lack of confidence or failure to meet disaster recovery objectives.

Recognizing Disaster Events and Activating Recovery Plans

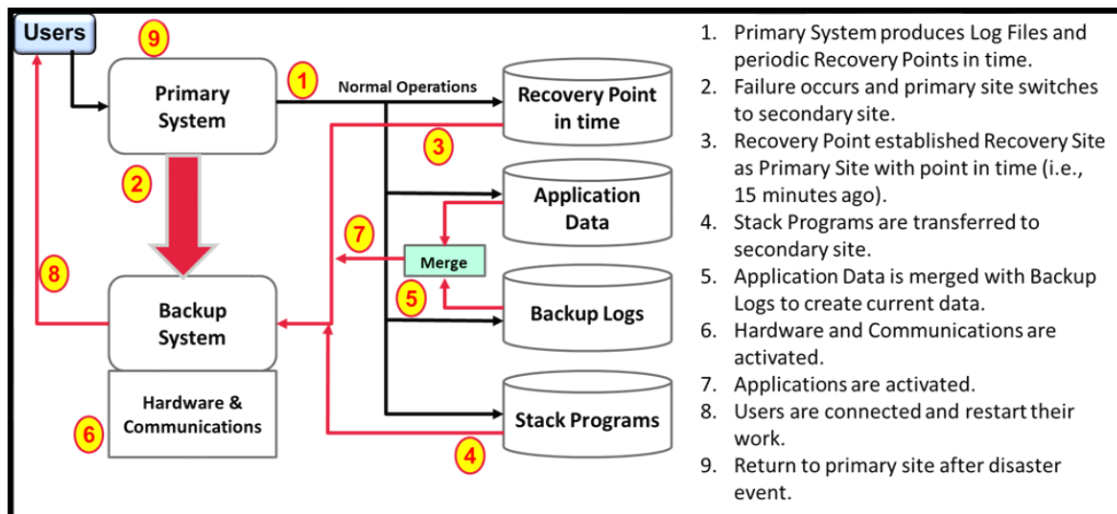
Activating a disaster recovery is costly and can sometimes be a false activation. It is important to train staff on how to recognize a disaster event and activate a recovery plan. Because of the cost of recoveries, a manager is usually required to declare a disaster and activate the disaster event. Recovery Plans should be clearly names so that the correct recovery plan can be selected for activated for a specific recovery event (Natural Disaster, System Disaster, etc.). Disaster events are often related to problems and incidents, so it stands to reason that an interface should be established between the problem/incident management system and the disaster recovery management system. If the failing

component owner decides that the time to resolve a problem will be longer than the RTO, then a recovery plan should be immediately enacted.

Problem/Incident Management and Disaster Recovery Management relationship



Recovery Process with Recovery Point



This final illustration demonstrates the steps needed to recover a site that has suffered from a malware virus, or ransomware attack. Use the Recovery Point to return to a time prior to the infection, then switch to the secondary site, recover data, connect users, and continue operations in the sequence shown.

Call to Action

If you want to discuss the topics included on this paper, please contact me at:

bronackt@dcag.com } bronackt@gmail.com

(917) 673-6992

Website: <https://www.dcag.com>