aws certified

Solutions Architect

Professional

DOLFINed

DESIGNING SOLUTIONS FOR HIGH AVAILABILITY AND BUSINESS CONTINUITY

ArchitectingOn AWS

Considerations/Pillars

Architecture Pillars and Principles

- Business Continuity and Disaster Recovery
  - Resiliency and fault Tolerance
  - Redundancy, and High Availability.
- Cost
- Performance
- Security
- Monitoring
- Scalability and Elasticity
- Ease of Deployment
- Migration and Hybrid architectures

Building Fault Tolerant Apps on AWS

Amazon Machine Image (AMI)

Fault Tolerance on AWS

Building Fault Tolerant Applications on AWS - AMI

Amazon Machine Image (AMI)

Is basically a software configuration that is applied to the EC2 instance and it includes: operating system, application server, and applications

- Creating a library of your own AMIs, is your first step towards building fault-tolerant applications on AWS.
- Your application should be comprised of at least one AMI that you have created.
  - Starting your application then is simply a matter of launching the AMI.
  - By doing this, you are able to quickly recover from failures;
    - If an instance fails, or is not behaving the way you want it to, you can simply launch another one based on the same template.

Fault Tolerance on AWS

Building Fault Tolerant Applications on AWS - AMI

Amazon Machine Image (AMI)

To minimize downtime, you might even always keep a spare instance running – ready to take over in the event of a failure

- This can be done efficiently using *elastic IP addresses*.
- You can easily fail over to a replacement instance or spare running instance by remapping your elastic IP address to the new instance **(Floating IP)**

- Being able to quickly launch replacement instances based on an AMI that you define is a critical first step towards fault tolerance.

Building Fault Tolerant Apps on AWS

Elastic Block Store (EBS)

Fault Tolerance on AWS

Building Fault Tolerant Applications on AWS – Storing Persistent Data

- Elastic Block Store (EBS)
  - Amazon EBS is especially suited for applications that require a database, a file system, or access to raw block level storage.
- Amazon EBS volumes are highly reliable, but to further mitigate the possibility of a failure, backups of these volumes can be created using a feature called ***snapshots*.**
- Snapshots can be used to create new Amazon EBS volumes, which are an exact replica of the original volume at the time the snapshot was taken.
- Because backups represent the on-disk state of the application, care must be taken to flush in-memory data to disk before initiating a snapshot.

Fault Tolerance on AWS

Building Fault Tolerant Applications on AWS – Storing Persistent Data

- EBS's fault tolerance:
  - EBS volumes store data redundantly, making them more durable than a typical hard drive.
  - A robust backup strategy will include an interval (time between backups, generally daily but perhaps more frequently for certain applications),
    - A retention period (dependent on the application and the business requirements for rollback), and a recovery plan.
  - Snapshots are stored for high-durability in Amazon S3.
  - EBS snapshots can be copied into other regions to create EBS volumes

Building Fault Tolerant Apps on AWS

Auto Scaling & ELB

Fault Tolerance on AWS

Building Fault Tolerant Applications on AWS – Auto Scaling

- Auto Scaling enables you to automatically scale your Amazon EC2 capacity up or down.
- Using AS in conjunction with Cloud Watch to control termination and launching new EC2 instances
- Since Auto Scaling will automatically detect failures and launch replacement instances,
  - If an instance is not behaving as expected (e.g., it is running with poor performance), you can simply terminate that instance and a new one will be launched.
- By using Auto Scaling, you can (and should) regularly turn your instances over to ensure that any leaks or degradation do not impact your application –
  - You can literally set expiry dates on your server instances to ensure they remain 'fresh.'

Fault Tolerance on AWS

Building Fault Tolerant Applications on AWS - ELB

- Elastic Load Balancing is an AWS product that distributes incoming traffic to your application across several Amazon EC2 instances.

- When you use Elastic Load Balancing, you are given a DNS host name – any requests sent to this host name are delegated to a pool of Amazon EC2 instances.

- Elastic Load balancing is bound to a region

- Elastic Load Balancing detects unhealthy instances within its pool of Amazon EC2 instances and automatically reroutes traffic to healthy instances, until the unhealthy instances have been restored.

Building Fault Tolerant Apps on AWS

Regions and Availability Zones

Fault Tolerance on AWS

Building Fault Tolerant Applications on AWS

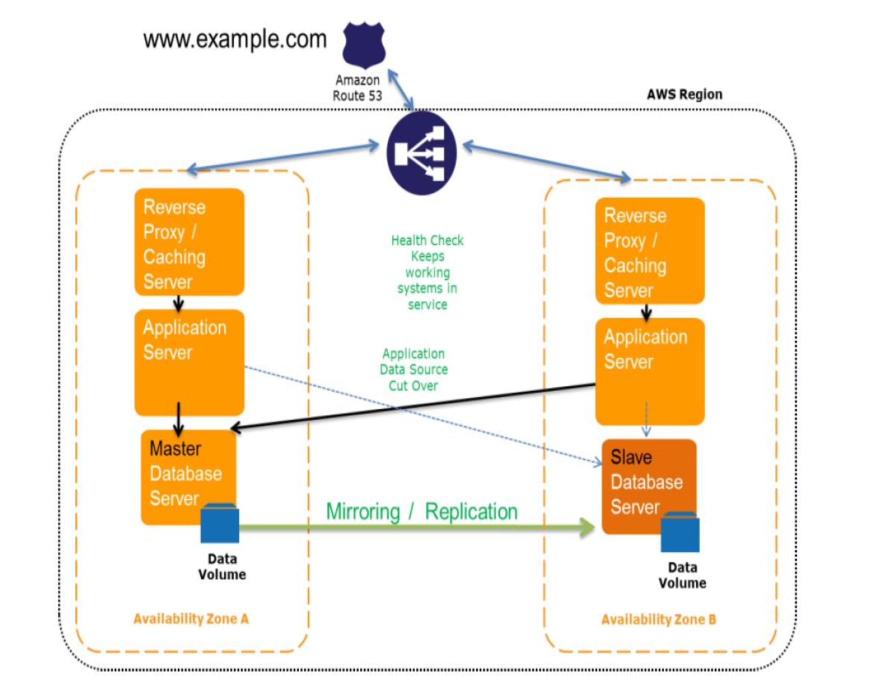
Regions and Availability Zones:

- Another key element to achieving greater fault tolerance is to distribute your application geographically.
- Amazon Web Services are available in geographic *Regions*.
- Regions consist of one or more Availability Zones, are geographically dispersed, and are in separate geographic areas or countries.
- Availability Zones are distinct locations that are engineered to be insulated from failures in other Availability Zones
- They provide inexpensive, low latency network connectivity to other Availability Zones in the same Region.
- By launching instances in separate Availability Zones, you can protect your applications from a failure (unlikely as it might be) that affects an entire zone.

Fault Tolerance on AWS

Building Fault Tolerant Applications on AWS – Multi AZ Architectures

- Within an AWS region,
  - The desired goal is to have an independent copy of each application stack in two or more Availability Zones.
- Use redundant instances for each tier (e.g. web, application, and database) of an application could be placed in distinct Availability Zones thereby creating a multi-site solution.
- To achieve even more fault tolerance with less manual intervention, you can use Elastic Load Balancing.
- Auto Scaling can work across multiple Availability Zones in an AWS Region,
  - This makes it easier to automate increasing and decreasing of capacity.



Building Fault Tolerant Applications on AWS

- Elastic Load Balancing can detect the health of Amazon EC2 instances.
  - When it detects unhealthy Amazon EC2 instances, it no longer routes traffic to those unhealthy instances.
  - Instead, it spreads the load across the remaining healthy instances.
  - If all of your Amazon EC2 instances in a particular Availability Zone are unhealthy, but you have set up instances in multiple Availability Zones, Elastic Load Balancing will route traffic to your healthy Amazon EC2 instances in those other zones.
- This multi-site solution is highly available, and by design will cope with individual component or even Availability Zone failures.
- You can also use Route53 to build fault tolerance applications across AWS Regions for higher availability and Disaster Recovery

Building Fault Tolerant Apps on AWS

Simple Storage Service

Building Fault Tolerant Applications on AWS

Amazon S3:

- Amazon Simple Storage Service (Amazon S3) is a deceptively simple web service that provides highly durable, fault-tolerant data/object storage.
- Amazon Web Services is responsible for maintaining availability and fault-tolerance; you simply pay for the storage that you use.
- Behind the scenes, Amazon S3 stores objects redundantly on multiple devices across multiple facilities in an Amazon S3 Region –
  - This caters for the case of a failure in an Amazon Web Service data center, where data will still be accessible.
- Amazon S3 is ideal for any kind of object data storage requirements that your application might have.

Fault Tolerance on AWS

Building Fault Tolerant Applications on AWS

- Amazon S3's Versioning feature allows you to retain prior versions of objects stored in S3
  - Versioning also protects against accidental deletions initiated by a misbehaving application. Versioning can be enabled for any of your S3 buckets.

- By using Amazon S3, you can delegate the responsibility of one critical aspect of fault-tolerance – data storage – to Amazon Web Services.

Building Fault Tolerant Apps on AWS

Relational DB Service

Building Fault Tolerant Applications on AWS

RDS

- In the context of building fault-tolerant and highly available applications, Amazon RDS offers several features to enhance the reliability of critical databases.
  - Automated backups:
    - Of your database enable point-in-time recovery for your database instance.
    - Amazon RDS will back up your database and transaction logs and store both for a user-specified retention period. This feature is enabled by default.
  - Manual Snapshots:
    - You can initiate snapshots of your DB Instance.
    - These full database backups will be stored by Amazon RDS until you explicitly delete them.
      - You can create a new DB Instance from a DB Snapshot whenever you desire.

Fault Tolerance on AWS

Building Fault Tolerant Applications on AWS

RDS

- Amazon RDS also supports a *Multi-AZ* deployment feature.
  - If this is enabled, a synchronous standby replica of your database is provisioned in a different Availability Zone.

- Updates to your DB Instance are **synchronously replicated across Availability Zones** to the standby in order to keep both databases in sync.

- In case of a failover scenario, the standby is promoted to be the primary and will handle your database operations.

- Running your DB Instance as a Multi-AZ deployment safeguards your data in the unlikely event of a DB Instance component failure or service health disruption in one Availability Zone.

Using AWS for Disaster Recovery (DR)

Part 1

Using AWS for DR

- Any event that has a negative impact on a company’s business continuity or finances could be termed a disaster. This includes:
  - Hardware or software failure, A network outage, A power outage,
  - Physical damage to a building like fire, earthquakes, hurricanes, or flooding,
  - Human error

- **Disaster recovery (DR)** is all about preparing for and recovering from a disaster.

- Question to be answered:
  - What are the best practices to improve the DR processes, from minimal investments to full-scale availability and fault tolerance, and
  - How to use AWS services to reduce cost and ensure business continuity during a DR event.

Using AWS for DR – RTO and RPO

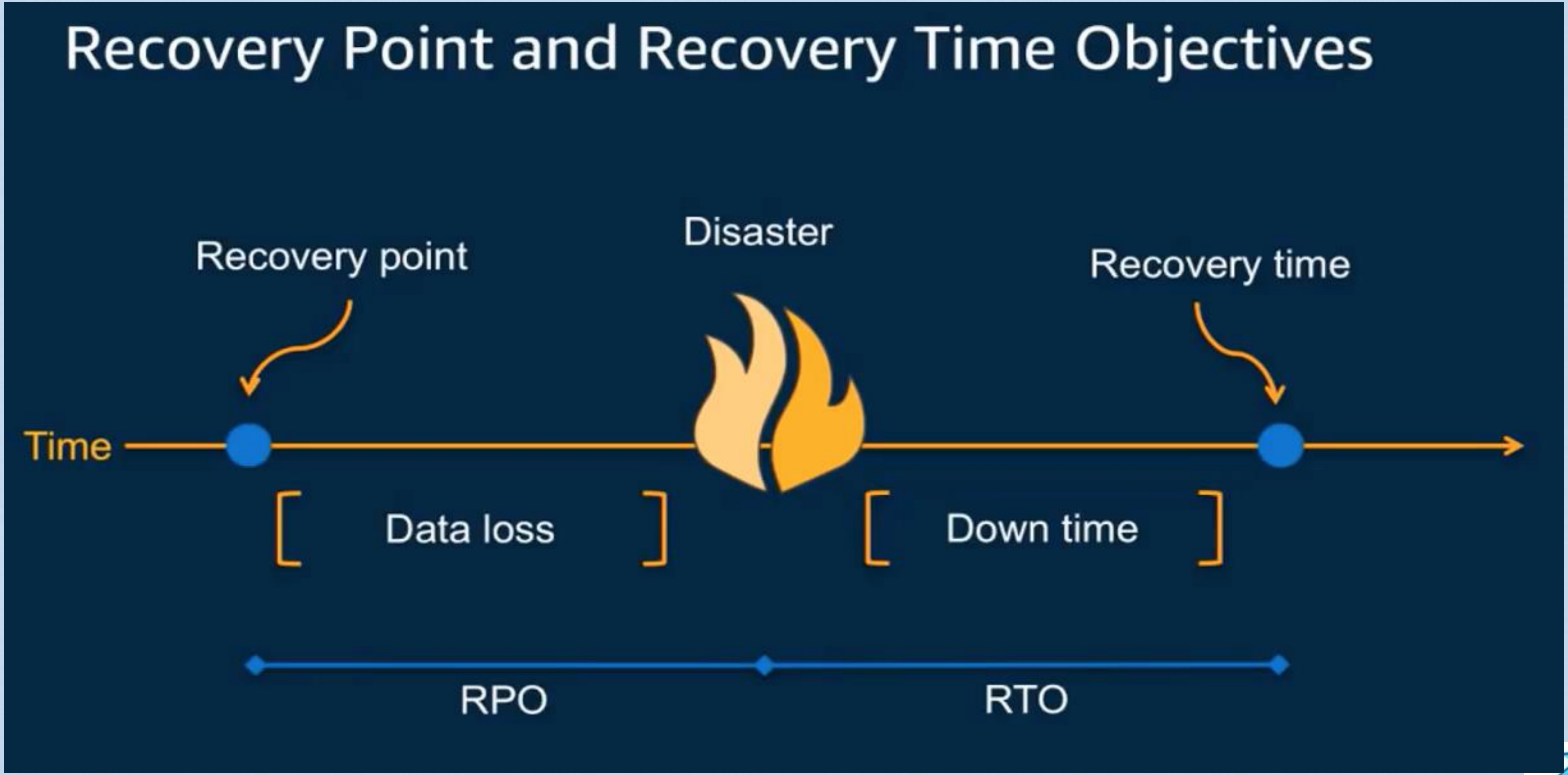
- **Recovery time objective** (RTO) —
  - The time it takes after a disruption to restore a business process to its service level, as defined by the operational level agreement (OLA).
  - For example, if a disaster occurs at 12:00 PM (noon) and the RTO is eight hours, the DR process should restore the business process to the acceptable service level by 8:00 PM.

- **Recovery point objective** (RPO) —
  - The acceptable amount of data loss measured in time.
  - For example, if a disaster occurs at 12:00 PM (noon) and the RPO is one hour, the system should recover all data that was in the system before 11:00 AM.
    - Data loss will span only one hour, between 11:00 AM and 12:00 PM (noon).

- A company typically decides on an acceptable RTO and RPO based on the financial impact to the business when systems are unavailable.

Disaster Recovery on AWS

Using AWS for DR – RTO and RPO



Disaster Recovery on AWS

Using AWS for DR – AWS Services for Backup and DR

- **Elastic Compute Cloud (Amazon EC2)**
  - Within minutes, you can create Amazon EC2 instances, which are virtual machines over which you have complete control.

- In the context of DR, the ability to rapidly create virtual machines that you can control is critical.

- Amazon Machine Images (AMIs) are preconfigured with operating systems, and some preconfigured AMIs might also include application stacks.

- In the context of DR, AWS strongly recommends that you configure and identify your own AMIs so that they can launch as part of your recovery procedure.
  - Such AMIs should be preconfigured with your operating system of choice plus appropriate pieces of the application stack.
  - You can copy your AMIs to other regions for DR purposes

Using AWS for Disaster Recovery (DR)

Part 2

Disaster Recovery on AWS

Using AWS for DR – AWS Services for Backup and DR

- **Storage**
  - **Simple Storage Service (S3) :**
    - Provides a highly durable storage infrastructure designed for mission-critical and primary data storage.
    - Objects are redundantly stored on multiple devices across multiple facilities within a region, designed to provide a durability of 99.999999999%.
    - AWS provides further protection for data retention and archiving through versioning in Amazon S3, AWS multi-factor authentication (AWS MFA), bucket policies, and AWS IAM
  - **Glacier:**
    - Provides extremely low-cost storage for data archiving and backup. Objects (or archives, as they are known in Amazon Glacier) are optimized for infrequent access, for which retrieval times of several hours are adequate.
    - Amazon Glacier is designed for the same durability as Amazon S3.

Using AWS for DR – AWS Services for Backup and DR

- **AWS Elastic Block Store (EBS):**
  - Provides the ability to create point-in-time snapshots of data volumes.
  - You can use the snapshots as the starting point for new Amazon EBS volumes.
  - You can protect your data for long-term durability because snapshots are stored within Amazon S3.
  - Amazon EBS volumes provide off-instance storage that persists independently from the life of an instance
    - It is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component.

Disaster Recovery on AWS

Using AWS for DR – AWS Storage Gateway

- AWS Storage Gateway supports three storage interfaces (or Storage Configurations): file gateway, volume gateway , and tape gateway.
  - Each gateway you have can provide one type of interface.

- The *volume gateway* provides block storage to your applications using the iSCSI protocol.
  - Data on the volumes is stored in Amazon S3.
    - To access your iSCSI volumes in AWS, you can take EBS snapshots which can be used to create EBS volumes.

- The *tape gateway* provides your backup application with an iSCSI virtual tape library (VTL) interface, consisting of a virtual media changer, virtual tape drives, and virtual tapes.
  - Virtual tape data is stored in Amazon S3 or can be archived to Amazon Glacier.

Using AWS for DR – Services for Backup and DR

- **Snowball :** Used to transfer Terabytes to Petabytes of data in and out of AWS
  - As a rule of thumb, if it takes more than one week to upload your data to AWS using the spare capacity of your existing Internet connection, then you should consider using Snowball.
  - Comes in three flavors, Snowball, Snowball Edge, and Snowmobile

Disaster Recovery on AWS

Using AWS for DR – AWS VM Import/Export

- VM Import/Export enables you to easily import virtual machine images from your existing environment to Amazon EC2 instances.

- You can also export the imported instances back to your on-premises virtualization infrastructure, allowing you to deploy workloads across your IT infrastructure.

- VM Import/Export is available at no additional charge beyond standard usage charges for Amazon EC2 and Amazon S3.

Using AWS for Disaster Recovery (DR)

Part 3

Using AWS for DR - Networking

- When you are dealing with a disaster, it’s very likely that you will have to modify network settings as your system is failing over to another site.

- The following AWS services and features enable you to manage and modify network settings.
  - **Amazon Route 53**
    - It gives developers and businesses a reliable, cost-effective way to route users to Internet applications.
    - Amazon Route 53 includes a number of global load-balancing capabilities (which can be effective when you are dealing with DR scenarios such as DNS endpoint health checks) and,
    - The ability to failover between multiple endpoints and even static websites hosted in Amazon S3.

Disaster Recovery on AWS

Using AWS for DR - Networking

- **Elastic IP addresses**
  - Elastic IP addresses enable you to mask instance or Availability Zone failures by programmatically remapping your public IP addresses to instances in your account in a particular region.
  - For DR, you can also pre-allocate some IP addresses for the most critical systems so that their IP addresses are already known before disaster strikes.

Using AWS for DR - Networking

- **Elastic Load Balancing**
  - Just as you can pre-allocate Elastic IP addresses, you can pre-allocate your load balancer so that its DNS name is already known, which can simplify the execution of your DR plan.

- **Amazon Virtual Private Cloud (Amazon VPC)**
  - In the context of DR, you can use Amazon VPC to extend your existing network topology to the cloud;
  - This can be especially appropriate when recovering enterprise applications that are typically on the internal network.

- Amazon Direct Connect makes it easy to set up a dedicated network connection from your premises to AWS.
  - This can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

Using AWS for DR - Database

- **Amazon Relational Database Service (Amazon RDS)**
  - You can use Amazon RDS either in the preparation phase for DR to hold your critical data in a database that is already running, or in the recovery phase to run your production database.
  - When you want to look at multiple regions, Amazon RDS gives you the ability to snapshot data from one region to another, and also to have a read replica running in another region.
- **Amazon DynamoDB**
  - You can also use it in the preparation phase to copy data to DynamoDB in another region or to Amazon S3.
  - During the recovery phase of DR, you can scale up seamlessly in a matter of minutes with a single click or API call.
  - You can also benefit from Global Tables (Cross Region Replication)

Disaster Recovery on AWS

Using AWS for DR - Database

- **Amazon Redshift**
  - You can use Amazon Redshift in the preparation phase to snapshot your data warehouse to be durably stored in Amazon S3 within the same region or copied to another region.
  - During the recovery phase of DR, you can quickly restore your data warehouse into the same region or within another AWS region.

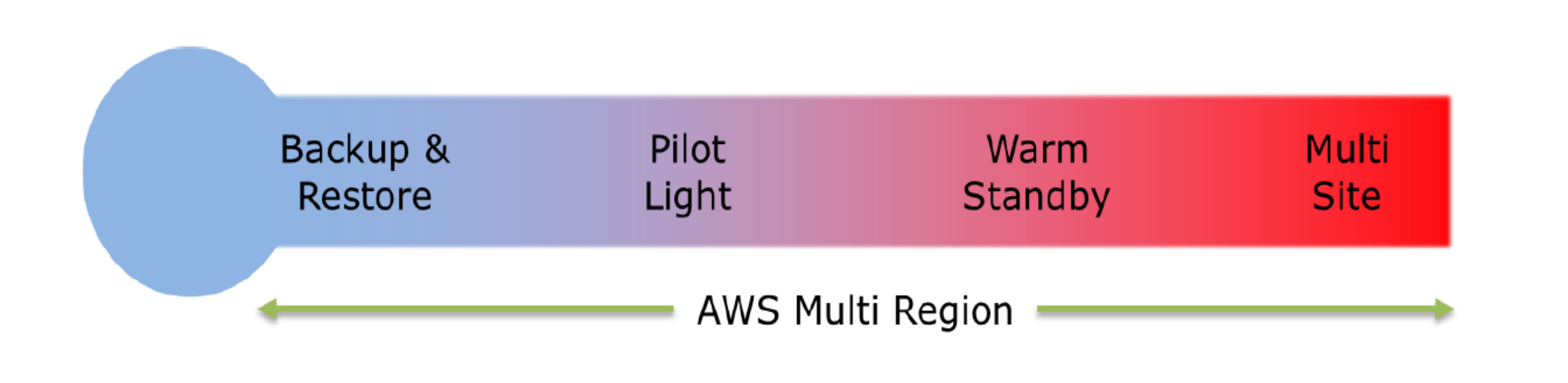
Using AWS for Disaster Recovery (DR)

DR Approaches

Disaster Recovery on AWS

Using AWS for DR – DR Approaches/Strategies

- Various approaches to DR:
  - Backup and Restore
  - Pilot Light
  - Warm standby
  - Multi Site

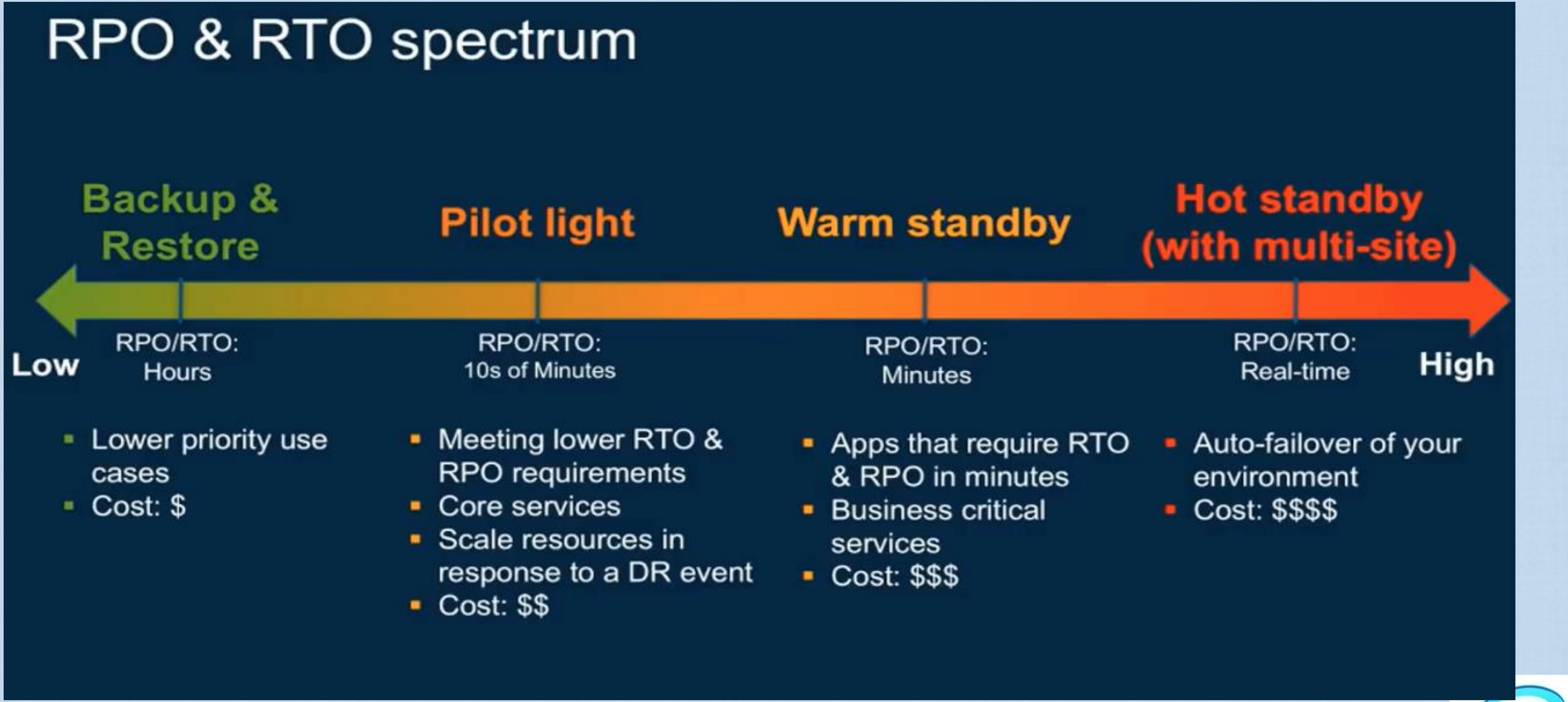


Slowest

Fastest

Disaster Recovery on AWS

Using AWS for DR – DR Approaches/Strategies



RPO & RTO spectrum

Low — High

Backup & Restore

RPO/RTO: Hours

- Lower priority use cases
- Cost: $

Pilot light

RPO/RTO: 10s of Minutes

- Meeting lower RTO & RPO requirements
- Core services
- Scale resources in response to a DR event
- Cost: $$

Warm standby

RPO/RTO: Minutes

- Apps that require RTO & RPO in minutes
- Business critical services
- Cost: $$$

Hot standby (with multi-site)

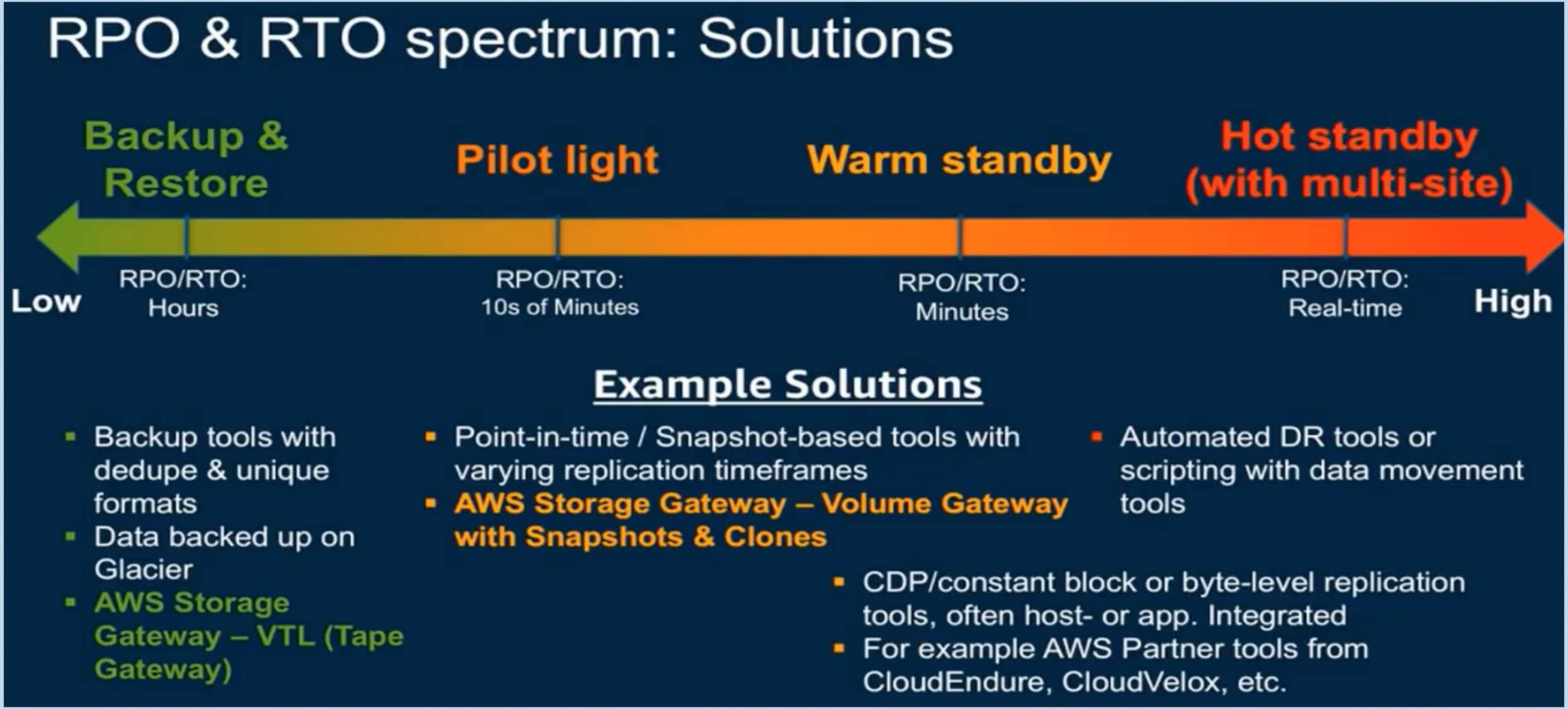
RPO/RTO: Real-time

- Auto-failover of your environment
- Cost: $$$$

Source aws.amazon.com

Disaster Recovery on AWS

Using AWS for DR – DR Approaches/Strategies



RPO & RTO spectrum: Solutions

Low

Backup & Restore
RPO/RTO:
Hours

Pilot light
RPO/RTO:
10s of Minutes

Warm standby
RPO/RTO:
Minutes

Hot standby (with multi-site)
RPO/RTO:
Real-time

High

Example Solutions

- Backup tools with dedupe & unique formats
- Data backed up on Glacier
- **AWS Storage Gateway – VTL (Tape Gateway)**

- Point-in-time / Snapshot-based tools with varying replication timeframes
- **AWS Storage Gateway – Volume Gateway with Snapshots & Clones**

- Automated DR tools or scripting with data movement tools

- CDP/constant block or byte-level replication tools, often host- or app. Integrated
- For example AWS Partner tools from CloudEndure, CloudVelox, etc.

Source aws.amazon.com

Disaster Recovery on AWS

Using AWS for DR – Backup and Restore

Key steps for backup and restore:

- Select an appropriate tool or method to back up your data into AWS.
- Ensure that you have an appropriate retention policy for this data.
- Ensure that appropriate security measures are in place for this data, including encryption and access policies.
- Regularly test the recovery of this data and the restoration of your system.

Using AWS for Disaster Recovery (DR)

- Replication Methods and Self Healing

Data Replication

- Many database systems support asynchronous data replication.

- The database replica can be located remotely, and the replica does not have to be completely synchronized with the primary database server.
  - This is acceptable in many scenarios, for example, as a backup source or reporting/read-only use cases.
  - In addition to database systems, you can also extend it to network file systems and data volumes.

Self Healing in AWS

- SQS to decouple
- CW and Auto Scaling terminating unhealthy instance
- Auto Scaling creating replacement EC2 instances to replace those terminated
- Amazon S3 also performs regular, systematic data integrity checks and is built to be automatically self-healing.
- Amazon Glacier performs regular, systematic data integrity checks and is built to be automatically self-healing.