MODULE 16 · STUDY NOTES

Planning for Disaster

Setting RPO and RTO from business needs, protecting each AWS service category across Regions, and choosing among backup & restore, pilot light, warm standby, and multi-site

A cloud architect designs to mitigate the risk of disaster and to recover quickly when one strikes — and, as with every other layer of an architecture, the right answer starts from the business, not the technology. Werner Vogels' maxim, *"Everything fails, all the time,"* frames the whole module: failures at small, large, and global scale are inevitable, so the question is how you prepare. This module builds the answer in four moves. First, it turns business impact into two numbers — the recovery point objective (RPO), the data you can afford to lose, and the recovery time objective (RTO), the downtime you can afford — and packages them, with a business impact analysis and risk assessment, into a business continuity plan (BCP). Second, it maps the AWS building blocks for protecting each service category across Regions: cross-Region replication and snapshots for storage, AMIs for compute, snapshots and global tables for databases, Route 53 and EventBridge for networking failover, and CloudFormation to redeploy whole environments. Third, it lays out the four DR patterns — backup and restore, pilot light, warm standby, and multi-site — each a different balance of cost, data loss, and recovery time. Finally, it connects the work to the AWS Well-Architected Framework: define recovery objectives, use a recovery strategy, test it, communicate with customers during outages, and keep an emergency access path open.

LEARNING OBJECTIVES

- Identify strategies for disaster planning, including recovery point objective (RPO) and recovery time objective (RTO) based on business requirements
- Identify disaster planning for Amazon Web Services (AWS) service categories
- Describe common patterns for backup and disaster recovery (DR) and how to implement them
- Use the AWS Well-Architected Framework principles when designing a disaster recovery plan

16.1 Disaster planning strategies

The starting point for designing around outages is the **customer's perspective** — which parts of the business they consider critical, nominally important, or nonessential — and the concepts apply whether the environment is mid-migration, fully cloud-based, or hybrid. Werner Vogels, VP and CTO at Amazon, has famously said *"Everything fails, all the time."* Failure is not an unlikely aberration to engineer away; it is a certainty to plan for. Architects respond in two directions at once: designing architectures that **avoid** disasters, and preparing plans to **recover** when disasters occur anyway.

TABLE 16.1 Failures occur at any scale – size and impact both matter

SCALE	WHAT HAPPENS	WATCH THE IMPACT, NOT JUST THE SIZE
Small-scale	A single server stops responding or goes offline	If that one server is the only host of customer records, it is not really a small event
Large-scale	Multiple resources are affected, perhaps across Availability Zones within a Region	An array of read-only replicas accessed once a day failing does not meet the criteria for a large event
Global-scale	Failure is widespread, affecting a large number of users and systems	Even an unlikely full-Region loss is possible — choose a DR site in another Region

THREE WAYS TO AVOID AND PLAN FOR DISASTER

- **Fault tolerance** — redundancy that lets the system withstand failure of one or more components (disks, servers, network) and keep functioning; production systems have defined uptime requirements.
- **Backup** — critical to protecting data and business continuity, but a challenge: data is generated at an exponentially growing pace while local disk density and durability are not keeping up.
- **Disaster recovery** — a set of policies and procedures for recovering vital technology infrastructure after any negative-impact event: hardware/software failure, network or power outage, physical damage (fire, flooding), or human error.

How much a business invests in disaster planning depends on the cost of a potential outage, and the design is dependent on which level of failure the business can tolerate. Four factors shape the strategy: **time dependency** (how fast must this be remedied to avoid business impact — a 9-to-5 business differs from a 24/7 one); **data loss** (how much *and what type* of data you can accept losing — 10-year-old tax records differ from certified birth records); **geographic location** (does it span multiple Regions, and do different Regions need different measures?); and **cost** (is the spend the correct level given the business impact and risk?). Developing a full plan takes time, but that should not stop you from starting: back up data stores, databases, and critical servers first, then set measures for each factor and improve incrementally.

The **recovery point objective (RPO)** is the maximum acceptable amount of data loss after an unplanned data-loss incident, expressed as an amount of time. If a disaster occurs 8 hours after the last backup, the data loss is everything generated in that window — an RPO of 8 hours. To calculate an RPO, first determine how much data loss is acceptable (from the BCP), then figure out how quickly that loss accumulates as a time measurement.

WORKED EXAMPLE – DETERMINING RPO

A genealogy company digitizing courthouse records decides that losing a maximum of **800 records** (for which it still holds the source files) is acceptable. Existing patterns show no more than **100 records are created per hour**, even at peak. So 800 ÷ 100 gives an **RPO of 8 hours**: implement a plan that backs up at least every 8 hours. If a disaster then strikes at 10 p.m., the system should be able to recover all data that was in the system before 2 p.m.

The **recovery time objective (RTO)** is the maximum acceptable amount of time after a disaster that a business process can remain out of commission — the time it takes to restore applications and recover data. If a disaster occurs at 10 p.m. and the RTO is 1 hour, the DR process should return the business process to an acceptable service level by 11 p.m. A company decides on acceptable RPO and RTO based on the **financial impact** of unavailability — lost business and reputational damage from downtime — and then IT plans cost-effective recovery solutions around those two targets.

WORKED EXAMPLE – DETERMINING RTO

A ticketing service for a local music venue determines that after **2 hours** of an outage it will begin to lose revenue from lost sales, and it wants to limit reputational risk because customers expect tickets at a set time. It sets an **RTO of 2 hours**. If a disaster occurs at 9 p.m., the system should be returned to service before 11 p.m.

KEY TERMS

Business continuity plan (BCP)
A system of prevention and recovery from potential threats to a company, consisting of a business impact analysis, a risk assessment, a disaster recovery plan, and evaluated and determined RPO and RTO. It is a living document, constantly evaluated and adjusted to the needs of the business.

DR plan is a subset of the BCP
Maintaining aggressive DR targets is pointless if the workload's objectives can't be met because of disaster impact on business elements external to the workload — an earthquake might stop you shipping products even if effective DR keeps the app running.

COMMON PITFALL

Don't confuse the two objectives. **RPO** is about *data* — how far back you must be able to recover, which sets how often you back up. **RTO** is about *time* — how long the process can be down, which sets the recovery strategy you choose. A tight RPO with a loose RTO (or vice versa) is entirely normal.

16.2 AWS disaster recovery planning

To scope DR properly, think about your use of AWS holistically. Most organizations combine services that fall into **five broad categories**, and a DR plan should **span more than one Region** across all of them. Although a full Region becoming unavailable is unlikely, it is possible — imagine a meteor strike — so AWS's many Regions let you choose a DR site in addition to your primary deployment. Your RPO and RTO guide the backup and restore plans in each category and shape the production architecture.

TABLE 16.2 The five AWS service categories a DR plan spans

SERVICE CATEGORY	EXAMPLE SERVICE	WHY IT MATTERS FOR DR
Storage	Amazon S3	Object, block, and file data must be replicated and backed up across Regions
Compute	Amazon EC2	Servers/containers must be rebuildable quickly from AMIs
Database	Amazon RDS	Databases need cross-Region snapshots, replicas, and multi-Region tables
Networking & Content Delivery	Amazon VPC	Traffic must be reroutable to the recovery site on failover
Deployment orchestration (Management & Governance)	AWS CloudFormation	Whole environments must be redeployable repeatably in minutes

Storage building blocks. Your AWS storage can combine block storage (**Amazon EBS**), file storage (**Amazon EFS**, **Amazon FSx**), and object storage (**Amazon S3**, **Amazon S3 Glacier**), and you may also connect an on-premises data center to the cloud. **AWS DataSync** moves large amounts of data online between on-premises NFS/SMB storage and Amazon S3, EFS, or FSx; it supports scripted copy jobs and scheduled transfers and can optionally use AWS Direct Connect links.

S3 CROSS-REGION REPLICATION (CRR)

- S3 buckets exist in a **specific Region**, chosen at creation. S3 provides **11 nines (99.999999999%) durability** for S3 Standard, S3 Standard-IA, S3 One Zone-IA, and Amazon S3 Glacier.
- S3 Standard, S3 Standard-IA, and S3 Glacier store objects across **a minimum of three Availability Zones** (miles apart within a Region), so they survive the loss of an entire AZ.
- For higher data security, configure **CRR**: add a replication configuration to the source bucket naming the destination bucket and an **IAM role** granting S3 permission to copy objects.
- Copied objects **retain their metadata**; the destination bucket can be a different **storage class** (e.g., S3 Standard replicated to S3 Glacier) and a different owner.
- **S3 Replication Time Control (S3 RTC)** replicates 99.99% (4 nines) of new objects within **15 minutes**, backed by an SLA.

EBS VOLUME SNAPSHOTS

- Snapshots are **incremental** point-in-time backups to S3 — they save only the blocks changed since the most recent snapshot, minimizing time and storage cost.
- Each snapshot holds everything needed to restore to a **new EBS volume** (an exact replica that loads lazily, usable immediately). Once a snapshot completes you can **copy it to another Region** or within the same Region.
- **Amazon Data Lifecycle Manager** automates snapshot creation, retention, and deletion to enforce a schedule, meet compliance, and cut storage costs.
- You **cannot** snapshot EC2 **instance store** volumes — create and format a new EBS volume, mount it, and copy the data over. (RDS offers DB snapshots; FSx for NetApp ONTAP offers NetApp snapshots.)

File system replication. DataSync moves data between two Amazon EFS or two FSx for Windows File Server file systems, or between on-premises storage and AWS, over Direct Connect or the internet. **FSx for Windows File Server** takes daily automatic backups (30-minute window, 7-day default retention) stored in S3. EFS and FSx replicate across AZs like most S3 classes; for multi-Region recovery, it is a best practice to **replicate EFS and FSx to a second Region using DataSync**. **AWS Backup** can manage EBS volume backups and automate EFS backups.

Recovering compute infrastructure. You can arrange **automatic recovery** of an EC2 instance when a system status check of the underlying hardware fails: the instance reboots (on new hardware if needed) but keeps its instance ID, IP addresses, EBS volume attachments, and configuration — so configure it to auto-start its services on initialization. **Amazon Machine Images (AMIs)** are preconfigured with an OS and sometimes an application stack; for DR, identify and maintain your own. A **golden AMI** is preconfigured with all necessary applications and services. Baking the latest code into a golden AMI means a new AMI on every code push (complicating DevOps); since configuration changes less often than code, it is usually easier to bake the AMI and use a **user data script** to pull the latest code and configuration at deploy time.

EVENTBRIDGE GLOBAL ENDPOINTS FOR REGIONAL FAILOVER

- A service disruption can interrupt event flow — producers in the primary Region cannot `PutEvents` to their event bus, and replication to the secondary Region is impacted.
- A **global endpoint** is a managed **Route 53 DNS endpoint** that routes events to the event bus in either Region depending on the **health of the primary Region**.
- The metric `IngestionToInvocationStartLatency` measures the time to invoke the first target after an event is ingested; sustained latency **over 30 seconds** may indicate a disruption and drive the failover.

TABLE 16.3 Networking services for resiliency and recovery

SERVICE	ROLE IN DISASTER RECOVERY
Route 53	DNS-based load balancing and routing; basic failover between endpoints or to a static website hosted in Amazon S3
Elastic Load Balancing (ELB)	Distributes incoming traffic across EC2 instances; pre-allocate it so its DNS name is already known, making DR implementation more straightforward
Amazon VPN	Extends an on-premises network to the cloud over a VPN connection — useful for recovering enterprise apps hosted on an internal network
AWS Direct Connect	A dedicated network connection for fast, consistent transfer between on-premises and AWS; reduces network costs and increases bandwidth

AMAZON RDS – DR FEATURES	AMAZON DYNAMODB – DR FEATURES
relational database recovery	*NoSQL recovery, scales in minutes*
▪ Save a **snapshot in a separate Region** (share a manual snapshot with up to 20 accounts) ▪ Use **read replicas** — read-only copies in the same or a different Region, updated asynchronously and promotable to standalone ▪ Use **Multi-AZ** deployments; **retain automated backups**	▪ **Back up entire tables** and create on-demand backups ▪ Use **point-in-time recovery (PITR)** to restore tables ▪ Use **global tables** for a multi-Region, multi-active database available even during a Region-level disaster

Replicating and redeploying environments. With **AWS CloudFormation** you model your entire infrastructure in a text template that becomes the single source of truth — duplicating complex production environments in minutes to a new Region or VPC, repeatably, with no manual actions. CloudFormation covers **resource configuration, not data**. **AWS Elastic Beanstalk** deploys or redeploys an application source bundle — it is for the **application, not the data**. **AWS OpsWorks** provides configuration management and automation, defining the environment as layers with automatic host replacement on instance failure; template with OpsWorks in the prep phase and combine it with CloudFormation in recovery.

COMMON PITFALL

Replication is **not** a substitute for a backup. Continuously replicated data (S3 CRR, RDS read replicas) offers no protection against **data corruption** — a corrupted source is faithfully copied to the replica. Also **version or snapshot** your data for point-in-time recovery, and back up the replicated data in the recovery site.

16.3 Disaster recovery patterns

For different combinations of RPO, RTO, and cost-effectiveness, organizations use four common DR patterns: **backup and restore**, **pilot light**, **warm standby**, and **multi-site**. Each is well-suited to a different requirement — some provide a faster RPO and RTO but cost more to maintain — and the selection depends on the business use case. These are examples; variations and combinations are possible, and if your application runs on AWS you can use multiple Regions and the same strategies still apply.

TABLE 16.4 The four DR patterns – architecture, objectives, cost

PATTERN	ARCHITECTURE	RPO / RTO	COST
Backup & restore	Back up data (e.g., to S3) and restore infrastructure and data when disaster strikes; nothing extra runs in between	Hours	Lowest
Pilot light	A minimal always-on core (typically the secondary database) kept up to date; provision the full environment around it on recovery	10s of minutes	Low
Warm standby	A scaled-down but **fully functional** stack always running; scale it up to full load on failover	Minutes	Higher
Multi-site	A **full-capacity active/active** duplicate running in a second Region at the same time as the primary	Near real-time	Highest

Backup and restore. The simplest approach mitigates data loss or corruption, a regional disaster (replicate to other Regions), or a single-AZ workload's lack of redundancy — but because data is backed up and sent offsite regularly, **restoring can take a long time**. Amazon S3 is a convenient, network-reachable destination: **DataSync or S3 Transfer Acceleration** can automate or speed the transfer, and an **S3 lifecycle configuration** moves aging backups to cheaper classes (S3 Glacier Flexible Retrieval or S3 Standard-IA). To restore, download from S3 to on-premises, or keep EC2 servers ready in a VPC in the DR Region to read the bucket and temporarily host applications.

TABLE 16.5 AWS Storage Gateway – three hybrid interfaces for backup and restore

INTERFACE	PROTOCOL(S)	BACKS DATA TO
File Gateway	NFS, SMB	Objects in Amazon S3 (accessed via an NFS mount point; local cache for active data)
Volume Gateway	iSCSI	Block volumes backed up as point-in-time EBS snapshots (accessible via Amazon EC2)
Tape Gateway	Virtual tape library (VTL)	Virtual tapes in Amazon S3, archived to Amazon S3 Glacier

Pilot light. A minimal backup version of your environment is **always running** — the analogy is a gas heater's pilot light, a small always-on flame that can ignite the whole furnace. Here the pilot light is typically the **secondary database**, always running and kept current. Recovery is faster than backup and restore because the **core is already running**: you rapidly provision a full production environment around it, starting the rest from preconfigured **AMIs** (or stopped instances) that connect to the database while **Route 53** routes traffic to the new servers. Regularly changing data is replicated to the pilot light; slower-changing data (OS, applications) is stored as AMIs. It is relatively inexpensive, but the secondary cannot handle full load and must **scale quickly**. The primary can be on-premises, another Region, or another AZ.

Warm standby. Like pilot light, but **more resources are already running** — a scaled-down yet **fully functional** environment always runs in the cloud on a minimum-sized fleet of the smallest EC2 instances, further cutting recovery time. It can double for non-production work (testing, QA, internal use). **Route 53** distributes requests between the main and backup systems; on disaster it switches to the secondary, which scales up by **adding EC2 instances to the load balancer** (or resizing to larger types). **Horizontal** scaling is preferred over **vertical** scaling. Watch **software licensing** — confirm your contracts support running the backup site.

Multi-site. A fully functional, full-capacity system runs in a second Region **at the same time** as the primary, in an **active-active** configuration. Because both sites carry full production capacity, use **weighted routing** (Route 53) to split traffic between them; on disaster, shift the DNS weighting entirely to the cloud and use **Amazon EC2 Auto Scaling** to reach full load (you may need application logic to cut over to the parallel, already-running database). Cost tracks normal-operation traffic and can be reduced with **EC2 Reserved Instances** for the always-on servers. Multi-site has the **least downtime** of all patterns but costs the most — an entire duplicate system, including its licensing.

TABLE 16.6 Implementing the four patterns – preparation and recovery

PATTERN	PREPARATION PHASE	IN CASE OF DISASTER
Backup & restore	Create backups of current systems, store them in Amazon S3, and document the restore procedure (which AMI to use/build, how to restore, how to route traffic, how to configure the deployment).	Retrieve backups from S3; start and restore the infrastructure (EC2 plus VPC, subnets, security groups — a CloudFormation stack for consistency across Regions, AMIs for the OS/packages); restore the system; route traffic (adjust DNS).
Pilot light	Configure EC2 instances to replicate servers; create and maintain AMIs of key servers where fast recovery is needed; regularly run, test, and update them; automate provisioning. Test often — if the source changed, the skeleton config might fail.	Bring up resources around the replicated core dataset; scale to handle current production traffic; switch over by adjusting DNS to the backup deployment (Route 53 failover routing between primary/secondary endpoints).
Warm standby	Like pilot light, but all components run 24/7 (not scaled for production). Conduct continuous testing — synthetic transactions verify Region readiness, or trickle a subset of production traffic to the DR site.	Immediately fail over the most critical production load and adjust DNS to AWS; with health checks configured, Route 53 failover routing sends traffic automatically. Then automatically scale to all production load.
Multi-site	Like warm standby, but configure the deployment for full scaling in and out, with servers running and ready. Choose a Route 53 policy — geolocation (route by request origin; keep data in a Region, control writes) or latency (shortest round-trip; best for performance).	One key step — immediately fail over all production load to the backup site.

Across the patterns, **cost increases and RTO decreases** from backup and restore toward multi-site: backup and restore is cheapest but slowest, while warm standby and multi-site buy a faster RTO with always-running servers. Whatever you choose, **consistently exercise** it: run **Game Day** exercises that simulate critical systems — or entire Regions — going offline, confirm that backups, snapshots, and AMIs actually restore data, **monitor your monitoring system**, and keep improving your RTO and RPO.

SAMPLE EXAM QUESTION, WORKED

A solutions architect must create a DR solution for business-critical applications. The maximum acceptable data loss is 7 minutes, and the solution must deploy a completely functional version of the applications to handle the majority of traffic immediately, then scale up to full capacity over time. Which DR pattern is the most cost-effective? The answer is **warm standby (C)**: it meets the 7-minute RPO, runs at low capacity and can scale within minutes, and costs less than multi-site. Backup and restore (A) cannot restore fast enough for business-critical apps; pilot light (B) has idle instances that cannot take the majority of traffic immediately; multi-site (D) meets the requirement but costs more than necessary.

COMMON PITFALL

Keep pilot light and warm standby distinct. In **pilot light**, only the critical **core** (the database) is always running; everything else is provisioned around it during recovery. In **warm standby**, **all** components run 24/7 but scaled down, so a fully functional stack is already handling — or ready to handle — traffic. That difference is exactly why warm standby recovers faster (and costs more).

16.4 Applying AWS Well-Architected Framework principles to disaster planning

The AWS Well-Architected Framework has **six pillars**, each with best practices and questions to consider; three are most relevant to disaster planning — **Reliability**, **Operational Excellence**, and **Security**. Building resilient workloads prepares for any event that prevents a workload from meeting its business objectives in its primary location. **Failure management** is one step toward resiliency, and planning for disaster recovery is one part of it — starting from backups and redundant components, with RTO and RPO set from business needs.

TABLE 16.7 Well-Architected best-practice approaches for disaster planning

PILLAR	BEST-PRACTICE APPROACH	WHAT TO DO
Reliability	Failure management — plan for disaster recovery	Define recovery objectives (RTO/RPO); use a defined recovery strategy; test the DR implementation
Operational Excellence	Operate — manage workload and operations events	Define a customer communication plan for outages
Security	Identity and access management — permissions management	Establish an emergency access process

RELIABILITY – PLAN FOR DISASTER RECOVERY

- **Define recovery objectives for downtime and data loss:** every workload gets an RTO and RPO based on business impact (monetary cost, loss of customer trust, operational issues, regulatory risk). Additional strategy considerations: cost constraints, workload dependencies, and operational requirements.
- **Use defined recovery strategies to meet the objectives:** choose backup and restore, pilot light, warm standby, or multi-site — a trade-off between reducing RTO/RPO and the strategy's cost and complexity. Mitigate a data disaster too: replicated data isn't protected from corruption unless you also version or snapshot it and back up the replicated data at the recovery site.
- **Test the DR implementation:** regularly test failover to the recovery site to verify it operates properly and meets RTO and RPO — the only error recovery that works is the path you test frequently.

OPERATIONAL EXCELLENCE – MANAGE WORKLOAD AND OPERATIONS EVENTS

- **Define a customer communication plan for outages:** communicate directly with users both when their services are impacted and when service returns to normal.
- Example: send an **email notification** describing the impaired functionality with a realistic restoration estimate, maintain a **status page** with real-time health, and **test the plan** in a development environment (for example, twice per year).

SECURITY – ESTABLISH AN EMERGENCY ACCESS PROCESS

- Create an **emergency access** process for the unlikely event of an issue with your centralized identity provider: if it fails or its federation configuration is modified, workforce users may be unable to **federate** into the cloud, so authorized administrators need an alternate path to fix it.
- Well-documented, well-tested emergency access reduces response time — **less downtime**, higher availability — and is exercised through the same Game Day and continuous-testing practices covered in this module.

COMMON PITFALL

Well-Architected DR is more than a running duplicate. A resilient design still fails the framework if recovery objectives were never defined from business impact, if the recovery path is never tested, if customers are left uninformed during an outage, or if a failed identity provider locks administrators out. Define objectives, pick a strategy, **test it**, communicate, and keep an emergency access door open.

Module summary

- Everything fails — assume failures at small, large, or global scale (weigh both size and impact), and a DR plan limits business and customer damage. Architects work backward from the business need using three levers: fault tolerance (redundancy that keeps functioning), backup (protecting data as it outgrows local disk durability), and disaster recovery (policies and procedures to recover vital infrastructure after any negative-impact event).
- RPO is the maximum acceptable data loss measured in time (look back to the last good copy — it sets backup frequency); RTO is the maximum acceptable downtime for a process (look forward to restoration — it sets the recovery strategy). Both come from financial impact and live in the BCP alongside a business impact analysis and risk assessment; the DR plan is a subset of the BCP.
- DR plans span more than one Region across five service categories: Storage (S3), Compute (EC2), Database (RDS), Networking & Content Delivery (VPC), and Deployment orchestration (CloudFormation). RPO and RTO guide backup and restore across each.
- Data-protection building blocks: S3 CRR (11-nines durability) and S3 RTC (99.99% within 15 minutes under an SLA); incremental EBS snapshots copyable across Regions and automated by Data Lifecycle Manager (you cannot snapshot instance store volumes); and EFS/FSx replication via DataSync and AWS Backup.
- Compute, network, and database recovery: EC2 automatic recovery and golden AMIs; Route 53 failover, ELB, Amazon VPN, and Direct Connect; EventBridge global endpoints; RDS snapshots, read replicas, and Multi-AZ; DynamoDB PITR and global tables; and CloudFormation, Elastic Beanstalk, and OpsWorks to redeploy environments.
- The four DR patterns, in rising cost and falling RTO: backup and restore (hours, cheapest, nothing extra running), pilot light (10s of minutes, only the core runs), warm standby (minutes, a scaled-down full stack runs

24/7), and multi-site (near real-time, a full active/active duplicate — costliest but least downtime). For hybrid backup and restore, AWS Storage Gateway offers File Gateway (NFS/SMB → S3), Volume Gateway (iSCSI → EBS snapshots), and Tape Gateway (VTL → S3, archived to S3 Glacier).

- Replication is not a backup (it copies corruption); version or snapshot for point-in-time recovery and test DR frequently with Game Day exercises. Well-Architected DR: define recovery objectives, use a recovery strategy and test it, plan customer communication for outages, and establish emergency access.

Review questions

1. Distinguish RPO from RTO, and say what each one drives.

Answer: RPO is the maximum acceptable data loss, measured in time — you look back to the last recoverable data, so it drives how often you back up. RTO is the maximum acceptable downtime after a disaster — you look forward to restoration, so it drives which recovery strategy (pattern) you choose. Both are set from the business/financial impact of an outage.

2. An application can lose at most 800 records and creates up to 100 records per hour. What RPO does that imply, and what does it mean for backups?

Answer: An RPO of 8 hours (800 ÷ 100). It means backing up at least every 8 hours; if a disaster occurred at 10 p.m., the system should recover all data that was in the system before 2 p.m.

3. Name the five AWS service categories a DR plan spans, with one example service each.

Answer: Storage (Amazon S3), Compute (Amazon EC2), Database (Amazon RDS), Networking & Content Delivery (Amazon VPC), and Deployment orchestration in Management & Governance (AWS CloudFormation).

4. Order the four DR patterns by cost and RTO, and give the always-running footprint of each.

Answer: Backup and restore — cheapest, RTO in hours, only backups stored. Pilot light — low cost, 10s of minutes, only the core (e.g., the database) running. Warm standby — higher cost, minutes, a scaled-down full stack running 24/7. Multi-site — costliest, near real-time, a full active/active duplicate running simultaneously.

5. Why is replication not a substitute for a backup?

Answer: Replicas copy the source faithfully, including corruption, so continuously replicated data (S3 CRR, RDS read replicas) offers no protection against data corruption. You must also version or snapshot the data for point-in-time recovery and back up the replicated data in the recovery site.

6. A business needs an RPO of 7 minutes for business-critical apps and must handle the majority of traffic immediately, then scale to full capacity — most cost-effective pattern, and why not the others?

Answer: Warm standby: it meets the 7-minute RPO, runs at low capacity but can scale within minutes, and costs less than multi-site. Backup and restore cannot restore fast enough; pilot light's idle core cannot handle the majority of traffic immediately; multi-site meets the requirement but costs more than necessary.

7. Which three Well-Architected best-practice approaches does this module map to disaster planning, and to which pillars?

Answer: Reliability — plan for disaster recovery (define recovery objectives, use a recovery strategy, test it). Operational Excellence — define a customer communication plan for outages. Security — establish an emergency access process.