**REL 1. How do you manage service quotas and constraints?**

For cloud-based workload architectures, there are service quotas (which are also referred to as service limits). These quotas exist to prevent accidentally provisioning more resources than you need and to limit request rates on API operations so as to protect services from abuse. There are also resource constraints, for example, the rate that you can push bits down a fiber-optic cable, or the amount of storage on a physical disk.

[AWS Live re:Inforce 2019 - Service Quotas](https://youtu.be/O9R5dWgtrVo?ref=wellarchitected)  
 [What Is Service Quotas?](https://docs.aws.amazon.com/servicequotas/latest/userguide/intro.html?ref=wellarchitected)  
 [AWS Service Quotas (formerly referred to as service limits)](http://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html?ref=wellarchitected)  
 [Amazon EC2 Service Limits](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-resource-limits.html?ref=wellarchitected)  
 [AWS Trusted Advisor Best Practice Checks (see the Service Limits section)](https://aws.amazon.com/premiumsupport/technology/trusted-advisor/best-practice-checklist/?ref=wellarchitected)  
 [AWS limit monitor on AWS answers](https://aws.amazon.com/answers/account-management/limit-monitor/?ref=wellarchitected)  
 [AWS Marketplace: CMDB products that help track limits](https://aws.amazon.com/marketplace/search/results?searchTerms=CMDB&ref=wellarchitected)  
 [APN Partner: partners that can help with configuration management](https://aws.amazon.com/partners/find/results/?keyword=Configuration+Management&ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

**Aware of service quotas and constraints**You are aware of your default quotas and quota increase requests for your [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) [architecture](https://wa.aws.amazon.com/wat.concept.architecture.en.html). You additionally know which resource constraints, such as disk or network, are potentially impactful.

**Manage service quotas across accounts and regions**If you are using multiple AWS accounts or [AWS Regions](https://wa.aws.amazon.com/wat.concept.region.en.html), ensure that you request the appropriate quotas in all environments in which your production [workloads](https://wa.aws.amazon.com/wat.concept.workload.en.html) run.

**Accommodate fixed service quotas and constraints through architecture**Be aware of unchangeable service quotas and physical resources, and architect to prevent these from impacting [reliability](https://wa.aws.amazon.com/wat.concept.c-reliability.en.html).

**Monitor and manage quotas**Evaluate your potential usage and increase your quotas appropriately allowing for planned growth in usage.

**Automate quota management**Implement tools to alert you when thresholds are being approached. By using AWS Service Quotas APIs, you can automate quota increase requests.

**Ensure that a sufficient gap exists between the current quotas and the maximum usage to accommodate failover**When a resource fails, it may still be counted against quotas until its successfully terminated. Ensure that your quotas cover the overlap of all failed resources with replacements before the failed resources are terminated. You should consider an [Availability Zone](https://wa.aws.amazon.com/wat.concept.az.en.html) failure when calculating this gap.

**REL 2. How do you plan your network topology?**

Workloads often exist in multiple environments. These include multiple cloud environments (both publicly accessible and private) and possibly your existing data center infrastructure. Plans must include network considerations such as intra- and inter-system connectivity, public IP address management, private IP address management, and domain name resolution.

[AWS re:Invent 2018: Advanced VPC Design and New Capabilities for Amazon VPC (NET303)](https://youtu.be/fnxXNZdf6ew?ref=wellarchitected)  
 [AWS re:Invent 2019: AWS Transit Gateway reference architectures for many VPCs (NET406-R1)](https://youtu.be/9Nikqn_02Oc?ref=wellarchitected)  
 [What is Amazon Route 53?](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html?ref=wellarchitected)  
 [What is Elastic Load Balancing?](https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html?ref=wellarchitected)  
 [What is Amazon CloudFront?](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html?ref=wellarchitected)  
 [What Is a Transit Gateway?](https://docs.aws.amazon.com/vpc/latest/tgw/what-is-transit-gateway.html?ref=wellarchitected)  
 [What Is Amazon VPC?](https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html?ref=wellarchitected)  
 [Working with Direct Connect Gateways](https://docs.aws.amazon.com/directconnect/latest/UserGuide/direct-connect-gateways.html?ref=wellarchitected)  
 [Using the Direct Connect Resiliency Toolkit to get started](https://docs.aws.amazon.com/directconnect/latest/UserGuide/resilency_toolkit.html?ref=wellarchitected)  
 [AWS Direct Connect Resiliency Recommendations](https://aws.amazon.com/directconnect/resiliency-recommendation/?ref=wellarchitected)  
 [Multiple data center HA network connectivity](https://aws.amazon.com/answers/networking/aws-multiple-data-center-ha-network-connectivity/?ref=wellarchitected)  
 [What Is AWS Global Accelerator?](https://docs.aws.amazon.com/global-accelerator/latest/dg/what-is-global-accelerator.html?ref=wellarchitected)  
 [VPC Endpoints and VPC Endpoint Services (AWS PrivateLink)](https://docs.aws.amazon.com/vpc/latest/userguide/endpoint-services-overview.html?ref=wellarchitected)  
 [Amazon Virtual Private Cloud Connectivity Options Whitepaper](https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/introduction.html?ref=wellarchitected)  
 [AWS Marketplace for Network Infrastructure](https://aws.amazon.com/marketplace/b/2649366011?ref=wellarchitected)  
 [APN Partner: partners that can help plan your networking](https://aws.amazon.com/partners/find/results/?keyword=network&ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

**Use highly available network connectivity for your workload public endpoints**These endpoints and the routing to them must be highly available. To achieve this, use highly available DNS, content delivery networks (CDNs), API Gateway, load balancing, or reverse proxies.

**Provision redundant connectivity between private networks in the cloud and on-premises environments**Use multiple [AWS Direct Connect](https://wa.aws.amazon.com/wat.concept.awsdirectconnect.en.html) (DX) connections or VPN tunnels between separately deployed private networks. Use multiple DX locations for high [availability](https://wa.aws.amazon.com/wat.concept.availability.en.html). If using multiple [AWS Regions](https://wa.aws.amazon.com/wat.concept.region.en.html), ensure redundancy in at least two of them. You might want to evaluate [AWS Marketplace](https://wa.aws.amazon.com/wat.concept.marketplace.en.html) appliances that terminate VPNs. If you use [AWS Marketplace](https://wa.aws.amazon.com/wat.concept.marketplace.en.html) appliances, deploy redundant instances for high [availability](https://wa.aws.amazon.com/wat.concept.availability.en.html) in different [Availability Zones](https://wa.aws.amazon.com/wat.concept.az.en.html).

**Ensure IP subnet allocation accounts for expansion and availability**[Amazon VPC](https://wa.aws.amazon.com/wat.concept.amazonvirtualprivatecloud.en.html) IP address ranges must be large enough to accommodate [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) requirements, including factoring in future expansion and allocation of IP addresses to subnets across [Availability Zones](https://wa.aws.amazon.com/wat.concept.az.en.html). This includes load balancers, [EC2 instances](https://wa.aws.amazon.com/wat.concept.ec2-instance.en.html), and container-based applications.

**Prefer hub-and-spoke topologies over many-to-many mesh**If more than two network address spaces (for example, VPCs and on-premises networks) are connected via VPC peering, [AWS Direct Connect](https://wa.aws.amazon.com/wat.concept.awsdirectconnect.en.html), or VPN, then use a hub-and-spoke model, like that provided by AWS Transit Gateway.

**Enforce non-overlapping private IP address ranges in all private address spaces where they are connected**The IP address ranges of each of your VPCs must not overlap when peered or connected via VPN. You must similarly avoid IP address conflicts between a VPC and on-premises environments or with other cloud providers that you use. You must also have a way to allocate private IP address ranges when needed.

**REL 3. How do you design your workload service architecture?**

Build highly scalable and reliable workloads using a service-oriented architecture (SOA) or a microservices architecture. Service-oriented architecture (SOA) is the practice of making software components reusable via service interfaces. Microservices architecture goes further to make components smaller and simpler.

[Amazon API Gateway: Configuring a REST API Using OpenAPI](https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-import-api.html?ref=wellarchitected)  
 [Implementing Microservices on AWS](https://docs.aws.amazon.com/whitepapers/latest/microservices-on-aws/introduction.html?ref=wellarchitected)  
 [Microservices on AWS](https://aws.amazon.com/microservices/?ref=wellarchitected)  
 [Microservices - a definition of this new architectural term](https://www.martinfowler.com/articles/microservices.html?ref=wellarchitected)  
 [Microservice Trade-Offs](https://martinfowler.com/articles/microservice-trade-offs.html?ref=wellarchitected)  
 [Bounded Context (a central pattern in Domain-Driven Design)](https://martinfowler.com/bliki/BoundedContext.html?ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

**Choose how to segment your workload**Monolithic [architecture](https://wa.aws.amazon.com/wat.concept.architecture.en.html) should be avoided. Instead, you should choose between SOA and microservices. When making each choice, balance the benefits against the complexities—what is right for a new product racing to first launch is different than what a [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) built to scale from the start needs. The benefits of using smaller segments include greater agility, organizational flexibility, and [scalability](https://wa.aws.amazon.com/wat.concept.scalability.en.html). Complexities include possible increased [latency](https://wa.aws.amazon.com/wat.concept.latency.en.html), more complex debugging, and increased operational burden

**Build services focused on specific business domains and functionality**SOA builds services with well-delineated functions defined by business needs. Microservices use domain models and bounded context to limit this further so that each service does just one thing. Focusing on specific functionality enables you to differentiate the [reliability](https://wa.aws.amazon.com/wat.concept.c-reliability.en.html) requirements of different services, and target investments more specifically. A concise business [problem](https://wa.aws.amazon.com/wat.concept.problem.en.html) and having a small team associated with each service also enables easier organizational scaling.

**Provide service contracts per API**Service contracts are documented agreements between teams on service integration and include a machine-readable API definition, rate limits, and [performance](https://wa.aws.amazon.com/wat.pillar.performance.en.html) expectations. A versioning strategy allows clients to continue using the existing API and migrate their applications to the newer API when they are ready. Deployment can happen anytime, as long as the contract is not violated. The service provider team can use the technology stack of their choice to satisfy the API contract. Similarly, the service consumer can use their own technology.

**REL 4. How do you design interactions in a distributed system to prevent failures?**

Distributed systems rely on communications networks to interconnect components, such as servers or services. Your workload must operate reliably despite data loss or latency in these networks. Components of the distributed system must operate in a way that does not negatively impact other components or the workload. These best practices prevent failures and improve mean time between failures (MTBF).

[AWS re:Invent 2019: Moving to event-driven architectures (SVS308)](https://youtu.be/h46IquqjF3E?ref=wellarchitected)  
 [AWS re:Invent 2018: Close Loops and Opening Minds: How to Take Control of Systems, Big and Small ARC337 (includes loose coupling, constant work, static stability)](https://youtu.be/O8xLxNje30M?ref=wellarchitected)  
 [AWS New York Summit 2019: Intro to Event-driven Architectures and Amazon EventBridge (MAD205)](https://youtu.be/tvELVa9D9qU?ref=wellarchitected)  
 [What Is Amazon EventBridge?](https://docs.aws.amazon.com/eventbridge/latest/userguide/what-is-amazon-eventbridge.html?ref=wellarchitected)  
 [What Is Amazon Simple Queue Service?](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/welcome.html?ref=wellarchitected)  
 [Amazon EC2: Ensuring Idempotency](https://docs.aws.amazon.com/AWSEC2/latest/APIReference/Run_Instance_Idempotency.html?ref=wellarchitected)  
 [The Amazon Builders' Library: Challenges with distributed systems](https://aws.amazon.com/builders-library/challenges-with-distributed-systems/?ref=wellarchitected)  
 [The Amazon Builders' Library: Reliability, constant work, and a good cup of coffee](https://aws.amazon.com/builders-library/reliability-and-constant-work/?ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

#### Identify which kind of distributed system is required Hard real-time distributed systems require responses to be given synchronously and rapidly, while soft real-time systems have a more generous time window of minutes or more for response. Offline systems handle responses through batch or asynchronous processing. Hard real-time distributed systems have the most stringent [reliability](https://wa.aws.amazon.com/wat.concept.c-reliability.en.html) requirements.

#### Implement loosely coupled dependencies Dependencies such as queuing systems, streaming systems, workflows, and load balancers are loosely coupled. Loose coupling helps isolate behavior of a [component](https://wa.aws.amazon.com/wat.concept.component.en.html) from other [components](https://wa.aws.amazon.com/wat.concept.component.en.html) that depend on it, increasing [resiliency](https://wa.aws.amazon.com/wat.concept.resiliency.en.html) and agility

#### Do constant work Systems can fail when there are large, rapid changes in load. For example, if your [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) is doing a health check that monitors the health of thousands of servers, it should send the same size payload (a full snapshot of the current state) each time. Whether no servers are failing, or all of them, the health check system is doing constant work with no large, rapid changes.

#### Make all responses idempotent An idempotent service promises that each request is completed exactly once, such that making multiple identical requests has the same effect as making a single request. An idempotent service makes it easier for a client to implement retries without fear that a request will be erroneously processed multiple times. To do this, clients can issue API requests with an idempotency token—the same token is used whenever the request is repeated. An idempotent service API uses the token to return a response identical to the response that was returned the first time that the request was completed.

**REL 5. How do you design interactions in a distributed system to mitigate or withstand failures?**

Distributed systems rely on communications networks to interconnect components (such as servers or services). Your workload must operate reliably despite data loss or latency over these networks. Components of the distributed system must operate in a way that does not negatively impact other components or the workload. These best practices enable workloads to withstand stresses or failures, more quickly recover from them, and mitigate the impact of such impairments. The result is improved mean time to recovery (MTTR).

[Retry, backoff, and jitter: AWS re:Invent 2019: Introducing The Amazon Builders’ Library (DOP328)](https://youtu.be/sKRdemSirDM?t=1884&ref=wellarchitected)  
 [Error Retries and Exponential Backoff in AWS](https://docs.aws.amazon.com/general/latest/gr/api-retries.html?ref=wellarchitected)  
 [Amazon API Gateway: Throttle API Requests for Better Throughput](https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-request-throttling.html?ref=wellarchitected)  
 [The Amazon Builders' Library: Timeouts, retries, and backoff with jitter](https://aws.amazon.com/builders-library/timeouts-retries-and-backoff-with-jitter/?ref=wellarchitected)  
 [The Amazon Builders' Library: Avoiding fallback in distributed systems](https://aws.amazon.com/builders-library/avoiding-fallback-in-distributed-systems?ref=wellarchitected)  
 [The Amazon Builders' Library: Avoiding insurmountable queue backlogs](https://aws.amazon.com/builders-library/avoiding-insurmountable-queue-backlogs?ref=wellarchitected)  
 [The Amazon Builders' Library: Caching challenges and strategies](https://aws.amazon.com/builders-library/caching-challenges-and-strategies/?ref=wellarchitected)  
 [Well-Architected lab: Level 300: Implementing Health Checks and Managing Dependencies to Improve Reliability](https://wellarchitectedlabs.com/Reliability/300_Health_Checks_and_Dependencies/README.html?ref=wellarchitected)  
 [CircuitBreaker (summarizes Circuit Breaker from “Release It!” book)](https://martinfowler.com/bliki/CircuitBreaker.html?ref=wellarchitected)  
 [Michael Nygard “Release It! Design and Deploy Production-Ready Software”](https://pragprog.com/titles/mnee2/release-it-second-edition/?ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

#### Implement graceful degradation to transform applicable hard dependencies into soft dependencies When a [component](https://wa.aws.amazon.com/wat.concept.component.en.html)'s dependencies are unhealthy, the [component](https://wa.aws.amazon.com/wat.concept.component.en.html) itself can still function, although in a degraded manner. For example, when a dependency call fails, failover to a predetermined static response.

#### Throttle requests This is a mitigation pattern to respond to an unexpected increase in demand. Some requests are honored but those over a defined limit are rejected and return a message indicating they have been throttled. The expectation on clients is that they will back off and abandon the request or try again at a slower rate.

#### Control and limit retry calls Use exponential backoff to retry after progressively longer intervals. Introduce jitter to randomize those retry intervals, and limit the maximum number of retries.

#### Fail fast and limit queues If the [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) is unable to respond successfully to a request, then fail fast. This allows the releasing of resources associated with a request, and permits the service to recover if it’s running out of resources. If the [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) is able to respond successfully but the rate of requests is too high, then use a queue to buffer requests instead. However, do not allow long queues that can result in serving stale requests that the client has already given up on.

#### Set client timeouts Set timeouts appropriately, verify them systematically, and do not rely on default values as they are generally set too high

#### Make services stateless where possible Services should either not require state, or should offload state such that between different client requests, there is no dependence on locally stored data on disk or in [memory](https://wa.aws.amazon.com/wat.concept.memory.en.html). This enables servers to be replaced at will without causing an [availability](https://wa.aws.amazon.com/wat.concept.availability.en.html) impact. [Amazon ElastiCache](https://wa.aws.amazon.com/wat.concept.elasticache.en.html) or [Amazon DynamoDB](https://wa.aws.amazon.com/wat.concept.dynamodb.en.html) are good destinations for offloaded state.

#### Implement emergency levers These are rapid processes that may mitigate [availability](https://wa.aws.amazon.com/wat.concept.availability.en.html) impact on your [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html). They can be operated in the absence of a [root cause](https://wa.aws.amazon.com/wat.concept.rca.en.html). An ideal emergency lever reduces the cognitive burden on the resolvers to zero by providing fully deterministic activation and deactivation criteria. Example levers include blocking all robot traffic or serving a static response. Levers are often manual, but they can also be automated.

**REL 6. How do you monitor workload resources?**

Logs and metrics are powerful tools to gain insight into the health of your workload. You can configure your workload to monitor logs and metrics and send notifications when thresholds are crossed or significant events occur. Monitoring enables your workload to recognize when low-performance thresholds are crossed or failures occur, so it can recover automatically in response.

[Using Amazon CloudWatch Metrics](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/working_with_metrics.html?ref=wellarchitected)  
 [Publishing Custom Metrics](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/publishingMetrics.html?ref=wellarchitected)  
 [Using Amazon CloudWatch Dashboards](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_Dashboards.html?ref=wellarchitected)  
 [Using Canaries (Amazon CloudWatch Synthetics)](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries.html?ref=wellarchitected)  
 [Amazon CloudWatch Logs Insights Sample Queries](https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/CWL_QuerySyntax-examples.html?ref=wellarchitected)  
 [AWS Systems Manager Automation](https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-automation.html?ref=wellarchitected)  
 [What is Amazon DevOps Guru?](https://docs.aws.amazon.com/devops-guru/latest/userguide/welcome.html?ref=wellarchitected)  
 [What is AWS X-Ray?](https://docs.aws.amazon.com/xray/latest/devguide/aws-xray.html?ref=wellarchitected)  
 [Debugging with Amazon CloudWatch Synthetics and AWS X-Ray](https://aws.amazon.com/blogs/devops/debugging-with-amazon-cloudwatch-synthetics-and-aws-x-ray/?ref=wellarchitected)  
 [The Amazon Builders' Library: Instrumenting distributed systems for operational visibility](https://aws.amazon.com/builders-library/instrumenting-distributed-systems-for-operational-visibility/?ref=wellarchitected)  
 [One Observability Workshop](https://observability.workshop.aws/?ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

#### Monitor all components for the workload (Generation) Monitor the [components](https://wa.aws.amazon.com/wat.concept.component.en.html) of the [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) with [Amazon CloudWatch](https://wa.aws.amazon.com/wat.concept.amazoncw.en.html) or third-party tools. Monitor AWS services with Personal Health Dashboard

#### Define and calculate metrics (Aggregation) Store log data and apply filters where necessary to calculate metrics, such as counts of a specific log [event](https://wa.aws.amazon.com/wat.concept.event.en.html), or [latency](https://wa.aws.amazon.com/wat.concept.latency.en.html) calculated from log [event](https://wa.aws.amazon.com/wat.concept.event.en.html) timestamps

#### Send notifications (Real-time processing and alarming) Organizations that need to know, receive notifications when significant [events](https://wa.aws.amazon.com/wat.concept.event.en.html) occur

#### Automate responses (Real-time processing and alarming) Use automation to take action when an [event](https://wa.aws.amazon.com/wat.concept.event.en.html) is detected, for example, to replace failed [components](https://wa.aws.amazon.com/wat.concept.component.en.html)

#### Analytics Collect log files and metrics histories and analyze these for broader trends and [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) insights

#### Conduct reviews regularly Frequently review how [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) monitoring is implemented and update it based on significant [events](https://wa.aws.amazon.com/wat.concept.event.en.html) and changes

#### Monitor end-to-end tracing of requests through your system Use [AWS X-Ray](https://wa.aws.amazon.com/wat.concept.awsxray.en.html) or third-party tools so that developers can more easily analyze and debug distributed systems to understand how their applications and its underlying services are performing

**REL 7. How do you design your workload to adapt to changes in demand?**

A scalable workload provides elasticity to add or remove resources automatically so that they closely match the current demand at any given point in time.

[AWS Auto Scaling: How Scaling Plans Work](https://docs.aws.amazon.com/autoscaling/plans/userguide/how-it-works.html?ref=wellarchitected)  
 [What Is Amazon EC2 Auto Scaling?](https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html?ref=wellarchitected)  
 [Managing Throughput Capacity Automatically with DynamoDB Auto Scaling](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/AutoScaling.html?ref=wellarchitected)  
 [What is Amazon CloudFront?](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html?ref=wellarchitected&ref=wellarchitected)  
 [Distributed Load Testing on AWS: simulate thousands of connected users](https://aws.amazon.com/solutions/distributed-load-testing-on-aws/?ref=wellarchitected)  
 [Telling Stories About Little's Law](https://brooker.co.za/blog/2018/06/20/littles-law.html?ref=wellarchitected)  
 [AWS Marketplace: products that can be used with auto scaling](https://aws.amazon.com/marketplace/search/results?searchTerms=Auto+Scaling&ref=wellarchitected)  
 [APN Partner: partners that can help you create automated compute solutions](https://aws.amazon.com/partners/find/results/?facets=%27Product%20:%20Compute%27&ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

#### Use automation when obtaining or scaling resources When replacing impaired resources or scaling your [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html), automate the process by using managed AWS services, such as [Amazon S3](https://wa.aws.amazon.com/wat.concept.amazonsimplestorageservice.en.html) and [AWS Auto Scaling](https://wa.aws.amazon.com/wat.concept.awsautoscaling.en.html). You can also use third-party tools and AWS SDKs to automate scaling.

#### Obtain resources upon detection of impairment to a workload Scale resources reactively when necessary if [availability](https://wa.aws.amazon.com/wat.concept.availability.en.html) is impacted, to restore [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) [availability](https://wa.aws.amazon.com/wat.concept.availability.en.html).

#### Obtain resources upon detection that more resources are needed for a workload Scale resources proactively to meet demand and avoid [availability](https://wa.aws.amazon.com/wat.concept.availability.en.html) impact.

#### Load test your workload Adopt a load testing methodology to measure if scaling activity meets [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) requirements.

**REL 8. How do you implement change?**

Controlled changes are necessary to deploy new functionality, and to ensure that the workloads and the operating environment are running known software and can be patched or replaced in a predictable manner. If these changes are uncontrolled, then it makes it difficult to predict the effect of these changes, or to address issues that arise because of them.

[AWS Summit 2019: CI/CD on AWS](https://youtu.be/tQcF6SqWCoY?ref=wellarchitected)  
 [What Is AWS CodePipeline?](https://docs.aws.amazon.com/codepipeline/latest/userguide/welcome.html?ref=wellarchitected)  
 [What Is CodeDeploy?](https://docs.aws.amazon.com/codedeploy/latest/userguide/welcome.html?ref=wellarchitected)  
 [Overview of a Blue/Green Deployment](https://docs.aws.amazon.com/codedeploy/latest/userguide/welcome.html?ref=wellarchitected#welcome-deployment-overview-blue-green)  
 [Deploying Serverless Applications Gradually](https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/automating-updates-to-serverless-apps.html?ref=wellarchitected)  
 [The Amazon Builders' Library: Ensuring rollback safety during deployments](https://aws.amazon.com/builders-library/ensuring-rollback-safety-during-deployments?ref=wellarchitected)  
 [The Amazon Builders' Library: Going faster with continuous delivery](https://aws.amazon.com/builders-library/going-faster-with-continuous-delivery/?ref=wellarchitected)  
 [Well-Architected lab: Level 300: Testing for Resiliency of EC2 RDS and S3](https://wellarchitectedlabs.com/Reliability/300_Testing_for_Resiliency_of_EC2_RDS_and_S3/README.html?ref=wellarchitected)  
 [Immutable Infrastructure: Reliability, consistency and confidence through immutability](https://medium.com/@adhorn/immutable-infrastructure-21f6613e7a23?ref=wellarchitected)  
 [CanaryRelease](https://martinfowler.com/bliki/CanaryRelease.html?ref=wellarchitected)  
 [AWS Marketplace: products that can be used to automate your deployments](https://aws.amazon.com/marketplace/search/results?searchTerms=DevOps&ref=wellarchitected)  
 [APN Partner: partners that can help you create automated deployment solutions](https://aws.amazon.com/partners/find/results/?keyword=devops&ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

#### Use runbooks for standard activities such as deployment [Runbooks](https://wa.aws.amazon.com/wat.concept.runbook.en.html) are the predefined steps used to achieve specific outcomes. Use [runbooks](https://wa.aws.amazon.com/wat.concept.runbook.en.html) to perform standard activities, whether done manually or automatically. Examples include deploying a [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html), patching it, or making DNS modifications.

#### Integrate functional testing as part of your deployment Functional tests are run as part of automated deployment. If success criteria are not met, the pipeline is halted or rolled back.

#### Integrate resiliency testing as part of your deployment [Resiliency](https://wa.aws.amazon.com/wat.concept.resiliency.en.html) tests (as part of chaos engineering) are run as part of the automated deployment pipeline in a pre-prod environment.

#### Deploy using immutable infrastructure This is a model that mandates that no updates, [security](https://wa.aws.amazon.com/wat.pillar.security.en.html) patches, or configuration changes happen in-place on production [workloads](https://wa.aws.amazon.com/wat.concept.workload.en.html). When a change is needed, the [architecture](https://wa.aws.amazon.com/wat.concept.architecture.en.html) is built onto new infrastructure and deployed into production.

#### Deploy changes with automation Deployments and patching are automated to eliminate negative impact.

**REL 9. How do you back up data?**

Back up data, applications, and configuration to meet your requirements for recovery time objectives (RTO) and recovery point objectives (RPO).

[What Is AWS Backup?](https://docs.aws.amazon.com/aws-backup/latest/devguide/whatisbackup.html?ref=wellarchitected)  
 [AWS re:Invent 2019: Deep dive on AWS Backup, ft. Rackspace (STG341)](https://youtu.be/av8DpL0uFjc?ref=wellarchitected)  
 [Well-Architected lab: Level 200: Testing Backup and Restore of Data](https://wellarchitectedlabs.com/Reliability/200_Testing_Backup_and_Restore_of_Data/README.html?ref=wellarchitected)  
 [Amazon S3: Protecting Data Using Encryption](https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html?ref=wellarchitected)  
 [Encryption for Backups in AWS](https://docs.aws.amazon.com/aws-backup/latest/devguide/encryption.html?ref=wellarchitected)  
 [On-demand backup and restore for DynamoDB](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/BackupRestore.html?ref=wellarchitected)  
 [Well-Architected lab: Testing Backup and Restore of Data](https://wellarchitectedlabs.com/reliability/200_labs/200_testing_backup_and_restore_of_data/?ref=wellarchitected)  
 [Well-Architected lab: Implementing Bi-Directional Cross-Region Replication (CRR) for Amazon S3](https://wellarchitectedlabs.com/reliability/200_labs/200_bidirectional_replication_for_s3/?ref=wellarchitected)  
 [EFS-to-EFS backup](https://aws.amazon.com/solutions/efs-to-efs-backup-solution/?ref=wellarchitected)  
 [AWS Marketplace: products that can be used for backup](https://aws.amazon.com/marketplace/search/results?searchTerms=Backup&ref=wellarchitected)  
 [APN Partner: partners that can help with backup](https://aws.amazon.com/partners/find/results/?keyword=Backup&ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

#### Identify and back up all data that needs to be backed up, or reproduce the data from sources [Amazon S3](https://wa.aws.amazon.com/wat.concept.amazonsimplestorageservice.en.html) can be used as a backup destination for multiple data sources. AWS services such as [Amazon EBS](https://wa.aws.amazon.com/wat.concept.ebs.en.html), [Amazon RDS](https://wa.aws.amazon.com/wat.concept.amazonrelationaldatabaseservice.en.html), and [Amazon DynamoDB](https://wa.aws.amazon.com/wat.concept.dynamodb.en.html) have built in capabilities to create backups. Third-party backup software can also be used. Alternatively, if the data can be reproduced from other sources to meet RPO, you might not require a backup

#### Secure and encrypt backups Detect access using authentication and authorization, such as AWS [IAM](https://wa.aws.amazon.com/wat.concept.iam.en.html), and detect data integrity compromise by using encryption.

#### Perform data backup automatically Configure backups to be taken automatically based on a periodic schedule, or by changes in the dataset. RDS instances, EBS volumes, DynamoDB tables, and S3 objects can all be configured for automatic backup. [AWS Marketplace](https://wa.aws.amazon.com/wat.concept.marketplace.en.html) solutions or third-party solutions can also be used.

#### Perform periodic recovery of the data to verify backup integrity and processes Validate that your backup process implementation meets your recovery time objectives (RTO) and recovery point objectives (RPO) by performing a recovery test.

**REL 10. How do you use fault isolation to protect your workload?**

Fault isolated boundaries limit the effect of a failure within a workload to a limited number of components. Components outside of the boundary are unaffected by the failure. Using multiple fault isolated boundaries, you can limit the impact on your workload.

[AWS re:Invent 2018: Architecture Patterns for Multi-Region Active-Active Applications (ARC209-R2)](https://youtu.be/2e29I3dA8o4?ref=wellarchitected)  
 [Shuffle-sharding: AWS re:Invent 2019: Introducing The Amazon Builders’ Library (DOP328)](https://youtu.be/sKRdemSirDM?t=1373&ref=wellarchitected)  
 [AWS re:Invent 2018: How AWS Minimizes the Blast Radius of Failures (ARC338)](https://youtu.be/swQbA4zub20?ref=wellarchitected)  
 [AWS re:Invent 2019: Innovation and operation of the AWS global network infrastructure (NET339)](https://youtu.be/UObQZ3R9_4c?ref=wellarchitected)  
 [What is AWS Outposts?](https://docs.aws.amazon.com/outposts/latest/userguide/what-is-outposts.html?ref=wellarchitected)  
 [Global Tables: Multi-Region Replication with DynamoDB](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GlobalTables.html?ref=wellarchitected)  
 [AWS Local Zones FAQ](https://aws.amazon.com/about-aws/global-infrastructure/localzones/faqs/?ref=wellarchitected)  
 [AWS Global Infrastructure](https://aws.amazon.com/about-aws/global-infrastructure?ref=wellarchitected)  
 [The Amazon Builders' Library: Workload isolation using shuffle-sharding](https://aws.amazon.com/builders-library/workload-isolation-using-shuffle-sharding/?ref=wellarchitected)  
 [Well-Architected lab: Fault isolation with shuffle sharding](https://wellarchitectedlabs.com/reliability/300_labs/300_fault_isolation_with_shuffle_sharding/?ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

#### Deploy the workload to multiple locations Distribute [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) data and resources across multiple [Availability Zones](https://wa.aws.amazon.com/wat.concept.az.en.html) or, where necessary, across [AWS Regions](https://wa.aws.amazon.com/wat.concept.region.en.html). These locations can be as diverse as required.

#### Select the appropriate locations for your multi-location deployment Always use multiple [AZs](https://wa.aws.amazon.com/wat.concept.az.en.html) where possible within an [AWS Region](https://wa.aws.amazon.com/wat.concept.region.en.html). For [workloads](https://wa.aws.amazon.com/wat.concept.workload.en.html) that require more [resiliency](https://wa.aws.amazon.com/wat.concept.resiliency.en.html), also use a multi-Region strategy, such as active-passive or active-active.

#### Automate recovery for components constrained to a single location If [components](https://wa.aws.amazon.com/wat.concept.component.en.html) of the [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) can only run in a single [Availability Zone](https://wa.aws.amazon.com/wat.concept.az.en.html) or on-premises data center, you must implement the capability to do a complete rebuild of the [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) within your defined recovery objectives.

#### Use bulkhead architectures to limit scope of impact Like the bulkheads on a ship, this pattern ensures that a failure is contained to a small subset of requests or users so that the number of impaired requests is limited, and most can continue without error. Bulkheads for data are often called [partitions](https://wa.aws.amazon.com/wat.concept.partition.en.html), while bulkheads for services are known as cells.

**REL 11. How do you design your workload to withstand component failures?**

Workloads with a requirement for high availability and low mean time to recovery (MTTR) must be architected for resiliency.

[Static stability in AWS: AWS re:Invent 2019: Introducing The Amazon Builders’ Library (DOP328)](https://youtu.be/sKRdemSirDM?t=704&ref=wellarchitected)  
 [AWS OpsWorks: Using Auto Healing to Replace Failed Instances](https://docs.aws.amazon.com/opsworks/latest/userguide/workinginstances-autohealing.html?ref=wellarchitected)  
 [What Is Amazon EventBridge?](https://docs.aws.amazon.com/eventbridge/latest/userguide/what-is-amazon-eventbridge.html?ref=wellarchitected)  
 [Amazon Route 53: Choosing a Routing Policy](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html?ref=wellarchitected)  
 [What Is AWS Global Accelerator?](https://docs.aws.amazon.com/global-accelerator/latest/dg/what-is-global-accelerator.html?ref=wellarchitected)  
 [The Amazon Builders' Library: Static stability using Availability Zones](https://aws.amazon.com/builders-library/static-stability-using-availability-zones?ref=wellarchitected)  
 [The Amazon Builders' Library: Implementing health checks](https://aws.amazon.com/builders-library/implementing-health-checks/?ref=wellarchitected)  
 [Well-Architected lab: Level 300: Implementing Health Checks and Managing Dependencies to Improve Reliability](https://wellarchitectedlabs.com/Reliability/300_Health_Checks_and_Dependencies/README.html?ref=wellarchitected)  
 [The Berkeley/Stanford Recovery-Oriented Computing (ROC) Project](http://roc.cs.berkeley.edu/?ref=wellarchitected)  
 [Multiple data center HA network connectivity](https://aws.amazon.com/answers/networking/aws-multiple-data-center-ha-network-connectivity/?ref=wellarchitected)  
 [AWS Marketplace: products that can be used for fault tolerance](https://aws.amazon.com/marketplace/search/results?searchTerms=fault+tolerance&ref=wellarchitected)  
 [APN Partner: partners that can help with automation of your fault tolerance](https://aws.amazon.com/partners/find/results/?keyword=automation&ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

#### Monitor all components of the workload to detect failures Continuously monitor the health of your [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) so that you and your automated systems are aware of degradation or complete failure as soon as they occur. Monitor for key [performance](https://wa.aws.amazon.com/wat.pillar.performance.en.html) indicators (KPIs) based on business value.

#### Fail over to healthy resources Ensure that if a resource failure occurs, that healthy resources can continue to serve requests. For location failures (such as [Availability Zone](https://wa.aws.amazon.com/wat.concept.az.en.html) or [AWS Region](https://wa.aws.amazon.com/wat.concept.region.en.html)) ensure you have systems in place to fail over to healthy resources in unimpaired locations.

#### Automate healing on all layers Upon detection of a failure, use automated capabilities to perform actions to remediate.

#### Rely on the data plane and not the control plane during recovery The control plane is used to configure resources, and the data plane delivers services. Data planes typically have higher [availability](https://wa.aws.amazon.com/wat.concept.availability.en.html) design goals than control planes and are usually less complex. When implementing recovery or mitigation responses to potentially [resiliency](https://wa.aws.amazon.com/wat.concept.resiliency.en.html)-impacting [events](https://wa.aws.amazon.com/wat.concept.event.en.html), using control plane [operations](https://wa.aws.amazon.com/wat.pillar.operationalExcellence.en.html) can lower the overall [resiliency](https://wa.aws.amazon.com/wat.concept.resiliency.en.html) of your [architecture](https://wa.aws.amazon.com/wat.concept.architecture.en.html). For example, you can rely on the Amazon Route 53 data plane to reliably route DNS queries based on health checks, but updating Route 53 routing policies uses the control plane, so do not rely on it for recovery.

#### Use static stability to prevent bimodal behavior Bimodal behavior is when your [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) exhibits different behavior under normal and failure modes, for example, relying on launching new instances if an [Availability Zone](https://wa.aws.amazon.com/wat.concept.az.en.html) fails. You should instead build [workloads](https://wa.aws.amazon.com/wat.concept.workload.en.html) that are statically stable and operate in only one mode. In this case, provision enough instances in each [Availability Zone](https://wa.aws.amazon.com/wat.concept.az.en.html) to handle the [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) load if one [AZ](https://wa.aws.amazon.com/wat.concept.az.en.html) were removed and then use [Elastic Load Balancing](https://wa.aws.amazon.com/wat.concept.elb.en.html) or Amazon Route 53 health checks to shift load away from the impaired instances.

#### Send notifications when events impact availability Notifications are sent upon the detection of significant [events](https://wa.aws.amazon.com/wat.concept.event.en.html), even if the issue caused by the [event](https://wa.aws.amazon.com/wat.concept.event.en.html) was automatically resolved.

**REL 12. How do you test reliability?**

After you have designed your workload to be resilient to the stresses of production, testing is the only way to ensure that it will operate as designed, and deliver the resiliency you expect.

[Well-Architected lab: Level 300: Testing for Resiliency of EC2 RDS and S3](https://wellarchitectedlabs.com/reliability/300_labs/300_testing_for_resiliency_of_ec2_rds_and_s3/?ref=wellarchitected)  
 [What is AWS Fault Injection Simulator (AWS FIS)?](https://docs.aws.amazon.com/fis/latest/userguide/what-is.html?ref=wellarchitected)  
 [Resilience Engineering: Learning to Embrace Failure](https://queue.acm.org/detail.cfm?id=2371297&ref=wellarchitected)  
 [AWS re:Invent 2019: Improving resiliency with chaos engineering (DOP309-R1)](https://youtu.be/ztiPjey2rfY?ref=wellarchitected)  
 [Continuous Delivery and Continuous Integration](https://docs.aws.amazon.com/codepipeline/latest/userguide/concepts-continuous-delivery-integration.html?ref=wellarchitected)  
 [Using Canaries (Amazon CloudWatch Synthetics)](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries.html?ref=wellarchitected)  
 [Use CodePipeline with AWS CodeBuild to test code and run builds](https://docs.aws.amazon.com/codebuild/latest/userguide/how-to-create-pipeline.html?ref=wellarchitected)  
 [Automate your operational playbooks with AWS Systems Manager](https://aws.amazon.com/about-aws/whats-new/2019/11/automate-your-operational-playbooks-with-aws-systems-manager/?ref=wellarchitected)  
 [Principles of Chaos Engineering](https://principlesofchaos.org/?ref=wellarchitected)  
 [Apache JMeter](https://github.com/apache/jmeter?ref=wellarchitected&ref=wellarchitected)  
 [Casey Rosenthal, Nora Jones. “Chaos Engineering” (August 2020)](https://www.oreilly.com/library/view/chaos-engineering/9781492043850/?ref=wellarchitected)  
 [AWS Marketplace: products that can be used for continuous integration](https://aws.amazon.com/marketplace/search/results?searchTerms=Continuous+integration&ref=wellarchitected)  
 [APN Partner: partners that can help with implementation of a continuous integration pipeline](https://aws.amazon.com/partners/find/results/?keyword=Continuous+Integration&ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

#### Use playbooks to investigate failures Enable consistent and prompt responses to failure scenarios that are not well understood, by documenting the investigation process in [playbooks](https://wa.aws.amazon.com/wat.concept.playbook.en.html). [Playbooks](https://wa.aws.amazon.com/wat.concept.playbook.en.html) are the predefined steps performed to identify the factors contributing to a failure scenario. The results from any process step are used to determine the next steps to take until the issue is identified or escalated.

#### Perform post-incident analysis Review customer-impacting [events](https://wa.aws.amazon.com/wat.concept.event.en.html), and identify the contributing factors and preventative action items. Use this information to develop mitigations to limit or prevent recurrence. Develop procedures for prompt and effective responses. Communicate contributing factors and corrective actions as appropriate, tailored to target audiences. Have a method to communicate these causes to others as needed.

#### Test functional requirements These include unit tests and integration tests that validate required functionality.

#### Test scaling and performance requirements This includes load testing to validate that the [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) meets scaling and [performance](https://wa.aws.amazon.com/wat.pillar.performance.en.html) requirements.

#### Test resiliency using chaos engineering Run tests that inject failures regularly into pre-production and production environments. Hypothesize how your [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) will react to the failure, then compare your hypothesis to the testing results and iterate if they do not match. Ensure that production testing does not impact users.

#### Conduct game days regularly Use [game days](https://wa.aws.amazon.com/wat.concept.gameday.en.html) to regularly exercise your procedures for responding to [events](https://wa.aws.amazon.com/wat.concept.event.en.html) and failures as close to production as possible (including in production environments) with the people who will be involved in actual failure scenarios. [Game days](https://wa.aws.amazon.com/wat.concept.gameday.en.html) enforce measures to ensure that production [events](https://wa.aws.amazon.com/wat.concept.event.en.html) do not impact users.

**REL 13. How do you plan for disaster recovery (DR)?**

Having backups and redundant workload components in place is the start of your DR strategy. RTO and RPO are your objectives for restoration of your workload. Set these based on business needs. Implement a strategy to meet these objectives, considering locations and function of workload resources and data. The probability of disruption and cost of recovery are also key factors that help to inform the business value of providing disaster recovery for a workload.

[AWS re:Invent 2018: Architecture Patterns for Multi-Region Active-Active Applications (ARC209-R2)](https://youtu.be/2e29I3dA8o4?ref=wellarchitected)  
 [AWS re:Invent 2019: Backup-and-restore and disaster-recovery solutions with AWS (STG208)](https://youtu.be/7gNXfo5HZN8?ref=wellarchitected)  
 [Disaster Recovery of Workloads on AWS: Recovery in the Cloud (AWS Whitepaper)](https://docs.aws.amazon.com/whitepapers/latest/disaster-recovery-workloads-on-aws/disaster-recovery-workloads-on-aws.html?ref=wellarchitected)  
 [AWS Architecture Blog: Disaster Recovery Series](https://aws.amazon.com/blogs/architecture/tag/disaster-recovery-series/?ref=wellarchitected)  
 [What Is AWS Backup?](https://docs.aws.amazon.com/aws-backup/latest/devguide/whatisbackup.html?ref=wellarchitected)  
 [What is Route 53 Application Recovery Controller?](https://docs.aws.amazon.com/r53recovery/latest/dg/what-is-route53-recovery.html?ref=wellarchitected)  
 [Remediating Noncompliant AWS Resources by AWS Config Rules](https://docs.aws.amazon.com/config/latest/developerguide/remediation.html?ref=wellarchitected)  
 [AWS Systems Manager Automation](https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-automation.html?ref=wellarchitected)  
 [AWS CloudFormation: Detect Drift on an Entire CloudFormation Stack](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/detect-drift-stack.html?ref=wellarchitected)  
 [Amazon RDS: Cross-region backup copy](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CopySnapshot.html?ref=wellarchitected)  
 [RDS: Replicating a Read Replica Across Regions](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html?ref=wellarchitected#USER_ReadRepl.XRgn)  
 [S3: Cross-Region Replication](http://docs.aws.amazon.com/AmazonS3/latest/dev/crr.html?ref=wellarchitected)  
 [Route 53: Configuring DNS Failover](http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-configuring.html?ref=wellarchitected)  
 [CloudEndure Disaster Recovery](https://aws.amazon.com/cloudendure-disaster-recovery/?ref=wellarchitected)  
 [How do I implement an Infrastructure Configuration Management solution on AWS?](https://aws.amazon.com/answers/configuration-management/aws-infrastructure-configuration-management/?ref=wellarchitected&ref=wellarchitected)  
 [CloudEndure Disaster Recovery to AWS](https://aws.amazon.com/marketplace/pp/B07XQNF22L?ref=wellarchitected)  
 [AWS Marketplace: products that can be used for disaster recovery](https://aws.amazon.com/marketplace/search/results?searchTerms=Disaster+recovery&ref=wellarchitected)  
 [APN Partner: partners that can help with disaster recovery](https://aws.amazon.com/partners/find/results/?keyword=Disaster+Recovery&ref=wellarchitected)

Afbeelding met tekst, schermafbeelding, computer, computer

Automatisch gegenereerde beschrijving

#### Define recovery objectives for downtime and data loss The [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) has a recovery time objective (RTO) and recovery point objective (RPO).

#### Use defined recovery strategies to meet the recovery objectives A disaster recovery (DR) strategy has been defined to meet your [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) objectives. Choose a strategy such as: backup and restore; standby (active/passive); or active/active.

#### Test disaster recovery implementation to validate the implementation Regularly test failover to DR to ensure that RTO and RPO are met.

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

#### Automate recovery Use AWS or third-party tools to automate system recovery and route traffic to the DR site or Region.