AWS Well-Architected Framework

July 2019

This document describes the AWS Well-Architected Framework, which enables you to review and improve your cloud-based architectures and better understand the business impact of your design decisions. We address general design principles as well as specific best practices and guidance in five conceptual areas that we define as the *pillars* of the Well-Architected Framework.

Notices

Introduction

The AWS Well-Architected Framework helps you understand the pros and cons of decisions you make while building systems on AWS. By using the Framework you will learn architectural best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud. It provides a way for you to consistently measure your architectures against best practices and identify areas for improvement. The process for reviewing an architecture is a constructive conversation about architectural decisions, and is not an audit mechanism. We believe that having well-architected systems greatly increases the likelihood of business success.

AWS Solutions Architects have years of experience architecting solutions across a wide variety of business verticals and use cases. We have helped design and review thousands of customers' architectures on AWS. From this experience, we have identified best practices and core strategies for architecting systems in the cloud.

The AWS Well-Architected Framework documents a set of foundational questions that allow you to understand if a specific architecture aligns well with cloud best practices. The framework provides a consistent approach to evaluating systems against the qualities you expect from modern cloud-based systems, and the remediation that would be required to achieve those qualities. As AWS continues to evolve, and we continue to learn more from working with our customers, we will continue to refine the definition of well-architected.

This framework is intended for those in technology roles, such as chief technology officers (CTOs), architects, developers, and operations team members. It describes AWS best practices and strategies to use when designing and operating a cloud workload, and provides links to further implementation details and architectural patterns. For more information, see the AWS Well-Architected homepage.

AWS also provides a service for reviewing your workloads at no charge. The AWS Well-Architected Tool (AWS WA Tool) is a service in the cloud that provides a consistent process for you to review and measure your architecture using the AWS Well-Architected Framework. The AWS WA Tool provides recommendations for making your workloads more reliable, secure, efficient, and cost-effective.

To help you apply best practices, we have created AWS Well-Architected Labs, which provides you with a repository of code and documentation to give you hands-on experience implementing best practices. We also have teamed up with select AWS Partner Network (APN) Partners, who are members of the AWS Well-Architected Partner program. These APN Partners have deep AWS knowledge, and can help you review and improve your workloads.

Definitions

Every day experts at AWS assist customers in architecting systems to take advantage of best practices in the cloud. We work with you on making architectural trade-offs as your designs evolve. As you deploy these systems into live environments, we learn how well these systems perform and the consequences of those trade-offs.

Based on what we have learned we have created the AWS Well-Architected Framework, which provides a consistent set of best practices for customers and partners to evaluate architectures, and provides a set of questions you can use to evaluate how well an architecture is aligned to AWS best practices.

The AWS Well-Architected Framework is based on five pillars — operational excellence, security, reliability, performance efficiency, and cost optimization.

Table 1. The pillars of the AWS Well-Architected Framework

Name	Description
Operational Excellence	The ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.
Security	The ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.
Reliability	The ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues.
Performance Efficiency	The ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve.
Cost Optimization	The ability to run systems to deliver business value at the lowest price point.

In the AWS Well-Architected Framework we use these terms

- A **component** is the code, configuration and AWS Resources that together deliver against a requirement. A component is often the unit of technical ownership, and is decoupled from other components.
- We use the term **workload** to identify a set of components that together deliver business value. The workload is usually the level of detail that business and technology leaders communicate about.

- **Milestones** mark key changes in your architecture as it evolves throughout the product lifecycle (design, testing, go live, and in production).
- We think about **architecture** as being how components work together in a workload. How components communicate and interact is often the focus of architecture diagrams.
- Within an organization the **technology portfolio** is the collection of workloads that are required for the business to operate.

When architecting workloads you make trade-offs between pillars based upon your business context. These business decisions can drive your engineering priorities. You might optimize to reduce cost at the expense of reliability in development environments, or, for mission-critical solutions, you might optimize reliability with increased costs. In ecommerce solutions, performance can affect revenue and customer propensity to buy. Security and operational excellence are generally not traded-off against the other pillars.

On Architecture

In on-premises environments customers often have a central team for technology architecture that acts as an overlay to other product or feature teams to ensure they are following best practice. Technology architecture teams are often composed of a set of roles such as Technical Architect (infrastructure), Solutions Architect (software), Data Architect, Networking Architect, and Security Architect. Often these teams use TOGAF or the Zachman Framework as part of an enterprise architecture capability.

At AWS, we prefer to distribute capabilities into teams rather than having a centralized team with that capability. There are risks when you choose to distribute decision making authority, for example, ensuring that teams are meeting internal standards. We mitigate these risks in two ways. First, we have *practices* [1] that focus on enabling each team to have that capability, and we put in place experts who ensure that teams raise the bar on the standards they need to meet. Second, we implement *mechanisms* [2] that carry out automated checks to ensure standards are being met. This distributed approach is supported by the Amazon leadership principles, and establishes a culture across all roles that *works back* [3] from the customer. Customer-obsessed teams build products in response to a customer need.

For architecture this means that we expect every team to have the capability to create architectures and to follow best practices. To help new teams gain these capabilities

[1] Ways of doing things, process, standards, and accepted norms.
[2] "Good intentions never work, you need good mechanisms to make anything happen" Jeff Bezos. This means replacing humans best efforts with mechanisms (often automated) that check for compliance with rules or process.
[3] Working backward is a fundamental part of our innovation process. We start with the customer and what they want, and let that define and guide our efforts.

or existing teams to raise their bar, we enable access to a virtual community of principal engineers who can review their designs and help them understand what AWS best practices are. The principal engineering community works to make best practices visible and accessible. One way they do this, for example, is through lunchtime talks that focus on applying best practices to real examples. These talks are recorded and can be used as part of onboarding materials for new team members.

AWS best practices emerge from our experience running thousands of systems at internet scale. We prefer to use data to define best practice, but we also use subject matter experts like principal engineers to set them. As principal engineers see new best practices emerge they work as a community to ensure that teams follow them. In time, these best practices are formalized into our internal review processes, as well as into mechanisms that enforce compliance. Well-Architected is the customer-facing implementation of our internal review process, where we have codified our principal engineering thinking across field roles like Solutions Architecture and internal engineering teams. Well-Architected is a scalable mechanism that lets you take advantage of these learnings.

By following the approach of a principal engineering community with distributed ownership of architecture, we believe that a Well-Architected enterprise architecture can emerge that is driven by customer need. Technology leaders (such as a CTOs or development managers), carrying out Well-Architected reviews across all your workloads will allow you to better understand the risks in your technology portfolio. Using this approach you can identify themes across teams that your organization could address by mechanisms, trainings, or lunchtime talks where your principal engineers can share their thinking on specific areas with multiple teams.

General Design Principles

The Well-Architected Framework identifies a set of general design principles to facilitate good design in the cloud:

- **Stop guessing your capacity needs**: Eliminate guessing about your infrastructure capacity needs. When you make a capacity decision before you deploy a system, you might end up sitting on expensive idle resources or dealing with the performance implications of limited capacity. With cloud computing, these problems can go away. You can use as much or as little capacity as you need, and scale up and down automatically.
- **Test systems at production scale**: In the cloud, you can create a production-scale test environment on demand, complete your testing, and then decommission the resources. Because you only pay for the test environment when it's running, you can simulate your live environment for a fraction of the cost of testing on premises.
- **Automate to make architectural experimentation easier**: Automation allows you to create and replicate your systems at low cost and avoid the expense of manual effort. You can track changes to your automation, audit the impact, and revert to previous parameters when necessary.
- **Allow for evolutionary architectures**: Allow for evolutionary architectures. In a traditional environment, architectural decisions are often implemented as static, one-time events, with a few major versions of a system during its lifetime. As a business and its context continue to change, these initial decisions might hinder the system's ability to deliver changing business requirements. In the cloud, the capability to automate and test on demand lowers the risk of impact from design changes. This allows systems to evolve over time so that businesses can take advantage of innovations as a standard practice.
- **Drive architectures using data**: In the cloud you can collect data on how your architectural choices affect the behavior of your workload. This lets you make fact-based decisions on how to improve your workload. Your cloud infrastructure is code, so you can use that data to inform your architecture choices and improvements over time.
- **Improve through game days**: Test how your architecture and processes perform by regularly scheduling game days to simulate events in production. This will help you understand where improvements can be made and can help develop organizational experience in dealing with events.

The Five Pillars of the Framework

Creating a software system is a lot like constructing a building. If the foundation is not solid structural problems can undermine the integrity and function of the building. When architecting technology solutions, if you neglect the five pillars of operational excellence, security, reliability, performance efficiency, and cost optimization it can become challenging to build a system that delivers on your expectations and requirements. Incorporating these pillars into your architecture will help you produce stable and efficient systems. This will allow you to focus on the other aspects of design, such as functional requirements.

Operational Excellence

The **Operational Excellence** pillar includes the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.

The operational excellence pillar provides an overview of design principles, best practices, and questions. You can find prescriptive guidance on implementation in the Operational Excellence Pillar whitepaper.

Design Principles

There are six design principles for operational excellence in the cloud:

- **Perform operations as code**: In the cloud, you can apply the same engineering discipline that you use for application code to your entire environment. You can define your entire workload (applications, infrastructure) as code and update it with code. You can implement your operations procedures as code and automate their execution by triggering them in response to events. By performing operations as code, you limit human error and enable consistent responses to events.
- **Annotate documentation**: In an on-premises environment, documentation is created by hand, used by people, and hard to keep in sync with the pace of change. In the cloud, you can automate the creation of annotated documentation after every build (or automatically annotate hand-crafted documentation). Annotated documentation can be used by people and systems. Use annotations as an input to your operations code.
- **Make frequent, small, reversible changes**: Design workloads to allow components to be updated regularly. Make changes in small increments that can be reversed if they fail (without affecting customers when possible).
- **Refine operations procedures frequently**: As you use operations procedures, look for opportunities to improve them. As you evolve your workload, evolve your

procedures appropriately. Set up regular game days to review and validate that all procedures are effective and that teams are familiar with them.

- **Anticipate failure**: Perform "pre-mortem" exercises to identify potential sources of failure so that they can be removed or mitigated. Test your failure scenarios and validate your understanding of their impact. Test your response procedures to ensure that they are effective, and that teams are familiar with their execution. Set up regular game days to test workloads and team responses to simulated events.
- **Learn from all operational failures**: Drive improvement through lessons learned from all operational events and failures. Share what is learned across teams and through the entire organization.

Definition

There are three best practice areas for operational excellence in the cloud:

- **Prepare**
- **Operate**
- **Evolve**

Operations teams need to understand their business and customer needs so they can effectively and efficiently support business outcomes. Operations creates and uses procedures to respond to operational events and validates their effectiveness to support business needs. Operations collects metrics that are used to measure the achievement of desired business outcomes. Everything continues to change—your business context, business priorities, customer needs, etc. It's important to design operations to support evolution over time in response to change and to incorporate lessons learned through their performance.

Best Practices

Prepare

Effective preparation is required to drive operational excellence. Business success is enabled by shared goals and understanding across the business, development, and operations. Common standards simplify workload design and management, enabling operational success. Design workloads with mechanisms to monitor and gain insight into application, platform, and infrastructure components, as well as customer experience and behavior.

Create mechanisms to validate that workloads, or changes, are ready to be moved into production and supported by operations. Operational readiness is validated through checklists to ensure a workload meets defined standards and that required procedures are adequately captured in runbooks and playbooks. Validate that there are sufficient trained personnel to effectively support the workload. Prior to transition, test responses to operational events and failures. Practice responses in supported environments through failure injection and game day events.

AWS enables operations as code in the cloud and the ability to safely experiment, develop operations procedures, and practice failure. Using AWS CloudFormation enables you to have consistent, templated, sandbox development, test, and production environments with increasing levels of operations control. AWS enables visibility into your workloads at all layers through various log collection and monitoring features. Data on use of resources, application programming interfaces (APIs), and network flow logs can be collected using Amazon CloudWatch, AWS CloudTrail, and VPC Flow Logs. You can use the collectd plugin, or the CloudWatch Logs agent, to aggregate information about the operating system into CloudWatch.

The following questions focus on these considerations for operational excellence. (For a list of operational excellence questions, answers, and best practices, see the Appendix.)

OPS 1: How do you determine what your priorities are?
Everyone needs to understand their part in enabling business success. Have shared goals in order to set priorities for resources. This will maximize the benefits of your efforts.
OPS 2: How do you design your workload so that you can understand its state?
Design your workload so that it provides the information necessary for you to understand its internal state (for example, metrics, logs, and traces) across all components. This enables you to provide effective responses when appropriate.
OPS 3: How do you reduce defects, ease remediation, and improve flow into production?
Adopt approaches that improve flow of changes into production, that enable refactoring, fast feedback on quality, and bug fixing. These accelerate beneficial changes entering production, limit issues deployed, and enable rapid identification and remediation of issues introduced through deployment activities.
OPS 4: How do you mitigate deployment risks?
Adopt approaches that provide fast feedback on quality and enable rapid recovery from changes that do not have desired outcomes. Using these practices mitigates the impact of issues introduced through the deployment of changes.
OPS 5: How do you know that you are ready to support a workload?
Evaluate the operational readiness of your workload, processes and procedures, and personnel to understand the operational risks related to your workload.

Implement the minimum number of architecture standards for your workloads. Balance the cost to implement a standard against the benefit to the workload and the burden upon operations. Reduce the number of supported standards to reduce the chance that lower-than-acceptable standards will be applied by error. Operations personnel are often constrained resources.

Invest in implementing operations activities as code to maximize the productivity of operations personnel, minimize error rates, and enable automated responses. Adopt deployment practices that take advantage of the elasticity of the cloud to facilitate pre-deployment of systems for faster implementations.

Operate

Successful operation of a workload is measured by the achievement of business and customer outcomes. Define expected outcomes, determine how success will be measured, and identify the workload and operations metrics that will be used in those calculations to determine if operations are successful. Consider that operational health includes both the health of the workload and the health and success of the operations acting upon the workload (for example, deployment and incident response). Establish baselines from which improvement or degradation of operations will be identified, collect and analyze your metrics, and then validate your understanding of operations success and how it changes over time. Use collected metrics to determine if you are satisfying customer and business needs, and identify areas for improvement.

Efficient and effective management of operational events is required to achieve operational excellence. This applies to both planned and unplanned operational events. Use established runbooks for well-understood events, and use playbooks to aid in the resolution of other events. Prioritize responses to events based on their business and customer impact. Ensure that if an alert is raised in response to an event, there is an associated process to be executed, with a specifically identified owner. Define in advance the personnel required to resolve an event and include escalation triggers to engage additional personnel, as it becomes necessary, based on impact (that is, duration, scale, and scope). Identify and engage individuals with the authority to decide on courses of action where there will be a business impact from an event response not previously addressed.

Communicate the operational status of workloads through dashboards and notifications that are tailored to the target audience (for example, customer, business, developers, operations) so that they may take appropriate action, so that their expectations are managed, and so that they are informed when normal operations resume.

Determine the root cause of unplanned events and unexpected impacts from planned events. This information will be used to update your procedures to mitigate future occurrence of events. Communicate root cause with affected communities as appropriate.

In AWS, you can generate dashboard views of your metrics collected from workloads and natively from AWS. You can leverage CloudWatch or third-party applications to aggregate and present business, workload, and operations level views of operations activities. AWS provides workload insights through logging capabilities including

AWS X-Ray, CloudWatch, CloudTrail, and VPC Flow Logs enabling the identification of workload issues in support of root cause analysis and remediation.

The following questions focus on these considerations for operational excellence.

OPS 6: How do you understand the health of your workload?
Define, capture, and analyze workload metrics to gain visibility to workload events so that you can take appropriate action.
OPS 7: How do you understand the health of your operations?
Define, capture, and analyze operations metrics to gain visibility to operations events so that you can take appropriate action.
OPS 8: How do you manage workload and operations events?
Prepare and validate procedures for responding to events to minimize their disruption to your workload.

Routine operations, as well as responses to unplanned events, should be automated. Manual processes for deployments, release management, changes, and rollbacks should be avoided. Releases should not be large batches that are done infrequently. Rollbacks are more difficult in large changes. Failing to have a rollback plan, or the ability to mitigate failure impacts, will prevent continuity of operations. Align metrics to business needs so that responses are effective at maintaining business continuity. One-time decentralized metrics with manual responses will result in greater disruption to operations during unplanned events.

Evolve

Evolution of operations is required to sustain operational excellence. Dedicate work cycles to making continuous incremental improvements. Regularly evaluate and prioritize opportunities for improvement (for example, feature requests, issue remediation, and compliance requirements), including both the workload and operations procedures. Include feedback loops within your procedures to rapidly identify areas for improvement and capture learnings from the execution of operations.

Share lessons learned across teams to share the benefits of those lessons. Analyze trends within lessons learned and perform cross-team retrospective analysis of operations metrics to identify opportunities and methods for improvement. Implement changes intended to bring about improvement and evaluate the results to determine success.

With AWS Developer Tools you can implement continuous delivery build, test, and deployment activities that work with a variety of source code, build, testing, and deployment tools from AWS and third parties. The results of deployment activities can be used to identify opportunities for improvement for both deployment and development. You can perform analytics on your metrics data integrating data from your operations and deployment activities, to enable analysis of the impact of those activities against business and customer outcomes. This data can be leveraged in cross-team retrospective analysis to identify opportunities and methods for improvement.

The following questions focus on these considerations for operational excellence.

OPS 9: How do you evolve operations?
Dedicate time and resources for continuous incremental improvement to evolve the effectiveness and efficiency of your operations.

Successful evolution of operations is founded in: frequent small improvements; providing safe environments and time to experiment, develop, and test improvements; and environments in which learning from failures is encouraged. Operations support for sandbox, development, test, and production environments, with increasing level of operational controls, facilitates development and increases the predictability of successful results from changes deployed into production.

Key AWS Services

The AWS service that is essential to Operational Excellence is **AWS CloudFormation**, which you can use to create templates based on best practices. This enables you to provision resources in an orderly and consistent fashion from your development through production environments. The following services and features support the three areas in operational excellence:

- **Prepare**: AWS Config and AWS Config rules can be used to create standards for workloads and to determine if environments are compliant with those standards before being put into production.
- **Operate**: Amazon CloudWatch allows you to monitor the operational health of a workload.
- **Evolve**: Amazon Elasticsearch Service (Amazon ES) allows you to analyze your log data to gain actionable insights quickly and securely.

Resources

Refer to the following resources to learn more about our best practices for Operational Excellence.

Documentation

- DevOps and AWS

Whitepaper

- Operational Excellence Pillar

Video

- DevOps at Amazon

Security

The **Security** pillar includes the ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.

The security pillar provides an overview of design principles, best practices, and questions. You can find prescriptive guidance on implementation in the Security Pillar whitepaper.

Design Principles

There are seven design principles for security in the cloud:

- **Implement a strong identity foundation**: Implement the principle of least privilege and enforce separation of duties with appropriate authorization for each interaction with your AWS resources. Centralize privilege management and reduce or even eliminate reliance on long-term credentials.
- **Enable traceability**: Monitor, alert, and audit actions and changes to your environment in real time. Integrate logs and metrics with systems to automatically respond and take action.
- **Apply security at all layers**: Rather than just focusing on protection of a single outer layer, apply a defense-in-depth approach with other security controls. Apply to all layers (e.g., edge network, VPC, subnet, load balancer, every instance, operating system, and application).
- **Automate security best practices**: Automated software-based security mechanisms improve your ability to securely scale more rapidly and cost effectively. Create secure architectures, including the implementation of controls that are defined and managed as code in version-controlled templates.
- **Protect data in transit and at rest**: Classify your data into sensitivity levels and use mechanisms, such as encryption, tokenization, and access control where appropriate.
- **Keep people away from data**: Create mechanisms and tools to reduce or eliminate the need for direct access or manual processing of data. This reduces the risk of loss or modification and human error when handling sensitive data.

- **Prepare for security events**: Prepare for an incident by having an incident management process that aligns to your organizational requirements. Run incident response simulations and use tools with automation to increase your speed for detection, investigation, and recovery.

Definition

There are five best practice areas for security in the cloud:

- **Identity and Access Management**
- **Detective Controls**
- **Infrastructure Protection**
- **Data Protection**
- **Incident Response**

Before you architect any system, you need to put in place practices that influence security. You will want to control who can do what. In addition, you want to be able to identify security incidents, protect your systems and services, and maintain the confidentiality and integrity of data through data protection. You should have a well-defined and practiced process for responding to security incidents. These tools and techniques are important because they support objectives such as preventing financial loss or complying with regulatory obligations.

The AWS Shared Responsibility Model enables organizations that adopt the cloud to achieve their security and compliance goals. Because AWS physically secures the infrastructure that supports our cloud services, as an AWS customer you can focus on using services to accomplish your goals. The AWS Cloud also provides greater access to security data and an automated approach to responding to security events.

Best Practices

Identity and Access Management

Identity and access management are key parts of an information security program, ensuring that only authorized and authenticated users are able to access your resources, and only in a manner that you intend. For example, you should define principals (that is, users, groups, services, and roles that take action in your account), build out policies aligned with these principals, and implement strong credential management. These privilege-management elements form the core of authentication and authorization.

In AWS, privilege management is primarily supported by the AWS Identity and Access Management (IAM) service, which allows you to control user and programmatic access to AWS services and resources. You should apply granular policies, which assign permissions to a user, group, role, or resource. You also have the ability to require strong password practices, such as complexity level, avoiding re-use, and enforcing multi-factor authentication (MFA). You can use federation with your existing directory service. For workloads that require systems to have access to AWS, IAM enables secure access through roles, instance profiles, identity federation, and temporary credentials.

The following questions focus on these considerations for security. (For a list of security questions, answers, and best practices, see the Appendix.)

SEC 1: How do you manage credentials and authentication?
Credentials and authentication mechanisms include passwords, tokens, and keys that grant access directly or indirectly in your workload. Protect credentials with appropriate mechanisms to help reduce the risk of accidental or malicious use.
SEC 2: How do you control human access?
Control human access by implementing controls inline with defined business requirements to reduce risk and lower the impact of unauthorized access. This applies to privileged users and administrators of your AWS account, and also applies to end users of your application
SEC 3: How do you control programmatic access?
Control programmatic or automated access with appropriately defined, limited, and segregated access to help reduce the risk of unauthorized access. Programmatic access includes access that is internal to your workload, and access to AWS related resources.

Credentials must not be shared between any user or system. User access should be granted using a least-privilege approach with best practices including password requirements and MFA enforced. Programmatic access including API calls to AWS services should be performed using temporary and limited-privilege credentials such as those issued by the AWS Security Token Service.

AWS provides resources that can help you with Identity and access management. To help learn best practices, explore our hands-on labs on managing credentials & authentication, controlling human access, and controlling programmatic access.

Detective Controls

You can use detective controls to identify a potential security threat or incident. They are an essential part of governance frameworks and can be used to support a quality process, a legal or compliance obligation, and for threat identification and response efforts. There are different types of detective controls. For example, conducting an inventory of assets and their detailed attributes promotes more effective decision making (and lifecycle controls) to help establish operational baselines. You can also use internal auditing, an examination of controls related to information systems, to ensure that practices meet policies and requirements and that you have set the correct automated alerting notifications based on defined conditions. These controls are important reactive factors that can help your organization identify and understand the scope of anomalous activity.

In AWS, you can implement detective controls by processing logs, events, and monitoring that allows for auditing, automated analysis, and alarming. CloudTrail logs, AWS API calls, and CloudWatch provide monitoring of metrics with alarming, and AWS Config provides configuration history. Amazon GuardDuty is a managed threat detection service that continuously monitors for malicious or unauthorized behavior to help you protect your AWS accounts and workloads. Service-level logs are also available, for example, you can use Amazon Simple Storage Service (Amazon S3) to log access requests.

The following questions focus on these considerations for security.

SEC 4: How do you detect and investigate security events?
Capture and analyze events from logs and metrics to gain visibility. Take action on security events and potential threats to help secure your workload.
SEC 5: How do you defend against emerging security threats?
Staying up to date with AWS and industry best practices and threat intelligence helps you be aware of new risks. This enables you to create a threat model to identify, prioritize, and implement appropriate controls to help protect your workload.

Log management is important to a well-architected design for reasons ranging from security or forensics to regulatory or legal requirements. It is critical that you analyze logs and respond to them so that you can identify potential security incidents. AWS provides functionality that makes log management easier to implement by giving you the ability to define a data-retention lifecycle or define where data will be preserved, archived, or eventually deleted. This makes predictable and reliable data handling simpler and more cost effective.

Infrastructure Protection

Infrastructure protection encompasses control methodologies, such as defense in depth, necessary to meet best practices and organizational or regulatory obligations. Use of these methodologies is critical for successful, ongoing operations in either the cloud or on-premises.

In AWS, you can implement stateful and stateless packet inspection, either by using AWS-native technologies or by using partner products and services available through the AWS Marketplace. You should use Amazon Virtual Private Cloud (Amazon VPC) to create a private, secured, and scalable environment in which you can define your topology—including gateways, routing tables, and public and private subnets.

The following questions focus on these considerations for security.

SEC 6: How do you protect your networks?
Public and private networks require multiple layers of defense to help protect from external and internal network-based threats.
SEC 7: How do you protect your compute resources?
Compute resources in your workload require multiple layers of defense to help protect from external and internal threats. Compute resources include EC2 instances, containers, AWS Lambda functions, database services, IoT devices, and more.

Multiple layers of defense are advisable in any type of environment. In the case of infrastructure protection, many of the concepts and methods are valid across cloud and on-premises models. Enforcing boundary protection, monitoring points of ingress and egress, and comprehensive logging, monitoring, and alerting are all essential to an effective information security plan.

AWS customers are able to tailor, or harden, the configuration of an Amazon Elastic Compute Cloud (Amazon EC2), Amazon EC2 Container Service (Amazon ECS) container, or AWS Elastic Beanstalk instance, and persist this configuration to an immutable Amazon Machine Image (AMI). Then, whether triggered by Auto Scaling or launched manually, all new virtual servers (instances) launched with this AMI receive the hardened configuration.

Data Protection

Before architecting any system, foundational practices that influence security should be in place. For example, data classification provides a way to categorize organizational data based on levels of sensitivity, and encryption protects data by way of rendering it unintelligible to unauthorized access. These tools and techniques are important because they support objectives such as preventing financial loss or complying with regulatory obligations.

In AWS, the following practices facilitate protection of data:

- As an AWS customer you maintain full control over your data.
- AWS makes it easier for you to encrypt your data and manage keys, including regular key rotation, which can be easily automated by AWS or maintained by you.
- Detailed logging that contains important content, such as file access and changes, is available.
- AWS has designed storage systems for exceptional resiliency. For example, Amazon S3 Standard, S3 Standard–IA, S3 One Zone-IA, and Amazon Glacier are all designed to provide 99.999999999% durability of objects over a given year. This durability level corresponds to an average annual expected loss of 0.000000001% of objects.

- Versioning, which can be part of a larger data lifecycle management process, can protect against accidental overwrites, deletes, and similar harm.
- AWS never initiates the movement of data between Regions. Content placed in a Region will remain in that Region unless you explicitly enable a feature or leverage a service that provides that functionality.

The following questions focus on these considerations for security.

SEC 8: How do you classify your data?
Classification provides a way to categorize data, based on levels of sensitivity, to help you determine appropriate protective and retention controls.
SEC 9: How do you protect your data at rest?
Protect your data at rest by defining your requirements and implementing controls, including encryption, to reduce the risk of unauthorized access or loss.
SEC 10: How do you protect your data in transit?
Protecting your data in transit by defining your requirements and implementing controls, including encryption, reduces the risk of unauthorized access or exposure.

AWS provides multiple means for encrypting data at rest and in transit. We build features into our services that make it easier to encrypt your data. For example, we have implemented server-side encryption (SSE) for Amazon S3 to make it easier for you to store your data in an encrypted form. You can also arrange for the entire HTTPS encryption and decryption process (generally known as SSL termination) to be handled by Elastic Load Balancing (ELB).

Incident Response

Even with extremely mature preventive and detective controls, your organization should still put processes in place to respond to and mitigate the potential impact of security incidents. The architecture of your workload strongly affects the ability of your teams to operate effectively during an incident, to isolate or contain systems, and to restore operations to a known good state. Putting in place the tools and access ahead of a security incident, then routinely practicing incident response through game days, will help you ensure that your architecture can accommodate timely investigation and recovery.

In AWS, the following practices facilitate effective incident response:

- Detailed logging is available that contains important content, such as file access and changes.
- Events can be automatically processed and trigger tools that automate responses through the use of AWS APIs.

- You can pre-provision tooling and a “clean room” using AWS CloudFormation. This allows you to carry out forensics in a safe, isolated environment.

The following questions focus on these considerations for security.

SEC 11: How do you respond to an incident?
Preparation is critical to timely investigation and response to security incidents to help minimize potential disruption to your organization.

Ensure that you have a way to quickly grant access for your InfoSec team, and automate the isolation of instances as well as the capturing of data and state for forensics.

Key AWS Services

The AWS service that is essential to Security is **AWS Identity and Access Management (IAM)**, which allows you to securely control access to AWS services and resources for your users. The following services and features support the five areas in security:

- **Identity and Access Management**: IAM enables you to securely control access to AWS services and resources. MFA adds an additional layer of protection on user access. AWS Organizations lets you centrally manage and enforce policies for multiple AWS accounts.
- **Detective Controls**: AWS CloudTrail records AWS API calls, AWS Config provides a detailed inventory of your AWS resources and configuration. Amazon GuardDuty is a managed threat detection service that continuously monitors for malicious or unauthorized behavior. Amazon CloudWatch is a monitoring service for AWS resources which can trigger CloudWatch Events to automate security responses.
- **Infrastructure Protection**: Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. Amazon CloudFront is a global content delivery network that securely delivers data, videos, applications, and APIs to your viewers which integrates with AWS Shield for DDoS mitigation. AWS WAF is a web application firewall that is deployed on either Amazon CloudFront or Application Load Balancer to help protect your web applications from common web exploits.
- **Data Protection**: Services such as ELB, Amazon Elastic Block Store (Amazon EBS), Amazon S3, and Amazon Relational Database Service (Amazon RDS) include encryption capabilities to protect your data in transit and at rest. Amazon Macie automatically discovers, classifies and protects sensitive data, while AWS Key Management Service (AWS KMS) makes it easy for you to create and control keys used for encryption.

- **Incident Response**: IAM should be used to grant appropriate authorization to incident response teams and response tools. AWS CloudFormation can be used to create a trusted environment or clean room for conducting investigations. Amazon CloudWatch Events allows you to create rules that trigger automated responses including AWS Lambda.

Resources

Refer to the following resources to learn more about our best practices for Security.

Documentation

- AWS Cloud Security
- AWS Compliance
- AWS Security Blog

Whitepaper

- Security Pillar
- AWS Security Overview
- AWS Security Best Practices
- AWS Risk and Compliance

Video

- AWS Security State of the Union
- Shared Responsibility Overview

Reliability

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

The reliability pillar provides an overview of design principles, best practices, and questions. You can find prescriptive guidance on implementation in the Reliability Pillar whitepaper.

Design Principles

There are five design principles for reliability in the cloud:

- **Test recovery procedures**: In an on-premises environment, testing is often conducted to prove the system works in a particular scenario. Testing is not typically used to validate recovery strategies. In the cloud, you can test how your system fails, and you can validate your recovery procedures. You can use automation to simulate different failures or to recreate scenarios that led to failures before. This exposes failure pathways that you can test and rectify before a real failure scenario, reducing the risk of components failing that have not been tested before.
- **Automatically recover from failure**: By monitoring a system for key performance indicators (KPIs), you can trigger automation when a threshold is breached. This allows for automatic notification and tracking of failures, and for automated recovery processes that work around or repair the failure. With more sophisticated automation, it's possible to anticipate and remediate failures before they occur.
- **Scale horizontally to increase aggregate system availability**: Replace one large resource with multiple small resources to reduce the impact of a single failure on the overall system. Distribute requests across multiple, smaller resources to ensure that they don't share a common point of failure.
- **Stop guessing capacity**: A common cause of failure in on-premises systems is resource saturation, when the demands placed on a system exceed the capacity of that system (this is often the objective of denial of service attacks). In the cloud, you can monitor demand and system utilization, and automate the addition or removal of resources to maintain the optimal level to satisfy demand without over- or under- provisioning.
- **Manage change in automation**: Changes to your infrastructure should be done using automation. The changes that need to be managed are changes to the automation.

Definition

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

- **Foundations**
- **Change Management**
- **Failure Management**

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

Best Practices

Foundations

Before architecting any system, foundational requirements that influence reliability should be in place. For example, you must have sufficient network bandwidth to your data center. These requirements are sometimes neglected (because they are beyond a single project's scope). This neglect can have a significant impact on the ability to deliver a reliable system. In an on-premises environment, these requirements can cause long lead times due to dependencies and therefore must be incorporated during initial planning.

With AWS, most of these foundational requirements are already incorporated or may be addressed as needed. The cloud is designed to be essentially limitless, so it is the responsibility of AWS to satisfy the requirement for sufficient networking and compute capacity, while you are free to change resource size and allocation, such as the size of storage devices, on demand.

The following questions focus on these considerations for reliability. (For a list of reliability questions, answers, and best practices, see the Appendix.)

REL 1: How do you manage service limits?
Default service limits exist to prevent accidental provisioning of more resources than you need. There are also limits on how often you can call API operations to protect services from abuse. If you are using AWS Direct Connect, you have limits on the amount of data you can transfer on each connection. If you are using AWS Marketplace applications, you need to understand the limitations of the applications. If you are using third-party web services or software as a service, you also need to be aware of the limits of those services.
REL 2: How do you manage your network topology?
Applications can exist in one or more environments: your existing data center infrastructure, publicly accessible public cloud infrastructure, or private addressed public cloud infrastructure. Network considerations such as intra- and inter-system connectivity, public IP address management, private address management, and name resolution are fundamental to using resources in the cloud.

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

Change Management

Being aware of how change affects a system allows you to plan proactively, and monitoring allows you to quickly identify trends that could lead to capacity issues or SLA breaches. In traditional environments, change-control processes are often manual and must be carefully coordinated with auditing to effectively control who makes changes and when they are made.

Using AWS, you can monitor the behavior of a system and automate the response to KPIs, for example, by adding additional servers as a system gains more users. You can control who has permission to make system changes and audit the history of these changes.

The following questions focus on these considerations for reliability.

REL 3: How does your system adapt to changes in demand?
A scalable system provides elasticity to add and remove resources automatically so that they closely match the current demand at any given point in time.
REL 4: How do you monitor your resources?
Logs and metrics are a powerful tool to gain insight into the health of your workloads. You can configure your workload to monitor logs and metrics and send notifications when thresholds are crossed or significant events occur. Ideally, when low-performance thresholds are crossed or failures occur, the workload has been architected to automatically self-heal or scale in response.
REL 5: How do you implement change?
Uncontrolled changes to your environment make it difficult to predict the effect of a change. Controlled changes to provisioned resources and workloads are necessary to ensure that the workloads and the operating environment are running known software and can be patched or replaced in a predictable manner.

When you architect a system to automatically add and remove resources in response to changes in demand, this not only increases reliability but also ensures that business success doesn't become a burden. With monitoring in place, your team will be automatically alerted when KPIs deviate from expected norms. Automatic logging of changes to your environment allows you to audit and quickly identify actions that might have impacted reliability. Controls on change management ensure that you can enforce the rules that deliver the reliability you need.

Failure Management

In any system of reasonable complexity it is expected that failures will occur. It is generally of interest to know how to become aware of these failures, respond to them, and prevent them from happening again.

With AWS, you can take advantage of automation to react to monitoring data. For example, when a particular metric crosses a threshold, you can trigger an automated action to remedy the problem. Also, rather than trying to diagnose and fix a failed resource that is part of your production environment, you can replace it with a new one and carry out the analysis on the failed resource out of band. Since the cloud enables you to stand up temporary versions of a whole system at low cost, you can use automated testing to verify full recovery processes.

The following questions focus on these considerations for reliability.

REL 6: How do you back up data?
Back up data, applications, and operating environments (defined as operating systems configured with applications) to meet requirements for mean time to recovery (MTTR) and recovery point objectives (RPO).
REL 7: How does your system withstand component failures?
If your workloads have a requirement, implicit or explicit, for high availability and low mean time to recovery (MTTR), architect your workloads for resilience and distribute your workloads to withstand outages.
REL 8: How do you test resilience?
Test the resilience of your workload to help you find latent bugs that only surface in production. Exercise these tests regularly.
REL 9: How do you plan for disaster recovery?
Disaster recovery (DR) is critical should restoration of data be required from backup methods. Your definition of and execution on the objectives, resources, locations, and functions of this data must align with RTO and RPO objectives.

Regularly back up your data and test your backup files to ensure you can recover from both logical and physical errors. A key to managing failure is the frequent and automated testing of systems to cause failure, and then observe how they recover. Do this on a regular schedule and ensure that such testing is also triggered after significant system changes. Actively track KPIs, such as the recovery time objective (RTO) and recovery point objective (RPO), to assess a system's resiliency (especially under failure-testing scenarios). Tracking KPIs will help you identify and mitigate single points of failure. The objective is to thoroughly test your system-recovery processes so that you are confident that you can recover all your data and continue to serve your customers, even in the face of sustained problems. Your recovery processes should be as well exercised as your normal production processes.

Key AWS Services

The AWS service that is essential to Reliability is **Amazon CloudWatch**, which monitors runtime metrics. The following services and features support the three areas in reliability:

- **Foundations**: AWS IAM enables you to securely control access to AWS services and resources. Amazon VPC lets you provision a private, isolated section of the

AWS Cloud where you can launch AWS resources in a virtual network. AWS Trusted Advisor provides visibility into service limits. AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards web applications running on AWS.

- **Change Management**: AWS CloudTrail records AWS API calls for your account and delivers log files to you for auditing. AWS Config provides a detailed inventory of your AWS resources and configuration, and continuously records configuration changes. Amazon Auto Scaling is a service that will provide an automated demand management for a deployed workload. Amazon CloudWatch provides the ability to alert on metrics, including custom metrics. Amazon CloudWatch also has a logging feature that can be used to aggregate log files from your resources.
- **Failure Management**: AWS CloudFormation provides templates for the creation of AWS resources and provisions them in an orderly and predictable fashion. Amazon S3 provides a highly durable service to keep backups. Amazon Glacier provides highly durable archives. AWS KMS provides a reliable key management system that integrates with many AWS services.

Resources

Refer to the following resources to learn more about our best practices for Reliability.

Documentation

- Service Limits
- Service Limits Reports Blog
- Amazon Virtual Private Cloud
- AWS Shield
- Amazon CloudWatch
- Amazon S3
- AWS KMS

Whitepaper

- Reliability Pillar
- Backup Archive and Restore Approach Using AWS
- Managing your AWS Infrastructure at Scale
- AWS Disaster Recovery
- AWS Amazon VPC Connectivity Options

Video

- How do I manage my AWS service limits?
- Embracing Failure: Fault-Injection and Service Reliability

Product

- AWS Premium Support
- Trusted Advisor

Performance Efficiency

The **Performance Efficiency** pillar includes the ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve.

The performance efficiency pillar provides an overview of design principles, best practices, and questions. You can find prescriptive guidance on implementation in the Performance Efficiency Pillar whitepaper.

Design Principles

There are five design principles for performance efficiency in the cloud:

- **Democratize advanced technologies**: Technologies that are difficult to implement can become easier to consume by pushing that knowledge and complexity into the cloud vendor's domain. Rather than having your IT team learn how to host and run a new technology, they can simply consume it as a service. For example, NoSQL databases, media transcoding, and machine learning are all technologies that require expertise that is not evenly dispersed across the technical community. In the cloud, these technologies become services that your team can consume while focusing on product development rather than resource provisioning and management.
- **Go global in minutes**: Easily deploy your system in multiple Regions around the world with just a few clicks. This allows you to provide lower latency and a better experience for your customers at minimal cost.
- **Use serverless architectures**: In the cloud, serverless architectures remove the need for you to run and maintain servers to carry out traditional compute activities. For example, storage services can act as static websites, removing the need for web servers, and event services can host your code for you. This not only removes the operational burden of managing these servers, but also can lower transactional costs because these managed services operate at cloud scale.

- **Experiment more often**: With virtual and automatable resources, you can quickly carry out comparative testing using different types of instances, storage, or configurations.
- **Mechanical sympathy**: Use the technology approach that aligns best to what you are trying to achieve. For example, consider data access patterns when selecting database or storage approaches.

Definition

There are four best practice areas for performance efficiency in the cloud:

- **Selection**
- **Review**
- **Monitoring**
- **Tradeoffs**

Take a data-driven approach to selecting a high-performance architecture. Gather data on all aspects of the architecture, from the high-level design to the selection and configuration of resource types. By reviewing your choices on a cyclical basis, you will ensure that you are taking advantage of the continually evolving AWS Cloud. Monitoring will ensure that you are aware of any deviance from expected performance and can take action on it. Finally, your architecture can make tradeoffs to improve performance, such as using compression or caching, or relaxing consistency requirements.

Best Practices

Selection

The optimal solution for a particular system will vary based on the kind of workload you have, often with multiple approaches combined. Well-architected systems use multiple solutions and enable different features to improve performance.

In AWS, resources are virtualized and are available in a number of different types and configurations. This makes it easier to find an approach that closely matches your needs, and you can also find options that are not easily achievable with on-premises infrastructure. For example, a managed service such as Amazon DynamoDB provides a fully managed NoSQL database with single-digit millisecond latency at any scale.

The following questions focus on these considerations for performance efficiency. (For a list of performance efficiency questions, answers, and best practices, see the Appendix.)

PERF 1: How do you select the best performing architecture?
Often, multiple approaches are required to get optimal performance across a workload. Well-architected systems use multiple solutions and enable different features to improve performance.

When you select the patterns and implementation for your architecture use a data-driven approach for the most optimal solution. AWS Solutions Architects, AWS Reference Architectures, and AWS Partner Network (APN) Partners can help you select an architecture based on what we have learned, but data obtained through benchmarking or load testing will be required to optimize your architecture.

Your architecture will likely combine a number of different architectural approaches (for example, event-driven, ETL, or pipeline). The implementation of your architecture will use the AWS services that are specific to the optimization of your architecture's performance. In the following sections we look at the four main resource types that you should consider (compute, storage, database, and network).

Compute

The optimal compute solution for a particular system may vary based on application design, usage patterns, and configuration settings. Architectures may use different compute solutions for various components and enable different features to improve performance. Selecting the wrong compute solution for an architecture can lead to lower performance efficiency.

In AWS, compute is available in three forms: instances, containers, and functions:

- **Instances** are virtualized servers and, therefore, you can change their capabilities with the click of a button or an API call. Because in the cloud resource decisions are no longer fixed, you can experiment with different server types. At AWS, these virtual server instances come in different families and sizes, and they offer a wide variety of capabilities, including solid-state drives (SSDs) and graphics processing units (GPUs).
- **Containers** are a method of operating system virtualization that allow you to run an application and its dependencies in resource-isolated processes.
- **Functions** abstract the execution environment from the code you want to execute. For example, AWS Lambda allows you to execute code without running an instance.

The following questions focus on these considerations for performance efficiency.

PERF 2: How do you select your compute solution?
The optimal compute solution for a system varies based on application design, usage patterns, and configuration settings. Architectures may use different compute solutions for various components and enable different features to improve performance. Selecting the wrong compute solution for an architecture can lead to lower performance efficiency.

When architecting your use of compute you should take advantage of the elasticity mechanisms available to ensure you have sufficient capacity to sustain performance as demand changes.

Storage

The optimal storage solution for a particular system will vary based on the kind of access method (block, file, or object), patterns of access (random or sequential), throughput required, frequency of access (online, offline, archival), frequency of update (WORM, dynamic), and availability and durability constraints. Well-architected systems use multiple storage solutions and enable different features to improve performance.

In AWS, storage is virtualized and is available in a number of different types. This makes it easier to match your storage methods more closely with your needs, and also offers storage options that are not easily achievable with on- premises infrastructure. For example, Amazon S3 is designed for 11 nines of durability. You can also change from using magnetic hard disk drives (HDDs) to SSDs, and easily move virtual drives from one instance to another in seconds.

The following questions focus on these considerations for performance efficiency.

PERF 3: How do you select your storage solution?
The optimal storage solution for a system varies based on the kind of access method (block, file, or object), patterns of access (random or sequential), required throughput, frequency of access (online, offline, archival), frequency of update (WORM, dynamic), and availability and durability constraints. Well-architected systems use multiple storage solutions and enable different features to improve performance and use resources efficiently.

When you select a storage solution, ensuring that it aligns with your access patterns will be critical to achieving the performance you want.

Database

The optimal database solution for a particular system can vary based on requirements for availability, consistency, partition tolerance, latency, durability, scalability, and query capability. Many systems use different database solutions for various subsystems and enable different features to improve performance. Selecting the wrong database solution and features for a system can lead to lower performance efficiency.

Amazon RDS provides a fully managed relational database. With Amazon RDS, you can scale your database's compute and storage resources, often with no downtime. Amazon DynamoDB is a fully managed NoSQL database that provides single-digit millisecond latency at any scale. Amazon Redshift is a managed petabyte-scale data warehouse that allows you to change the number or type of nodes as your performance or capacity needs change.

The following questions focus on these considerations for performance efficiency.

PERF 4: How do you select your database solution?
The optimal database solution for a system varies based on requirements for availability, consistency, partition tolerance, latency, durability, scalability, and query capability. Many systems use different database solutions for various sub-systems and enable different features to improve performance. Selecting the wrong database solution and features for a system can lead to lower performance efficiency.

Although a workload's database approach (RDBMS, NoSQL) has significant impact on performance efficiency, it is often an area that is chosen according to organizational defaults rather than through a data-driven approach. As with storage, it is critical to consider the access patterns of your workload, and also to consider if other non-database solutions could solve the problem more efficiently (such as using a search engine or data warehouse).

Network

The optimal network solution for a particular system will vary based on latency, throughput requirements and so on. Physical constraints such as user or on-premises resources will drive location options, which can be offset using edge techniques or resource placement.

In AWS, networking is virtualized and is available in a number of different types and configurations. This makes it easier to match your networking methods more closely with your needs. AWS offers product features (for example, Enhanced Networking, Amazon EBS-optimized instances, Amazon S3 transfer acceleration, dynamic Amazon CloudFront) to optimize network traffic. AWS also offers networking features (for example, Amazon Route 53 latency routing, Amazon VPC endpoints, and AWS Direct Connect) to reduce network distance or jitter.

The following questions focus on these considerations for performance efficiency.

PERF 5: How do you configure your networking solution?
The optimal network solution for a system varies based on latency, throughput requirements, and so on. Physical constraints such as user or on-premises resources drive location options, which can be offset using edge techniques or resource placement.

When selecting your network solution, you need to consider location. With AWS, you can choose to place resources close to where they will be used to reduce distance. By taking advantage of Regions, placement groups, and edge locations you can significantly improve performance.

Review

When architecting solutions, there is a finite set of options that you can choose from. However, over time new technologies and approaches become available that could improve the performance of your architecture.

Using AWS, you can take advantage of our continual innovation, which is driven by customer need. We release new Regions, edge locations, services, and features regularly. Any of these could positively improve the performance efficiency of your architecture.

The following questions focus on these considerations for performance efficiency.

PERF 6: How do you evolve your workload to take advantage of new releases?
When architecting workloads, there are finite options that you can choose from. However, over time, new technologies and approaches become available that could improve the performance of your workload.

Understanding where your architecture is performance-constrained will allow you to look out for releases that could alleviate that constraint.

Monitoring

After you have implemented your architecture you will need to monitor its performance so that you can remediate any issues before your customers are aware. Monitoring metrics should be used to raise alarms when thresholds are breached. The alarm can trigger automated action to work around any badly performing components.

Amazon CloudWatch provides the ability to monitor and send notification alarms. You can use automation to work around performance issues by triggering actions through Amazon Kinesis, Amazon Simple Queue Service (Amazon SQS), and AWS Lambda.

The following questions focus on these considerations for performance efficiency.

PERF 7: How do you monitor your resources to ensure they are performing as expected?
System performance can degrade over time. Monitor system performance to identify this degradation and remediate internal or external factors, such as the operating system or application load.

Ensuring that you do not see too many false positives, or are overwhelmed with data, is key to having an effective monitoring solution. Automated triggers avoid human error and can reduce the time to fix problems. Plan for game days where you can conduct simulations in the production environment to test your alarm solution and ensure that it correctly recognizes issues.

Tradeoffs

When you architect solutions, think about tradeoffs so you can select an optimal approach. Depending on your situation you could trade consistency, durability, and space versus time or latency to deliver higher performance.

Using AWS, you can go global in minutes and deploy resources in multiple locations across the globe to be closer to your end users. You can also dynamically add read-only replicas to information stores such as database systems to reduce the load on the primary database. AWS also offers caching solutions such as Amazon ElastiCache, which provides an in-memory data store or cache, and Amazon CloudFront, which caches copies of your static content closer to end users. Amazon DynamoDB Accelerator (DAX) provides a read-through/write-through distributed caching tier in front of Amazon DynamoDB, supporting the same API, but providing sub-millisecond latency for entities that are in the cache.

The following questions focus on these considerations for performance efficiency.

PERF 8: How do you use tradeoffs to improve performance?
When architecting solutions, actively considering tradeoffs enables you to select an optimal approach. Often you can improve performance by trading consistency, durability, and space for time and latency.

Tradeoffs can increase the complexity of your architecture and require load testing to ensure that a measurable benefit is obtained.

Key AWS Services

The AWS service that is essential to Performance Efficiency is **Amazon CloudWatch**, which monitors your resources and systems, providing visibility into your overall performance and operational health. The following services and features support the four areas in performance efficiency:

- **Selection**
  - **Compute**: Auto Scaling is key to ensuring that you have enough instances to meet demand and maintain responsiveness.
  - **Storage**: Amazon EBS provides a wide range of storage options (such as SSD and provisioned input/output operations per second (PIOPS)) that allow you to optimize for your use case. Amazon S3 provides serverless content delivery, and Amazon S3 transfer acceleration enables fast, easy, and secure transfers of files over long distances.
  - **Database**: Amazon RDS provides a wide range of database features (such as PIOPS and read replicas) that allow you to optimize for your use case. Amazon DynamoDB provides single-digit millisecond latency at any scale.
  - **Network**: Amazon Route 53 provides latency-based routing. Amazon VPC endpoints and AWS Direct Connect can reduce network distance or jitter.
- **Review**: The AWS Blog and the What's New section on the AWS website are resources for learning about newly launched features and services.
- **Monitoring**: Amazon CloudWatch provides metrics, alarms, and notifications that you can integrate with your existing monitoring solution, and that you can use with AWS Lambda to trigger actions.
- **Tradeoffs**: Amazon ElastiCache, Amazon CloudFront, and AWS Snowball are services that allow you to improve performance. Read replicas in Amazon RDS can allow you to scale read-heavy workloads.

Resources

Refer to the following resources to learn more about our best practices for Performance Efficiency.

Documentation

- Amazon S3 Performance Optimization
- Amazon EBS Volume Performance

Whitepaper

- Performance Efficiency Pillar

Video

- AWS re:Invent 2016: Scaling Up to Your First 10 Million Users (ARC201)
- AWS re:Invent 2017: Deep Dive on Amazon EC2 Instances

Cost Optimization

The **Cost Optimization** pillar includes the ability to run systems to deliver business value at the lowest price point.

The cost optimization pillar provides an overview of design principles, best practices, and questions. You can find prescriptive guidance on implementation in the Cost Optimization Pillar whitepaper.

Design Principles

There are five design principles for cost optimization in the cloud:

- **Adopt a consumption model**: Pay only for the computing resources that you require and increase or decrease usage depending on business requirements, not by using elaborate forecasting. For example, development and test environments are typically only used for eight hours a day during the work week. You can stop these resources when they are not in use for a potential cost savings of 75% (40 hours versus 168 hours).
- **Measure overall efficiency**: Measure the business output of the workload and the costs associated with delivering it. Use this measure to know the gains you make from increasing output and reducing costs.
- **Stop spending money on data center operations**: AWS does the heavy lifting of racking, stacking, and powering servers, so you can focus on your customers and organization projects rather than on IT infrastructure.
- **Analyze and attribute expenditure**: The cloud makes it easier to accurately identify the usage and cost of systems, which then allows transparent attribution of IT costs to individual workload owners. This helps measure return on investment (ROI) and gives workload owners an opportunity to optimize their resources and reduce costs.
- **Use managed and application level services to reduce cost of ownership**: In the cloud, managed and application level services remove the operational burden of maintaining servers for tasks such as sending email or managing databases. As managed services operate at cloud scale, they can offer a lower cost per transaction or service.

Definition

There are four best practice areas for cost optimization in the cloud:

- **Expenditure Awareness**
- **Cost-Effective Resources**
- **Matching supply and demand**
- **Optimizing Over Time**

As with the other pillars, there are tradeoffs to consider. For example, do you want to prioritize for speed to market or for cost? In some cases, it's best to prioritize for speed—going to market quickly, shipping new features, or simply meeting a deadline—rather than investing in upfront cost optimization. Design decisions are sometimes guided by haste as opposed to empirical data, as the temptation always exists to overcompensate “just in case” rather than spend time benchmarking for the most cost-optimal workload over time. This often leads to drastically over-provisioned and under-optimized deployments, which remain static throughout their life cycle. The following sections provide techniques and strategic guidance for the initial and ongoing cost optimization of your deployment.

Best Practices

Expenditure Awareness

The increased flexibility and agility that the cloud enables encourages innovation and fast-paced development and deployment. It eliminates the manual processes and time associated with provisioning on-premises infrastructure, including identifying hardware specifications, negotiating price quotations, managing purchase orders, scheduling shipments, and then deploying the resources. However, the ease of use and virtually unlimited on-demand capacity requires a new way of thinking about expenditures.

Many businesses are composed of multiple systems run by various teams. The capability to attribute resource costs to the individual organization or product owners drives efficient usage behavior and helps reduce waste. Accurate cost attribution allows you to know which products are truly profitable, and allows you to make more informed decisions about where to allocate budget.

In AWS you can use Cost Explorer to track your spend, and gain insights into exactly where you spend. Using AWS Budgets, you can send notifications if your usage or costs are not inline with your forecasts. You can use tagging on resources to apply business and organization information to your usage and cost; this provides additional insights to optimization from an organization perspective.

The following questions focus on these considerations for cost optimization. (For a list of cost optimization questions, answers, and best practices, see the Appendix.)

COST 1: How do you govern usage?
Establish policies and mechanisms to ensure that appropriate costs are incurred while objectives are achieved. By employing a checks-and-balances approach, you can innovate without overspending.
COST 2: How do you monitor usage and cost?
Establish policies and procedures to monitor and appropriately allocate your costs. This allows you to measure and improve the cost efficiency of this workload.
COST 3: How do you decommission resources?
Implement change control and resource management from project inception to end-of-life. This ensures you shut down or terminate unused resources to reduce waste.

You can use cost allocation tags to categorize and track your AWS usage and costs. When you apply tags to your AWS resources (such as EC2 instances or S3 buckets), AWS generates a cost and usage report with your usage and your tags. You can apply tags that represent organization categories (such as cost centers, workload names, or owners) to organize your costs across multiple services.

Combining tagged resources with entity lifecycle tracking (employees, projects) makes it possible to identify orphaned resources or projects that are no longer generating value to the organization and should be decommissioned. You can set up billing alerts to notify you of predicted overspending, and the AWS Simple Monthly Calculator allows you to calculate your data transfer costs.

Cost-Effective Resources

Using the appropriate instances and resources for your workload is key to cost savings. For example, a reporting process might take five hours to run on a smaller server but one hour to run on a larger server that is twice as expensive. Both servers give you the same outcome, but the smaller server incurs more cost over time.

A well-architected workload uses the most cost-effective resources, which can have a significant and positive economic impact. You also have the opportunity to use managed services to reduce costs. For example, rather than maintaining servers to deliver email, you can use a service that charges on a per-message basis.

AWS offers a variety of flexible and cost-effective pricing options to acquire instances from EC2 and other services in a way that best fits your needs. *On-Demand Instances* allow you to pay for compute capacity by the hour, with no minimum commitments required. *Reserved Instances* allow you to reserve capacity and offer savings of up to 75% off On-Demand pricing. With Spot Instances, you can leverage unused

Amazon EC2 capacity and offer savings of up to 90% off On-Demand pricing. *Spot Instances* are appropriate where the system can tolerate using a fleet of servers where individual servers can come and go dynamically, such as stateless web servers, batch processing, or when using HPC and big data.

Appropriate service selection can also reduce usage and costs; such as CloudFront to minimize data transfer, or completely eliminate costs, such as utilizing Amazon Aurora on RDS to remove expensive database licensing costs.

The following questions focus on these considerations for cost optimization.

COST 4: How do you evaluate cost when you select services?
Amazon EC2, Amazon EBS, and Amazon S3 are building-block AWS services. Managed services, such as Amazon RDS and Amazon DynamoDB, are higher level, or application level, AWS services. By selecting the appropriate building blocks and managed services, you can optimize this workload for cost. For example, using managed services, you can reduce or remove much of your administrative and operational overhead, freeing you to work on applications and business-related activities.
COST 5: How do you meet cost targets when you select resource type and size?
Ensure that you choose the appropriate resource size for the task at hand. By selecting the most cost effective type and size, you minimize waste.
COST 6: How do you use pricing models to reduce cost?
Use the pricing model that is most appropriate for your resources to minimize expense.
COST 7: How do you plan for data transfer charges?
Ensure that you plan and monitor data transfer charges so that you can make architectural decisions to minimize costs. A small yet effective architectural change can drastically reduce your operational costs over time.

By factoring in cost during service selection, and using tools such as Cost Explorer and AWS Trusted Advisor to regularly review your AWS usage, you can actively monitor your utilization and adjust your deployments accordingly.

Matching supply and demand

Optimally matching supply to demand delivers the lowest cost for a workload, but there also needs to be sufficient extra supply to allow for provisioning time and individual resource failures. Demand can be fixed or variable, requiring metrics and automation to ensure that management does not become a significant cost.

In AWS, you can automatically provision resources to match demand. Auto Scaling and demand, buffer, and time-based approaches allow you to add and remove resources as needed. If you can anticipate changes in demand, you can save more money and ensure your resources match your workload needs.

The following questions focus on these considerations for cost optimization.

COST 8: How do you match supply of resources with demand?
For a workload that has balanced spend and performance, ensure that everything you pay for is used and avoid significantly underutilizing instances. A skewed utilization metric in either direction has an adverse impact on your organization, in either operational costs (degraded performance due to over-utilization), or wasted AWS expenditures (due to over-provisioning).

When designing to match supply against demand, actively think about the patterns of usage and the time it takes to provision new resources.

Optimizing Over Time

As AWS releases new services and features, it is a best practice to review your existing architectural decisions to ensure they continue to be the most cost-effective. As your requirements change, be aggressive in decommissioning resources, entire services, and systems that you no longer require.

Managed services from AWS can significantly optimize the workload, so it is essential to be aware of new managed services and features as they become available. For example, running an Amazon RDS database can be cheaper than running your own database on Amazon EC2.

The following questions focus on these considerations for cost optimization.

COST 9: How do you evaluate new services?
As AWS releases new services and features, it is a best practice to review your existing architectural decisions to ensure they continue to be the most cost effective.

When regularly reviewing your deployments, assess how newer services can help save you money. For example, Amazon Aurora on RDS can reduce costs for relational databases

Key AWS Services

The tool that is essential to Cost Optimization is **Cost Explorer**, which helps you gain visibility and insights into your usage, across your workloads and throughout your organization. The following services and features support the four areas in cost optimization:

- **Expenditure Awareness**: AWS Cost Explorer allows you to view and track your usage in detail. AWS Budgets notify you if your usage or spend exceeds actual or forecast budgeted amounts.

- **Cost-Effective Resources**: You can use Cost Explorer for Reserved Instance recommendations, and see patterns in how much you spend on AWS resources over time. Use Amazon CloudWatch and Trusted Advisor to help right size your resources. You can use Amazon Aurora on RDS to remove database licensing costs. AWS Direct Connect and Amazon CloudFront can be used to optimize data transfer.
- **Matching supply and demand**: Auto Scaling allows you to add or remove resources to match demand without overspending.
- **Optimizing Over Time**: The AWS News Blog and the What's New section on the AWS website are resources for learning about newly launched features and services. AWS Trusted Advisor inspects your AWS environment and finds opportunities to save you money by eliminating unused or idle resources or committing to Reserved Instance capacity.

Resources

Refer to the following resources to learn more about our best practices for Cost Optimization.

Documentation

- Analyzing Your Costs with Cost Explorer
- AWS Cloud Economics Center
- AWS Detailed Billing Reports

Whitepaper

- Cost Optimization Pillar

Video

- Cost Optimization on AWS

Tool

- AWS Total Cost of Ownership (TCO) Calculators
- AWS Simple Monthly Calculator

The Review Process

The review of architectures needs to be done in a consistent manner, with a blame-free approach that encourages diving deep. It should be a light-weight process (hours not days) that is a conversation and not an audit. The purpose of reviewing an architecture is to identify any critical issues that might need addressing or areas that could be improved. The outcome of the review is a set of actions that should improve the experience of a customer using the workload.

As discussed in the "On Architecture" section, you will want each team member to take responsibility for the quality of its architecture. We recommend that the team members who build an architecture use the Well-Architected Framework to continually review their architecture, rather than holding a formal review meeting. A continuous approach allows your team members to update answers as the architecture evolves, and improve the architecture as you deliver features.

AWS Well-Architected is aligned to the way that AWS reviews systems and services internally. It is premised on a set of design principles that influences architectural approach, and questions that ensure that people don't neglect areas that often featured in Root Cause Analysis (RCA). Whenever there is a significant issue with an internal system, AWS service, or customer we look at the RCA to see if we could improve the review processes we use.

Reviews should be applied at key milestones in the product lifecycle, early on in the design phase to avoid *one-way doors* [1] that are difficult to change, and then before the go live date. Post go live your workload will continue to evolve as you add new features and change technology implementations. The architecture of a workload changes over time. You will need to follow good hygiene practices to stop its architectural characteristics from degrading as you evolve it. As you make significant architecture changes you should follow a set of hygiene processes including a Well-Architected review.

If you want to use the review as a one-time snapshot or independent measurement you will want to ensure you have all the right people in the conversation. Often we find that reviews are the first time that a team truly understands what they have implemented. An approach that works well when reviewing another team's workload is to have a series of informal conversations about their architecture where you can glean the answers to most questions. You can then follow up with one or two meetings where you can gain clarity or dive deep on areas of ambiguity or perceived risk.

Here are some suggested items to facilitate your meetings:

[1] Many decisions are reversible, two-way doors. Those decisions can use a light-weight process. One-way doors are hard or impossible to reverse and require more inspection before making them.

- A meeting room with whiteboards
- Print outs of any diagrams or design notes
- Action list of questions that require out-of-band research to answer (for example, "did we enable encryption or not?")

After you have done a review you should have a list of issues that you can prioritize based on your business context. You will also want to take into account the impact of those issues on the day-to-day work of your team. If you address these issues early you could free up time to work on creating business value rather than solving recurring problems. As you address issues you can update your review to see how the architecture is improving.

While the value of a review is clear after you have done one, you may find that a new team might be resistant at first. Here are some objections that can be handled through educating the team on the benefits of a review:

- "We are too busy!" (Often said when the team is getting ready for a big launch.)
  - If you are getting ready for a big launch you will want it to go smoothly. The review will allow you to understand any problems you might have missed.
  - We recommend that you carry out reviews early in the product lifecycle to uncover risks and develop a mitigation plan aligned with the feature delivery roadmap.
- "We don't have time to do anything with the results!" (Often said when there is an immovable event, such as the Super Bowl, that they are targeting.)
  - These events can't be moved. Do you really want to go into it without knowing the risks in your architecture? Even if you don't address all of these issues you can still have playbooks for handling them if they materialize
- "We don't want others to know the secrets of our solution implementation!"
  - If you point the team at the questions in the Well-Architected Framework, they will see that none of the questions reveal any commercial or technical propriety information.

As you carry out multiple reviews with teams in your organization you might identify thematic issues. For example, you might see that a group of teams has clusters of issues in a particular pillar or topic. You will want to look at all your reviews in a holistic manner, and identify any mechanisms, training, or principal engineering talks that could help address those thematic issues.

Conclusion

The AWS Well-Architected Framework provides architectural best practices across the five pillars for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud. The Framework provides a set of questions that allows you to review an existing or proposed architecture. It also provides a set of AWS best practices for each pillar. Using the Framework in your architecture will help you produce stable and efficient systems, which allow you to focus on your functional requirements.

Contributors

The following individuals and organizations contributed to this document:

- 'Fitz' Philip Fitzsimons: Sr. Manager Well-Architected, Amazon Web Services
- Brian Carlson: Operations Lead Well-Architected, Amazon Web Services
- Ben Potter: Security Lead Well-Architected, Amazon Web Services
- Rodney Lester: Reliability Lead Well-Architected, Amazon Web Services
- John Ewart: Performance Lead Well-Architected, Amazon Web Services
- Nathan Besh: Cost Lead Well-Architected, Amazon Web Services
- Jon Steele: Sr. Technical Account Manager, Amazon Web Services
- Ryan King: Technical Program Manager, Amazon Web Services
- Erin Rifkin: Senior Product Manager, Amazon Web Services
- Max Ramsay: Principal Security Solutions Architect, Amazon Web Services
- Scott Paddock: Security Solutions Architect, Amazon Web Services
- Callum Hughes: Solutions Architect, Amazon Web Services

Further Reading

AWS Well-Architected Partner program

AWS Well-Architected Tool

AWS Well-Architected homepage

Cost Optimization Pillar whitepaper

Operational Excellence Pillar whitepaper

Performance Efficiency Pillar whitepaper

Reliability Pillar whitepaper

Security Pillar whitepaper

Document Revisions

Table 2. Major revisions:

Date	Description
July 2019	Addition of AWS Well-Architected Tool, links to AWS Well-Architected Labs, and AWS Well-Architected Partners, minor fixes to enable multiple language version of framework.
November 2018	Review and rewrite of most questions and answers, to ensure questions focus on one topic at a time. This caused some previous questions to be split into multiple questions. Added common terms to definitions (workload, component etc). Changed presentation of question in main body to include descriptive text.
June 2018	Updates to simplify question text, standardize answers, and improve readability.
November 2017	Operational Excellence moved to front of pillars and rewritten so it frames other pillars. Refreshed other pillars to reflect evolution of AWS.
November 2016	Updated the Framework to include operational excellence pillar, and revised and updated the other pillars to reduce duplication and incorporate learnings from carrying out reviews with thousands of customers.
November 2015	Updated the Appendix with current Amazon CloudWatch Logs information.
October 2015	Original publication.

Appendix: Questions, Answers, and Best Practices

Operational Excellence

Prepare

OPS 1 How do you determine what your priorities are?

Everyone needs to understand their part in enabling business success. Have shared goals in order to set priorities for resources. This will maximize the benefits of your efforts.

Best practices:

- **Evaluate external customer needs**: Involve key stakeholders, including business, development, and operations teams, to determine where to focus operations efforts on external customer needs. This will ensure that you have a thorough understanding of the operations support that is required to achieve business outcomes.
- **Evaluate internal customer needs**: Involve key stakeholders, including business, development, and operations teams, when determining where to focus operations efforts on internal customer needs. This will ensure that you have a thorough understanding of the operations support that is required to achieve business outcomes.
- **Evaluate compliance requirements**: Evaluate external factors, such as regulatory compliance requirements and industry standards, to ensure that you are aware of guidelines or obligations that may mandate or emphasize specific focus. If no compliance requirements are identified, ensure that you apply due diligence to this determination.
- **Evaluate threat landscape**: Evaluate threats to the business (for example, competition, business risk and liabilities, operational risks, and information security threats), so that you can include their impact when determining where to focus operations efforts.
- **Evaluate tradeoffs**: Evaluate the impact of tradeoffs between competing interests, to help make informed decisions when determining where to focus operations efforts. For example, accelerating speed to market for new features may be emphasized over cost optimization.
- **Manage benefits and risks**: Manage benefits and risks to make informed decisions when determining where to focus operations efforts. For example, it may be beneficial to deploy a system with unresolved issues so that significant new features can be made available to customers.

OPS 2 How do you design your workload so that you can understand its state?

Design your workload so that it provides the information necessary for you to understand its internal state (for example, metrics, logs, and traces) across all components. This enables you to provide effective responses when appropriate.

Best practices:

- **Implement application telemetry**: Instrument your application code to emit information about its internal state, status, and achievement of business outcomes. For example, queue depth, error messages, and response times. Use this information to determine when a response is required.
- **Implement and configure workload telemetry**: Design and configure your workload to emit information about its internal state and current status. For example, API call volume, http status codes, and scaling events. Use this information to help determine when a response is required.
- **Implement user activity telemetry**: Instrument your application code to emit information about user activity. For example, click streams, or started, abandoned, and completed transactions. Use this information to help understand how the application is used, patterns of usage, and to determine when a response is required.
- **Implement dependency telemetry**: Design and configure your workload to emit information about the status of resources it depends on. Examples of these are external databases, DNS, and network connectivity. Use this information to determine when a response is required.
- **Implement transaction traceability**: Implement your application code and configure your workload components to emit information about the flow of transactions across the workload. Use this information to determine when a response is required and to assist in identifying the root cause of issues.

OPS 3 How do you reduce defects, ease remediation, and improve flow into production?

Adopt approaches that improve flow of changes into production, that enable refactoring, fast feedback on quality, and bug fixing. These accelerate beneficial changes entering production, limit issues deployed, and enable rapid identification and remediation of issues introduced through deployment activities.

Best practices:

- **Use version control**: Use version control to enable tracking of changes and releases.
- **Test and validate changes**: Test and validate changes to help limit and detect errors. Automate testing to reduce errors caused by manual processes, and reduce the level of effort to test.
- **Use configuration management systems**: Use configuration management systems to make and track configuration changes. These systems reduce errors caused by manual processes and reduce the level of effort to deploy changes.
- **Use build and deployment management systems**: Use build and deployment management systems. These systems reduce errors caused by manual processes and reduce the level of effort to deploy changes.
- **Perform patch management**: Perform patch management to gain features, address issues, and remain compliant with governance. Automate patch management to reduce errors caused by manual processes, and reduce the level of effort to patch.
- **Share design standards**: Share best practices across teams to increase awareness and maximize the benefits of development efforts.
- **Implement practices to improve code quality**: Implement practices to improve code quality and minimize defects. For example, test-driven development, code reviews, and standards adoption.
- **Use multiple environments**: Use multiple environments to experiment, develop, and test your workload. Use increasing levels of controls to gain confidence your workload will operate as intended.
- **Make frequent, small, reversible changes**: Frequent, small, and reversible changes reduce the scope and impact of a change. This eases troubleshooting, enables faster remediation, and provides the option to roll back a change.
- **Fully automate integration and deployment**: Automate build, deployment, and testing of the workload. This reduces errors caused by manual processes and reduces the effort to deploy changes.

OPS 4 How do you mitigate deployment risks?

Adopt approaches that provide fast feedback on quality and enable rapid recovery from changes that do not have desired outcomes. Using these practices mitigates the impact of issues introduced through the deployment of changes.

Best practices:

- **Plan for unsuccessful changes**: Plan to revert to a known good state, or remediate in the production environment if a change does not have the desired outcome. This preparation reduces recovery time through faster responses.
- **Test and validate changes**: Test changes and validate the results at all lifecycle stages, to confirm new features and minimize the risk and impact of failed deployments.
- **Use deployment management systems**: Use deployment management systems to track and implement change. This reduces errors cause by manual processes and reduces the effort to deploy changes.
- **Test using limited deployments**: Test with limited deployments alongside existing systems to confirm desired outcomes prior to full scale deployment. For example, use deployment canary testing or one-box deployments.
- **Deploy using parallel environments**: Implement changes onto parallel environments, and then transition to the new environment. Maintain the prior environment until there is confirmation of successful deployment. Doing so minimizes recovery time by enabling rollback to the previous environment.
- **Deploy frequent, small, reversible changes**: Use frequent, small, and reversible changes to reduce the scope of a change. This results in easier troubleshooting and faster remediation with the option to roll back a change.
- **Fully automate integration and deployment**: Automate build, deployment, and testing of the workload. This reduces errors cause by manual processes and reduces the effort to deploy changes.
- **Automate testing and rollback**: Automate testing of deployed environments to confirm desired outcomes. Automate rollback to previous known good state when outcomes are not achieved to minimize recovery time and reduce errors caused by manual processes.

OPS 5 How do you know that you are ready to support a workload?

Evaluate the operational readiness of your workload, processes and procedures, and personnel to understand the operational risks related to your workload.

Best practices:

- **Ensure personnel capability**: Have a mechanism to validate that you have an appropriate number of trained personnel to provide support for operational needs. Train personnel and adjust personnel capacity as necessary to maintain effective support.
- **Ensure consistent review of operational readiness**: Ensure you have a consistent review of your readiness to operate a workload. Review must include at a minimum the operational readiness of the teams and the workload, and security considerations. Implement review activities in code and trigger automated review in response to events where appropriate, to ensure consistency, speed of execution, and reduce errors caused by manual processes.
- **Use runbooks to perform procedures**: Runbooks are documented procedures to achieve specific outcomes. Enable consistent and prompt responses to well-understood events by documenting procedures in runbooks. Implement runbooks as code and trigger the execution of runbooks in response to events where appropriate, to ensure consistency, speed responses, and reduce errors caused by manual processes.
- **Use playbooks to identify issues**: Playbooks are documented processes to investigate issues. Enable consistent and prompt responses to failure scenarios by documenting investigation processes in playbooks. Implement playbooks as code and trigger playbook execution in response to events where appropriate, to ensure consistency, speed responses, and reduce errors caused by manual processes.
- **Make informed decisions to deploy systems and changes**: Evaluate the capabilities of the team to support the workload and the workload's compliance with governance. Evaluate these against the benefits of deployment when determining whether to transition a system or change into production. Understand the benefits and risks to make informed decisions.

Operate

OPS 6 How do you understand the health of your workload?

Define, capture, and analyze workload metrics to gain visibility to workload events so that you can take appropriate action.

Best practices:

- **Identify key performance indicators**: Identify key performance indicators (KPIs) based on desired business and customer outcomes. Evaluate KPIs to determine workload success.
- **Define workload metrics**: Define workload metrics to measure the achievement of KPIs. Define workload metrics to measure the health of the workload. Evaluate metrics to determine if the workload is achieving desired outcomes, and to understand the health of the workload.
- **Collect and analyze workload metrics**: Perform regular proactive reviews of metrics to identify trends and determine where appropriate responses are needed.
- **Establish workload metrics baselines**: Establish baselines for metrics to provide expected values as the basis for comparison and identification of under and over performing components.
- **Learn expected patterns of activity for workload**: Establish patterns of workload activity to determine when behavior is outside of the expected values so that you can respond appropriately if required.
- **Alert when workload outcomes are at risk**: Raise an alert when workload outcomes are at risk so that you can respond appropriately if required.
- **Alert when workload anomalies are detected**: Raise an alert when workload anomalies are detected so that you can respond appropriately if required.
- **Validate the achievement of outcomes and the effectiveness of KPIs and metrics** : Create a business-level view of your workload operations to help you determine if you are satisfying needs and to identify areas that need improvement to reach business goals. Validate the effectiveness of KPIs and metrics and revise them if necessary.

OPS 7 How do you understand the health of your operations?

Define, capture, and analyze operations metrics to gain visibility to operations events so that you can take appropriate action.

Best practices:

- **Identify key performance indicators**: Identify key performance indicators (KPIs) based on desired business and customer outcomes. Evaluate KPIs to determine operations success.
- **Define operations metrics**: Define operations metrics to measure the achievement of KPIs. Define operations metrics to measure the health of the operations. Evaluate metrics to determine if the operations are achieving desired outcomes, and to understand operations health.
- **Collect and analyze operations metrics**: Perform regular proactive reviews of metrics to identify trends and determine where appropriate responses are needed.
- **Establish operations metrics baselines**: Establish baselines for metrics to provide expected values as the basis for comparison and identification of under and over performing processes.
- **Learn the expected patterns of activity for operations**: Establish baselines for metrics to provide expected values as the basis for comparison.
- **Alert when operations outcomes are at risk**: Raise an alert when operations outcomes are at risk so that you can respond appropriately if required.
- **Alert when operations anomalies are detected**: Raise an alert when operations anomalies are detected so that you can respond appropriately if required.
- **Validate the achievement of outcomes and the effectiveness of KPIs and metrics** : Create a business-level view of your operations activities to help you determine if you are satisfying needs and to identify areas that need improvement to reach business goals. Validate the effectiveness of KPIs and metrics and revise them if necessary.

OPS 8 How do you manage workload and operations events?

Prepare and validate procedures for responding to events to minimize their disruption to your workload.

Best practices:

- **Use processes for event, incident, and problem management**: Have processes to address observed events, events that require intervention (incidents), and events that require intervention and either recur or cannot currently be resolved (problems). Use these processes to mitigate the impact of these events on the business and your customers by ensuring timely and appropriate responses.
- **Use a process for root cause analysis**: Have a process to identify and document the root cause of an event so that you can develop mitigations to limit or prevent recurrence and you can develop procedures for prompt and effective responses. Communicate root cause as appropriate, tailored to target audiences.
- **Have a process per alert**: Have a well-defined response (runbook or playbook), with a specifically identified owner, for any event for which you raise an alert. This ensures effective and prompt responses to operations events and prevents actionable events from being obscured by less valuable notifications.
- **Prioritize operational events based on business impact**: Ensure that when multiple events require intervention, those that are most significant to the business are addressed first. For example, impacts can include loss of life or injury, financial loss, or damage to reputation or trust.
- **Define escalation paths**: Define escalation paths in your runbooks and playbooks, including what triggers escalation, and procedures for escalation. Specifically identify owners for each action to ensure effective and prompt responses to operations events.
- **Enable push notifications**: Communicate directly with your users (for example, with email or SMS) when the services they use are impacted, and when the services return to normal operating conditions, to enable users to take appropriate action.
- **Communicate status through dashboards**: Provide dashboards tailored to their target audiences (for example, internal technical teams, leadership, and customers) to communicate the current operating status of the business and provide metrics of interest.
- **Automate responses to events**: Automate responses to events to reduce errors caused by manual processes, and to ensure prompt and consistent responses.

Evolve

OPS 9 How do you evolve operations?

Dedicate time and resources for continuous incremental improvement to evolve the effectiveness and efficiency of your operations.

Best practices:

- **Have a process for continuous improvement**: Regularly evaluate and prioritize opportunities for improvement to focus efforts where they can provide the greatest benefits.
- **Implement feedback loops**: Include feedback loops in your procedures and workloads to help you identify issues and areas that need improvement.
- **Define drivers for improvement**: Identify drivers for improvement to help you evaluate and prioritize opportunities.
- **Validate insights**: Review your analysis results and responses with cross-functional teams and business owners. Use these reviews to establish common understanding, identify additional impacts, and determine courses of action. Adjust responses as appropriate.
- **Perform operations metrics reviews**: Regularly perform retrospective analysis of operations metrics with cross-team participants from different areas of the business. Use these reviews to identify opportunities for improvement, potential courses of action, and to share lessons learned.
- **Document and share lessons learned**: Document and share lessons learned from the execution of operations activities so that you can use them internally and across teams.
- **Allocate time to make improvements**: Dedicate time and resources within your processes to make continuous incremental improvements possible.

Security

Identity and Access Management

SEC 1 How do you manage credentials and authentication?

Credentials and authentication mechanisms include passwords, tokens, and keys that grant access directly or indirectly in your workload. Protect credentials with appropriate mechanisms to help reduce the risk of accidental or malicious use.

Best practices:

- **Define identity and access management requirements**: Identity and access management configurations need to be defined to meet your organizational, legal, and compliance requirements.
- **Secure AWS root user**: Secure the AWS root user with MFA, no access keys, and limit its use to help secure your AWS account.
- **Enforce use of multi-factor authentication**: Enforce multi-factor authentication (MFA) with software or hardware mechanisms to provide additional access control.
- **Automate enforcement of access controls**: Enforce access controls through automated tools and by reporting irregularities. This helps you maintain your credential management requirements.
- **Integrate with centralized federation provider**: Integrate with a federated identity provider or directory service to authenticate all users in a centralized place. This reduces the requirement for multiple credentials and reduces management complexity.
- **Enforce password requirements**: Enforce policies for minimum length, complexity, and reuse of passwords to help protect against brute force and other password attacks.
- **Rotate credentials regularly**: Rotate credentials regularly to help reduce the risk of old credentials being used by unauthorized systems or users.
- **Audit credentials periodically**: Audit credentials to ensure the defined controls (for example, MFA) are enforced, rotated regularly, and have appropriate access level.

SEC 2 How do you control human access?

Control human access by implementing controls inline with defined business requirements to reduce risk and lower the impact of unauthorized access. This applies to privileged users and administrators of your AWS account, and also applies to end users of your application

Best practices:

- **Define human access requirements**: Clearly define access requirements for users based on job function to reduce the risks from unnecessary privileges.
- **Grant least privileges**: Grant users only the minimum privileges you have defined to reduce the risk of unauthorized access.
- **Allocate unique credentials for each individual**: Credentials are not shared between any users to help segregation of users and traceability.
- **Manage credentials based on user lifecycles**: Integrate access management with user lifecycle. For example, decommission a user to revoke unused and unnecessary credentials when a user leaves or changes roles.
- **Automate credential management**: Automate credential management to enforce minimum privileges and disable unused credentials. Automate auditing, reporting, and management of the lifecycle of users.
- **Grant access through roles or federation**: Use IAM roles instead of IAM users or static access keys to allow for secure cross-account access and federated users.

SEC 3 How do you control programmatic access?

Control programmatic or automated access with appropriately defined, limited, and segregated access to help reduce the risk of unauthorized access. Programmatic access includes access that is internal to your workload, and access to AWS related resources.

Best practices:

- **Define programmatic access requirements**: Clearly define access requirements for automated or programmatic access to reduce the risks from unnecessary privileges.
- **Grant least privileges**: Grant automated or programmatic access only the minimum privileges you have defined to reduce the risk of unauthorized access.
- **Automate credential management**: Automate credential management to enforce minimum privileges and disable unused credentials. Automate auditing, reporting, and management of dynamic authentication.
- **Allocate unique credentials for each component**: Credentials are not shared between any component to help segregation and traceability. For example, use different IAM roles for AWS Lambda functions and EC2 instances.
- **Grant access through roles or federation**: Use IAM roles or federation instead of IAM users or static access keys to allow for secure programmatic access.
- **Implement dynamic authentication**: Credentials are dynamically acquired and frequently rotated by a service or system.

Detective Controls

SEC 4 How do you detect and investigate security events?

Capture and analyze events from logs and metrics to gain visibility. Take action on security events and potential threats to help secure your workload.

Best practices:

- **Define requirements for logs**: Define requirements for retention and access control for logs to meet your organizational, legal, and compliance requirements.
- **Define requirements for metrics**: Collecting metrics and defining baselines allows you to gain insights to potential security threats.
- **Define requirements for alerts**: Define who should receive alerts and what they should do with the alerts they receive.
- **Configure service and application logging**: Configure logging throughout the workload, including application logs, AWS services logs, and resource logs.
- **Analyze logs centrally**: All logs should be collected centrally and automatically analyzed to detect anomalies and indicators of malicious activity or compromise.
- **Automate alerting on key indicators**: Key indicators, including metrics and events related to security, should be monitored and trigger automated alerts based on thresholds.
- **Develop investigation processes**: Develop processes to investigate different types of events, including escalation paths for incident response processes.

SEC 5 How do you defend against emerging security threats?

Staying up to date with AWS and industry best practices and threat intelligence helps you be aware of new risks. This enables you to create a threat model to identify, prioritize, and implement appropriate controls to help protect your workload.

Best practices:

- **Keep up to date with organizational, legal, and compliance requirements**: Stay up to date with organizational, legal, and compliance requirements that enable you to adjust your security posture to comply.
- **Keep up to date with security best practices**: Stay up to date with both AWS and industry security best practices to evolve protection of the workload.
- **Keep up to date with security threats**: Understand attack vectors by staying up to date with the latest security threats. This helps you implement detective and preventive controls.
- **Evaluate new security services and features regularly**: Evaluate security services from both AWS and APN Partners, including new features to limit risk of threats.
- **Define and prioritize risks using a threat model**: Use a threat model to identify and maintain an up to date register of potential threats. Prioritize your threats and adjust your security posture to respond.
- **Implement new security services and features**: Adopt security services and features to implement controls that help protect your workload.

Infrastructure Protection

SEC 6 How do you protect your networks?

Public and private networks require multiple layers of defense to help protect from external and internal network-based threats.

Best practices:

- **Define network protection requirements**: Define controls for protection of your networks to meet your organizational, legal, and compliance requirements.
- **Limit exposure**: Limit the exposure of the workload to the internet and internal networks by only allowing minimum required access.
- **Automate configuration management**: Enforce and validate secure configurations automatically by using a configuration management service or tool to reduce human error.
- **Automate network protection**: Automate protection mechanisms to provide a self defending network based on threat intelligence and anomaly detection.
- **Implement inspection and protection**: Inspect and filter your traffic at the application level; for example, by using a web application firewall, to help protect against threats.
- **Control traffic at all layers**: Apply controls for controlling both ingress and egress traffic, including data loss prevention. For Amazon Virtual Private Cloud (VPC) this includes security groups, Network ACLs, and subnets. For AWS Lambda, consider running in your private VPC to control traffic.

SEC 7 How do you protect your compute resources?

Compute resources in your workload require multiple layers of defense to help protect from external and internal threats. Compute resources include EC2 instances, containers, AWS Lambda functions, database services, IoT devices, and more.

Best practices:

- **Define compute protection requirements**: Define the protection controls for your compute resources to meet your organizational, legal, and compliance requirements.
- **Scan for and patch vulnerabilities**: Frequently scan for and patch vulnerabilities in your code base and in your infrastructure to help protect against new threats.
- **Automate configuration management**: Enforce and validate secure configurations automatically by using a configuration management service or tool to reduce human error.
- **Automate compute protection**: Automate defense with intrusion prevention, including the ability to protect against unknown threats; for example, use virtual patching.
- **Reduce attack surface**: Reduce the attack surface, including hardening EC2 operating systems and configuring container and serverless resources, to help limit exposure.
- **Implement managed services**: Implement services that manage resources, such as Amazon RDS, AWS Lambda, and Amazon ECS, to reduce security maintenance tasks.

Data Protection

SEC 8 How do you classify your data?

Classification provides a way to categorize data, based on levels of sensitivity, to help you determine appropriate protective and retention controls.

Best practices:

- **Define data classification requirements**: Define data classification requirements to meet your organizational, legal, and compliance requirements.
- **Define data protection controls**: Protect data according to its classification level; for example, secure publicly accessible data by using best practices while protecting sensitive data with additional controls.
- **Implement data identification**: Classify data with easily identifiable indicators; for example, use tags for Amazon S3 buckets and objects that classify the data in the buckets.
- **Automate identification and classification**: Automate identification and classification of data to reduce the risk of human error.
- **Identify the types of data**: Be aware of the types of data in your workload to help you implement controls to meet organizational, legal, and compliance requirements.

SEC 9 How do you protect your data at rest?

Protect your data at rest by defining your requirements and implementing controls, including encryption, to reduce the risk of unauthorized access or loss.

Best practices:

- **Define data management and protection at rest requirements**: Define data management and protection at rest requirements, such as encryption and data retention, to meet your organizational, legal, and compliance requirements.
- **Implement secure key management**: Encryption keys must be stored securely, and rotated with strict access control; for example, by using a key management service such as AWS Key Management Service. Consider using different keys for segregation of different data classification levels and retention requirements.
- **Enforce encryption at rest**: Enforce your defined encryption requirements based on the latest standards and best practices to help protect your data at rest.
- **Enforce access control**: Enforce access control with least privileges and mechanisms, including backups, isolation, and versioning, to help protect your data at rest. Consider what data you have that is publicly accessible.
- **Provide mechanisms to keep people away from data**: Keep all users away from directly accessing sensitive data. For example, provide a dashboard instead of direct access to a data store, and provide tools to indirectly manage the data.

SEC 10 How do you protect your data in transit?

Protecting your data in transit by defining your requirements and implementing controls, including encryption, reduces the risk of unauthorized access or exposure.

Best practices:

- **Define data protection in transit requirements**: Define data protection in transit requirements, such as encryption standards, based on data classification to meet your organizational, legal, and compliance requirements. Best practices are to encrypt and authenticate all traffic, and to enforce the latest standards and ciphers.
- **Implement secure key and certificate management**: Store encryption keys and certificates securely and rotate them with strict access control; for example, by using a certificate management service such as AWS Certificate Manager.
- **Enforce encryption in transit**: Enforce your defined encryption requirements based on the latest standards and best practices to help you meet your organizational, legal, and compliance requirements.
- **Automate detection of data leak**: Use a tool or detection mechanism to automatically detect attempts to move data outside of defined boundaries; for example, to detect a database system that is copying data to an unknown host.
- **Authenticate network communications**: Verify the identity of communications by using protocols, such as Transport Layer Security (TLS) or IPsec, to reduce the risk of data tampering or loss.

Incident Response

SEC 11 How do you respond to an incident?

Preparation is critical to timely investigation and response to security incidents to help minimize potential disruption to your organization.

Best practices:

- **Identify key personnel and external resources**: Identify internal and external personnel and resources that would help your organization respond to an incident.
- **Identify tooling**: Identify AWS, partner, and open source tools that would help your organization respond to an incident.
- **Develop incident response plans**: Create incident response plans, starting with the most likely scenarios for your workload and organization. Include how you would communicate and escalate both internally and externally.
- **Automate containment capability**: Automate containment of an incident to reduce response times and organizational impact.
- **Identify forensic capabilities**: Identify the forensic investigation capabilities that are available, including external specialists.
- **Pre-provision access**: Ensure that security personnel have the correct access pre-provisioned into AWS so that an appropriate response can be made to an incident.
- **Pre-deploy tools**: Ensure that security personnel have the right tools pre-deployed into AWS so that an appropriate response can be made to an incident.
- **Run game days**: Practice incident response game days (simulations) regularly, incorporate lessons learned into plans, and continuously improve responses and plans.

Reliability

Foundations

REL 1 How do you manage service limits?

Default service limits exist to prevent accidental provisioning of more resources than you need. There are also limits on how often you can call API operations to protect services from abuse. If you are using AWS Direct Connect, you have limits on the amount of data you can transfer on each connection. If you are using AWS Marketplace applications, you need to understand the limitations of the applications. If you are using third-party web services or software as a service, you also need to be aware of the limits of those services.

Best practices:

- **Aware of limits but not tracking them**: You are aware there are limits, but are not tracking your current limits.
- **Monitor and manage limits**: Evaluate your potential usage, increase your regional limits appropriately, and allow planned growth in usage.
- **Use automated monitoring and management of limits**: Implement tools to alert you when thresholds are being approached. You will want to use a distribution mechanism to alert a responsible group until the limit increase request can be automated.
- **Accommodate fixed service limits through architecture**: Be aware of unchangeable service limits and architect around these.
- **Ensure a sufficient gap between the current service limit and the maximum usage to accommodate failover**: When a resource fails it may still be counted against limits until it is successfully terminated. Ensure your limits cover the overlap of all failed resources with replacements before the failed resources are terminated. You should consider an Availability Zone failure when calculating this gap.
- **Manage service limits across all relevant accounts and regions**: If you are using multiple AWS accounts or AWS Regions, ensure you request the same limits in all environments in which you run your production workloads.

REL 2 How do you manage your network topology?

Applications can exist in one or more environments: your existing data center infrastructure, publicly accessible public cloud infrastructure, or private addressed public cloud infrastructure. Network considerations such as intra- and inter-system connectivity, public IP address management, private address management, and name resolution are fundamental to using resources in the cloud.

Best practices:

- **Use highly available connectivity between private addresses in public clouds and on-premises environment**: Use multiple AWS Direct Connect (DX) circuits and multiple VPN tunnels between separately deployed private IP address spaces. Use multiple DX locations for high availability. If you use multiple AWS Regions, you will also need multiple DX locations in at least 2 regions. You may want to evaluate AWS Marketplace appliances that terminate VPNs. If you use AWS Marketplace appliances, deploy redundant instances for high availability in different Availability Zones.
- **Use highly available network connectivity for the users of the workload**: Use a highly available DNS, CloudFront, API Gateway, load balancing, and reverse proxy as the public facing endpoint of your application. You may want to evaluate AWS Marketplace appliances for load balancing or proxying.
- **Enforce non-overlapping private IP address ranges in multiple private address spaces where they are connected**: The IP ranges of each of your VPCs must not conflict if they are peered or connected via VPN. The same is true for private connectivity to your on-premises environments and other cloud providers. You must have a way to allocate private IP ranges when needed.
- **Ensure IP subnet allocation accounts for expansion and availability**: Individual Amazon VPC IP address ranges must be large enough to accommodate an application's requirements, including factoring in future expansion and allocation of IP addresses to subnets across Availability Zones. This includes load balancers, AWS Lambda functions, EC2 instances, and container-based applications. Additionally, keep some IP addresses available for possible future expansion.

Change Management

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

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

Best practices:

- **Procure resources automatically when scaling a workload up or down**: Use services that automatically scale, such as Amazon S3, Amazon CloudFront, Amazon Auto Scaling, and AWS Lambda. You can also use third-party tools and AWS SDKs to automate scaling.
- **Procure resources upon detection of lack of service within a workload**: Scaling of resources is performed manually when availability is impacted.
- **Procure resources manually upon detection that more resources may be needed soon for a workload**: Manually scale compute and storage as need is anticipated.
- **Load test the workload**: Adopt a load testing methodology to measure if scaling activity will meet workload requirements.

REL 4 How do you monitor your resources?

Logs and metrics are a powerful tool to gain insight into the health of your workloads. You can configure your workload to monitor logs and metrics and send notifications when thresholds are crossed or significant events occur. Ideally, when low-performance thresholds are crossed or failures occur, the workload has been architected to automatically self-heal or scale in response.

Best practices:

- **Monitor the workload in all tiers**: Monitor the tiers of the workload with Amazon CloudWatch or third-party tools. Monitor AWS services with Personal Health Dashboard.
- **Send notifications based on the monitoring**: Organizations that need to know receive notifications when significant events occur.
- **Perform automated responses on events**: Use automation to take action when an event is detected; for example, to replace failed components.
- **Conduct reviews regularly**: Frequently review the monitoring of the workload based on significant events and changes to evaluate the architecture and implementation.

REL 5 How do you implement change?

Uncontrolled changes to your environment make it difficult to predict the effect of a change. Controlled changes to provisioned resources and workloads are necessary to ensure that the workloads and the operating environment are running known software and can be patched or replaced in a predictable manner.

Best practices:

- **Deploy changes in a planned manner**: Deployments and patching follow a documented process.
- **Deploy changes with automation**: Deployments and patching are automated.

Failure Management

REL 6 How do you back up data?

Back up data, applications, and operating environments (defined as operating systems configured with applications) to meet requirements for mean time to recovery (MTTR) and recovery point objectives (RPO).

Best practices:

- **Identify all data that needs to be backed up and perform backups or reproduce the data from sources**: Back up important data using Amazon S3, Amazon EBS snapshots, or third-party software. Alternatively, if the data can be reproduced from sources to meet RPO, you may not require a backup.
- **Perform data backup automatically or reproduce the data from sources automatically**: Automate backups or the reproduction from sources using AWS features (for example, snapshots of Amazon RDS and Amazon EBS, versions on Amazon S3, etc.), AWS Marketplace solutions, or third-party solutions.
- **Perform periodic recovery of the data to verify backup integrity and processes**: Validate that your backup process implementation meets Recovery Time Objective and Recovery Point Objective through a recovery test.
- **Secure and encrypt backups or ensure the data is available from a secure source for reproduction**: Detect access via authentication and authorization like AWS IAM, and detect data integrity compromise by using encryption.

REL 7 How does your system withstand component failures?

If your workloads have a requirement, implicit or explicit, for high availability and low mean time to recovery (MTTR), architect your workloads for resilience and distribute your workloads to withstand outages.

Best practices:

- **Monitor all layers of the workload to detect failures**: Continuously monitor the health of your system and report degradation as well as complete failure.
- **Implement loosely coupled dependencies**: Dependencies such as queuing systems, streaming systems, workflows, and load balancers are loosely coupled.
- **Implement graceful degradation to transform applicable hard dependencies into soft dependencies**: When a component's dependencies are unhealthy, the component itself does not report as unhealthy. It can continue to serve requests in a degraded manner.
- **Automating complete recovery because technology constraints exist in parts or all of the workload requiring a single location**: Elements of the workload can only run in one Availability Zone or one data center, requiring you to implement a complete rebuild of the workload with defined recovery objectives.
- **Deploy the workload to multiple locations**: Distribute workload load across multiple Availability Zones and AWS Regions (for example, DNS, ELB, Application Load Balancer, and API Gateway). These locations can be as diverse as needed.
- **Automate healing on all layers**: Use automated capabilities upon detection of failure to perform an action to remediate.
- **Send notifications upon availability impacting events**: Notifications are sent upon detection of significant events, even if the issue was automatically healed.

REL 8 How do you test resilience?

Test the resilience of your workload to help you find latent bugs that only surface in production. Exercise these tests regularly.

Best practices:

- **Use playbooks for unanticipated failures**: You have playbooks for failure scenarios that have not been anticipated to identify root causes and assist in strategies for prevention or mitigation.
- **Conduct root cause analysis (RCA) and share results**: Review system failures based on significant events to evaluate the architecture and identify the root cause. Have a method to communicate these causes to others as needed.
- **Inject failures to test resiliency**: Test failures regularly, ensuring coverage of failure pathways.
- **Conduct game days regularly**: Use game days to regularly exercise your failure procedures with the people who will be involved in actual failure scenarios.

REL 9 How do you plan for disaster recovery?

Disaster recovery (DR) is critical should restoration of data be required from backup methods. Your definition of and execution on the objectives, resources, locations, and functions of this data must align with RTO and RPO objectives.

Best practices:

- **Define recovery objectives for downtime and data loss**: The workload 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 objectives.
- **Test disaster recovery implementation to validate the implementation**: Regularly test failover to DR to ensure that RTO and RPO are met.
- **Manage configuration drift on all changes**: Ensure that AMIs and the system configuration state are up-to-date at the DR site or region, as well as the limits on AWS services.
- **Automate recovery**: Use AWS or third-party tools to automate system recovery.

Performance Efficiency

Selection

PERF 1 How do you select the best performing architecture?

Often, multiple approaches are required to get optimal performance across a workload. Well-architected systems use multiple solutions and enable different features to improve performance.

Best practices:

- **Understand the available services and resources**: Learn about and understand the wide range of services and resources available to you on AWS. Identify which services and configuration options are relevant to your workload and understand how to use them to achieve optimal performance.
- **Define a process for architectural choices**: Use existing internal experience and knowledge of AWS, or use external resources, such as published use cases, relevant documentation, or whitepapers to define a process to choose resources and services. For example, a process that encourages experimentation and benchmarking different services as they might be used in your workload.
- **Factor cost or budget into decisions**: Workloads often have budgets that they need to operate within, and the budget is a critical part of efficient operation. Use internal cost controls and consider budget when selecting resource types and sizes based on predicted resource needs.
- **Use policies or reference architectures**: Use internal policies or existing reference architectures to make the best architectural choices for your workload. Evaluate which services and configuration are best for your workload to maximize performance and efficiency.
- **Use guidance from AWS or an APN Partner**: Use AWS resources, such as Solutions Architects, or an APN Partner to guide your decisions. These resources can help review and suggest improvements to your architecture to achieve an optimal level of performance.
- **Benchmark existing workloads**: Benchmark the performance of an existing workload to understand how it performs on AWS. Use the data collected from these benchmarks to drive architectural decisions.
- **Load test your workload**: Deploy the latest version of your system on AWS using different resource types and sizes, and use monitoring to capture performance metrics that identify bottlenecks or excess capacity. Use this information when designing or improving your architecture and resource selection based on how it performs.

PERF 2 How do you select your compute solution?

The optimal compute solution for a system varies based on application design, usage patterns, and configuration settings. Architectures may use different compute solutions for various components and enable different features to improve performance. Selecting the wrong compute solution for an architecture can lead to lower performance efficiency.

Best practices:

- **Evaluate the available compute options**: Look at and understand the performance characteristics of the compute-related options available to you. Know how instances, containers, and functions work and what advantages, or disadvantages, they bring to your workload.
- **Understand the available compute configuration options**: Understand how various options complement your workload and which configuration options are best for your system. Examples of these options include instance family, sizes, features (GPU, I/O), function sizes, container instances, single versus multi-tenancy, and so on.
- **Collect compute-related metrics**: One of the best ways to understand how your systems are performing is to record and track the true utilization of various resources. This data can then be fed back into making more accurate determinations of resource requirements.
- **Determine the required configuration by right-sizing**: Analyze the various performance characteristics of your workload and how these characteristics relate to memory, network, and CPU usage. Use this data when choosing resources that best match your workload's profile. For example, a memory-intensive workload, such as a database, could be served best by the r-family of instances, while a bursting workload may benefit more from an elastic container system such as Amazon Elastic Container Service.
- **Use the available elasticity of resources**: AWS provides the flexibility to expand or reduce your resources dynamically through a variety of mechanisms (for example: AWS Auto Scaling, Amazon Elastic Container Service, and AWS Lambda) to meet changes in demand. Combined with compute-related metrics, a workload can automatically respond to these changes and utilize the optimal set of resources to achieve its goal.
- **Re-evaluate compute needs based on metrics**: Use system-level metrics to identify the behavior and requirements of your workload over time. Evaluate your workload's needs by comparing the available resources with these requirements and make changes to your compute environment to best match your workload's profile. For example, over time a system may be observed to be more memory-intensive than initially thought, so moving to a different instance family or size may improve both performance and efficiency.

PERF 3 How do you select your storage solution?

The optimal storage solution for a system varies based on the kind of access method (block, file, or object), patterns of access (random or sequential), required throughput, frequency of access (online, offline, archival), frequency of update (WORM, dynamic), and availability and durability constraints. Well-architected systems use multiple storage solutions and enable different features to improve performance and use resources efficiently.

Best practices:

- **Understand storage characteristics and requirements**: Understand the different characteristics (for example, shareable, file size, cache size, access patterns, latency, throughput, and persistence of data) that are required to select the services that best fit your workload, such as Amazon S3, Amazon EBS, Amazon Elastic File System (Amazon EFS), and Amazon EC2 instance store.
- **Evaluate available configuration options**: Evaluate the various characteristics and configuration options and how they relate to storage. Understand where and how to use PIOPS, SSDs, magnetic storage, Amazon S3, Amazon Glacier, or ephemeral storage to optimize storage space and performance for your workload.
- **Make decisions based on access patterns and metrics**: Choose storage systems and configure them by considering how the workload accesses data. Make performance improvements, such as choosing caching services or instances that best match your access patterns, utilizing optimal key distributions when storing data in Amazon S3 or DynamoDB, striping storage volumes, or partitioning data based on system measurements. Increase storage efficiency by choosing object storage, such as Amazon S3, or block storage, such as Amazon Elastic Block Store. Configure the storage options you choose to match your data access patterns.

PERF 4 How do you select your database solution?

The optimal database solution for a system varies based on requirements for availability, consistency, partition tolerance, latency, durability, scalability, and query capability. Many systems use different database solutions for various sub-systems and enable different features to improve performance. Selecting the wrong database solution and features for a system can lead to lower performance efficiency.

Best practices:

- **Understand data characteristics**: Understand the different characteristics of data in your workload. Determine if the workload requires transactions, how it interacts with data, what its performance demands are, and so on. Use this data to select the best performing database approach for your workload (for example, relational databases, NoSQL, data warehouses, or in-memory storage).
- **Evaluate the available options**: Evaluate the services and storage options that are available as part of the selection process for your workload's storage mechanisms. Understand how, and when, to use a given service or system for data storage. Learn about available configuration options that can further optimize database performance or efficiency, such as PIOPs, memory and compute resources, caching, and so on.
- **Collect and record database performance metrics**: Use tools, libraries, and systems that record performance measurements related to database performance. For example, measure transactions per second, slow queries, or system latency introduced when accessing the database. Use this data to understand the performance of your database systems.
- **Choose data storage based on access patterns**: Use the access patterns of the workload to decide which services and technologies to use. For example, utilize a relational database for workloads that require transactions, or a key-value store that provides higher throughput but is eventually consistent where applicable.
- **Optimize data storage based on access patterns and metrics**: Use performance characteristics and access patterns that optimize how data is stored or queried to achieve the best possible performance. Measure how optimizations such as indexing, key distribution, data warehouse design, or caching strategies affect system performance or overall efficiency.

PERF 5 How do you configure your networking solution?

The optimal network solution for a system varies based on latency, throughput requirements, and so on. Physical constraints such as user or on-premises resources drive location options, which can be offset using edge techniques or resource placement.

Best practices:

- **Understand how networking impacts performance**: Analyze and understand how network-related decisions impact workload performance. For example, network latency often impacts the user experience, and using the wrong protocols can starve network capacity through excessive overhead.
- **Understand available product options**: Understand the service-level features that are available to optimize network-related performance; for example, EC2 instance network capability, enhanced networking, Amazon EBS-optimized instances, Amazon S3 Transfer Acceleration, and dynamic content delivery with Amazon CloudFront.
- **Evaluate available networking features**: Evaluate networking features in AWS that increase performance. Measure the impact of these features through testing, metrics, and analysis. For example, take advantage of network-level features that are available (including Amazon Route 53 latency-based routing, Amazon VPC endpoints, or AWS Direct Connect) to reduce latency, network distance, or jitter.
- **Use minimal network ACLs**: Design your network to minimize the number of ACLs while still meeting requirements. Having too many ACLs can negatively impact network performance, reducing system performance or efficiency.
- **Leverage encryption offloading and load-balancing**: Use load balancing for offloading encryption termination (TLS) to improve performance and to manage and route traffic effectively. Distribute traffic across multiple resources or services to allow your workload to take advantage of the elasticity that AWS provides.
- **Choose network protocols to improve performance**: When choosing protocols for communication between systems and networks, make decisions based on how those protocols will impact workload performance.
- **Choose location based on network requirements**: Use the location options available (for example, AWS Region, Availability Zone, placement groups, and edge locations) to reduce network latency or improve throughput.
- **Optimize network configuration based on metrics**: Use data that is collected and analyzed to make informed decisions about optimizing your network configuration. Measure the impact of those changes and use the impact measurements to make future decisions.

Review

PERF 6 How do you evolve your workload to take advantage of new releases?
When architecting workloads, there are finite options that you can choose from. However, over time, new technologies and approaches become available that could improve the performance of your workload.

Best practices:

- **Keep up-to date on new resources and services**: Evaluate ways to improve performance as new services, design patterns, or product offerings become available. Consider how these could improve performance or increase the efficiency of the workload through ad-hoc evaluation, internal discussion, or external analysis.
- **Define a process to improve workload performance**: Define a process to evaluate new services, design patterns, resource types, and configurations as they become available. For example, run existing performance tests on new instance offerings to determine which improvements in performance or efficiency would be gained by using them.
- **Evolve workload performance over time**: As an organization, use the information gathered through the evaluation process to actively drive adoption of new services or resources when they are available to improve performance or make efficiency gains in your workload.

Monitoring

PERF 7 How do you monitor your resources to ensure they are performing as expected?

System performance can degrade over time. Monitor system performance to identify this degradation and remediate internal or external factors, such as the operating system or application load.

Best practices:

- **Record performance-related metrics**: Use Amazon CloudWatch, a third-party service, or self-managed monitoring tools to record performance-related metrics. For example, record database transactions, slow queries, I/O latency, HTTP request throughput, service latency, or other key data.
- **Analyze metrics when events or incidents occur**: In response to (or during) an event or incident, use monitoring dashboards or reports to understand and diagnose the impact. These views provide insight into which portions of the workload are not performing at expected levels.
- **Establish KPIs to measure workload performance**: Identify the KPIs that indicate whether the system is performing as intended. For example, an API-based workload may use overall response latency as an indication of overall performance, and an e-commerce site may choose to use the number of purchases being made as its KPI.
- **Use monitoring to generate alarm-based notifications**: Using the performance-related KPIs that you defined, use a monitoring system that generates alarms automatically when these measurements are outside expected boundaries.
- **Review metrics at regular intervals**: As routine maintenance or in response to events or incidents, review which metrics are collected. Use these reviews to identify which metrics were key in addressing issues and which additional metrics, if they were being tracked, would help to identify, address, or prevent issues.
- **Monitor and alarm proactively**: Use KPIs, combined with monitoring and alerting systems, to proactively address performance-related issues. Use alarms to trigger automated actions to remediate issues where possible; escalate the alarm to those able to respond if automated response is not possible. For example, a system that can predict expected KPI values and alarm when they breach certain thresholds, or a tool that can automatically halt or roll back deployments if KPIs are outside of expected values.

Tradeoffs

PERF 8 How do you use tradeoffs to improve performance?
When architecting solutions, actively considering tradeoffs enables you to select an optimal approach. Often you can improve performance by trading consistency, durability, and space for time and latency.

Best practices:

- **Understand the areas where performance is most critical**: Understand and identify areas where increasing the performance of your workload will have a positive impact on efficiency or the customer experience. For example, a website that has a lot of customer interaction would benefit from using edge services such as Amazon CloudFront to move content delivery closer to customers.
- **Learn about design patterns and services**: Research and understand the various design patterns and services that help improve workload performance. As part of the analysis, identify what you may be trading to achieve higher performance. For example, using Amazon ElastiCache can help to reduce load placed on database systems; however, it requires some engineering to implement safe caching, or possible introduction of eventual consistency in some areas.
- **Identify how tradeoffs impact customers and efficiency**: When evaluating performance-related improvements, consider how those choices will impact customers and workload efficiency. For example, if using key-value storage such as Amazon DynamoDB would significantly increase system performance, it is also important to evaluate how the eventually consistent nature of Amazon DynamoDB could affect customers.
- **Measure the impact of performance improvements**: As changes are made to improve performance, evaluate metrics and data that were collected to determine the impact that the performance improvement had on the workload, its components, and any customers. This measurement helps you understand the improvements that result from the tradeoff, and helps you determine if any negative side-effects were introduced.
- **Use various performance-related strategies**: Where applicable, utilize a number of strategies to improve performance. For example, use strategies like caching data to prevent excessive network or database calls, using read-replicas for database engines to improve read rates, sharding or compressing data where possible to reduce data volumes, and buffering and streaming of results as they are available to avoid blocking.

Cost Optimization

Expenditure Awareness

COST 1 How do you govern usage?

Establish policies and mechanisms to ensure that appropriate costs are incurred while objectives are achieved. By employing a checks-and-balances approach, you can innovate without overspending.

Best practices:

- **Develop policies based on your organization requirements**: Develop policies that define how resources are managed by your organization. Policies should cover cost aspects of resources and workloads, including creation, modification and decommission over the resource lifetime. Also develop cost targets and goals for workloads.
- **Implement an account structure**: Implement a structure of accounts that maps to your organization. This assists in allocating and managing costs throughout your organization.
- **Implement groups and roles**: Implement groups and roles that align to your policies and control who can create, modify, or decommission instances and resources in each group; for example, development, test, and production groups. This applies to AWS services and third-party solutions.
- **Implement cost controls**: Implement controls based on organization policies and defined groups and roles. These ensure that costs are only incurred as defined by organization requirements; for example, control access to regions or resource types with IAM policies.
- **Track project lifecycle**: Track, measure, and audit the lifecycle of projects, teams, and environments to avoid using and paying for unnecessary resources.

COST 2 How do you monitor usage and cost?

Establish policies and procedures to monitor and appropriately allocate your costs. This allows you to measure and improve the cost efficiency of this workload.

Best practices:

- **Configure AWS Cost and Usage Report**: Configure the AWS Cost and Usage Report to capture detailed usage and billing information.
- **Identify cost attribution categories**: Identify organization categories that could be used to allocate cost within your organization.
- **Establish organization metrics**: Establish the organization metrics that are required for this workload. Example metrics of a workload are customer reports produced or web pages served to customers.
- **Define and implement tagging**: Define a tagging schema based on organization, and workload attributes, and cost allocation categories. Implement tagging across all resources.
- **Configure billing and cost management tools**: Configure AWS Cost Explorer and AWS Budgets inline with your organization policies.
- **Report and notify on cost optimization**: Configure AWS Budgets to provide notifications on cost and usage against targets. Have regular meetings to analyze this workload's cost efficiency and to promote cost aware culture.
- **Monitor cost proactively**: Implement tooling and dashboards to monitor cost proactively for this workload; do not just look at costs and categories when you receive notifications. This helps to identify positive trends and promote them throughout your organization.
- **Allocate costs based on workload metrics**: Allocate this workload's costs by metrics or business outcomes to measure workload cost efficiency. Implement a process to analyze the AWS Cost and Usage Report with Amazon Athena, which can provide insight and charge back capability.

COST 3 How do you decommission resources?

Implement change control and resource management from project inception to end-of-life. This ensures you shut down or terminate unused resources to reduce waste.

Best practices:

- **Track resources over their life time**: Define and implement a method to track resources and their associations with systems, over their life time. You can use tagging to identify the workload or function of the resource.
- **Implement a decommissioning process**: Implement a process to identify and decommission orphaned resources.
- **Decommission resources in an unplanned manner**: Decommission resources on an unplanned basis. This is typically triggered by events such as periodic audits and is usually performed manually.
- **Decommission resources automatically**: Design your workload to gracefully handle resource termination as you identify and decommission non-critical resources, resources that are not required, or resources with low utilization.

Cost-Effective Resources

COST 4 How do you evaluate cost when you select services?

Amazon EC2, Amazon EBS, and Amazon S3 are building-block AWS services. Managed services, such as Amazon RDS and Amazon DynamoDB, are higher level, or application level, AWS services. By selecting the appropriate building blocks and managed services, you can optimize this workload for cost. For example, using managed services, you can reduce or remove much of your administrative and operational overhead, freeing you to work on applications and business-related activities.

Best practices:

- **Identify organization requirements for cost**: Work with team members to define the balance between cost optimization and other pillars, such as performance and reliability, for this workload.
- **Analyze all components of this workload**: Ensure every workload component is analyzed, regardless of current size or current costs. Review effort should reflect potential benefit, such as current and projected costs.
- **Perform a thorough analysis of each component**: Look at overall cost to the organization of each component. Look at total cost of ownership by factoring in cost of operations and management, especially when using managed services. Review effort should reflect potential benefit; for example, time spent analyzing is proportional to component cost.
- **Select components of this workload to optimize cost inline with organization priorities**: Factor in cost when selecting all components. This includes using application level and managed services such as Amazon RDS, Amazon DynamoDB, Amazon SNS, and Amazon SES to reduce overall organization cost. Use serverless and containers for compute, such as AWS Lambda, Amazon S3 for static websites, and Amazon ECS. Minimize license costs by using open source software, or software that does not have license fees. For example Amazon Linux for compute workloads, or migrate databases to Amazon Aurora.
- **Perform cost analysis for different usage over time**: Workloads can change over time, and some services or features are more cost effective at different usage levels. By performing the analysis on each component over time and at projected usage, you ensure this workload remains cost effective over its lifetime.

COST 5 How do you meet cost targets when you select resource type and size?

Ensure that you choose the appropriate resource size for the task at hand. By selecting the most cost effective type and size, you minimize waste.

Best practices:

- **Perform cost modeling**: Identify organization requirements and perform cost modeling of the workload and each of its components. Perform benchmark activities for the workload under different predicted loads and compare the costs. The modeling effort should reflect potential benefit; for example, time spent is proportional to component cost.
- **Select resource type and size based on estimates**: Estimate resource size or type based on workload and resource characteristics; for example, compute, memory, throughput, or write intensive. This estimate is typically made using a previous version of the workload (such as an on-premises version), using documentation, or using other sources of information about the workload.
- **Select resource type and size based on metrics**: Use metrics from the currently running workload to select the right size and type to optimize for cost. Appropriately provision throughput, sizing, and storage for services such as Amazon EC2, Amazon DynamoDB, Amazon EBS (PIOPS), Amazon RDS, Amazon EMR, and networking. This can be done with a feedback loop such as automatic scaling, or by a manual process of re-sizing.

COST 6 How do you use pricing models to reduce cost?

Use the pricing model that is most appropriate for your resources to minimize expense.

Best practices:

- **Perform pricing model analysis**: Perform an analysis on the workload using the Reserved Instance Recommendations feature in AWS Cost Explorer.
- **Implement different pricing models, with low coverage**: Implement reserved capacity, Spot Instances, Spot Blocks or Spot Fleet, in the workload but with low coverage, at less than 80 percent of overall recommendations.
- **Implement regions based on cost**: Resource pricing can be different in each region. Factoring in region cost ensures you pay the lowest overall price for this workload.
- **Implement pricing models for all components of this workload**: Permanently running resources have high coverage with reserved capacity, with at least 80 percent of recommendations implemented. Short term capacity is configured to use Spot Instances, Spot Blocks or Spot Fleet. On demand is only used for short-term workloads that cannot be interrupted, and do not run long enough for reserved capacity: typically 25 to 75 percent of the year, depending on the resource type.

COST 7 How do you plan for data transfer charges?

Ensure that you plan and monitor data transfer charges so that you can make architectural decisions to minimize costs. A small yet effective architectural change can drastically reduce your operational costs over time.

Best practices:

- **Perform data transfer modeling**: Gather organization requirements and perform data transfer modeling of the workload and each of its components. This identifies the lowest cost point for its current data transfer requirements.
- **Select components to optimize data transfer cost**: All components are selected, and architecture is designed to reduce data transfer costs. This includes using components such as WAN optimization and Multi-AZ configurations.
- **Implement services to reduce data transfer costs**: Implement services to reduce data transfer; for example, using a CDN such as Amazon CloudFront to deliver content to end users, caching layers using Amazon ElastiCache, or using AWS Direct Connect instead of VPN for connectivity to AWS.

Matching supply and demand

COST 8 How do you match supply of resources with demand?

For a workload that has balanced spend and performance, ensure that everything you pay for is used and avoid significantly underutilizing instances. A skewed utilization metric in either direction has an adverse impact on your organization, in either operational costs (degraded performance due to over-utilization), or wasted AWS expenditures (due to over-provisioning).

Best practices:

- **Perform an analysis on the workload demand**: Analyze the demand of the workload over time. Ensure the analysis covers seasonal trends and accurately represents operating conditions over the full workload lifetime. Analysis effort should reflect potential benefit; for example, time spent is proportional to the workload cost.
- **Provision resources reactively or unplanned**: Resource levels change due to demand, however provisioning is in an unplanned manner, typically manually, and triggered by adverse events or changes in the workload. Resourcing is slow to change, and typically results in over or under provisioning.
- **Provision resources dynamically**: Resources are provisioned in a planned manner. This can be demand-based, such as through automatic scaling; buffer-based, where demand is spread over time with lower overall resourcing used; or time-based, where demand is predictable and resources are provided based on time. These methods result in the least amount of over or under provisioning.

Optimizing Over Time

COST 9 How do you evaluate new services?

As AWS releases new services and features, it is a best practice to review your existing architectural decisions to ensure they continue to be the most cost effective.

Best practices:

- **Establish a cost optimization function**: Create a team that regularly reviews cost and usage across the organization.
- **Develop a workload review process**: Develop a process that defines the criteria and process for workload review. The review effort should reflect potential benefit; for example, core workloads or workloads with a value of over 10% of the bill are reviewed quarterly, while workloads below 10% are reviewed annually.
- **Review and implement services in an unplanned way**: Adopt new services in an unplanned way.
- **Review and analyze this workload regularly**: Existing workloads are regularly reviewed as per defined processes.
- **Keep up to date with new service releases**: Consult regularly with experts or APN Partners to consider which services and features provide lower cost. Review AWS blogs and other information sources.