AWS Design Principles:

There are five 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 run process by initiating them in response to events. By performing operations as code, you limit human error and achieve consistent responses to events.
- Make frequent, small, reversible changes: Design workloads to permit 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 verify that they are effective, and that teams are familiar with their process. 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.

6 Pillars of AWS well architected framework:

1. Operational Excellence

The Operational Excellence pillar includes the ability to support development and run workloads effectively, gain insight into their operation, and continuously improve supporting processes and procedures to delivery business value. You can find prescriptive guidance on implementation in the Operational Excellence Pillar whitepaper.

Design Principles

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

- Perform operations as code
- Make frequent, small, reversible changes
- Refine operations procedures frequently
- Anticipate failure

- Learn from all operational failures

Best Practices

Operations teams need to understand their business and customer needs so they can support business outcomes. Ops creates and uses procedures to respond to operational events, and validates their effectiveness to support business needs. Ops also collects metrics that are used to measure the achievement of desired business outcomes.

Everything continues to change—your business context, business priorities, and customer needs. It's important to design operations to support evolution over time in response to change, and to incorporate lessons learned through their performance.

2. Security

The Security pillar includes the ability to protect data, systems, and assets to take advantage of cloud technologies to improve your security. 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
- Enable traceability
- Apply security at all layers
- Automate security best practices
- Protect data in transit and at rest
- Keep people away from data
- Prepare for security events

Best Practices

Before you architect any workload, you need to put in place practices that influence security. You'll 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.

3. Reliability

The Reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it's expected to. This includes the ability to operate and test the workload through its total lifecycle. You can find prescriptive guidance on implementation in the Reliability Pillar whitepaper.

Design Principles

There are five design principles for reliability in the cloud:

- Automatically recover from failure
- Test recovery procedures

- Scale horizontally to increase aggregate workload availability
- Stop guessing capacity
- Manage change in automation

Best Practices

Before building 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). With AWS, however, most of the foundational requirements are already incorporated or can be addressed as needed.

The cloud is designed to be nearly limitless, so it's the responsibility of AWS to satisfy the requirement for sufficient networking and compute capacity, leaving you free to change resource size and allocations on demand.

A reliable workload starts with upfront design decisions for both software and infrastructure. Your architecture choices will impact your workload behavior across all six AWS Well-Architected pillars. For reliability, there are specific patterns you must follow, such as loosely coupled dependencies, graceful degradation, and limiting retries.

Changes to your workload or its environment must be anticipated and accommodated to achieve reliable operation of the workload. Changes include those imposed on your workload, like a spikes in demand, as well as those from within such as feature deployments and security patches.

Low-level hardware component failures are something to be dealt with every day in an on-premises data center. In the cloud, however, these are often abstracted away. Regardless of your cloud provider, there is the potential for failures to impact your workload. You must therefore take steps to implement resiliency in your workload, such as fault isolation, automated failover to healthy resources, and a disaster recovery strategy.

4. 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. 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
- Go global in minutes
- Use serverless architectures
- Experiment more often
- Consider mechanical sympathy

Best Practices

Take a data-driven approach to building 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.

Reviewing your choices on a regular basis ensures you are taking advantage of the continually evolving AWS Cloud. Monitoring ensures you are aware of any deviance from expected performance. Make trade-offs in your architecture to improve performance, such as using compression or caching, or relaxing consistency requirements

The optimal solution for a particular workload varies, and solutions often combine multiple approaches. AWS Well-Architected workloads use multiple solutions and enable different features to improve performance

5. Cost Optimization

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

- Implement cloud financial management
- Adopt a consumption model
- Measure overall efficiency
- Stop spending money on undifferentiated heavy lifting
- Analyze and attribute expenditure

Best Practices

As with the other pillars, there are trade-offs to consider. For example, do you want to optimize for speed to market or for cost? In some cases, it's best to optimize for speed—going to market quickly, shipping new features, or simply meeting a deadline—rather than investing in up-front cost optimization.

Design decisions are sometimes directed by haste rather than data, and as the temptation always exists to overcompensate rather than spend time benchmarking for the most cost-optimal deployment. This might lead to over-provisioned and under-optimized deployments.

Using the appropriate services, resources, and configurations for your workloads is key to cost savings

6. Sustainability

The discipline of sustainability addresses the long-term environmental, economic, and societal impact of your business activities. You can find prescriptive guidance on implementation in the Sustainability Pillar whitepaper.

Design Principles

There are six design principles for sustainability in the cloud:

- Understand your impact
- Establish sustainability goals
- Maximize utilization
- Anticipate and adopt new, more efficient hardware and software offerings
- Use managed services
- Reduce the downstream impact of your cloud workloads

Best Practices

Choose AWS Regions where you will implement workloads based on your business requirements and sustainability goals.

User behavior patterns can help you identify improvements to meet sustainability goals. For example, scale infrastructure down when not needed, position resources to limit the network required for users to consume them, and remove unused assets.

Implement software and architecture patterns to perform load smoothing and maintain consistent high utilization of deployed resources. Understand the performance of your workload components, and optimize the components that consume the most resources.

Analyze data patterns to implement data management practices that reduce the provisioned storage required to support your workload. Use lifecycle capabilities to move data to more efficient, less performant storage when requirements decrease, and delete data that's no longer required.

Analyze hardware patterns to identify opportunities that reduce workload sustainability impacts by minimizing the amount of hardware needed to provision and deploy. Select the most efficient hardware for your individual workload.

In your development and deployment process, identify opportunities to reduce your sustainability impact by making changes, such as updating systems to gain performance efficiencies and manage sustainability impacts. Use automation to manage the lifecycle of your development and test environments, and use managed device farms for testing.

Adding extra points:

1. Always Scalable
    - Automated - using autoscaling group
    - Manual
2. Disposable servers: You can easily get rid of servers if not required
3. Automation
    - Automate infra
    - Use ansible, terraform
    - Use cloud formation
    - Automate task
4. Loose Coupling: The components of the applications called elements depend upon each other to the least extent practicable.
5. Service not servers: Talk in terms of service and not servers
6. Diverse databases: Use databases based on the use cases
7. Managing increasing volumes of data
8. Optimize cost
9. Caching
10. Security