AWS Solution Architect Associate Certification Training – Module 30

30. Well-Architected Framework: The 5 Pillars of the AWS

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 architecturewill help you produce stable and efficient systems. This will allow you to focus on the other aspects of design, such as functional requirements.

1. 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.

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.

2. Security

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

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.

3. 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.

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.

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.

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.

5. Cost Optimization

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

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 workloadowners. 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.