AWS Well-Architected Framework

Week 7.2

What you will Learn

- Describe the AWS Well-Architected Framework.
- Describe the features of the Well-Architected Framework.
- Explore the six pillars of the Well-Architected Framework.

AWS Well-Architected Framework

- Creating a software system is a lot like constructing a building. Structural issues can compromise the building’s stability and functionality if the foundation is not solid.
- The AWS Well-Architected Framework documents a set of foundational questions that enable you to understand whether 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.

AWS Well-Architected Framework

- The Well-Architected Framework helps cloud architects assess and improve their architectures and get a better understanding of how their design decisions can impact their business.
- It provides a set of questions developed by AWS experts to help customers think critically about their architecture, such as "Does your infrastructure follow best practices?"
- As cloud technologies continue to evolve, and as AWS continues to learn more from working with customers, the definition of well-architected is continually being improved and refined.

Well-Architected Framework Pillars

- AWS Well-Architected Framework helps you design your architecture from six different perspectives or pillars.
- The pillars are Operational Excellence, Security, Reliability, Performance Efficiency, Cost Optimization and Sustainability.
- We would learn about each pillar in more detail and discuss the design principles for each pillar.

Operational Excellence

- The Operational Excellence pillar focuses on running workloads effectively, gaining insights into systems operations to deliver business value and continually improving supporting processes and procedures.
- Key topics include managing and automating changes, responding to events, and defining standards to successfully manage daily operations.
- Some 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

Operational Excellence Design Principles

- Perform operations as code
  - Define your entire workload (that is, applications and infrastructure) as code and update it with code.
  - Implement operations procedures as code and configure them to automatically trigger in response to events.
  - By performing operations as code, you limit human error and enable consistent responses to events.
- Make frequent, small, reversible changes
  - Design workloads to enable components to be updated regularly.
  - Make changes in small increments that can be reversed if they fail (without affecting customers when possible).

Operational Excellence Design Principles

- Refine operations procedures frequently
  - Look for opportunities to improve operations procedures.
  - Evolve your procedures appropriately as your workloads evolve.
  - Set up regular game days to review all procedures, validate their effectiveness, and ensure that teams are familiar with them.
  - A game day simulates a failure or event to test systems, processes, and team responses.
- Anticipate failure
  - Identify potential sources of failure so that they can be removed or mitigated.
  - Test 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.

Operational Excellence Design Principles

- 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.
- To learn more, refer to the Operational Excellence Pillar whitepaper

Security

- The Security pillar involves the ability to monitor and protect systems while delivering business value through risk assessments and mitigation strategies.
- Key topics include protecting the confidentiality and integrity of data, identifying and managing who can do what (or privilege management), protecting systems, and establishing controls to detect security events.
- Some design principles for security in the cloud include:
  - Apply security at all layers.
  - Enable Traceability.
  - Implement the principle of least privilege.
  - Secure your system.
  - Automate security best practices.

Security Design Principles

- Apply security at all layers
  - You want to make sure that you have multiple layers of defense by securing your infrastructure everywhere and at every layer.
  - Implement security within and between your resources.
  - This ensures that your environment and components are secured from each other as well.
- Enable traceability
  - Enable traceability through logging and auditing all actions or changes to your environment.
- Implement the principle of least privilege
  - Make sure that authorization within your environment is adequate.
  - Also, make sure that you are implementing strong logical access controls to your AWS resources that grant the minimum permissions that are needed for business requirements.

Security Design Principles

- Secure your system
  - Focus on securing your system.
  - With the AWS shared responsibility model, you can focus clearly on securing your application, data, and operating systems.
- Automate security best practices
  - Software-based security mechanisms improve your ability to securely scale more rapidly and cost-effectively.
  - For example, create and save a patched, hardened image of a virtual server so that when you need an image, you can use that image automatically to create a new instance.
  - Another best practice is to automate the response to both routine and anomalous security events.
- To learn more refer to the Security Pillar whitepaper

Reliability

- The Reliability pillar is concerned with the ability of a system to recover from infrastructure or service failures and to dynamically acquire computing resources to meet demand and mitigate disruptions.
- It encompasses the ability of a workload to perform its intended function correctly and consistently when it’s expected to.
- Reliability can help you recover from failures and meet demand.
- Reliability in the cloud comprises three areas, careful evaluation of each of these areas will enable you to anticipate, respond to, and prevent failures;
  - Foundations.
  - Change management.
  - Failure management.

Reliability Areas

- Foundations
  - To achieve reliability, your architecture and system must have a well-planned foundation that can handle changes in demand, or with requirements, and also detect a failure and automatically heal itself.
  - Before architecting any sort of structure, it is critical to look at the foundation, foundational requirements that influence reliability should be in place.
- Change management
  - With change management, it is important to fully understand how change can affect your system.
  - If you plan proactively and monitor your systems, you can accommodate change and adjust to it quickly and reliably.

Reliability Areas

- Failure management
  - To make sure that your architecture is reliable, it is key to anticipate, become aware of, respond to, and prevent failures from happening.
  - In a cloud environment, you can take advantage of automation with monitoring, replacing systems in your environment, and later troubleshooting failed systems—all at a lower cost, all while still being reliable.

Reliability Design Principles

- Some design principles for reliability in the cloud include:
  - Testing recovery procedures.
  - Automatically recover from failure.
  - Scale horizontally to increase aggregate system availability.
  - Stop guessing capacity.
  - Manage change in automation.
- Testing recovery procedures
  - In the cloud, users can test how systems fail, and they can validate their recovery procedures.
  - Users can simulate and expose different failures, and then resolve them before a real failure occurs.

Reliability Design Principles

- Automatically recover from failure
  - In the AWS Cloud, users can trigger automated responses when thresholds are breached.
  - It is thus possible to anticipate and remediate failures before they occur.
- Scale horizontally to increase aggregate system availability
  - When you have one large resource, it is beneficial to replace that large resource with multiple small resources to reduce the impact of a single point of failure on the overall system.
  - The goal is to scale horizontally and distribute requirements among multiple small resources.

Reliability Design Principles

- Stop guessing capacity
  - In the cloud environment, you can monitor demand and system utilization, and automate the addition or removal of resources.
  - This ensures that you have the optimal level to satisfy your demand without over-provisioning or under-provisioning.
- Manage change in automation
  - Changes to your architectures and infrastructure should be made using automation.
  - With this, you only need to manage change to your automation, not each system or resource.
- To learn more refer to the Reliability Pillar whitepaper

Performance Efficiency

- The Performance Efficiency pillar refers to using computing resources efficiently while meeting system requirements.
- At the same time, it is important to maintain that efficiency as demand fluctuates and technologies evolve.
- Factors that influence performance efficiency in the cloud include:
  - Selection.
  - Review.
  - Monitoring.
  - Tradeoffs.

Performance Efficiency

- Selection
  - It is important to choose the best solution that will optimize your architecture.
  - Solutions vary based on the kind of workload you have, and AWS enables you to customize your solutions in many different ways and configurations.
- Review
  - You can continually innovate your solutions and take advantage of the newer technologies and approaches that become available.
  - Any of these newer releases could improve the performance efficiency of your architecture.

Performance Efficiency

- Monitoring
  - After you implement your architecture, you must monitor performance to ensure that you can remediate any issues before customers are affected and aware of them.
  - With AWS, you can use automation and monitor your architecture with tools such as Amazon CloudWatch, Amazon Kinesis, Amazon Simple Queue Service (Amazon SQS), and AWS Lambda.
- Tradeoffs
  - An example of a tradeoff that ensures an optimal approach is trading consistency, durability, and space against time or latency to deliver higher performance.

Performance Efficiency Design Principles

- Some design principles for performance efficiency in the cloud include:
  - Democratize advanced technologies.
  - Go global in minutes.
  - Use serverless architectures.
  - Experiment more often.
  - Have mechanical sympathy.
- Democratize advanced technologies
  - Technologies that are difficult to implement can become simpler to consume by pushing that knowledge and complexity into the cloud vendor's domain.
  - Instead of having your IT team learn how to host and run a new technology, they can consume it as a service.

Performance Efficiency Design Principles

- Go global in minutes
  - With AWS, you can easily deploy your system in multiple AWS Regions around the world while providing lower latency and a better experience for your customers at a minimal cost.
- Use serverless architectures
  - Serverless computing is a cloud computing runtime model where the cloud provider dynamically manages the allocation of machine resources.
  - Pricing is based on the actual amount of resources consumed by an application, instead of on pre-purchased units of capacity.
  - In the cloud, serverless computing enables you to reduce the need to run and maintain traditional servers for compute activities.
  - It also removes the operational burden and can lower transactional costs.

Performance Efficiency Design Principles

- Experiment more often.
  - With virtualization, you can quickly carry out testing to enhance efficiency.
- Have mechanical sympathy.
  - Mechanical sympathy is when you use a tool or system with an understanding of how it operates best.
  - This principle suggests that you use the technology approach that best aligns with what you are trying to achieve.
  - "Understanding how a car works make you a better driver."
- To learn more refer to the Performance Efficiency Pillar whitepaper

Cost Optimization

- Cost optimization refers to the ability to avoid or eliminate unneeded expenses and resources.
- Deliver business value at the lowest price point.
- The four areas that comprise the Cost Optimization pillar include:
  - Using cost-effective resources.
  - Matching supply with demand.
  - Increasing expenditure awareness.
  - Optimizing over time

Cost Optimization Areas

- Using cost-effective resources
  - A fully cost-optimized system will use all resources to achieve the best outcome at the lowest possible price point, while still meeting your functional requirements.
  - One of the key parts to cost savings is making sure that your systems are using the appropriate services, resources and configurations.
  - As a user, you want to focus on the details such as provisioning, sizing, purchasing options, and other specifics to ensure you have the best architecture for your needs.
- Matching supply with demand
  - With AWS, you can use the elasticity of cloud architecture to meet demands as they change.
  - You can scale and be notified by other services to adjust your supply when demand changes.

Cost Optimization Areas

- Expenditure awareness
  - Being fully aware of what spending and cost drivers are happening with your business is critical.
  - You can enhance the cost optimization of your architecture in the cloud by being able to see, understand, and break down the current costs; predict future costs; and plan accordingly.
- Optimize over time
  - With all the tools and different approaches, you can measure, monitor, and improve your architecture from the data that you collect by using AWS.

Cost Optimization Design Principles

- Some design principles for cost optimization in the cloud include:
  - Adopt a consumption model.
  - Measure overall efficiency.
  - Stop spending money on undifferentiated heavy lifting.
  - Analyze and attribute expenditure.
- Adopt a consumption model
  - With the consumption model, you pay only for the computing resources you use.
  - You can then increase or decrease resources depending on business requirements.

Cost Optimization Design Principles

- Measure overall efficiency
  - It is important to measure the business output of systems and the costs that are associated with delivering them.
  - Take this measurement to understand how to make gains from increasing output and reducing costs.
- Stop spending money on undifferentiated heavy lifting
  - With AWS, you no longer need to do the heavy lifting of racking, stacking, and powering servers.
  - Instead, you can completely focus on your customers and business projects instead of the IT infrastructure.

Cost Optimization Design Principles

- Analyze and attribute expenditure
  - With the cloud, it is easier to accurately identify the usage and cost of systems.
  - Customers can measure their return on investment, which enables them to optimize resources and reduce costs.
- To learn more refer to the Cost Optimization Pillar whitepaper

Sustainability

- The sustainability pillar focuses on minimizing the environmental impacts of running cloud workloads.
- Key topics include a shared responsibility model for sustainability, understanding impact, and maximizing utilization to minimize required resources and reduce downstream impacts.
- Some design principles for sustainability in the cloud include:
  - 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

Sustainability Design Principles

- Understand your impact
  - Choose AWS Regions where you will implement workloads based on your business requirements and sustainability goals.
- Establish sustainability goals
  - Establish a set of goals for your cloud workloads.
  - These goals will help you to understand the impact of your workloads and how to minimize the impact.
- Maximize utilization
  - Maximize the utilization of your cloud workloads.
  - This will help you to reduce the cost of your workload.
  - For example, you can use dedicated hardware and software to reduce the cost of your workload.

Sustainability Design Principles

- Anticipate and adopt new, more efficient hardware and software offerings
  - With AWS, you can use the latest hardware and software offerings to reduce the cost of your workloads.
  - 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 workload.
- Use managed services
  - The cloud has many managed services to remove the operational burden of maintaining servers for tasks like sending email messages or managing databases.
- To learn more refer to the Sustainability Pillar whitepaper

Key Takeaways

- The Well-Architected Framework provides a consistent approach to evaluating cloud architectures and guidance to help implement designs.
- The Well-Architected Framework documents a set of foundational questions that enable you to understand if a specific architecture aligns well with cloud best practices.
- The Well-Architected Framework is organized into six pillars: Operational Excellence, Security, Reliability, Performance Efficiency, Cost Optimization and Sustainability.
- Each pillar includes a set of design principles and best practices.
- To learn more, refer to the AWS Well-Architected webpage.