

The Well Architected Framework

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

The Well Architected Framework helps you understand the pros and cons of decisions you make while building systems on AWS or Azure. As with most of this course we will be using AWS as the service provider.

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. We believe that having well-architected systems greatly increases the likelihood of business success.

As Solutions Architects my team and I have years of experience architecting solutions across a wide variety of business verticals and use cases. We have helped design and architect 70+ of customers' architectures on AWS, Azure and a few on Oracle. 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 section of the course is intended for those in technology roles, such as chief technology officers (CTOs), architects, developers, TPMs, SDMs and operations team members.

Every day we 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.

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

Pillar 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 avoid or eliminate unneeded cost or suboptimal resources.

When architecting solutions 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. It is preferable 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, it is encouraged to have practices that focus on enabling each team to have that capability, to access to experts who ensure that teams raise the bar on the standards they need to meet.

Second, is to put in place mechanisms that carry out automated checks to ensure standards are being met.

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.

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.

The Five Pillars of the Well Architected 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.