How to start building on AWS: Need-to-know fundamentals

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully-featured services from data centers globally. For those used to the traditional on-premises way of doing things, however, getting up to speed on the AWS platform may seem daunting at first.

With this in mind, let's take a look at some fundamentals that will help you better understand and navigate the AWS Cloud, and begin building on it. We'll start by examining the AWS Well-Architected Framework, which describes the key concepts, design principles, and best practices for designing and running workloads in the AWS Cloud. Then we'll look at the development tools available to help you begin building applications and the role AWS partners can play in supporting your journey.

Understanding the AWS Well-Architected Framework

The distillation of over a decade of experience building scalable applications on the cloud, the **AWS Well-Architected Framework** provides architectural best practices across five pillars for designing and operating reliable, secure, efficient, and cost-effective cloud systems.

Incorporating these pillars into your architecture will help yield stable and efficient systems and enable you to focus on the other aspects of design, such as functional requirements.

Let's examine the pillars one by one.

THE 5 PILLARS ARE

1. **Operational Excellence**
2. **Security**
3. **Reliability**
4. **Performance Efficiency**
5. **Cost Optimization**

Operational Excellence

This pillar encompasses the ability to support development and run workloads effectively, gain insight into their operation, and continuously improve supporting processes and procedures to deliver business value.

CONTEXT AND CONSIDERATIONS

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

Because change happens—business context, priorities, customer needs—it's important to design Operations so that it supports evolution in response to change and incorporates lessons learned through performance.

When thinking about operational excellence in the cloud, it's useful to think in terms of automation. Human error is the primary cause of defects and operational incidents. The more operations that can be automated, the less chance there is for human error. In addition to preventing error, automation helps you continuously improve your internal processes and promotes a set of repeatable best practices that can be applied across your entire organization.

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 operational failures

Security

This pillar includes the ability to protect data, systems, and assets to take advantage of cloud technologies to improve security.

CONTEXT AND CONSIDERATIONS

Before you architect any workload, you need to put in place practices that influence security. You'll want to control who can do what, be equipped to identify and respond to security incidents, protect your systems and services, and maintain the confidentiality and integrity of data through data protection.

These practices are important because they support objectives such as preventing financial loss and complying with regulatory obligations.

When thinking about cloud security, it's useful to adopt the zero trust model, where all application components and services are considered discrete and potentially malicious entities. This involves the underlying network fabric, any agents that have access to your resources, and the software that runs inside your service.

Finally, it's worth noting that within AWS, security and compliance is a shared responsibility between AWS and the customer. AWS is responsible for the security of the cloud, including the physical infrastructure, software, and networking capabilities; while the customer is responsible for security in the cloud. This includes the configuration of cloud services, the application software, and the management of sensitive data.

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

Reliability

This pillar focuses on ensuring a workload performs its intended function correctly and consistently when it's expected to. A resilient workload quickly recovers from failures to meet business and customer demand.

CONTEXT AND CONSIDERATIONS

When thinking about reliability in the cloud, it's useful to think in terms of blast radius—the maximum impact that might be sustained in the event of a system failure. To build reliable systems, you want to minimize the blast radius of any individual component.

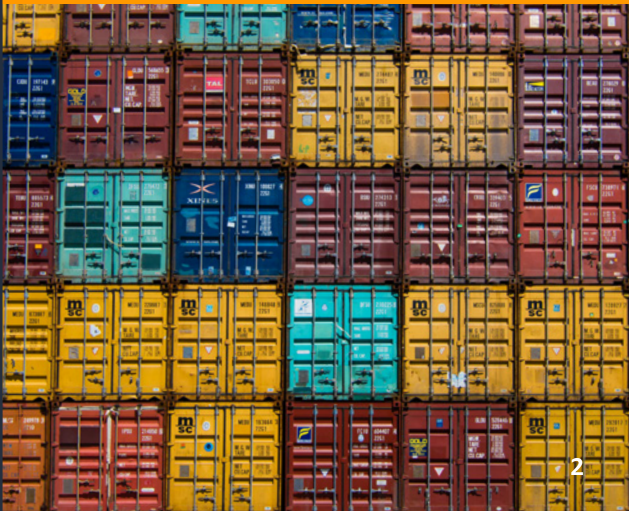
Before architecting any system, foundational requirements that influence reliability should be in place. For example, your data center must have sufficient network bandwidth. These requirements are sometimes neglected because they are beyond a single project's scope, and that can impact system reliability.

With AWS, most of these foundational requirements are already incorporated or can be addressed as required. The cloud is designed to be essentially limitless, so it's 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.

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



Performance Efficiency

This pillar focuses on using IT and computing resources efficiently.

CONTEXT AND CONSIDERATIONS

AWS recommends taking a data-driven approach to building a high-performance architecture. This means gathering 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, meanwhile, ensures you are taking advantage of the 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. Well-Architected workloads use multiple solutions and enable different features to improve performance.

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

Cost Optimization

This pillar helps you achieve your desired business outcomes while minimizing costs.

CONTEXT AND CONSIDERATIONS

When thinking about cost optimization in the cloud, it’s useful to think of cloud-related spend in terms of OpEx instead of CapEx. OpEx is an ongoing pay-as-you-go model whereas CapEx is a one-time purchase model.

Traditional IT costs on on-premises data centers are mostly CapEx. You pay for all your capacity upfront regardless of whether or not you use it. Purchasing new servers was a lengthy process that involved getting sign-off from multiple parties. That’s because CapEx costs were often significant and mistakes costly. Moreover, once you purchased servers, they could take weeks to arrive.

In AWS, your costs are OpEx. You pay on an ongoing basis for the capacity that you use. Provisioning new servers can be done in real-time by engineering, and without the need for a lengthy approval process. This is because OpEx costs are much smaller and can be reversed if requirements change. Because you only pay for what you use, any excess capacity can simply be stopped and terminated. When you do decide to use a service, provisioning can be done in the order of seconds and minutes.

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

Tools to facilitate building on AWS

Once you're ready to build based on AWS Well-Architected Framework principles, it's easy to develop applications on AWS. That's because you can leverage existing skillsets by using preferred programming languages and tools.

AWS offers five types of developer tools. These are:

Infrastructure as Code (IaC)

Three different tools are available for IaC: AWS Cloud Formation lets you describe your resources in YAML or Json, and then provisions them for you; AWS Serverless Application Model is an open source framework for building serverless applications; and AWS Cloud Development Kit is an open source software development framework that makes it easy to define cloud applications in familiar programming languages.

Integrated Development Environment (IDE) and IDE toolkits

AWS Cloud9 is a cloud-based IDE that allows you to run, write, and debug code in your AWS environment. You can share your development environment with your team to compare programs and track each other's input in real time. You can also use AWS toolkits for familiar desktop IDEs to provide an integrated experience for developing serverless applications. AWS toolkits are open source plugins, making it easier to create, debug, and deploy on AWS.

Software Development Kit (SDK) and Command Line Interface (CLI)

AWS SDKs are offered in many popular programming languages. They take the complexity out of coding by providing abstract APIs and giving you easy access to AWS services and resources. AWS CLIs, meanwhile, enable you to control multiple AWS services from from the command line and automate them through scripts. Finally, AWS Cloud Show gives you a browser-based shell containing an AWS CLI and other popular tools that are ready to use.

CI/CD

AWS development tools feature integrated CI/CD to enable teams to deliver code changes more frequently and reliably. You can also integrate popular AWS partner and open source solutions with AWS developer tools so you don't have to migrate to a new tool.

Next generation DevOps services

AWS services such as Amazon CodeGuru offer intelligent recommendations for improving code quality and identifying an application's most expensive lines of code.

Enlisting the support of AWS Well-Architected Partners

When implementing AWS Well-Architected Framework principles, you can also choose to enlist the support of AWS Well-Architected Partner Program members. These partners have in-depth training on the AWS Well-Architected Framework and can help you implement AWS best practices, measure your workloads, and make improvements where required.

Customers who have engaged with AWS partners have realized significant cost savings, improved the performance of their applications, and eliminated security risks and other high risk issues.

Conclusion

Building on AWS is easy and efficient once you get familiar with fundamentals such as the AWS Well-Architected Framework and familiarize yourself with the tools available. And for those who don't want to go it alone, there are a wide range of AWS partners available to support your cloud journey.

To learn more about how to get started with the AWS Cloud, please see the resources below:

https://aws.amazon.com/getting-started/fundamentals-core-concepts
https://aws.amazon.com/getting-started
https://docs.aws.amazon.com/general/latest/gr/glos-chap.html