**5 Pillars of the AWS Well-Architected Framework**

## 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**. You can find prescriptive guidance on implementation in the [Operational Excellence Pillar whitepaper](https://d1.awsstatic.com/whitepapers/architecture/AWS-Operational-Excellence-Pillar.pdf).

### Design Principles

There are **six design** principles for operational excellence in the cloud:

* Perform **operations as code**
* **Annotate** documentation
* 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, customer needs, etc. It’s important to design operations to support evolution over time in response to change and to incorporate lessons learned through their performance.

## **Security**

The security pillar includes the ability to **protect information**, **systems**, and **assets** while **delivering business value** through **risk assessments** and **mitigation strategies**. You can find prescriptive guidance on implementation in the [Security Pillar whitepaper](https://d1.awsstatic.com/whitepapers/architecture/AWS-Security-Pillar.pdf).

### Design Principles

There are six 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**
* Prepare for **security events**

### Best Practices

Before you architect any system, you need to put in place practices that influence security. You will 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](https://aws.amazon.com/compliance/shared-responsibility-model/) enables organizations to achieve **security and compliance** goals.

Because AWS physically secures the infrastructure that supports our cloud services, you can focus on using services to accomplish your goals.

## **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. You can find prescriptive guidance on implementation in the [Reliability Pillar whitepaper](https://d1.awsstatic.com/whitepapers/architecture/AWS-Reliability-Pillar.pdf).

### Design Principles

There are five design principles for reliability in the cloud:

* **Test recovery** procedures
* Automatically **recover** from failure
* Scale **horizontally** to increase aggregate system availability
* **Stop guessing** capacity
* **Manage change** in **automation**

### Best Practices

To achieve **reliability**, a system must have a well-planned foundation and monitoring in place, with mechanisms for handling changes in demand or requirements. The system should be designed to detect **failure** and **automatically** heal itself.

## **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](https://d1.awsstatic.com/whitepapers/architecture/AWS-Performance-Efficiency-Pillar.pdf).

### 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
* Mechanical sympathy

### Best Practices

Take a data-driven approach to selecting 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.

By reviewing your choices on a cyclical basis, you will ensure you are taking advantage of the continually evolving AWS cloud. Monitoring will ensure you are aware of any deviance from expected performance and can take action on it.

## **5. Cost Optimization**

The cost optimization pillar includes the ability to avoid or eliminate unneeded cost or suboptimal resources. You can find prescriptive guidance on implementation in the [Cost Optimization Pillar whitepaper](https://d1.awsstatic.com/whitepapers/architecture/AWS-Cost-Optimization-Pillar.pdf).

### Design Principles

There are five design principles for cost optimization in the cloud:

* Adopt a **consumption(વપરાસ)** model
* Measure overall **efficiency**
* **Stop spending money** on data center operations
* Analyze and attribute expenditure
* Use managed services to reduce cost of ownership

### Best Practices

As with the other pillars, there are tradeoffs 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 upfront cost optimization.

Design decisions are sometimes guided by haste as opposed to empirical data, as the temptation always exists to overcompensate “just in case” rather than spend time benchmarking for the most cost-optimal deployment. This often leads to drastically over-provisioned and under-optimized deployments.

Using the appropriate instances and resources for your system is key to cost savings. For example, a reporting process might take five hours to run on a smaller server but one hour to run on a larger server that is twice as expensive. Both servers give you the same outcome, but the smaller one will incur more cost over time. A well-architected system will use the most cost-effective resources, which can have a significant and positive economic impact.