

## Five Pillars of AWS Well-Architected Framework

AWS Well-Architected Framework is basically a body of knowledge that describes the various design principles, key concepts, design and architectural best practices that can help companies design and run highly efficient workloads in the AWS platform.

This framework ensures that the company's cloud architecture is in accordance with the AWS best practices. It also comes with related AWS features, services and tools that you can utilize to measure the overall efficiency of your design.

The AWS Well-Architected Framework will empower you to improve your existing IT infrastructure in terms of your overall operations, security, reliability, efficiency, cost optimization, and sustainability.

5 Pillars of AWS Well-Architected Framework are as follows:

- **Operational Excellence**
- **Security**
- **Reliability**
- **Performance Efficiency**
- **Cost Optimization**
- **Sustainability**

### Operational Excellence

Operational excellence brings business value to the process by supporting the development and efficient execution of workloads.

### Design Principles:

- **Operate as codes:** Define and update the entire workload as code. Execute the operation processes as code and automate the execution of processes by activating them. The benefit of operating procedures as code is that it reduces human errors and provides consistent responses to the events. It could be reused again for creating similar architectures.

- **Build reversible changes frequently:** Keep the components updated by designing workloads appropriately. Build reversible changes gradually so that they can be altered later if they fail.
- **Frequent revision of operational procedures:** Improvise the operations frequently. Update the processes along with the workload. Carry out periodic reviewing and validation of the ongoing processes and keep the teams updated.
- **Predict the possibilities of failure:** Develop strategies to anticipate the possible breakdowns and sort them accordingly. Evaluate the procedures repeatedly so that you get a proper understanding of their impact to prevent future losses.
- **Improvise the operations in case of failure:** Try to recover and improve the existing procedures, if they failed.

## Security

**Security helps you to secure your data and assets by using cloud technologies for protection.**

### Design Principles:

- **Strengthen the foundation of identity:** Use the concept of least privilege while communicating with your AWS resources, and implement and authorize separation of tasks. The goal is to remove the use of long-term static credentials through identity management.
- **Trace the procedures:** Real-time monitoring, alerting, and auditing of activities and changes in the environment. Evaluate logs and automate metric systems.
- **Employ protection throughout the board:** Use numerous security measures to implement a defense-in-depth strategy. Implement this approach on all layers such as AWS VPC, edge of the network, load balancing, operating system, and computer service.

- **Automate the security:** Software-based security methods that are programmed increase the capacity to expand safely and efficiently. Produce secure architectures that include features that are specified and managed as code in version-controlled templates.
- **Secure data during transit and inactivity:** Create sensitivity levels for your data and use techniques such as encryption, tokenization, and access control, as and when needed.
- **Protect data from people:** Restrict access to direct or manual data processing using procedures and tools. When managing sensitive data, the danger of misuse, alteration, or manual errors is reduced.
- **Prepare for potential security threats:** Be prepared for upcoming threats by establishing incident management and investigative policies and procedures in line with your company's needs. To improve your detection, investigation, and recovery times, do incident response simulations and employ automated technology.

## Reliability

Reliability refers to workload's capacity to fulfill its intended purpose accurately and consistently when required. This includes the ability to run the workload and test it during its entire life span. This blog provides in-depth, best-practice recommendations for deploying dependable workloads on AWS.

## Design Principles:

- **Recovery from failure:** Activate the automation of workload once the key performance indicator (KPI) reaches the threshold.
  - KPIs must be on commercial value rather than technical elements of the performance of service. It enables automatic failure notice, tracking, and automated recovery methods to work around or correct the problem.
  - It is feasible to predict and correct errors before they happen with more advanced automation.

- **Procedures for test recovery:** In an on-premises system, testing is to be done to ensure that the workload works in a specific situation. Identify the reason for the failure of workload and evaluate your recovery processes in the cloud.
  - You can use automation to mimic various failures or reproduce circumstances that have previously resulted in a loss. This method reveals failure paths that may be tested and fixed before an actual failure occurs, lowering risk.
- **Scaling of workload availability:** To decrease the impact of a single failure on the total workload, replace one resource with several tiny ones. Distribute requests among several smaller resources to avoid having a single point of failure.
- **Stop speculating on capacity:** When the demands placed on a workload exceed the capacity, it is called resource saturation; it is a common cause of failure in on-premises workloads, which needs to be avoided.
  - You can monitor requirement and workload utilization in the cloud and automate the addition and removal of resources to keep demand at an appropriate level without over- or under-provisioning.
- **Implement changes using automation:** Make automated changes to your infrastructure. Handle automation changes and these changes may then be monitored and assessed.

## **Performance Efficiency**

Performance efficiency involves the capacity to employ computer resources efficiently to fulfill system needs and to maintain the efficiency as demand changes and technology advances.

### **Design Principles:**

- **Employ latest technologies:** Delegate complicated duties to your cloud vendor to make technically advanced deployment easier for your team. Recognize technology as a service rather than just requiring your IT personnel to learn about hosting and administering a new technology.

- NoSQL, database, media transcoding, and machine learning are services that your team may use in the cloud, allowing them to concentrate on product creation rather than resource allocation and administration.
- **Take it to the international level:** By employing your workload across various AWS regions across the world, you may offer your clients lower latency and a better experience at a cheaper cost.
- **Use serverless architectures:** For typical computational operations, serverless architectures eliminate the need to run and manage physical servers.
  - Serverless storage applications, for instance, can serve as static web pages without the need for web servers, while event services can host code.
- Perhaps because managed services run at a cloud-scale, this alleviates the operational strain of managing physical servers while also potentially lowering transaction costs.
- **Prototype frequently:** With virtualized and automatable assets, you can easily compare alternative types of instances, storage, or settings.

The best practices of performance efficiency are selection, review, monitoring, and trade-offs.

## Cost Optimization

The capacity to manage systems that offer business value at the lowest cost is part of cost optimization.

## Design Principles:

- **Deploy Cloud Financial Management:** Investing in Cloud Financial Management or cost optimization will help you to achieve financial success and increase the company value in the cloud.
  - Your company must devote time and money to developing capabilities in this new technology and use the management sector.

- To become a cost-effective company, you must create opportunities through knowledge building, programmes, resources, and workflows, similar to your security or operational excellence capabilities.
- **Employ a consumption model:** Spend just for the software applications that you need and adjust use based on business needs rather than relying on extensive forecasts.
  - During the workweek development, generally, environments are used for eight hours a day only. You may save up to 75 percent on these resources by turning them off when not in use.
- **Measure total efficiency:** Calculate the workload business output and the expenses involved with completing it. Use this metric to see how much money you save by boosting results and cutting costs.
- **Verify expenses:** The cloud makes it simpler to precisely identify system use and expenditures, allowing for clear IT cost attribution to individual task owners. This enables the professionals to track their return on investment (ROI) and manage their resources while lowering expenses.

#### Others:

#### Operational Excellence:

**Organization** – AWS Cloud Compliance, AWS Trusted Advisor, AWS Organizations

- **Prepare** – AWS Config
- **Operate** – Amazon CloudWatch
- **Evolve** – Amazon Elasticsearch Service
- **Cloud Formation**

#### Security:

Security – AWS Shared Responsibility Model, AWS Config, AWS Trusted Advisor

Identity and Access Management – IAM, Multi-Factor Authentication, AWS Organizations

Detective Controls – AWS CloudTrail, AWS Config, Amazon GuardDuty

Infrastructure Protection – Amazon VPC, Amazon CloudFront with AWS Shield, AWS WAF

Data Protection – ELB, Amazon Elastic Block Store (Amazon EBS), Amazon S3, and Amazon Relational Database Service (Amazon RDS) encryption, Amazon Macie, AWS Key Management Service (AWS KMS)

Incident Response – IAM, Amazon CloudWatch Events

### **Reliability:**

Foundations – IAM, Amazon VPC, AWS Trusted Advisor, AWS Shield

Change Management – AWS CloudTrail, AWS Config, Auto Scaling, Amazon CloudWatch

Failure Management – AWS CloudFormation, Amazon S3, AWS KMS, Amazon Glacier

Workload Architecture – AWS SDK, AWS Lambda

Performance Efficiency:

Selection – Auto Scaling for Compute, Amazon EBS and S3 for Storage, Amazon RDS and DynamoDB for Database, Route53, VPC, and AWS Direct Connect for Network

Review – AWS Blog and What's New section of the website

Monitoring – Amazon CloudWatch

Tradeoffs – Amazon ElastiCache, Amazon CloudFront, AWS Snowball, Amazon RDS read replicas.

### **Cost Optimization:**

Cloud Financial Management – Amazon QuickSight, AWS Cost and Usage Report (CUR)

Cost-Effective Resources – Cost Explorer, Amazon CloudWatch and Trusted Advisor, Amazon Aurora for RDS, AWS Direct Connect with Amazon CloudFront

Matching supply and demand – Auto Scaling

Expenditure Awareness – AWS Cost Explorer, AWS Budgets

Optimizing Over Time – AWS News Blog and the What's New section on the AWS website, AWS Trusted Advisor

**Sustainability:**

Region Selection – AWS Global Infrastructure

User Behavior Patterns – Auto Scaling, Elastic Load Balancing

Software and Architecture Patterns – AWS Design Principles

Data Patterns – Amazon EBS, Amazon EFS, Amazon FSx, Amazon S3

Hardware Patterns – Amazon EC2, AWS Elastic Beanstalk

Development and Deployment Process – AWS CloudFormation