**AWS Well Architected**

**General Design Principles**

**1.Stop guessing your capacity needs**

**2.Test systems at production scale**

**3.Automate to make architectural experimentation easier**

**4.Allow for evolutionary architectures**

**5.Drive architectures using data**

**6.Improve through game days**

The Five Pillars of the Framework

1.Operational Excellence

2. Security

3. Reliability

4. Performance Efficiency

5. Cost Optimization

Operational Excellence

Design Principles

•**Perform operations as code**:

•**Annotate documentation:**

**•Make frequent, small, reversible changes:**

•**Refine operations procedures frequently:**

•**Anticipate failure**:

•**Learn from all operational failures**:

**Operational Excellence: Definition**

Three best practice areas for operational excellence in the cloud:

•**Prepare**

**OPS 1: How do you determine what your priorities are?**

**OPS 2: How do you design your workload so that you can understand its state?**

**OPS 3: How do you reduce defects, ease remediation, and improve flow into production?**

**OPS 4: How do you mitigate deployment risks?**

**OPS 5: How do you know that you are ready to support a workload?**

•**Operate**

**OPS 6: How do you understand the health of your workload?**Define, capture, and analyze workload

**OPS 7: How do you understand the health of your operations?**

**OPS 8: How do you manage workload and operations events?**

**•Evolve**

**OPS 9: How do you evolve operations?**

Key AWS Services

**The AWS service that is essential to Operational Excellence is AWS *CloudFormation***,**which you can use to create templates based on best practices.** This enables you to provision resources in an orderly and consistent fashion from your development through production environments. The following services and features support the three areas in operational excellence:

•**Prepare**: **AWS Config** and AWS Config rules can be used to create standards for workloads and to determine if environments are compliant with those standardsbefore being put into production.

•**Operate**: **Amazon CloudWatch** allows you to monitor the operational health of a workload.

• **Evolve**: **Amazon Elasticsearch Service (Amazon ES)** allows you to analyze your log data to gain actionable insights quickly and securely.

Security

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:**

Security Definition

There are five best practice areas for security in the cloud:

**• Identity and Access Management**

**SEC 1: How do you manage credentials and authentication? SEC 2: How do you control human access?**

**SEC 3: How do you control programmatic access?**

**• Detective Controls**

**SEC 4: How do you detect and investigate security events?**

**SEC 5: How do you defend against emerging security threats?**

**• Infrastructure Protection**

**SEC 4: How do you detect and investigate security events?**

**SEC 5: How do you defend against emerging security threats?**

**• Data Protection**

**SEC 6: How do you protect your networks?**

**SEC 7: How do you protect your compute resources?**

**• Incident Response**

**SEC 8: How do you classify your data?**

**SEC 9: How do you protect your data at rest?**

**SEC 10: How do you protect your data in transit?**

**SEC 11: How do you respond to an incident?**

Key AWS Services – Security

**•Identity and Access Management – IAM**

•**Detective Controls**: **AWS Config** provides **CloudTrail** a detailed inventory of your AWS resources and configuration.

Amazon **GuardDuty** is a managed threat detection service monitors for malicious or unauthorized behavior.

Amazon **CloudWatch** is a monitoring service for AWS resources which can trigger CloudWatch Events to automate security responses.

• **Infrastructure Protection**: **AWS Shield** for DDoS mitigation. **AWS WAF** is a web application firewall that is deployed on either Amazon CloudFront or Application Load Balancer to help protect your webapplications from common web exploits.

• **Data Protection**: Services such as ELB, Amazon Elastic Block Store (Amazon EBS),Amazon S3, and Amazon Relational Database Service (Amazon RDS) include **encryption** capabilities to protect your data in transit and at rest. **Amazon Macie** automatically discovers, classifies and protects sensitive data, while **AWS Key Management Service (AWS KMS)** makes it easy for you to create and control keysused for encryption.18

• **Incident Response**:. AWS CloudFormation can be used to create a trusted environment or clean room for conducting investigations.

Resources – Security

Documentation

• AWS Cloud Security

• AWS Compliance

• AWS Security Blog

Whitepaper

• Security Pillar

• AWS Security Overview

• AWS Security

Best Practices

• AWS Risk and Compliance

Video

• AWS Security State of the Union

• Shared Responsibility Overview

**Reliability**

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:**

**Definition**

There are three best practice areas for reliability in the cloud:

• **Foundations**

**REL 1: How do you manage service limits?**

**REL 2: How do you manage your network topology?**

**• Change Management**

**REL 3: How does your system adapt to changes in demand?**

**REL 4: How do you monitor your resources?**

**REL 5: How do you implement change?**

**• Failure Management**

**REL 6: How do you back up data?**

**REL 7: How does your system withstand component failures?**

**REL 9: How do you plan for disaster recovery?**

Key AWS Services

**Foundations**: AWS IAM, AWS Shield for DDOS

**Change Management**: AWS Config and AWS Cloudtrail, Amazon Auto Scaling

**Failure Management**: AWS CloudFormation

Resources

Refer to the following resources to learn more about our best practices for Reliability.

Documentation

• Service Limits

• Service Limits Reports Blog

• Amazon Virtual Private Cloud

• AWS Shield

• Amazon CloudWatch

• Amazon S3

• AWS KMS Whitepaper

• Reliability Pillar

• Backup Archive and Restore Approach Using AWS

• Managing your AWS Infrastructure at Scale

• AWS Disaster Recovery

• AWS Amazon VPC Connectivity Options

Video

• How do I manage my AWS service limits?

• Embracing Failure: Fault-Injection and Service ReliabilityProduct

• AWS Premium Support

• Trusted Advisor

**Performance Efficiency**

**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:**

**Definition**

In AWS, compute is available in three forms: instances, containers, and functions

There are four best practice areas for performance efficiency in the cloud:

• **Selection**

**PERF 1: How do you select the best performing architecture?**

**COMPUTE-** available in three forms:

1.**Instances -** are virtualized servers

2.**Containers** - method of operating system

3.**Functions –** abstract execution

environment from code (Lambda)

**PERF 2: How do you select your compute solution?**

**STORAGE** - lock, file, or object

**PERF 3: How do you select your storage solution?**

**DATABASE –** availability, consistency, partition tolerance, latency, durability, scalability,and query capability

**PERF 4: How do you select your database solution?**

**NETWORK –**

**PERF 5: How do you configure your networking solution?**

• **Review**

**PERF 6: How do you evolve your workload to take advantage of new releases?**

• **Monitoring**

**PERF 7: How do you monitor your resources to ensure they are performing as expected?**

• **Tradeoffs**

**PERF 8: How do you use tradeoffs to improve performance?**

Key AWS Services

AWS service that is essential to Performance Efficiency is Amazon CloudWatch

The following services and features support the four areas in performance efficiency:

• **Selection**

•**Compute:** Auto Scaling

•**Storage**: Amazon EBS

S3 for serverless content delivery, and S3 transfer acceleration

•**Database**: Amazon RDS

AmazonDynamoDB

• **Network**: Amazon Route 53

Amazon VPC endpoints

and AWS Direct Connect

• **Review**: **The AWS Blog and the What's New section**

• **Monitoring**: Amazon CloudWatch

• **Tradeoffs**: **Amazon ElastiCache, Amazon CloudFront, and AWS Snowball** to improve performance.

Resources

Refer to the following resources to learn more about our best practices for

Performance Efficiency.

Documentation

• Amazon S3 Performance Optimization

• Amazon EBS Volume

Performance

Whitepaper

• Performance Efficiency Pillar

Video

• AWS re:Invent 2016: Scaling Up to Your First 10 Million Users (ARC201)

• AWS re:Invent 2017: Deep Dive on Amazon EC2 Instances

**Cost Optimization**

**Design Principles**

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

**• Adopt a consumption model:** Pay for what you need

**• Measure overall efficiency:**

**• Stop spending money on data center operations:**

**• Analyze and attribute expenditure:** (ROI)

**• Use managed and application level services to reduce cost of ownership:**

Definition

There are four best practice areas for cost optimization in the cloud:

**• Expenditure Awareness**

**COST 1: How do you govern usage?**

**COST 2: How do you monitor usage and cost?**

**COST 3: How do you decommission resources?**

**• Cost-Effective Resources**

**COST 4: How do you evaluate cost when you select services?**

**COST 5: How do you meet cost targets when you select resource type and size?**

**COST 6: How do you use pricing models to reduce cost? COST 7: How do you plan for data transfer charges?**

**• Matching supply and demand**

**COST 8: How do you match supply of resources with demand?**

**• Optimizing Over Time**

**COST 9: How do you evaluate new services?**

**Key AWS Services**

**• Expenditure Awareness:** **AWS Cost Explorer** allows you to view and track your usage in detail.

**AWS Budgets notify you if your usage or spend exceeds** actual orforecast budgeted amounts.

**• Cost-Effective Resources:** You can use **Cost Explorer for Reserved Instance recommendations**, and see patterns in how much you spend on AWS resources over time. Use **Amazon CloudWatch** and **Trusted Advisor** to help right size your resources. You can use **Amazon Aurora on RDS to remove database licensing costs.**

**AWS Direct Connect and Amazon CloudFront can be used to optimize data transfer.**

**•Matching supply and demand:** **Auto Scaling**

**•Optimizing Over Time:** **The AWS News Blog and the What's New section on the AWS website** are resources for learning about newly launched features and services.**AWS Trusted Advisor** inspects your AWS environment and finds opportunities to save you money by eliminating unused or idle resources or committing to Reserved Instance capacity.

**Resources**

Refer to the following resources to learn more about our best practices for Cost Optimization.

Documentation

• Analyzing Your Costs with Cost Explorer

• AWS Cloud Economics Center

• AWS Detailed Billing Reports

Whitepaper

• Cost Optimization Pillar

Video

• Cost Optimization on AWSTool

• AWS Total Cost of Ownership (TCO) Calculators

• AWS Simple Monthly Calculator

**1. 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.

**2. 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.

Before architecting any system, foundational requirements that influence reliability should be in place. For example, you must have sufficient network bandwidth to your data center. These requirements are sometimes neglected (because they are beyond a single project’s scope). This neglect can have a significant impact on the ability to deliver a reliable system. In an on-premises environment, these requirements can cause long lead times due to dependencies and therefore must be incorporated during initial planning.

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

**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.

Finally, your architecture can make tradeoffs to improve performance, such as using compression or caching, or relaxing consistency requirements.  
The optimal solution for a particular system will vary based on the kind of workload you have, often with multiple approaches combined. Well-architected systems use multiple solutions and enable different features to improve performance.

**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 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.