## Solution Fundamentals

## Importance of Architecture in design

## Key elements of cloud

## network

The network **serves as the linkage between the end users consuming cloud services and the provider's data centers providing the cloud services**. In addition, in large-scale cloud data centers, tens of thousands of compute and storage nodes are connected by a data center network to deliver a single-purpose cloud service

## Compute instances

Generally 1 instance == 1 machine/server (often a virtual machine).

See e.g. <http://aws.amazon.com/ec2/instance-types/> and <https://developers.google.com/appengine/docs/adminconsole/instances>

3

The hierarchy is in this way, cloud->data centers->host computers->instances(Virtual Machines).

Consider an example to understand each term. Consider a public cloud like AWS or Google App Engine, each public cloud will have many data centers at different geographical locations where there would be many servers, computers, disks for storage of data etc. and other hardware components which are required to provide the cloud services.

In each data centers there would be a group(cluster) of dedicated hardware which provides specialized services or processes and these are known as hosts.

The instance type determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities and are grouped in instance families based on these capabilities. Instance are a kind of virtual environment which are used for running the users process or application.

Whenever a user wants to avail a particular service or wants to deploy a certain kind of app on the cloud, then the user needs to create an instance of that particular type.

For further information refer to the following links For aws: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html>

For Google Cloud Platform: <https://cloud.google.com/appengine/docs/java/how-instances-are-managed>

## What is auto scaling

AWS Auto Scaling **monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost**. Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes.

# [AWS Scale out , Scale Up](https://stackoverflow.com/questions/42034688/aws-scale-out-scale-up)

Q.

In AWS, we come across scaling up (Adding more storage i.e from t1.small to t2.medium or t2.large) and scaling out is adding up of instances (adding EC2 instances or other). How are these related to Horizontal scaling and vertical scaling. Also, what is preferred to be used more in Recovery and Backups, Volume management more often while the condition is to minimize the cost of the infrastructure maintenance.

Ans:

Scaling up is when you change the instance types within your Auto Scaling Group to a higher type (for example: changing an instance from a m4.large to a m4.xlarge), scaling down is to do the reverse.

Scaling out is when you add more instances to your Auto Scaling Group and scaling in is when you reduce the number of instances in your Auto Scaling Group.

When you scale out, you distribute your load and risk which in turn provides a more resilient solution, here is an example:

Let's say you have an ASG with 4x m4.xlarge instances. If one fails that means you lost 25% of your processing capability, it doesn't matter that these are sizeable instances with a good amount of CPU and Ram, the fact is by having bigger instance types but less of them you increase the impact of a failure.

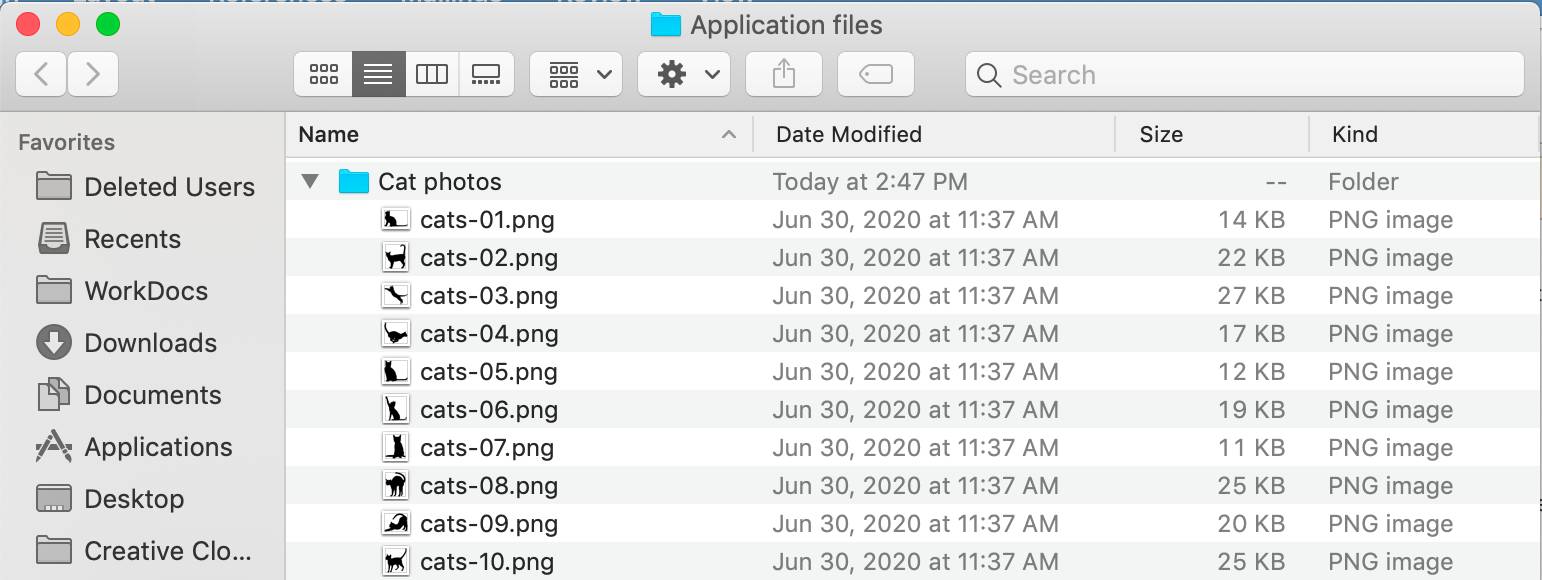
However if you had say 8x m4.large instead, your total compute is the same as 4x m4.xlarge however if 1 instance dies then you only lose 12.5% of your resources.

Typically its better to use more smaller instances than less larger ones, so you will see that its more common to "scale-out" to meet demand than it is to "scale-up".

One last consideration is, in order to scale-up/scale-down you have to restart the instance, so there is a service impact when you scale-up/scale-down. There is no such impact when you scale-in/scale-out howeve

## Storage Types

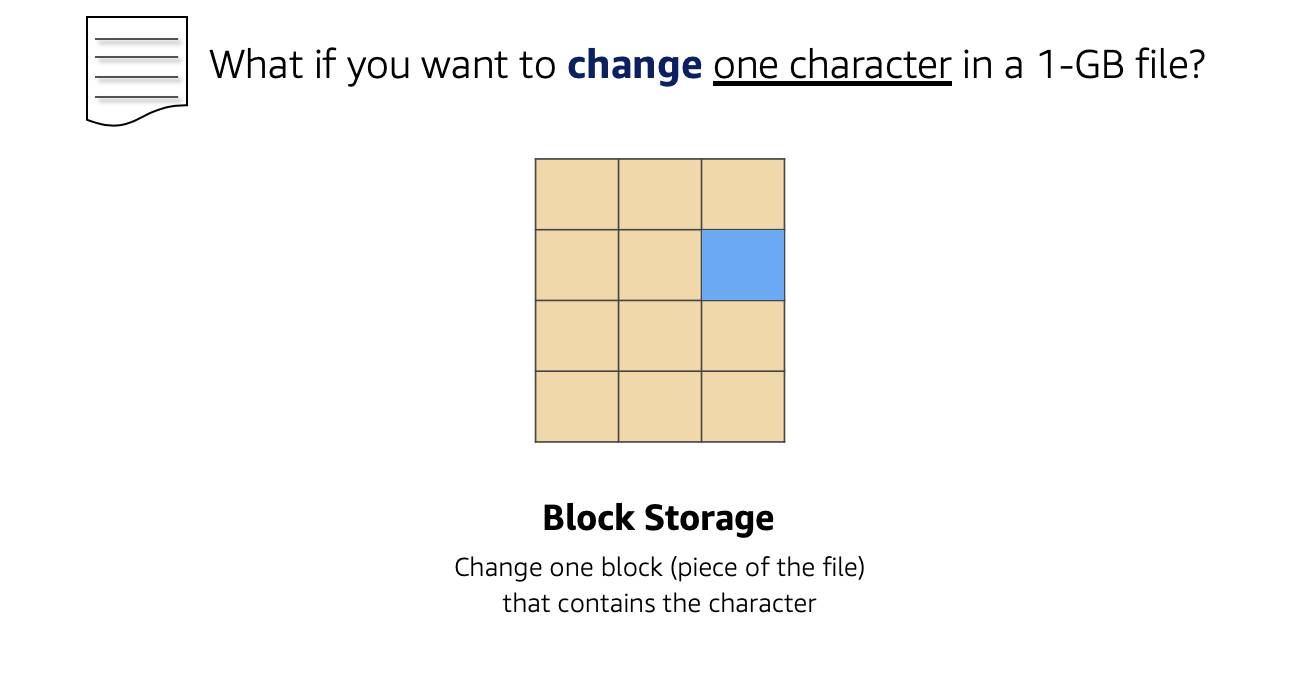
AWS storage services are grouped into three different categories**: block storage, file storage, and object storage**. **File Storage** You may be familiar with file storage if you’ve interacted with file storage systems like Windows File Explorer or Finder on MacOS. You place your files in a tree-like hierarchy that consists of folders and subfolders. For example, if you have hundreds of cat photos on your laptop, you may want to create a folder called Cat photos, and place those images inside that folder to organize them. Since you know these images will be used in an application, you may want to place the cat photos folder inside another folder called Application files.



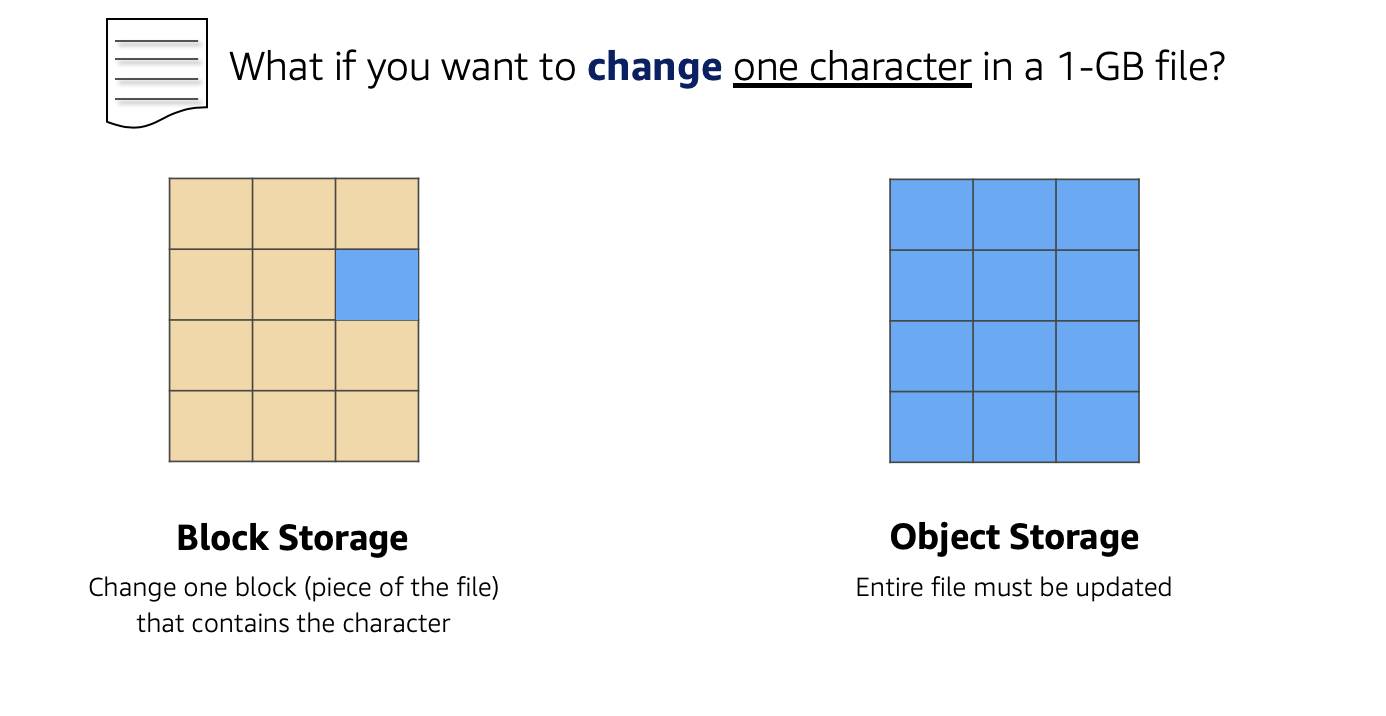
Each file has metadata such as file name, file size, and the date the file was created. The file also has a path, for example, computer/Application\_files/Cat\_photos/cats-03.png. When you need to retrieve a file, your system can use the path to find it in the file hierarchy. File storage is ideal when you require centralized access to files that need to be easily shared and managed by multiple host computers. Typically, this storage is mounted onto multiple hosts and requires file locking and integration with existing file system communication protocols. Common use cases for file storage include:

* Large content repositories
* Development environments
* User home directories

**Block Storage** While file storage treats files as a singular unit, **block storage splits files into fixed-size chunks of data called** **blocks** that have their own addresses. Since each block is addressable, blocks can be retrieved efficiently. When data is requested, these addresses are used by the storage system to organize the blocks in the correct order to form a complete file to present back to the requestor. Outside of the address, there is no additional metadata associated with each block. So, when you want to change a character in a file, you just change the block, or the piece of the file, that contains the character. This ease of access is why block storage solutions are fast and use less bandwidth.



Since block storage is optimized for low-latency operations, it is a typical storage choice for high-performance enterprise workloads, such as databases or enterprise resource planning (ERP) systems, that require low-latency storage. **Object Storage** Objects, much like files, are also treated as a single unit of data when stored. However, unlike file storage, **these objects are stored in a flat structure instead of a hierarchy**. Each object is a file with a unique identifier. This identifier, along with any additional metadata, is bundled with the data and stored. **Changing just one character in an object is more difficult than with block storage.** When you want to change one character in a file, the entire file must be updated.



With object storage, you can store almost any type of data, and there is no limit to the number of objects stored, making it easy to scale. Object storage is generally useful when storing large data sets, unstructured files like media assets, and static assets, such as photos. **Relate Back to Traditional Storage Systems** If you’ve worked with storage on-premises, you may already be familiar with block, file, and object storage. Consider the following technologies and how they relate to systems you may have seen before.

* Block storage in the cloud is analogous to **direct-attached storage (DAS) or a storage area network (SAN)**.
* File storage systems are often supported with a **network attached storage (NAS)** server.

Adding more storage in a traditional data center environment is a more rigid process, as you need to purchase, install, and configure these storage solutions. With cloud computing, the process is more flexible. You can create, delete, and modify storage solutions all within a matter of minutes.

**Resources**

* [*External Site:* AWS: What Is Cloud Storage](https://aws.amazon.com/what-is-cloud-storage/)
* [*External Site:* AWS: Types of Cloud Storage](https://aws.amazon.com/what-is-cloud-object-storage/#types)

## Database services

**Database as a Service (DBaaS) :**  
Like[SaaS, PaaS and IaaS](https://www.geeksforgeeks.org/cloud-based-services/)of cloud computing we can consider DBaaS (also known as Managed Database Service) as a cloud computing service. It allows users associated with database activities to access and use a cloud database system without purchasing it.

DBaaS and cloud database comes under Software as a Service (SaaS) whose demand is growing so fast  
In simple we can say Database as a Service (DBaaS) is self service/ on demand database consumption coupled with automation of operations. As we know cloud computing services are like pay per use so DBaaS also based on same payment structure like how much you will use just pay for your usage. This DBaaS provides same function as like standard traditional and relational database models. So using DBaaS, organizations can avoid data base configuration, management, upgradation and security.

DBaaS consists of an info manager element, that controls all underlying info instances via API. This API is accessible to the user through a management console, typically an online application, that the user might use to manage and assemble the info and even provision or deprovision info instances.

**Key Characteristics of DBaaS :**

* A fully managed info service helps to line up, manage, and administer your info within the cloud and conjointly offer services for hardware provisioning and Backup.
* DBaaS permits the availability of info’s effortlessly to Database shoppers from numerous backgrounds and IT expertise.
* Provides on demand services.
* Supported the resources offered, it delivers a versatile info platform that tailors itself to the environment’s current desires.
* A team of consultants at your disposal, endlessly watching the Databases.
* Automates info administration and watching.
* Leverages existing servers and storage.

**How does DBaaS work ?**  
It is a service that is added to our databases which make our daily tasks easier. It eliminates tedious and time-consuming administration tasks and makes our tasks simpler and more flexible. Now most of the organizations are going for DBaaS as it helps organizations to accelerate their business performance by starting their working with database more easily and running the workloads without delay.

Once we move our database to the cloud, we have the option to add software deployment as a service. Doing so simplifies the processes required to make information available through Internet-based communications. Storage consolidation can also be useful for moving company databases to the cloud.

**Advantages of DBaaS :**

1. DBaaS is responsible of the info supplier to manage and maintain info hardware and code.
2. The hefty power bills for ventilation and cooling bills to stay the servers running area unit eliminated.
3. An organization that subscribes to DBaaS is free from hiring info developers or constructing a info system in-house.
4. Make use of the most recent automation, straightforward outs of clouds area unit possible at low price and fewer time.
5. Human resources needed to manage the upkeep of the system is eliminated.
6. Since DBaaS is hosted off-site, the organization is free from the hassles of power or network failure.
7. Explore the portfolio of Oracle info as a service.

**Disadvantages of DBaaS :**

1. Traditional enterprises may have objections to cloud-based services generally.
2. In case of significant failure of the DBaaS server or network, the organization might lose its knowledge.
3. Companies already equipped with resources and IT-related human resources might not realize DBaaS solutions economically viable.
4. Intrinsic network connected problems with cloud can impact the performance of a DBaaS.
5. Features offered within the typical RDBMS might not perpetually be offered during a DBaaS system.
6. The use of DBaaS may result in revenue loss in alternative areas of code updates and hardware management.

## Security

## DDoS attack meaning

In a distributed denial-of-service (DDoS) attack, an attacker overwhelms its target with unwanted internet traffic so that normal traffic can’t reach its intended destination.

From a high level, a DDoS or DoS attack is like an unexpected traffic jam caused by hundreds of bogus ride-share requests. The requests appear to be legitimate to ride-share services, and they dispatch drivers for pickup that inevitably clog up the city streets. This prevents regular legitimate traffic from arriving at its destination.

During a DDoS attack, attackers use large numbers of exploited machines and connected devices across the internet — including Internet of Things (IoT) devices, smartphones, personal computers, and network servers — to send a flood of traffic to targets.

A DDoS or DoS attack is like a traffic jam

A DDoS attack on a company’s website, web application, APIs, network, or data center infrastructure can cause downtime and prevent legitimate users from buying products, using a service, getting information, or any other access.

## Why do I need DDoS protection?

DDoS(Distributed Denial of Service) attacks **corrupt a network by attacking nodes present in the network thus blocking incoming traffic to websites and risking the loss of confidential data**. These attacks can shut down a website thus affecting the business directly.

## What is AWS security hub

AWS Security Hub is **a cloud security posture management service that performs security best practice checks, aggregates alerts, and enables automated remediation**.

What does Azure Security Center do?

Microsoft Azure Security Center is a set of tools for **monitoring and managing the security of virtual machines and other cloud computing resources within the Microsoft Azure public cloud**. Administrators access the Azure Security Center through the Azure management portal.

What is Google security command center?

Security Command Center helps you strengthen your security posture by evaluating your security and data attack surface; providing asset inventory and discovery; identifying misconfigurations, vulnerabilities and threats; and helping you mitigate and remediate risks.

## Identity service

What is the IAM AWS?

AWS Identity and Access Management (IAM) is **a web service that helps you securely control access to AWS resources**. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

What is Azure Active Directory?

Azure Active Directory (Azure AD) is **a cloud-based identity and access management service**. This service helps your employees access external resources, such as Microsoft 365, the Azure portal, and thousands of other SaaS applications

**Google Identity and Access Management** (IAM) lets administrators authorize who can take action on specific resources, giving you full control and visibility to manage Google Cloud resources centrally.

## The 6 Pillars of the AWS Well-Architected Framework

## 1. Operational Excellence

The Operational Excellence pillar includes the ability to support development and run workloads effectively, gain insight into their operation, and continuously improve supporting processes and procedures to delivery business value. You can find prescriptive guidance on implementation in the [Operational Excellence Pillar whitepaper](https://docs.aws.amazon.com/wellarchitected/latest/operational-excellence-pillar/welcome.html).

### 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 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, and customer needs. 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 data, systems, and assets to take advantage of cloud technologies to improve your security. You can find prescriptive guidance on implementation in the [Security Pillar whitepaper](https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html).

### 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

### Best Practices

Before you architect any workload, you need to put in place practices that influence security. You’ll 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 that adopt the cloud to achieve their security and compliance goals. Because AWS physically secures the infrastructure that supports our cloud services, as an AWS customer you can focus on using services to accomplish your goals. The AWS Cloud also provides greater access to security data and an automated approach to responding to security events.

## 3. Reliability

The Reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it’s expected to. This includes the ability to operate and test the workload through its total lifecycle. You can find prescriptive guidance on implementation in the [Reliability Pillar whitepaper](https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/welcome.html).

### 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

### Best Practices

Before building 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). With AWS, however, most of the foundational requirements are already incorporated or can be addressed as needed.

The cloud is designed to be nearly limitless, so it’s the responsibility of AWS to satisfy the requirement for sufficient networking and compute capacity, leaving you free to change resource size and allocations on demand.

A reliable workload starts with upfront design decisions for both software and infrastructure. Your architecture choices will impact your workload behavior across all six AWS Well-Architected pillars. For reliability, there are specific patterns you must follow, such as loosely coupled dependencies, graceful degradation, and limiting retries.

Changes to your workload or its environment must be anticipated and accommodated to achieve reliable operation of the workload. Changes include those imposed on your workload, like a spikes in demand, as well as those from within such as feature deployments and security patches.

Low-level hardware component failures are something to be dealt with every day in an on-premises data center. In the cloud, however, these are often abstracted away. Regardless of your cloud provider, there is the potential for failures to impact your workload. You must therefore [take steps to implement resiliency](https://docs.aws.amazon.com/whitepapers/latest/disaster-recovery-workloads-on-aws/shared-responsibility-model-for-resiliency.html) in your workload, such as fault isolation, automated failover to healthy resources, and a disaster recovery strategy.

## 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://docs.aws.amazon.com/wellarchitected/latest/performance-efficiency-pillar/welcome.html).

### 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

### Best Practices

Take a data-driven approach to building 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.

Reviewing your choices on a regular basis ensures you are taking advantage of the continually 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. AWS Well-Architected workloads use multiple solutions and enable different features to improve performance

## 5. Cost Optimization

The Cost Optimization pillar includes the ability to run systems to deliver business value at the lowest price point. You can find prescriptive guidance on implementation in the [Cost Optimization Pillar whitepaper](https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/welcome.html).

### 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

### Best Practices

As with the other pillars, there are trade-offs 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 up-front cost optimization.

Design decisions are sometimes directed by haste rather than data, and as the temptation always exists to overcompensate rather than spend time benchmarking for the most cost-optimal deployment. This might lead to over-provisioned and under-optimized deployments.

Using the appropriate services, resources, and configurations for your workloads is key to cost savings

## 6. Sustainability

The discipline of sustainability addresses the long-term environmental, economic, and societal impact of your business activities. You can find prescriptive guidance on implementation in the [Sustainability Pillar whitepaper](https://docs.aws.amazon.com/wellarchitected/latest/sustainability-pillar/sustainability-pillar.html).

### Design Principles

There are six design principles for sustainability in the cloud:

* Understand your impact
* Establish sustainability goals
* Maximize utilization
* Anticipate and adopt new, more efficient hardware and software offerings
* Use managed services
* Reduce the downstream impact of your cloud workloads

### Best Practices

Choose AWS Regions where you will implement workloads based on your business requirements and sustainability goals.

User behavior patterns can help you identify improvements to meet sustainability goals. For example, scale infrastructure down when not needed, position resources to limit the network required for users to consume them, and remove unused assets.

Implement software and architecture patterns to perform load smoothing and maintain consistent high utilization of deployed resources. Understand the performance of your workload components, and optimize the components that consume the most resources.

Analyze data patterns to implement data management practices that reduce the provisioned storage required to support your workload. Use lifecycle capabilities to move data to more efficient, less performant storage when requirements decrease, and delete data that’s no longer required.

Analyze hardware patterns to identify opportunities that reduce workload sustainability impacts by minimizing the amount of hardware needed to provision and deploy. Select the most efficient hardware for your individual workload.

In your development and deployment process, identify opportunities to reduce your sustainability impact by making changes, such as updating systems to gain performance efficiencies and manage sustainability impacts. Use automation to manage the lifecycle of your development and test environments, and use managed device farms for testing.