CHAPTER ONE

Getting Started with Cloud Concepts

Welcome to Chapter 1. In this chapter, you learn what Cloud computing can offer your business. Specifically, this chapter covers the following topics:

- Benefits of Cloud computing
- AWS Cloud economics
- The total cost of ownership
- Reducing costs by moving to the Cloud
- Cloud architecture design principles
- The Well-Architected Framework

By the end of this chapter, you will be able to define the Cloud and describe its many uses. Let's get started!

What Is Cloud Computing?

Imagine you're planning a wonderful camping trip with your friends. You're into the concept of "roughing it," but you have your limits! Do you really need to sew your own sleeping bags, craft your own tents, drag some rocks around to make a fire pit, and rub two sticks together to make a campfire? Probably not! Instead, you might take a trip to your local camping supply store and buy a pup tent kit, a mess kit, and one of those cool camp cook stoves with a little butane fuel container. Once you're geared up, you can free your attention to enjoying the great outdoors in the company of your friends!

Cloud computing is a little like that visit to the store to source your gear, rather than building it all yourself. Instead of managing your own servers in your own office, you can use **the Cloud**, which is a set of services available to you to conduct your business's needs over the Internet. You can store files in the Cloud, manage data in various regions, connect your systems together, and do much more by using Cloud computing. You probably use it every day without even realizing it.

Whenever you open a mobile app to reserve a spot at your local campground, you are using the Cloud to make the connection between the device you hold in your hand, the Internet, and the campground's system for taking your reservation. This is the difference between the client (your phone and its browser or app) and the server (the machine that exists somewhere else and handles your requests).

The Cloud—the machine or "server" that exists in a data center somewhere not too far from you—takes care of the transactions needed to make your camping outing perfect, including gathering your personal information, storing it

safely and privately, notifying the campground to reserve a designated spot, and transferring your payment from your bank account to theirs. It's such a seamless experience that people hardly think about it nowadays, but the technology behind it is worth a deep dive.

Cloud computing brings with it a lot of benefits that can help any business or individual succeed in achieving their goals. It has six specific benefits that are worth understanding well. Diving deep

into these topics will guide you as you start your journey to becoming a Cloud practitioner. So let's go!

Global Reach

If you are managing your own servers, including the hardware, software, and networking capabilities that you need to keep your business systems running, imagine how complex your task becomes when you need to reach customers in the far corners of the Earth. Cloud computing offers the ability to "go global" as quickly as you like.

The idea of going global isn't so much that your systems, now ported to the cloud, aren't already accessible in many different parts of the Internet-connected world. The benefit to Cloud computing on AWS is that you can do it *well*. By deploying your business systems globally, their *latency* is reduced.

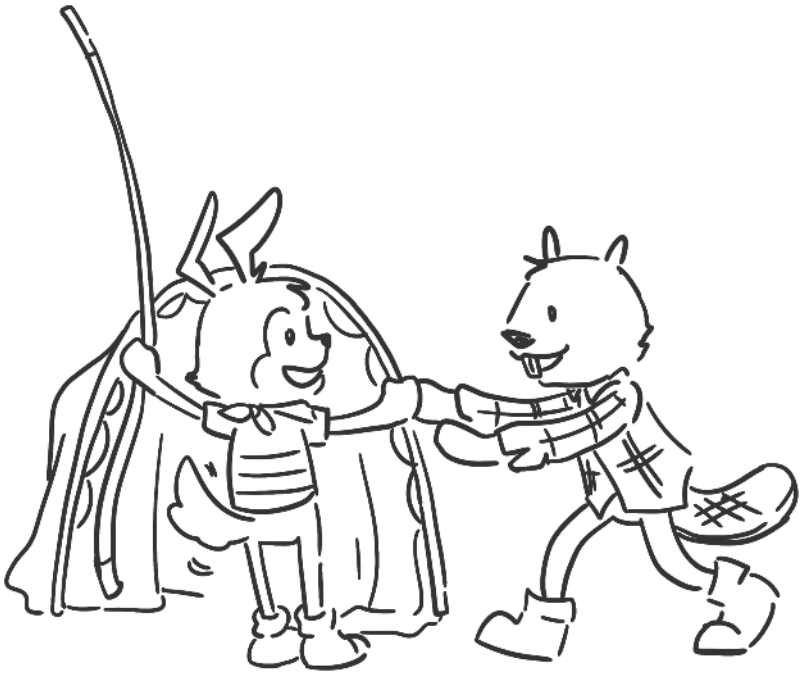
NOTE: *Latency* is the amount of time needed for information to travel across a network. If an application is deployed closer to the user, it takes less time to arrive.

Going global allows folks to deliver a fast experience to their customers worldwide.

Economies of Scale

When you think of yourself and your capabilities, you're only able to achieve a limited number of things each day. That's normal! You need to eat, rest, and practice some self-care during your busy day. There's an expression, however, that "many hands make light work." Within the context of Cloud computing, the same analogy applies.

A huge company such as Amazon can serve thousands upon thousands of customers, and because they have this scale, they can spread costs across many customers and reduce pricing for their services. In this way, we talk about how the AWS Cloud benefits its customers due to its massive economies of scale. More customers means more services delivered at better prices because their usage is combined in the Cloud and prices can be better managed at scale. It's a bit like asking your friend group to pool their money to pay for a campsite rental for a weekend camping trip.



Agility

Everyone loves convenience. The connected world in which many people are fortunate to live allows a car to be summoned at the press of a button, rather than having to rent one every time you visit a new city or own a car if you don't use it all the time. In Cloud computing, similar agility is displayed when a developer can quickly spin up a database and provision it for use by a company in just a few clicks.

We are so used to doing this nowadays that it seems trivial, but imagine the days before Cloud computing made it easy. You can make decisions much more quickly now that so many resources are a click away on an Internet-enabled device connected to the Cloud. Your ability to conduct experiments and change your mind is enhanced as well. Within a Cloud computing context, you are able to "try before you buy," sampling the ability of a various Cloud-connected service before deciding if it's right for you, turning it off if it doesn't meet your needs, or ramping up your usage if it proves just right.

Elasticity, Scalability, and Availability

When you are trying to make plans to build something for use by different types of customers worldwide, it can be daunting to try to make plans about how much compute you need, how much storage you need, what services you have to leverage and how they should all connect. Cloud computing solves this problem in a really neat way.

NOTE: In the context of the Cloud, **compute** refers to the processing power, memory, networking capability, and storage used for software computation.

AWS offers a suite of highly available online services that powers workloads, providing secure, *elastic* (resizable) compute capacity in the Cloud. You learn more about these services in Chapter 4.

Instead of asking you to plan up front how much computing power your applications need, you can set up your Cloud computing infrastructure mirroring your current needs and then quickly add more capacity as the demand for your services expands or contracts. With Cloud computing, this is possible. You can quickly scale your infrastructure's size and scope up and down, depending on the resources needed or not needed at a given time.

Cloud computing offers the promise of high availability, meaning your sites will be up and running with minimal downtime with a very high percentage of time that your workloads in the Cloud are available to be used.

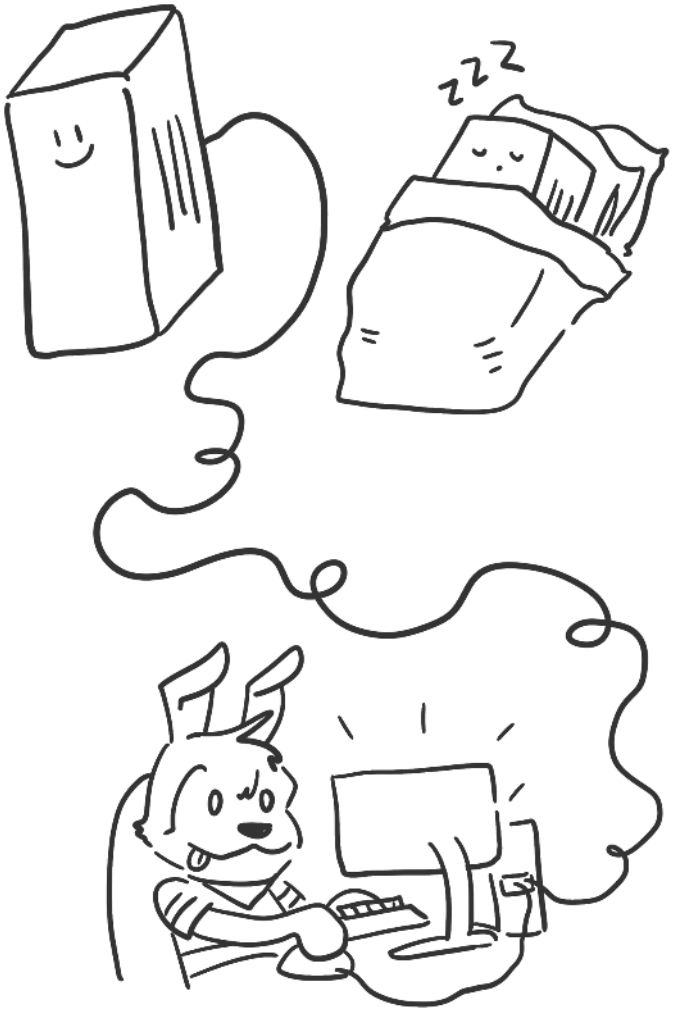
NOTE: In Cloud computing, a **workload** is defined as "a set of components that together bring business value." Websites and dashboards reflecting the output of databases are examples of workloads.

Availability is the amount of time your workloads are available to be used, divided by a given period, such as a month or a year. So, if your website, for example, is down for five minutes every year for scheduled or unscheduled reasons, its availability is described as 99.999 percent. This would be described as highly available, with "five nines" of availability, and would be a promise you could make to your customers about your reliability. Since you pay more for business systems that require very high uptime, you need to decide how much downtime you can tolerate in order to budget correctly.

How does AWS make sure that services can be available? In general, it helps enable three elements.

- That there are no single points of failure
- That systems are redundant and fail over at reliable crossover points

- That failures are detected as they take place so that they can be tracked



Pay-As-You-Go Pricing

Imagine you know you want to create an online business, website, or place to store digital assets,

but you have no idea where to start. You could consider buying all the hardware and software, installing, patching, maintaining, and storing it in your office. This is called a **fixed expense**. Or, you could look to the Cloud to take care of these tasks. The variable nature of Cloud computing means you can focus your budget on **variable expenses** and pay as you go, building your business in the Cloud.

Moving from fixed expenses is one of Cloud computing's core benefits. The main idea behind this shift is embracing the concept that you want to move away from large one-time purchases and instead embrace expenses that fluctuate, based on usage. Cloud computing offers services that help you scale your expenses and manage your budget according to the way you intend to use them.

Focus on Business, Not Managing Infrastructure

Before the days of Cloud computing, if you wanted a place to store your data, run a website, and handle incoming and outgoing transactions either locally or over the Internet, you had to buy or rent your own server. You probably then had to store it somewhere cool and safe, install software that you needed to keep it up-to-date, and connect it to other servers via networks that you might have had to set up yourself.

With Cloud computing, these tasks can be outsourced. Instead of buying your own equipment and running your own private data center, you can benefit from the huge data centers available for you.

What Are Cloud Economics?

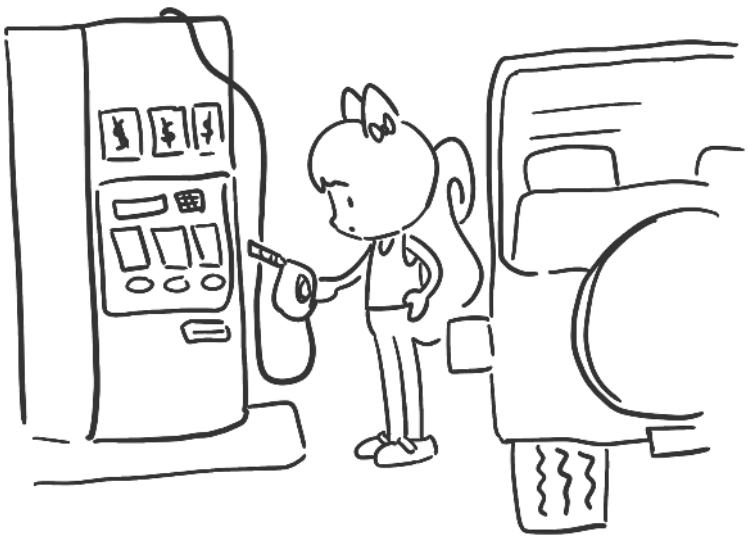
When we talk about *Cloud economics,* what are we referring to? It's a term used to define the financial benefits and costs associated with using Cloud computing, as opposed to other ways of

managing infrastructure. By moving to the Cloud, how much money is your business going to spend and, ideally, save? Quantifying the return on investment (ROI) of this method of hosting infrastructure is an important aspect of building a business that will scale.

This section quantifies the various aspects of Cloud economics and explains how they pertain to your use cases.

The Total Cost of Ownership

How much does it cost to host your business, school, or private enterprise in the Cloud? To answer this question, understanding a little vocabulary is necessary. Calculating the total cost of ownership (TCO) of your on-premises, hybrid, or Cloud infrastructure involves understanding the costs incurred to host, run, and manage the life span of your Cloud workloads. TCO in general is calculated as a sum of your initial cost and your maintenance costs, minus the asset's devaluation over time. But let's start by understanding a few baseline concepts that impact that cost.



NOTE: You can use the handy AWS Pricing Calculator to help forecast and analyze the cost

of running AWS workloads at `https://calculator.aws`.

OpEx

Operational expense (OpEx) refers to the ongoing expenses incurred when running a business. A useful example is an everyday expense that you have to pay to run a server, including the electrical bill and the bill you pay to keep the server room cool. Any rent you have to pay to keep your business up and running, such as renting the server hardware and insuring it, as well as meeting payroll and managing inventory, are all part of your OpEx. If there's an ongoing expense that's needed to keep your organization's lights on day in and day out, that's OpEx. Your TCO is impacted by the

long-term operational expenses that add up as your business grows and scales.

CapEx

Capital expense (CapEx), on the other hand, defines those large expenses that occur less regularly, such as buying a building, vehicles, or machinery to support your business. Fixed assets, often termed *property, plant, and equipment* (PP&E), are another example of CapEx. In general, consider CapEx expenses as the spending you have to do on physical assets. Your TCO is impacted immediately by CapEx purchases.

In the context of Cloud computing, it's vital to understand how your shift from an on-premises model into a Cloud model will impact your budget. **Cloud computing, with its flexible pay-as-you-go model, allows you to shift your expenses from CapEx to OpEx.** This shift will have implications for your budget forecasting and taxes, so it's important to understand the difference and how it will impact your bottom line. Managing the balance between your CapEx and OpEx costs is an important aspect of managing

your budget and predicting long-term impacts to your TCO.

Labor Costs

No matter how much automation is available, there are still labor costs associated with doing business, whether in the Cloud or on-premises.

But maintaining your own data center has significant costs that can be mitigated by moving to the Cloud. To run your data center on-premises, you need to budget for a mix of CapEx and OpEx. In particular, you need staff to support and maintain the following:

- Server and network infrastructure
- Physical security
- Facility maintenance
- Climate control
- Fire suppression systems
- Backup power systems

Each business should understand the budgetary ramifications of moving their infrastructure fully or partially into the Cloud, as the labor costs shift from a more local, in-house expenditure to being attached to a data center.

Software Licensing

An important aspect of provisioning infrastructure is paying for the software license that makes them usable for the tasks required. Licenses can be tricky to manage, as they are priced according

to the numbers of users and the type of hardware required. If you are creating a new environment in the Cloud, provisioning software to support it includes licensing it.

AWS supports several types of licensing models to help you better manage your software. AWS License Manager can help you manage this process.

- **BYOL:** Bring your own license. If you already own a software license, you can port it into the Cloud.
- **Granted licenses:** These licenses come with software purchased from AWS Marketplace.
- **LI:** An AWS License Included model for new environments, such as a license for Microsoft SQL Server.

The details involved in licensing can be daunting, and in an on-premises environment they are up to you to handle. By using Cloud services to handle licensing, you have more flexibility as your business grows to help your software evolve to suit.

How Can You Reduce Costs?

No business ever in the history of business has wanted, for the fun of it, to pay more than was strictly necessary for a given service. Their profitability depends on their ability to balance their costs against their revenue. So what are some tips that Cloud computing can offer to help control infrastructure costs? Let's dive in.

Right-Sizing Infrastructure

It's a gamble when setting up an on-premises server room where you have to "rack and stack" all your hardware. What if you set up an immense server room with a huge amount of compute, network, and storage, and then your business shrinks or changes in scope entirely? What if you don't buy enough hardware and your company goes viral? What if you accidentally install the wrong software on your boxes or experience a catastrophic outage causing significant down-time? Making guesses about how much stor-age and compute you need is a risky business for your business. It's better to "right-size" your infrastructure.

The right-sizing process in AWS Cloud involves making sure your capacity matches your work-loads in a granular fashion, and it's a great way to control costs. By continually watching your infra-structure for opportunities to efficiently increase or decrease capacity to match workloads, you can fine-tune your operational expenditures. By continually monitoring and enforcing a sys-tem of tagging your resources, you can right-size

your infrastructure when you build it natively in the Cloud or when you move your infrastructure from on-premises into the Cloud. In this way, you avoid wasting money on unused or underused resources.

Automation

It's one thing to build an infrastructure native to the Cloud or to port your on-premises setup to the Cloud. You can take your Cloud presence to the next level by automating many steps

required for its maintenance. Automation involves the execution of a process behind the scenes, either via a timer, in response to an event, or ad hoc, taking the error-prone human guesswork out of maintenance. One service that supports automation is AWS CloudFormation, designed to help manage your infrastructure using templates, automating resource management across your organization.

A good example of one type of automation is a sort of "dress rehearsal" of building, deploying, and rolling back instances of services to ensure that changes can be deployed into production seamlessly, avoiding costly downtime. Another example of this type is automating the application of patches to operating systems. You could automate scaling your systems' capacity as demand rises and falls to avoid more costly guesswork. You could also automatically deploy security improvements across the board to make sure your entire infrastructure is equally secure. You could automate the documentation of your patches, changes, and fixes to make sure they are reproducible for the next cycle, and you could make sure, via automation, that your system can

recover quickly from errors, freeing you to concentrate on scaling your business, rather than worrying about manually managing the infrastructure that supports it.

Reduce Compliance Scope

One aspect of running a business is keeping in compliance with various regulations. Examples include the Health Insurance Portability and Accountability Act (HIPAA), which regulates how patient data is handled, and Payment Card

Industry Data Security Standard (PCI) compliance, which regulates the handling of credit cards. General Data Protection Regulation (GDPR) is a compliance standard required in the European Union. Making sure your business stays compliant with all these systems can be very expensive and involves extensive reporting. AWS supports more than 140 compliance standards, helping you manage your responsibilities for keeping in compliance with various tools you can use in the Cloud. Managing compliance is part of the *shared responsibility model*, which you learn about later.

Managed Services

Some AWS services offer **managed services**, which abstract away some of their complexity so that you can focus on their business value. Their infrastructure and detailed setup is taken care of behind the scenes. An example of this type of service includes the database service

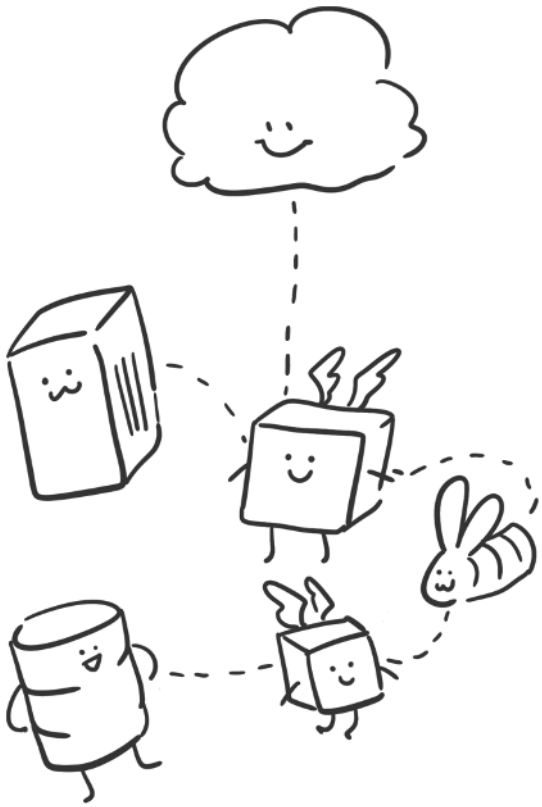
DynamoDB, which hides the infrastructure setup, and the Relational Database Service (RDS) where the database engine is managed. While these services can involve higher pricing, the invisible underlying infrastructure involved in setting them up and managing them behind the scenes—which you don't have to worry about—makes adopting them a cost savings in many instances.

How Should You Design Your Cloud Architecture?

It can seem daunting to think about designing an entire Cloud architecture from scratch, but take a moment to think about all the times you've interacted with a suboptimal system on the Internet. Remember when you stared at a page on the

Web, waiting for images to download, or experienced an outage that caused you to lose work?

Outages can be catastrophic, grounding thousands of flights, or they can last just a few seconds, causing loading to stop for a short time. If you are in the business of managing a company or service that's reliant on the Internet, it's up to you to make sure you design a robust infrastructure. The following sections explain four best practices to help you do that.

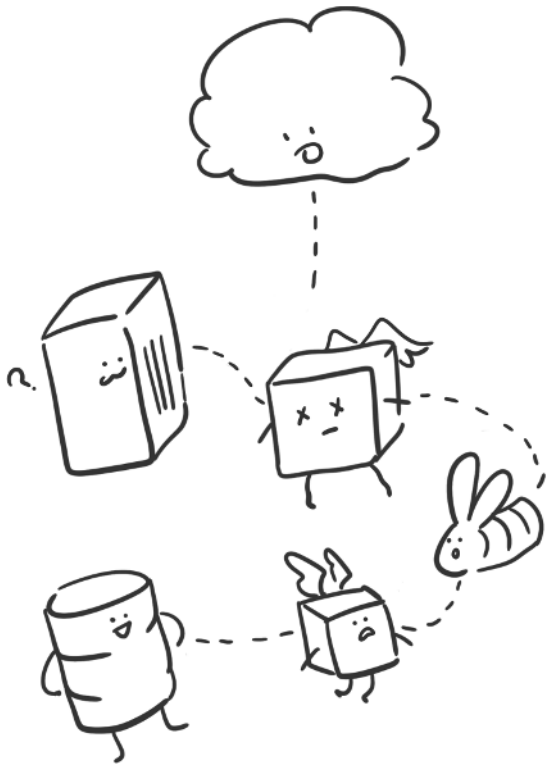


Design for Failure

Outages. Everyone hates them, from customers who can't add their item to their cart to retailers who find that their sales pipeline suddenly stops flowing. How can you design your infrastructure to *expect* failure and handle failure when it happens? Failure management is part of the Reliability pillar of the *Well-Architected Framework,* which is covered in the next section. But it's also necessary to expect failures to happen and to adapt your infrastructure design to handle it well. In other words, think pessimistically and design for fault-tolerance. There are several techniques available to ensure this best practice:

- Building a **distributed architecture** that avoids single points of failure. Use multiple availability zones (covered more in Chapter 3) so that if there's a failure in one, your workload can "fail over" to another. Failures tend to be contained to single AZs.
- Make sure your data and components are **redundant**, replicated across multiple resources, using load balancers, for example, to distribute traffic across redundant resources.

- **Isolate failures** so that their impact is minimized. Use *microservices*—small, independent services you can connect to via APIs—for example, which can be designed to fail without taking down an entire system.
- Plan for disasters and for **disaster recovery**, using well-documented best practices.
- Test your system's resilience using tooling to simulate failures; this technique is called **chaos engineering**.



Decouple Components vs. Monolithic Architecture

You're going to need to understand the trade-offs between creating a monolithic versus a decoupled architecture as you plan for failure.

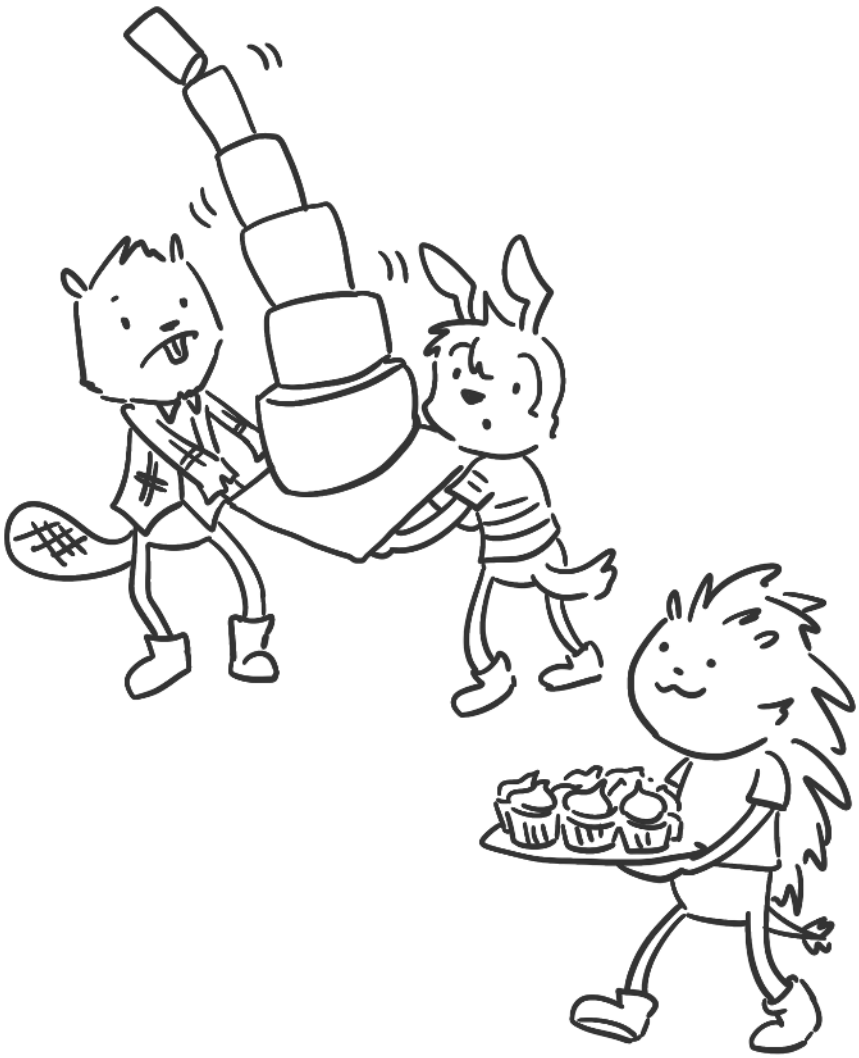
A **monolithic** architecture is described as one workload that does many things. It grows as business needs grow, and its processes are tightly coupled or connected. If one part of it goes down, the rest will, too.

A **decoupled** architecture can be made of many microservices, which, despite their name, aren't necessarily small. They are, instead, autonomous and specialized. Their job is to encapsulate a business capability. They should be able to be built, tested, deployed, and changed independently.

An example of a monolithic service is a service that encapsulates users, posts, and responses to posts.

A decoupled microservice architecture would split those services into three: users, posts, responses. If one goes down, the rest don't go down with it. This is good infrastructure design.

NOTE: Using AWS services such as Lambda and Amazon Elastic Containers are good choices when building a decoupled microservice architecture.



Implement Elasticity in the Cloud vs. On-Premises

The **elasticity** of Cloud computing is one of its great capabilities. If there's a sudden surge of traffic in an on-premises data center and servers go down, the failure can be much more challenging to a business than when one part of an elastic Cloud architecture fails. In an elastic, Cloud-enabled architecture, when one part fails, another part can become available as new resources are provisioned. Traffic can re-route to these new resources, and the failure is invisible to customers.

Think Parallel

Finally, it's important to think in parallel when designing your Cloud architecture. This means designing your system to be able to handle multiple requests simultaneously. Services can be paired to offload jobs to each other and ensure that the workload is handled seamlessly.

An example might be one microservice designed to intake images from hundreds of concurrent users and store them in a database, while a second service waits for a response from

each upload and analyzes it for its content using machine learning. Moving away from designing sequential processes and designing instead toward a multithreaded, asynchronous architecture—for example, using a multithreaded process to put images into S3 while asynchronously reacting to each put—will improve its speed and reliability.

The Well-Architected Framework

There's a useful *mental model* called the Well-Architected Framework that can help you design and operate your business systems in the Cloud. This framework sets you up for success by helping you understand the pros and cons of decisions you have to make while building in the Cloud. It is divided into six sections or pillars, and

it's really helpful to understand these well as core concepts for the Cloud practitioner to master. The following sections discuss these pillars.

Pillar 1: Operational Excellence

The Operational Excellence pillar is about improvement, not only of technology but also about people and processes. You should be able to run your workloads, learn about how well or poorly they operate, and improve them throughout their lifecycle. Check out this pillar's five design principles:

- **Operations as code:** You can design your whole infrastructure using version-controlled code,

which allows you to update it more easily and use it for automating operational procedures.

- **Frequent, small changes:** Design your workloads so that you can update them frequently with the ability to reverse your changes without issues.
- **Frequent refinements:** Small, incremental changes help you refine your procedures gradually.
- **Anticipate failure:** In life, optimism is great, but in infrastructure and application code, it's better to expect the worst and be ready to learn from failures. You can set up tests to see how well your workloads and your team respond to disasters.
- **Learn from failures:** Also as in life, share what you learned from your failures across your team and your organization so the same mistake won't be made twice!

Pillar 2: Security

A key aspect of AWS Cloud computing is its high emphasis on security. This pillar focuses on protecting information and systems. It's important for customers to be sure that their data and

infrastructure are secure. Moving to the Cloud, in fact, can act as a means to enhance security. These seven principles outline how security is handled in the AWS Cloud:

- **Implement a strong identity policy:** This means using the *principle of least privilege*, meaning that you ensure that each element of your architecture has appropriate authorization policies in place.
- **Enable traceability:** Make sure you can trace any changes or events in your architecture by capturing logs and metrics.
- **Apply security at all layers:** Each layer of your architecture should have multiple security controls.
- **Automate security:** Use software to manage your security seamlessly.
- **Protect data in transit and data at rest:** Understand the various ways your data moves through your system and protect it by classifying it according to sensitivity. This is a good opportunity to use encryption to protect data in transit and access management for data at rest.

- **Keep people away from data:** Human error can often cause security incidents, so use tools to avoid the need to manipulate data manually.
- **Prepare for security events:** Expect the best, but prepare for the worst. Make sure your systems are ready for incidents and run simulations to make sure that such problems can be detected, investigated, and recovered from.

Pillar 3: Reliability

Reliability in the Cloud means that workloads are monitored for their availability and they can recover from disruptions. A reliable system also has tests to determine and mitigate potential points of failure. Agility, elasticity, and scalability also factor in to ensure the reliability of a system, and these aspects can all be automated to reduce human error. A good example of a reliable system

is one where when one component sees a spike in demand—a video goes viral, for example—an automation adds resources to support this additional load and removes the resources when the load goes down.

The Reliability pillar includes these five design principles:

- **Automatic recovery:** Use automation to monitor your workloads and trigger a notification and recovery process to repair a failure.
- **Testable procedures:** Make sure your recovery procedures are solid by setting up tests to simulate failures and recovery solutions.
- **Horizontal scaling:** Instead of having one large resource, use many smaller resources in your architecture so that failures won't impact such a large area.
- **Elastic capacity:** When demand exceeds supply, automate expanding or removal of resources.
- **Automated changes:** Use automation as well to make changes to your infrastructure.

Pillar 4: Performance Efficiency

When we talk about performance efficiency, we're referring to using your available resources in the most efficient way to meet your workload's requirements. Performance has to do with a system's response speed and use of memory. This efficiency needs to keep up with any changes in your system's demand and any changes in technology, as well. This pillar includes these five principles:

- **Democratize technology:** Let AWS take care of managing complex tasks in the Cloud, rather than asking in-house staff to manage it. This way, you can focus on your business needs and your customer's experience.

- **Go global in minutes:** Deploy your workloads around the world using multiple AWS Regions. Your customers will appreciate the lower latency and faster service.

NOTE **AWS Regions** are physical locations around the world that hold a cluster of data centers that host AWS resources. Inside each region there are at least three Availability Zones, discussed in Chapter 3.

- **Go serverless:** Using serverless technologies means you don't have to maintain physical servers or manage and pay for physical server operations.

NOTE: **Serverless** means using architectures that remove the need for servers—an S3 storage bucket instead of a web server, for example.

- **Experiment!** With your Cloud environment, you can test and experiment to see what configurations work best.
- **Consider mechanical sympathy:** This is a fancy way to say "Use the tool that's right for

the job." For example, use a NoSQL database when it makes sense over a relational database.

Pillar 5: Cost Optimization

The Cost Optimization pillar is all about—you guessed it—delivering the highest value at the lowest price point. Learn how to save money by following these five design principles:

- **Invest in knowledge:** Make sure that your business is well equipped to make good decisions to optimize cost.

- **Adopt a consumption model:** While you might be tempted to forecast how much compute you might need, it's better to lean on the elasticity of the Cloud and increase and decrease usage proactively to save.
- **Measure efficiency:** Knowledge is power, so measure and keep track of your costs and savings as you increase and decrease compute, networking, storage, and governance.
- **Stop spending money on IT infrastructure:** Let AWS do that for you! You don't need to rack and stack your own data centers if you've invested in the Cloud, just like you don't need to construct your own tent if a ready-made one is available.

- **Analyze:** Identify which elements of your workloads are consuming so you can know where to optimize

them to reduce costs and attribute those costs to the appropriate people running them.

Pillar 6: Sustainability

According to the United Nations, sustainable development is "development that meets the needs of the present without compromising the ability of future generations to meet their own needs." In this spirit, this sixth pillar of the Well-Architected Framework includes the goal of helping make organizations more sustainable via reducing the harms that infrastructure can cause to our environment.

Take a look at these best practices:

- **Understand the impact:** Know the impact of your workloads on the environment so that you can target inefficient ones and make their impact less harmful.
- **Establish goals:** Establish sustainability goals and work toward meeting them on a timeline that works for you.
- **Maximize utilization:** Fewer resources that are utilized at a higher percentage are more efficient than more workloads that are less utilized.

- **Adopt efficient hardware and software:** Be aware and ready to implement more efficient services that become available to you.
- **Use managed services:** When many customers share workloads, they are more efficient, so prioritize shared, managed services.
- **Reduce downstream impact:** Constantly upgrading devices (like the one you used to reserve your campground) causes environmental impact, so make sure your infrastructure can be used on current devices, continually testing to ensure their compatibility.