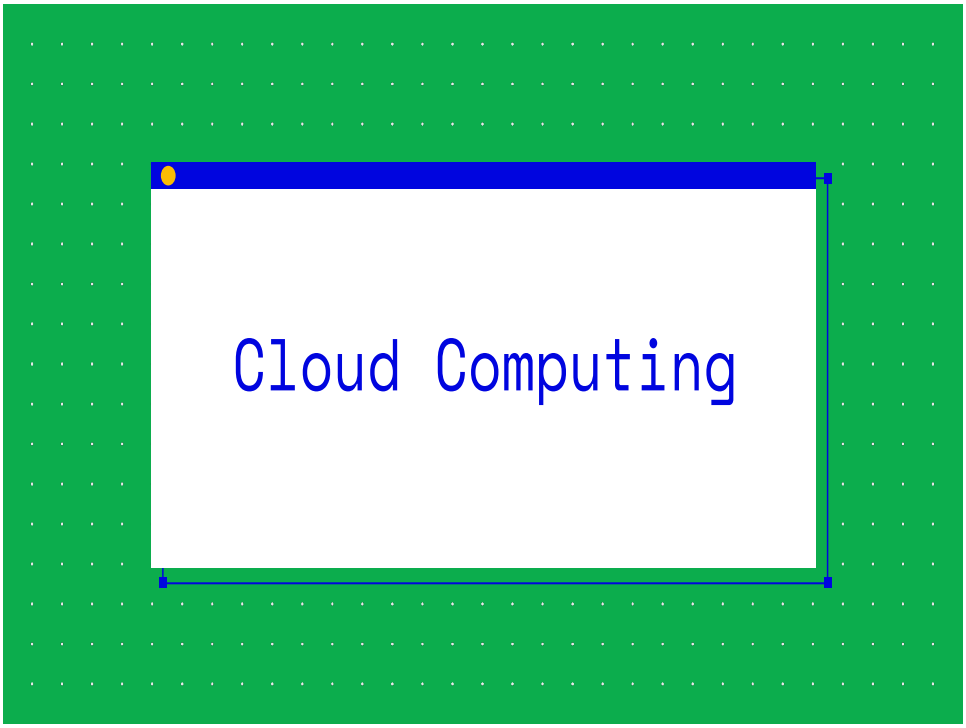
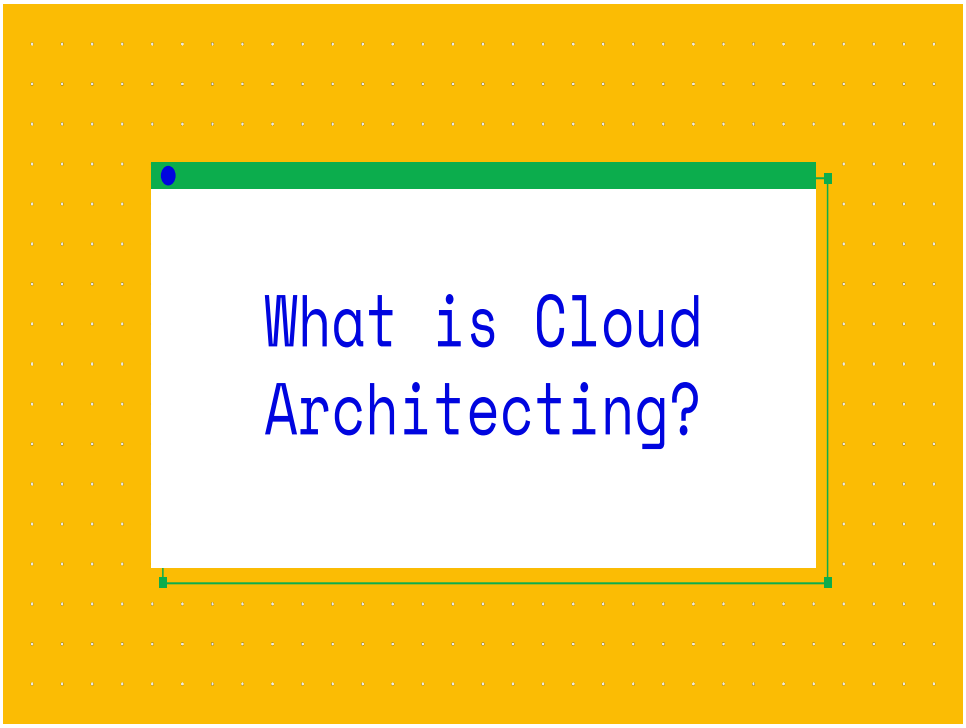

Cloud Computing

Agenda

1. What is Cloud Computing?
2. What is Cloud Architecting?
3. AWS Well-Architected Framework
4. Best Practices
5. AWS Global Infrastructure
6. Questions


What is Cloud
Architecting?

Architectural Need

Around 2000, Amazon was struggling to make its shopping website highly available and scalable

- To understand what **cloud architecting** is and why it's important,
  - first consider an example of what **software development** is like in its absence.
- **Amazon**
  - **Founded** by Jeff Bezos from his garage in Bellevue, Washington, on July 5, **1994**.
  - Around **2000**, Amazon was trying to **create an ecommerce service** that would enable third-party sellers to build their own online shopping sites on top of the Amazon ecommerce engine.
  - The company was struggling to make its new shopping website highly available and scalable.

Origins of AWS (1 / 2)

According to AWS CEO Andy Jassy, at the time,

- Amazon ecommerce tools were “a jumbled mess”
- **Apps** & **Architectures** were built without proper planning
- was difficult to separate services from each other

Solution:

- Amazon created a set of well-documented APIs,
- which became the company’s standard for service development

- In a TechCrunch interview about the genesis of AWS, AWS Chief Executive Officer (CEO) Andy Jassy said that in the beginning, the Amazon ecommerce tools were a “jumbled mess.”
    - Applications and architectures were being built without proper planning.
    - Jassy also said that it was “a huge challenge to separate the various services to make a centralized development platform.”
- The solution to this problem was to create a set of well-documented Application Programming Interfaces (APIs) to organize the development environment.

Origins of AWS (2 / 2)

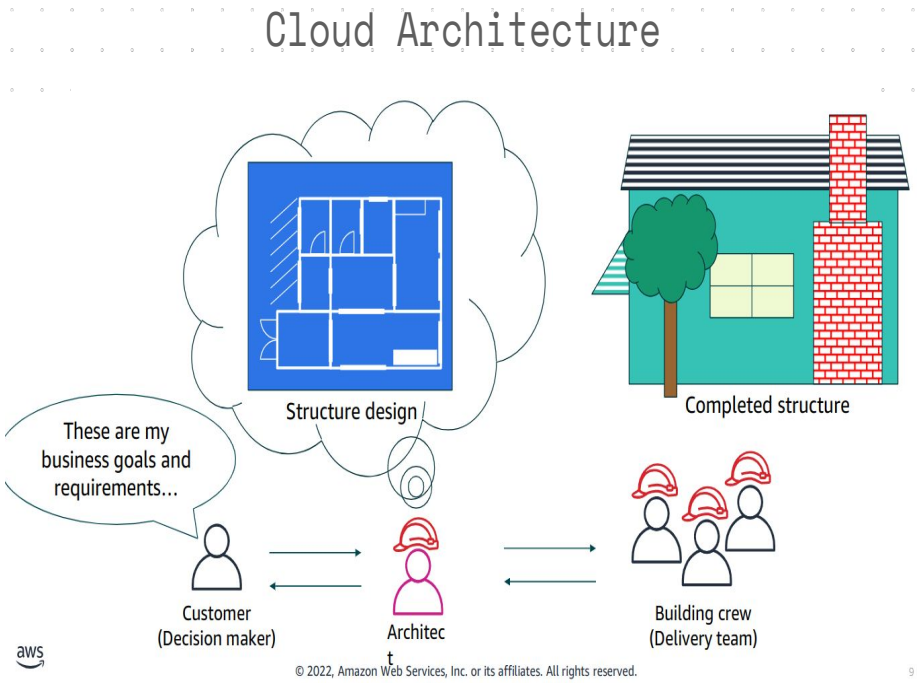
Amazon still struggled to build apps quickly

- Database, compute, and storage components took 3 months to build
- Each team built their own resources,
  - with **no** planning for **scalability** or **re-usability**

Solution:

- Amazon built internal services to create
  - HA, scalable, and reliable architectures on top of its infrastructure.
- In **2006**, Amazon started selling these services as **AWS**.

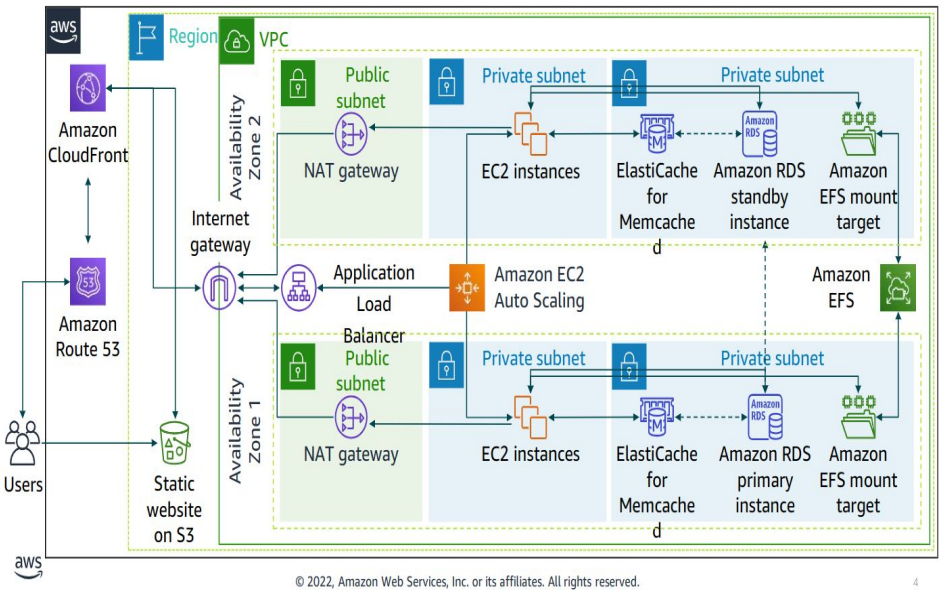
- However, Amazon still struggled to build applications quickly as the company grew and more software engineers were hired.
  - It took 3 months to build database, compute, and storage components for an entire project that was expected to take 3 months.
  - Each team built their own resources without planning for scalability or reusability.
- The solution was to build internal services to create highly available, scalable, and reliable architectures on top of the Amazon infrastructure.
- In 2006, Amazon started selling these services as AWS

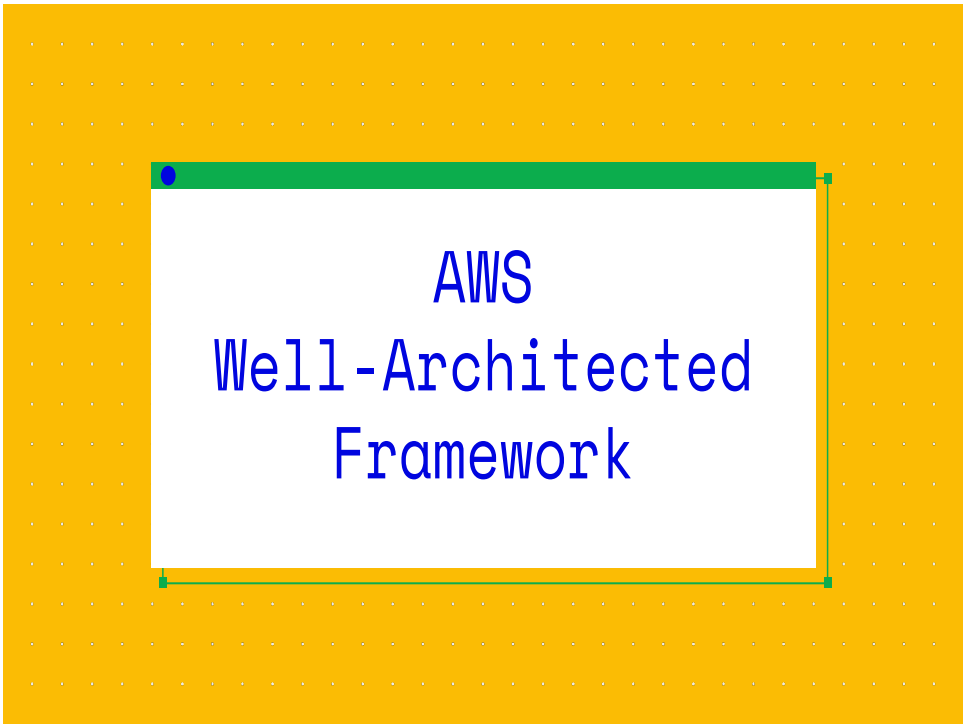


- So, what is cloud architecture?
    - Cloud architecture is the practice of applying cloud characteristics to a solution that uses cloud services and features to meet an organization's technical needs and business use cases.
    - A solution is similar to a blueprint for a building.
- Software systems require architects to manage their size and complexity.
- Cloud architects:
    - Engage with decision makers to identify the business goals and the capabilities that need improvement.
    - Ensure alignment between technology deliverables of a solution and the business goals.
    - Work with delivery teams that are implementing the solution to ensure that the technology features are appropriate.

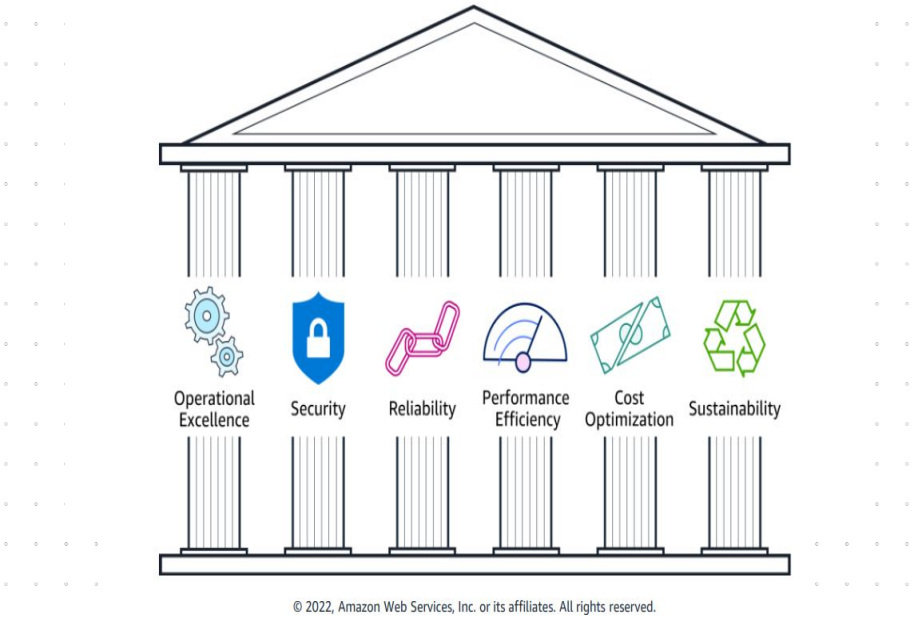
- Having well-architected systems increases the likelihood that the technology deliverables will help meet business goals

Example of a Cloud Architecture




AWS
Well-Architected
Framework

Pillars of the AWS Well-Architected Framework



- The **AWS Well-Architected Framework** is organized into six pillars:
    - Operational Excellence,
    - Security,
    - Reliability,
    - Performance Efficiency,
    - Cost Optimization, and
    - Sustainability.
- The first five pillars have been part of the framework since the framework's introduction in **2015**.
- The sustainability pillar was added as the sixth pillar in **2021** to help organizations learn how to minimize the environmental impacts of running cloud workloads.

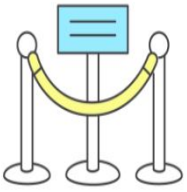
- The **AWS Well-Architected Framework** is a guide that provides a consistent approach to evaluate cloud architectures, and guidance to help implement designs.
    - It documents a set of foundational questions and best practices that enable you to understand if a specific architecture aligns well with cloud best practices.
    - AWS developed this framework after reviewing thousands of customer architectures on AWS.

Operational Excellence Pillar

- The ability to run and monitor systems
- To continuously improve supporting process and procedures

Deployed

Updated



Operated

- The **Operational Excellence Pillar** addresses the ability to run systems and gain insight into their operations to deliver business value.
    - It also addresses the ability to continuously improve supporting processes and procedures.
- When you design a workload for operations, you must be aware of how it will be deployed, updated, and operated.
    - Implement engineering practices that align with defect reductions and quick, safe fixes.
    - Enable observation with logging, instrumentation, and business and technical **metrics** so that you can gain insight into what is happening inside your architecture.

- In AWS, you can view your entire **workload** (apps, infrastructure, policy, governance, and operations) **as code**.
    - It can all be defined in and updated using code.
    - This means that you can apply the same engineering discipline that you use for app code to every element of your stack.

Reliability Pillar

- Recover quickly from infrastructure or service disruptions
- Dynamically acquire computing resources to meet demand
- Mitigate disruptions such as:
  - Misconfigurations
  - Transient network issues

- It can be difficult to ensure reliability in a traditional environment.
  - Issues arise from single points of failure, lack of automation, and lack of elasticity.
- By applying the best practices outlined in the **Reliability Pillar**, you can prevent many of these issues.
  - It will help you and your customers to have a properly designed architecture with respect to high availability, fault tolerance, and overall redundancy

Performance Efficiency Pillar

- Choose efficient resources and maintain their efficiency as demand changes
- Democratize advanced technologies
- Employ mechanical sympathy
  - use a tool or system with an understanding of how it operates best
  - use the technology approach that aligns best to what you are trying to achieve

- When you consider performance, you want to maximize your performance by using computation resources efficiently.
  - You also want to maintain that efficiency as the demand changes.
- It is also important to democratize advanced technologies.
  - In situations where technology is difficult to implement yourself,consider using a vendor.
  - By implementing the technology for you, the vendor handles the complexity and the knowledge, freeing your team to focus on more value-added work.
- Mechanical sympathy is when you use a tool or system with an understanding of how it operates best.
  - Use the technology approach that aligns best to what you are trying to achieve.
  - For example, consider data access patterns when you select database or storage approaches.

Cost Optimization Pillar

- Measure efficiency
- Eliminate unneeded expense
  - Understanding how efficient your current architecture is in relation to your goals
  - can remove unneeded expense
- Consider using **managed service**
  - they operate at cloud scale
  - they can offer a lower cost per transaction or service

- Cost optimization is an ongoing requirement of any good architectural design.
- The process is iterative, and it should be refined and improved throughout your production lifetime.
- Understanding how efficient your current architecture is in relation to your goals can remove unneeded expense.
- Consider using managed services because they operate at cloud scale, and they can offer a lower cost per transaction or service

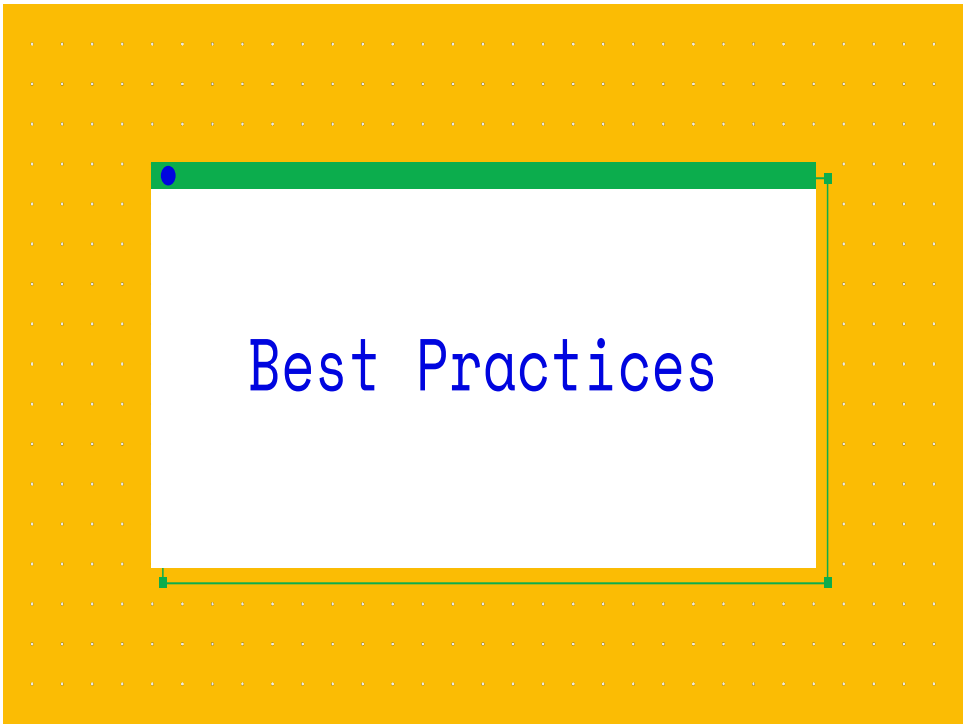
Sustainability Pillar (1/2)

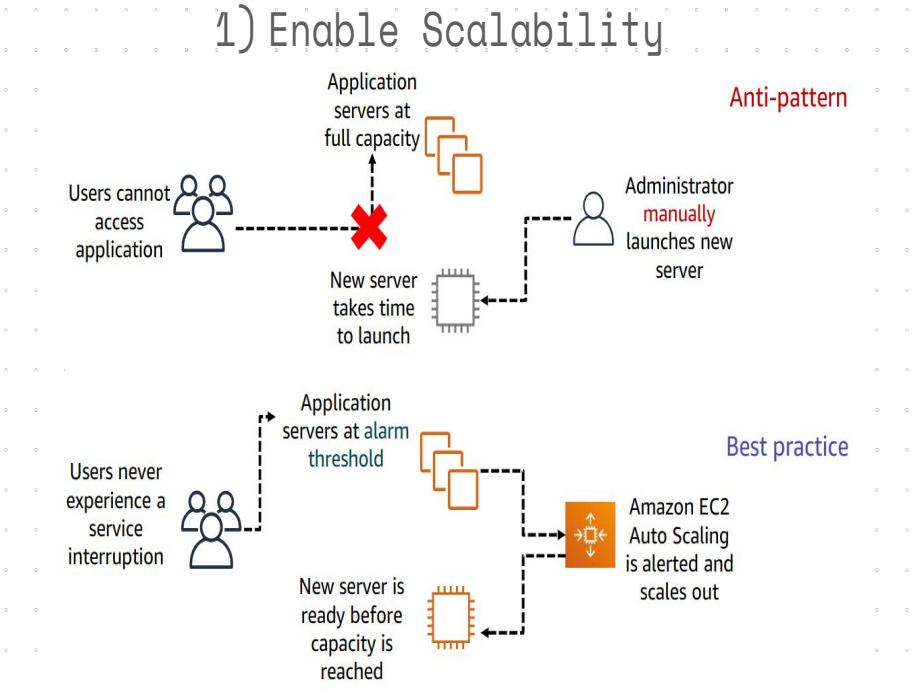
- Understand your impact
- Establish sustainability goals
- Maximize utilization
- Anticipate and adopt new, more efficient hardware and software offerings
- Reduce the downstream impact of your cloud workloads

Sustainability Pillar (2/2)

- Build architectures that maximize efficiency and reduce waste.
- Focused on energy reduction and efficiency across all components of a workload
- This effort can range from the
  - initial selection of an **efficient programming language**,
  - adoption of **modern algorithms**,
  - use of **efficient data storage** techniques,
  - deploying to correctly sized and **efficient compute infrastructure**

- **Sustainability Pillar** addresses the ability to build architectures that **maximize efficiency and reduce waste**.
- **Sustainability** in the cloud is a continuous effort focused primarily on **energy reduction and efficiency** across all components of a workload by achieving the maximum benefit from the resources provisioned and minimizing the total resources required.
- This effort can range from the
  - initial selection of an **efficient programming language**,
  - adoption of **modern algorithms**,
  - use of **efficient data storage** techniques,
  - deploying to correctly sized and **efficient compute infrastructure**, and
  - minimizing requirements for high-powered end-user hardware

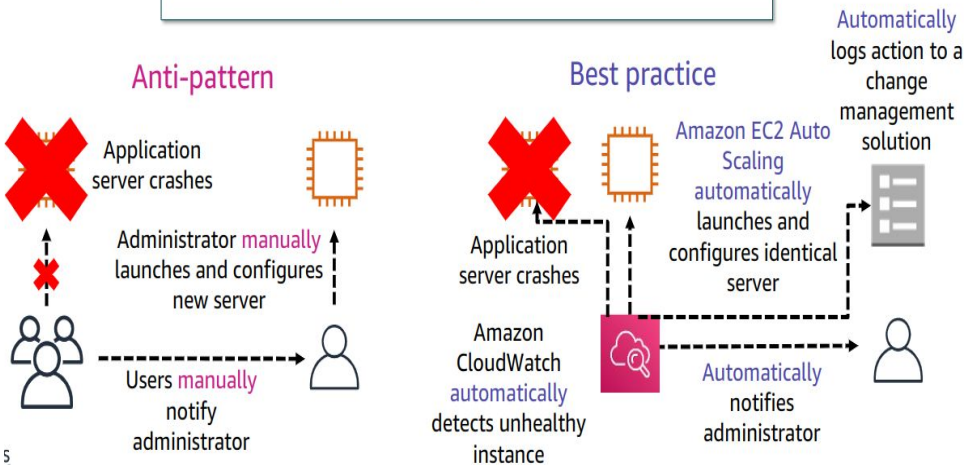

Best Practices



- Ensure that your architecture can handle changes in demand

2) Automate Environment

Where possible, automate the provisioning, termination, and configuration of resources.



3) Treat Resources as Disposable

Take advantage of the dynamically provisioned nature of cloud

Anti-pattern

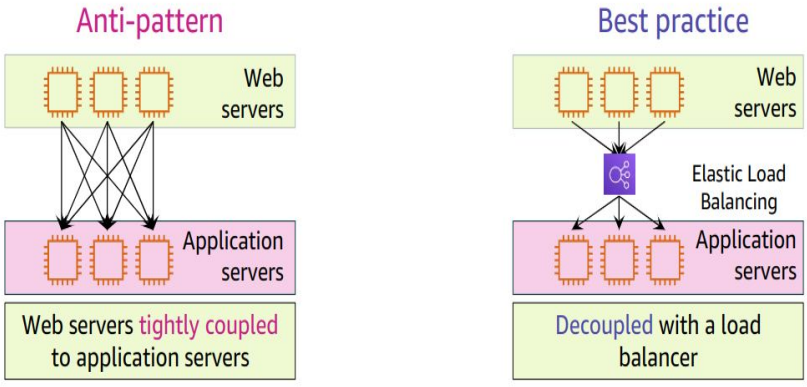
- Over time, different servers end up with different configurations
- Resources run when they're not needed
- Hardcoded IP addresses prevent flexibility
- It can be difficult or inconvenient to test new updates on hardware that's in use

Best practice

- Automate deployment of new resources with identical configurations
- Terminate resources that are not in use
- Switch to new IP addresses automatically
- Test updates on new resources, and then replace old resources with updated ones

5) Design Services, not Servers

Design architectures with independent components.



5) Design Services, not Servers

Use the breadth of cloud services

Don't limit your infrastructure to servers

Anti-pattern

- Simple applications run on persistent servers
- Applications communicate directly with one another
- Static web assets are stored locally on instances
- Backend servers handle user authentication and user state storage

Best practice

- When appropriate, consider using containers or a serverless solution
- Message queues handle communication between applications
- Static web assets are stored externally, such as on Amazon Simple Storage Service (Amazon S3)
- User authentication and user state storage are handled by managed AWS services

6) Choose the Right Database Solution

What to consider when choosing a database solution

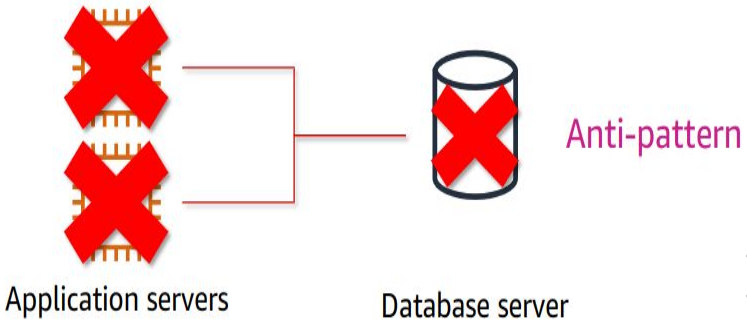
- Read and write needs
- Total storage requirements
- Durability requirements
- Latency requirements
- Maximum concurrent users to support
- Nature of queries
- ...

- It is important that you choose the right database solution.
- In traditional data centers and on-premises environments, limits on available hardware and licenses can constrain your choice of a data store solution.

7) Avoid Single Points of Failure

- Assume everything fails. Then, design backward

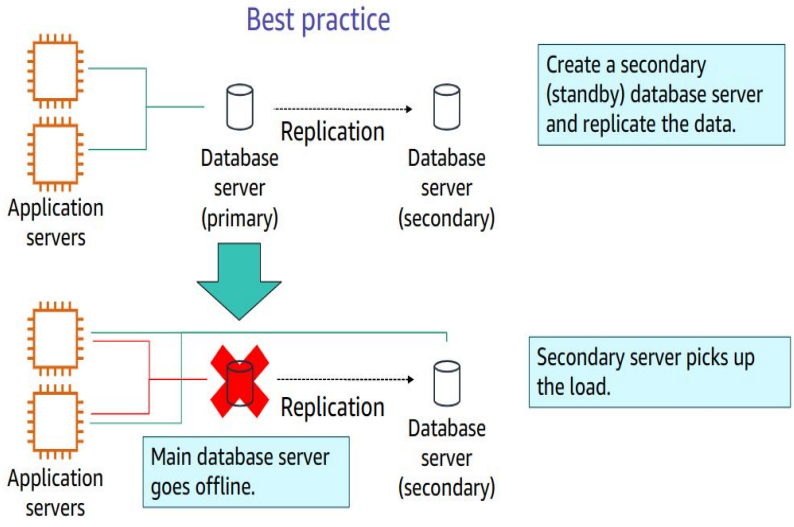
- Where possible,
  - use redundancy to prevent single points from bringing down an entire system



Application servers

Database server

7) Avoid Single Points of Failure



8) Optimize for Cost

What to consider to optimize the cost of cloud resources

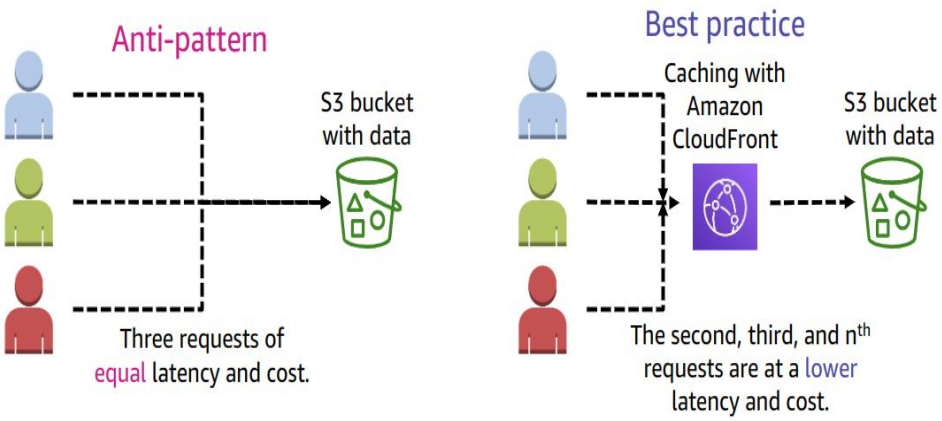
- Are my resources the right size and type for the job?
- What metrics should I monitor?
- How do I make sure to turn off resources that are not in use?
- Can I replace any of my servers with managed services

- Take advantage of cloud flexibility to increase your cost efficiency

9) Use Caching

Caching

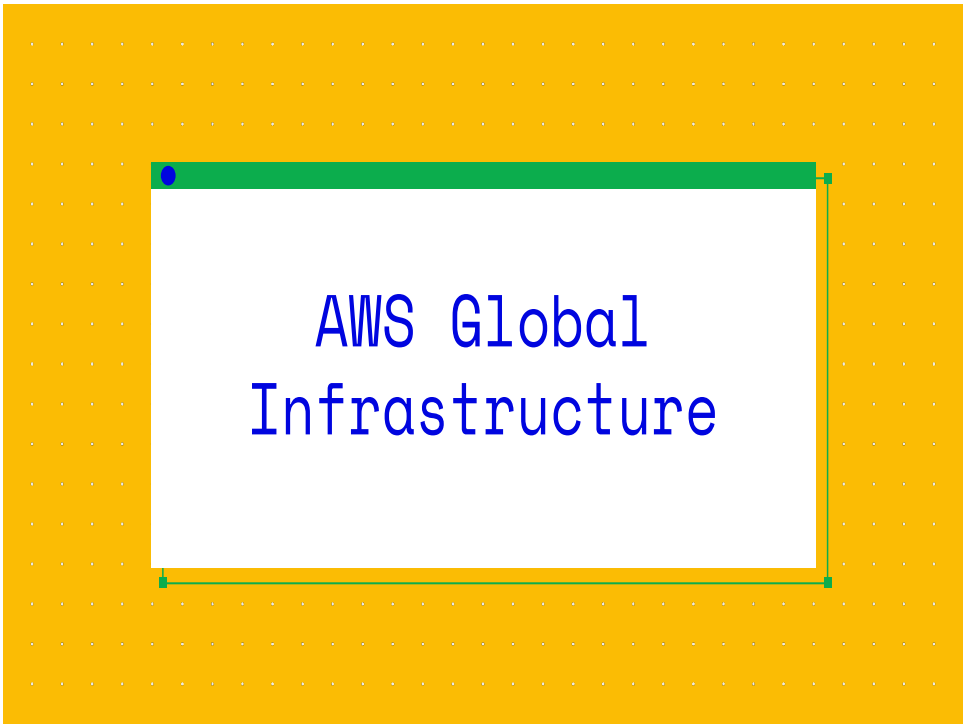
- minimizes redundant data retrieval operations
- improves performance and cost



10) Secure Your Entire Infrastructure

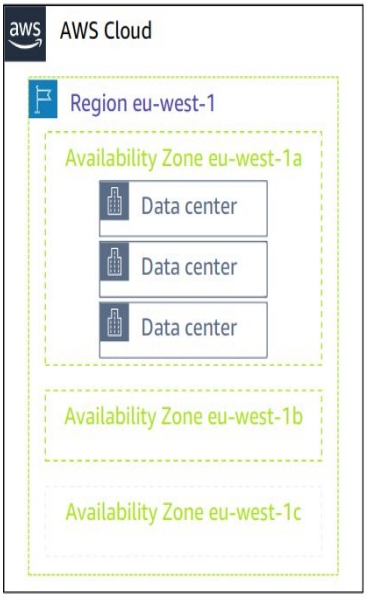
Build security into every layer of your infrastructure

- Isolate parts of your infrastructure
- **Encrypt data** in transit and at rest
- Enforce access control granularly, using the **principle of least privilege**
- Use multi-factor authentication (**MFA**)
- Use **managed services**
- **Log** access of resources
- **Automate** your deployments to keep security consisten


AWS Global
Infrastructure

AWS Availability Zones (AWS AZs)

- Each Availability Zone
  - Made up of one or more data centers
  - Designed for fault isolation
  - Interconnected with other AZs in a Region
    - using high-speed private links
- You can choose your AZs
- **Recommendation** replicate across AZs for resiliency

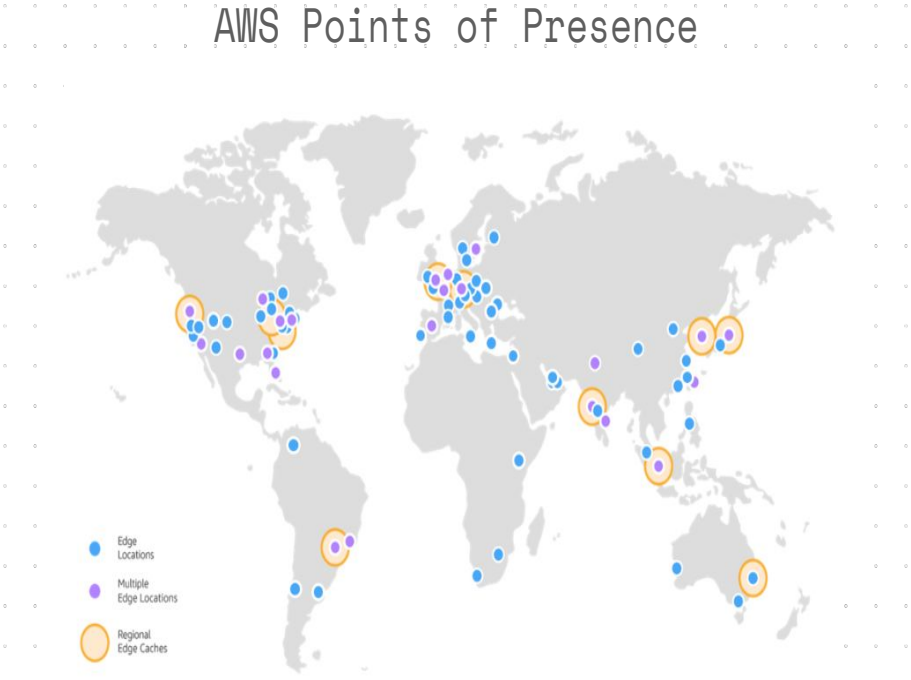


- **Resiliency**
  - The capacity to withstand or to recover quickly from difficulties
  - The ability of a substance or object to spring back into shape
- **Toughness**
  - the remarkable resilience of so many institutions

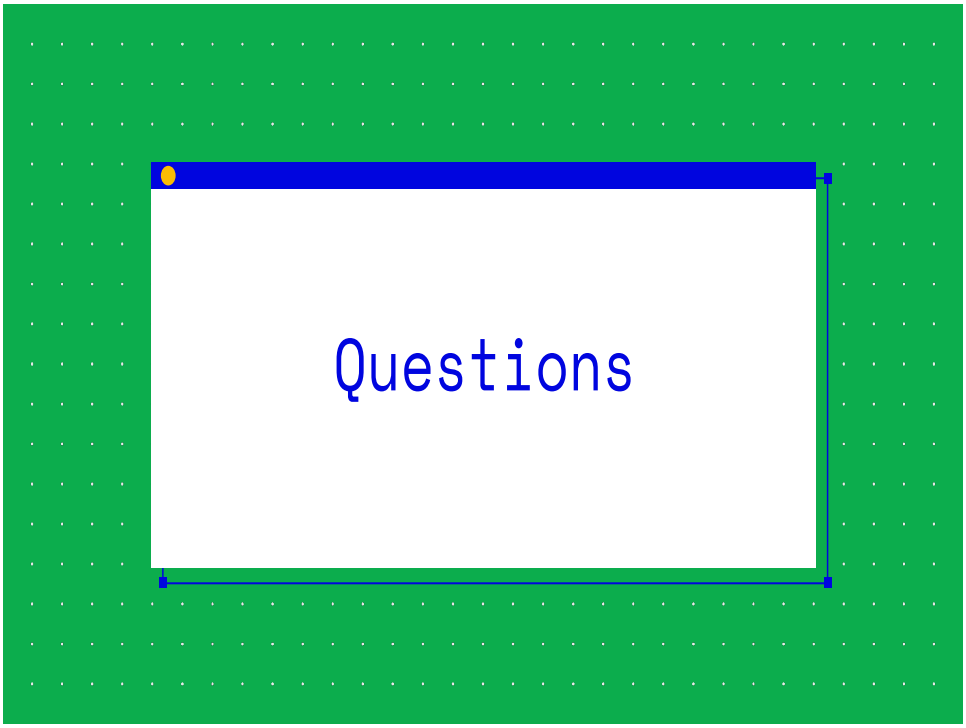
- **Resiliency in cloud computing**
  - Your organization's ability to handle failure while remaining functional
  - The goal is not to avoid failure but to accept it as inevitable and construct your cloud services to respond to it

AWS Data Centers

- **Data centers** are where the data resides and data processing occurs
- A **data center** typically has tens of thousands of servers
- All **data centers** are online and serving customers
- Usually Cloud Service Providers (CSPs)
  - use custom network equipment
  - has a customized network protocol stack



- To deliver content to end users with lower latency, Amazon **CloudFront** uses a global network that includes over **200 Points of Presence** that are comprised of
    - **Edge Locations**
        - located in North America, Europe, Asia, Australia, South America, the Middle East, Africa, and China
        - support AWS services like Amazon **Route 53** and **Amazon CloudFront**
    - **Regional Edge Caches**
        - are used by default with Amazon **CloudFront**
        - used when you have content that is not accessed frequently enough to remain in an **Edge Location**
        - absorb the content and provide an alternative to fetching the content from the origin server


Questions

Links

https://github.com/FCAI-B/cloud