



CSCI 5902 Adv. Cloud Architecting
Fall 2023
Instructor: Lu Yang

Module 2 Introducing Cloud Architecting
Sep 15, 2023

Housekeeping and feedback



1. First tutorial today. The topic is Introduction to Azure.
2. Difference of AWS Academy Cloud Architecting course
and AWS Academy Learner Lab.

challenge labs

AWS Academy Cloud Architecting

Module 2: Introducing Cloud Architecting

Module overview



Sections

1. What is cloud architecting?
2. The Amazon Web Services (AWS) Well-Architected Framework
3. Best practices for building solutions on AWS
4. AWS global infrastructure

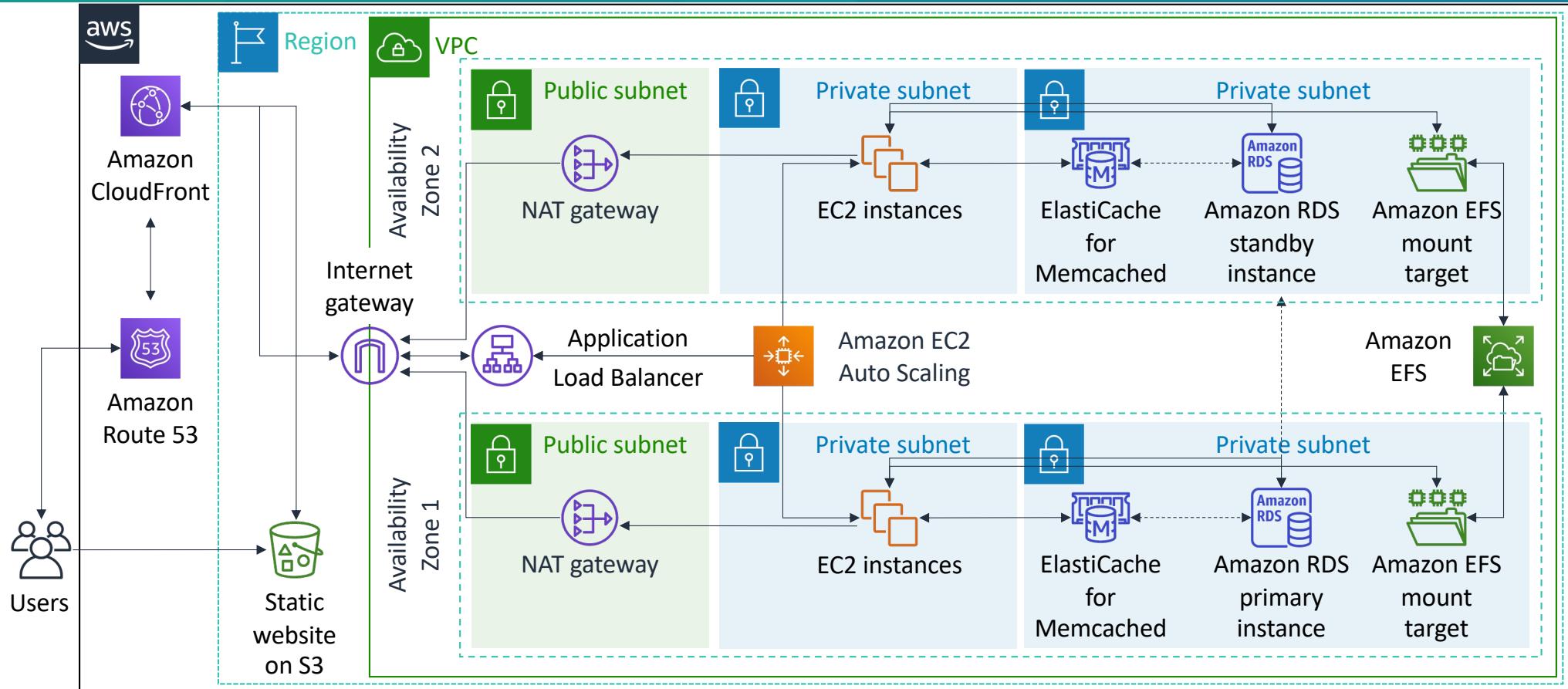
Module objectives



At the end of this module, you should be able to:

- Define cloud architecture
- Describe how to design and evaluate architectures using the AWS Well-Architected Framework
- Explain best practices for building solutions on AWS
- Describe how to make informed decisions on where to place AWS resources

A large architecture



Module 2: Introducing Cloud Architecting

Section 1: What is cloud architecting?

Architectural need



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

Origins of AWS



- According to AWS CEO Andy Jassy, at the time, Amazon ecommerce tools were “a jumbled mess”
 - Applications and architectures were built without proper planning
 - It 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

Problems persisted

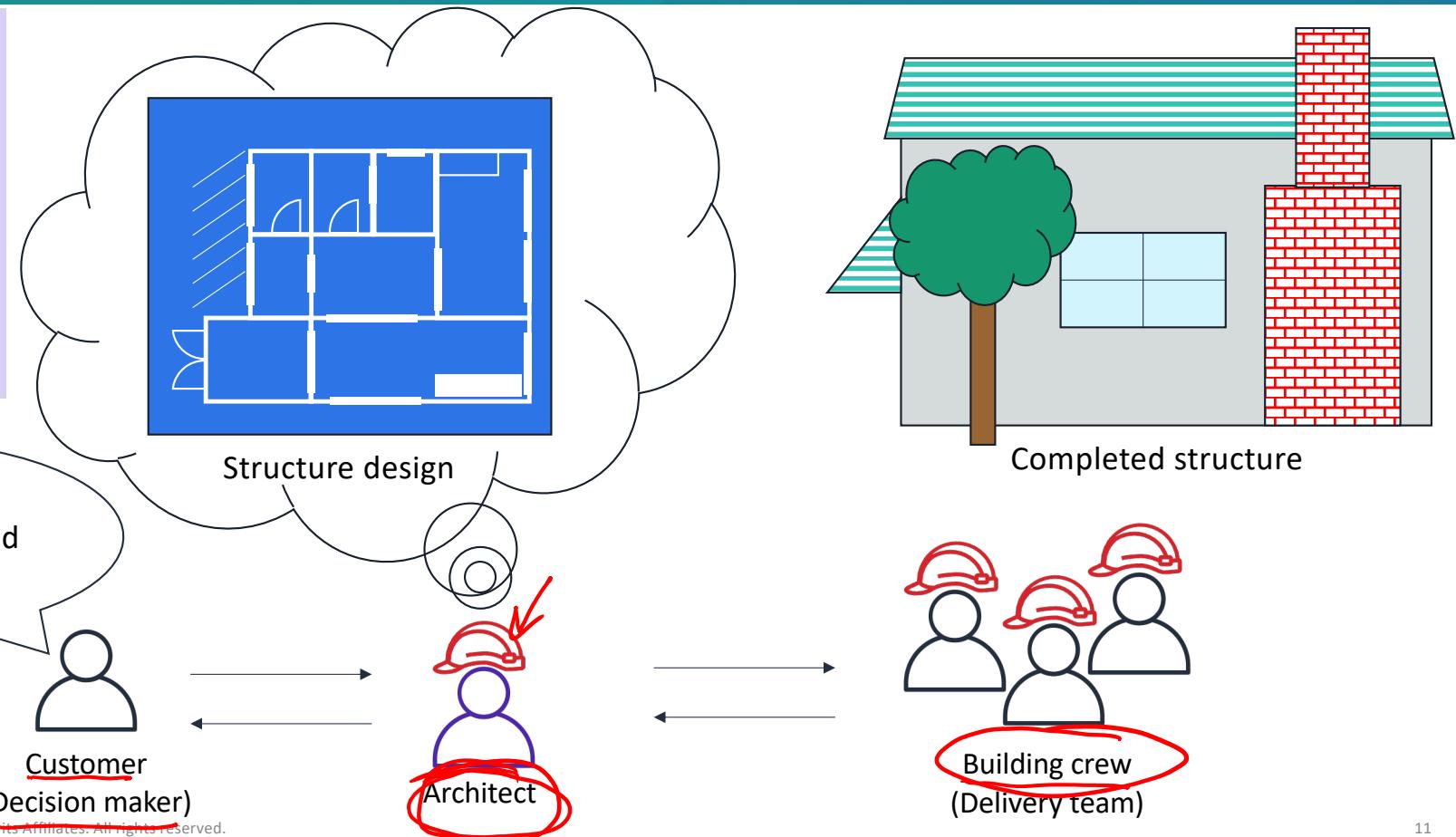


- Amazon still struggled to build applications quickly.
 - Database, compute, and storage components took 3 months to build.
 - Each team built their own resources, with no planning for scalability or re-usability.
- Solution: Amazon built internal services to create highly available, scalable, and reliable architectures on top of its infrastructure. In 2006, Amazon started selling these services as AWS.

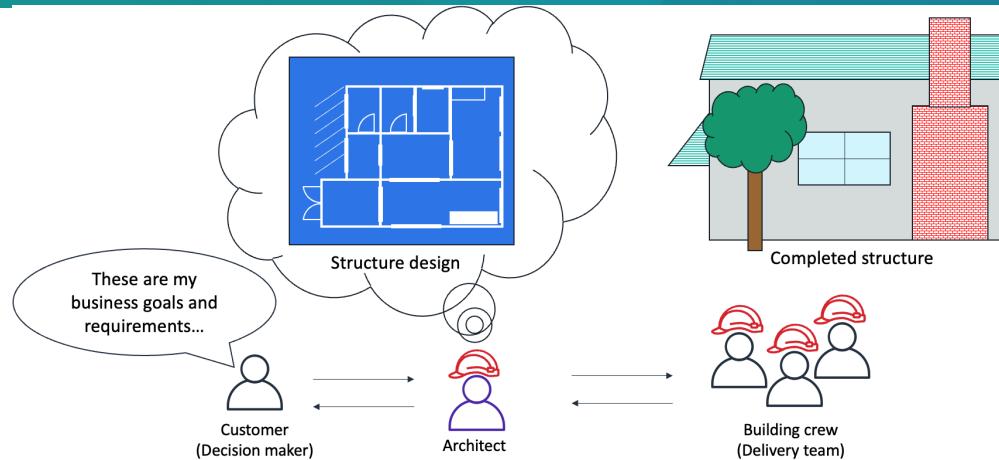
Cloud architecture



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.



Cloud architecture



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.

Section 1 key takeaways



13

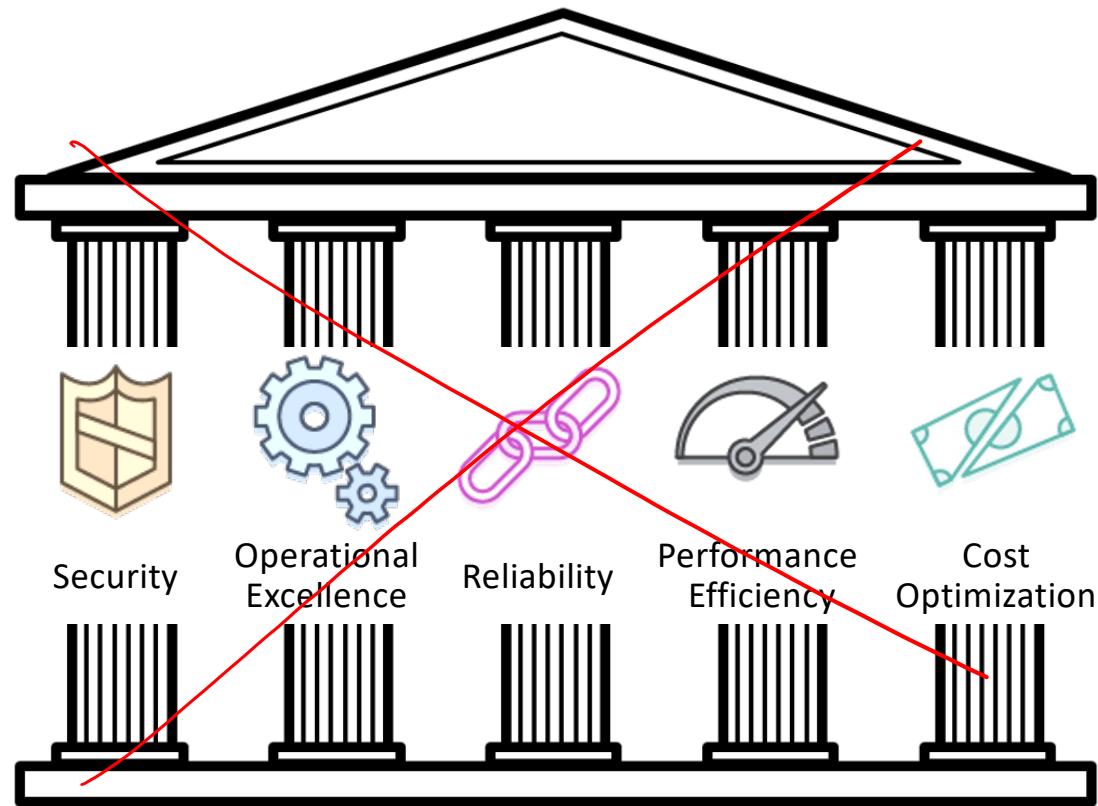


- Cloud architecture is the practice of applying cloud characteristics to a solution that uses cloud services and tools to meet an organization's technical needs and business use cases
- You can use AWS services to create highly available, scalable, and reliable architectures

Module 2: Introducing Cloud Architecture

Section 2: The AWS Well-Architected Framework

The AWS Well-Architected Framework



The AWS Well-Architected Framework



6 pillars of the framework



Figure from: <https://www.romexsoft.com/blog/six-pillars-of-the-aws-well-architected-framework-the-impact-of-its-usage-when-building-saas-application/>

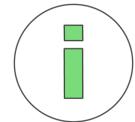
AWS Official Docs of Well-Architected Framework:

<https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html>

Security pillar



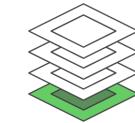
- The ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.
- There are six best practice areas and tools for security in the cloud:
 - **Security** – AWS Shared Responsibility Model, AWS Config, AWS Trusted Advisor
 - **Identity and Access Management** – IAM, Multi-Factor Authentication, AWS Organizations
 - **Detective Controls** – AWS CloudTrail, AWS Config, Amazon GuardDuty
 - **Infrastructure Protection** – Amazon VPC, Amazon CloudFront with AWS Shield, AWS WAF
 - **Data Protection** – ELB, Amazon Elastic Block Store (Amazon EBS), Amazon S3, and Amazon Relational Database Service (Amazon RDS) encryption, Amazon Macie, AWS Key Management Service (AWS KMS)
 - **Incident Response** – IAM, Amazon CloudWatch Events
- Key AWS service:
 - **AWS Identity and Access Management (IAM)**



Identity foundation



Traceability



Security at all layers



Risk assessment
and mitigation
strategies

Operational Excellence pillar



- The ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.
- There are four best practice areas and tools for operational excellence in the cloud:
 - **Organization** – AWS Cloud Compliance, AWS Trusted Advisor, AWS Organizations
 - **Prepare** – AWS Config
 - **Operate** – Amazon CloudWatch
 - **Evolve** – Amazon Elasticsearch Service
- Key AWS service:
 - **AWS CloudFormation** for creating templates. (See AWS Management Tools Cheat Sheet)

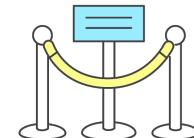
IaC



Deployed



Updated

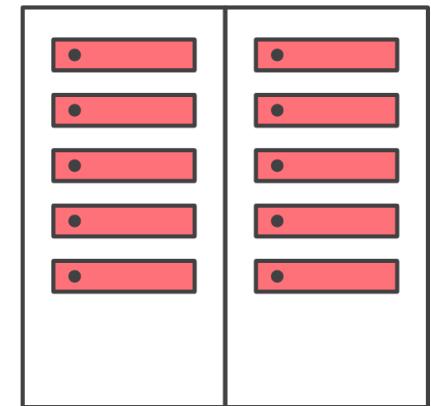


Operated

Reliability pillar



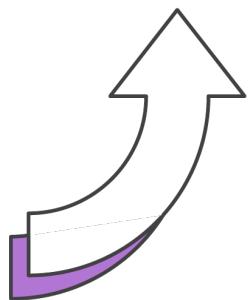
- Recover quickly from infrastructure or service disruptions ?
- Dynamically acquire computing resources to meet demand
- Mitigate disruptions such as: misconfigurations or transient network issues
- There are four best practice areas and tools for reliability in the cloud:
 - **Foundations** – IAM, Amazon VPC, AWS Trusted Advisor, AWS Shield
 - **Change Management** – AWS CloudTrail, AWS Config, Auto Scaling, Amazon CloudWatch
 - **Failure Management** – AWS CloudFormation, Amazon S3, AWS KMS, Amazon Glacier
 - **Workload Architecture** – AWS SDK, AWS Lambda
- Key AWS service:
 - Amazon CloudWatch



Performance Efficiency pillar



- Use computing resources efficiently to meet system requirements
- Maintain their efficiency as demand changes as technologies evolve
- There are four best practice areas for performance efficiency in the cloud:
 - **Selection** – Auto Scaling for Compute, Amazon EBS and S3 for Storage, Amazon RDS and DynamoDB for Database, Route53, VPC, and AWS Direct Connect for Network
 - **Review** – AWS Blog and What's New section of the website
 - **Monitoring** – Amazon CloudWatch
 - **Tradeoffs** – Amazon Elasticache, Amazon CloudFront, AWS Snowball, Amazon RDS read replicas.
- Key AWS service:
 - **Amazon CloudWatch**



Cost Optimization pillar



- Measure efficiency
- Eliminate unneeded expense 50%
- Consider using managed services
- There are five best practice areas and tools for cost optimization in the cloud:
 - **Cloud Financial Management** – Amazon QuickSight, AWS Cost and Usage Report (CUR)
 - **Cost-Effective Resources** – Cost Explorer, Amazon CloudWatch and Trusted Advisor, Amazon Aurora for RDS, AWS Direct Connect with Amazon CloudFront
 - **Matching supply and demand** – Auto Scaling
 - **Expenditure Awareness** – AWS Cost Explorer, AWS Budgets
 - **Optimizing Over Time** – AWS News Blog and the What's New section on the AWS website, AWS Trusted Advisor
- Key AWS service:
 - Cost Explorer



Sustainability pillar



- The ability to increase efficiency across all components of a workload by maximizing the benefits from the provisioned resources.
- There are six best practice areas for sustainability in the cloud:
 - **Region Selection** – AWS Global Infrastructure
 - **User Behavior Patterns** – Auto Scaling, Elastic Load Balancing
 - **Software and Architecture Patterns** – AWS Design Principles
 - **Data Patterns** – Amazon EBS, Amazon EFS, Amazon FSx, Amazon S3
 - **Hardware Patterns** – Amazon EC2, AWS Elastic Beanstalk
 - **Development and Deployment Process** – AWS CloudFormation
- Key AWS service:
 - **Amazon EC2 Auto Scaling**

The AWS Well-Architected Tool



AWS
Well-Architected
Tool

- Helps you review the state of your workloads and compares them to the latest AWS architectural best practices
- Gives you access to knowledge and best practices used by AWS architects, when you need it
- Delivers an action plan with step-by-step guidance on how to build better workloads for the cloud
- Provides a consistent process for you to review and measure your cloud architectures

<https://aws.amazon.com/well-architected-tool/>

SAA Exam Content Outline



Domain	% of Exam
Domain 1: Design Secure Architectures	30%
Domain 2: Design Resilient Architectures	26%
Domain 3: Design High-Performing Architectures	24%
Domain 4: Design Cost-Optimized Architectures	20%
TOTAL	100%

https://d1.awsstatic.com/training-and-certification/docs-sa-assoc/AWS-Certified-Solutions-Architect-Associate_Exam-Guide.pdf

Section 2 key takeaways



- The AWS Well-Architected Framework provides a consistent approach to evaluate cloud architectures and guidance to help implement designs
- The AWS Well-Architected Framework is organized into ~~six pillars~~
- Each pillar documents a set of foundational questions that enable you to understand if a specific architecture aligns well with cloud best practices
- The AWS Well-Architected Tool helps you review the state of your workloads and compares them to the latest AWS architectural best practices

Module 2: Introducing Cloud Architecting

Section 3: Best practices for building solutions on AWS

Design tradeoffs



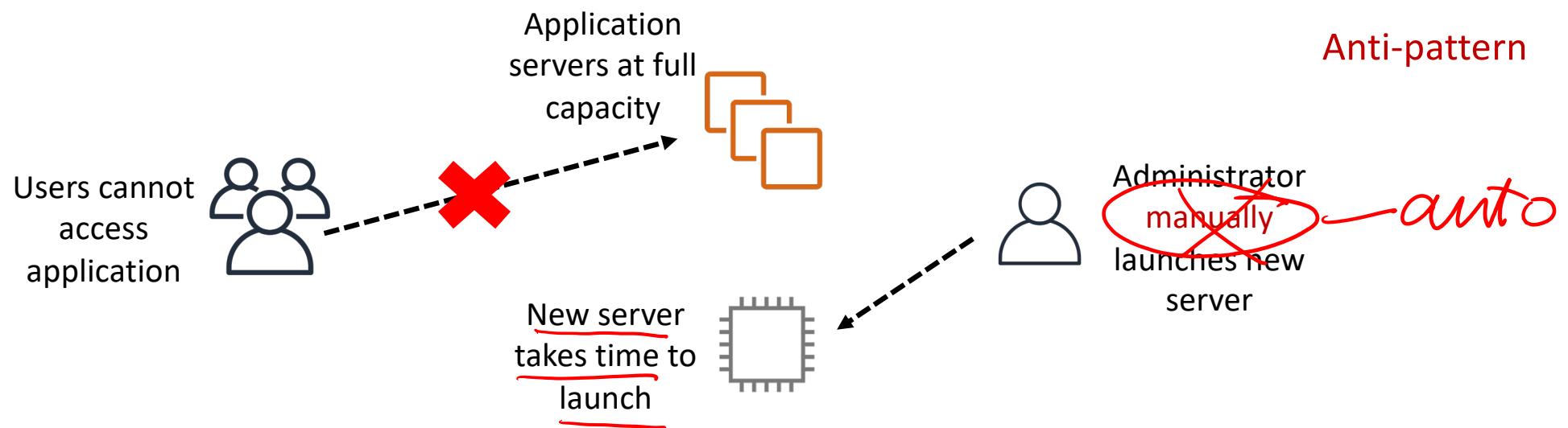
- Evaluate tradeoffs so you can select an optimal approach
- Examples of tradeoffs include:
 - Trade consistency, durability, and space for time and latency to deliver higher performance
 - Prioritize speed to market of new features over cost
- Base design decisions on empirical data



1. Enable scalability (1 of 2)



Ensure that your architecture can handle changes in demand.



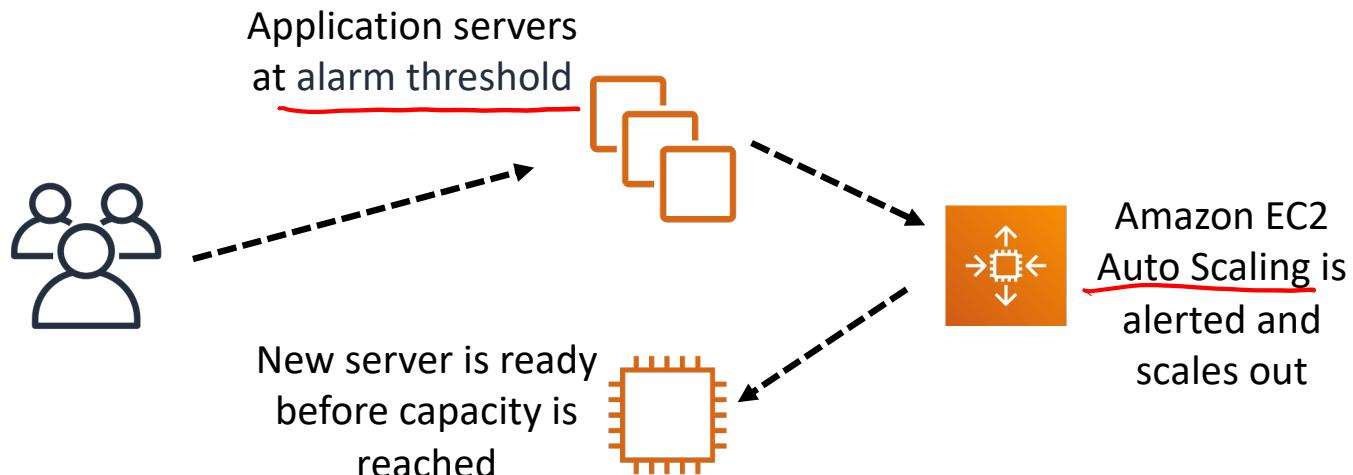
1. Enable scalability (2 of 2)



Ensure that your architecture can handle changes in demand.

Best practice

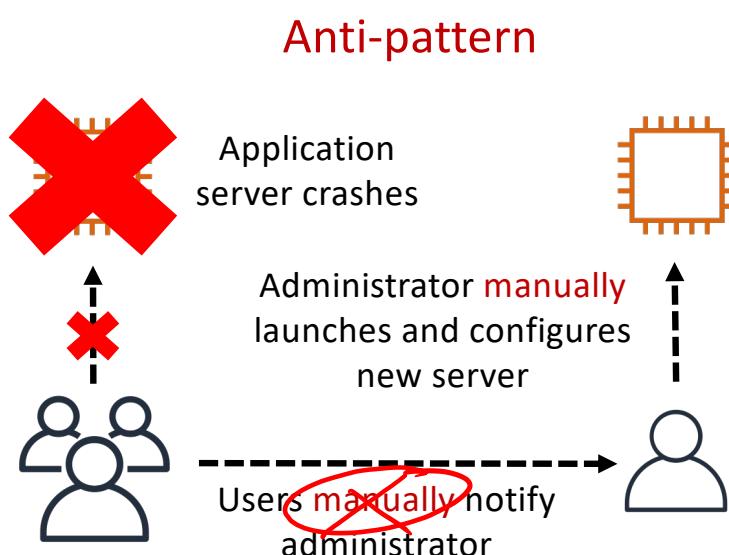
Users never experience a service interruption



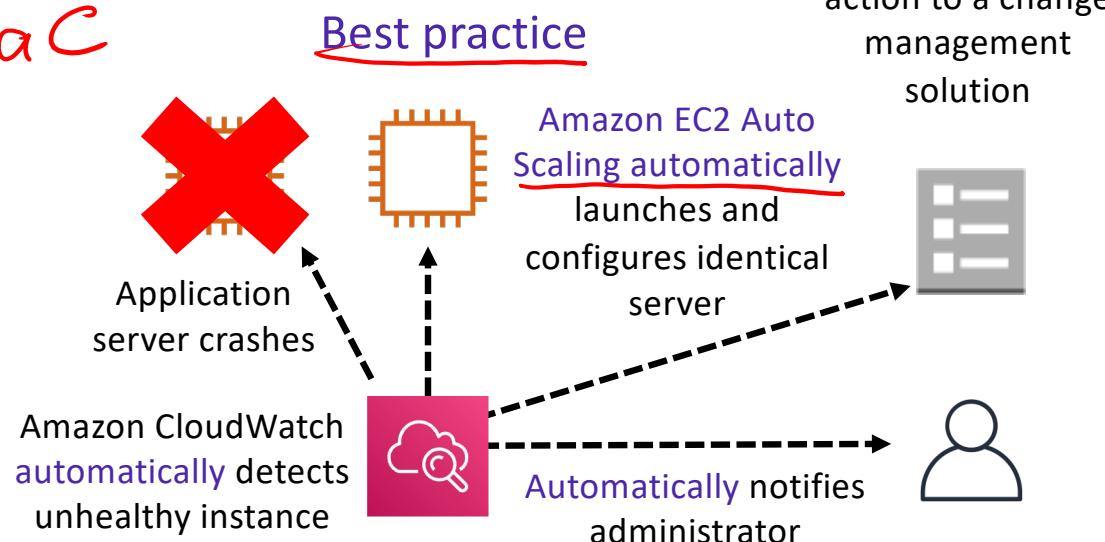
2. Automate your environment



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



IaC



3. Treat resources as disposable



Take advantage of the dynamically provisioned nature of cloud computing.

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

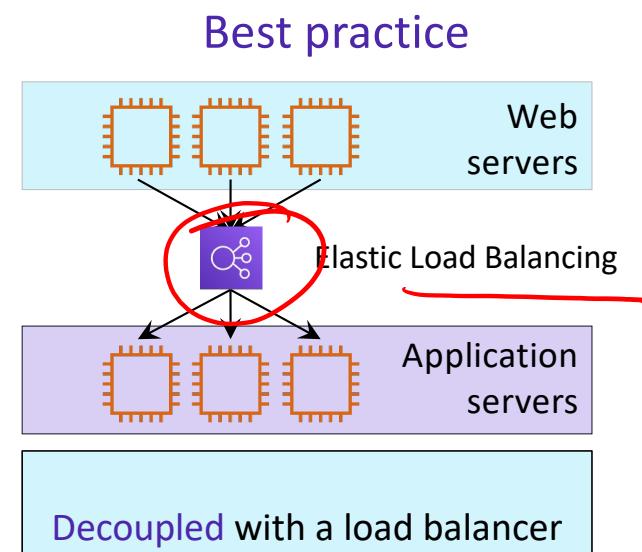
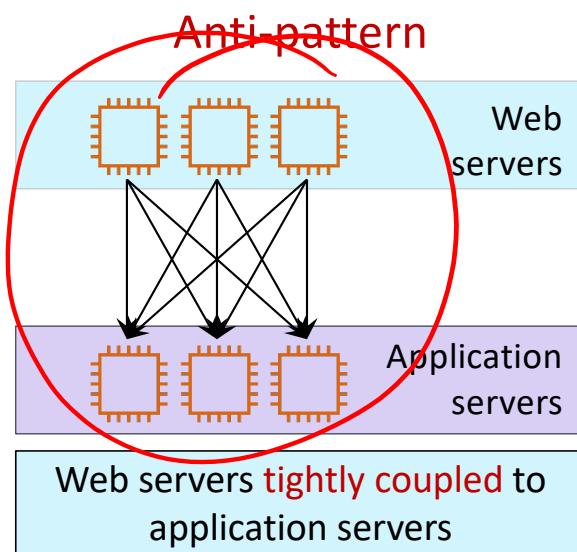
Best practice

- 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

4. Use loosely coupled components



Design architectures with independent components.



5. Design services, not servers



*Use the breadth of AWS 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



*Match technology to the workload, not
the other way around.*

Things to consider:

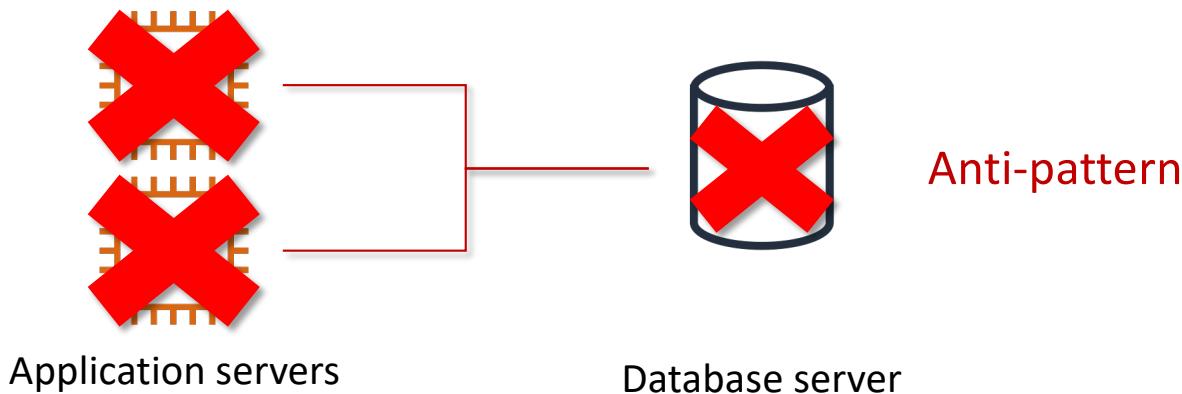
- Read and write needs
- Total storage requirements
- Typical object size and nature of access to these objects
- Durability requirements
- Latency requirements
- Maximum concurrent users to support
- Nature of queries
- Required strength of integrity controls

7. Avoid single points of failure (1 of 2)

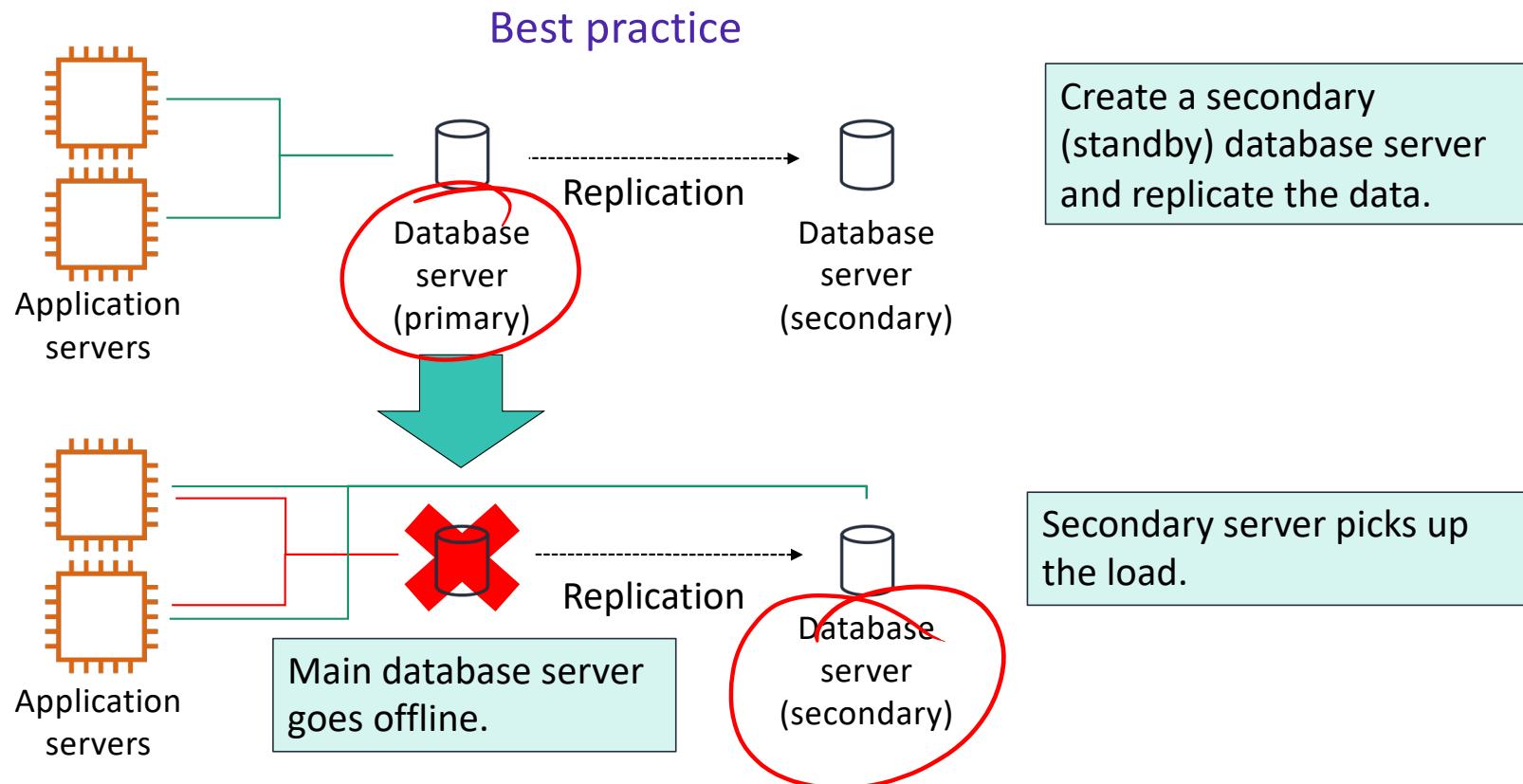


*Assume everything fails.
Then, design backward.*

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



7. Avoid single points of failure (2 of 2)



8. Optimize for cost



Take advantage of the flexibility of AWS to increase your cost efficiency.

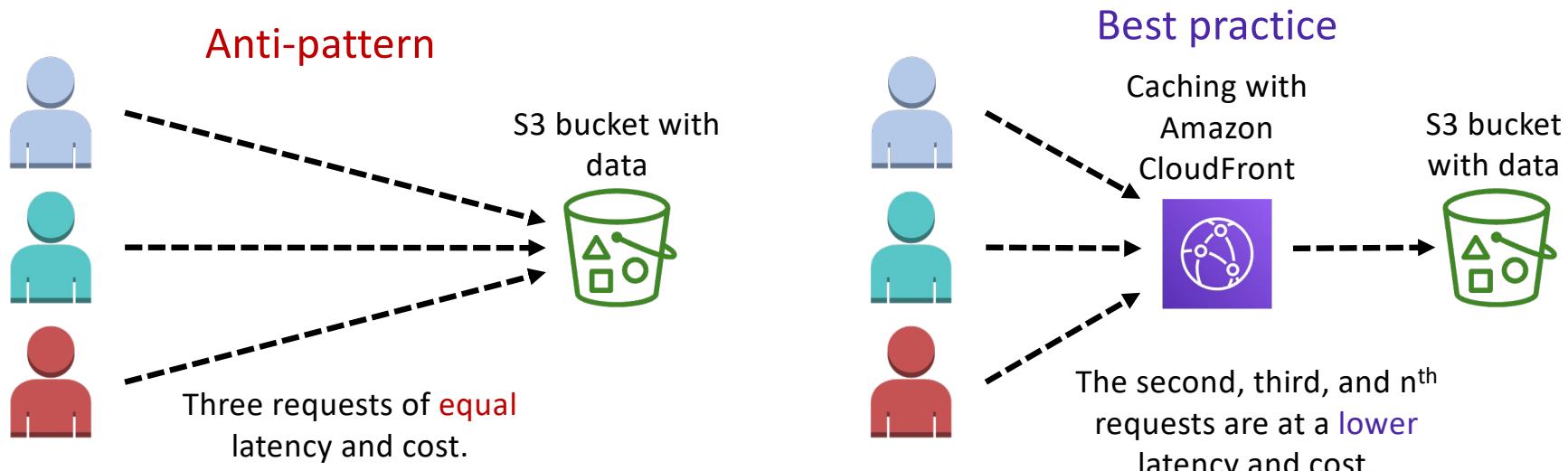
Things to consider:

- 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?
- How often will I need to use this resource?
- Can I replace any of my servers with managed services?

9. Use caching



Caching minimizes redundant data retrieval operations, improving performance and cost.



10. Secure your entire infrastructure



Build security into every layer of your infrastructure.

Things to consider:

- 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 consistent

Section 3 key takeaways



40



- As you design solutions, evaluate tradeoffs and base your decisions on empirical data
- Follow these best practices when building solutions on AWS –
 - Enable scalability
 - Automate your environment
 - Treat resources as disposable
 - Use loosely-coupled components
 - Design services, not servers
 - Choose the right database solution
 - Avoid single points of failure
 - Optimize for cost
 - Use caching
 - Secure your entire infrastructure

Module 2: Introducing Cloud Architecting

Section 4: AWS global infrastructure

AWS Regions



- An AWS Region is a geographical area
- Each AWS Region consists of two or more Availability Zones
- Communication between Regions uses AWS backbone network infrastructure
- You enable and control data replication across Regions
never

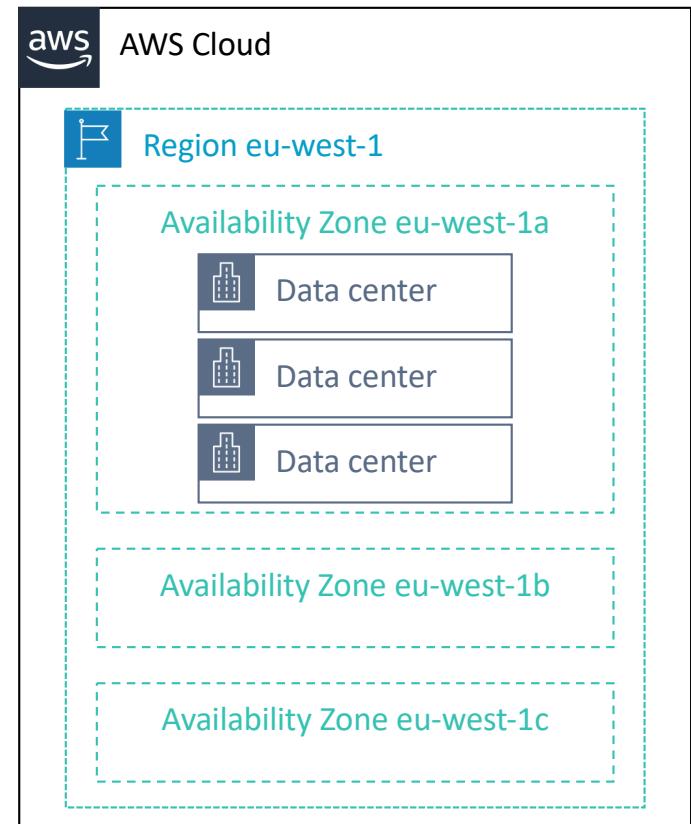


Example: London Region

AWS Availability Zones



- Each Availability Zone is –
 - Made up of one or more data centers
 - Designed for fault isolation
 - Interconnected with other Availability Zones in a Region using high-speed private links
- For certain services, you can choose your Availability Zones
- AWS recommends replicating across Availability Zones for resiliency



AWS Local Zones



- Enable you to run **latency-sensitive** portions of applications closer to end users and resources in a specific geography
- Are an extension of an AWS Region where you can use AWS services in **geographic proximity to end users**
- Let you place AWS compute, storage, database, and other select services closer to large population, industry, and IT centers where no Region exists today
- Are managed and supported by AWS
- **Los Angeles (LA) AWS Local Zone** is available by invitation

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
- AWS custom network equipment –
 - Is sourced from multiple ODMs
 - Has a customized network protocol stack



AWS Global Infrastructure



(32)

31 Launched Regions
each with multiple Availability Zones

(33)

35 Local Zones
29 Wavelength Zones
for ultralow latency applications

(102)

99 Availability Zones

550+

(450+) Points of Presence
400+ Edge Locations and 13 Regional Edge Caches

245 Countries and Territories Served

115

Direct Connect Locations

<https://aws.amazon.com/about-aws/global-infrastructure/>

AWS Global Infrastructure Map

The AWS Cloud spans 99 Availability Zones within 31 geographic regions around the world, with announced plans for 15 more Availability Zones and 5 more AWS Regions in Canada, Israel, Malaysia, New Zealand, and Thailand.



List view

Regions Coming soon

Section 4 key takeaways



48



- The AWS global infrastructure consists of **Regions**, **Availability Zones**, and **edge locations**
- Your choice of a Region is typically based on **compliance requirements** or to **reduce latency**
- Each **Availability Zone** is physically separate from other Availability Zones and has redundant power, networking, and connectivity
- Edge locations and **Regional edge caches** improve performance by caching content closer to users

Module 2: Introducing Cloud Architecting

Module wrap-up

Module summary



In summary, in this module, you learned how to:

- Define cloud architecture
- Describe how to design and evaluate architectures using the AWS Well-Architected Framework
- Explain best practices for building solutions on AWS
- Describe how to make informed decisions on where to place AWS resources