

Architecting for Reliability on AWS

ARCHITECTING FOR AVAILABILITY



Ben Piper

AUTHOR, AWS CERTIFIED SOLUTIONS ARCHITECT STUDY GUIDE

benpiper.com

Reliability or resiliency

The ability of an application to avoid and recover from failure

Availability

The percentage of time that an application is performing as expected

Poor performance implies low availability

Uptime isn't the same as availability

Decide how much availability
you need before designing
your AWS environment

Availability and Annual Downtime

99.0%

3 days
15 hours

99.9%

5 hours
45 minutes

99.99%

About 1 hour

99.999%

5 minutes

Architecting for 99.9% Availability

Traditional Application

Runs on Windows or Linux

You can “lift and shift” it to AWS without changing its code

- Example: SQL-backed web application

How you design for availability depends on the architecture of the application

Web-based Video Processing Application



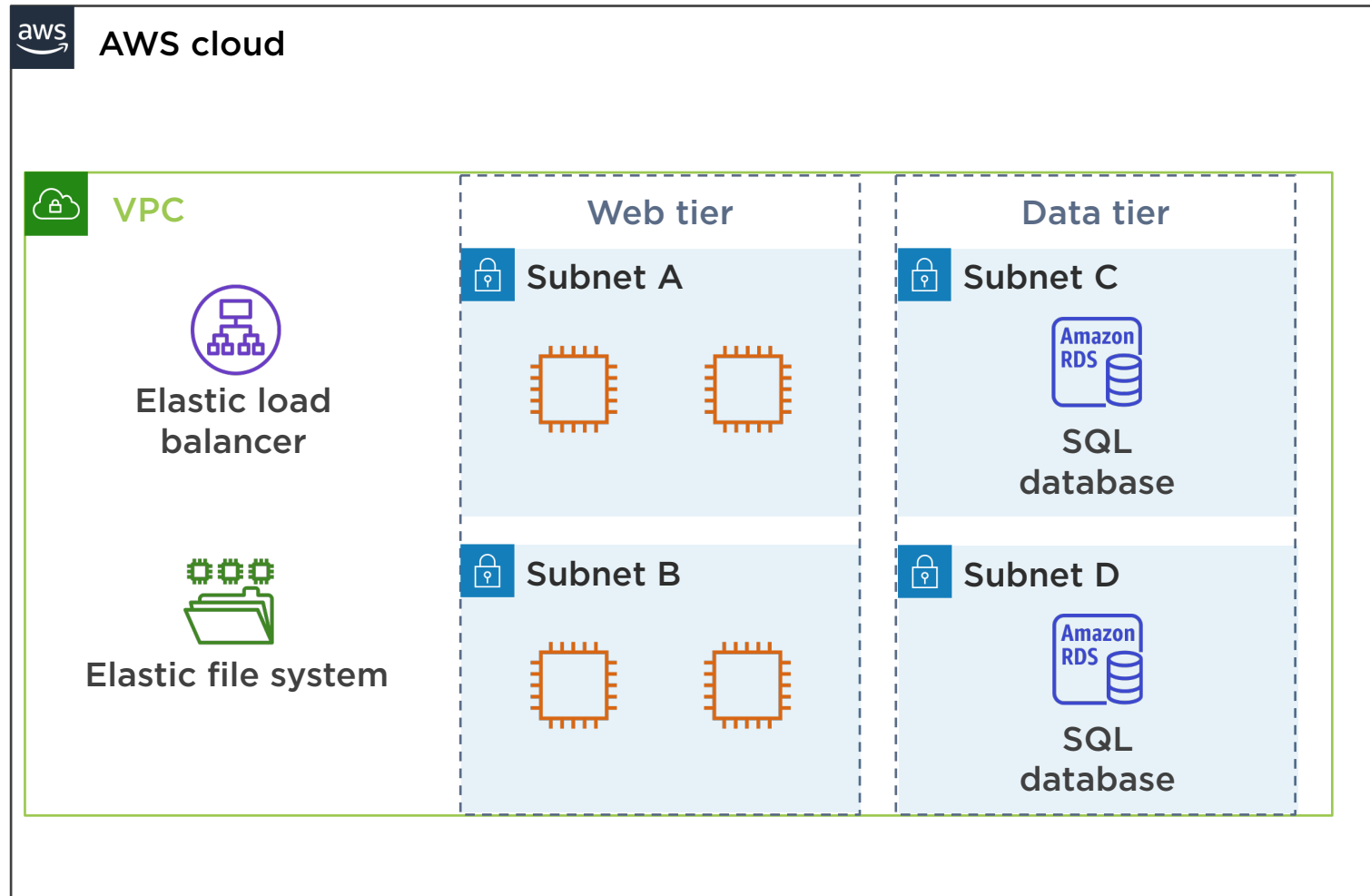
Front end

- Web interface
- Video processing logic

Back end

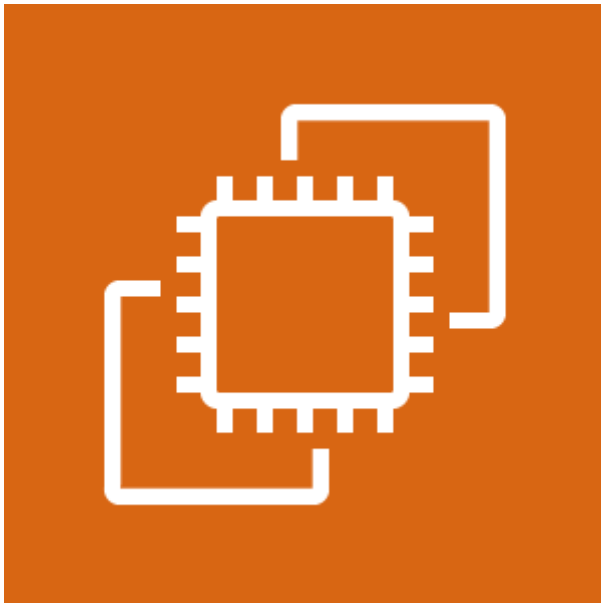
- SQL database

AWS Architecture




EC2 instance

Calculating Availability—EC2

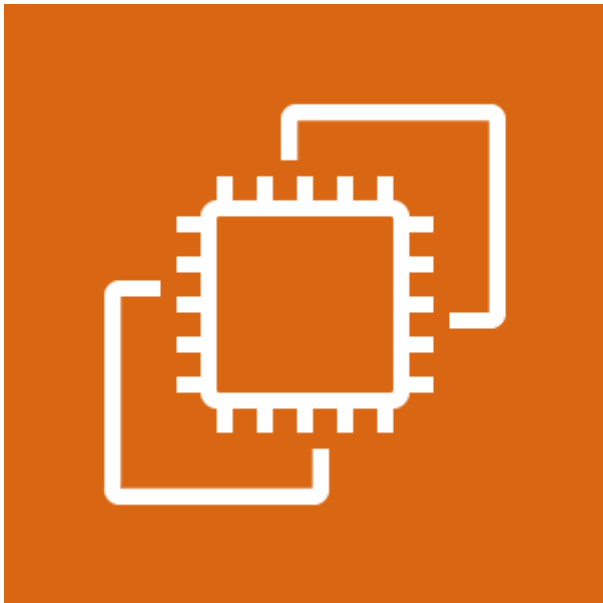


The service level agreement (SLA) for each service includes its annual availability

Availability of an EC2 instance is 90%

Failure rate of each instance is 10%

Calculating Availability—Redundant Instances



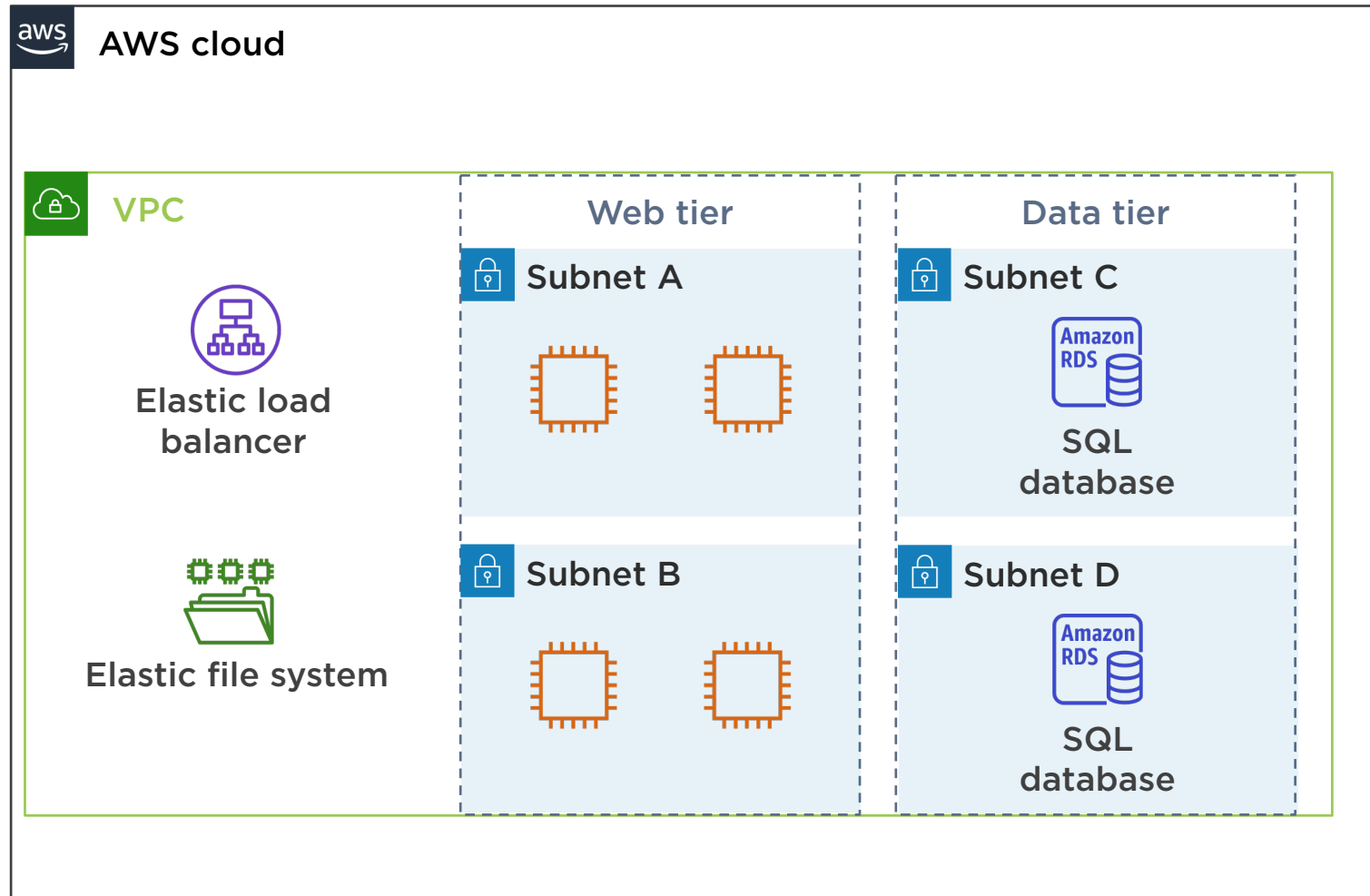
Multiply failure rate of four instances

$$- 10\% \times 10\% \times 10\% \times 10\% = 0.01\%$$

Subtract product from 100%

$$- 100\% - 0.01\% = 99.99\%$$

AWS Architecture




EC2 instance

Calculating Availability—Elastic Load Balancer

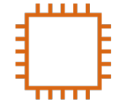
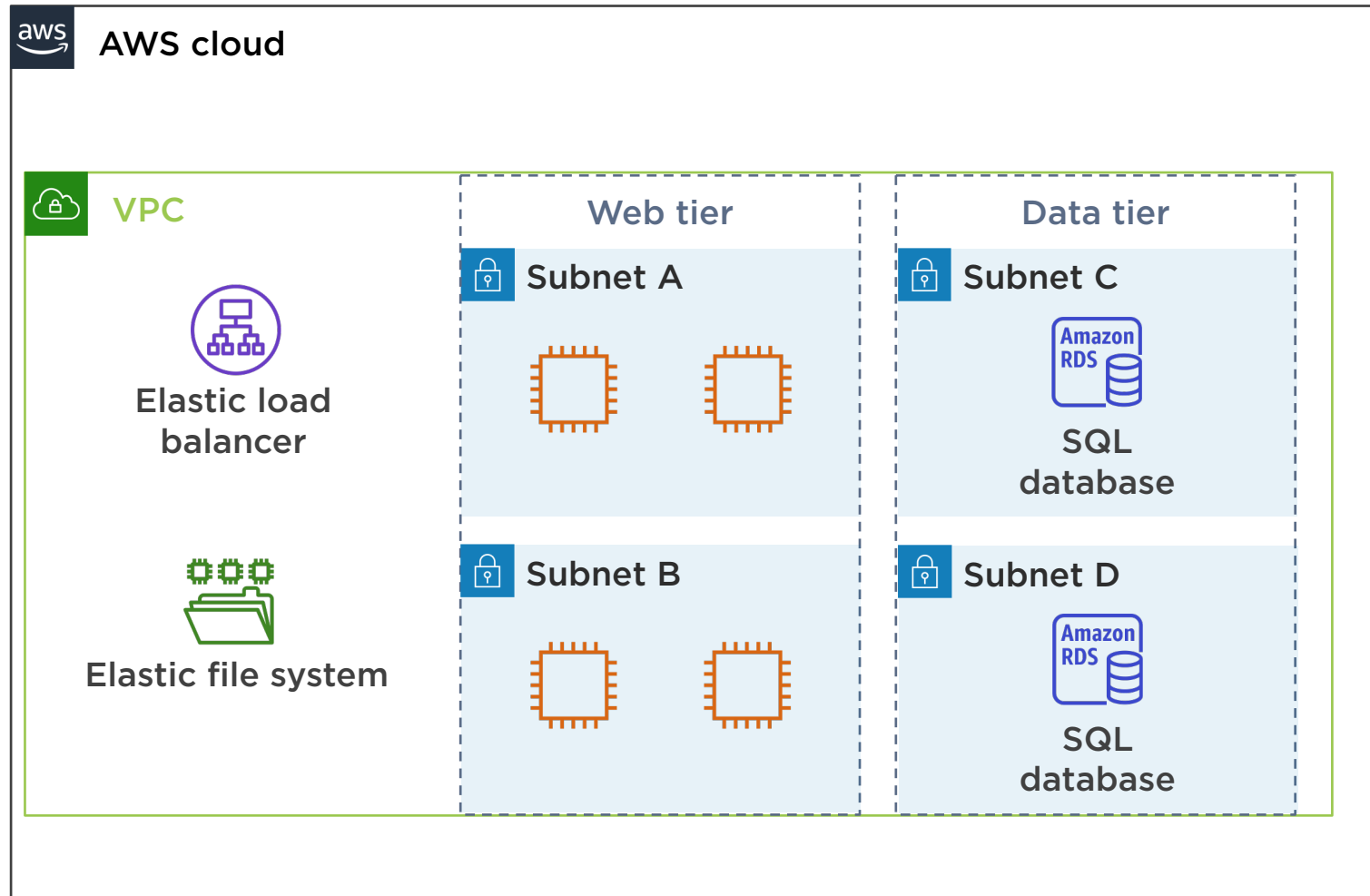


ELB availability is 99.99%

Multiply this by the availability of instances

$$- 99.99\% \times 99.99\% = 99.998\%$$

AWS Architecture



EC2 instance

Calculating Availability—Elastic File System



EFS availability is 99.99%

$$99.998\% \times 99.99\% = 99.97\%$$

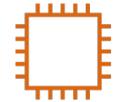
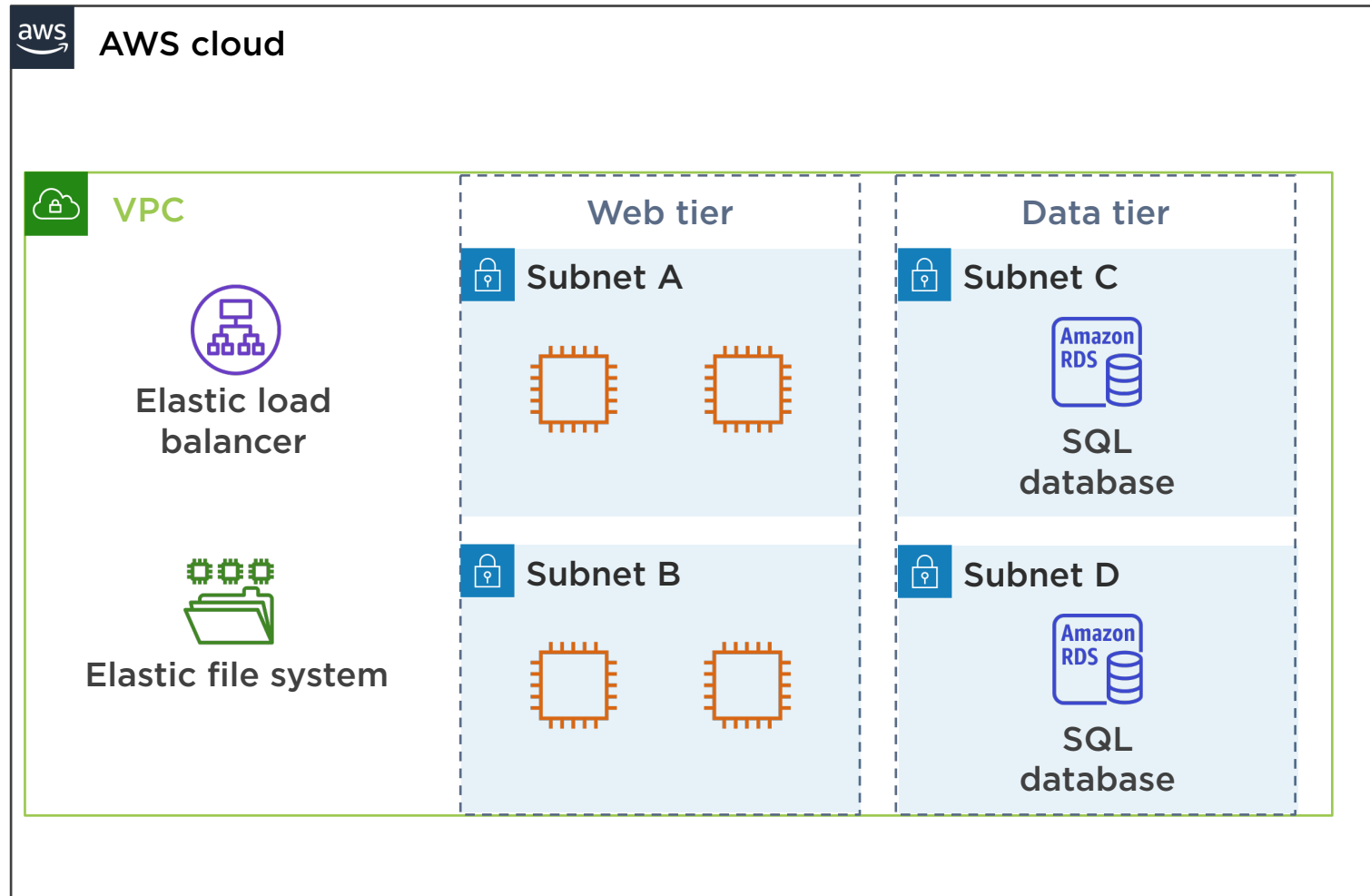
EFS or S3?

To use S3, application must be designed for it

EFS provides a network file system (NFS) volume

NFS is an established standard that most Linux distributions support

AWS Architecture



EC2 instance

Relational Database Service



RDS offers managed database engines

- MySQL, MariaDB, PostgreSQL, Microsoft SQL server, and Oracle

AWS manages database infrastructure and backups

Calculating Availability—RDS

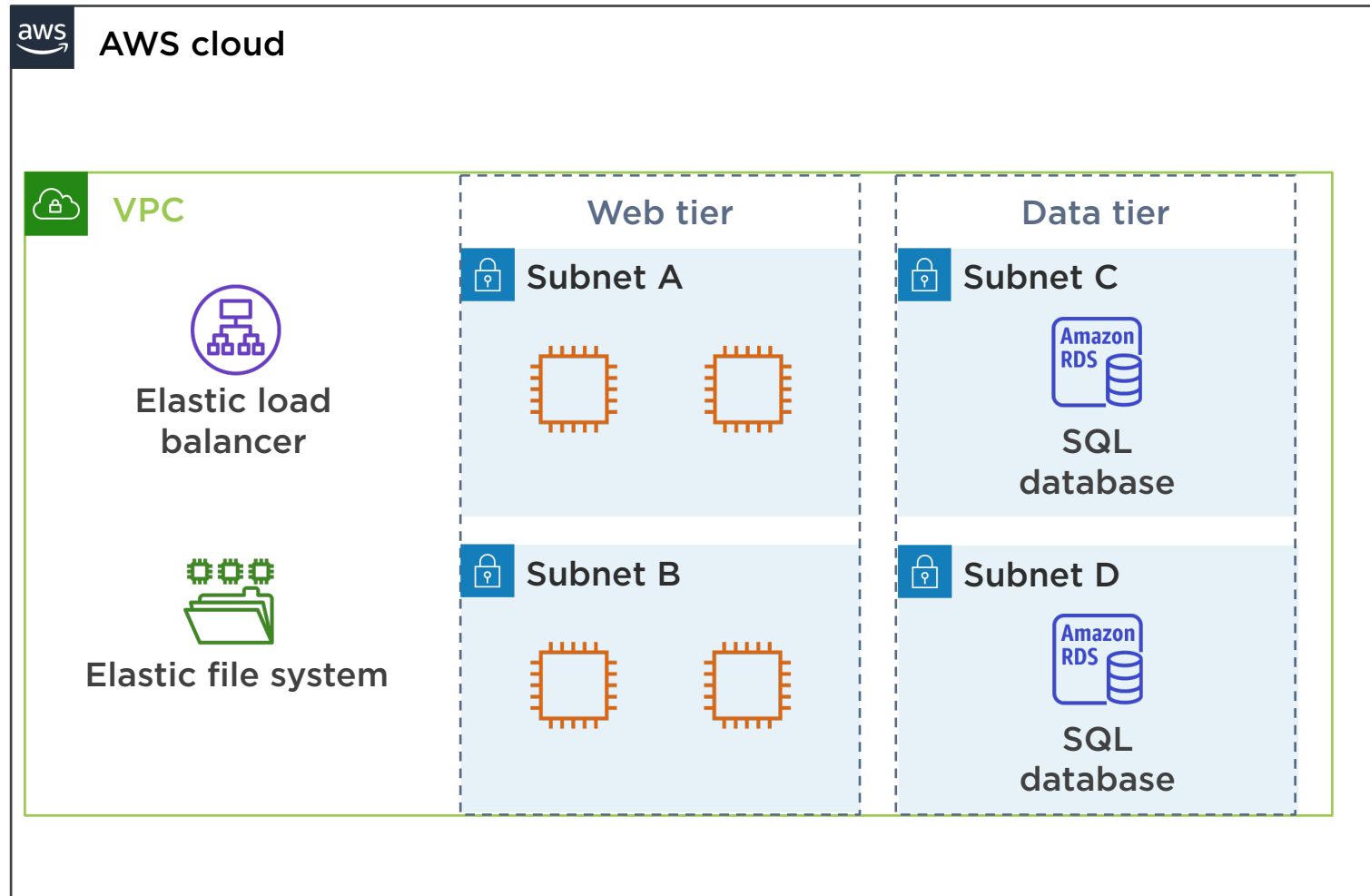


RDS multi-AZ availability is 99.95%

$$99.95\% \times 99.97\% = 99.92\%$$

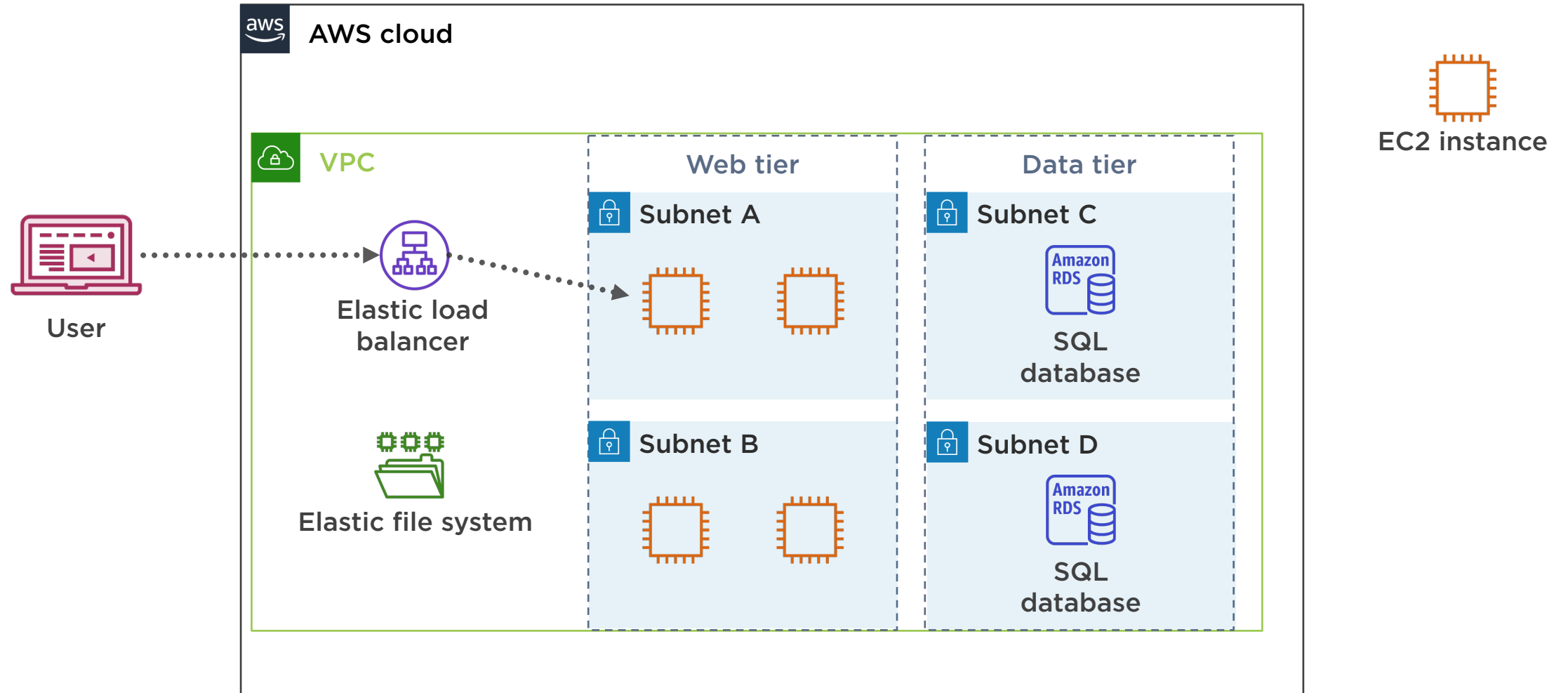
Loose Coupling

Loose Coupling

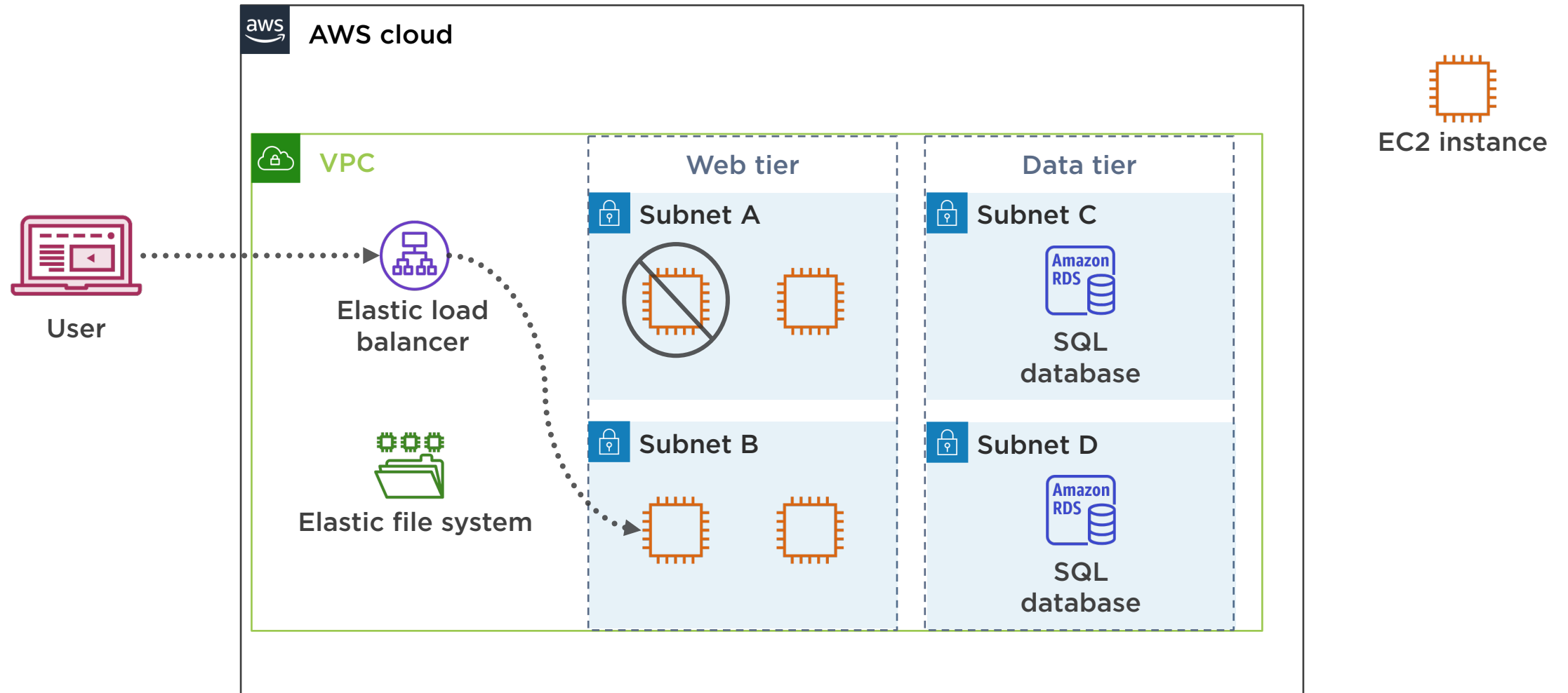



EC2 instance

Loose Coupling



Loose Coupling



Loose vs. Tight Coupling

Loose coupling

One component *doesn't* depend on a specific component

URL points to ELB

One-to-many relationship

Tight coupling

One component *does* depend on a specific component

URL points directly to instance

One-to-one relationship

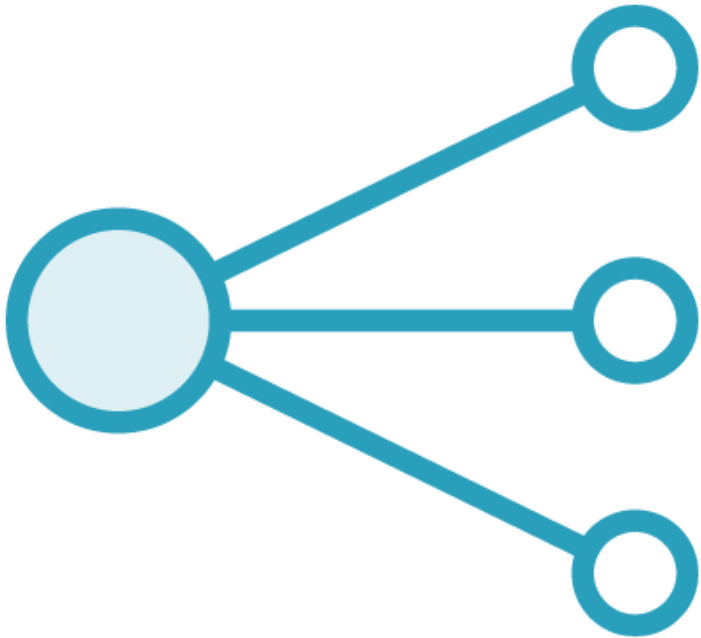
What About EFS?



Elastic services are always composed of redundant components

Elastic services are always loosely coupled with other resources

Redundancy and Loose Coupling



Redundancy → loose coupling → availability

Loose Coupling and Performance



Independently scale resources without impacting availability

Performance and availability are linked

- Poor performance leads to low availability
- Good performance helps maintain availability

Simple Queue Service

Monolithic Architecture

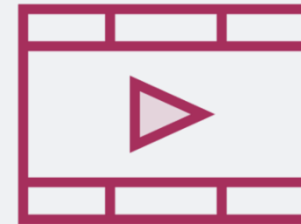


EC2 instance

Web service



Video processing
service



Microservices Architecture



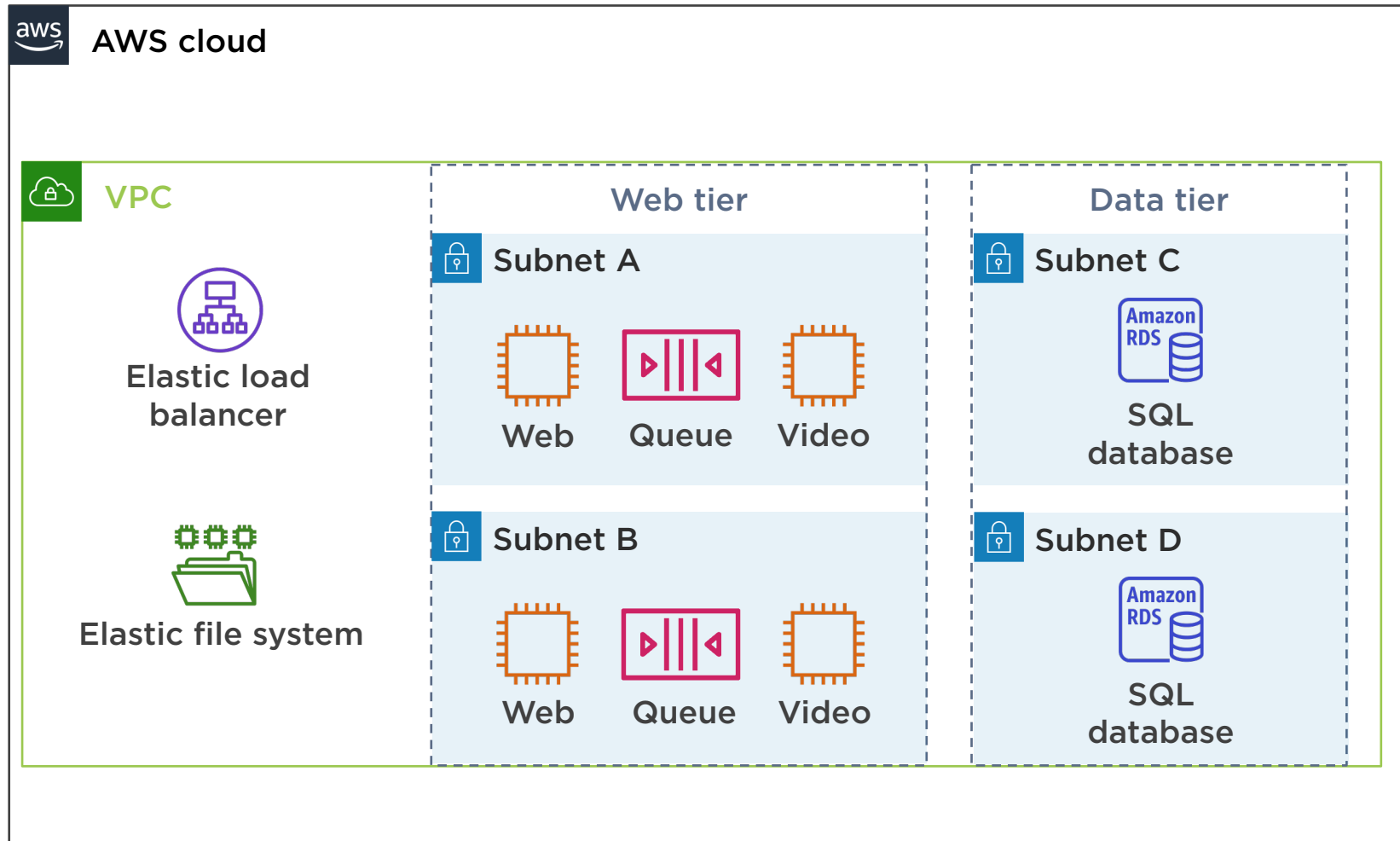
Benefits of Microservices

Easier updates

Availability

Scalability

Microservices Architecture



Elastic Container Service

Containers



Behave like virtual machines

Let you run multiple microservices on a single instance

Processes running inside the container are isolated from the host

Docker



Image contains everything the service needs

Launch containers from an image

Like launching EC2 instances from an AMI

Benefits of Containers—Ease of Deployment



Build an image

Deploy it to instances

Launch containers from it

Benefits of Containers—Availability



Having multiple containers on a single instance can give you redundancy

Elastic Container Service



ECS clusters contain EC2 instances

ECS can store containers in an elastic container registry (ECR)

Elastic Container Service



A task defines how to launch containers

- Image to use
- CPU
- Memory
- Instance-to-container port mappings
- Storage mappings

Cloud Native Applications

Cloud Native Applications



Depend on a cloud service that can't be deployed on-premises

- Examples: SQS, S3, DynamoDB

“Services instead of servers”

Availability not necessarily higher than traditional applications

Cloud Native Application Example



Lambda

Video processing



S3

Web asset and
video storage



DynamoDB

Database

Lambda



Managed “serverless” compute service

Supports many programming languages

- C#
- Java
- Python
- Go
- PowerShell

Availability is 99.95%

Simple Storage Service (S3)

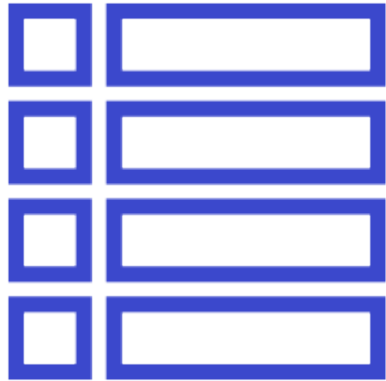


Unlimited file storage

Static web hosting

Availability is 99.9%

DynamoDB

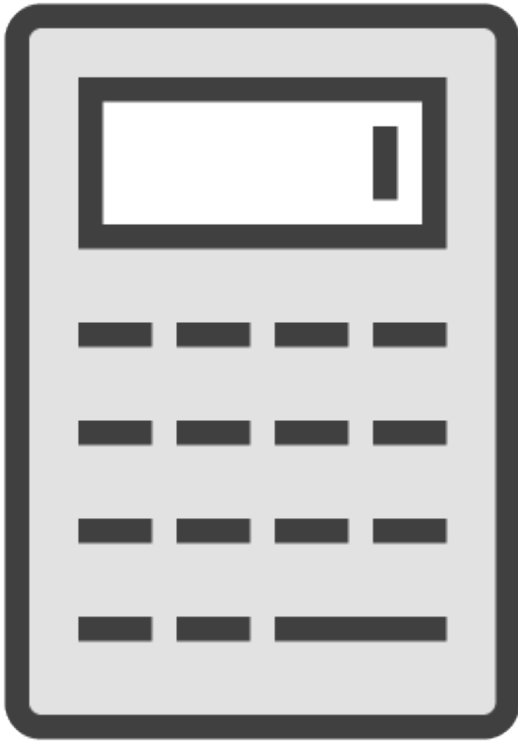


Managed, nonrelational database

Global tables feature replicates tables across multiple regions

Availability with global tables is 99.999%

Calculating Availability



Multiply the availability of services

$$- 99.95 \times 99.9 \times 99.999 = 99.85\%$$

Cloud Native Architecture

Advantages

Scalability

Performance

Convenience

Disadvantages

Cloud vendor lock-in

Slightly lower availability

Improve Availability by Using Multiple Regions



Calculate the failure rate of each region

- $100\% - 99.85\% = 0.15\%$

Multiply the failure rate of both regions

- $0.15\% \times 0.15\% = 0.0225\%$

Subtract the product from 100%

- $100\% - 0.0225\% \approx 99.999\%$

Trusted Advisor

Service Limits



You can't provision an unlimited number of AWS resources because of service limits

Check Trusted Advisor to find out your service limits

Demo



View Trusted Advisor to see how many EC2 instances we can launch per region

Key Takeaways

Summary



Availability is the percentage of time an application is performing as expected

Summary



**Decide your availability requirements
before beginning your design**

Summary



To achieve high availability, use

- Redundancy
- Loose coupling

Summary



To calculate the availability of redundant components

- Multiply the failure rate of the components
- Subtract that number from 100%

Summary



To calculate the availability of hard dependencies, multiply the availability of the components together

Summary



The Simple Queue Service (SQS) can act as a go-between for loosely coupled services

- Sending service places message in a queue
- Receiving service polls the queue for new messages

Useful for online voting applications

Summary



**The Elastic Container Service (ECS)
deploys microservices using
Docker containers**

**Can improve availability by running
multiple containers on a single instance**

Summary



Service limits constrain availability

Use Trusted Advisor to assess utilization against service limits

Coming up Next



Setting up your AWS environment