

# AWS State, Local, and Education Learning Days

Salt Lake City

11:30am – 12:30pm

**200**  
level

Cloud architectural patterns:

Master Cloud Architecture: Build Secure, Scalable Solutions with AWS Best Practices and Enterprise-Grade Design Strategies.



# Cloud architectural patterns:

## Platform and application best practices

### Ash Naik

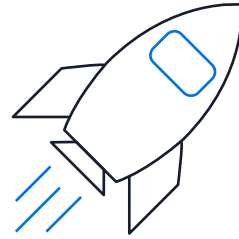
Solutions Architect  
Amazon Web Services  
ashjnaik@amazon.com

### Ryan Bachand

Solutions Architect  
Amazon Web Services  
bacryan@amazon.com



# Why Cloud Architecture Matters



Build and deploy faster

---



Lower or mitigate risks

---



Make informed decisions

---



**Also -**

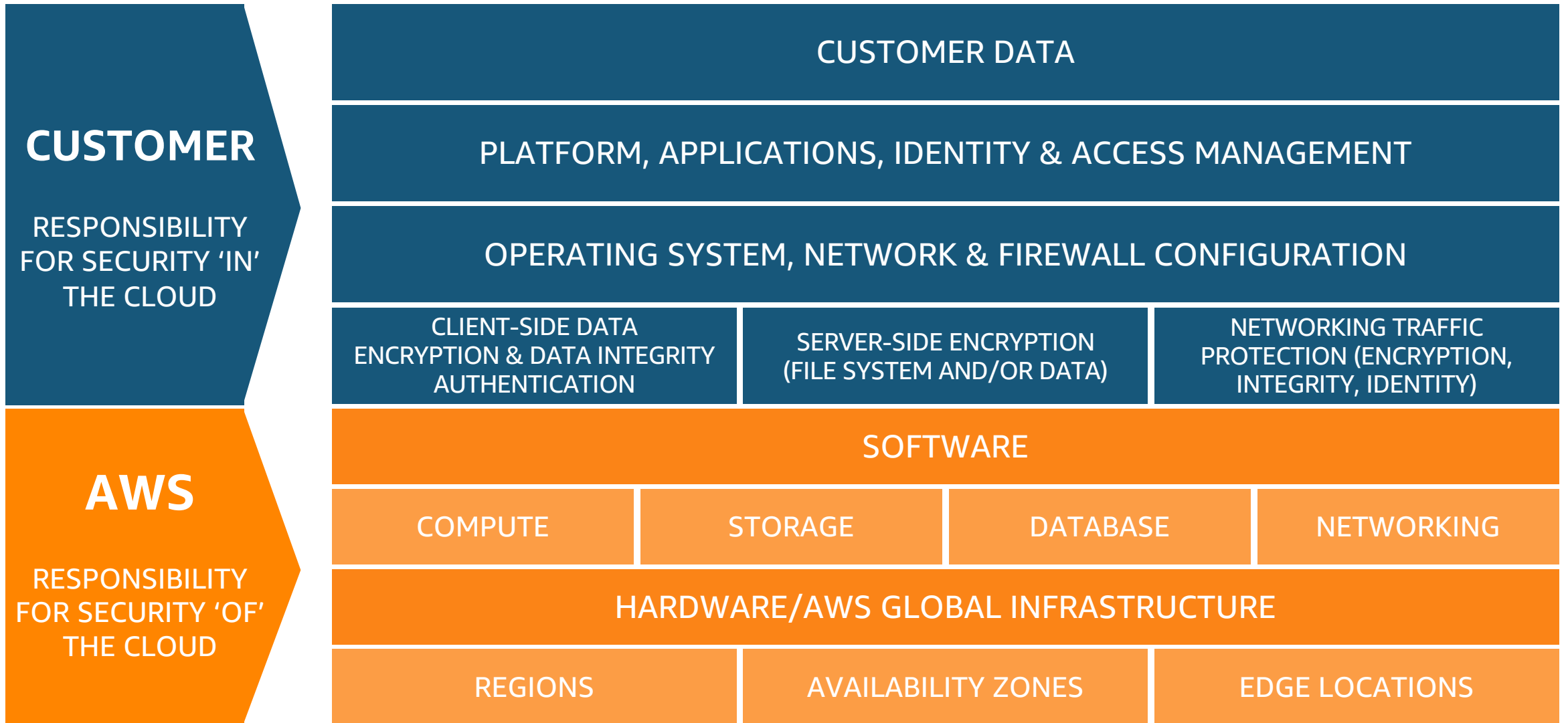
**“Everything fails,  
all the time.”**

**Werner Vogels**  
CTO, Amazon.com

aws

ed the Turing tes

# AWS Shared Responsibility Model



# The Situation

Meet Bob – Our new Junior developer, and taco enthusiast.



Bob just joined the team three months ago, fresh out of college. When he's not dreaming about finding the perfect taco truck, he's eager to prove himself as a developer. His manager just gave him his first solo project: deploying a WordPress site for a major government service that's expected to go viral once citizens discover how much time it will save them.

The conversation went something like this:

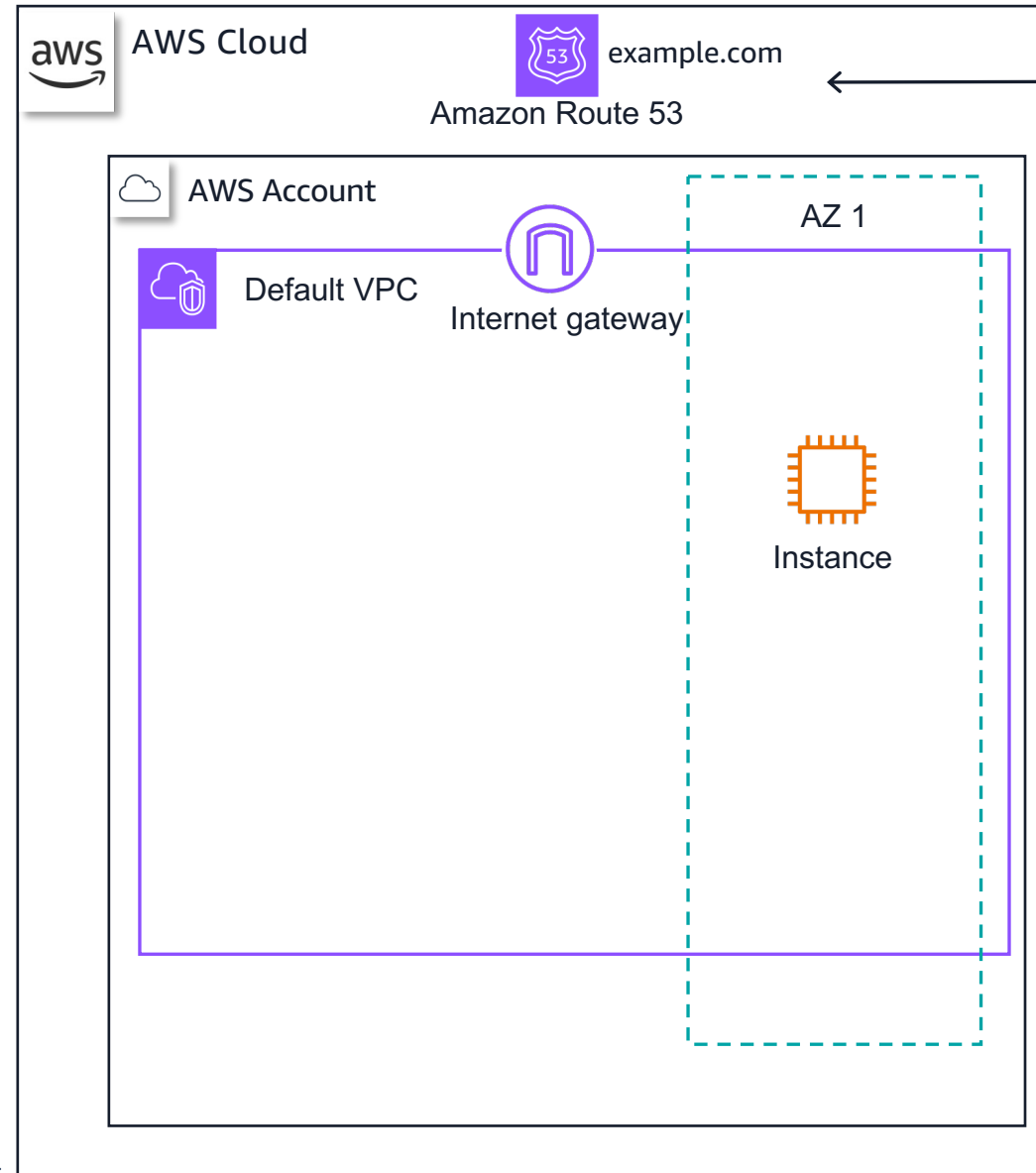
Manager: "Bob, can you get this WordPress site up in AWS for the client?"

Bob (confidently): "No problem! I've built WordPress sites on my laptop before!"

Bob gets to work. Here's what he did:

- Launched a single t2.micro EC2 instance in the default VPC
- Installed Apache, MySQL, and PHP directly on the instance
- Opened port 80 and 22 to 0.0.0.0/0 in the security group
- Proudly told his manager "It's live!" while heading out for his celebratory taco lunch

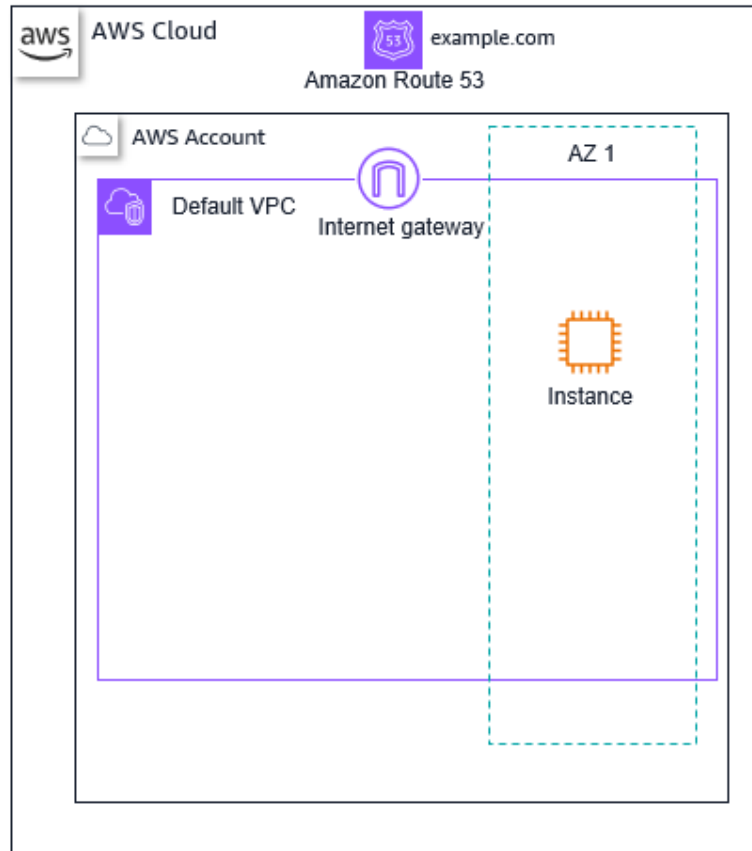
# Bob's Architecture



# Pillars of the AWS well-architected framework



# Bobs Bad Day

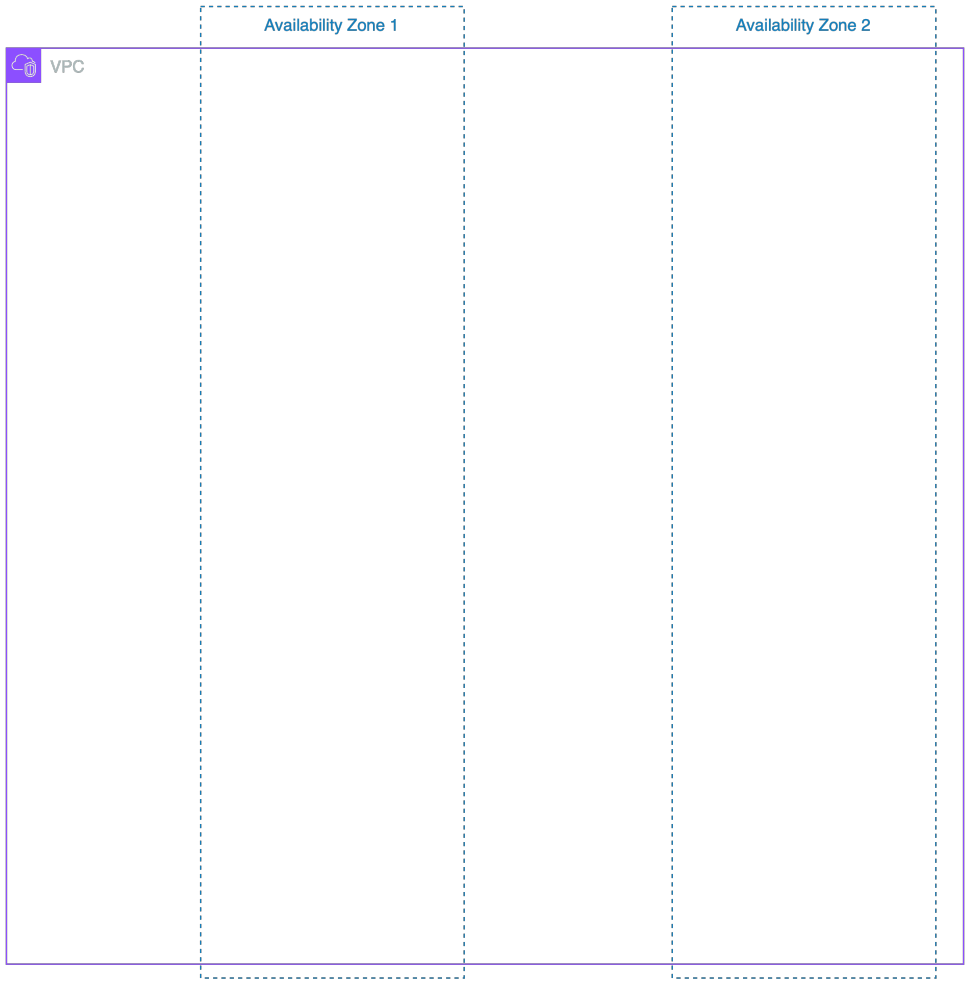


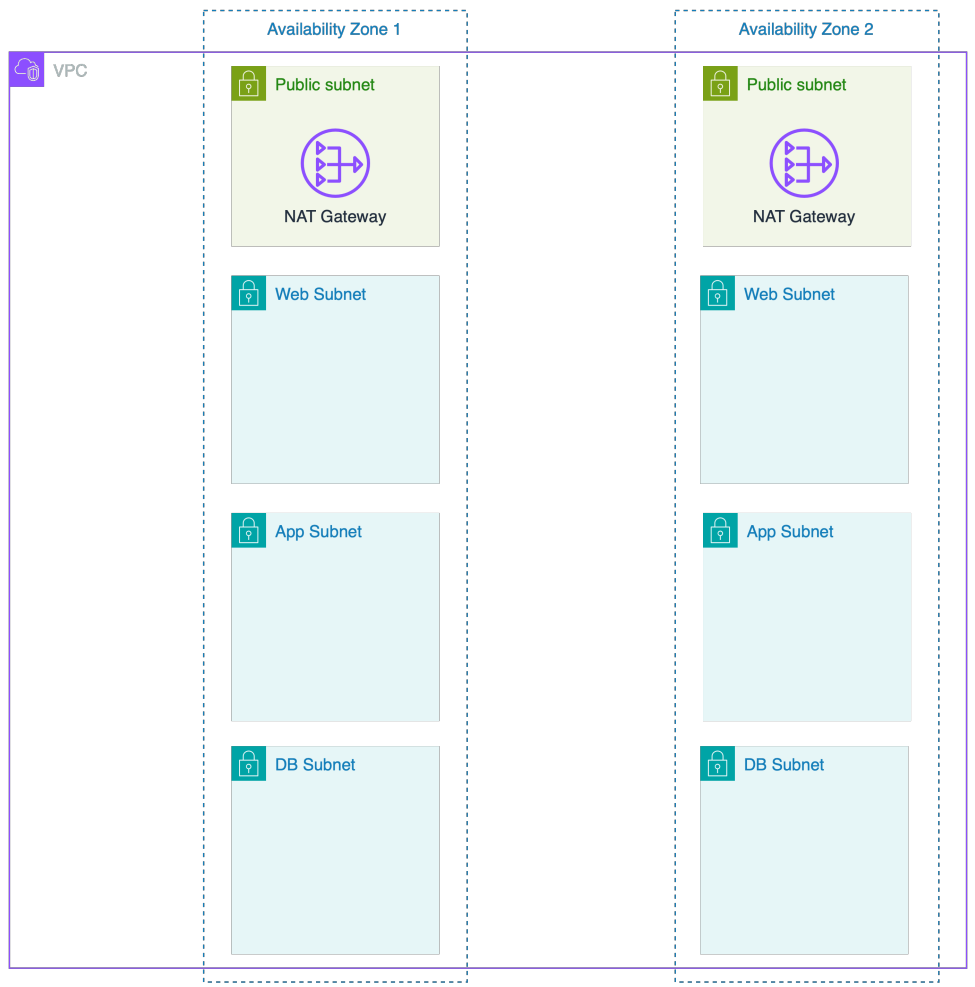
**Security**

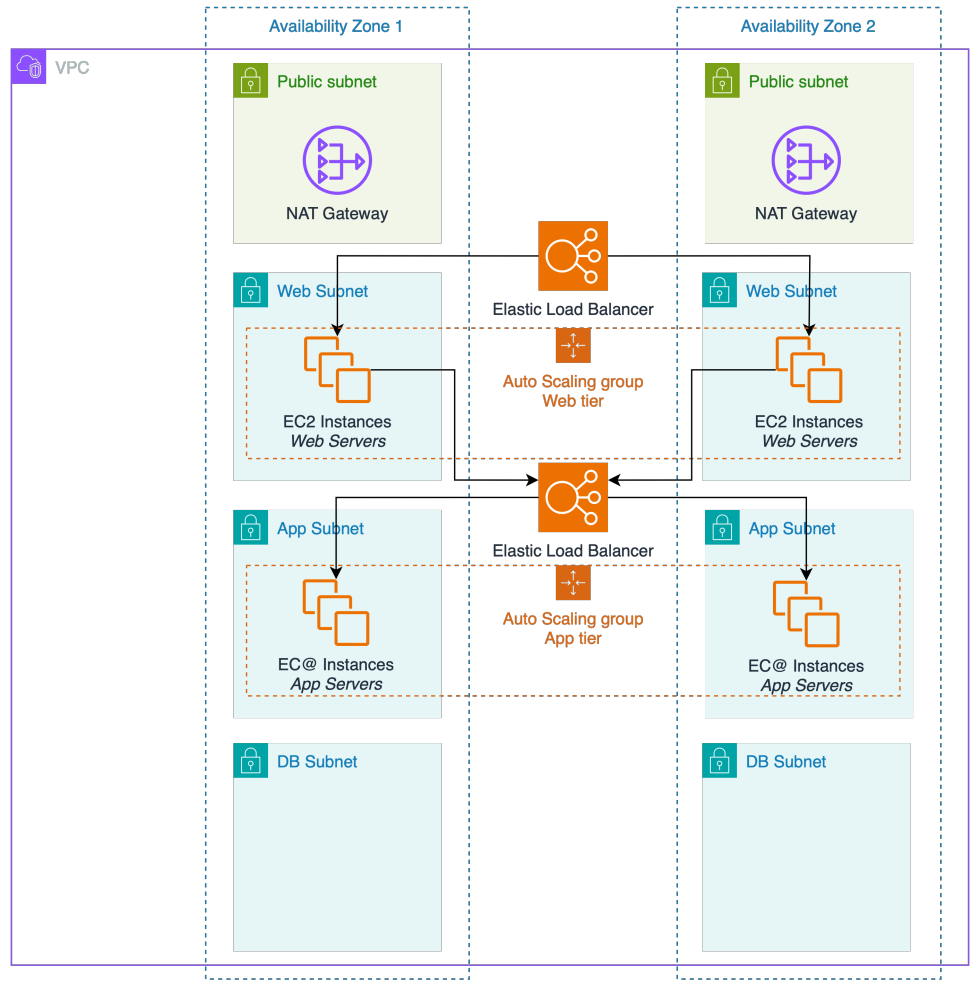
**Reliability**

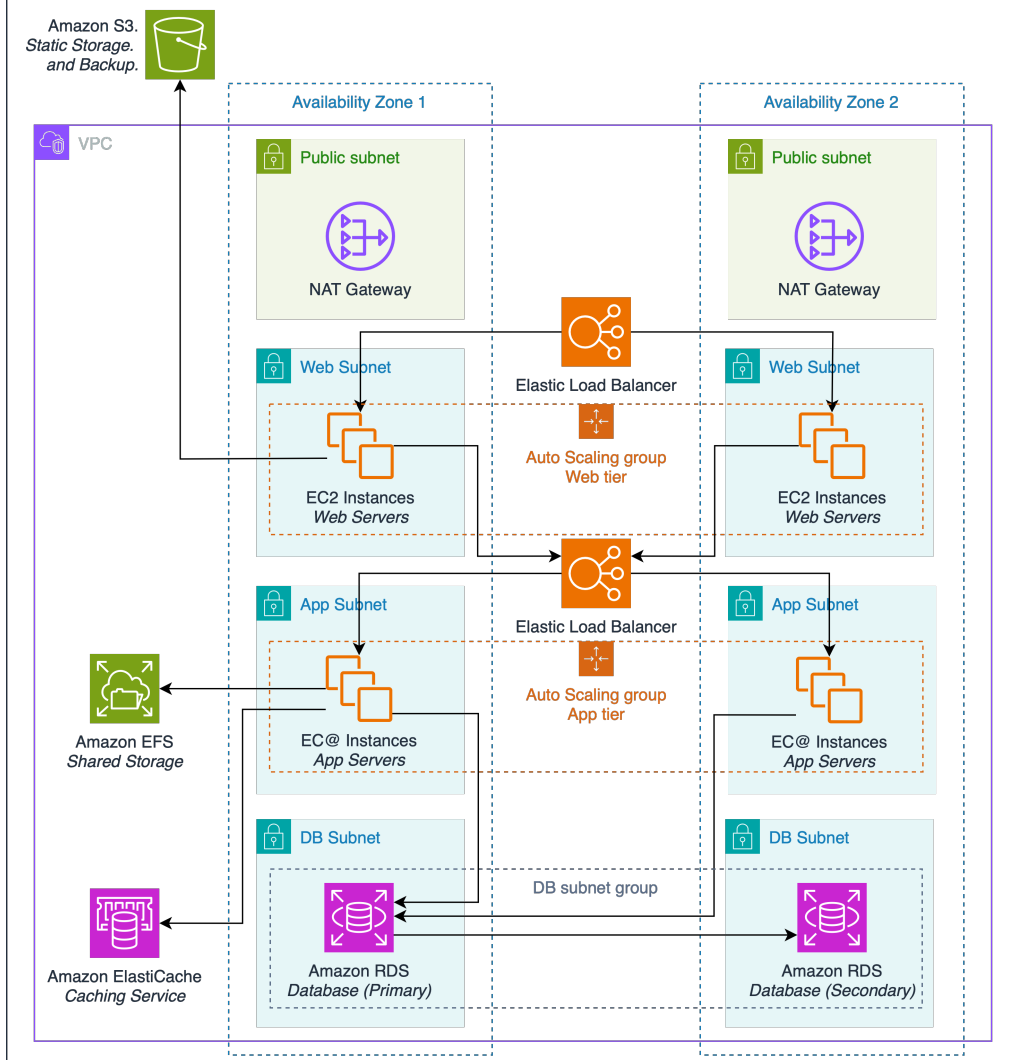
**Operational Excellence**

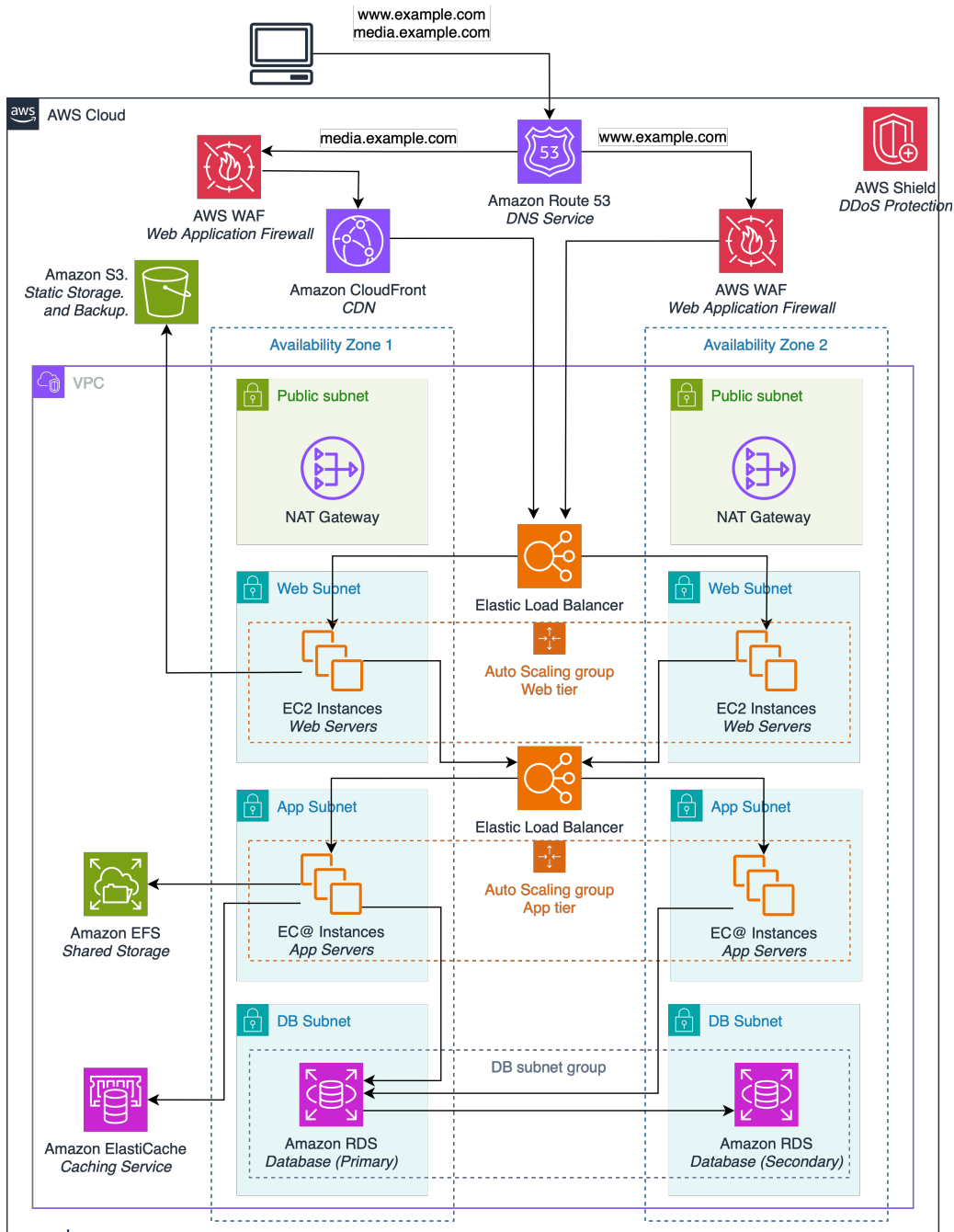
**Performance Efficiency /  
Cost Optimization**











# Security – Top design principles



- **Implement a strong identity foundation**
- **Maintain 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**

# AWS Security Services



## Identity and access management

AWS Identity and Access Management (IAM)  
AWS Single Sign-On  
AWS Organizations  
AWS Directory Service  
Amazon Cognito  
AWS Resource Access Manager



## Detective controls

AWS Security Hub  
Amazon GuardDuty  
Amazon Inspector  
Amazon CloudWatch  
AWS Config  
AWS CloudTrail  
VPC Flow Logs  
AWS IoT Device Defender  
AWS Security Lake  
Amazon GuardDuty



## Infrastructure protection

Network Edge Protection  
AWS Firewall Manager  
AWS Network Firewall  
AWS Shield Advanced  
AWS WAF  
Amazon Route 53 DNS Firewall  
AWS Verified Access  
Amazon VPC  
AWS PrivateLink  
AWS Systems Manager



## Data protection

Amazon Macie  
AWS Key Management Service (KMS)  
AWS CloudHSM  
AWS Certificate Manager  
AWS Secrets Manager  
AWS VPN  
Server-Side Encryption  
\*AWS Clean Rooms (Preview)



## Incident response

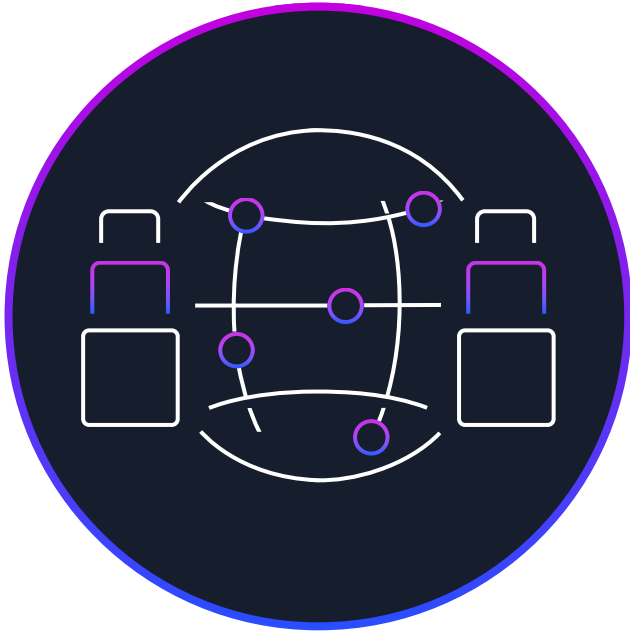
Amazon Detective  
Amazon EventBridge  
AWS Backup  
AWS Security Hub  
AWS Elastic Disaster Recovery



## Privacy and Compliance

AWS Artifact  
AWS Audit Manager  
Amazon CloudWatch  
AWS CloudTrail  
AWS Config  
AWS Security Hub  
AWS Systems Manager  
AWS Verified Permissions  
AWS Wickr

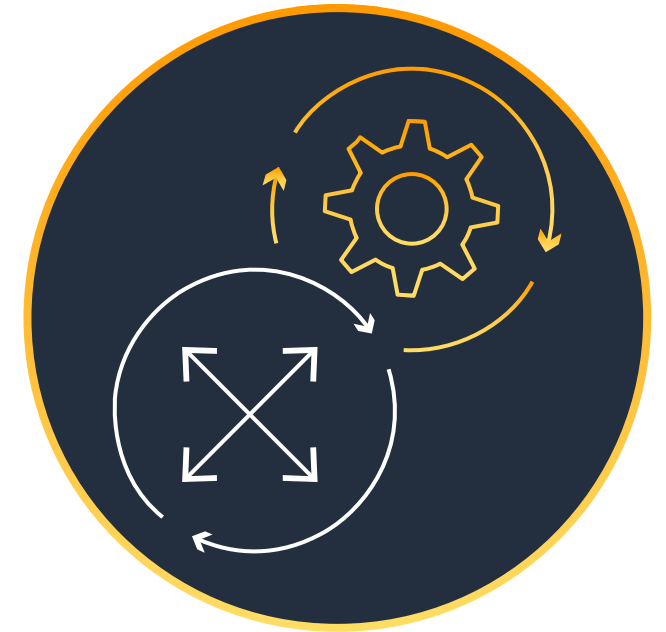
# Reliability – Top design principles



- ⌘ Automatically recover from failure
- ⌘ Test recovery procedures
- ⌘ Scale horizontally to increase aggregate workload availability
- ⌘ Stop guessing capacity
- ⌘ Manage change through automation

# Operational Excellence

- ⌘ Perform operations as code
- ⌘ Make frequent, small, reversible changes
- ⌘ Refine operation procedures frequently
- ⌘ Anticipate failure
- ⌘ Learn from all operational failures
- ⌘ Use managed services



# Performance efficiency & cost optimization



- ⌘ Democratize advanced technologies
- ⌘ Go global in minutes
- ⌘ Use serverless architectures
- ⌘ Experiment more often
- ⌘ Consider mechanical sympathy

**...but how do you scale this?**

# Use accounts as building blocks

## Account limits

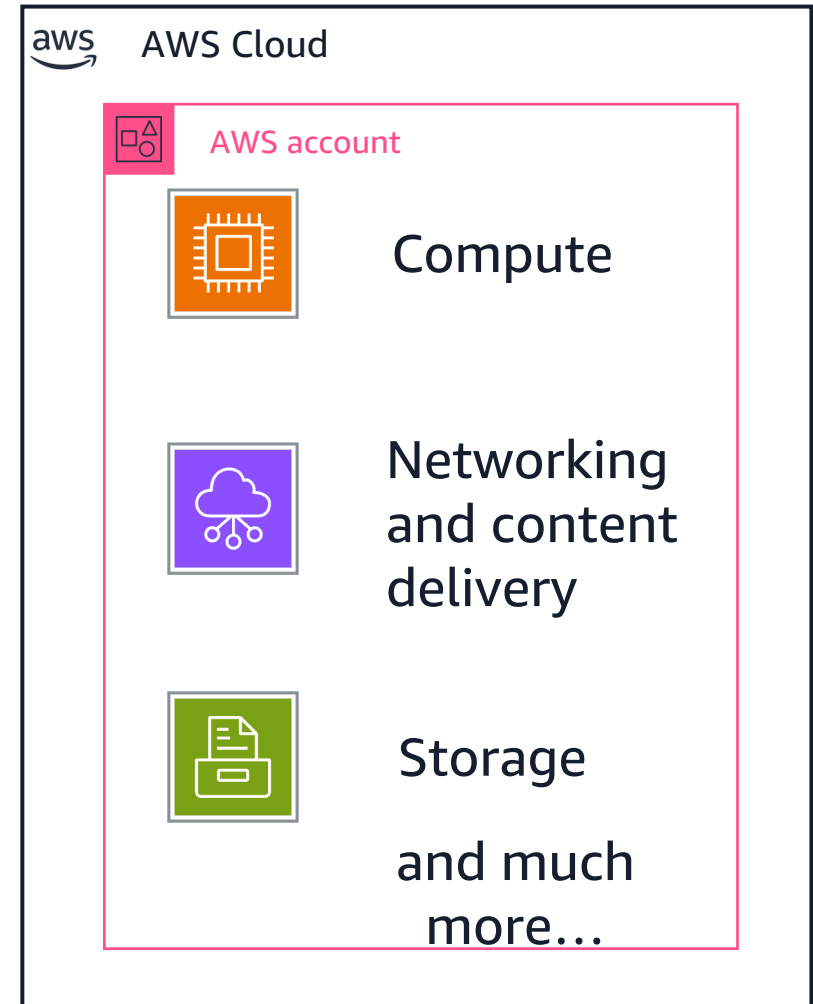
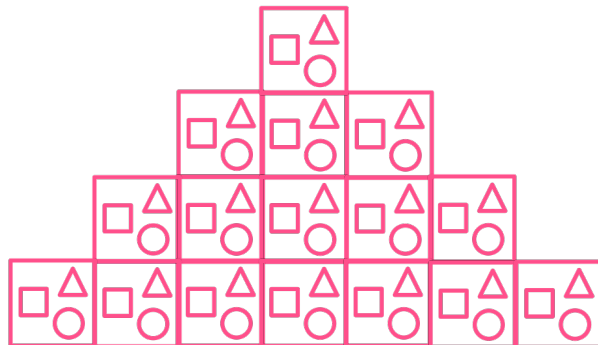
Quotas

## Security

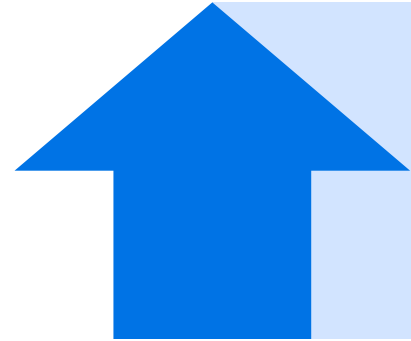
Natural boundaries,  
isolation

## Compliance/ business processes

Billing, custom  
requirements

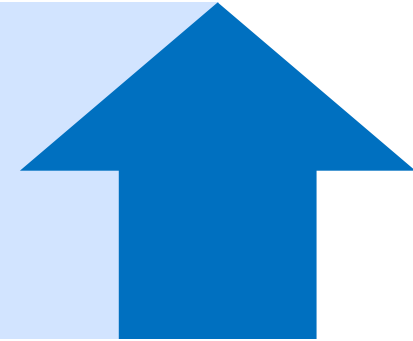


# Business agility and governance control



With AWS Control Tower, you don't have to choose between agility and control

**You can have both**



## Governance

---

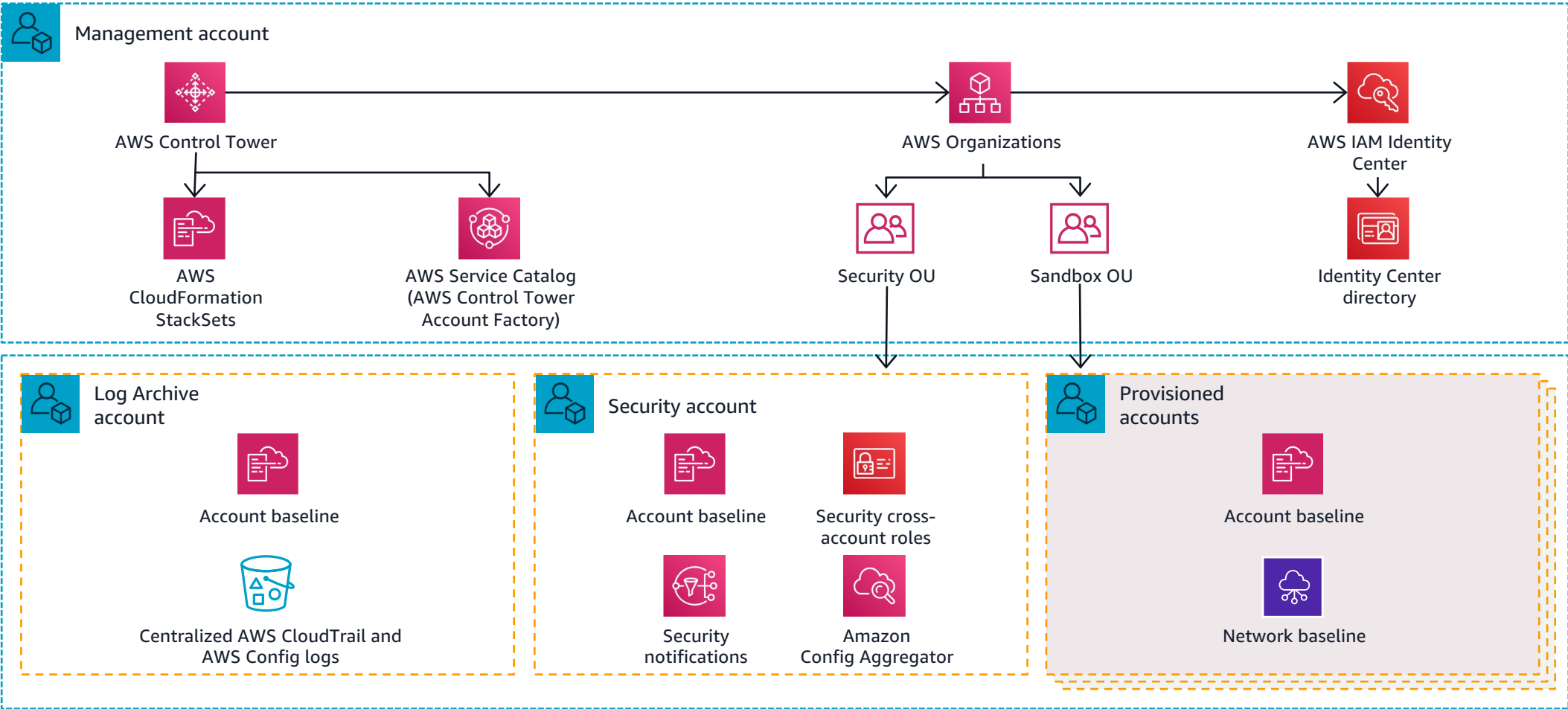
- Security
- Compliance
- Operations
- Spend management

## Agility

---

- Self-service access
- Experiment fast
- Respond quickly to change

# Landing zone foundation of AWS Control Tower





# Thank you!

## Ash Naik

Solutions Architect  
AWS  
ashjnaik@amazon.com

## Ryan Bachand

Solutions Architect  
AWS  
bacryan@amazon.com

Please complete the survey  
for this session



**Track: Cloud Fundamental**  
**Session: Cloud architectural patterns: Platform  
and application best practices**

Coming up NEXT

1:30pm – 3:00pm

**300**  
level

**Cloud Lab  
Potpourri**

Hands-On Cloud  
Adventure:  
Customize Your AWS  
Learning with  
Interactive Technical  
Workshops