AWS re:Invent

Being Well-Architected in the cloud

Julien Simon, Principal Technical Evangelist, AWS

@julsimon

julsimon@amazon.com

amazon web services

Customer Challenges

Faster response to change in market

Delivery time

Change Management

Reduce human errors

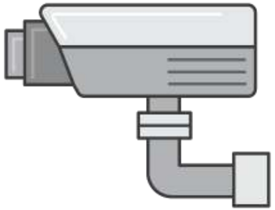
Scaling to demand

Faster recovery

High availability

Automation

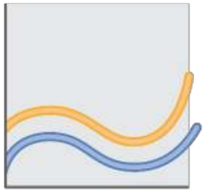
AWS Design Principles



Security by design

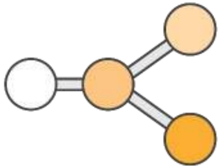
Test systems at scale

Data-driven architectures



Stop guessing capacity needs

Automate to enable experimentation



Allow for evolution

AWS Regions and AZs

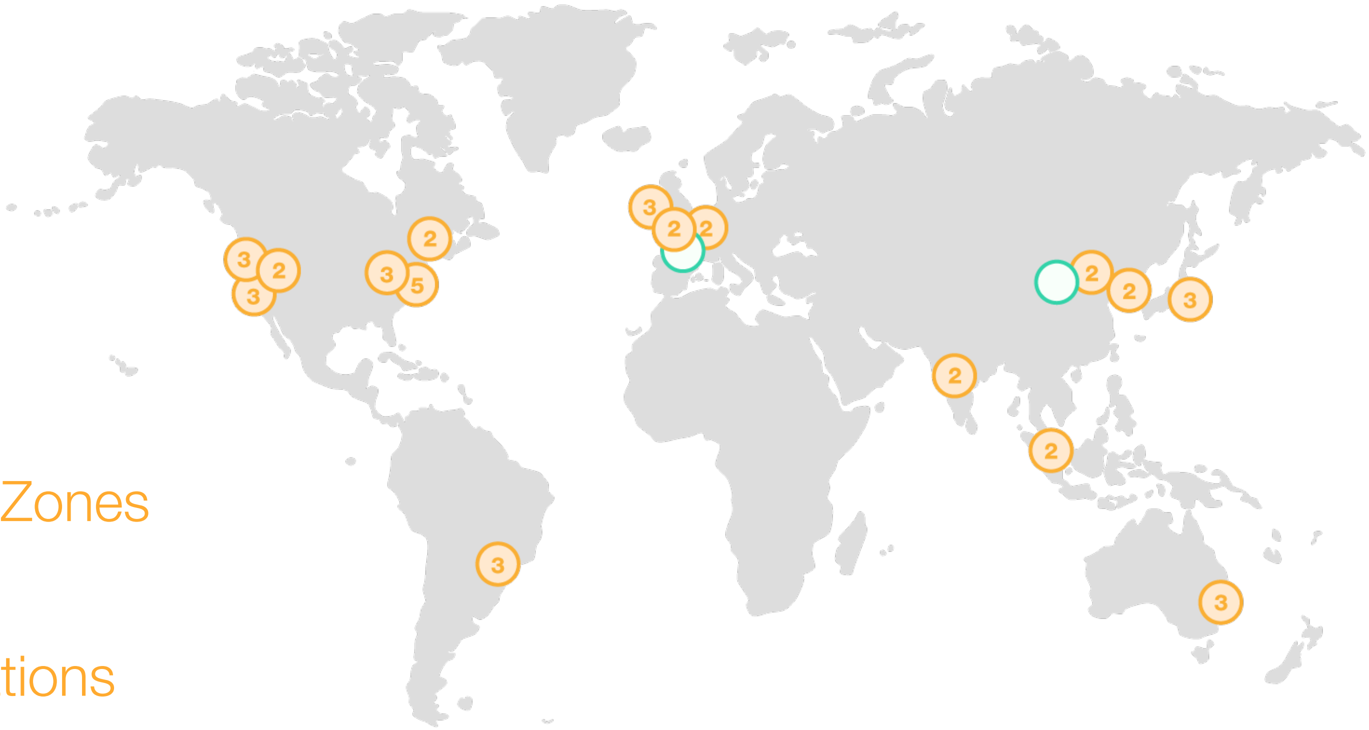
16 Regions

+2 coming

🇫🇷 🇨🇳

42 Availability Zones

68 Edge Locations



AWS well-architected framework

Set of questions you can use to evaluate how well an architecture is aligned to AWS best practices

Security

Reliability

Performance efficiency

Cost optimization

Operational excellence

The Security Pillar

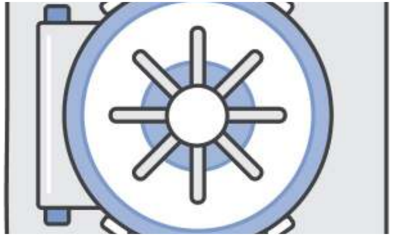
Security pillar

Protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies

Security at all layers

Enable traceability

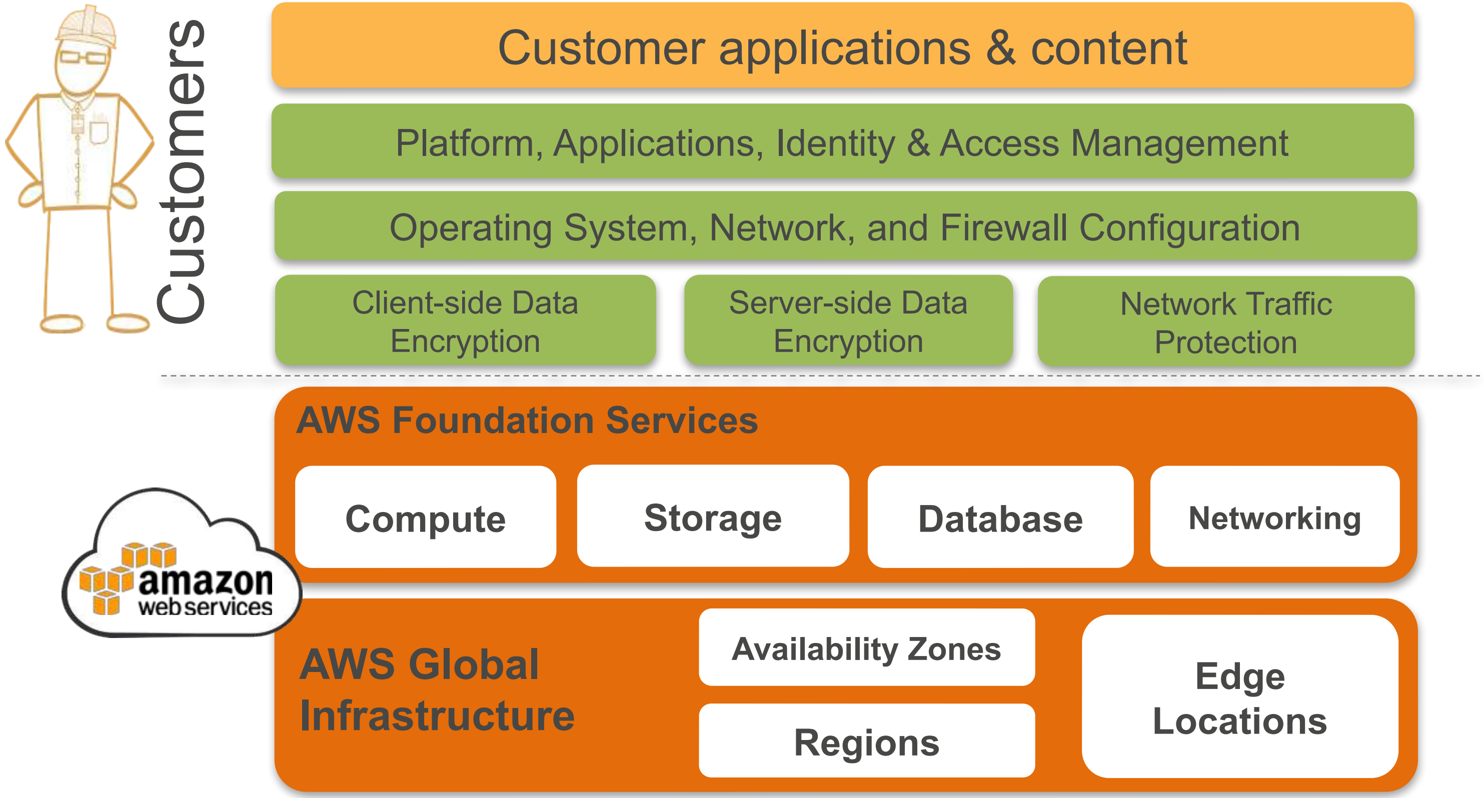
Implement a principle of least privilege



Focus on securing system

Automate security best practices

Shared Responsibility



Customers

Customer applications & content

Platform, Applications, Identity & Access Management

Operating System, Network, and Firewall Configuration

Client-side Data Encryption

Server-side Data Encryption

Network Traffic Protection

amazon web services

AWS Foundation Services

Compute

Storage

Database

Networking

AWS Global Infrastructure

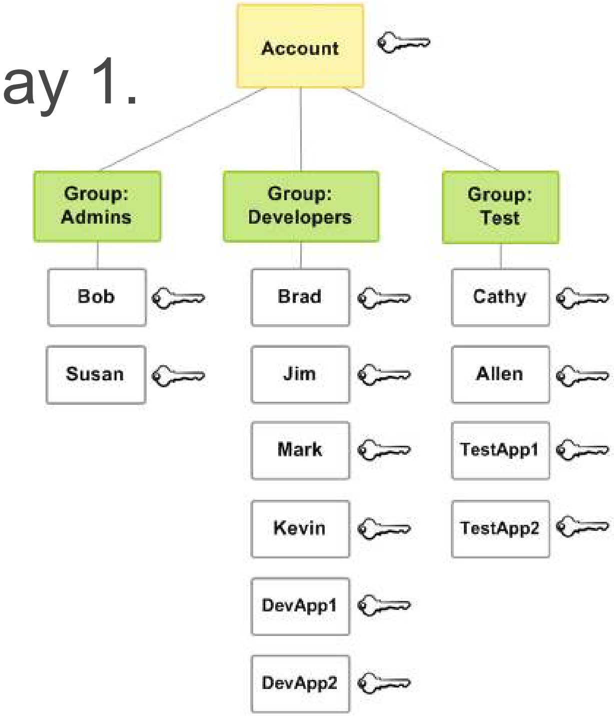
Availability Zones

Regions

Edge Locations

Credentials

- Enforce MFA for everyone from day 1.
- Use AWS IAM Users and Roles from day 1.
- Enforce strong passwords.
- Protect and rotate credentials.
- No access keys in code.



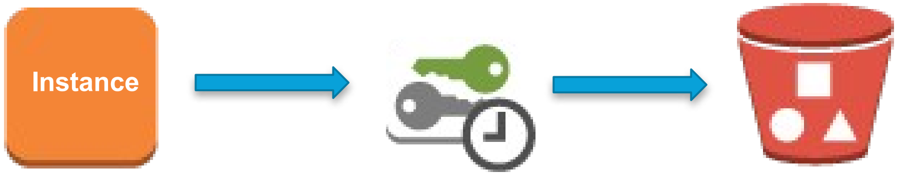
EC2 Role

1: Create EC2 role

Create role in IAM service with limited policy

2: Launch EC2 instance

Launch instance with role



3: App retrieves credentials

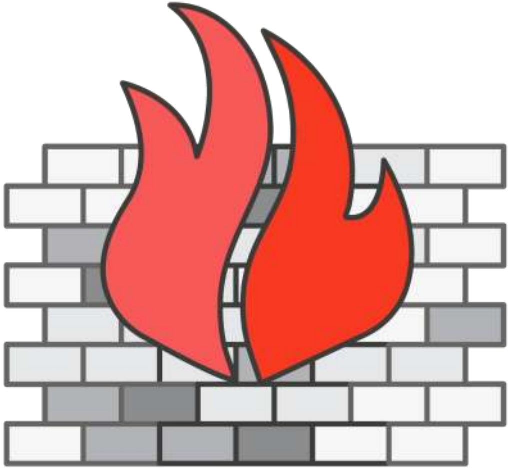
Using AWS SDK application retrieves temporary credentials

4: App accesses AWS resource(s)

Using AWS SDK application uses credentials to access resource(s)

Network and Boundary

- Security groups are built-in stateful firewalls
- Divide layers of the stack into subnets
- Use a bastion host for access
- Implement host based controls



Monitoring and Auditing

- Capture & audit AWS CloudTrail, Amazon VPC and Amazon CloudWatch logs.
- Collect all logs centrally.
- Setup alerts.

Amazon Virtual Private Cloud

AWS Identity & Access Manager

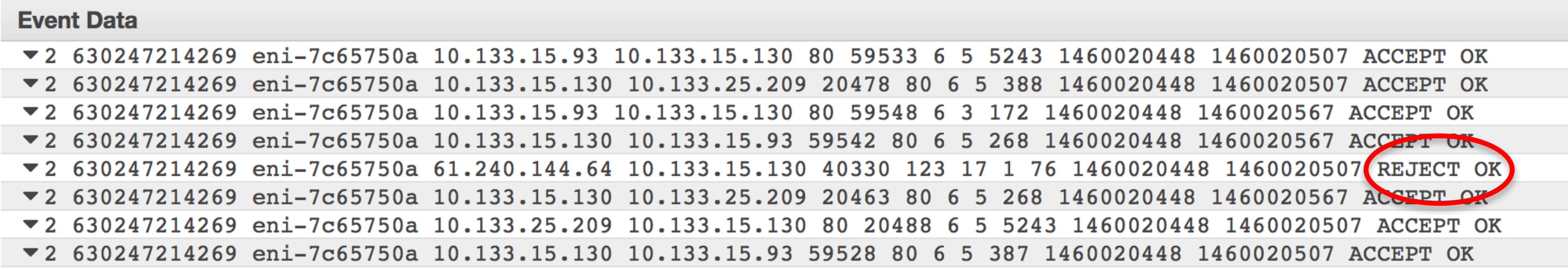
AWS Key Management Service

AWS CloudTrail

AWS Config

Monitoring and Auditing

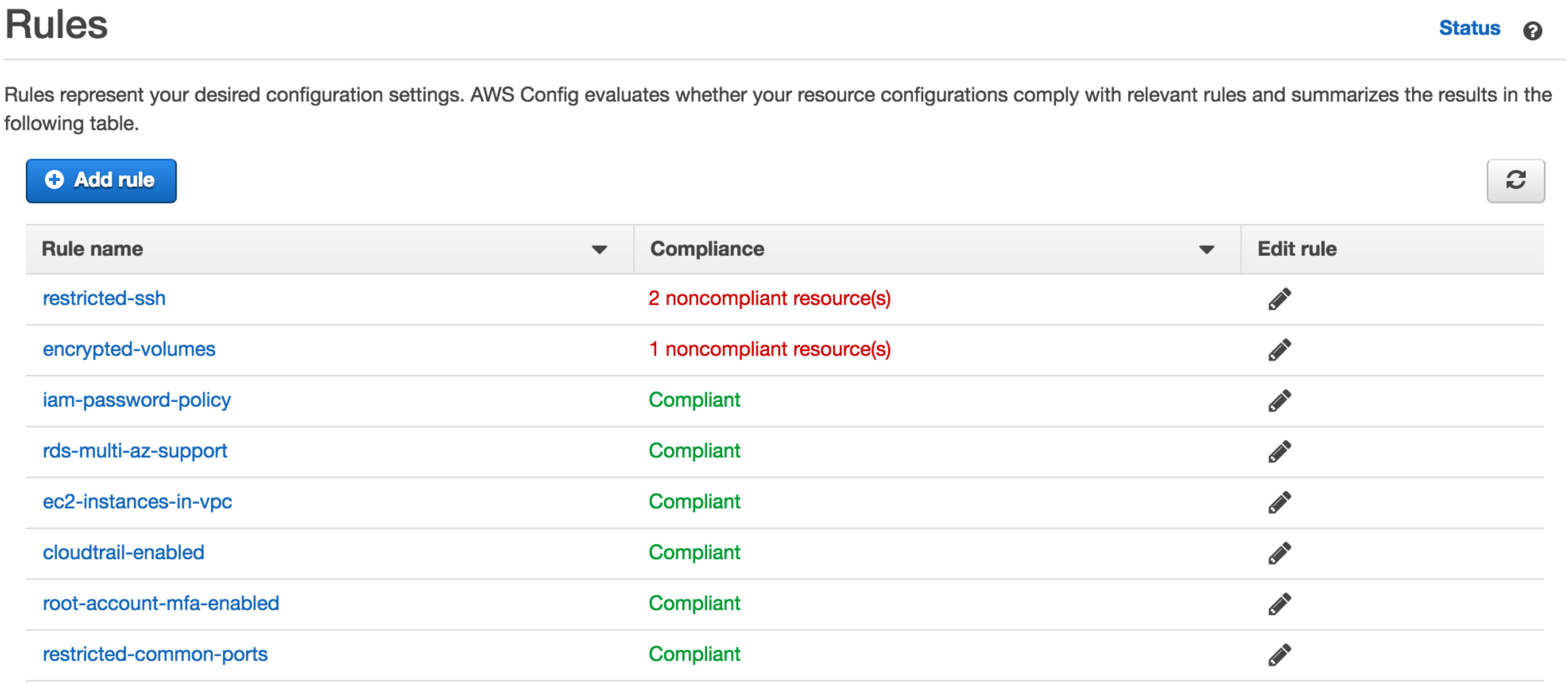
- Amazon VPC Flow Logs – Developers Best Friend



Event Data

```
▼2 630247214269 eni-7c65750a 10.133.15.93 10.133.15.130 80 59533 6 5 5243 1460020448 1460020507 ACCEPT OK
▼2 630247214269 eni-7c65750a 10.133.15.130 10.133.25.209 20478 80 6 5 388 1460020448 1460020507 ACCEPT OK
▼2 630247214269 eni-7c65750a 10.133.15.93 10.133.15.130 80 59548 6 3 172 1460020448 1460020567 ACCEPT OK
▼2 630247214269 eni-7c65750a 10.133.15.130 10.133.15.93 59542 80 6 5 268 1460020448 1460020567 ACCEPT OK
▼2 630247214269 eni-7c65750a 61.240.144.64 10.133.15.130 40330 123 17 1 76 1460020448 1460020507 REJECT OK
▼2 630247214269 eni-7c65750a 10.133.15.130 10.133.25.209 20463 80 6 5 268 1460020448 1460020567 ACCEPT OK
▼2 630247214269 eni-7c65750a 10.133.25.209 10.133.15.130 80 20488 6 5 5243 1460020448 1460020507 ACCEPT OK
▼2 630247214269 eni-7c65750a 10.133.15.130 10.133.15.93 59528 80 6 5 387 1460020448 1460020507 ACCEPT OK
```

Verify everything, always, with AWS Config



Rules

Rules represent your desired configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and summarizes the results in the following table.

Rule name	Compliance	Edit rule
restricted-ssh	2 noncompliant resource(s)	✎
encrypted-volumes	1 noncompliant resource(s)	✎
iam-password-policy	Compliant	✎
rds-multi-az-support	Compliant	✎
ec2-instances-in-vpc	Compliant	✎
cloudtrail-enabled	Compliant	✎
root-account-mfa-enabled	Compliant	✎
restricted-common-ports	Compliant	✎

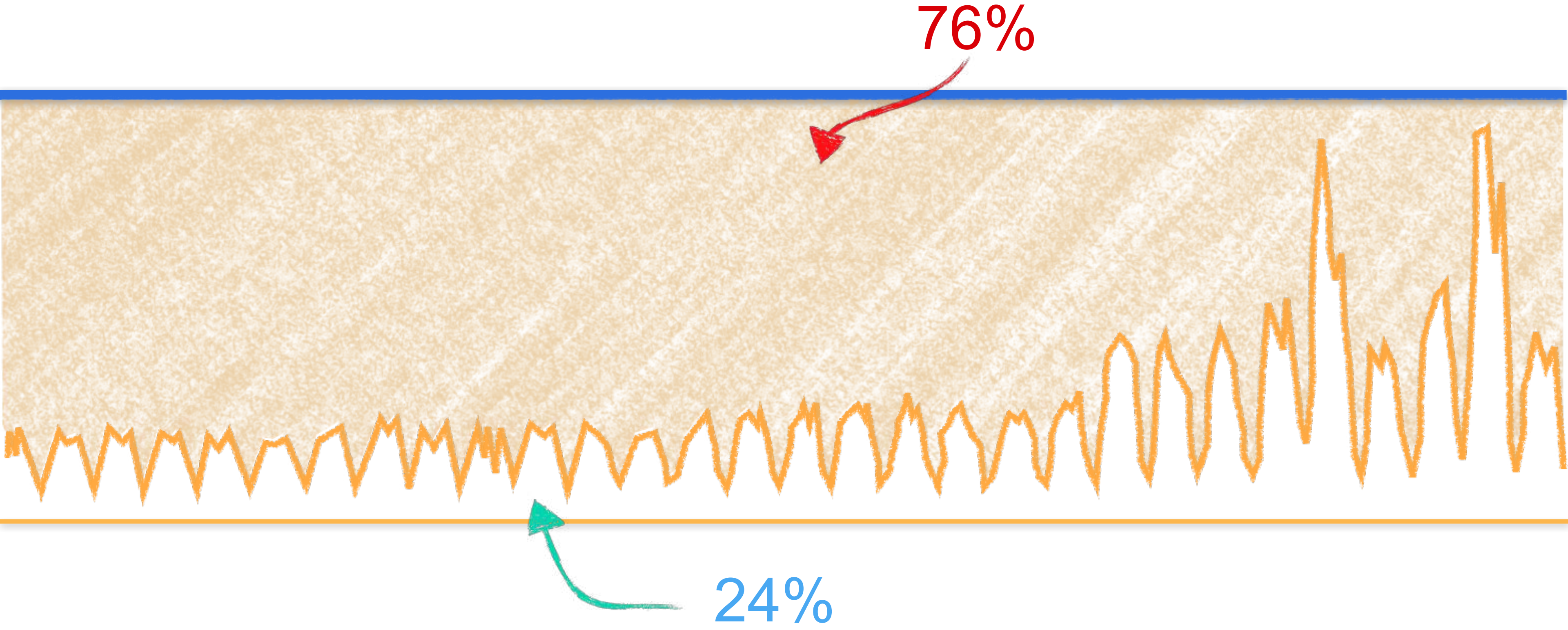
The Reliability Pillar

Reliability pillar

Ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues

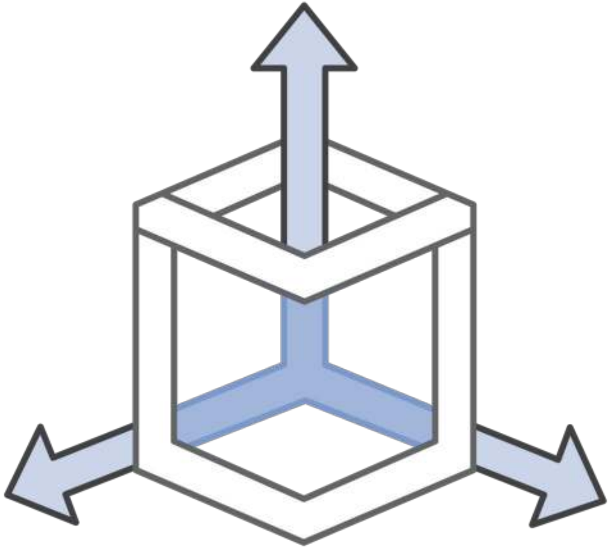
- Test recovery procedures
- Automatically recover from failure
- Scale horizontally to increase availability
- Stop guessing capacity

Utilization vs Provisioned capacity

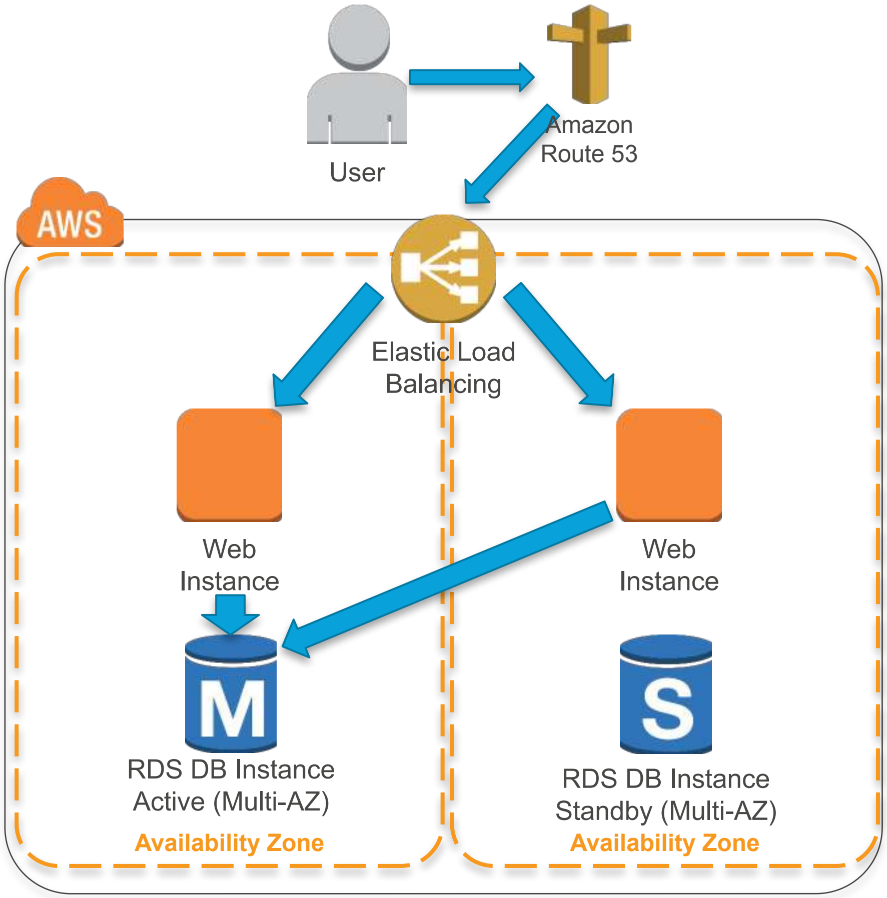


High Availability

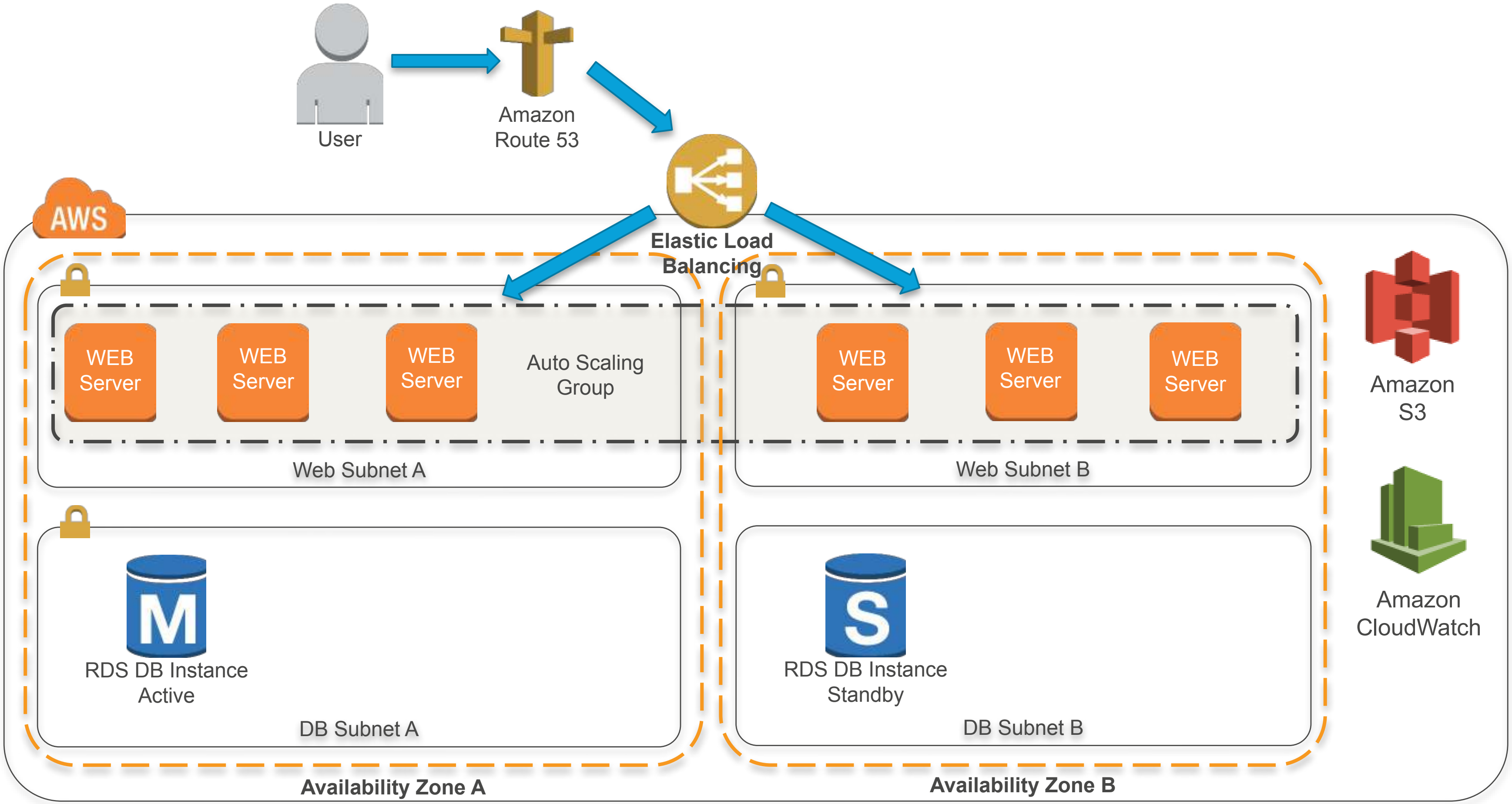
- No Single Point of Failure
- Multiple Availability Zones
- Load Balancing
- Auto Scaling and Healing



Multi-AZ Architecture



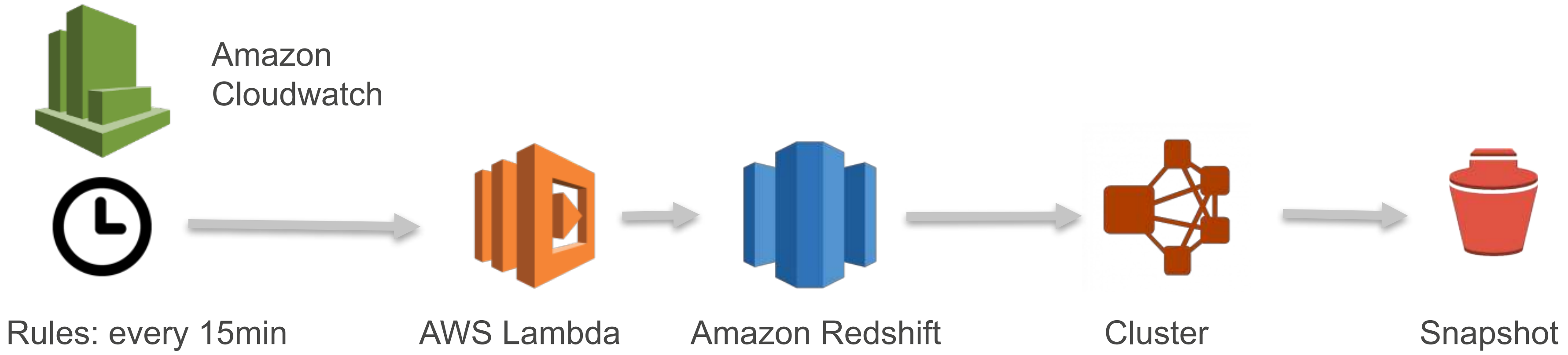
Multi-AZ, Load Balanced, Auto Scaled



Backup and DR

- Define Objectives
- Backup Strategy
- Periodic Recovery Testing
- Automated Recovery
- Periodic Reviews

Automated backups using AWS Lambda



Amazon Cloudwatch

Rules: every 15min

AWS Lambda

Amazon Redshift

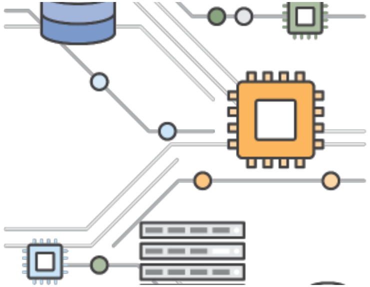
Cluster

Snapshot

The Performance Pillar

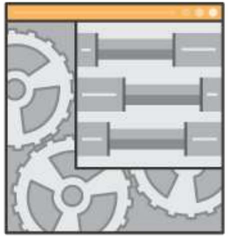
Performance efficiency pillar

Efficiently use of computing resources to meet requirements, and maintaining that efficiency as demand changes and technologies evolve



Democratize advanced technologies

Go global in minutes



Use server-less architectures

Experiment more often

Right Sizing

- Reference Architecture
- Quick Start Reference Deployments
- Benchmarking
- Load Testing
- Cost / Budget
- Monitoring and Notification

Proximity and Caching

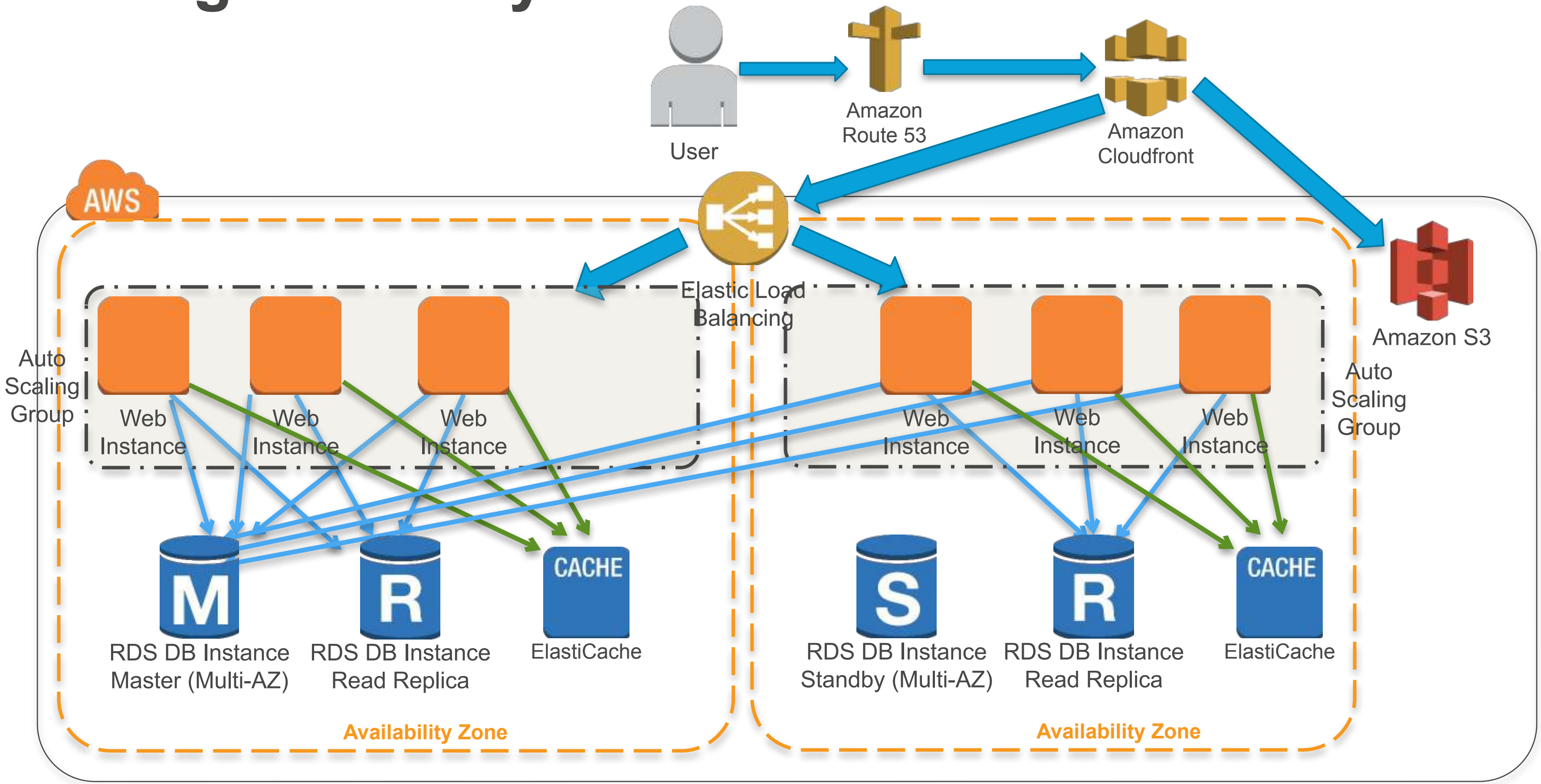
- Content Delivery Network (CDN)
- Database Caching
- Reduce Latency
- Pro-active Monitoring and Notification

Amazon
CloudFront

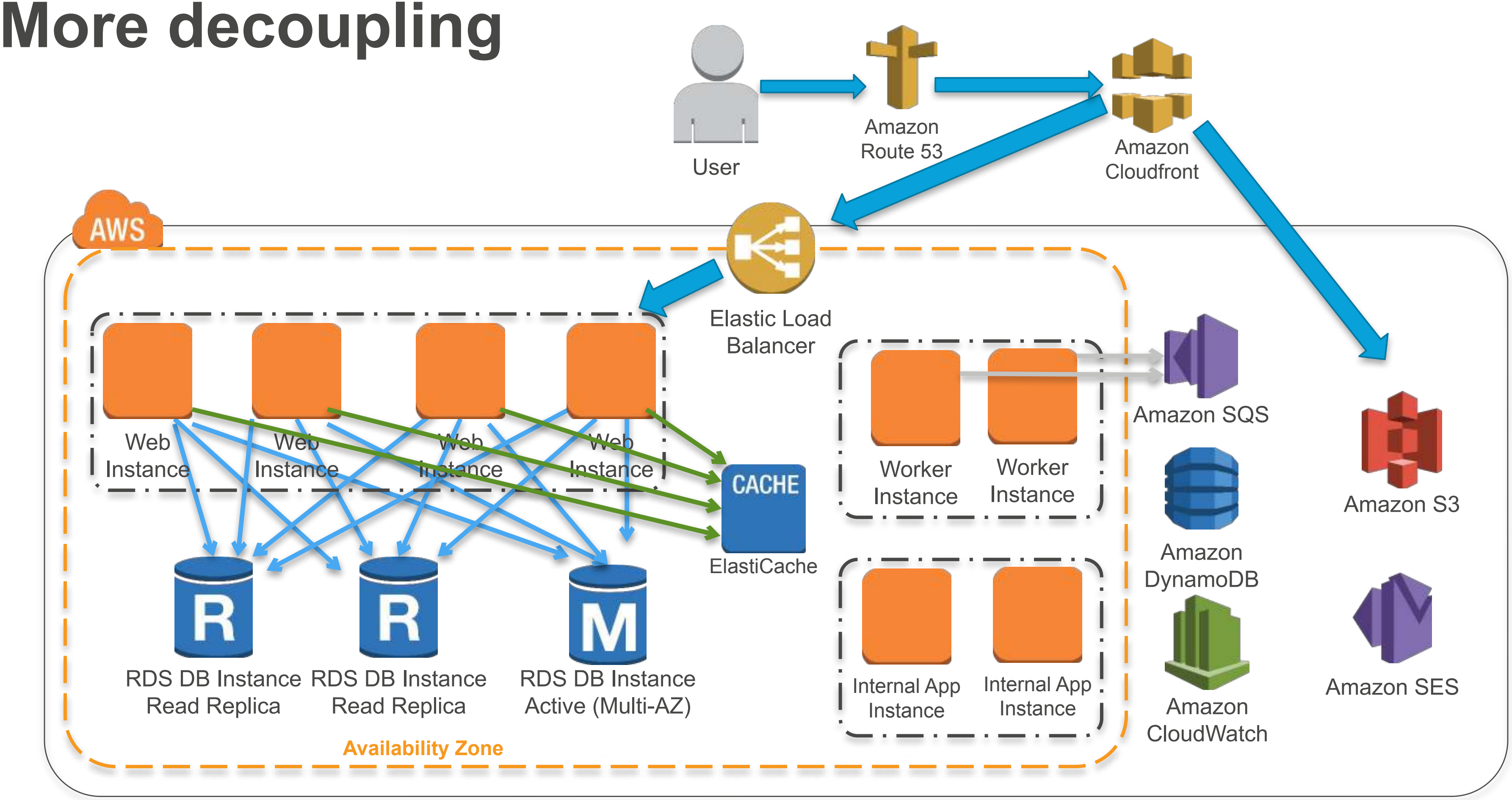
Amazon
ElastiCache

RDS DB
instance read
replica

Scaling all the layers

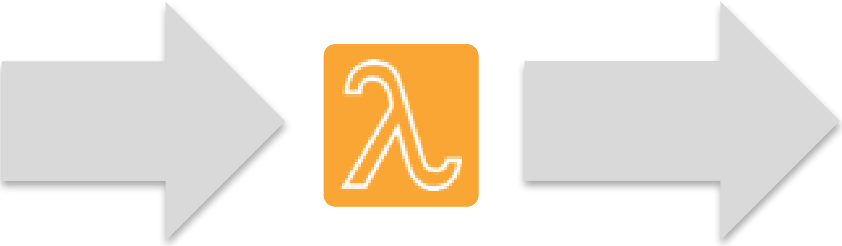


More decoupling



AWS Lambda

Functions are the unit of deployment and scaling.

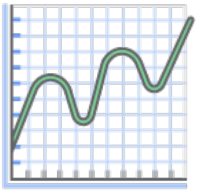


Invocation

“Lambda functions”

Action

No servers to manage



Continuous scaling

Never pay for idle – no cold servers

The Cost Optimization Pillar

Cost optimization pillar

Assess your ability to avoid or eliminate unneeded costs or suboptimal resources, and use those savings on differentiated benefits for your business

Analyze and attribute expenditure

Managed services to reduce TCO

Adopt a consumption model

Benefits from economies of scale

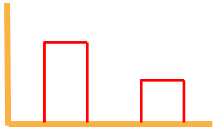
Stop spending money on data center operations

Pricing Model

- On Demand

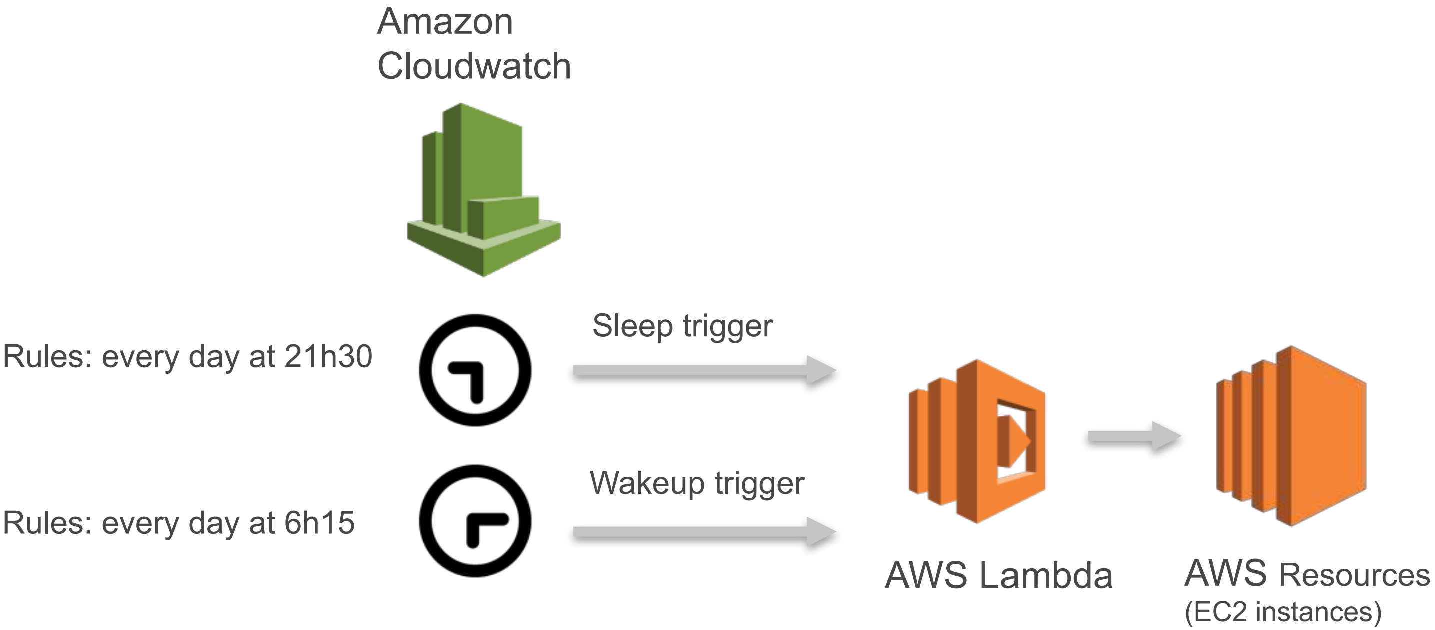
- Reserved

- Spot



- Dedicated

Auto Start/Shutdown of Instances



Amazon Cloudwatch

Rules: every day at 21h30

Sleep trigger

Rules: every day at 6h15

Wakeup trigger

AWS Lambda

AWS Resources
(EC2 instances)

Managed Services

- Let AWS do the heavy lifting.
- Databases, caches and big data solutions.
- Application Level Services.

Amazon RDS

Amazon DynamoDB

Amazon Redshift

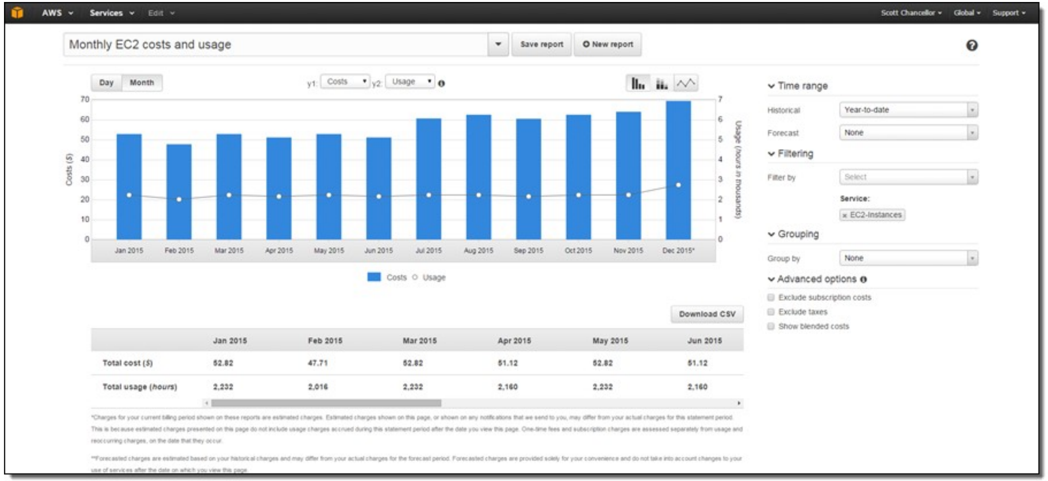
Amazon ElastiCache

AWS Elastic Beanstalk

Amazon Elasticsearch Service

Manage Expenditure

- Tag Resources
- Track Project Lifecycle
- Profile Applications vs Cost
- Monitor Usage & Spend



Monthly EC2 costs and usage

Time range

Historical: Year-to-date

Forecast: None

Filtering

Filter by: Select

Service: EC2-Instances

Grouping

Group by: None

Advanced options

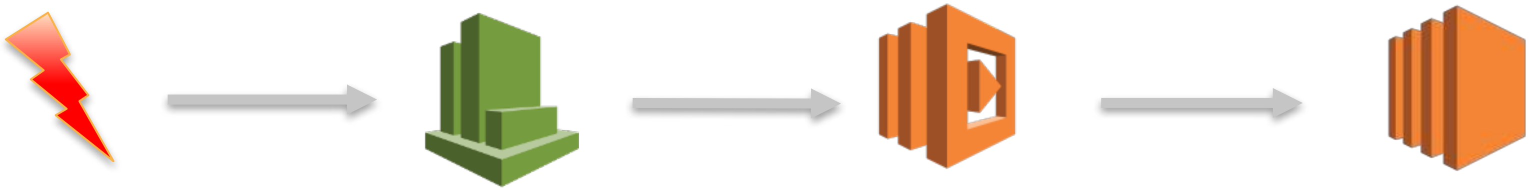
□ Exclude subscription costs
□ Exclude taxes
□ Show blended costs

	Jan 2015	Feb 2015	Mar 2015	Apr 2015	May 2015	Jun 2015
Total cost ($)	52.82	47.71	52.82	51.12	52.82	51.12
Total usage (*hours*)	2,232	2,016	2,232	2,160	2,232	2,160

*Charges for your current billing period shown on these reports are estimated charges. Estimated charges shown on this page, or shown on any notifications that we send to you, may differ from your actual charges for this statement period. This is because estimated charges presented on this page do not include usage charges accrued during this statement period after the date you view this page. One-time fees and subscription charges are assessed separately from usage and reoccurring charges, on the date that they occur.

**Forecasted charges are estimated based on your historical charges and may differ from your actual charges for the forecast period. Forecasted charges are provided solely for your convenience and do not take into account changes to your use of services after the date on which you view this page.

Auto Tagging resources as they start



Events:
RunInstances

Amazon
Cloudwatch

AWS Lambda

EC2 Instances
Tag:
Owner = userName
PrincipalId = aws:userid

The Operational Excellence Pillar

Operational excellence pillar

Operational practices and procedures used to manage production workloads

Perform operations with code

Align operations processes to business objectives

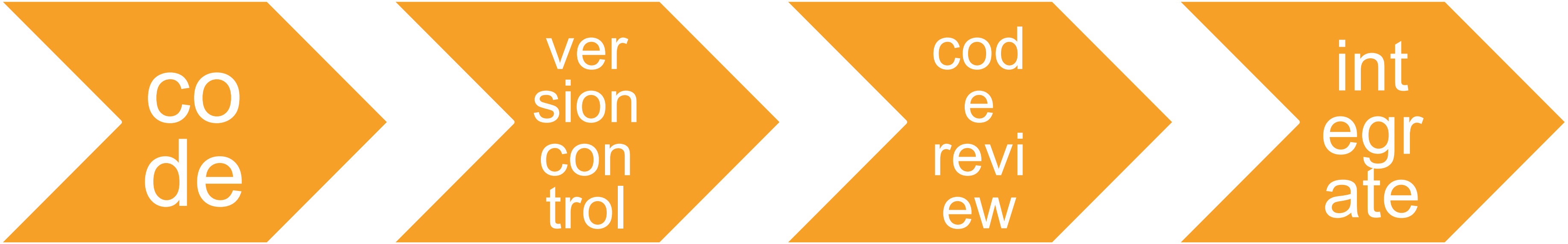
Make regular, small, incremental changes

Test for responses to unexpected events

Learn from operational events and failures

Keep operations procedures current

Infrastructure-as-code workflow



code

version control

code review

integrate

“It’s all software”

- Create templates of your infrastructure.
- Version control/replicate/update templates like code.
- Integrates with development, CI/CD, management tools

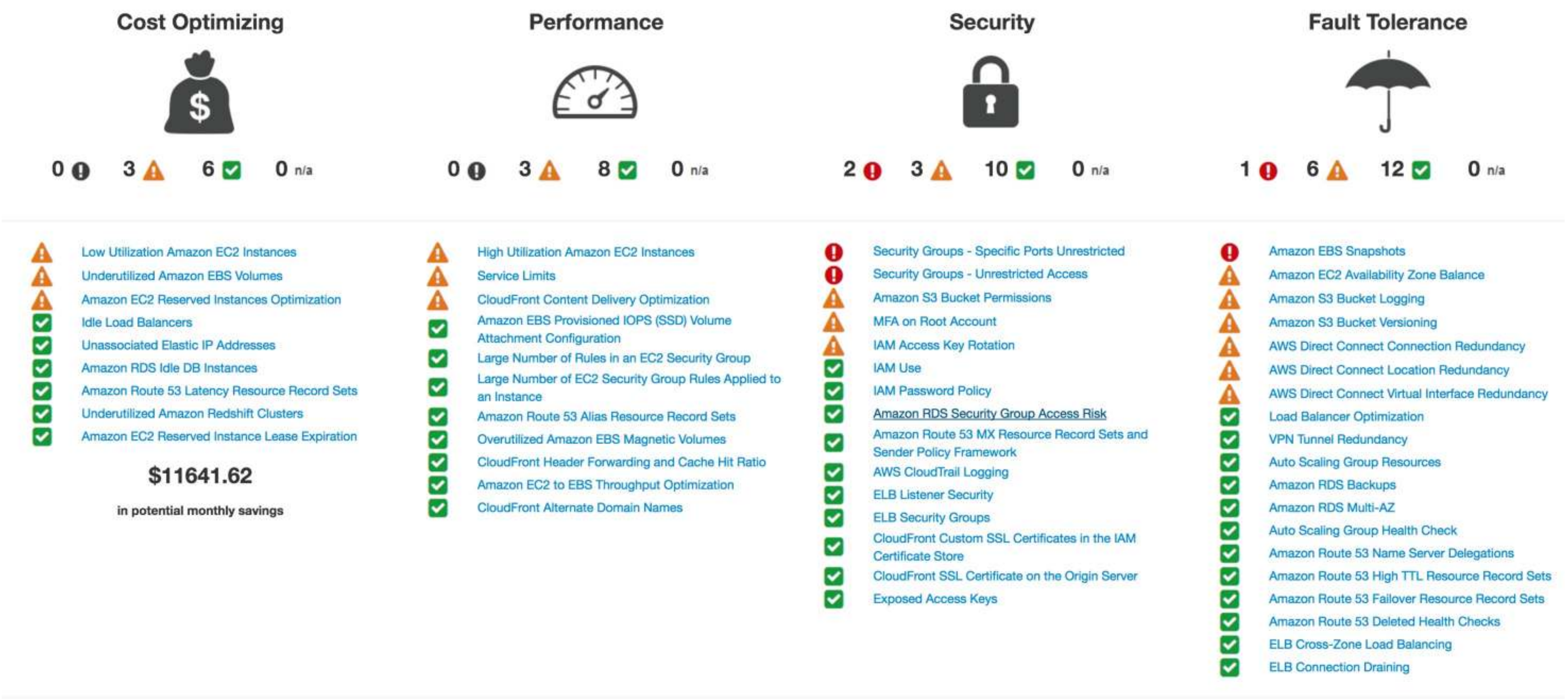
AWS CloudFormation

Some tips … from my own experience

- Architecture as code – code everything.
- Automate everything: “Invest time to save time”
- Don’t reinvent the wheel; managed services are your best friends.
- Embrace security early on.
- Test your DR strategy regularly.
- Serverless architectures free you from managing infrastructure.
- Did I mention automation?

And don’t forget …

Trusted Advisor



Cost Optimizing

0 ❗ 3 ⚠ 6 ☑ 0 n/a

- ⚠ Low Utilization Amazon EC2 Instances
- ⚠ Underutilized Amazon EBS Volumes
- ⚠ Amazon EC2 Reserved Instances Optimization
- ☑ Idle Load Balancers
- ☑ Unassociated Elastic IP Addresses
- ☑ Amazon RDS Idle DB Instances
- ☑ Amazon Route 53 Latency Resource Record Sets
- ☑ Underutilized Amazon Redshift Clusters
- ☑ Amazon EC2 Reserved Instance Lease Expiration

$11641.62

in potential monthly savings

Performance

0 ❗ 3 ⚠ 8 ☑ 0 n/a

- ⚠ High Utilization Amazon EC2 Instances
- ⚠ Service Limits
- ⚠ CloudFront Content Delivery Optimization
- ☑ Amazon EBS Provisioned IOPS (SSD) Volume Attachment Configuration
- ☑ Large Number of Rules in an EC2 Security Group
- ☑ Large Number of EC2 Security Group Rules Applied to an Instance
- ☑ Amazon Route 53 Alias Resource Record Sets
- ☑ Overutilized Amazon EBS Magnetic Volumes
- ☑ CloudFront Header Forwarding and Cache Hit Ratio
- ☑ Amazon EC2 to EBS Throughput Optimization
- ☑ CloudFront Alternate Domain Names

Security

2 ❗ 3 ⚠ 10 ☑ 0 n/a

- ❗ Security Groups - Specific Ports Unrestricted
- ❗ Security Groups - Unrestricted Access
- ⚠ Amazon S3 Bucket Permissions
- ⚠ MFA on Root Account
- ⚠ IAM Access Key Rotation
- ☑ IAM Use
- ☑ IAM Password Policy
- ☑ Amazon RDS Security Group Access Risk
- ☑ Amazon Route 53 MX Resource Record Sets and Sender Policy Framework
- ☑ AWS CloudTrail Logging
- ☑ ELB Listener Security
- ☑ ELB Security Groups
- ☑ CloudFront Custom SSL Certificates in the IAM Certificate Store
- ☑ CloudFront SSL Certificate on the Origin Server
- ☑ Exposed Access Keys

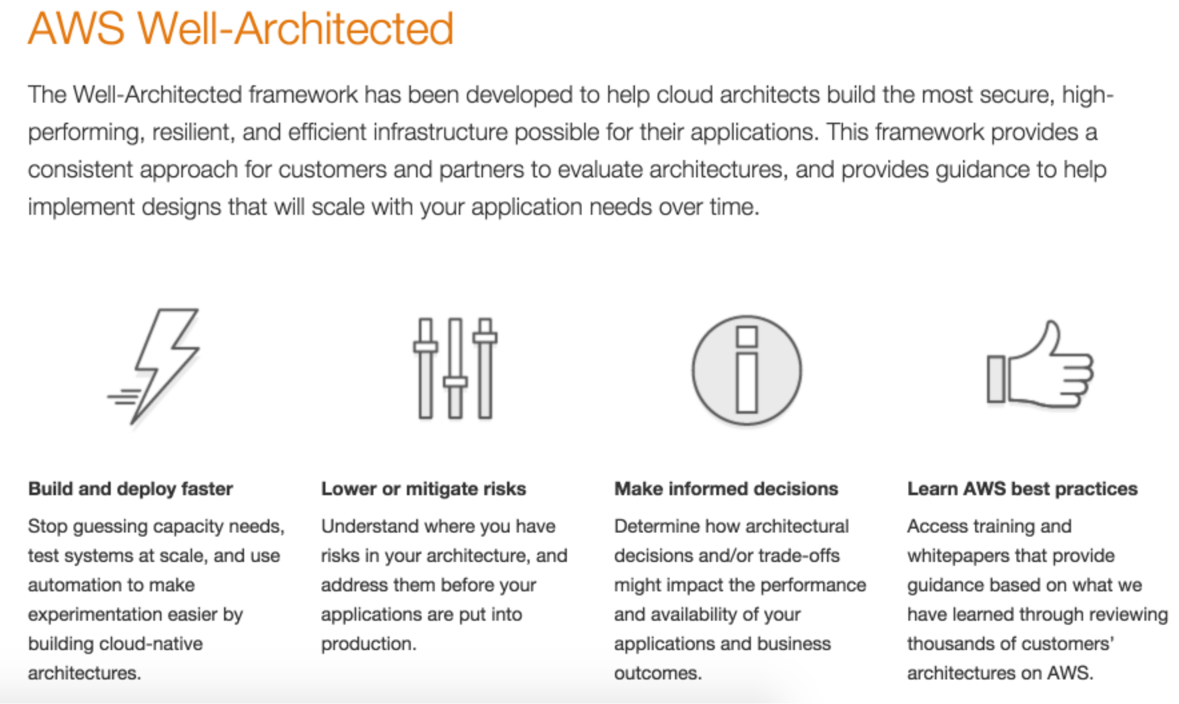
Fault Tolerance

1 ❗ 6 ⚠ 12 ☑ 0 n/a

- ❗ Amazon EBS Snapshots
- ⚠ Amazon EC2 Availability Zone Balance
- ⚠ Amazon S3 Bucket Logging
- ⚠ Amazon S3 Bucket Versioning
- ⚠ AWS Direct Connect Connection Redundancy
- ⚠ AWS Direct Connect Location Redundancy
- ⚠ AWS Direct Connect Virtual Interface Redundancy
- ☑ Load Balancer Optimization
- ☑ VPN Tunnel Redundancy
- ☑ Auto Scaling Group Resources
- ☑ Amazon RDS Backups
- ☑ Amazon RDS Multi-AZ
- ☑ Auto Scaling Group Health Check
- ☑ Amazon Route 53 Name Server Delegations
- ☑ Amazon Route 53 High TTL Resource Record Sets
- ☑ Amazon Route 53 Failover Resource Record Sets
- ☑ Amazon Route 53 Deleted Health Checks
- ☑ ELB Cross-Zone Load Balancing
- ☑ ELB Connection Draining

Resources

https://aws.amazon.com/well-architected/



AWS Well-Architected

The Well-Architected framework has been developed to help cloud architects build the most secure, high-performing, resilient, and efficient infrastructure possible for their applications. This framework provides a consistent approach for customers and partners to evaluate architectures, and provides guidance to help implement designs that will scale with your application needs over time.

Build and deploy faster

Stop guessing capacity needs, test systems at scale, and use automation to make experimentation easier by building cloud-native architectures.

Lower or mitigate risks

Understand where you have risks in your architecture, and address them before your applications are put into production.

Make informed decisions

Determine how architectural decisions and/or trade-offs might impact the performance and availability of your applications and business outcomes.

Learn AWS best practices

Access training and whitepapers that provide guidance based on what we have learned through reviewing thousands of customers' architectures on AWS.

AWS re:Invent

Thank you!

@julsimon

✉ julsimon@amazon.com