
AWS
re:Invent

ARC207

Architecting resiliency, recovery, and reliability: Capital One

Tanusree McCabe

Senior Distinguished Engineer, Architect

Capital One

Don Barber

Principal Solutions Architect

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Why is this important?

Where Capital One was

Defining site reliability engineering (SRE) pillars: Resiliency, recovery, and reliability

Using the 3 R’s to design systems

Where Capital One is today



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why is this important?

According to Gartner, the average cost of IT downtime is $5,600 per minute, or about $300,000 per hour

AWS enables users to meet their resiliency, recovery, and reliability goals with a variety of services and features

Resiliency, recovery, and reliability fall into 2 of the 5 pillars of the AWS Well-Architected Framework (**operational excellence** and **reliability**)

Users often ask for best practices and other advice

Capital One has led the way in many of these best practices

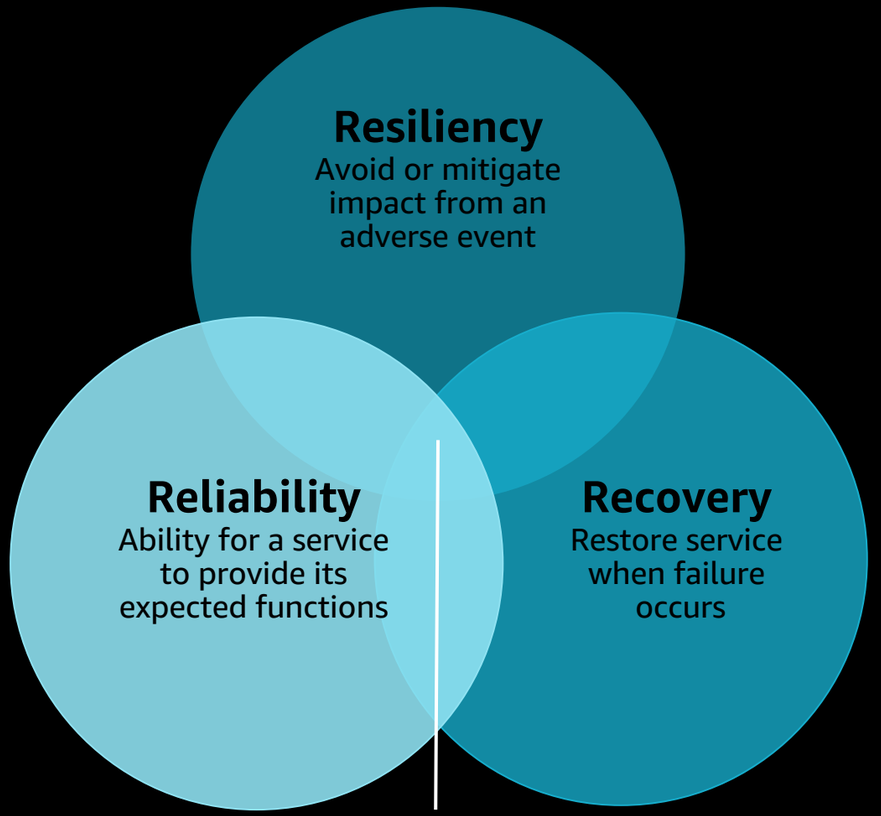
Where Capital One was

Capital One is focused on providing a great customer experience

In 2015, Capital One defined resiliency tiers for on-premises, colocated, and cloud applications to help achieve this focus

In 2018, the resiliency tiers were updated to emphasize high availability using cloud-native capabilities such as scaling and multi-region to orient toward future solutions (cloud)

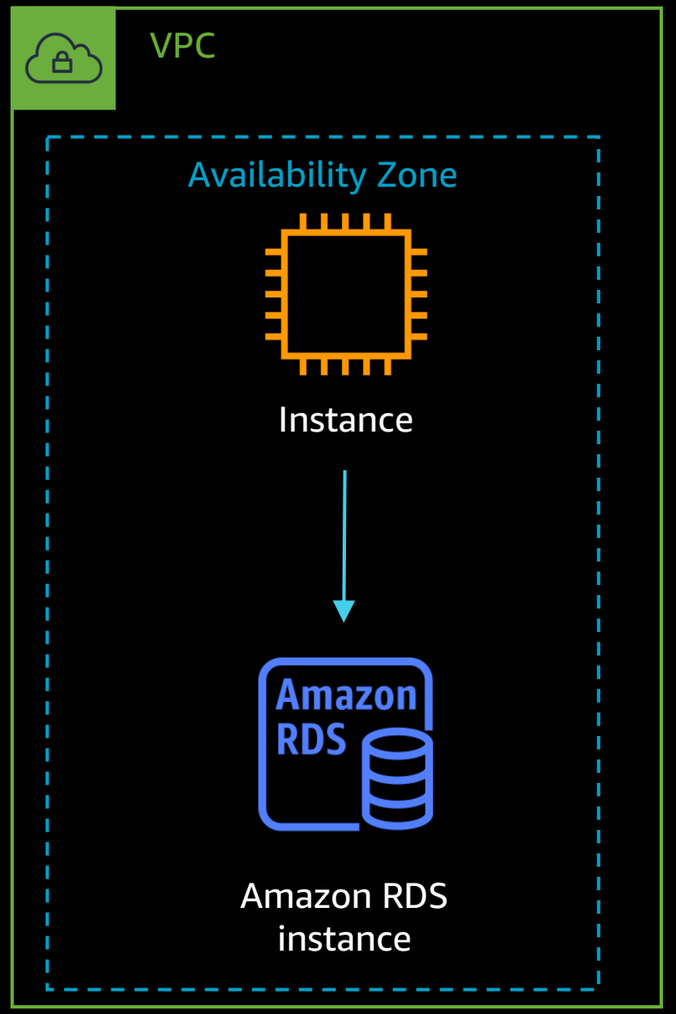
How Capital One defines pillars of SRE



Stability

Business requirements define goals (for example, RTO, RPO, SLO)

Using 3 R's to define system



Apply the resiliency lens: Considerations

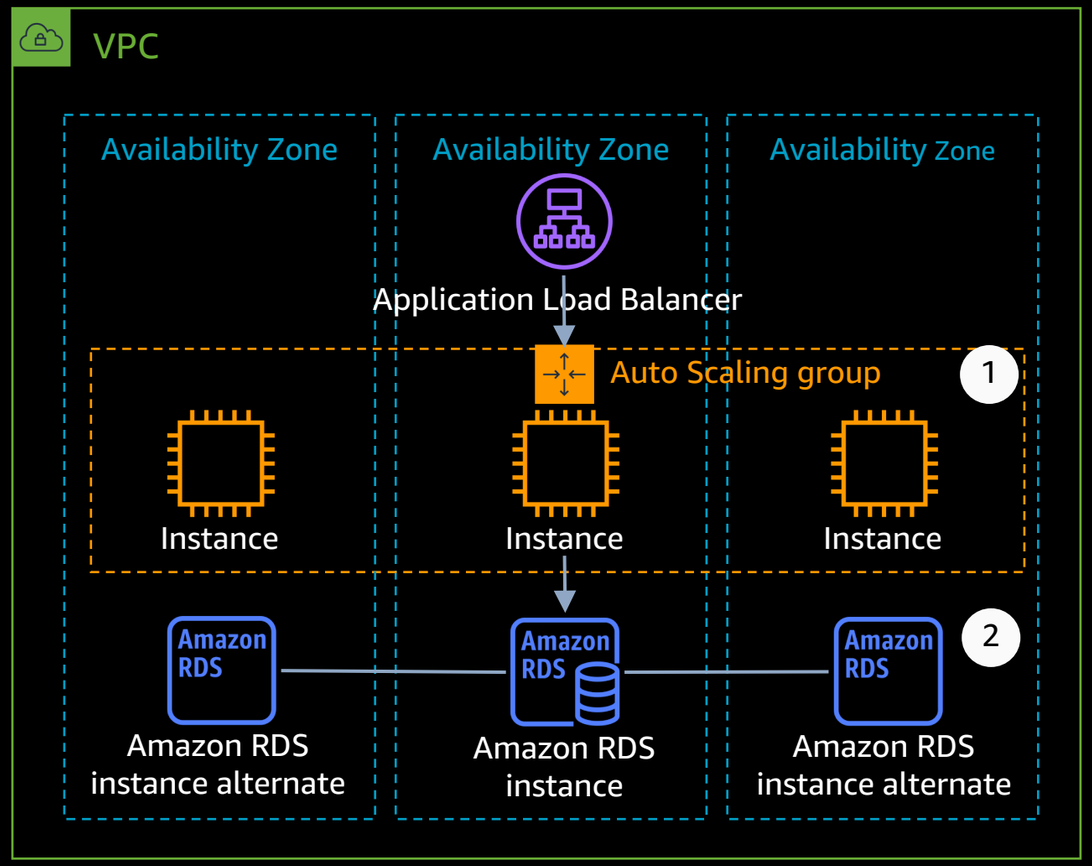
Scaling: Can this application scale horizontally up or down in response to changes?

Redundancy: Are redundancies used to increase availability in the case of an outage?

Capacity: Is there enough capacity in the services used to allow for scale? For example, do the subnets have enough free IP space? Are the AWS API rate limits well off from being hit?

Apply the resiliency lens: Result

Our simple application has now been extended to be “highly available”:



1. Elasticity with auto scaling
2. Multi-AZ data layer with built-in data replication

Apply the recovery lens: Considerations

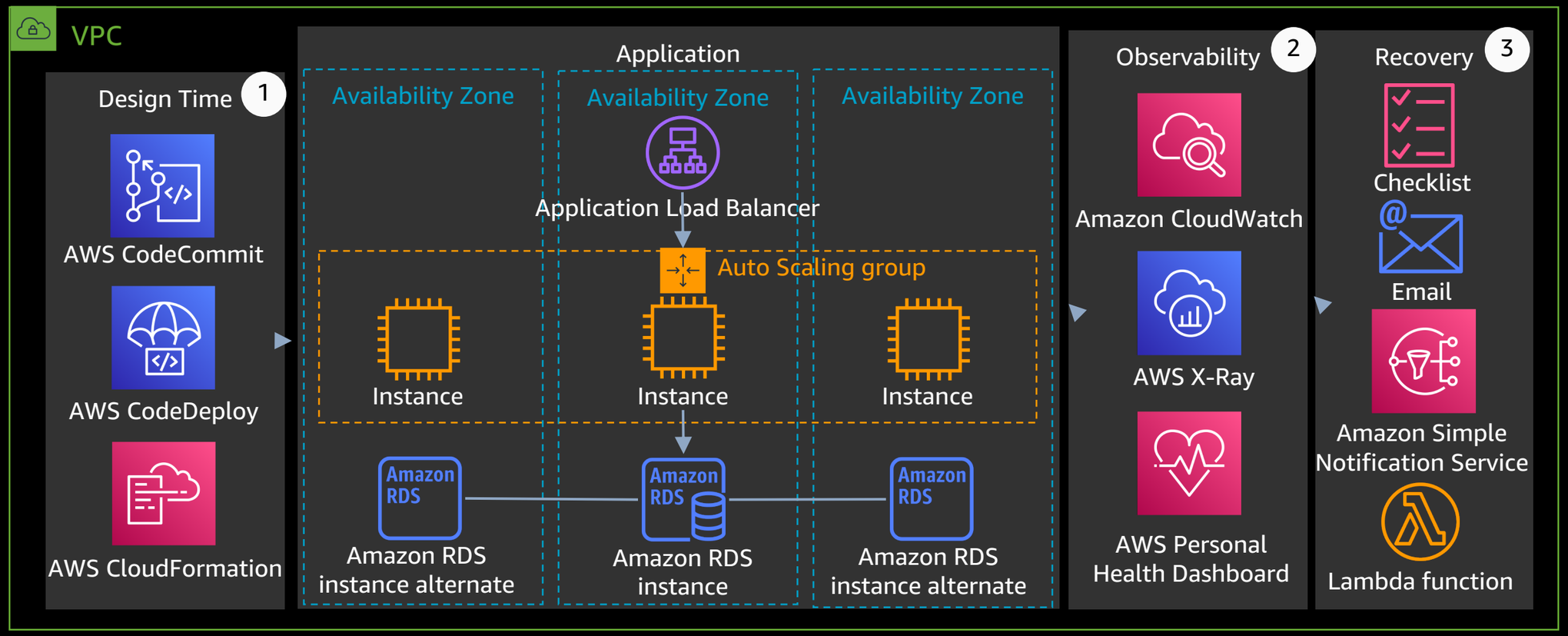
Failure points: What are the possible failure points, and how will we know about them? Which failure points are critical and which are not? What is the probability and impact of that failure?

Recovery stages: What are the stages of recovery such that critical functionality is restored as quickly as possible, with full restoration afterward? What can be done to recover automatically vs. gracefully degrade and perhaps require human intervention?

Strategy and procedures: What is the failover strategy? What are the standard recovery procedures and have they been tested?

Apply the recovery lens: Result

Our highly available application is augmented to support “recovery”:



1. Design time automation
2. Observability via tuned telemetry
3. Recovery defined for manual and automated means

Apply the reliability lens: Considerations

Service level objectives (SLOs): Is this the right level of investment for the expected user need? What are the SLOs for the critical transactions / user journeys? Does that require more redundancy? Is one AWS Region sufficient?

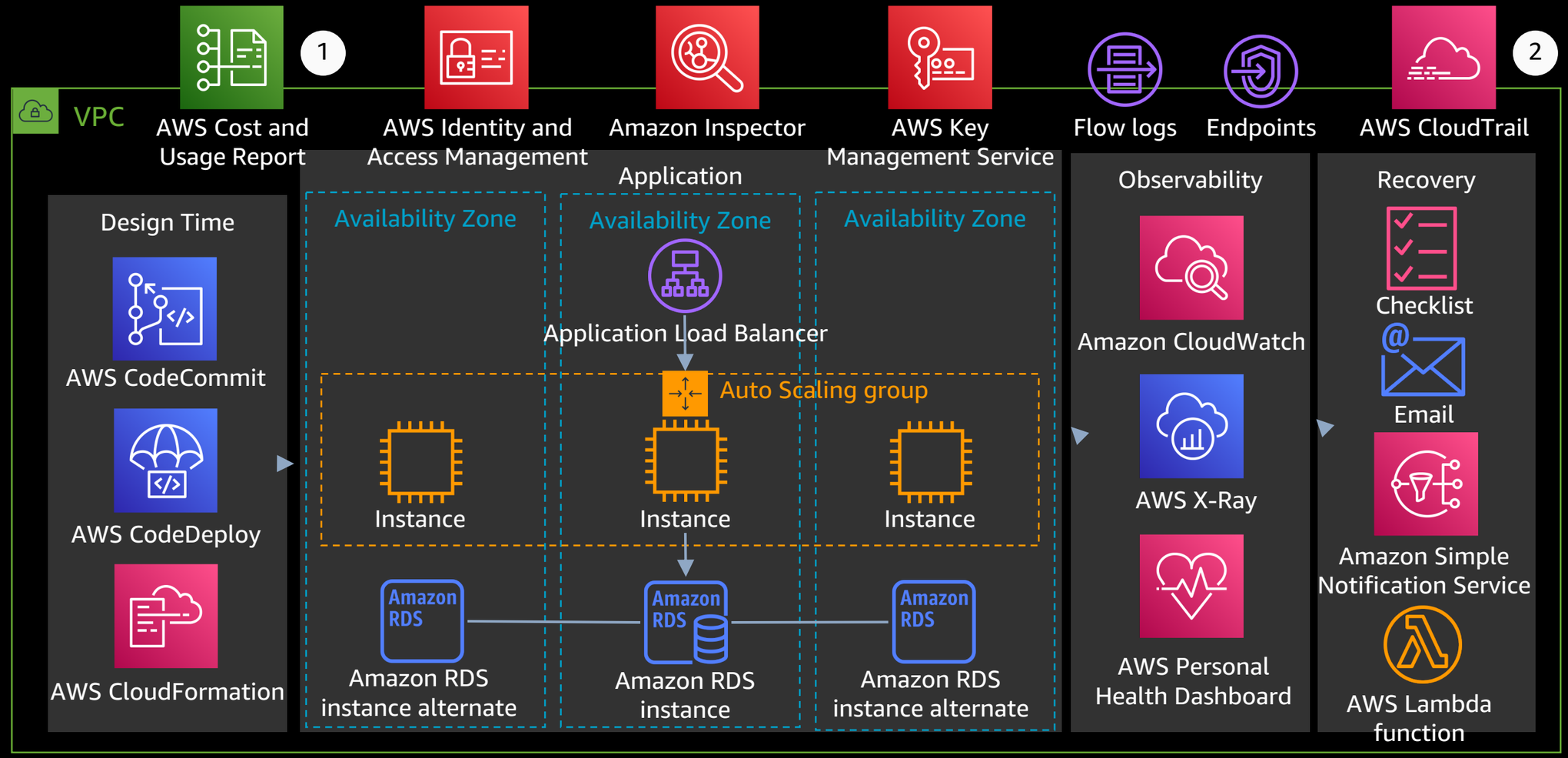
Security: What are the security implications, and how can greater security be achieved?

Performance: What are the performance requirements? Are the resources sized appropriately? Has performance testing occurred?

Cost: Based on user expectations, is the architecture cost-conscious? Have the resources been selected to maximize cost-efficiency?

Apply the reliability lens: Result

Our resilient and recoverable application is augmented to support “reliability”:



1 Cost optimization 2 Security guardrails for preventive and detective controls

Where Capital One is today

In 2020, Capital One is all in on cloud and is steadily incorporating SRE

We are maturing how we apply resiliency, recoverability, and reliability lenses to increase the stability of our internal and end user–facing services

Areas of focus include

- Data layer patterns (for example, resiliency, reliability, durability)
- Dependency management (for example, data freshness, third-party)
- Architecture principles (for example, observability, stability)

Architecture resources

AWS Well-Architected Framework

The official best practices for architecting in the AWS Cloud

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

AWS Well-Architected Labs

Hands-on labs to help you learn, measure, and build using architectural best practices

https://wellarchitectedlabs.com/

AWS Architecture Center

Official AWS repository for all architecture resources

https://aws.amazon.com/architecture/

AWS Solutions Library

Vetted reference implementations and well-architected patterns

https://aws.amazon.com/solutions/

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.