



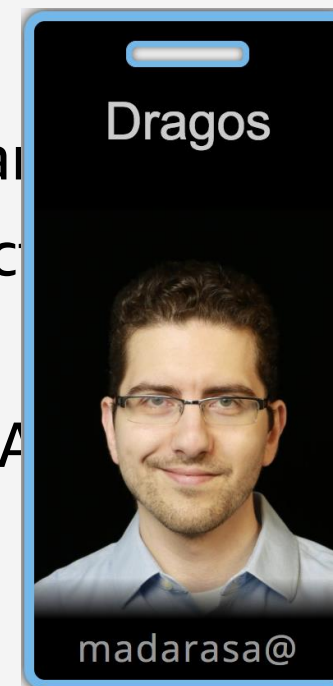
# Are you Well-Architected?

**Dragos Madarasan**  
Solutions Architect, AWS

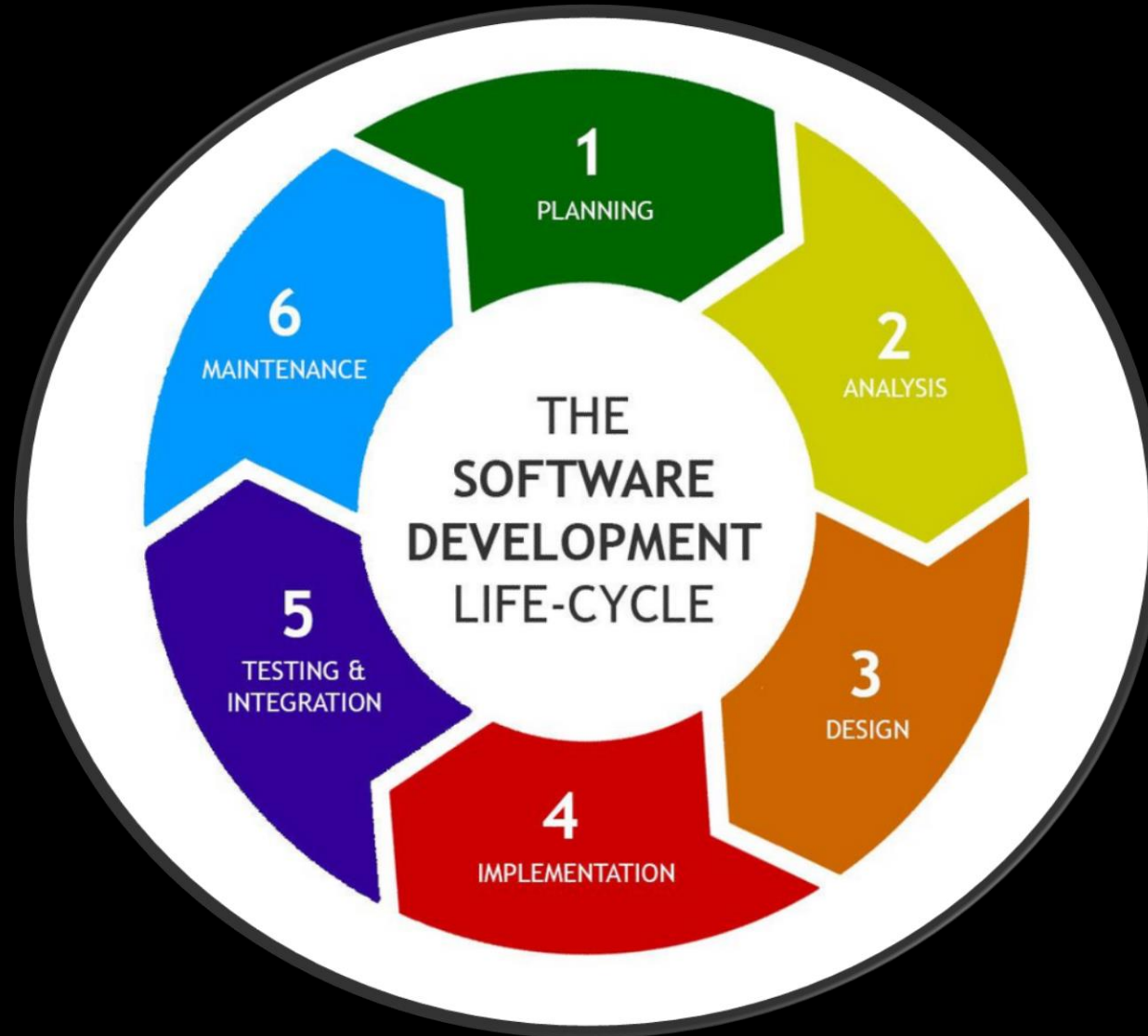


## About me

- Solutions Architect @ AWS, covering Romania and Hungary
- Work with and support clients, partners, NGOs, Public Sector
- Based in Munich, Germany, often in Romania
- Previously Support Engineer and ProServe Consultant @ AWS



# Building an application

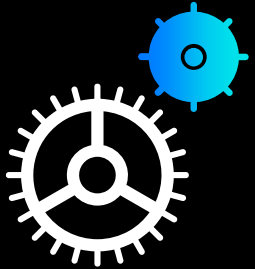


When you look at the system your team is building, can you answer the question:

“Are you Well-Architected?”

---

# Are you Well-Architected?



Operations



Security



Reliability



Performance  
efficiency



Cost  
optimization

# Are you Well-Architected?



Operations



Security



Reliability



Performance  
efficiency



Cost  
optimization



Review  
process



Consistent



Technology  
portfolio



# AWS Well-Architected

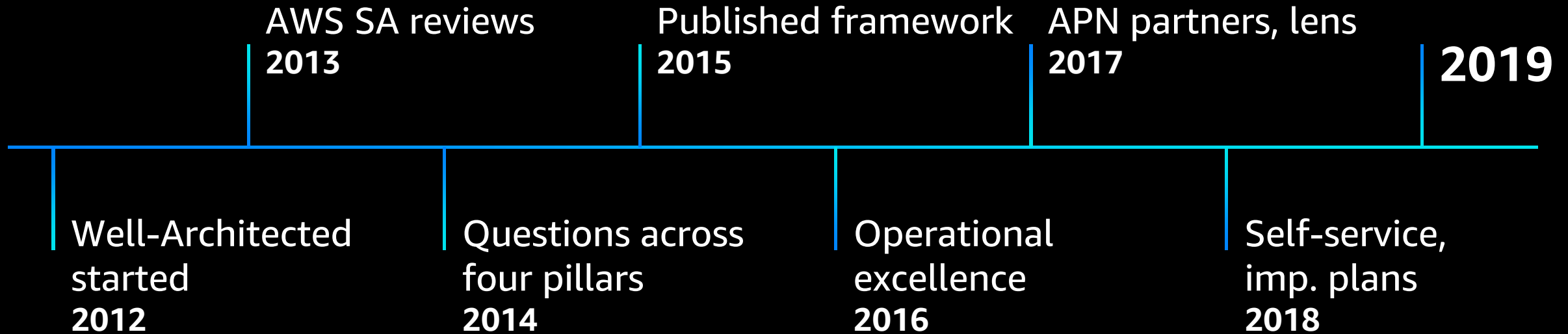
<https://aws.amazon.com/well-architected/>

---

# A little bit of history

---

# History



# AWS Well-Architected Framework

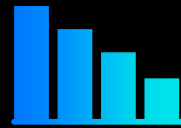


# Why AWS Well-Architected Framework?



Build and deploy faster

---



Lower or mitigate risks

---



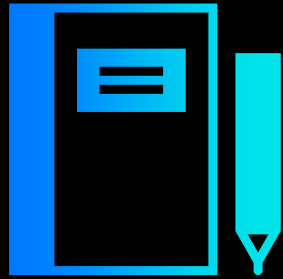
Make informed decisions

---



Learn AWS best practices

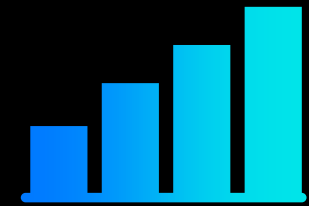
# A mechanism for your cloud journey



Learn



Measure



Improve

# What is the AWS Well-Architected Framework?



Pillars

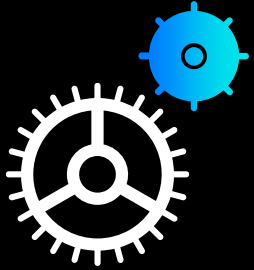


Design principles



Questions

# Pillars of AWS Well-Architected



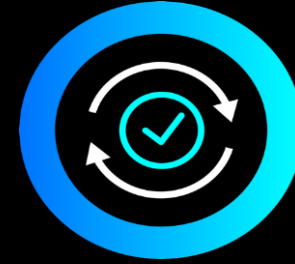
Operational  
excellence



Security



Reliability



Performance  
efficiency



Cost  
optimization

# Design principles



General  
design principles



Pillar-specific  
design principles

Automate responses to security events: Monitor and automatically trigger responses to event-driven, or condition-driven, alerts

# General design principles

Stop guessing your capacity needs

---

Test systems at production scale

---

Automate to make architectural experimentation easier

---

Allow for evolutionary architectures

---

Drive architectures using data

---

Improve through game days

---



# Design principles for security

Implement a strong identity foundation

---

Enable 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

---



# Design principles for reliability

Test recovery procedures

---

Automatically recover from failure

---

Scale horizontally to increase aggregate system availability

---

Stop guessing capacity

---

Manage change in automation

---



# Questions

## Failure management

### REL 7 How does your system withstand component failures?

---

*If your workloads have a requirement, implicit or explicit, for high availability and low mean time to recovery (MTTR), architect your workloads for resiliency and distribute your workloads to withstand outages.*

Best practices:

- **Monitoring is done at all layers of the workload to detect failures:** Continuously monitor the health of your system and report degradation as well as complete failure.
- **Deployed to multiple Availability Zones; Multiple AWS Regions if required:** Distribute workload load across multiple Availability Zones and AWS Regions (for example, DNS, ELB, Application Load Balancer, API Gateway).
- **Has loosely coupled dependencies:** Dependencies such as queuing systems, streaming systems, workflows, and load balancers are loosely coupled.
- **Has implemented graceful degradation:** When a component's dependencies are unhealthy, the component itself does not report as unhealthy. It can continue to serve requests in a degraded manner.
- **Automated healing implemented on all layers:** Use automated capabilities upon detection of failure to perform an action to remediate.
- **Notifications are sent upon availability impacting events:** Notifications are sent upon detection of any significant events, even if it was automatically healed.

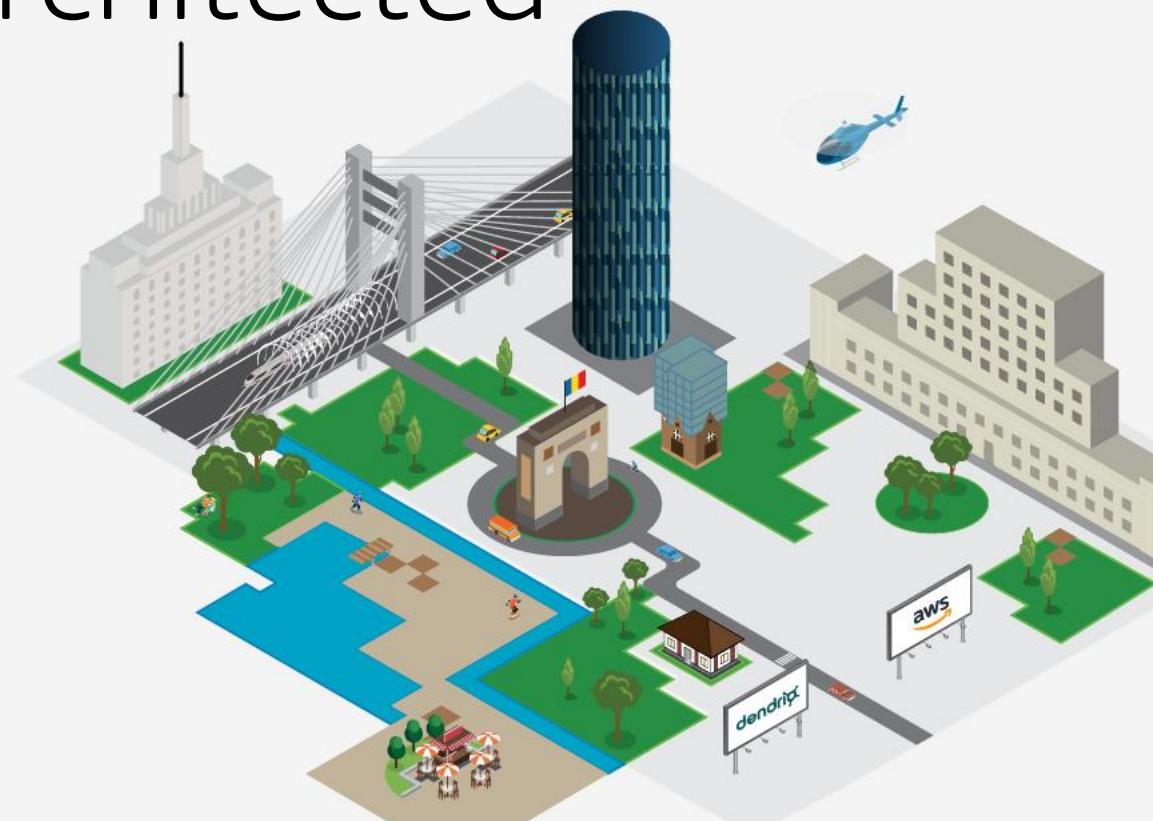
Pillar area

Question

Context

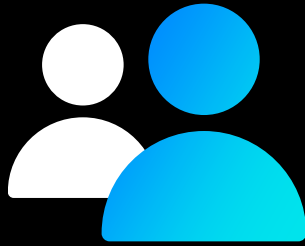
Best practices

# Applying AWS Well-Architected



# Intent of review

Not an audit



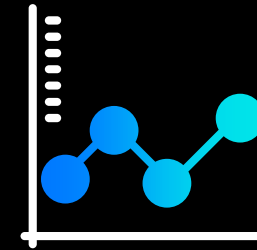
Working together to  
improve

Not architecture astronauts



Pragmatic,  
proven advice

Not a one-time check

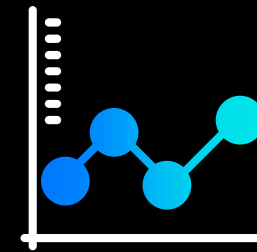


Throughout  
lifecycle

# Not a one-time check

- Read whitepaper before design
  - At minimum: questions
- Iterate through lifecycle
  - Halfway (or sooner)
  - Close to launch
  - Post-launch

Not a one-time check



Throughout  
lifecycle

# Review choice

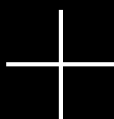
Your team technical  
and business leads

AWS  
WA tool

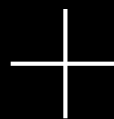
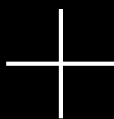
AWS  
APN partner

AWS solutions  
architect

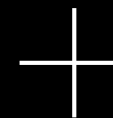
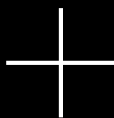
Self-service



Partner



AWS SA



# Review choice

## Self-service

Full control of how often and when

---

## Partner

When you have a shortage of skilled resource, can address issues

---

## AWS SA

Critical workloads, advice on improvement plan

---

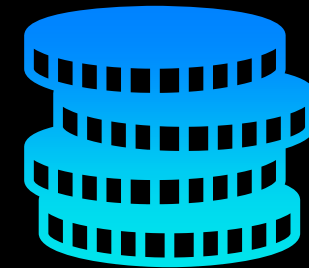
# Partner Well-Architected reviews



Engage with a  
Well-Architected Partner  
for a free review



Results including  
statement of work (SoW)  
for improvements



Approve SoW within  
30 days receive \$5K  
in AWS credits

# Well-Architected Partners

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



# Tips

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

---



## Whitepapers (PDF, Kindle)

- Framework
  - Per pillar (operational excellence, reliability, security, performance efficiency, cost optimization)
  - Lens (serverless, HPC, IoT)
- 



## Training (framework, pillars, review process, tool)

---



## Website

- Glossary
  - Videos
  - Map
-

# Resources



# Framework has Q&A

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



Amazon Web Services AWS Well-Architected Framework

## Appendix: Questions, Answers, and Best Practices

### Operational Excellence

#### Prepare

**OPS 1 How do you determine what your priorities are?**

*Everyone needs to understand their part in enabling business success. Have shared goals in order to set priorities for resources. This will maximize the benefits of your efforts.*

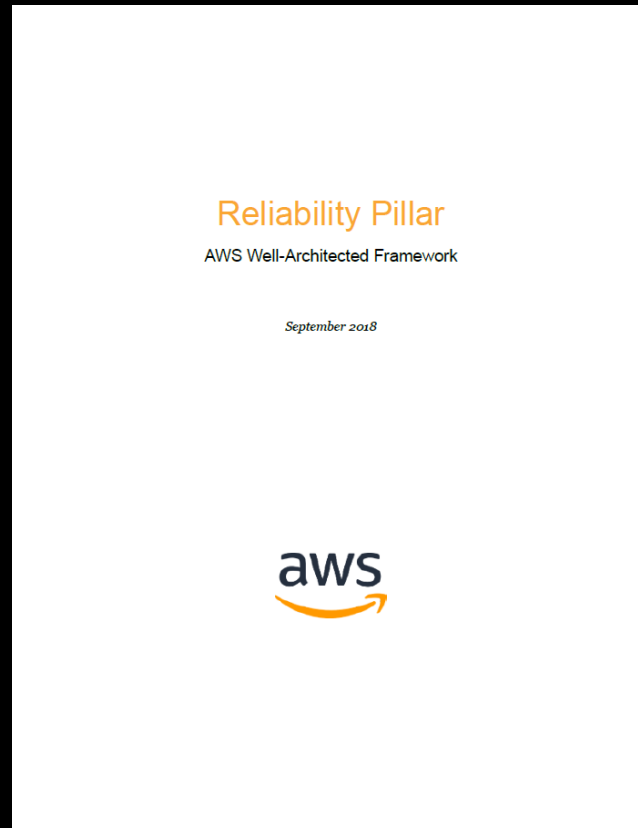
Best practices:

- **Evaluate external customer needs:** Involve key stakeholders, including business, development, and operations teams, to determine where to focus operations efforts on external customer needs. This will ensure that you have a thorough understanding of the operations support that is required to achieve business outcomes.
- **Evaluate internal customer needs:** Involve key stakeholders, including business, development, and operations teams, when determining where to focus operations efforts on internal customer needs. This will ensure that you have a thorough understanding of the operations support that is required to achieve business outcomes.
- **Evaluate compliance requirements:** Evaluate external factors, such as regulatory compliance requirements and industry standards, to ensure that you are aware of guidelines or obligations that may mandate or emphasize specific focus. If no compliance requirements are identified, ensure that you apply due diligence to this determination.
- **Evaluate threat landscape:** Evaluate threats to the business (for example, competition, business risk and liabilities, operational risks, and information security threats), so that you can include their impact when determining where to focus operations efforts.
- **Evaluate tradeoffs:** Evaluate the impact of tradeoffs between competing interests, to help make informed decisions when determining where to focus operations efforts. For example, accelerating speed to market for new features may be emphasized over cost optimization.
- **Manage benefits and risks:** Manage benefits and risks to make informed decisions when determining where to focus operations efforts. For example, it may be beneficial to deploy a system with unresolved issues so that significant new features can be made available to customers.

aws 46

# Pillar has detail

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



## Contents

Introduction

Reliability

Design Principles

Definition

Foundation – Limit Management

Foundation - Networking

Application Design for High Availability

Understanding Availability Needs

Application Design for Availability

Operational Considerations for Availability

Example Implementations for Availability Goals

Dependency Selection

Single Region Scenarios

Multi-Region Scenarios

Conclusion

Contributors

Document Revisions

Appendix A: Designed-For Availability for Select AWS Services

# Lens has reference architecture

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

## Serverless Applications Lens

AWS Well-Architected Framework

November 2018



### Amazon Web Services – AWS Well-Architected Lens – Serverless Applications

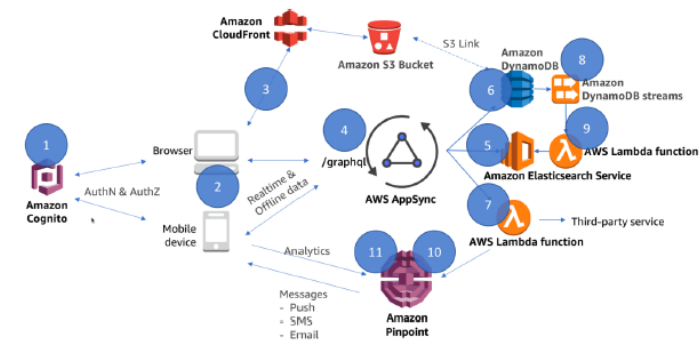


Figure 4: Reference architecture for a mobile backend

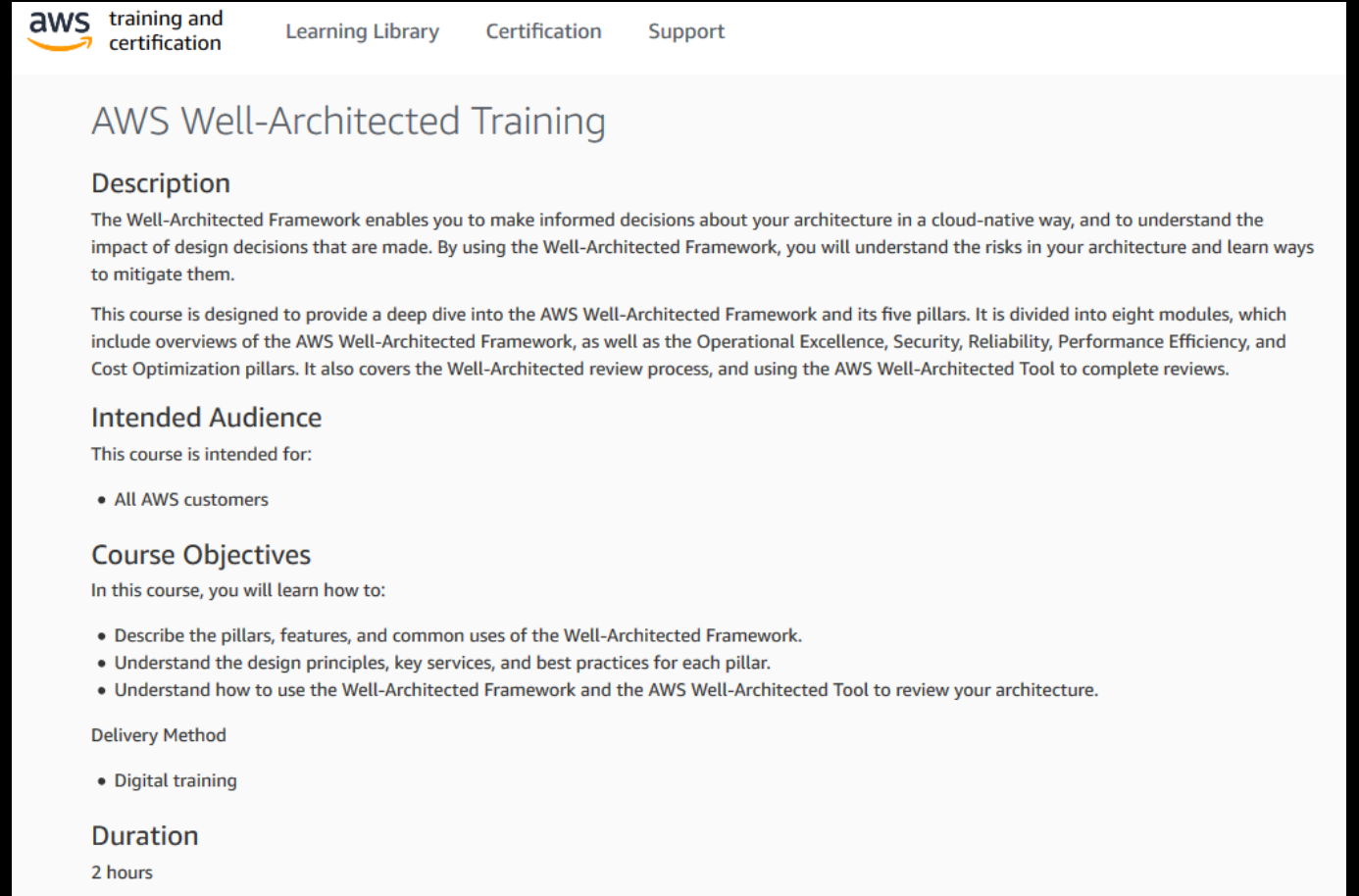
1. **Amazon Cognito** is used for user management and as an identity provider for your mobile application. Additionally, it allows mobile users to leverage existing social identities such as Facebook, Twitter, Google+, and Amazon to sign in.
2. **Mobile users** interact with the mobile application backend by performing GraphQL operations against AWS AppSync and AWS service APIs (for example, Amazon S3 and Amazon Cognito).
3. **Amazon S3** stores mobile application static assets including certain mobile user data such as profile images. Its contents are securely served via CloudFront.
4. **AWS AppSync** hosts GraphQL HTTP requests and responses to mobile users. In this scenario, data from AWS AppSync is real-time when devices are connected, and data is available offline as well. Data sources can be **Amazon DynamoDB**, **Amazon Elasticsearch Service**, or **AWS Lambda functions**.
5. **Amazon Elasticsearch Service** acts as a main search engine for your mobile application as well as analytics.



# Free training

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

The Framework  
Operational Excellence  
Security  
Reliability  
Performance Efficiency  
Cost Optimization  
Well-Architected Review  
AWS Well-Architected Tool



The screenshot shows the AWS training and certification page for the Well-Architected Framework. The page includes a navigation bar with links to Learning Library, Certification, and Support. The main content area is titled 'AWS Well-Architected Training' and contains sections for Description, Intended Audience, Course Objectives, Delivery Method, and Duration.

**aws** training and certification    Learning Library    Certification    Support

## AWS Well-Architected Training

### Description

The Well-Architected Framework enables you to make informed decisions about your architecture in a cloud-native way, and to understand the impact of design decisions that are made. By using the Well-Architected Framework, you will understand the risks in your architecture and learn ways to mitigate them.

This course is designed to provide a deep dive into the AWS Well-Architected Framework and its five pillars. It is divided into eight modules, which include overviews of the AWS Well-Architected Framework, as well as the Operational Excellence, Security, Reliability, Performance Efficiency, and Cost Optimization pillars. It also covers the Well-Architected review process, and using the AWS Well-Architected Tool to complete reviews.

### Intended Audience

This course is intended for:

- All AWS customers

### Course Objectives

In this course, you will learn how to:

- Describe the pillars, features, and common uses of the Well-Architected Framework.
- Understand the design principles, key services, and best practices for each pillar.
- Understand how to use the Well-Architected Framework and the AWS Well-Architected Tool to review your architecture.

### Delivery Method

- Digital training

### Duration

2 hours

# Pillar priority

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

The screenshot shows the 'Well-Architected Tool' interface. On the left is a navigation sidebar with 'Dashboard' and 'Workloads'. The main content area is titled 'Whisky App' and has a breadcrumb trail 'Well-Architected Tool > Workloads > Whisky App'. Below the title are four tabs: 'Review' (selected), 'Improvement plan', 'Milestones', and 'Properties'. A prominent blue box with an information icon contains the message 'Continue review' and 'Your workload review is not complete. To complete your workload review, click the Continue review button.' Below this is a 'Review overview' section with 'Overall status' set to 'Unanswered' and 'Review notes' which is currently empty.

The screenshot shows the 'Edit improvement plan configuration' dialog box. The title is 'Edit improvement plan configuration'. Below the title is a section titled 'Improvement plan configuration'. Underneath, the text reads 'Pillar priority' and 'Put the pillars in order from most to least important for this workload'. There are five rows, each representing a pillar with a drag handle (three horizontal lines) on the left and up/down arrows on the right. The pillars are ordered from top to bottom: Security, Reliability, Operational Excellence, Performance Efficiency, and Cost Optimization. At the bottom right of the dialog are 'Cancel' and 'Save' buttons.

# Well-Architected content website

<https://wa.aws.amazon.com/>

AWS Well-Architected Framework



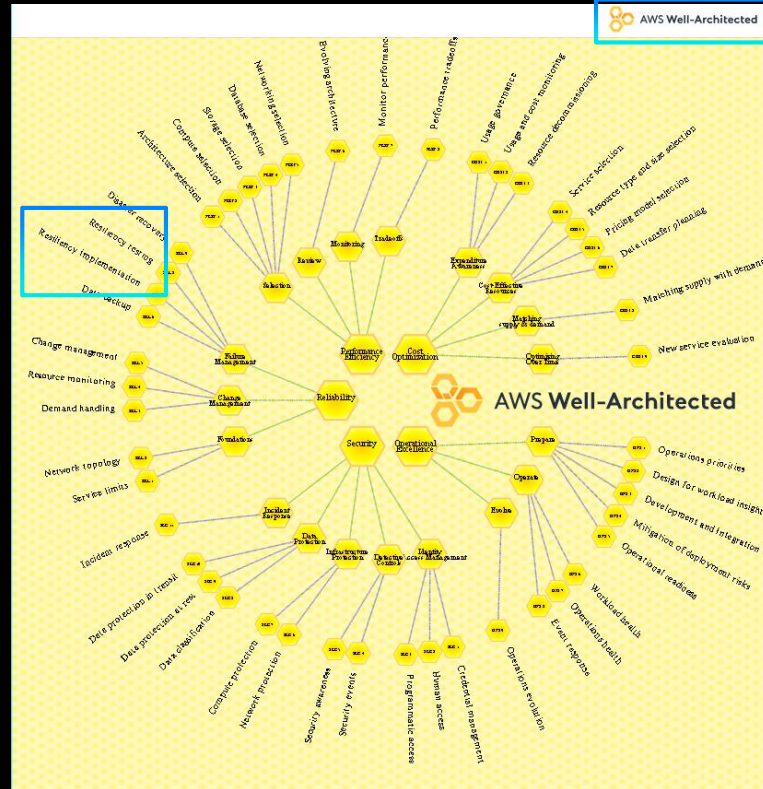
## AWS Well-Architected Framework

### Abstract

This document describes the AWS Well-Architected Framework, which enables you to review and improve your cloud-based architectures and better understand the business impact of your design decisions. We address general design principles as well as specific best practices and guidance in five conceptual areas that we define as the *pillars* of the Well-Architected Framework.

### Contents

- Introduction
  - Definitions
  - On Architecture
  - General Design Principles
- The Five Pillars of the Framework
  - Operational Excellence
  - Security

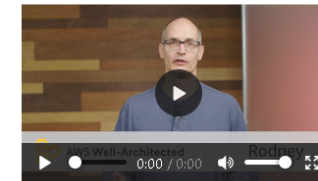


AWS Well-Architected Framework > The Five Pillars of the Framework

> Reliability > How does your system withstand component failures?



**REL 7: How does your system withstand component failures?**  
*If your workloads have a requirement, implicit or explicit, for high availability and low mean time to recovery (MTTR), architect your workloads for resilience and distribute your workloads to withstand outages.*



### Resources

- ▶ [Performing chaos at Netflix scale](#)
- 📄 [AWS global infrastructure](#)
- 📄 [Global tables](#)
- 🎯 [Multiple data center HA network connectivity](#)
- 🛒 [AWS Marketplace: products that can be used for fault tolerance](#)
- 👤 [APN Partner: partners that can help with automation of your](#)

# Well-Architected Labs

<https://github.com/aws-labs/aws-well-architected-labs>

Documentation and code to help you learn, measure, and build using architectural best practices. <http://aws.amazon.com/well-architected>

aws well-architected security lab

71 commits 1 branch 0 releases 5 contributors View license

Branch: master New pull request Find file Clone or download

natbesh COST 200\_4 - added wording around sample files Latest commit 435d62b 12 hours ago

github	Creating initial file from template	9 months ago
Cost	COST 200_4 - added wording around sample files	12 hours ago
Security	New security lab!	7 days ago
CODE_OF_CONDUCT.md	Public release	8 months ago
CONTRIBUTING.md	Public release	8 months ago
LICENSE-Apache	Public release	8 months ago
LICENSE-MITnoAttr	Public release	8 months ago
NOTICE	Public release	8 months ago
README.md	updated base readme, added cost fundamentals 200_3	a month ago

README.md

## AWS Well-Architected Labs

### Introduction

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.

Branch: master aws-well-architected-labs / Security /

Ben Potter New security lab!

- 100 - AWS Account & Root User Updates to sec labs
- 100 - Basic Identity & Access Management User, Group, Role Updates to sec labs
- 200 - Automated Deployment of Detective Controls Update Lab Guide.md
- 200 - Automated Deployment of IAM Groups and Roles Minor updates to IAM
- 200 - Basic EC2 with WAF Protection Formatting updates
- 200 - CloudFront with WAF Protection Major updates incl WAF la
- 300 - IAM Permission Boundaries Delegating Role Creation New security lab!
- README.md New security lab!

README.md

## AWS Well-Architected Security Labs

# Demo

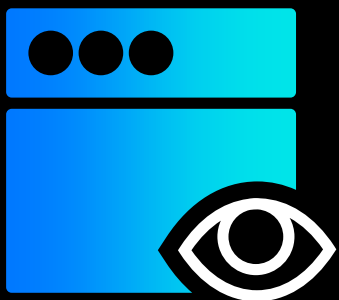


# Best practices

- Read the whitepaper before you start
- Iterate through the questions
  - Identify actions and backlog
- Review pre-launch
- Review/iterate post launch

# Getting started

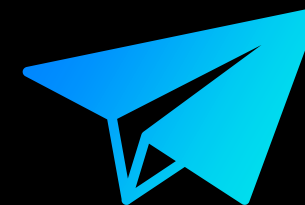
<https://aws.amazon.com/well-architected/>



Read online resources



Account team or partner



Review first workload

# Next step

Start using the tool:

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

# Bucharest AWS User Group



## Other sessions today

**11:00-12:00 Playing Lego with Virtualization components**

**13:00-14:00 Chaos Engineering: why breaking things should be practiced**

**14:00-14:45 Amazon Game Tech: detect anomalies, increase player engagement and optimize costs with Machine Learning**

**15:00-15:45 Advanced traffic routing using AWS Global Accelerator and Amazon Route 53**

**16:00-17:00 How to build a CI/CD pipeline to deploy database changes across multiple accounts**

**13:00-17:00 Solution Architect Associate Certification Exam Preparation (workshop room)**

# Thank you!

**Dragos Madarasan**  
Solutions Architect

