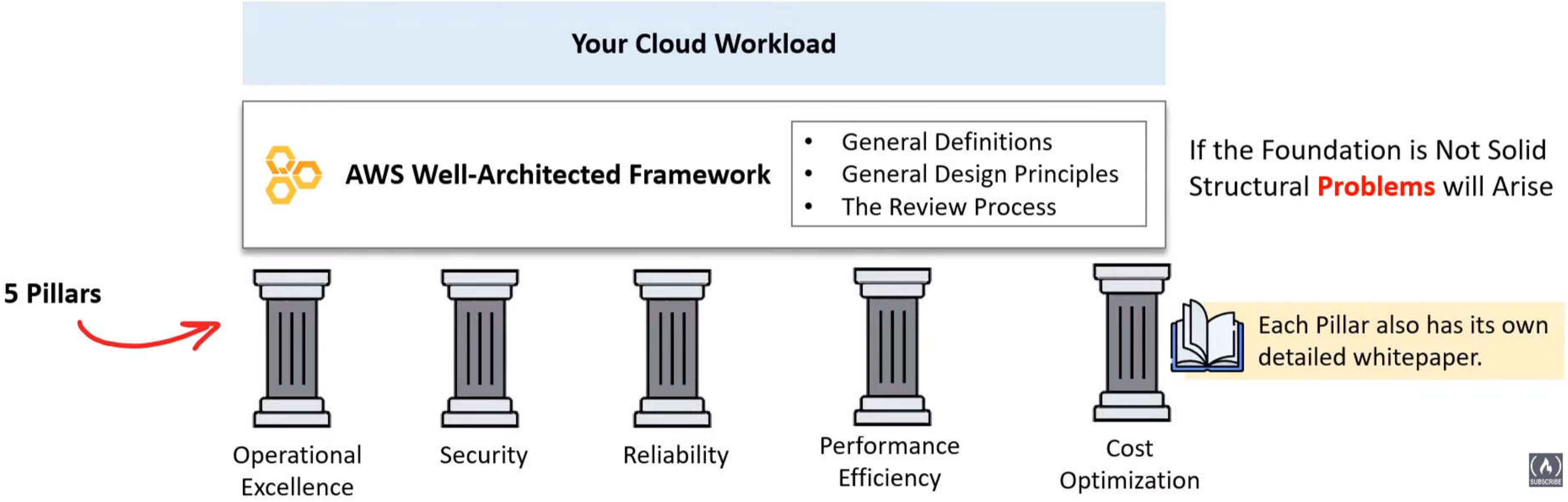
AWS Well-Architected Framework

The AWS Well-Architected Framework is a Whitepaper created by AWS to help customers build using best-practices defined by AWS.

aws.amazon.com/architecture/well-architected

The framework is divided into 5 sections called pillars which address different aspects or "lenses" that can be applied to a cloud workload.



Your Cloud Workload

AWS Well-Architected Framework

- General Definitions
- General Design Principles
- The Review Process

If the Foundation is Not Solid Structural **Problems** will Arise

5 Pillars

Operational Excellence

Security

Reliability

Performance Efficiency

Cost Optimization

Each Pillar also has its own detailed whitepaper.

AWS Well-Architected – General Definitions

*Business Value

Operational Excellent Pillar — Run and monitor systems

Security Pillar — Protect data and systems, mitigate risk

Reliability Pillar — Mitigate and recover from disruptions

Performance Efficiency Pillar — Use computing resources effectively

Cost Optimization Pillar — Get the lowest price

***Trade-Off Pillars Based on Business Context**

General Definitions

Component — Code, Configuration and AWS Resource against a requirement

Workload — A set of components that work together to deliver business value

Milestones — Key changes of your architecture through product life cycle

Architecture — **How** components work together **in a** workload

Technology Portfolio — A collection of workloads required for the business to operate

AWS Well-Architected – On Architecture

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/clf-c01

The AWS Well-Architected Framework is designed around a different kind of team structure.
Enterprises generally have centralized teams with specific roles where AWS has distrubted teams with flexible roles.
Distributed teams can come with new risks, AWS mitigates these with Practices, Mechanisms and Leadership Principles

On-Premise Enterprise

VS

Centralized team consisting of:

- Technical Architect (infrastructure)
- Solution Architect (software)
- Data Architect
- Networking Architect
- Security Architect

Managed by either **TOGAF** or **Zachman Framework**

Amazon Web Services

Distributed teams consisting of:

- Practices
  - Team Experts (Raise the Bar)
- Mechanisms
  - Automated Checks for Standards
- *Amazon Leadership Principle

Supported by a virtual community of **SMEs**, **Principle Engineers**
eg. lunchtime talks - recycled into onboarding material

Amazon Leadership Principles

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/clf-c01

The **Amazon Leadership Principles are a set of principles** used during the company **decision-making, problem-solving, simple brainstorming, and hiring**.

1. Customer Obsession
2. Ownership
3. Invent and Simplify
4. Are Right, A Lot
5. Learn and Be Curious
6. Hire and Develop the Best
7. Insist on the Highest Standards
8. Think Big
9. Bias for Action
10. Frugality
11. Earn Trust
12. Dive Deep
13. Have Backbone; Disagree and Commit
14. Deliver Results
15. Strive to be Earth's Best Employer
16. Success and Scale Bring Broad Responsibility

You can read in detail about all 16 here: https://www.amazon.jobs/en/principles

AWS Well-Architected – General Design Principles

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/clf-c01

Stop guessing your capacity needs

eg. Cloud computing you use as little or much based **on demand.**

Test systems at production scale

eg. Clone production env to testing, Tear down testing not in use to save money.

Automate to make architectural experimentation easier

eg. Using CloudFormation with ChangeSets, StackUpdate and Drift Detection

Allow for evolutionary architectures

eg. CI/CD, rapid or nightly releases, Lambdas deprecating run-times forcing you to evolve

Drive architectures using data

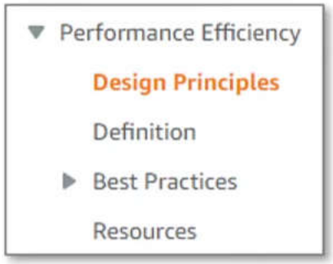
eg. CloudWatch, Cloud Trail automatically turned on collecting data

Improve through game days

eg. simulate traffic on production or purposely kill EC2 instances to see test recovery

AWS Well-Architected – Anatomy of a Pillar

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/clf-c01



Performance Efficiency

A Pillar of the Well-Architected Framework is **structured** as follows:

- Design Principles
  - A list of design principles that need to be considered during implementation
- Definition
  - overview of the best practice categories
- Best Practices
  - detailed information about each best practice with AWS Services
- Resources
  - Additional documentation, whitepapers and videos to implement this pillar

AWS Well-Architected – Design Principles

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/clf-c01

Operational Excellence Design Principles

Perform operations as code
Apply the same engineering discipline you would to application code to your cloud infrastructure.
By treating your operations as code you can limit human error and enable consistent responses to events.
eg. Infrastructure as Code

Make frequent, small, reversible changes
Design workloads to allow components to be updated regulary.
eg. rollbacks, incremental changes, Blue/Green, CI/CD

Refine operations procedures frequently
Look for continuous opportunities to improve your operations
eg. Use game days to simulate traffic or event failure on your production workloads

Anticipate failure
Perform post-mortems on system failures to better improve, write test code, kill production serves to test recovery

Learn from all operational failures
share lessons learned in a knowledge base for operational events and failures across your entire organization

AWS Well-Architected – Design Principles

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/clf-c01

Security Design Principles

Implement a strong identity foundation
Implement Principle of Least Privilege (PoLP). Use Centralized identity. Avoid Long-lived credentials

Enable traceability
Monitor alert and audit actions and changes to your environment in real-time
Integrate log and metric collection and automate investigation and remediation

Apply security at all layers
Take Defense in depth approach with multiple security controls for everything eg. Edge Network, VPC, Load Balancing Instances, OS, Application Code

Automate security best practices

Protect data in transit and at rest

Keep people away from data

Prepare for security events
Incident management systems and investigation policy and processes. Tools to detect, investigate and recover from incidences

AWS Well-Architected – Design Principles

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/clf-c01

Reliability Design Principles

Automatically recover from failure
Monitor Key Performance Indicators (KPIs) and trigger automation when threshold is breached.

Test recovery procedures
Test how your workload fails, and you validate your recovery procedures.
You can use automation to simulate different failures or to recreate scenarios that led to failures before.

Scale horizontally to increase aggregate system availability
Replace one large resource with multiple small resources to reduce the impact of a single failure on the overall workload.
Distribute requests across multiple, smaller resources to ensure that they don't share a common point of failure.

Stop guessing capacity
In on-premise it takes a lot of guess work to determine the elasticity of your workload demands.
With Cloud you don't need to guess how much you need because you can request the right size of resources on-demand.

Manage change in automation
Making changes via Infrastructure as Code, will allow for a formal process to track and review infrastructure

AWS Well-Architected – Design Principles

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/clf-c01

Performance Efficiency Design Principles

Democratize advanced technologies:
Focus on product development rather than procurement, provisioning and management of services.
Take advantage of advanced technology specialized and optimized for your use-case with on-demand cloud services.

Go global in minutes
Deploying your workload in multiple AWS Regions around the world allows you to provide lower latency and a better experience for your customers at minimal cost.

Use serverless architectures:
Serverless architectures remove the need for you to run and maintain physical servers for traditional compute activities.
Removes the operational burden of managing physical servers, and can lower transactional costs because managed services operate at cloud scale.

Experiment more often:
With virtual and automatable resources, you can quickly carry out comparative testing using different types of instances, storage, or configurations.

Consider mechanical sympathy
Understand how cloud services are consumed and always use the technology approach that aligns best with your workload goals. For example, consider data access patterns when you select database or storage approaches.

AWS Well-Architected – Design Principles

Cost Optimization Design Principles

Implement Cloud Financial Management:
Dedicate time and resources to build capability Cloud Financial Management and Cost Optimization tooling.

Adopt a consumption model
Pay only for the computing resources that you require and increase or decrease usage depending on business requirements

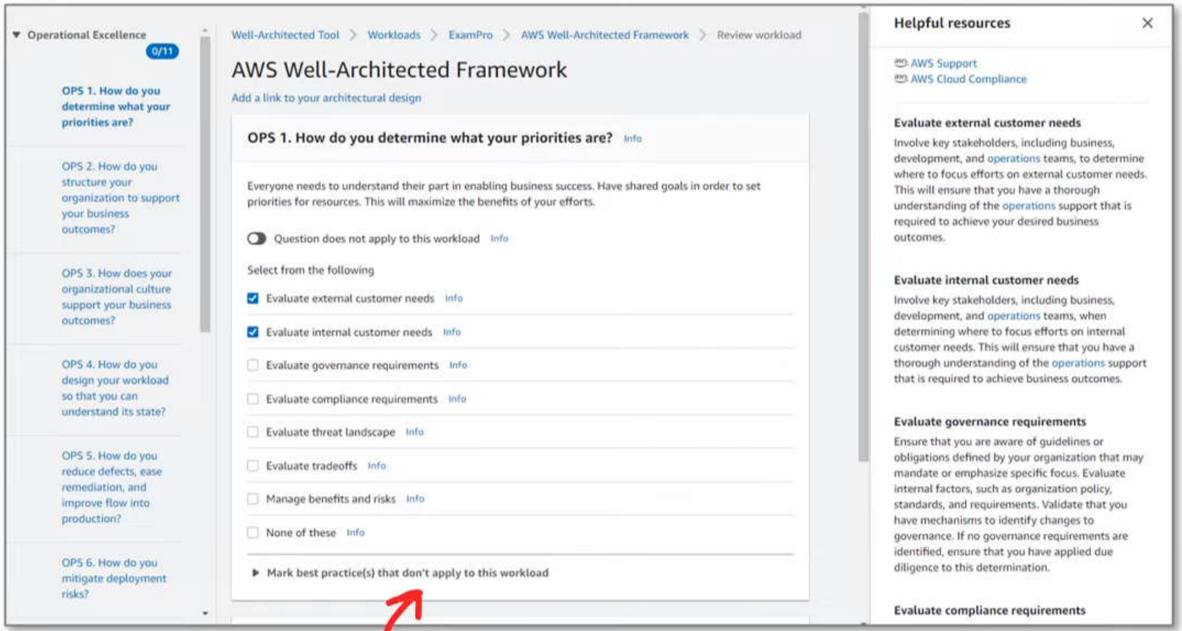
Measure overall efficiency
Measure the business output of the workload and the costs associated with delivering it.
Use this measure to know the gains you make from increasing output and reducing costs.

Stop spending money on undifferentiated heavy lifting
AWS does the heavy lifting of data center operations like racking, stacking, and powering servers.
It also removes the operational burden of managing operating systems and applications with managed services.
This allows you to focus on your customers and business projects rather than on IT infrastructure.

Analyze and attribute expenditure
The cloud makes it easier to accurately identify the usage and cost of systems, which then allows transparent attribution of IT costs to individual workload owners. This helps measure return on investment (ROI) and gives workload owners an opportunity to optimize their resources and reduce costs.

AWS Well-Architected Tool

The Well-Architected Tool is **an auditing tool** to be used to asset your cloud workloads for alignment with the AWS Well Architected Framework.



▼ Operational Excellence 0/11

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

OPS 2. How do you structure your organization to support your business outcomes?

OPS 3. How does your organizational culture support your business outcomes?

OPS 4. How do you design your workload so that you can understand its state?

OPS 5. How do you reduce defects, ease remediation, and improve flow into production?

OPS 6. How do you mitigate deployment risks?

Well-Architected Tool > Workloads > ExamPro > AWS Well-Architected Framework > Review workload

AWS Well-Architected Framework

Add a link to your architectural design

OPS 1. How do you determine what your priorities are? Info

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.

○ Question does not apply to this workload Info

Select from the following

☑ Evaluate external customer needs Info

☑ Evaluate internal customer needs Info

□ Evaluate governance requirements Info

□ Evaluate compliance requirements Info

□ Evaluate threat landscape Info

□ Evaluate tradeoffs Info

□ Manage benefits and risks Info

□ None of these Info

▶ Mark best practice(s) that don't apply to this workload

Helpful resources ×

AWS Support
AWS Cloud Compliance

Evaluate external customer needs

Involve key stakeholders, including business, development, and operations teams, to determine where to focus efforts on external customer needs. This will ensure that you have a thorough understanding of the operations support that is required to achieve your desired business outcomes.

Evaluate internal customer needs

Involve key stakeholders, including business, development, and operations teams, when determining where to focus 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 governance requirements

Ensure that you are aware of guidelines or obligations defined by your organization that may mandate or emphasize specific focus. Evaluate internal factors, such as organization policy, standards, and requirements. Validate that you have mechanisms to identify changes to governance. If no governance requirements are identified, ensure that you have applied due diligence to this determination.

Evaluate compliance requirements

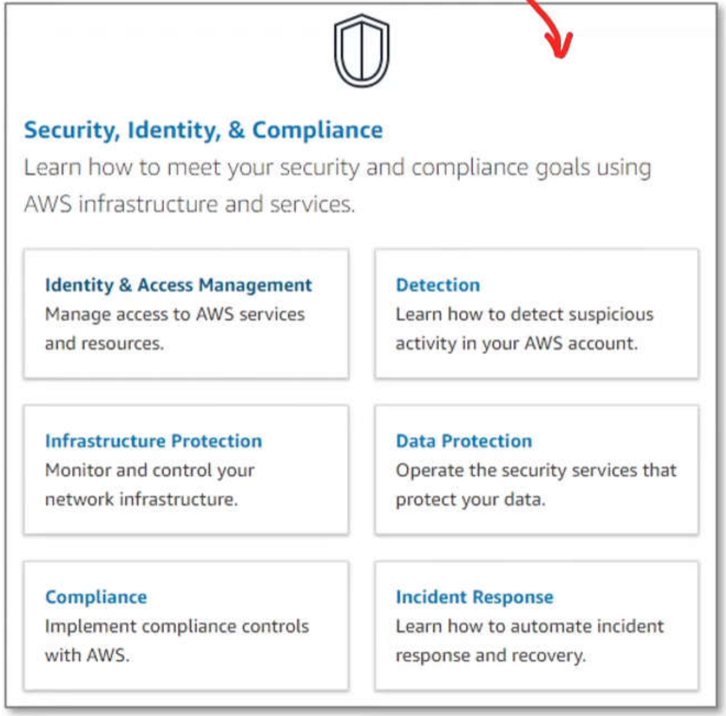
Its essentially **a checklist**, with nearby references to help you assemble a report to share with executives and key stake-holders

AWS Architecture Center

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/clf-c01

The AWS Architecture Center is a web-portal that contains **best practices** and **reference architectures** for a variety of different workloads.

aws.amazon.com/architecture



Security, Identity, & Compliance

Learn how to meet your security and compliance goals using AWS infrastructure and services.

Identity & Access Management
Manage access to AWS services and resources.

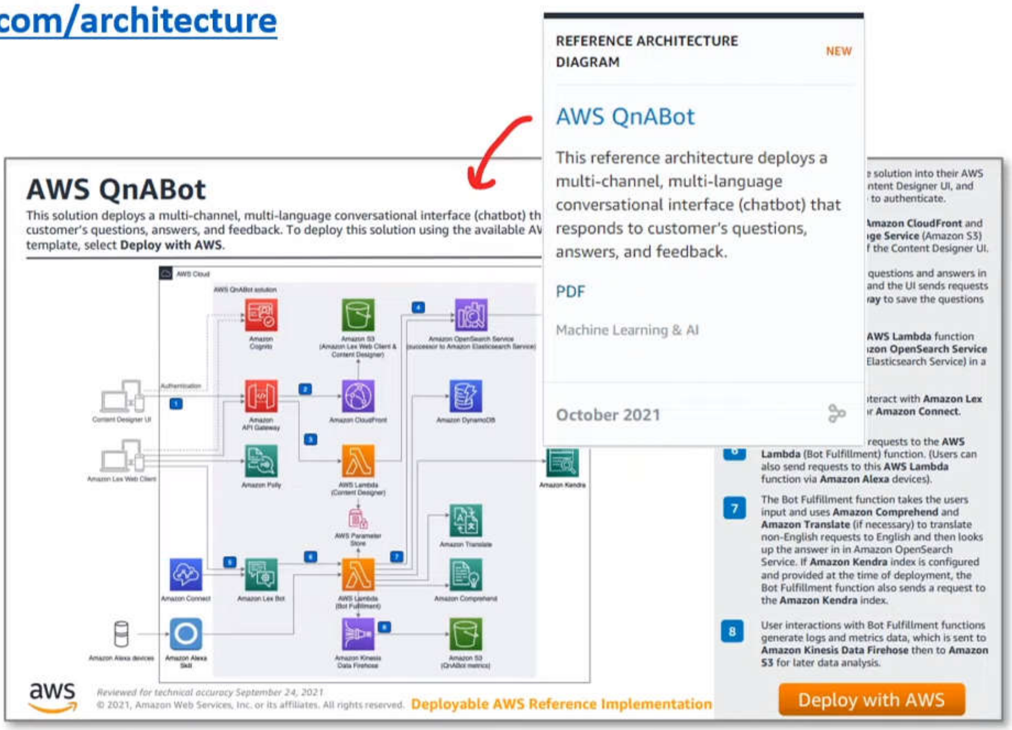
Detection
Learn how to detect suspicious activity in your AWS account.

Infrastructure Protection
Monitor and control your network infrastructure.

Data Protection
Operate the security services that protect your data.

Compliance
Implement compliance controls with AWS.

Incident Response
Learn how to automate incident response and recovery.



REFERENCE ARCHITECTURE DIAGRAM NEW

AWS QnABot

This reference architecture deploys a multi-channel, multi-language conversational interface (chatbot) that responds to customer's questions, answers, and feedback.

PDF

Machine Learning & AI

October 2021

AWS QnABot

This solution deploys a multi-channel, multi-language conversational interface (chatbot) th
customer's questions, answers, and feedback. To deploy this solution using the available AV
template, select **Deploy with AWS**.

e solution into their AWS
ntent Designer UI, and
to authenticate.

Amazon CloudFront and
ge Service (Amazon S3)
f the Content Designer UI.

questions and answers in
and the UI sends requests
ay to save the questions

AWS Lambda function
zon OpenSearch Service
Elasticsearch Service) in a

teract with **Amazon Lex**
r **Amazon Connect**.

6 requests to the **AWS Lambda** (Bot Fulfillment) function. (Users can also send requests to this **AWS Lambda** function via **Amazon Alexa** devices).

7 The Bot Fulfillment function takes the users input and uses **Amazon Comprehend** and **Amazon Translate** (if necessary) to translate non-English requests to English and then looks up the answer in in Amazon OpenSearch Service. If **Amazon Kendra** index is configured and provided at the time of deployment, the Bot Fulfillment function also sends a request to the **Amazon Kendra** index.

8 User interactions with Bot Fulfillment functions generate logs and metrics data, which is sent to **Amazon Kinesis Data Firehose** then to **Amazon S3** for later data analysis.

aws *Reviewed for technical accuracy September 24, 2021*
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Deployable AWS Reference Implementation

Deploy with AWS