

# What is the Well-Architected Framework and why does it matter?

AWS CLOUD CONCEPTS

Hatim Khouzaimi

Data Scientist



# What is the Well-Architected Framework?

- Result of **AWS experts' experience** with many customers and partners
- Consistent approach to evaluate:
  - IT architectures on AWS
  - Organization around architectures
- Implement designs that **maximize the value of the cloud** and scale over time
  - Structured around 6 pillars:
    - Operational excellence
    - Security
    - Reliability
    - Performance efficiency
    - Cost optimization
    - Sustainability

# What is a workload

- **Component:**
  - **Code** (e.g. python code for Lambda), **configuration** (e.g. template file for CloudFormation) and **AWS resources** (e.g. RDS database)
  - Executes certain tasks to satisfy a specific requirement
  - Independent unit, decoupled from other components
- **Workload:**
  - Set of components
  - Deliver business value
    - Ex: Python on Lambda and RDS database, both deployed through a CloudFormation template file, to **process clients' orders faster**
  - Level of detail where communication between technology and business leaders happen

# Operational excellence



- Run workloads effectively
- Gain insights into their operations
- Continuously improve processes to deliver business value

<sup>1</sup> People vector created by pch.vector (<https://www.freepik.com/vectors/people>)

# Security

- Leverage cloud technologies to protect:
  - data
  - systems
  - assets
- Improve companies' security posture



<sup>1</sup> Business vector created by jcomp (<https://www.freepik.com/vectors/business>)

# Reliability

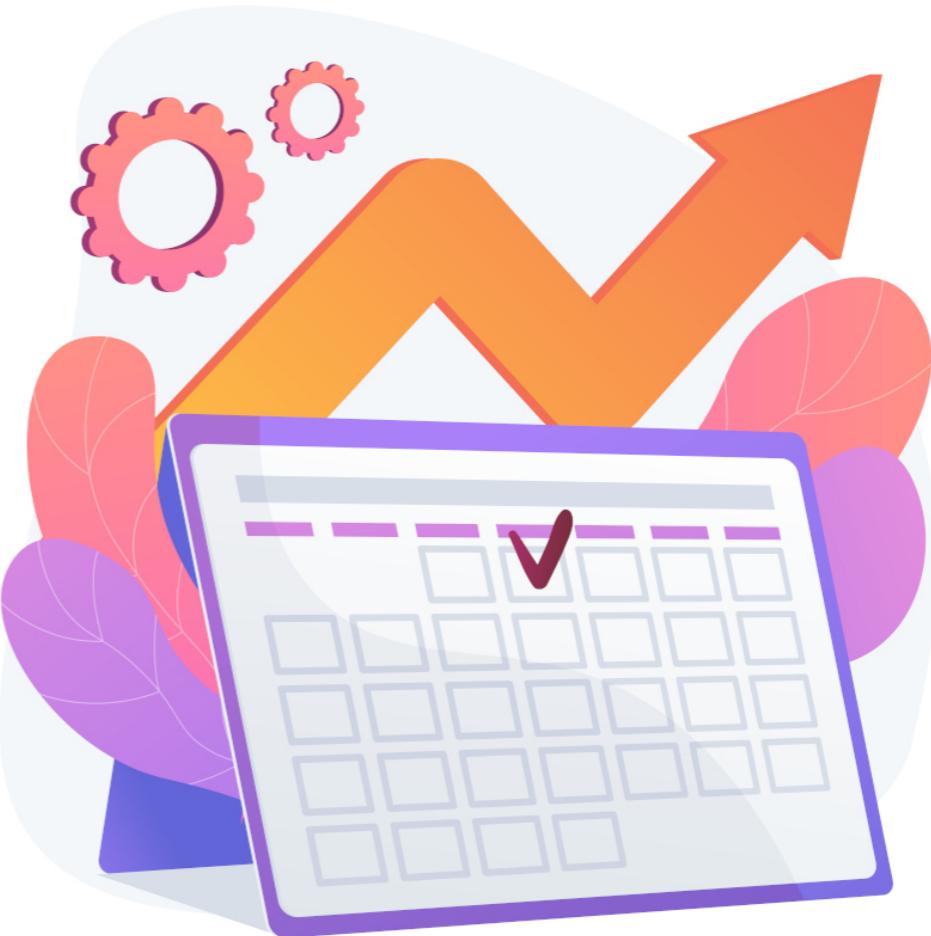


- Ability of workloads to correctly perform tasks when intended
- Administrators and developers ability to:
  - Operate workloads
  - Perform functional testing
  - Throughout the whole workload lifecycle

<sup>1</sup> Hand vector created by macrovector (<https://www.freepik.com/vectors/hand>)

# Performance efficiency

- Efficient usage of resources to meet system requirements
- Maintaining efficiency as:
  - Demand changes
  - Technology evolves



<sup>1</sup> Calendar vector created by vectorjuice (<https://www.freepik.com/vectors/calendar>)

# Cost optimization



- Cost-aware workloads
- Continuous ROI improvement:
  - Cost minimization while achieving business objectives

<sup>1</sup> Business vector created by studiogstock (<https://www.freepik.com/vectors/business>)

# Sustainability

- Understanding and quantifying the **environmental impact** of workload
  - Energy consumption
  - Carbon emissions
- Implementing design principles and best practices to **reduce this impact**



<sup>1</sup> Green vector created by freepik (<https://www.freepik.com/vectors/green>)

# The benefits of the Well-Architected Framework

- Accelerate the build and deploy process
- Better risk management:
  - Mitigate risks before problems arise
  - Quickly fix issues when they happen
- Cloud-native applications:
  - On the cloud since development
  - Faster deployment and scaling
- Properly assess technology evolution impact
- Constantly evolving framework:
  - Continuous improvement mindset

# Pillar structure

- Areas of focus
  - Best practices
  - Design patterns
  - AWS resources and services
- 
- **Note:** Some best practices can serve several pillars

# **Let's practice!**

**AWS CLOUD CONCEPTS**

# Operational excellence and sustainability

AWS CLOUD CONCEPTS



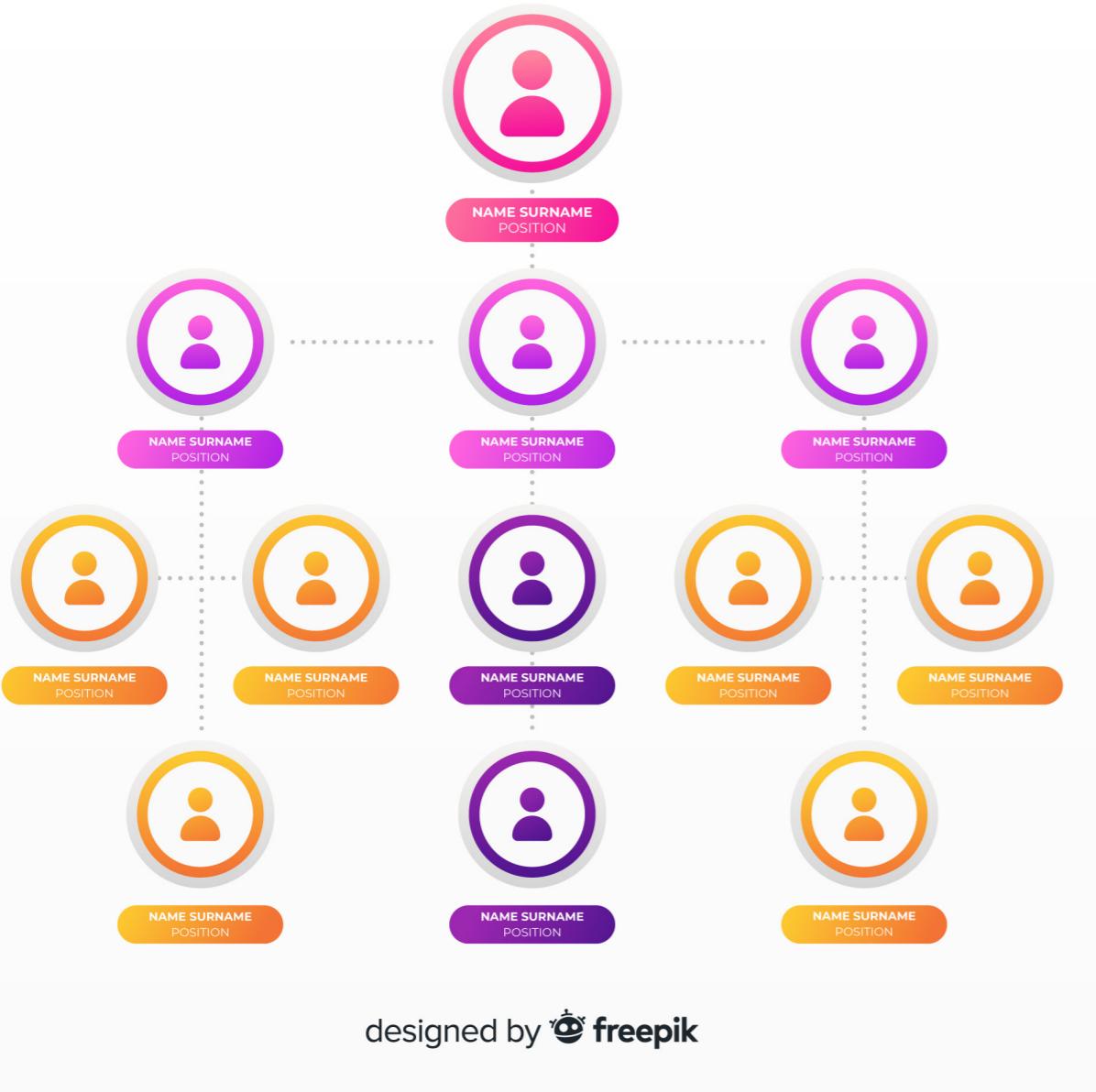
Hatim Khouzaimi

Data Scientist

# Operational excellence focus areas

- Organization
- Prepare
- Operate
- Evolve

# Organization



- **Clear objectives and priorities** based on:
  - Internal and external customers' needs
  - Leadership team's requirements
  - Compliance requirements
  - Threat landscape
- **Shared understanding** of business goals between teams
- Good understanding of each **team's role**
- Encourage teams to **experiment**, take risks and **escalate** concerns

<sup>1</sup> Infographic vector created by pikisuperstar (<https://www.freepik.com/vectors/infographic>)

# Prepare

- Design your workloads to emit information about their internal state
- Facilitate changes into production:
  - Version control
  - Test and validation automation
- Plan for recovery after unsuccessful change
- Processes to assess when a workload is production-ready



<sup>1</sup> Education vector created by vectorjuice (<https://www.freepik.com/vectors/education>)

# Operate



- Define clear KPIs and metrics related to:
  - Workload health (e.g. error rate, response time)
  - Operations health (e.g. successful vs. failed deployments)
- Regularly collect and analyze them
- Set alerts when KPIs are at risk or when anomalies arise
- Have a process for each alert
- Automate responses to events

<sup>1</sup> Abstract vector created by vectorjuice (<https://www.freepik.com/vectors/abstract>)

# Evolve

- Constantly **analyze** operations
- **Learn** from failures:
  - Post-incident analysis
- Document and share learning experiences
  - Feedback loops
- Proactively allocate time to continuously **improve and adapt** your processes



<sup>1</sup> Business vector created by jcomp (<https://www.freepik.com/vectors/business>)

# Some AWS resources for operational excellence

- AWS Compliance
- AWS Trusted Advisor
- AWS Organizations
- AWS Control Tower
- AWS Service Catalog
- AWS Partner Network (APN)
- AWS Systems Manager
- Amazon CloudWatch
- Amazon EventBridge
- AWS X-Ray
- AWS CloudFormation
- AWS CodeBuild
- AWS CodePipeline
- AWS CodeDeploy
- AWS Config
- AWS Personal Health Dashboard

# Design principles for operational excellence

1. Perform **operations as code**
2. Make **small, frequent, reversible** changes
3. **Refine** operations procedures **frequently**
4. **Anticipate** failure
5. **Learn from all operational failures**

# Sustainability focus areas

- Region selection
- User behavior patterns
- Software and architecture patterns
- Data patterns
- Hardware patterns
- Development and deployment process

# Region selection



- Include sustainability factors in your choice of regions:
  - Some regions are near **renewable energy** projects
  - Some regions publish lower **carbon intensity** reports than others
- Check the following link:  
<https://sustainability.aboutamazon.com/all-the-globe>

<sup>1</sup> Business vector created by macrovector\_official (<https://www.freepik.com/vectors/business>)

# User behavior patterns

- Minimize unused infrastructure by adapting to user load
- Incorporate **sustainability goals** in your **Service-Level Agreements (SLAs)**
- Reduce the distance network traffic must travel by **adapting geographic placement** to user locations



<sup>1</sup> Mobile website vector created by vectorjuice (<https://www.freepik.com/vectors/mobile-website>)

# Software and architecture patterns

- **Code optimization** to lower time and resource usage
- Remove or refactor idle or low usage components and workloads
- Use technologies and software patterns that minimize data processing and storage requirements



<sup>1</sup> Business vector created by vectorjuice (<https://www.freepik.com/vectors/business>)

# Data patterns

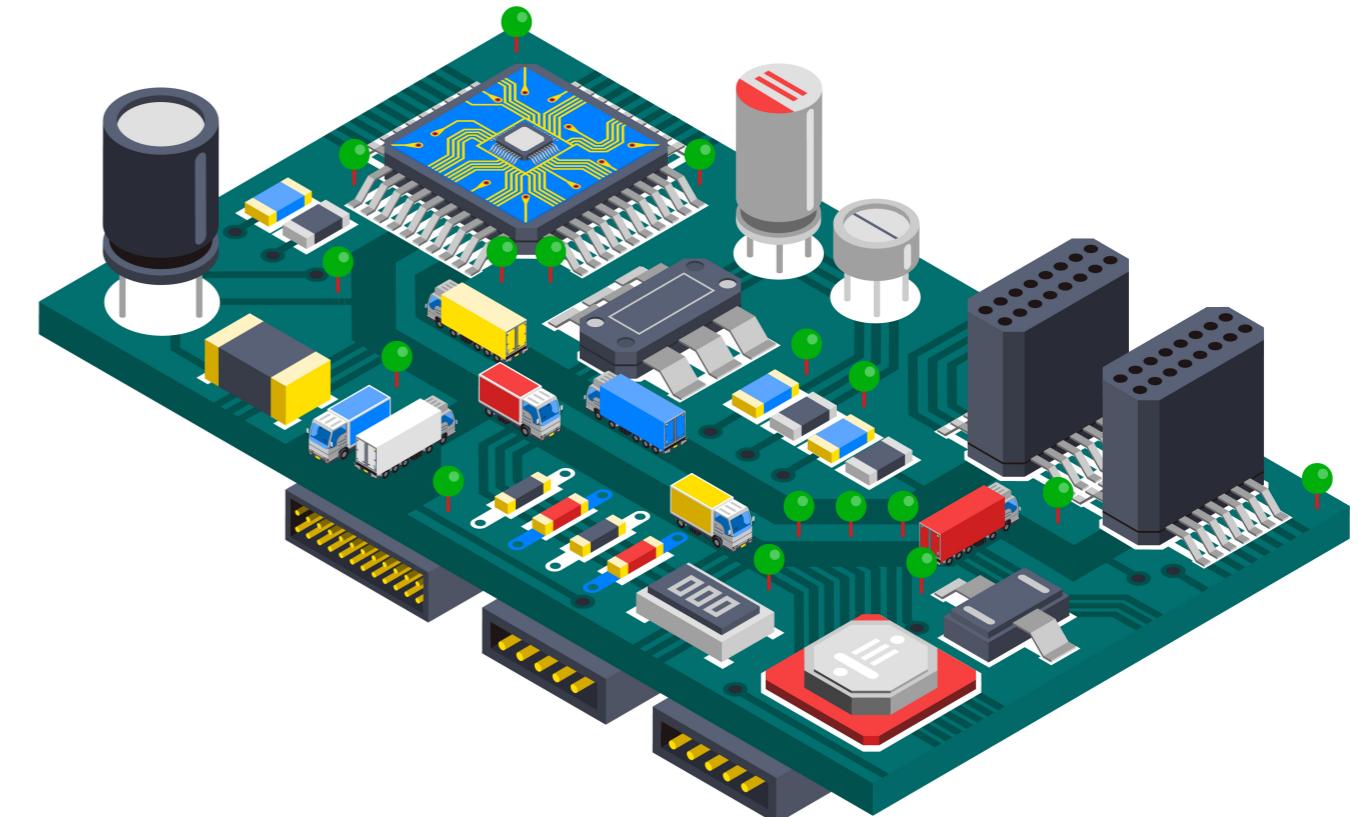


- Limit redundant and delete unnecessary data
- Minimize data traffic across networks
- Back up only when difficult/impossible to recreate data

<sup>1</sup> Data vector created by storyset (<https://www.freepik.com/vectors/data>)

# Hardware patterns

- Use the most **energy-efficient instances**
  - Keep up to date regarding new instances improvements
- Only use GPUs for the necessary time
- Use **managed services**:
  - Shifts sustainability optimization responsibility to AWS
  - Distributes sustainability impact across all hardware users



<sup>1</sup> Abstract vector created by macrovector (<https://www.freepik.com/vectors/abstract>)

# Development and deployment process



- Evaluate the sustainability impact before performing new deployments
- Provision build environment resources only when needed
- Test new features using managed device farms

<sup>1</sup> Abstract vector created by vectorjuice (<https://www.freepik.com/vectors/abstract>)

# Some AWS resources for sustainability

- Amazon around the globe
- Renewable energy methodology
- Amazon CloudWatch
- Amazon CloudFront
- AWS Systems Manager Fleet Manager
- Amazon CodeGuru Profiler
- AWS Device Farm
- Amazon Athena Compression Support
- AWS Compute Optimizer

# Design principles for sustainability

- Understand your **impact**
- Establish **sustainability goals**
- Maximize utilization
- Anticipate and adopt new, more **efficient hardware and software offerings**
- Use **managed services**
- Reduce the **downstream impact** of your cloud workloads

# **Let's practice!**

**AWS CLOUD CONCEPTS**

# Security and reliability

AWS CLOUD CONCEPTS



Hatim Khouzaimi

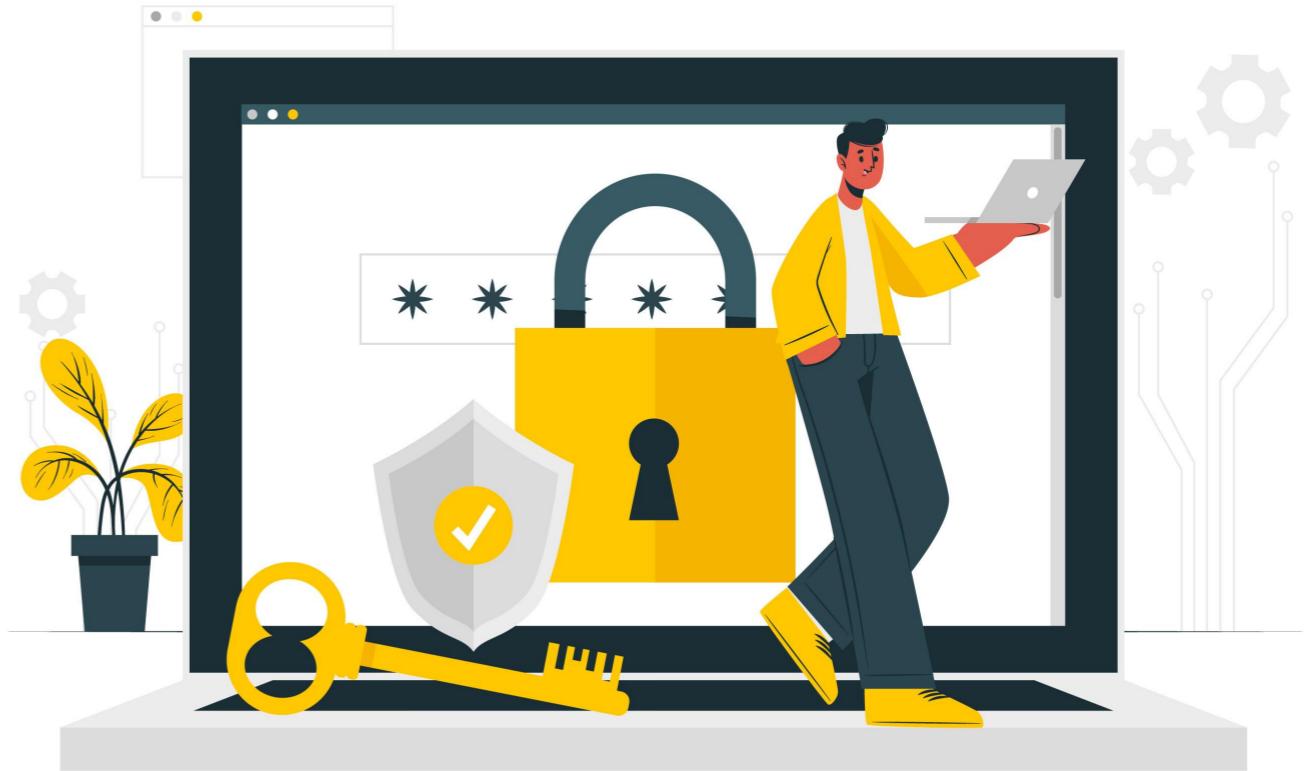
Data Scientist

# Security focus areas

- Foundations
- Identity and access management
- Detection
- Infrastructure protection
- Data protection
- Incident response

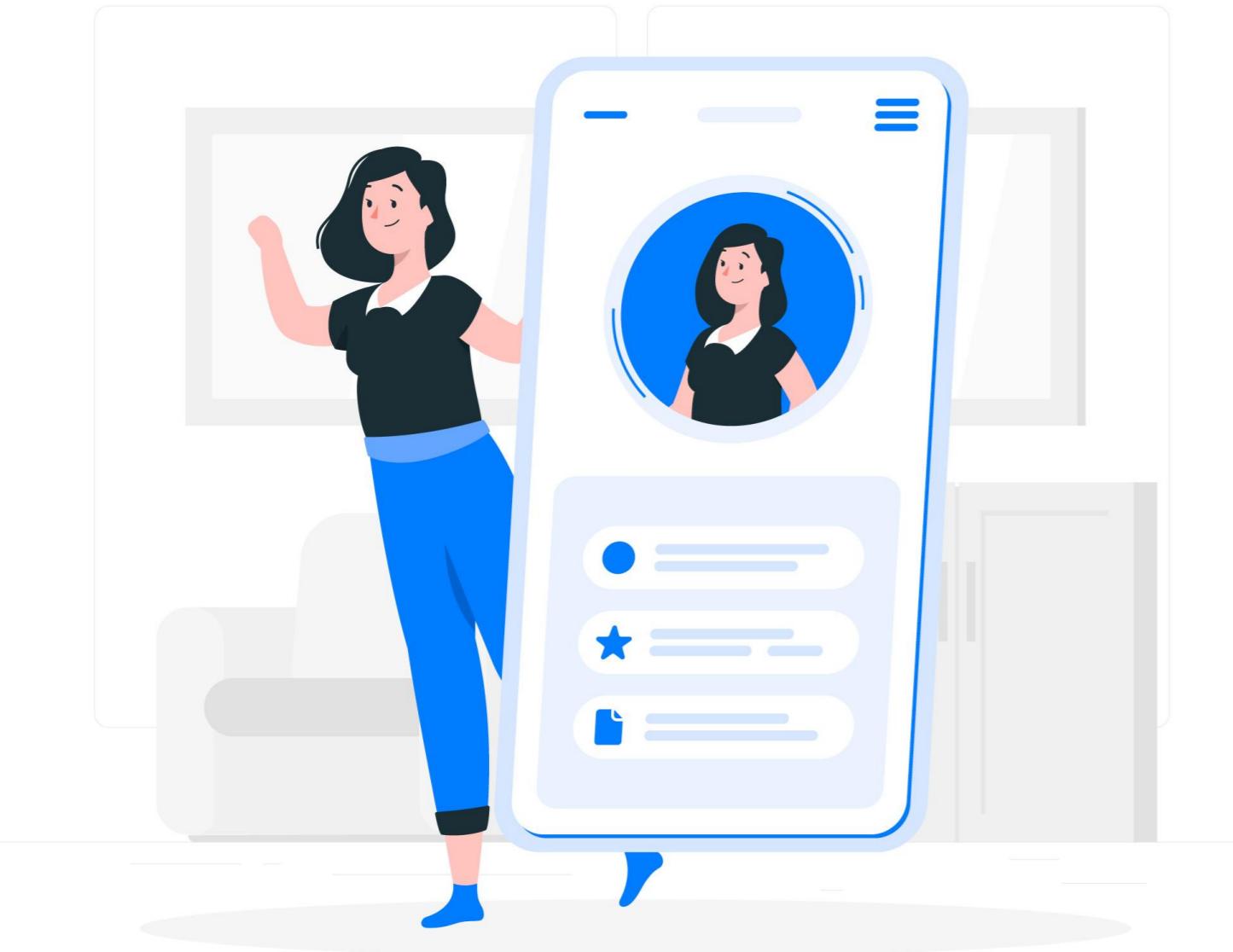
# Foundations

- AWS secures the cloud, you secure your application
- Keep contact information accurate: AWS can reach out to you quickly in case of abuse or compromise
- Stay up to date on security matters
- Automate security processes
- Separate accounts for workloads but centralize environment management using AWS Organizations



<sup>1</sup> Technology vector created by stories (<https://www.freepik.com/vectors/technology>)

# Identity and access management



- Centralized identity management
- Carefully grant permissions to human and machine identities in a fine-grained fashion
- Strong sign-in for identities with high privileges

<sup>1</sup> Social media vector created by stories (<https://www.freepik.com/vectors/social-media>)

# Detection

- Continuous logging and analysis for malicious activity detection
- Actionable security events
- Automated response to event



<sup>1</sup> Business vector created by vectorjuice (<https://www.freepik.com/vectors/business>)

# Infrastructure protection



- Several network layers:
  - Ex: using subnets with no direct access to the Internet
- Frequent scans and patches for code vulnerabilities, software integrity validations
- Managed services

<sup>1</sup> Clouds vector created by vectorjuice (<https://www.freepik.com/vectors/clouds>)

# Data protection

- Data categorization based on criticality and sensitivity
- For both data at rest and in transit:
  - Encryption
  - Key management
  - Certificate management
  - Access control enforcement



<sup>1</sup> People vector created by pch.vector (<https://www.freepik.com/vectors/people>)

# Incident response



- Clear objectives and documented plans for security incident responses
- Pre-deployed and ready to use investigation tools
- Incident simulation through game days
- Containment and recovery automation

<sup>1</sup> Technology vector created by vectorjuice (<https://www.freepik.com/vectors/technology>)

# Some AWS resources for security

- AWS Artifact
- AWS Organizations
- AWS IAM
- AWS Secrets Manager
- AWS Resource Access Manager
- AWS CloudTrail
- AWS GuardDuty
- AWS EventBridge
- AWS Transit Gateway
- AWS WAF
- AWS CodeGuru
- AWS Signer
- AWS KMS
- AWS Macie
- AWS Certificate Manager
- AWS VPN
- AWS Trusted Advisor
- AWS Detective
- AWS Security Hub

# Security design principles

1. Implement a **strong identity** foundation
2. Enable **traceability**
3. Apply security at **all layers**
4. **Automate** security best practices
5. **Protect** data in transit and at rest
6. **Keep people away** from data
7. **Prepare** for security events

# Reliability focus areas

- Foundations
- Workload architecture
- Change management
- Failure management

# Foundations

- Knowing and monitoring service quotas and limitations
- Network topology planning:
  - High connectivity for public endpoints
  - Redundant connectivity between private networks



<sup>1</sup> Business vector created by freepik (<https://www.freepik.com/vectors/business>)

# Workload architecture

- Distributed systems to **prevent failures**:
  - Microservices (single function building blocks)
  - API communication
- **Mitigating failures**:
  - Request throttling (slowing down request processing)
  - Request timeouts



<sup>1</sup> Business vector created by vectorjuice (<https://www.freepik.com/vectors/business>)

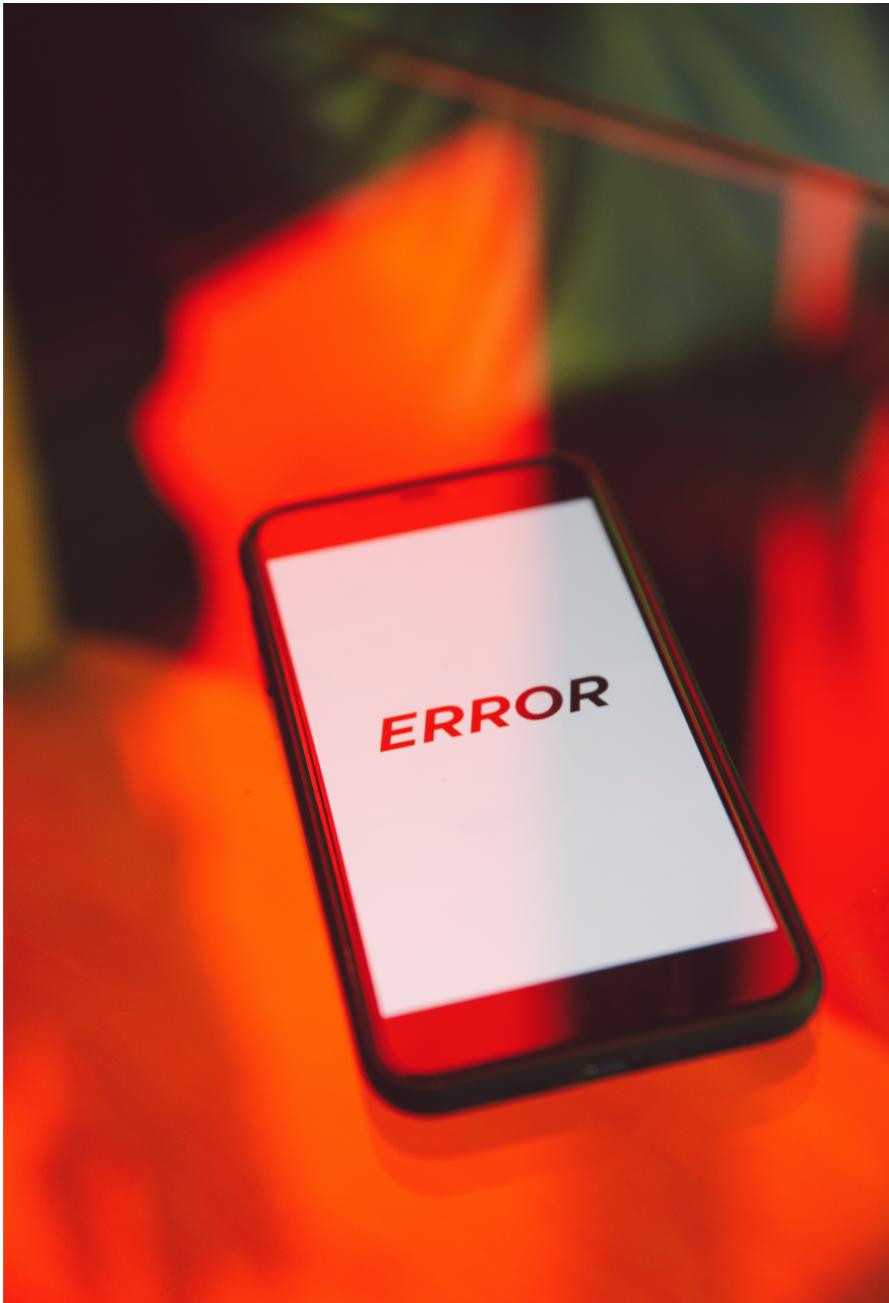
# Change management

- Automatic adaptation to **change in demand**:
  - Automatic provisioning and scaling
- Processes for **change in implementation**:
  - Automatic testing
  - Automatic deployment



<sup>1</sup> <https://www.pexels.com/photo/shallow-focus-photo-of-change-4502492/>

# Failure management



- Automatic data backup
- Multiple location deployment
- Automatic failure detection and failover
- Testing, simulation and game days:
  - Performance requirements
  - Resiliency
- Plan for disaster recovery

<sup>1</sup> <https://www.pexels.com/photo/black-smartphone-displaying-error-3747139/>

# Some AWS resources for reliability

- AWS service by service documentation
- AWS Trusted Advisor
- AWS Direct Connect
- AWS X-Ray
- AWS API Gateway
- AWS Elastic Load Balancing (ELB)
- AWS CloudTrail
- AWS CloudWatch
- AWS Auto Scaling
- AWS CodeDeploy
- AWS CodePipeline
- AWS S3
- AWS KMS
- AWS CloudFormation
- AWS Route 53
- AWS Fault Injection Simulator
- AWS Backup

# Reliability design principles

1. **Automatically recover** from failure
2. **Test** recovery procedures
3. **Scale horizontally** to increase aggregate workload availability
4. **Stop guessing** capacity
5. **Automate** change management

# **Let's practice!**

**AWS CLOUD CONCEPTS**

# Performance efficiency and cost optimization

AWS CLOUD CONCEPTS



Hatim Khouzaimi

Data Scientist

# Performance efficiency best practices four areas

- Selection
- Review
- Monitoring
- Trade-offs

# Selection

- Understand the available resources
- Evaluate the options and possible configurations, and their impact on performance
- Make decisions based on performance metrics
- Use guidance from AWS or partner



<sup>1</sup> Photo by Steve Johnson from Pexels

# Review



- Have a consistent performance review process based on well defined metrics:
  - Ex: Deming's PDCA (Plan Do Check Act)
- Stay up to date and keep track of new resources:
  - Use the ones that improve performance

<sup>1</sup> Photo by Michael Burrows from Pexels

# Monitoring

- Collect performance data and analyze it
- Use real-time processing and alarming
- Establish Key Performance Indicators (KPIs) and review them regularly



<sup>1</sup> Photo by fauxels from Pexels

# Trade-offs



- Understand:
  - The areas where performance is most **critical**
  - The **levers** to trade-off in order to maximize performance on those areas
    - Consistency
    - Durability
    - Space
    - Latency
- **Measure** the impact of performance improvements

<sup>1</sup> Photo by cottonbro from Pexels

# AWS resources for performance efficiency

- AWS Solutions Architects
- AWS Architecture Center
- AWS Partner Network (APN)
- AWS CloudWatch
- AWS X-Ray
- AWS Route 53
- AWS Direct Connect
- AWS Global Accelerator
- AWS Transit Gateway
- AWS CloudFront
- AWS Outposts
- AWS Local Zones
- AWS Wavelength
- AWS CloudWatch

# Performance efficiency design principles

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

# Cost optimization focus areas

- Practice cloud financial management
- Expenditure and usage awareness
- Cost-effective resources
- Manage demand and supplying resources
- Optimize over time

# Practice cloud financial management

- Bridge the gap between finance and technology teams
- Build a cost-aware culture and processes
- Establish cloud budget and forecasts
- Quantify business value generated by cost optimization



<sup>1</sup> Photo by Karolina Grabowska from Pexels

# Expenditure and usage awareness



- Define goals and set usage quotas
- Establish costs and usage reports
- Decommission unused resources

<sup>1</sup> Photo by cottonbro from Pexels

# Cost effective resources

- Evaluate the cost of services
- Select the optimal configuration:
  - Type of resource
  - Size of resource
  - Number of resource
  - Pricing model



<sup>1</sup> Photo by Karolina Grabowska from Pexels

# Manage demand and supplying resources



- Manage demand:
  - Throttling requests
  - Buffer-based
- Dynamic supply:
  - Demand-based
  - Time-based

<sup>1</sup> Photo by Anna Shvets from Pexels

# Optimize over time

- Implement consistent cost review processes
- Stay up to date
- Implement new services



<sup>1</sup> Photo by Lorenzo from Pexels

# AWS resources for cost optimization

- AWS Cost Explorer
- AWS Budgets
- AWS Organization
- AWS Control Tower
- AWS Cost & Usage Report (CUR)
- AWS IAM
- AWS Cost Management Blog
- AWS Auto Scaling
- AWS Simple Monthly Calculator
- AWS Pricing Calculator
- AWS Managed Services
- AWS License Manager
- AWS Compute Optimizer
- AWS CloudWatch

# Cost optimization design principles

1. Implement cloud financial management
2. Adopt a consumption model
3. Measure overall efficiency
4. Stop spending money on undifferentiated heavy lifting
5. Analyze and attribute expenditure

# **Let's practice!**

**AWS CLOUD CONCEPTS**

# Wrap-up

## AWS CLOUD CONCEPTS



**Hatim Khouzaimi**

Data Scientist

# Congratulations



# Chapter 1: Introduction to AWS



○ Regions  
○ Coming Soon

A screenshot of the AWS Management Console homepage. The top navigation bar includes 'Services', 'Resource Groups', 'Indeed', and 'Support'. The main area features a search bar, a 'Build a solution' section with cards for launching a virtual machine, building a web app, connecting an IoT device, starting a development project, and registering a domain, along with a 'See more' link; a 'Learn to build' section with categories like Websites and Web Apps, Storage, Databases, DevOps, Machine Learning, and Big Data; and a sidebar for 'Access resources on the go', 'Explore AWS' (Amazon Redshift, AWS Lambda, AWS Fargate, Amazon S3), 'AWS Marketplace', and 'Have feedback?'.

A terminal window titled 'James (aws)' showing the execution of an AWS CLI command. The command is '\$ aws dynamodb wizard new-table'. The user is prompted to enter the table name ('MyTable') and primary key ('pk'). It then asks if they want to add a sort key, select read/write capacity mode, and choose server-side encryption settings. The 'KMS - Customer managed CMK' option is highlighted in the dropdown menu.

# Chapter 2: AWS Cloud Value Framework

## Cost Savings (TCO)

### What is it?

Infrastructure cost savings/  
avoidance from moving to the  
cloud.

## Staff Productivity

### What is it?

Efficiency improvement  
by function on a task-by-  
task basis.

## Operational Resilience

### What is it?

Benefit of improved availability,  
security, and compliance.

## Business Agility

### What is it?

Deploying new features/  
applications faster and reducing  
errors

# Chapter 3: Overview of AWS core services

## ▼ All services

### Compute

- EC2
- Elastic Container Service
- Lambda
- Elastic Beanstalk
- ECR

### Storage

- S3
- Glacier
- Storage Gateway

### Database

- RDS
- DynamoDB
- ElastiCache
- Amazon Redshift

### Migration

- Database Migration Service
- Server Migration Service
- Snowball

### Networking & Content Delivery

- VPC
- API Gateway
- Direct Connect

### Developer Tools

- CodeDeploy

### Management Tools

- CloudWatch
- CloudFormation
- CloudTrail

### Config

- Systems Manager
- Trusted Advisor

### Machine Learning

- Amazon SageMaker
- Amazon Polly
- Rekognition
- Amazon Translate

### Analytics

- EMR
- Elasticsearch Service
- Kinesis

### Security, Identity & Compliance

- IAM
- GuardDuty
- Inspector
- Certificate Manager
- CloudHSM
- Directory Service

### Application Integration

- Step Functions
- Simple Notification Service
- Simple Queue Service
- SWF

### Internet of Things

- IoT Core
- IoT Device Management

# Chapter 4: The Well-Architected Framework



<sup>1</sup> <https://aws.amazon.com/architecture/well-architected/>

# Other resources

- AWS Well-Architected Tool
- AWS Well-Architected Labs
- AWS Knowledge Center
- AWS Discussion Forums
- AWS Support Center
- AWS Builder's Library
- AWS Blog
- Official AWS Podcast

# What now?



# **Best of luck**

**AWS CLOUD CONCEPTS**