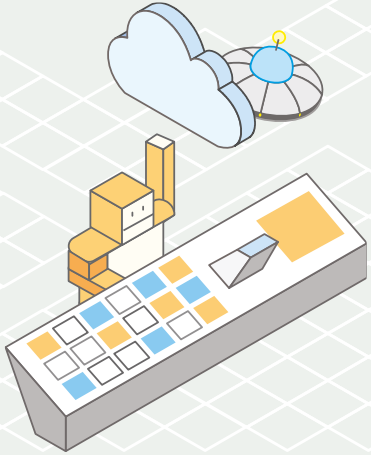
Pop-up Loft
LONDON

Being Well-Architected in the Cloud

Adrian Hornsby, Technical Evangelist @ AWS

Twitter: @adhorn
Email: adhorn@amazon.com



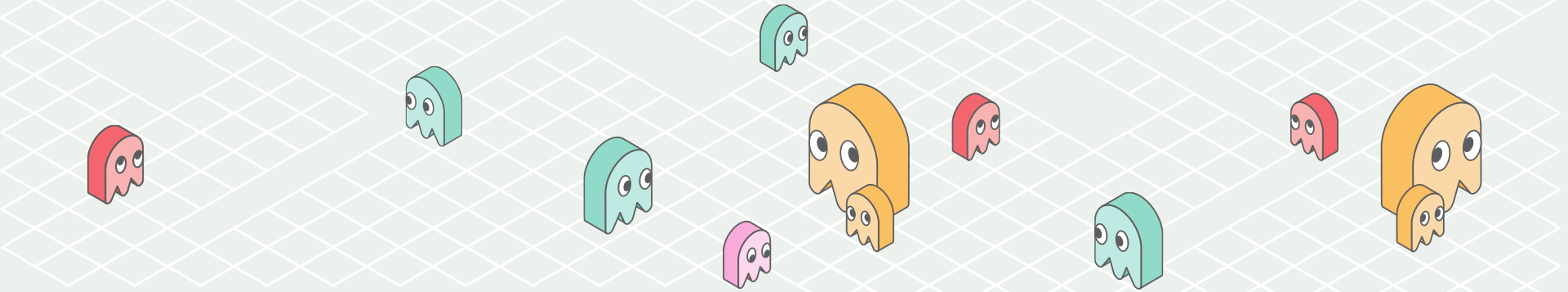
- Technical Evangelist, Developer Advocate,
  … Software Engineer
- Own bed in Finland
- Previously:
  - Solutions Architect @AWS
  - Lead Cloud Architect @Dreambroker
  - Director of Engineering, Software Engineer, DevOps, Manager, ... @Hdm
  - Researcher @Nokia Research Center
  - and a bunch of other stuff.
- Climber, like Ginger shots.

What to expect from the session

1. What is the Well-Architected framework
2. Framework Overview
3. How to be Well-Architected
4. Conclusion

aws | Pop-up Loft
LONDON

What is the Well-Architected Framework?

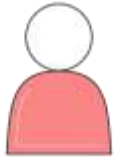


Customer Challenges

Faster response to change in market

Delivery time

Change Management



Reduce human errors

Scaling to demand

Faster recovery

High availability

Automation

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

Menu | aws | Products | Solutions | Pricing | Getting Started | Documentation | More | English | My Account | Sign In to the Console

ABOUT AWS

AWS Well-Architected

RELATED LINKS

- AWS Well-Architected
- AWS Economics Center
- Security & Compliance
- AWS Products & Services
- AWS Solutions
- Case Studies

Manage Your Resources

Sign In to the Console

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.

Build using a structured approach

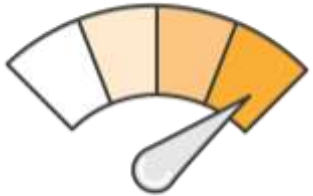
Download the Whitepaper

AWS well-architected framework

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

Security

Reliability



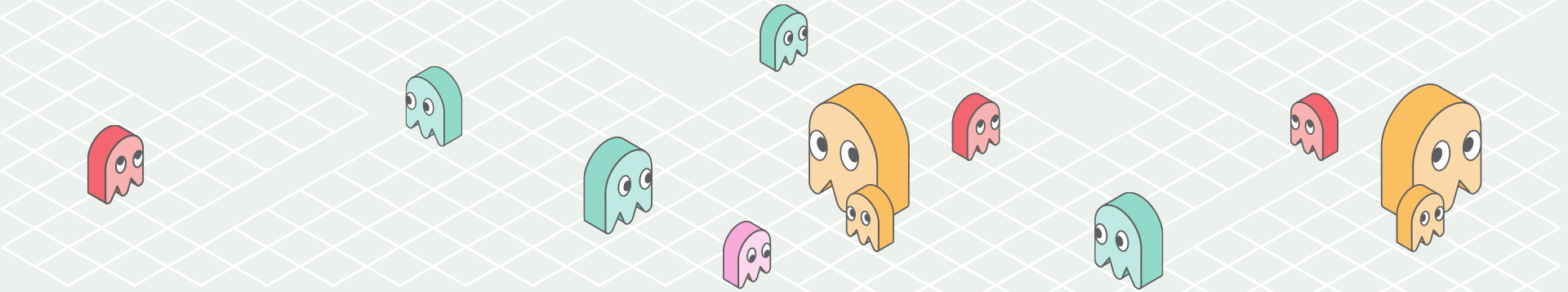
Performance efficiency

Cost optimization

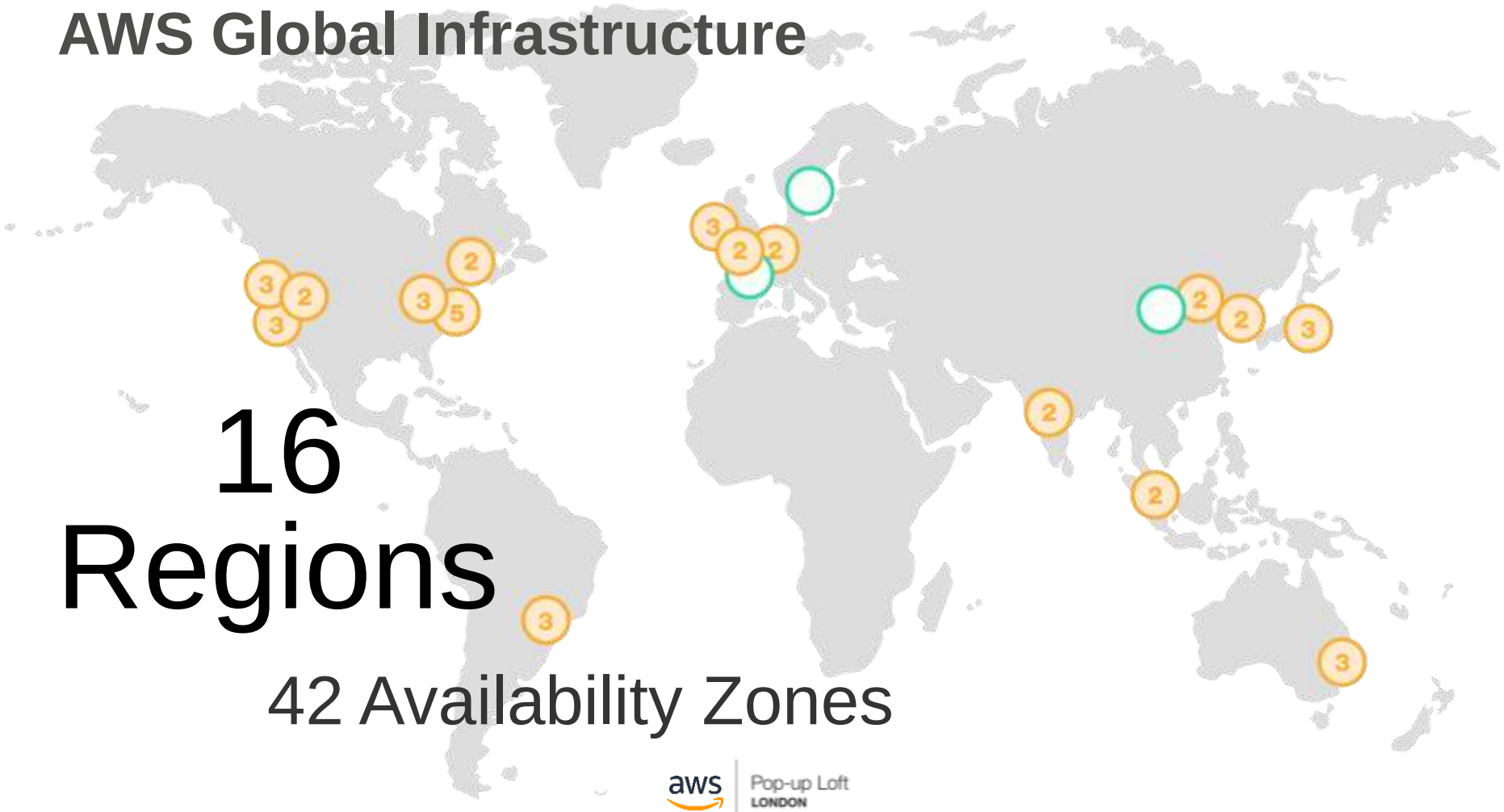
Operational excellence

aws | Pop-up Loft
LONDON

Couple of fundamentals



AWS Global Infrastructure

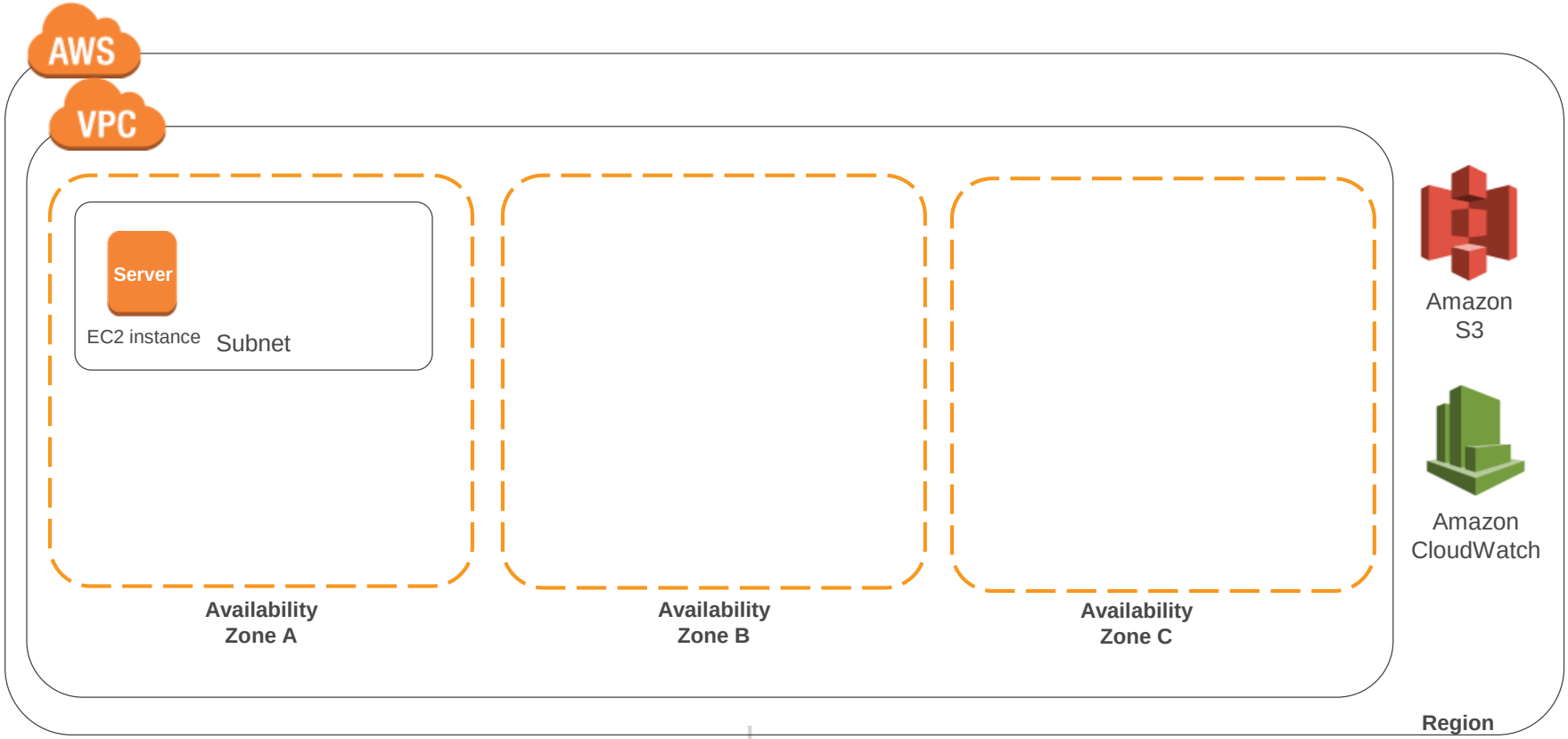


16
Regions

42 Availability Zones

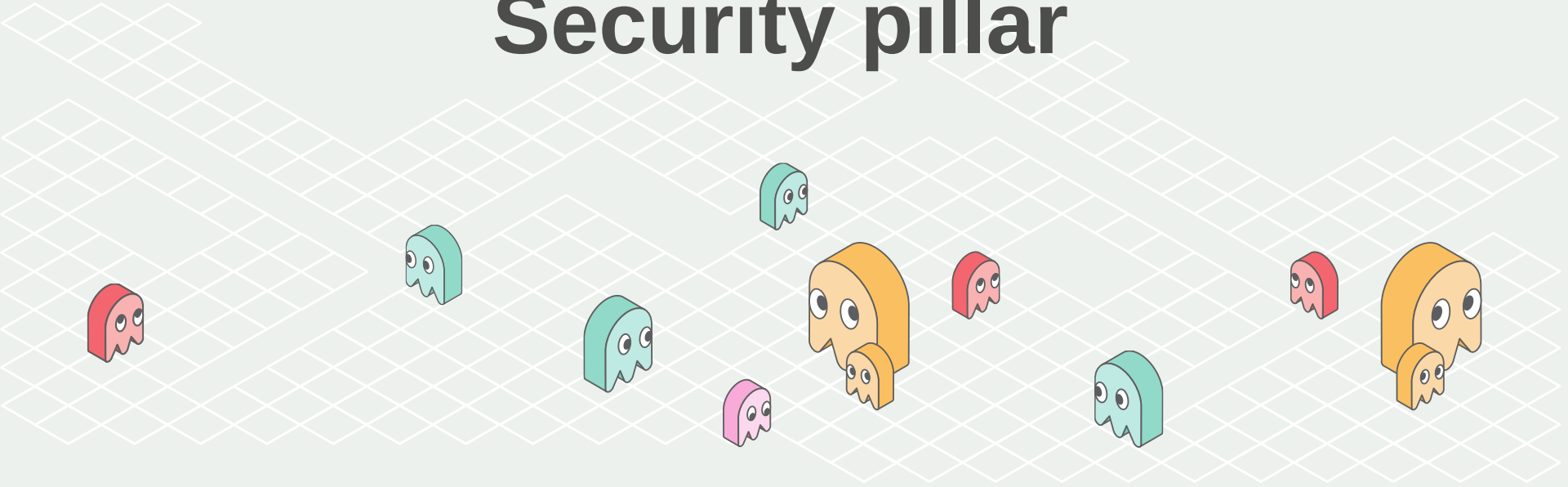
aws | Pop-up Loft
LONDON

Building Blocks



aws | Pop-up Loft
LONDON

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

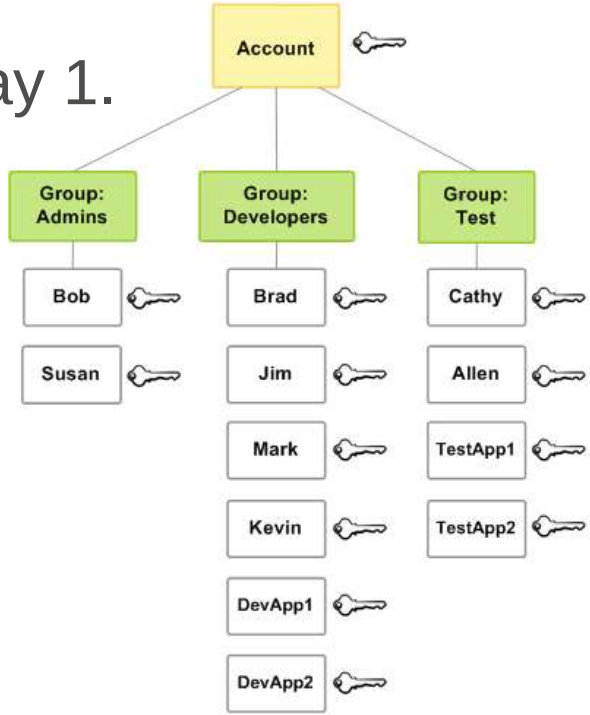
Availability Zones

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

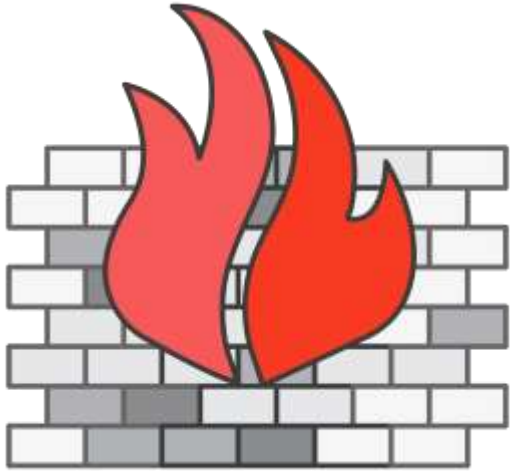
4: App accesses AWS resource(s)
Using AWS SDK application uses credentials to access resource(s)

IAM Policies

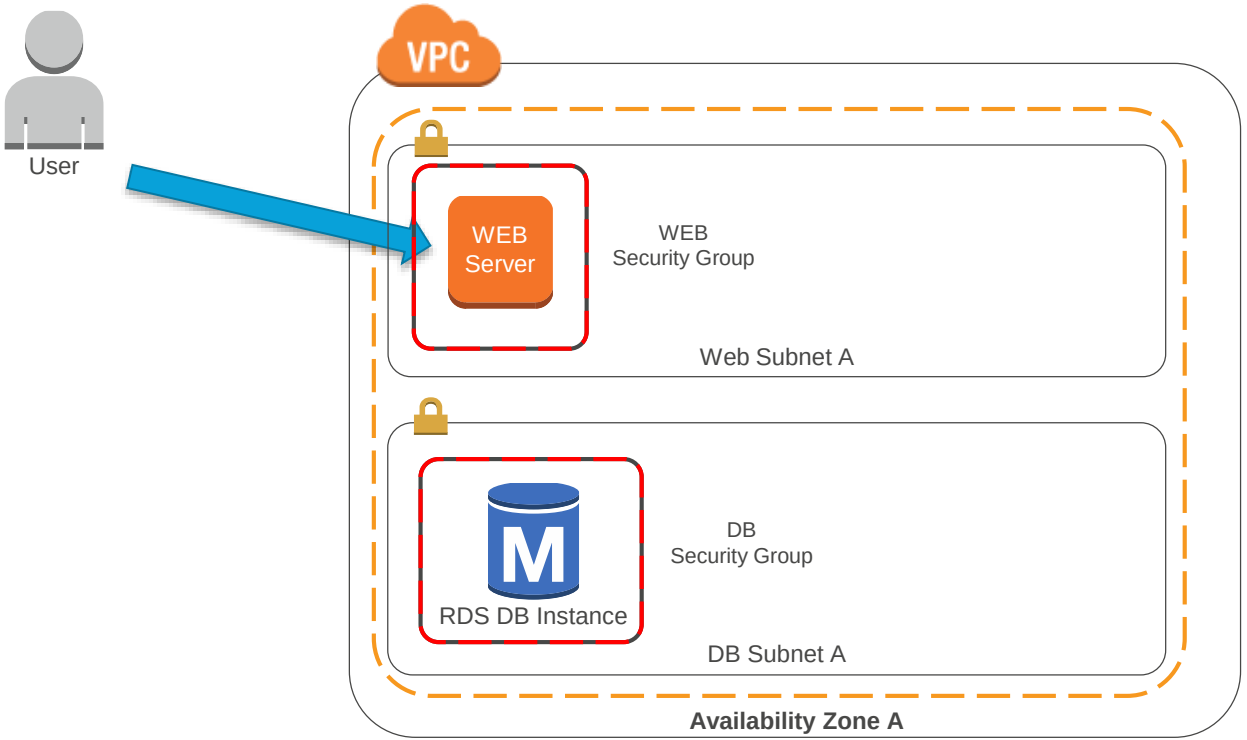
```
{
"Version": "2012-10-17",
"Statement":
    [
        {
            "Sid": "AddPerm",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::YOUR_BUCKET_NAME/*"
        }
    ]
}
```

Network and Boundary

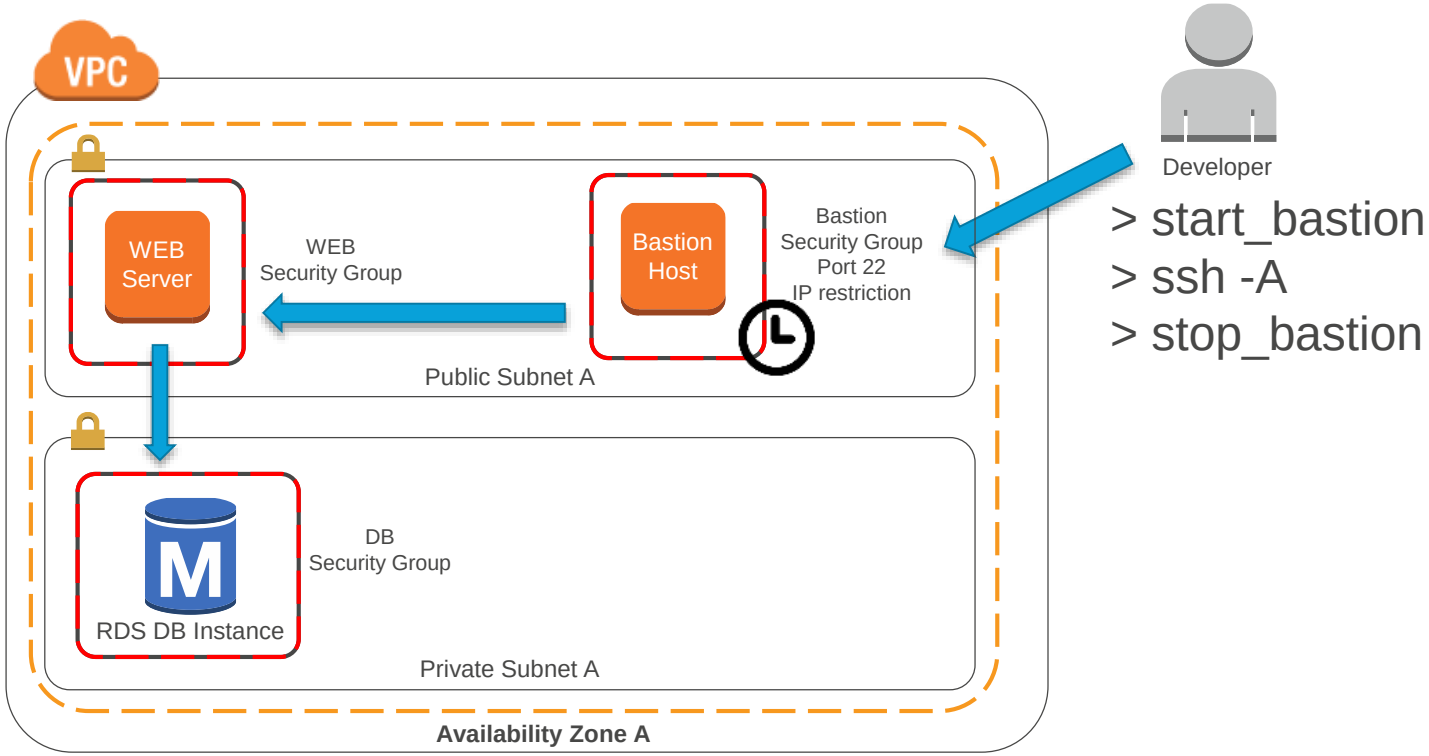
- Security groups are built-in stateful firewalls
- Divide layers of the stack into subnets
- Use a bastion host for access
- Implement host based controls



Layers with Security Groups



Bastion Host & Security Groups



Monitoring and Auditing

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

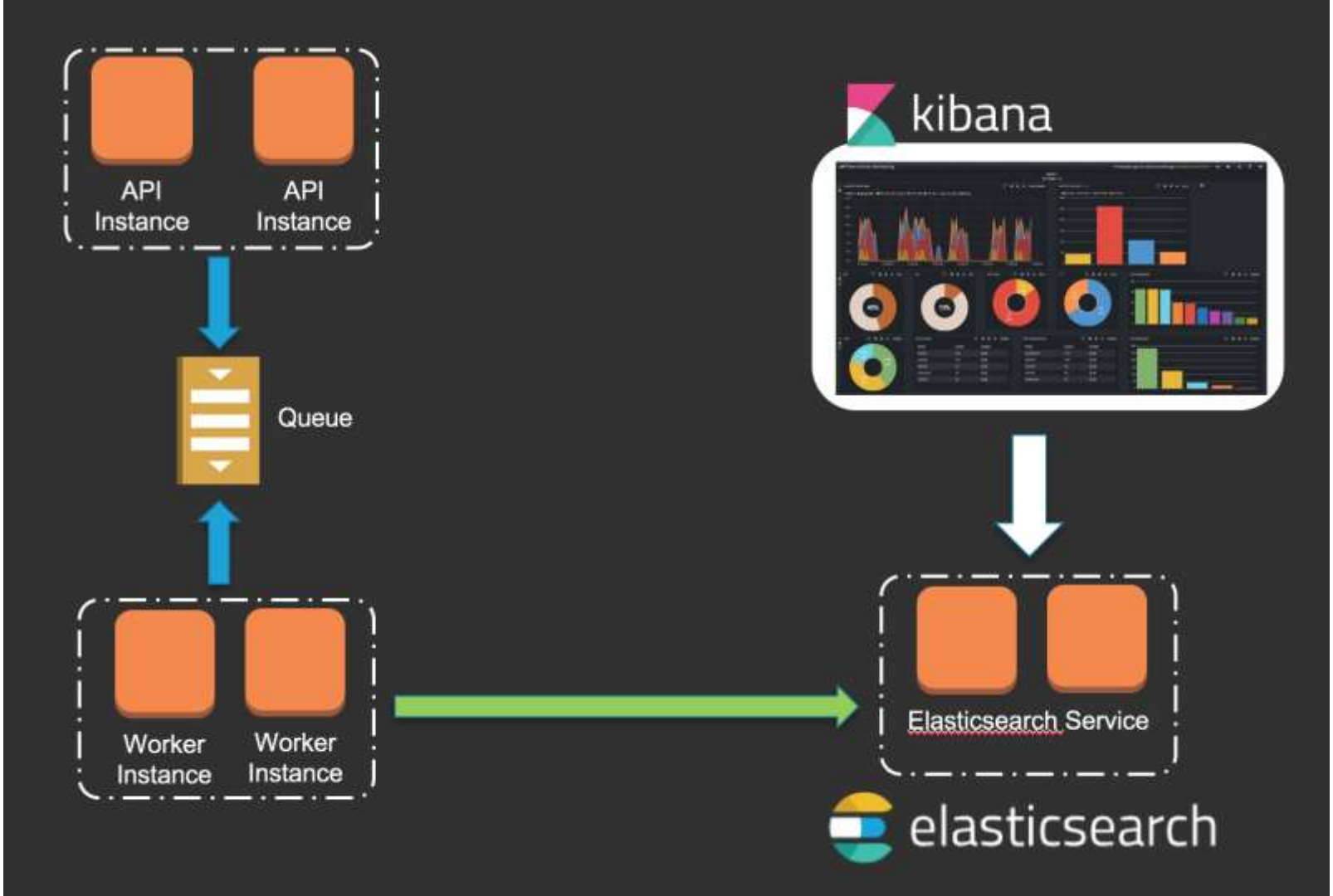
Amazon Virtual Private Cloud

AWS Identity & Access Manager

AWS Key Management Service

AWS CloudTrail

AWS Config


API Instance
API Instance
Queue
Worker Instance
Worker Instance
kibana
Elasticsearch Service
elasticsearch

This repository Search Pull requests Issues Marketplace Explore

adhorn / logtoes

Unwatch 1 Star 2 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

Demo of Asynchronous pattern (worker) using Python Flask & Celery

Edit

python celery flask flask-api gunicorn asynchronous worker-service worker aws elasticsearch redis Manage topics

14 commits 1 branch 0 releases 1 contributor Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

adhorn minor typo and adding licence term Latest commit ed1512f on 25 May

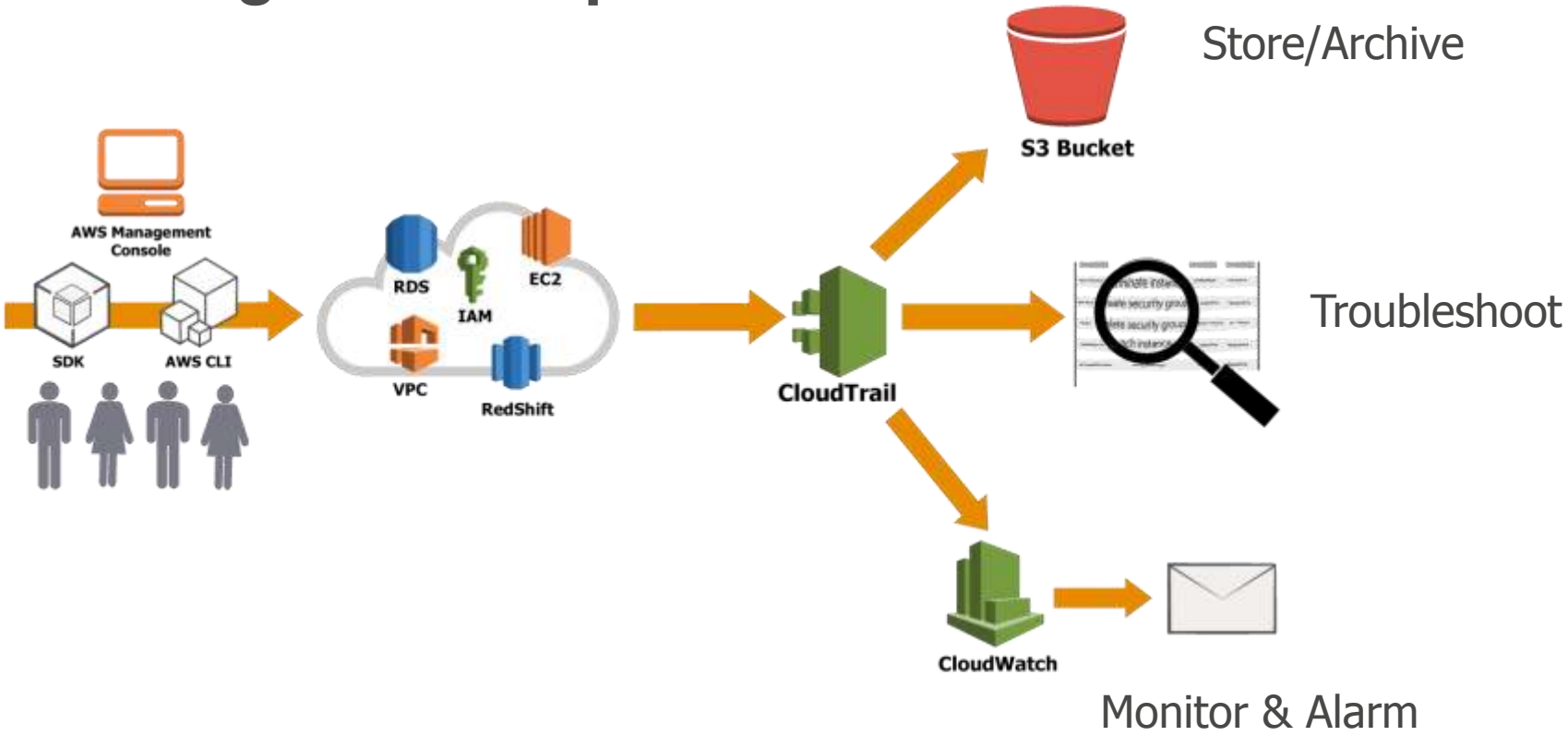
logs	Initial commit	4 months ago
logtoes	minor typo and adding licence term	3 months ago
pics	adding kibana visu	3 months ago
.gitignore	polishing the readme file	3 months ago
LICENCE	minor typo and adding licence term	3 months ago
README.md	adding kibana visu	3 months ago
requirements.txt	changing Elasticsearch driver to use the https API since AWS Elastics...	3 months ago
start_celery.py	Initial commit	4 months ago
start_flask.py	Initial commit	4 months ago

README.md

LogToES:

Simple demo of the asynchronous worker pattern using Flask and Celery. This demo demonstrates the use of a python decorator to send API logs to Elasticsearch in real-time for analysis.

Audit logs for all operations



Verify everything, always, with AWS Config

Rules

Status

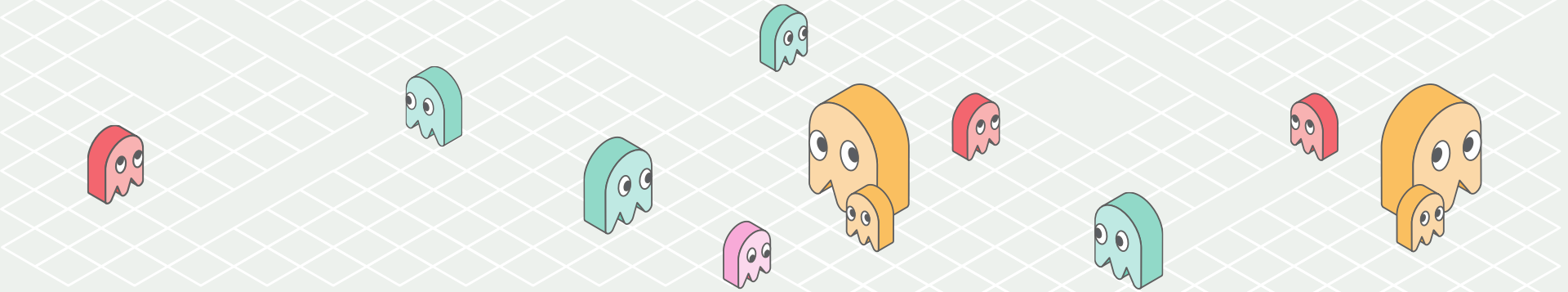
Rules represent your desired configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and summarizes the results in the following table.

Add rule

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	

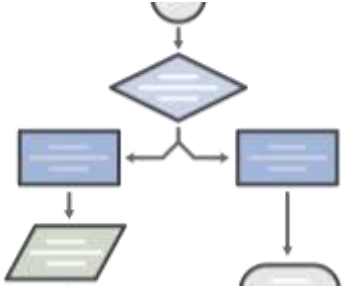
aws | Pop-up Loft
LONDON

Reliability pillar



Reliability pillar

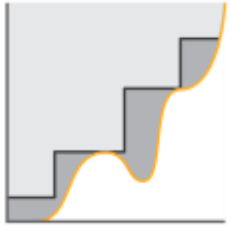
Ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues



Test recovery procedures

Automatically recover from failure

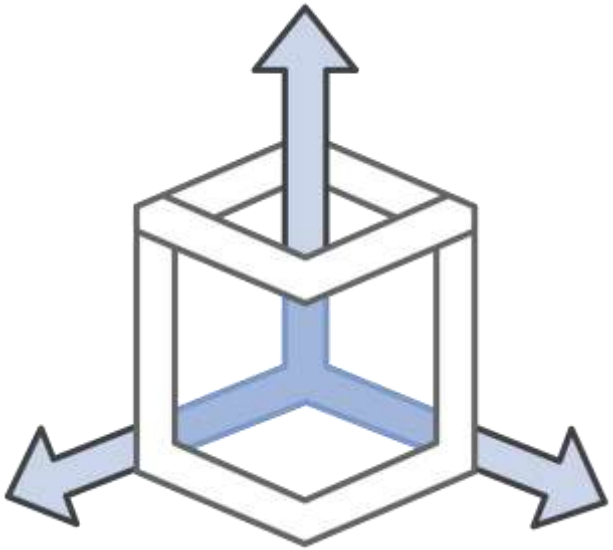
Scale horizontally to increase availability



Stop guessing capacity

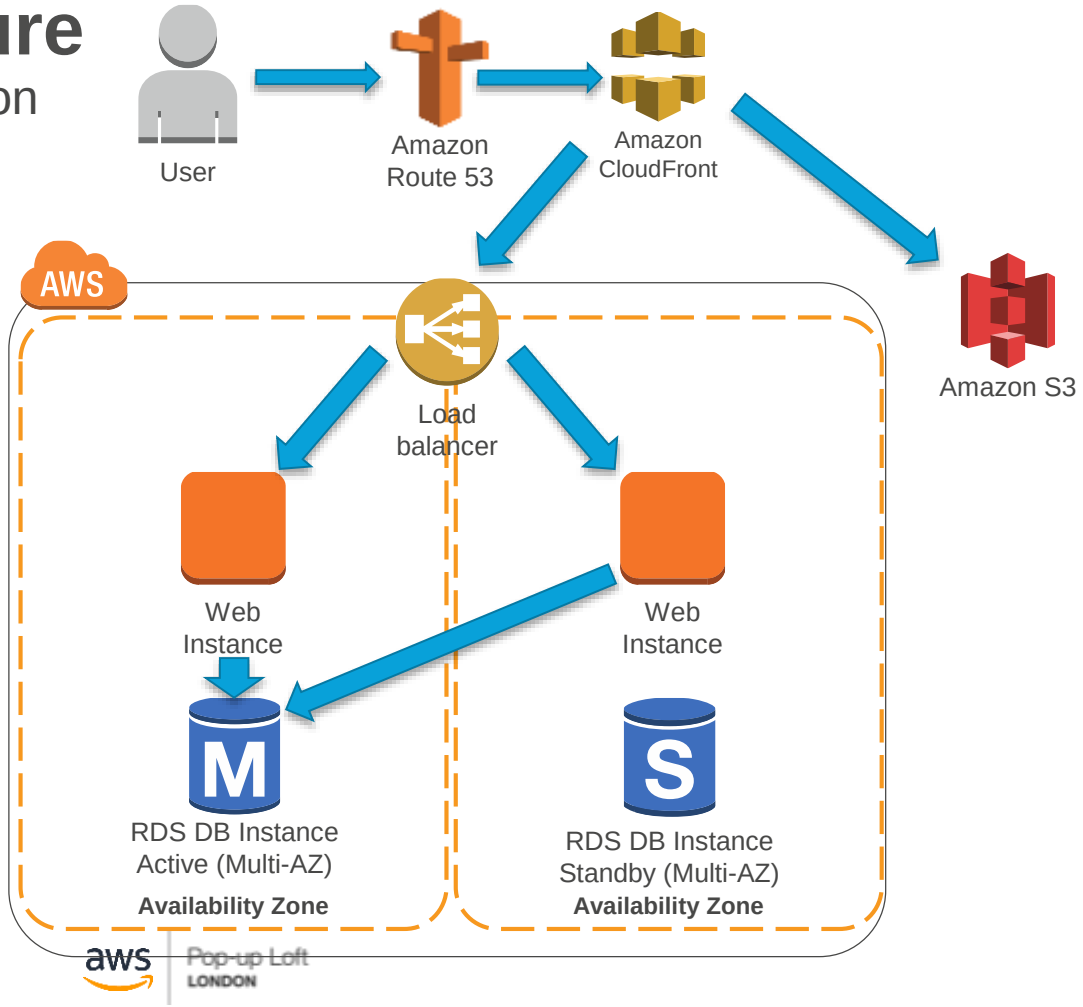
High Availability

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

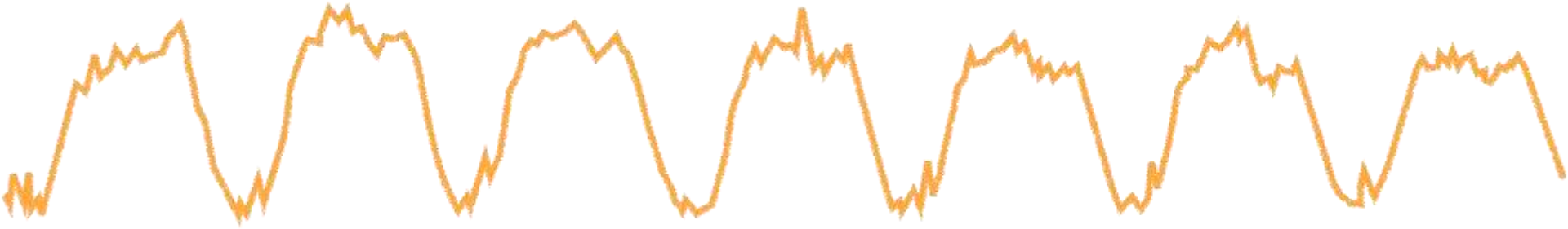


Multi-AZ Architecture

Available & redundant application



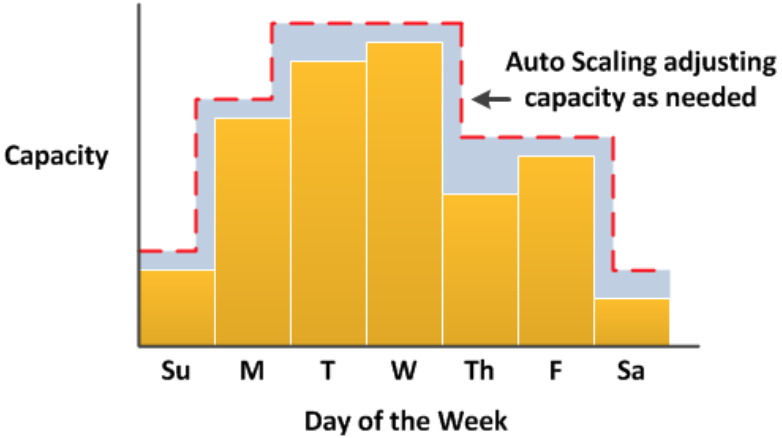
Weekly traffic pattern



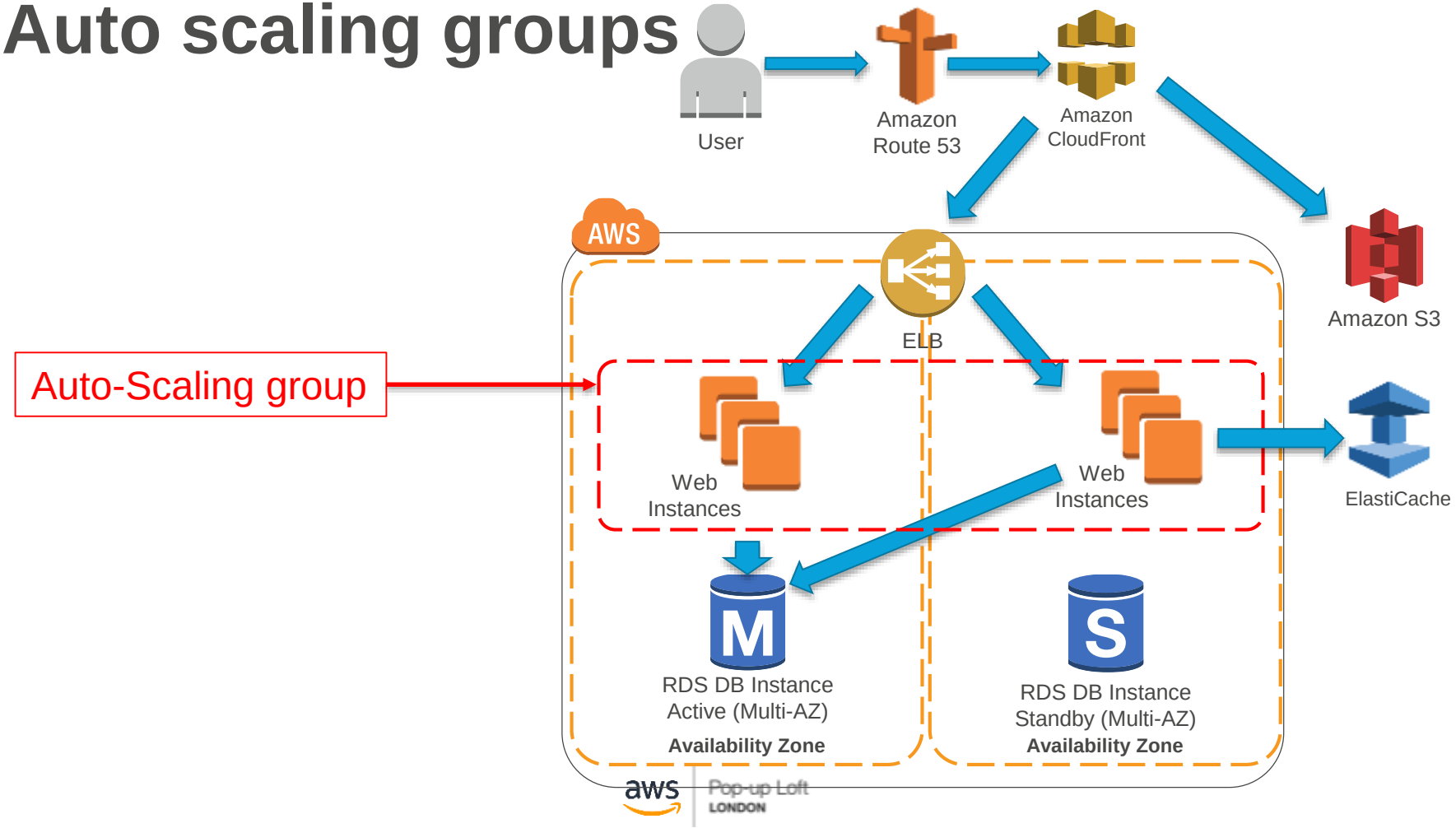
Sunday Monday Tuesday Wednesday Thursday Friday Saturday

Auto Scaling

- Maintain your Amazon EC2 instance availability
- Automatically Scale Up and Down your EC2 Fleet
- Scale based on CPU, Memory or Custom metrics

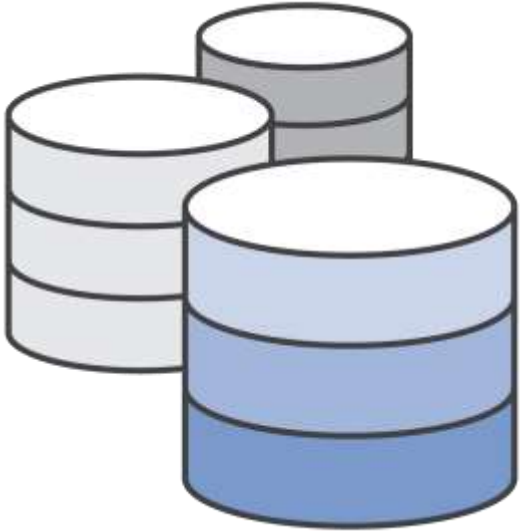


Auto scaling groups



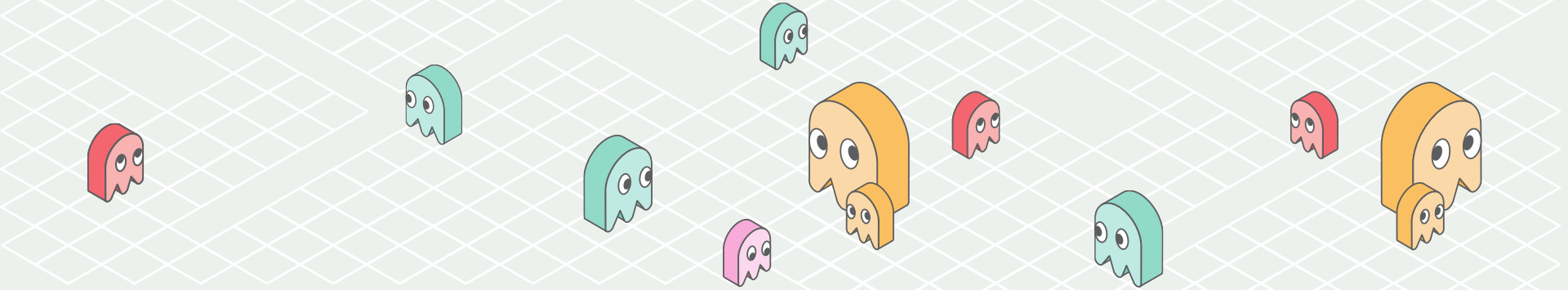
Backup and DR

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



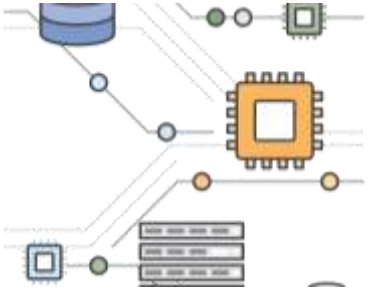
aws | Pop-up Loft
LONDON

Performance efficiency pillar



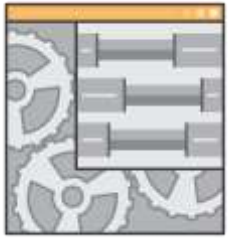
Performance efficiency pillar

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



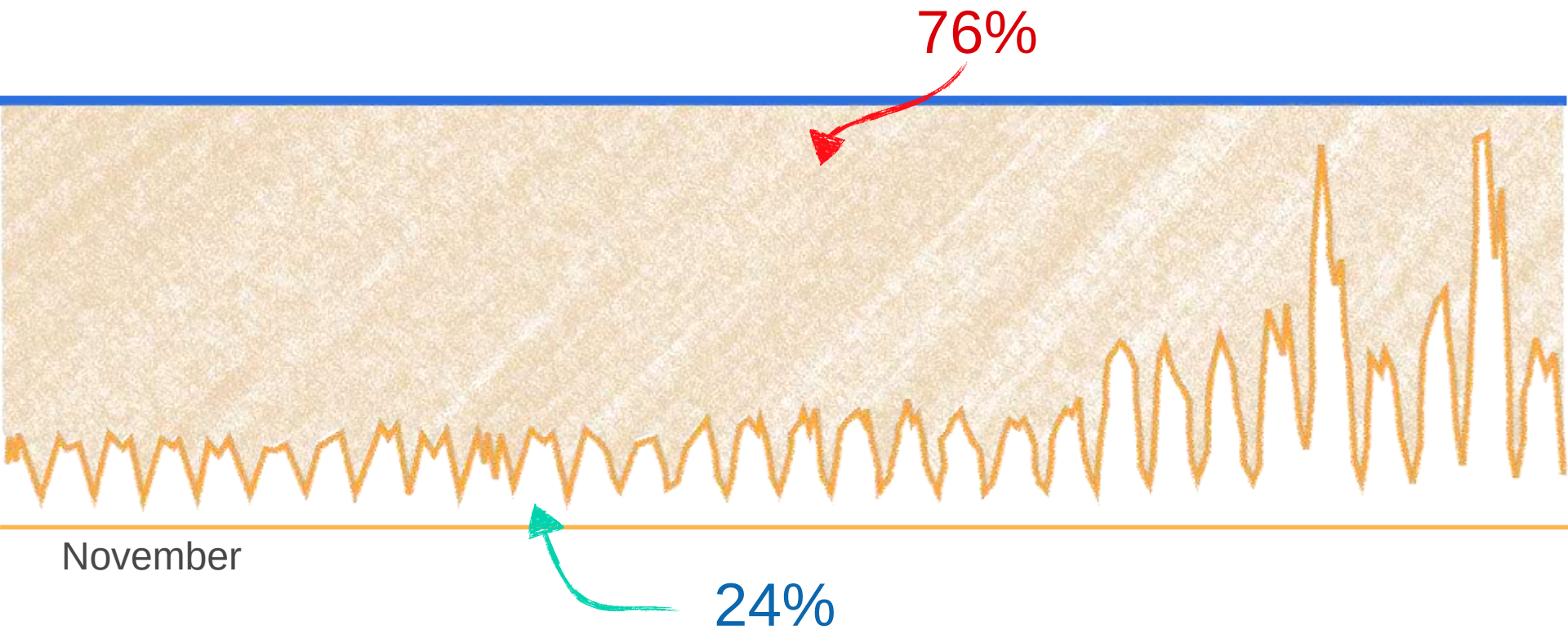
Use the right architectures for your backend and databases

Experiment more often

Right Sizing

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

Utilization vs Provisioned capacity



Proximity and Caching

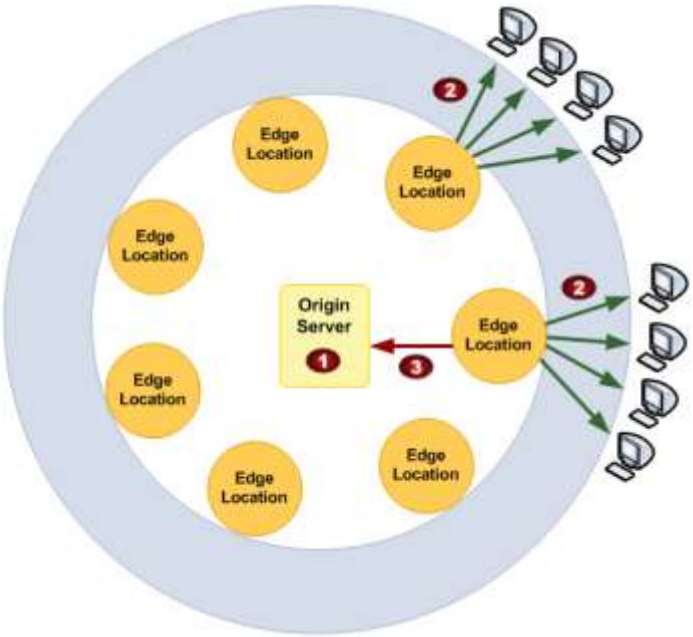
- Content Delivery Network (CDN)
- Database Caching
- Reduce Latency
- Pro-active Monitoring and Notification

Amazon CloudFront

Amazon ElastiCache

RDS DB instance read replica

Amazon CloudFront (CDN)

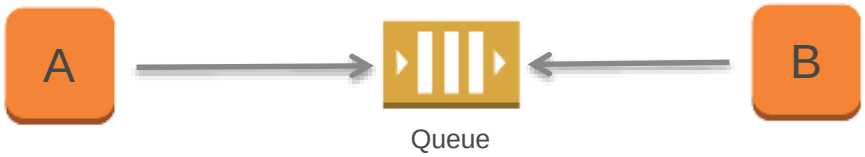


- Cache content at the edge for faster delivery
- Lower load on origin
- Dynamic and static content
- Streaming video
- Custom SSL certificates
- Low TTLs

Asynchronous patterns

Message passing

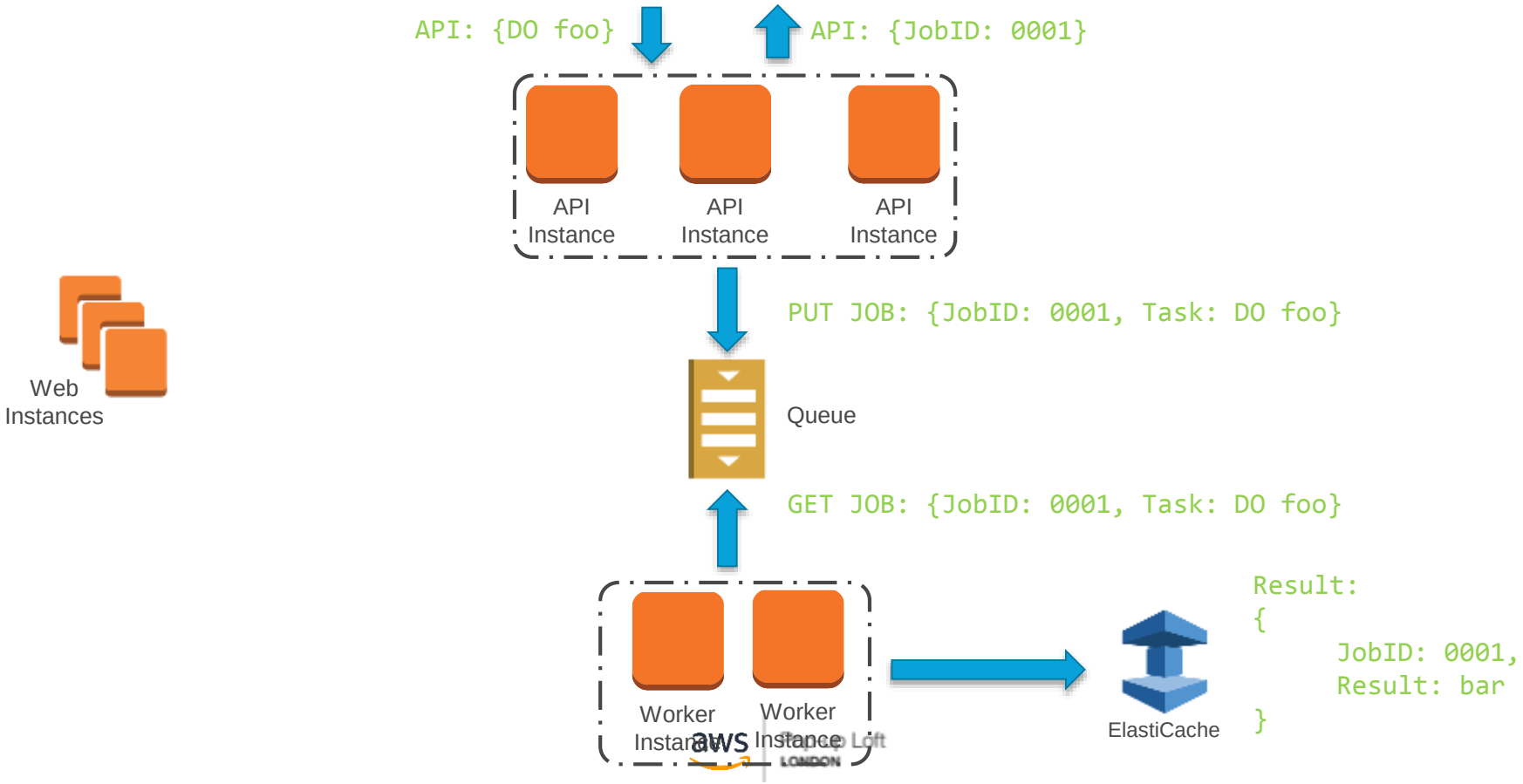
Listener



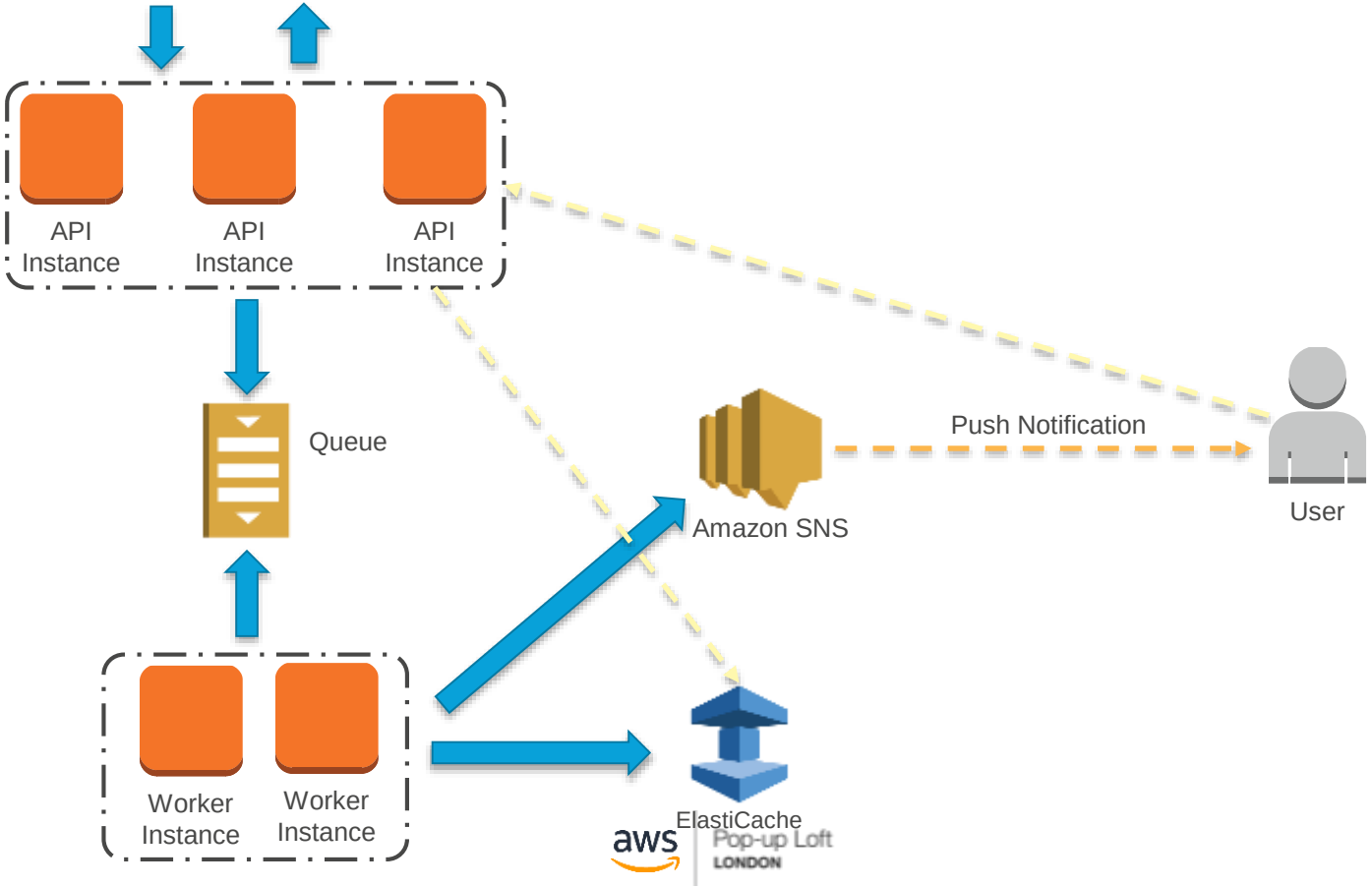
SNS, SQS, Redis, RabbitMQ

Pub-Sub

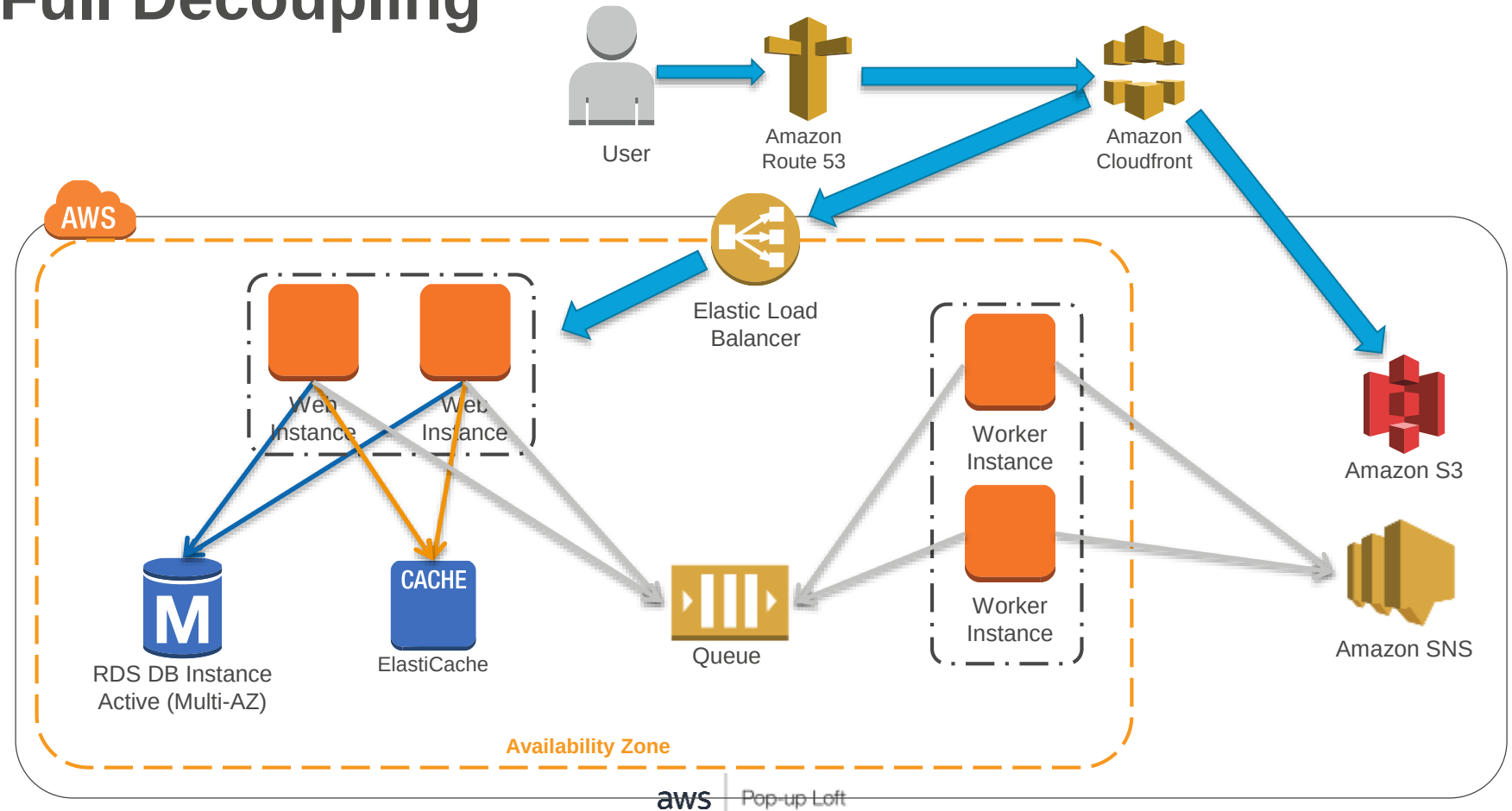
Async. Architecture (part 1)



Async. Architecture (part 2)



Full Decoupling



Event-driven patterns

Event driven

Event on B by A triggers C

Invocation

Lambda functions

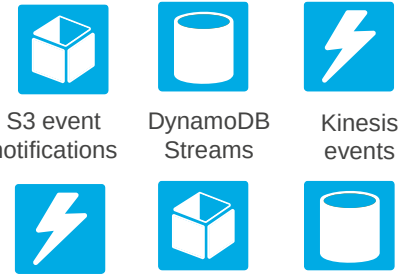
aws | Pop-up Loft LONDON

Action

How Lambda works

Invoked in response to events

- **Changes in data**
- **Changes in state**



S3 event notifications

DynamoDB Streams

Kinesis events

SNS events

CloudTrail events

Cognito events

Custom events

CloudWatch events

Lambda functions

Access any service, including your own

Any custom

Such as…

SNS

DynamoDB

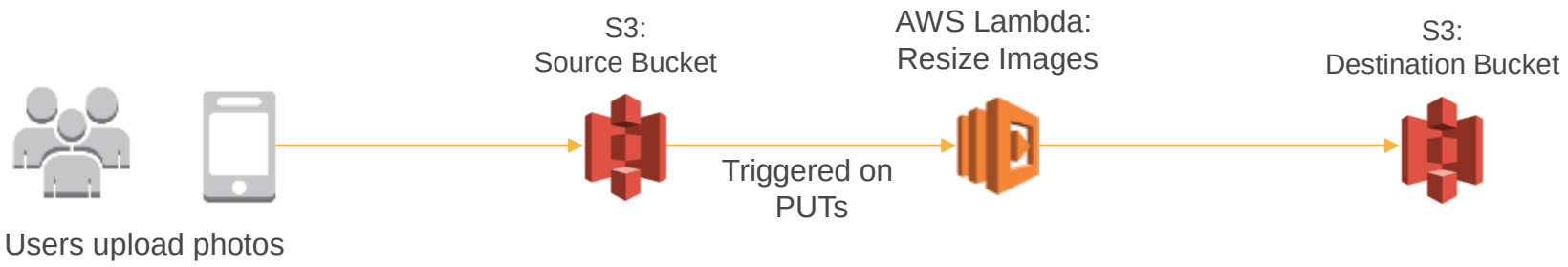
Lambda

Redshift

Kinesis

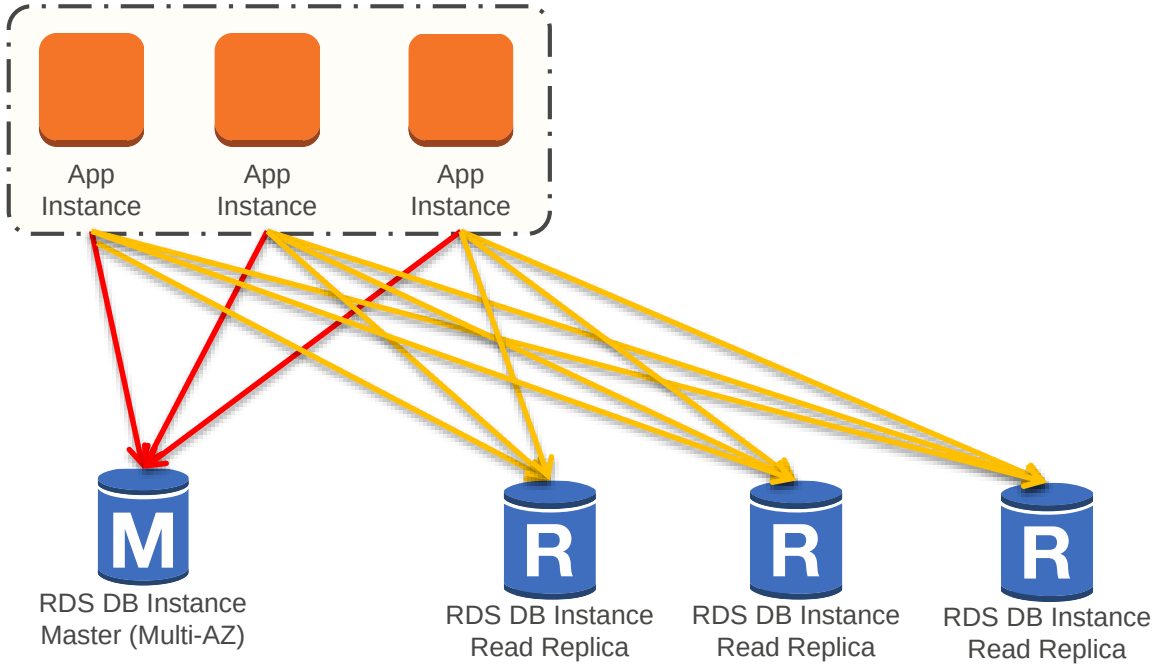
S3

Event-driven using Lambda

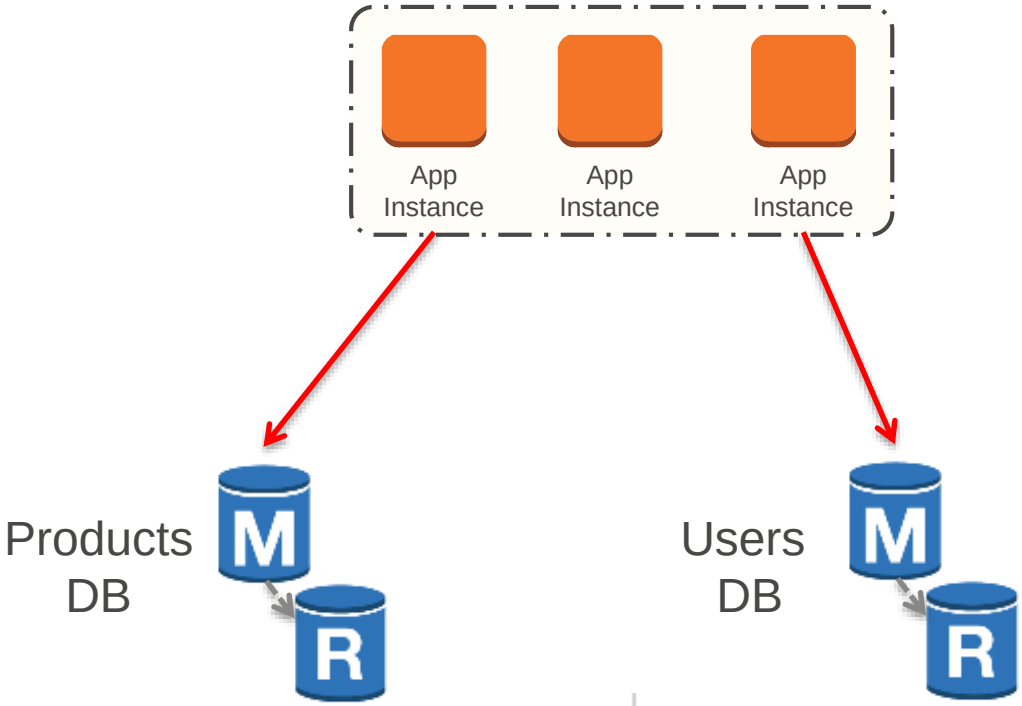


Databases

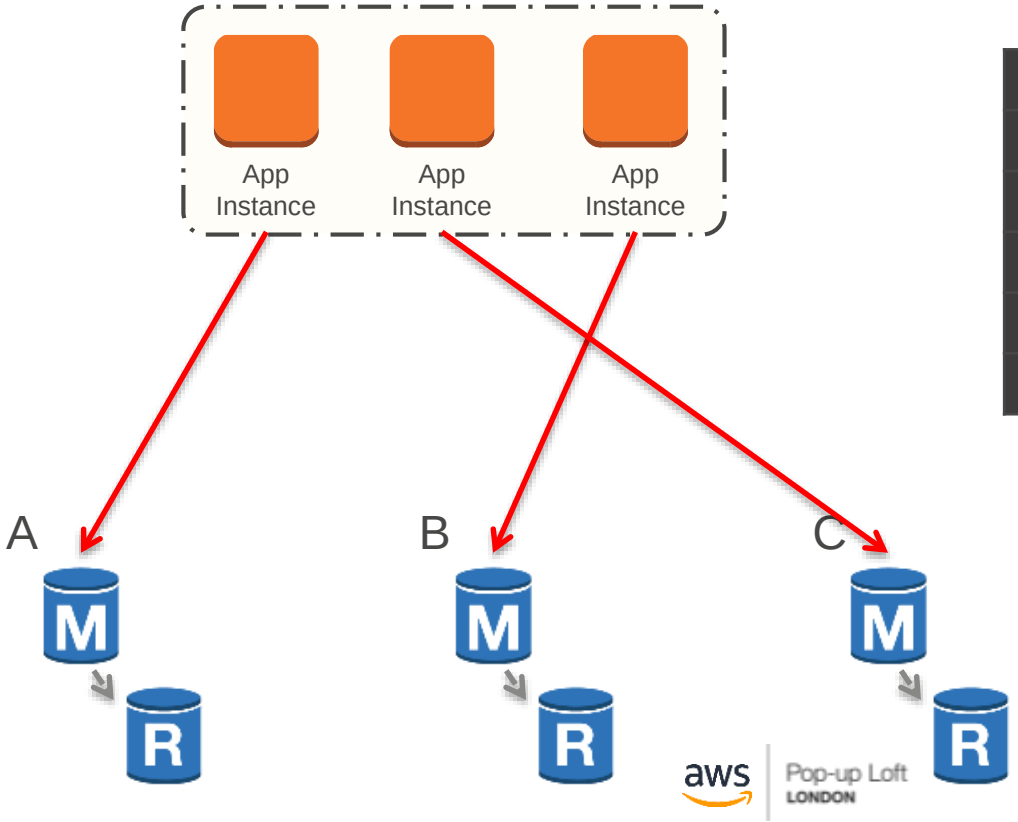
Read / Write Sharding



Database Federation



Database Sharding



User	ShardID
002345	A
002346	B
002347	C
002348	B
002349	A

Specialized Database

NoSQL

Graph DB

Database specialization example: Redis

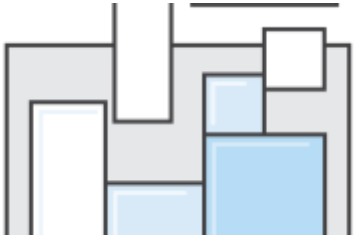
In-memory data structure store, used as a database, cache and message broker.
Specialized in data structures such as

- string
- hashes
- lists
- sets
- sorted sets with range queries
- bitmaps
- hyperloglogs
- geospatial indexes with radius queries

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

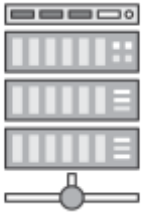
Analyze and attribute expenditure



Managed services to reduce TCO

Adopt a consumption model

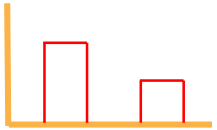
Benefits from economies of scale



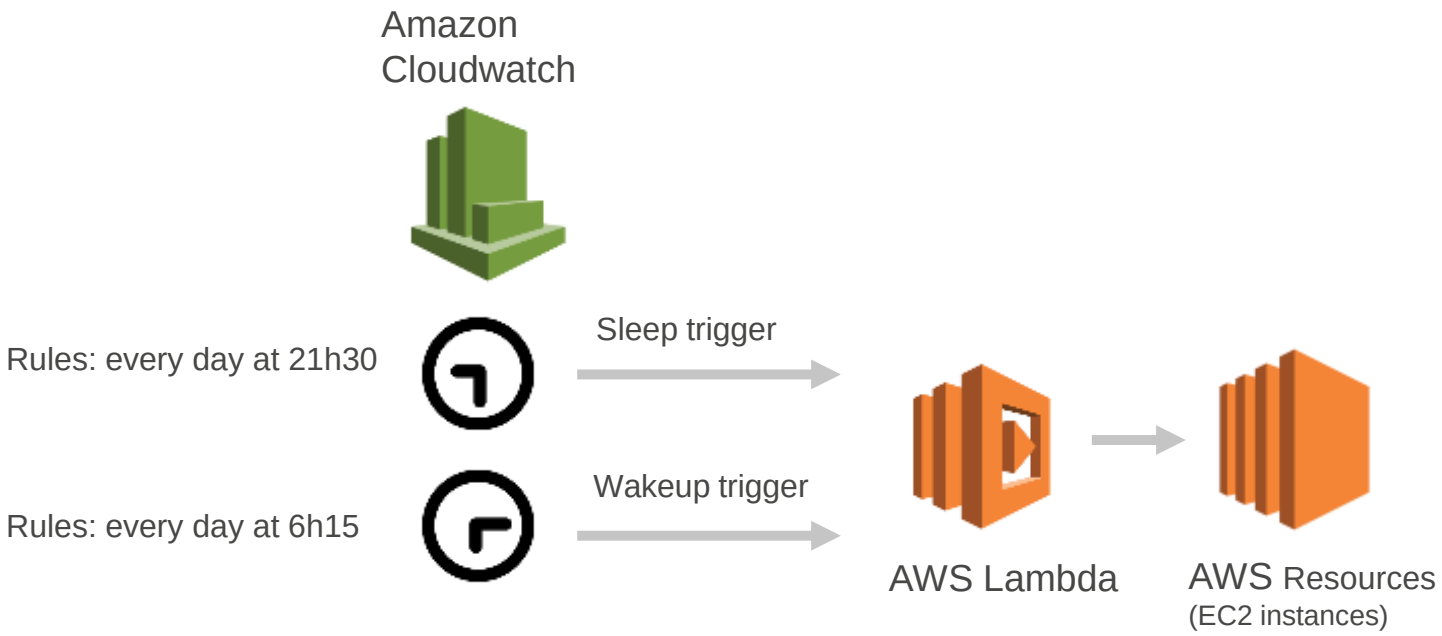
Stop spending money on data center operations

Pricing Model

- On Demand
- Reserved
- Spot
- Dedicated



Auto Start/Shutdown of Instances



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

Auto Tagging resources as they start

Events:
RunInstances

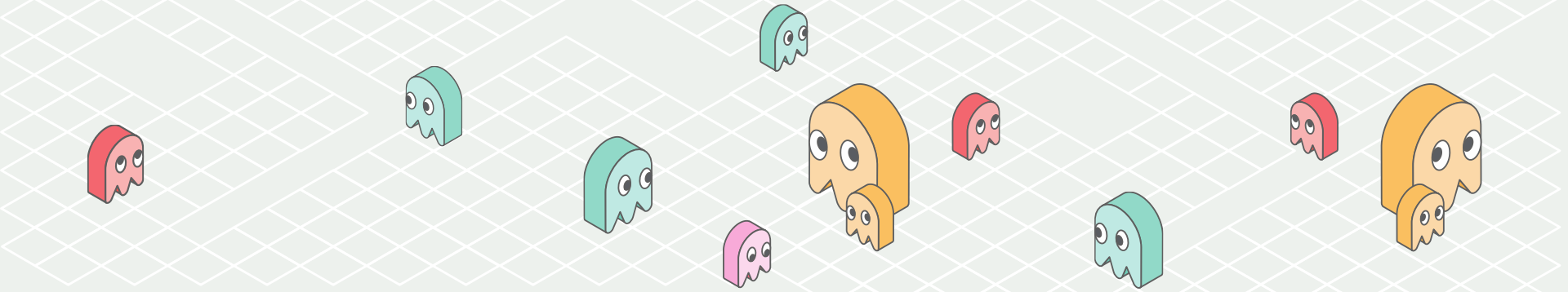
Amazon
Cloudwatch

AWS Lambda

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

aws | Pop-up Loft
LONDON

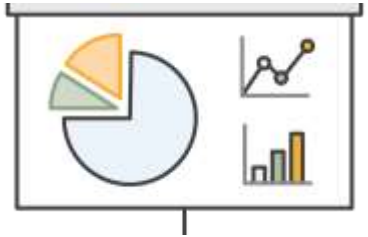
Operational excellence pillar



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

“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?

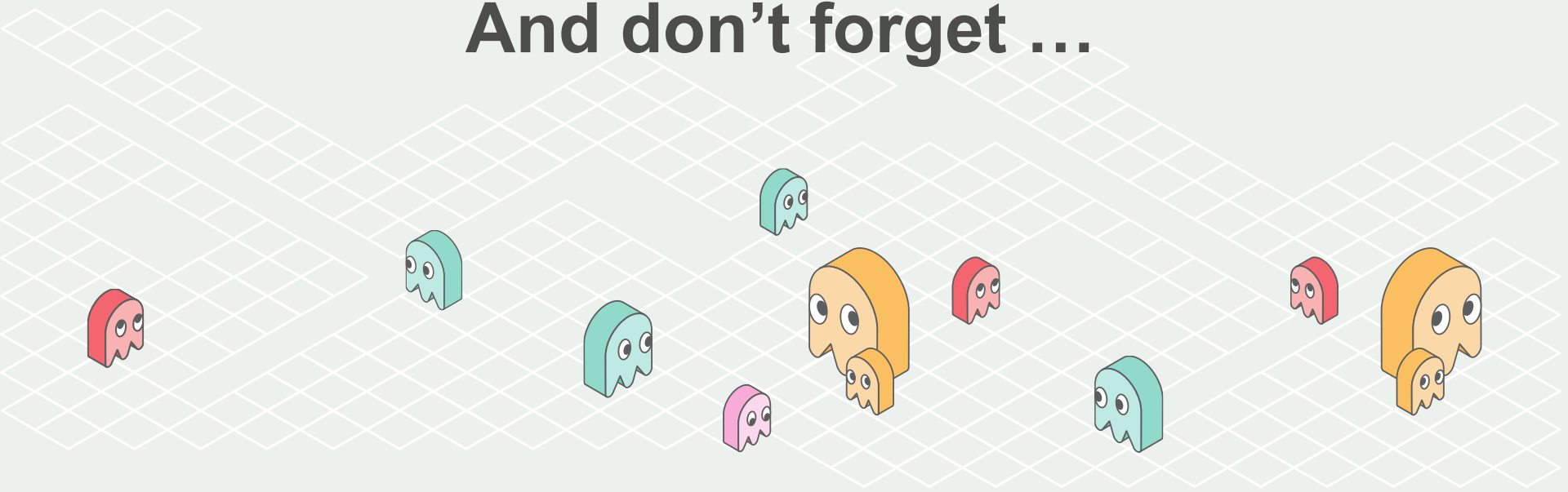
The “Must” from Day 1

Operational Excellence

- High quality code
- Version controlled
- CI/CD pipeline
- Infrastructure as code
- Security at every layer
- Cost conscious
- Test & Monitor everything
- DR procedure

aws | Pop-up Loft
LONDON

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 | Pop-up Loft
LONDON

Questions?

Twitter: @adhorn
Email: adhorn@amazon.com