

The background features a dark blue gradient with abstract geometric shapes. On the left, a large triangle is outlined in orange, with a diagonal line crossing it. On the right, a curved shape transitions from orange to blue. The text is centered in the upper right area.

# AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

ARC308

# Improve workload resiliency using shuffle sharding

Julien Lépine

Director, Solutions Architecture  
AWS

Mahanth Jayadeva

Solutions Architect, Well-Architected  
AWS



# Action plan

- Dealing with failures
- Sharding
- Shuffle sharding
- Summary of tests and observations

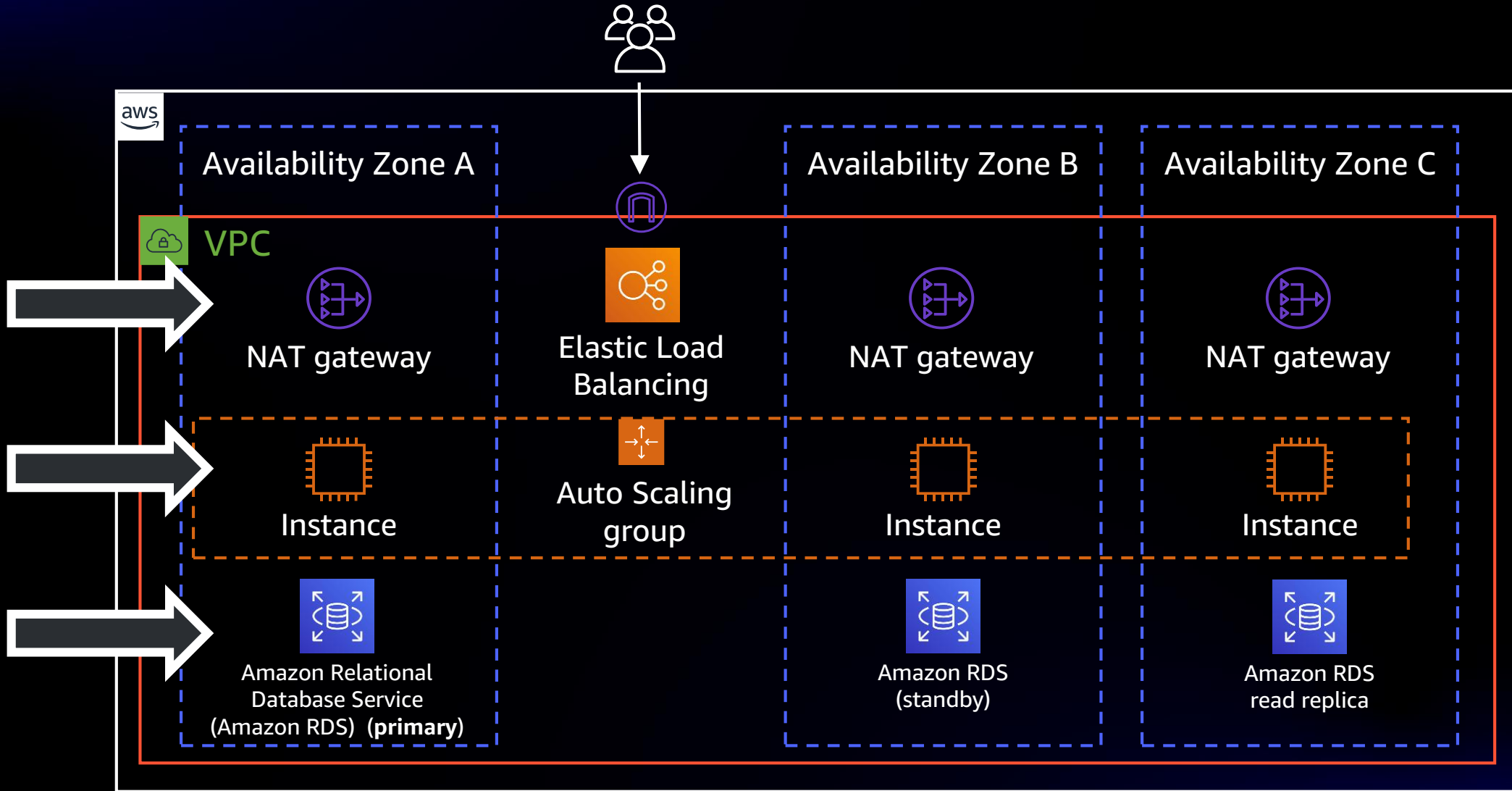
**“We needed to build systems  
that embrace failure as a  
natural occurrence”**



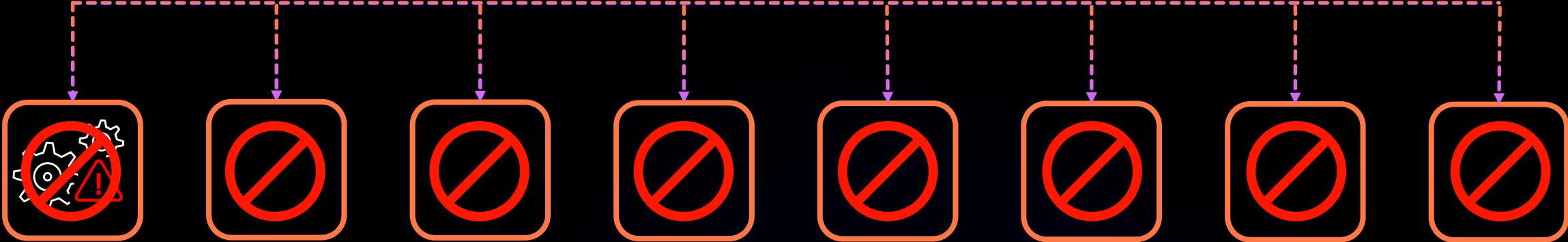
Werner Vogels  
CTO  
Amazon.com



# Traditional architecture



# Traditional architecture

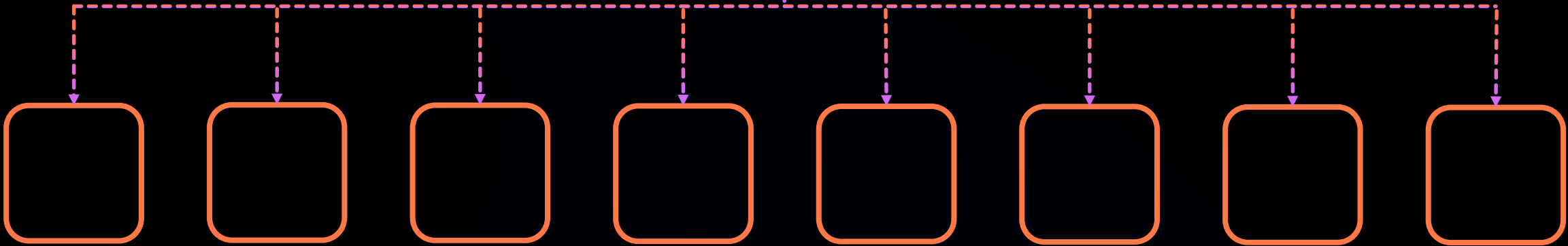


Scope of impact = All customers

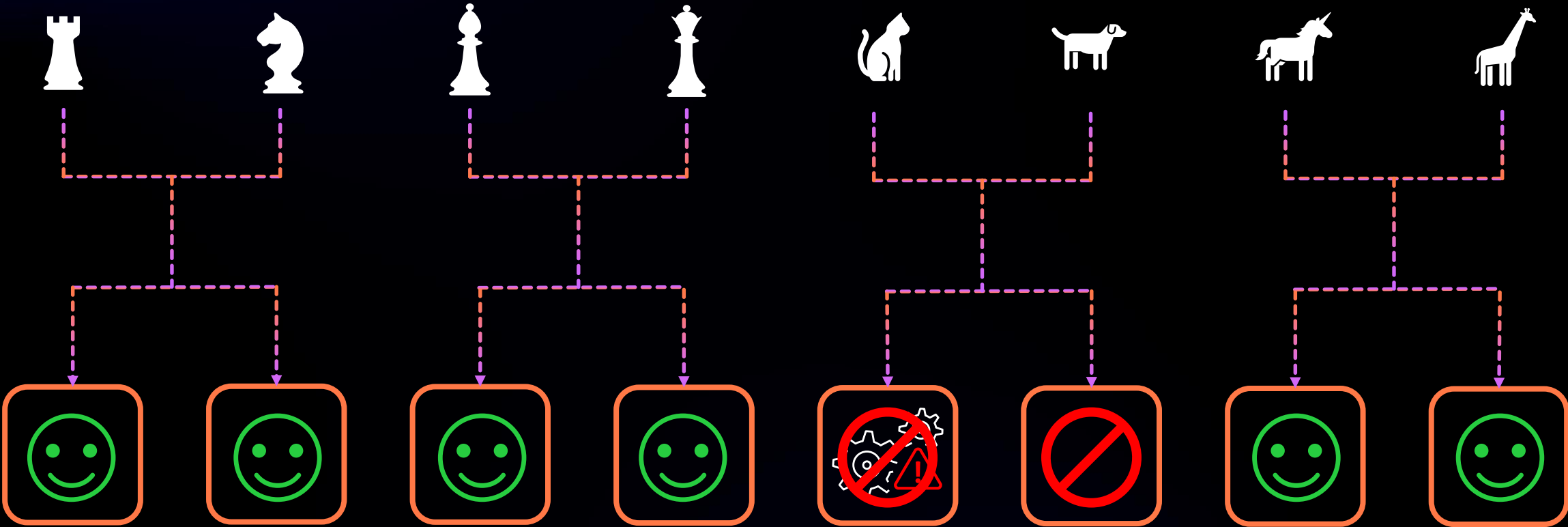
# Can we do better?



# Sharding



# Sharding



$$\text{Scope of impact} = \frac{\text{Customers}}{\text{Shards}}$$

# Hands-on lab

AWS ACCOUNTS ARE PROVIDED FOR YOU



AWS Account

In the Lab Guide,

*Always* choose the directions that say...

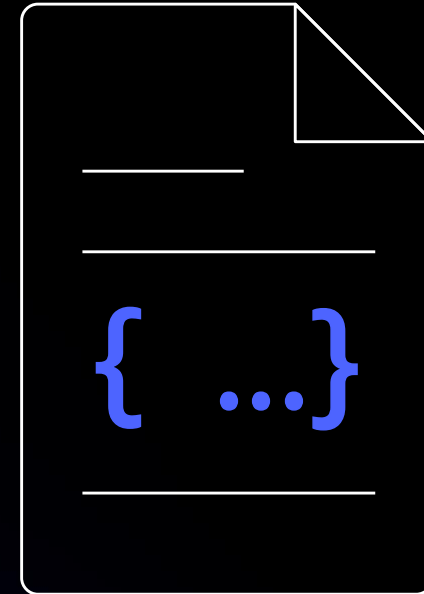
If you are attending an in-person workshop and were provided with an AWS account by the instructor:

Jot down this hash code (somewhere you can easily copy/paste it later)

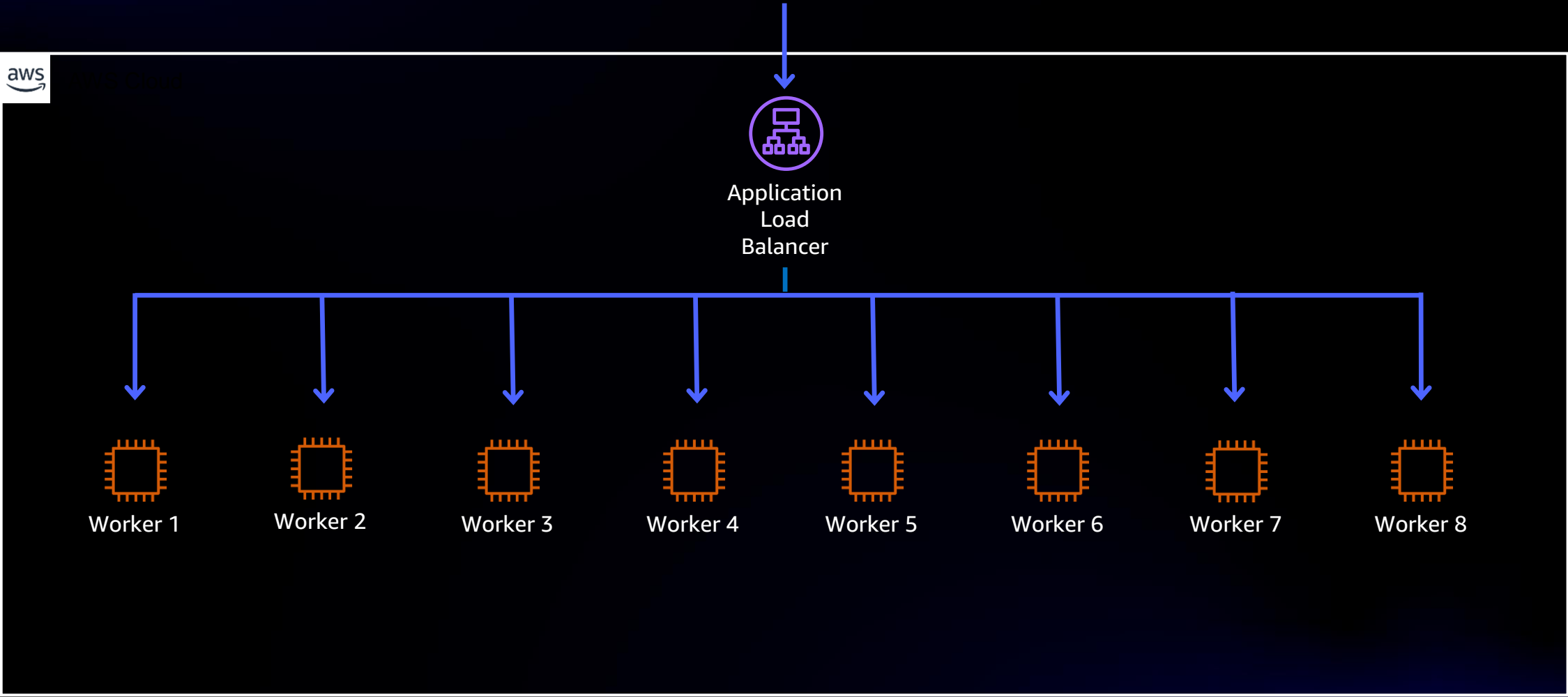
4eb8-1b23b431e4-8b

# Hands-on lab

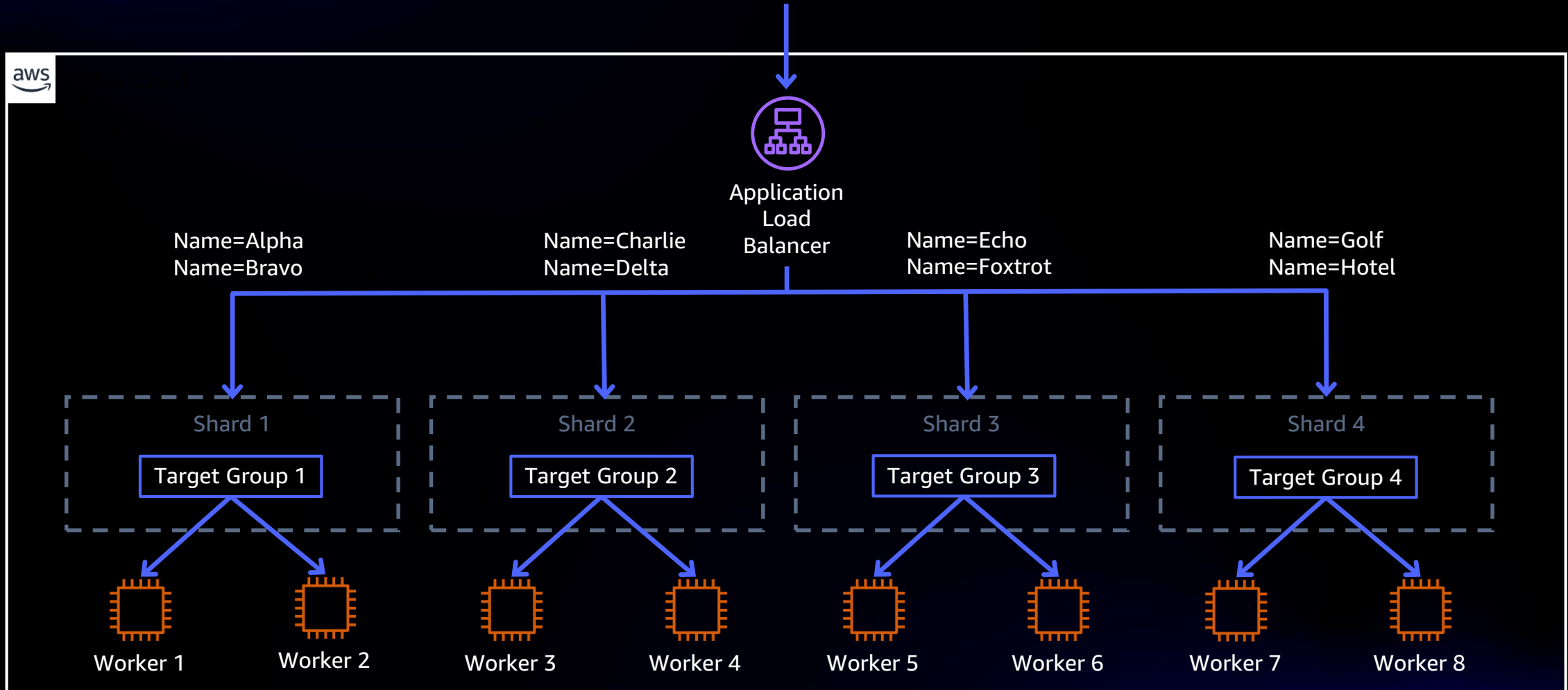
- Go to <https://go.aws/wa-shuffle-sharding>
- Go through sections
  1. Deploy workload
  2. Impact of failures
  3. Implement sharding
  4. Impact of failures – Sharding
- Regroup in 30 minutes



# Traditional architecture



# Architecture - Sharding

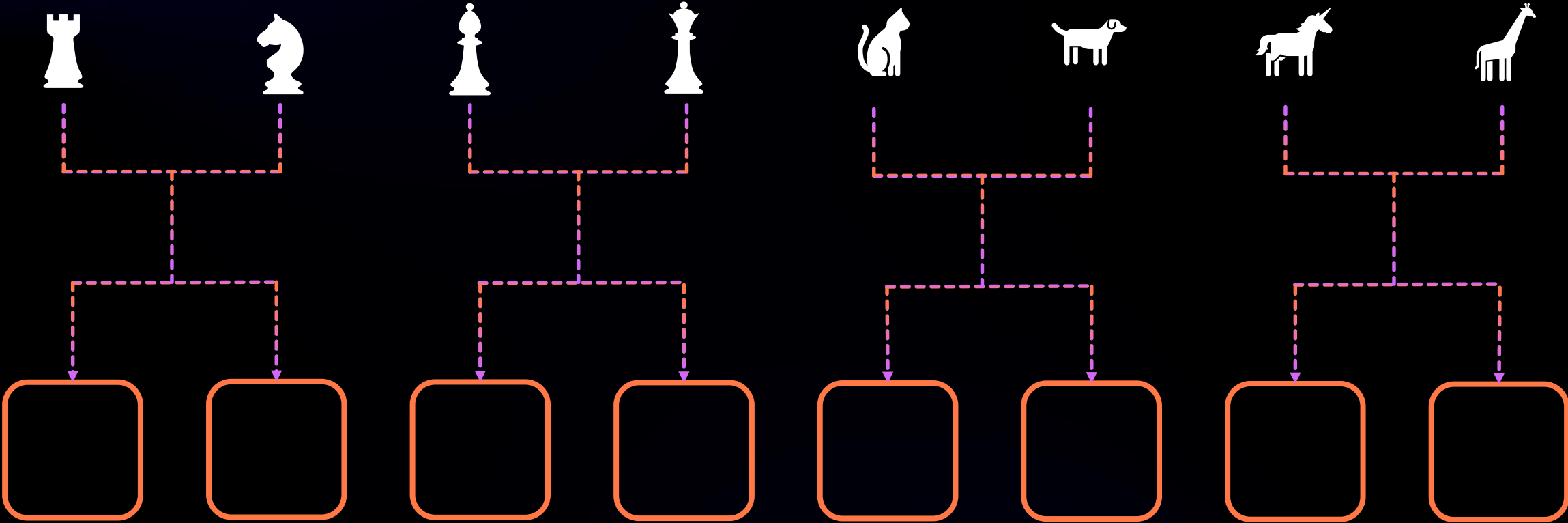


# Observations

- Traditional architecture – complete outage
- Sharding
  - Impact localized to customers on the same shard
  - 25% of customers affected

# The magic of math!

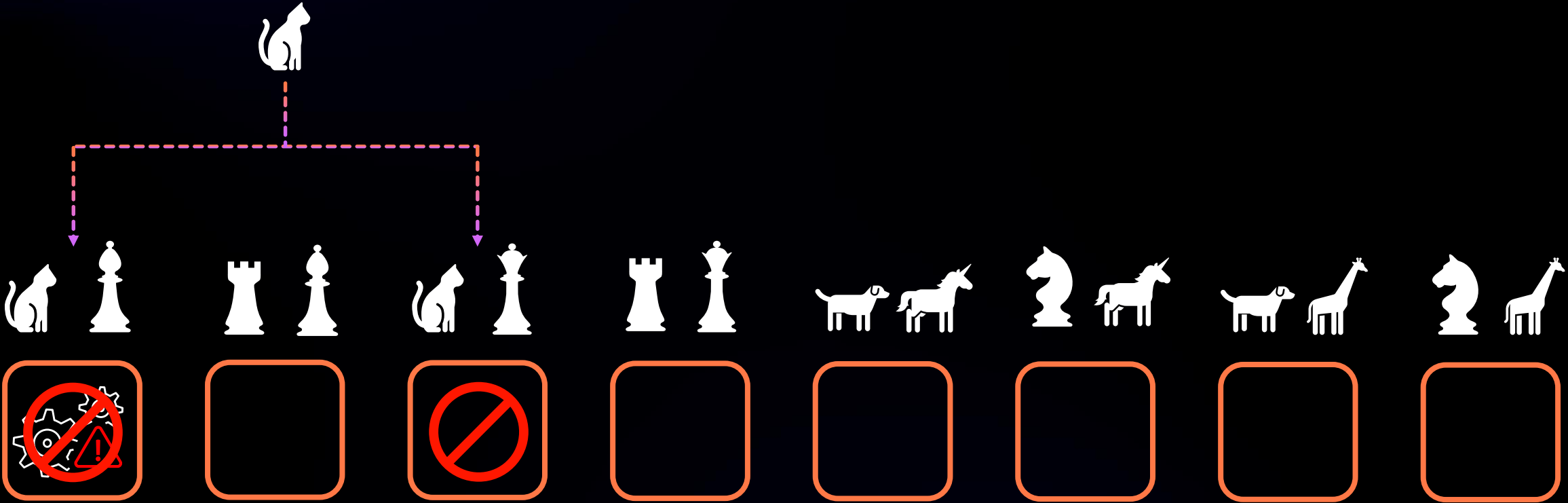
# Shuffle sharding



# Shuffle sharding



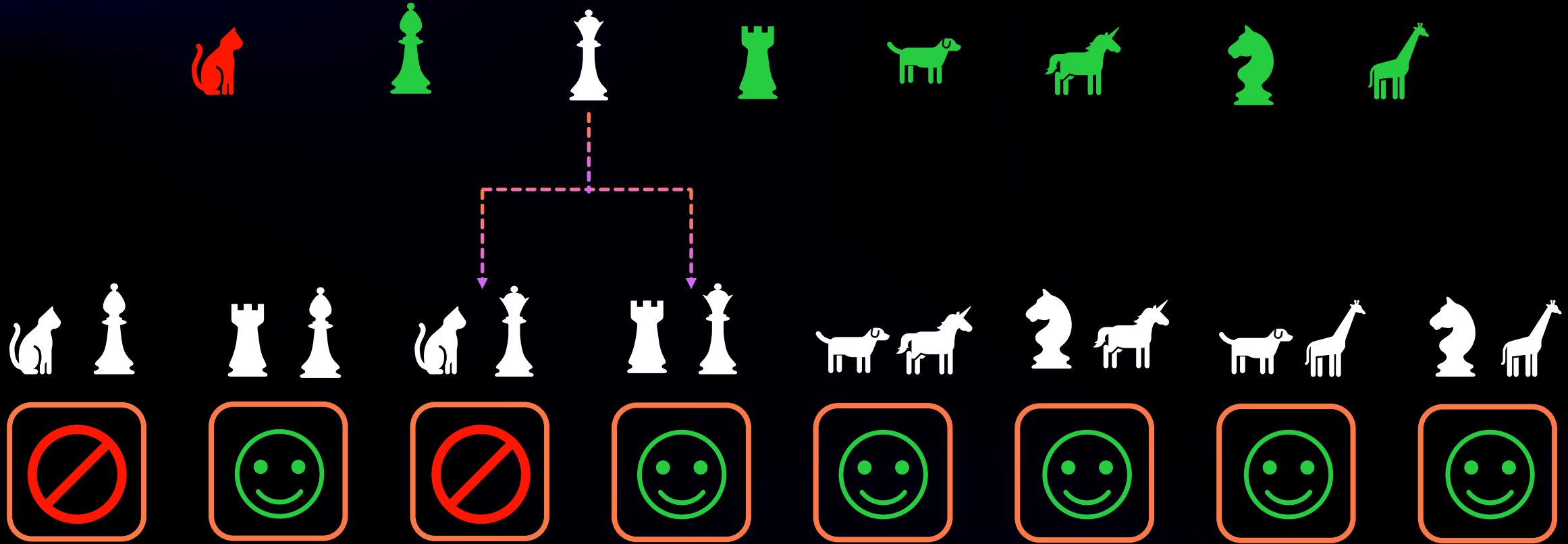
# Shuffle sharding



# Shuffle sharding



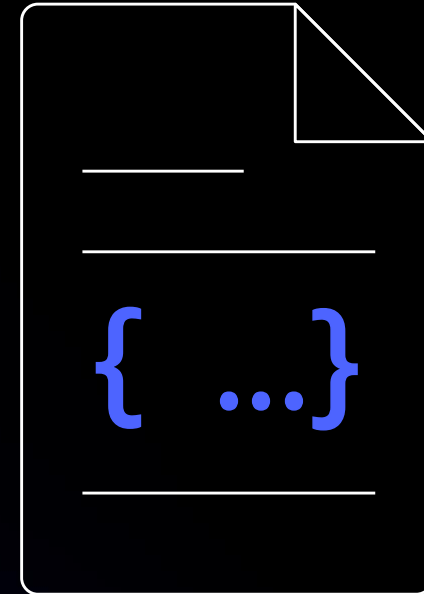
# Shuffle sharding



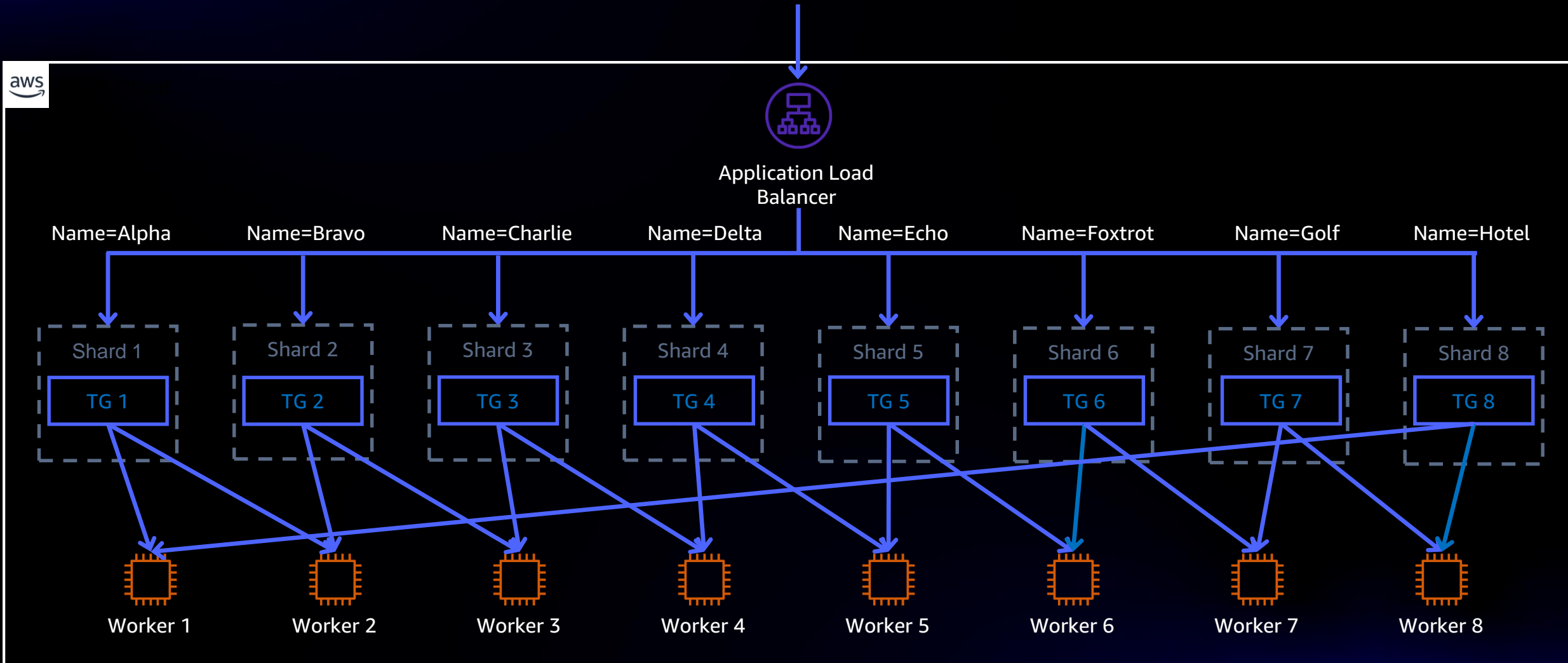
$$\text{Scope of impact} = \frac{\text{Customers}}{\text{Combinations}}$$

# Hands-on lab

- Go back to the lab
- Go through sections
  5. Implement shuffle sharding
  6. Impact of failures – shuffle sharding
- Regroup in 20 minutes



# Architecture – Shuffle sharding



# Observations

- Traditional architecture – complete outage
- Sharding
  - Impact localized to customers on the same shard
  - 25% of customers affected
- Shuffle sharding
  - Impact localized to customers having the same combination of nodes
  - 12.5% of customers affected

# Scope of impact – Shuffle sharding

N – number of worker nodes

S – shard size (workers per shard)

$$\text{Scope of impact} = \frac{1}{\text{Combinations}} \times 100$$

$$\text{Combinations} = \frac{N!}{S!(N-S)!}$$

$$\text{Scope of impact} = \frac{1}{28} \times 100$$

$$\text{Combinations} = \frac{8!}{2!(8-2)!}$$

3.6%

# Observations

$$\text{Impact} = \frac{1}{C(\text{\# of nodes, shard size})}$$



Overlap	% customers
0	53.6%
1	42.8%
2	3.6%

8 nodes, 2 nodes per shard

# Scope of impact – Shuffle sharding

$$\text{Scope of impact} = \frac{1}{\text{Combinations}} \times 100$$

$$\text{Scope of impact} = \frac{1}{8} \times 100$$

12.5%

# Observations

Overlap	% customers
0	77%
1	21%
2	1.8%
3	0.06%
4	0.0006%
5	0.0000013%

100 nodes, 5 nodes per shard



Amazon Route 53

2048 worker nodes, each shuffle shard uses 4 worker nodes

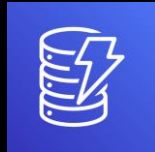
1 chance in 730 billion

# Implementation considerations

- Placement



Amazon Aurora

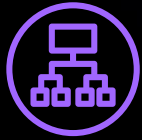


Amazon DynamoDB



# Implementation considerations

- Placement
- Router



Application Load  
Balancer



Amazon Route 53



# Implementation considerations

- Placement
- Router
- Control plane



AWS Cloud Development Kit  
(AWS CDK)



AWS CloudFormation



AWS CodePipeline



Amazon CloudWatch

# AWS Well-Architected – Reliability

## AWS Well-Architected Framework

[Add a link to your architectural design](#)

### REL 10. How do you use fault isolation to protect your workload? [Info](#)

Fault isolated boundaries limit the effect of a failure within a workload to a limited number of components. Components outside of the boundary are unaffected by the failure. Using multiple fault isolated boundaries, you can limit the impact on your workload.

Question does not apply to this workload [Info](#)

Select from the following

- Deploy the workload to multiple locations [Info](#)
- Automate recovery for components constrained to a single location [Info](#)
- Use bulkhead architectures [Info](#)



# AWS Well-Architected – Reliability

- 25 Regions and 81 Availability Zones
- Amazon EC2 Auto Scaling, Amazon RDS failover
- Infrastructure as code
- Shuffle sharding



# Resources

## **Amazon Builders' Library: Workload isolation using shuffle sharding**

How Amazon Route 53 achieves resiliency and provides a 100% SLA using shuffle sharding

## **Amazon Route 53 Infima**

Library for managing service-level fault isolation using Amazon Route 53

## **AWS Well-Architected Framework**

Curated list of best practices for designing, building, and operating workloads on AWS

## **AWS Well-Architected Labs**

Hands-on labs to help you learn how to implement architectural best practices



# Thank you!

Julien Lépine

Twitter: @julien\_lepine

Mahanth Jayadeva

Twitter: @MahanthJayadeva

