
AWS
re:Invent

ARC303

Reliability, consistency, and confidence through immutability

Adrian Hornsby
Principal Developer Advocate
Amazon Web Services

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

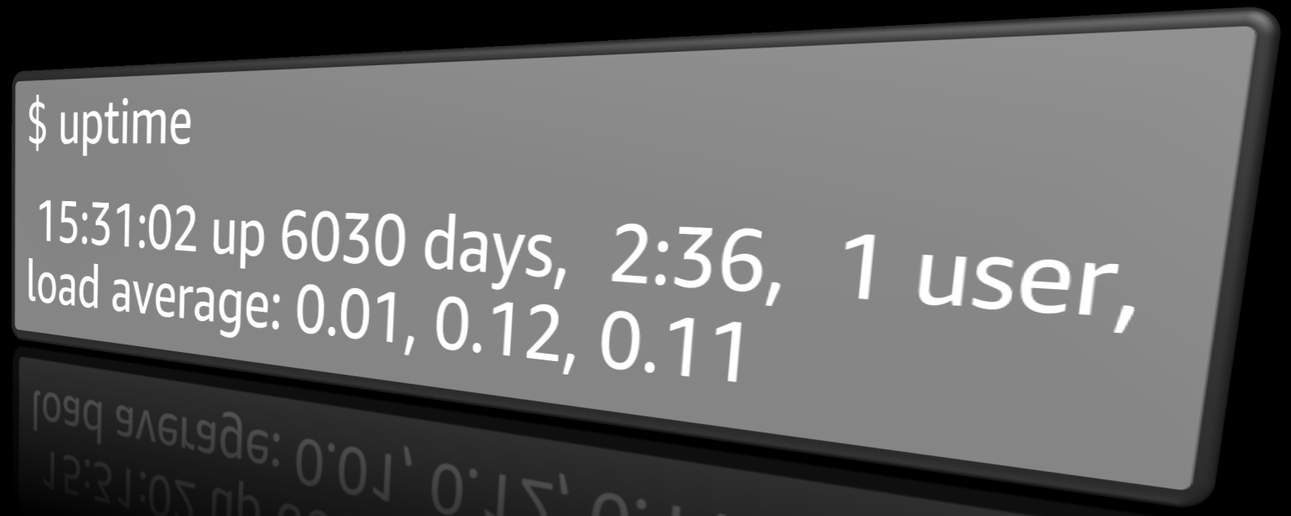
Immutable:
not capable of or susceptible to change

Traditional infrastructures

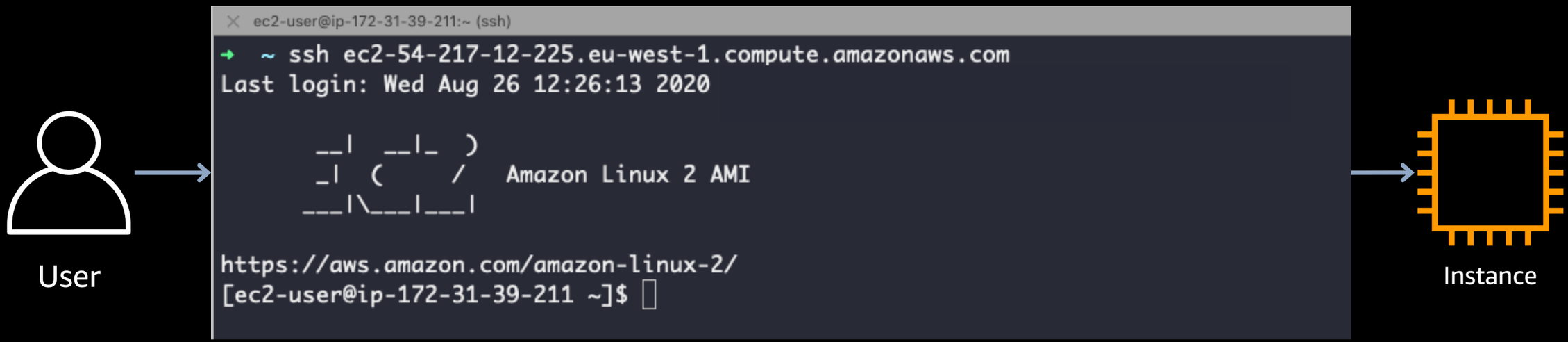


© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

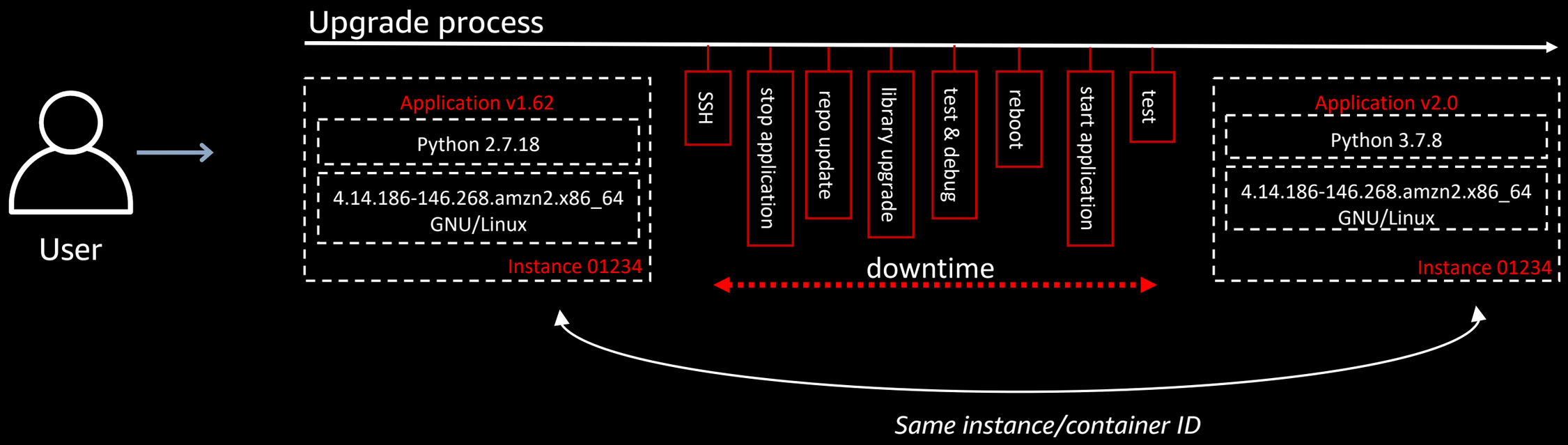
Uptime


$ uptime
15:31:02 up 6030 days, 2:36, 1 user,
load average: 0.01, 0.12, 0.11

ssh

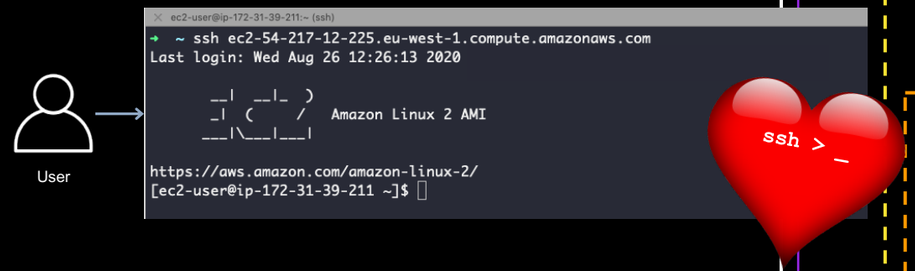

ec2-user@ip-172-31-39-211:~ (ssh)
~ ssh ec2-54-217-12-225.eu-west-1.compute.amazonaws.com
Last login: Wed Aug 26 12:26:13 2020
__| __|_)
_| (/ Amazon Linux 2 AMI
___|___|___|
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-39-211 ~]$
User
Instance

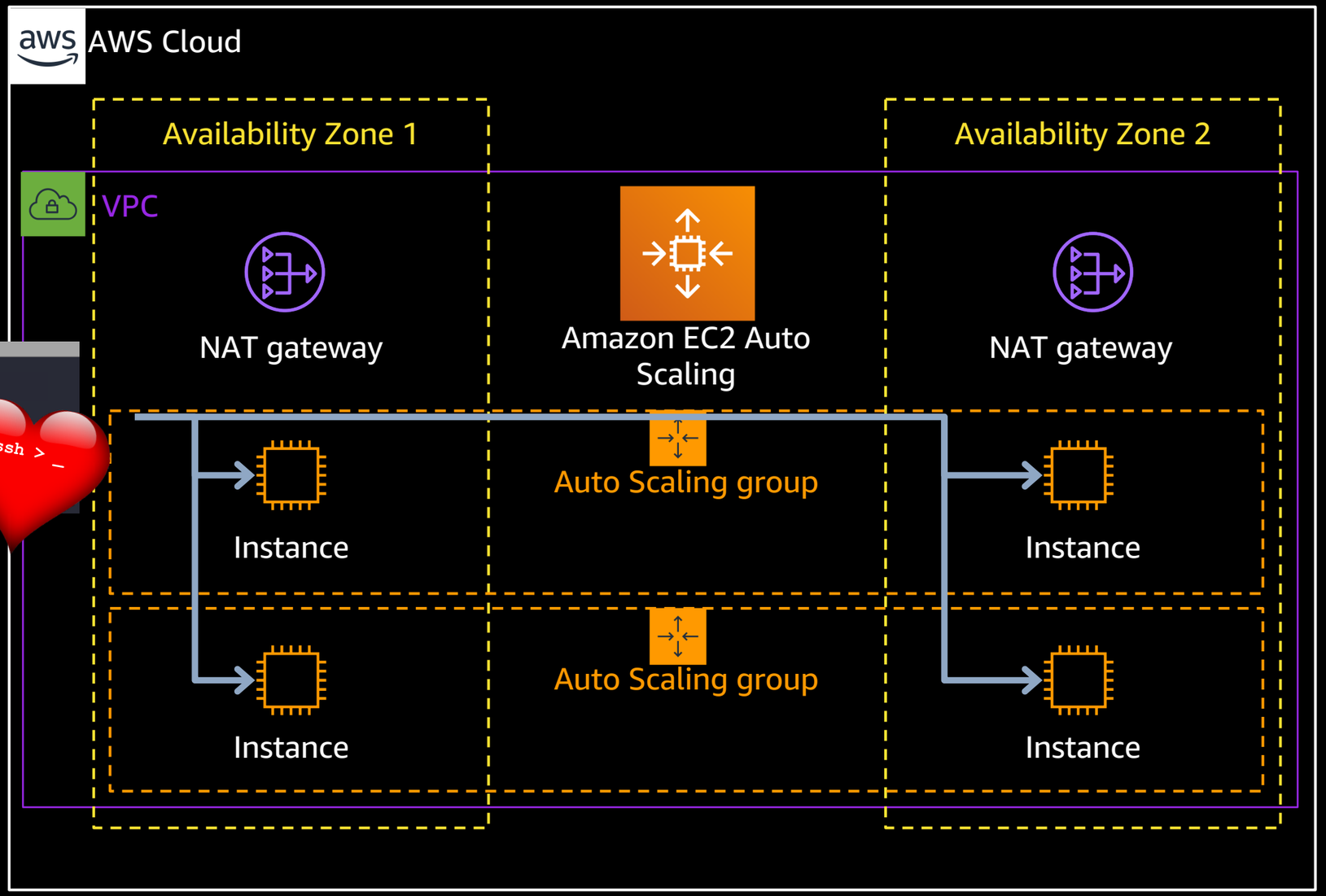
In-place server update



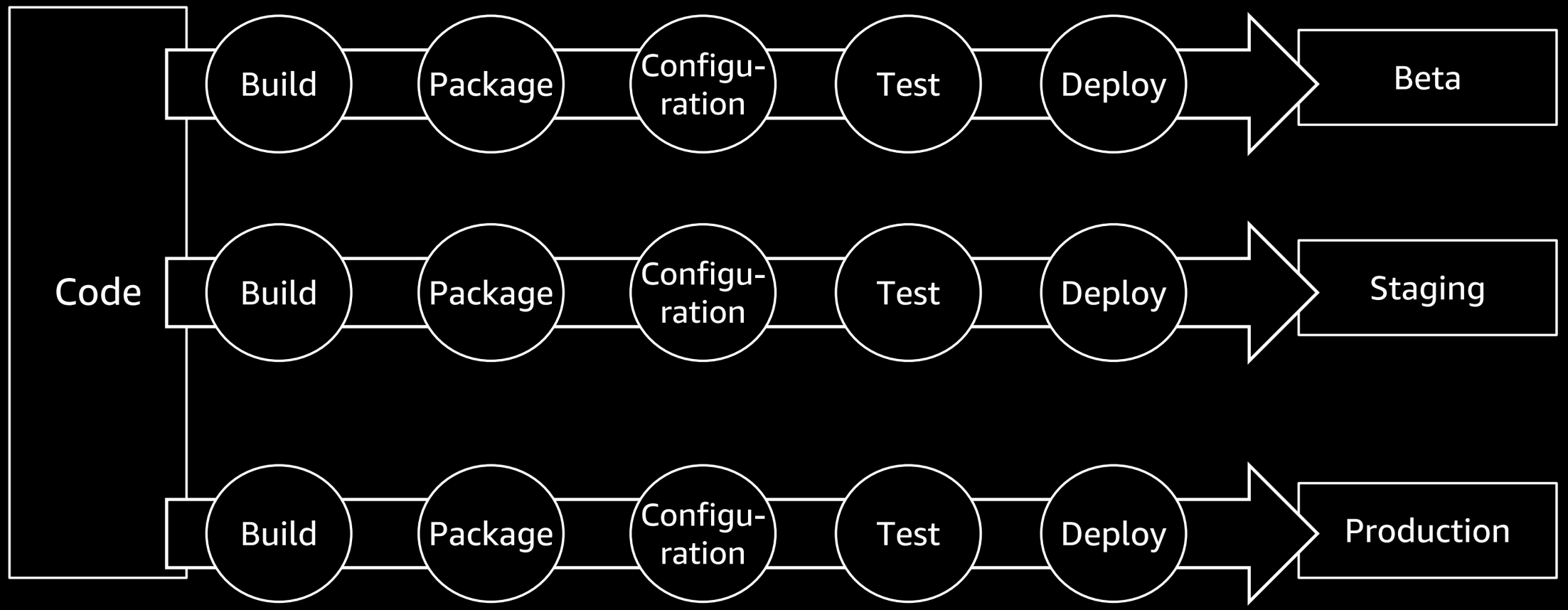
ssh > _ love syndrome

Leads to configuration drifts, and more





Mutable deployments pipelines



Running commands on Linux instance at launch with user data and shell scripts

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum install -y httpd mariadb-server
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

```
>>> pip install -r requirements.txt
>>> npm install
>>> docker build
```



NPM ERR!

How one programmer broke the internet by deleting a tiny piece of code

```
1  module.exports = leftpad;
2  function leftpad (str, len, ch) {
3    str = String(str);
4    var i = -1;
5    if (!ch && ch !== 0) ch = ' ';
6    len = len - str.length;
7    while (++i < len) {
8      str = ch + str;
9    }
10   return str;
11 }
12
13
14
15
16
17
18
19
20
```

By **Keith Collins**
Tech Reporter

March 27, 2016 · This article is more than 2 years old.

A man in Oakland, California, disrupted web development around the world last week by deleting 11 lines of code.

The story of how 28-year-old Azer Koçulu briefly broke the internet



A discussion about the breaking of the Internet

Mike Roberts Follow
Mar 23, 2016 · 5 min read

Hey everyone — I'm the head of messenger at Kik. I wish this didn't have to be my first post on Medium, but open source is something that I care about. I've published a few meager open source projects in the past, things that aren't groundbreaking but that I thought might be useful to other people, and I rely on countless others every day.

I found out about this problem like a lot of you, when our builds started failing because we use the extremely helpful JSCS. Through a long chain of dependencies, JSCS relied on left-pad@0.0.3, which was removed by the author yesterday. Our team was confused at the time as well.

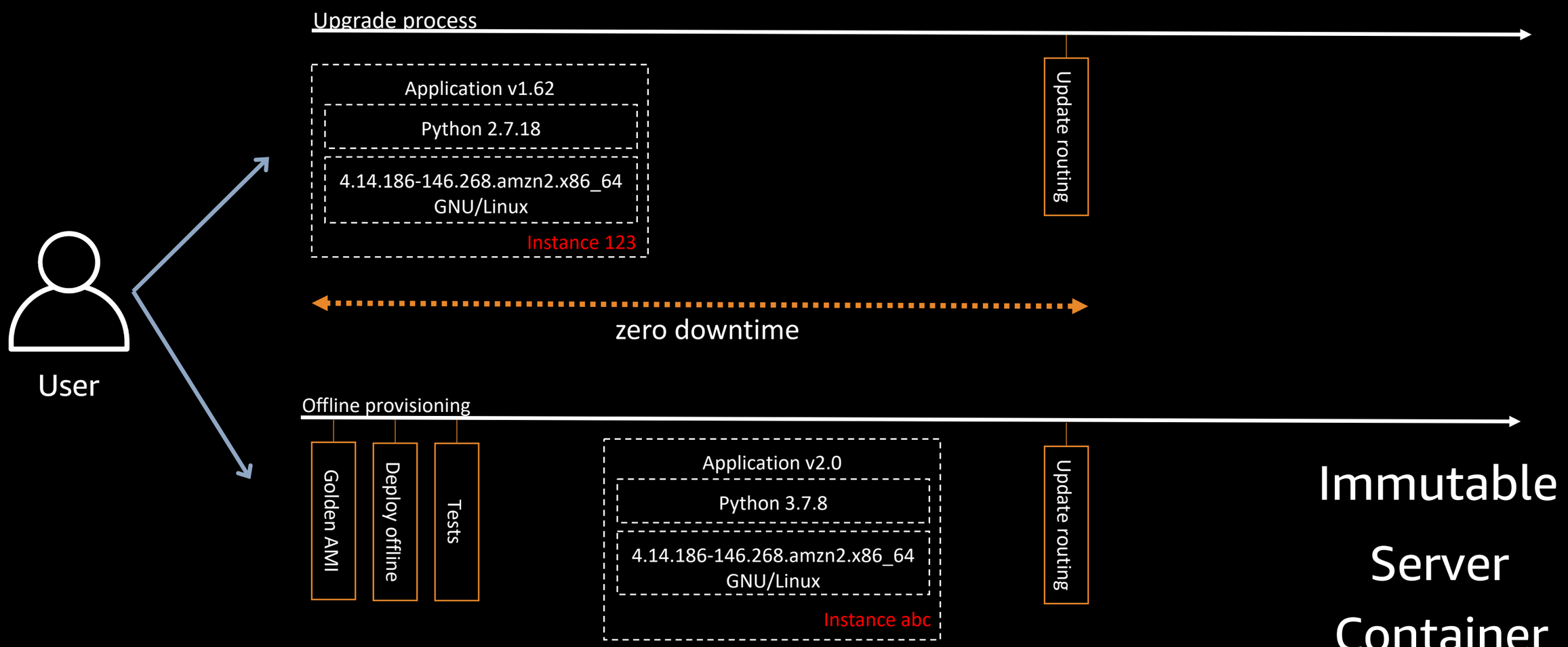
https://medium.com/@mproberts/a-discussion-about-the-breaking-of-the-internet
https://qz.com/646467/how-one-programmer-broke-the-internet-by-deleting-a-tiny-piece-of-code/

Immutable infrastructure



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Immutable



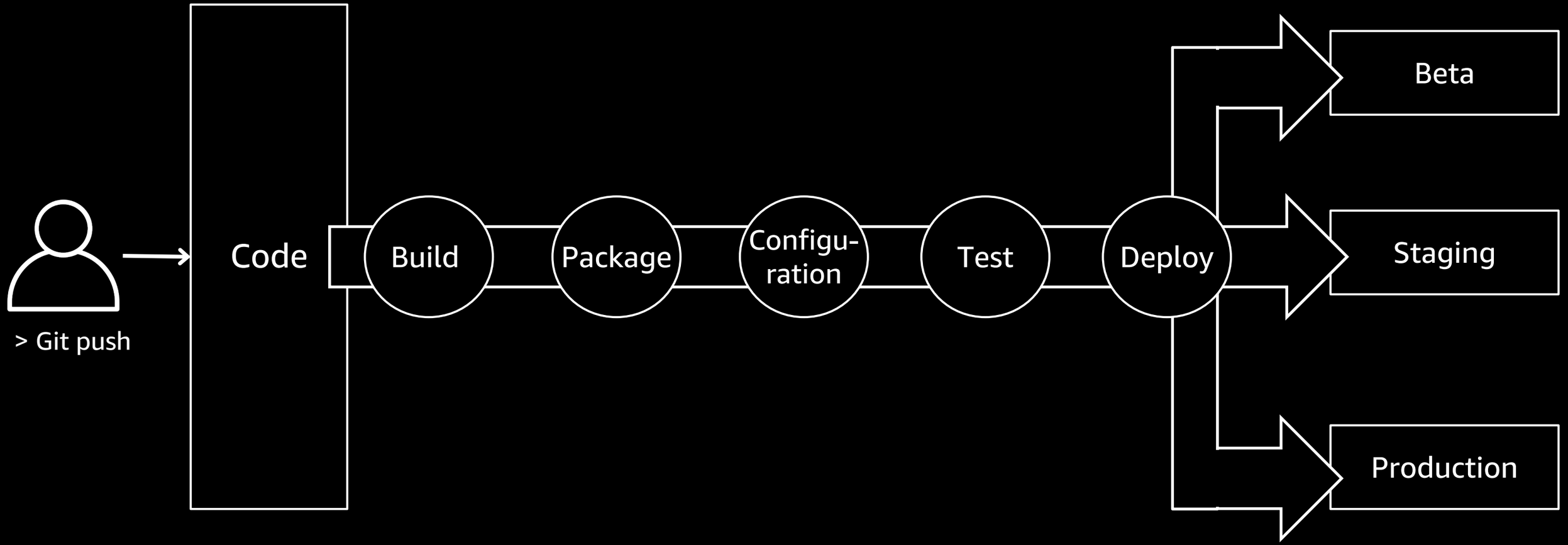
Immutable

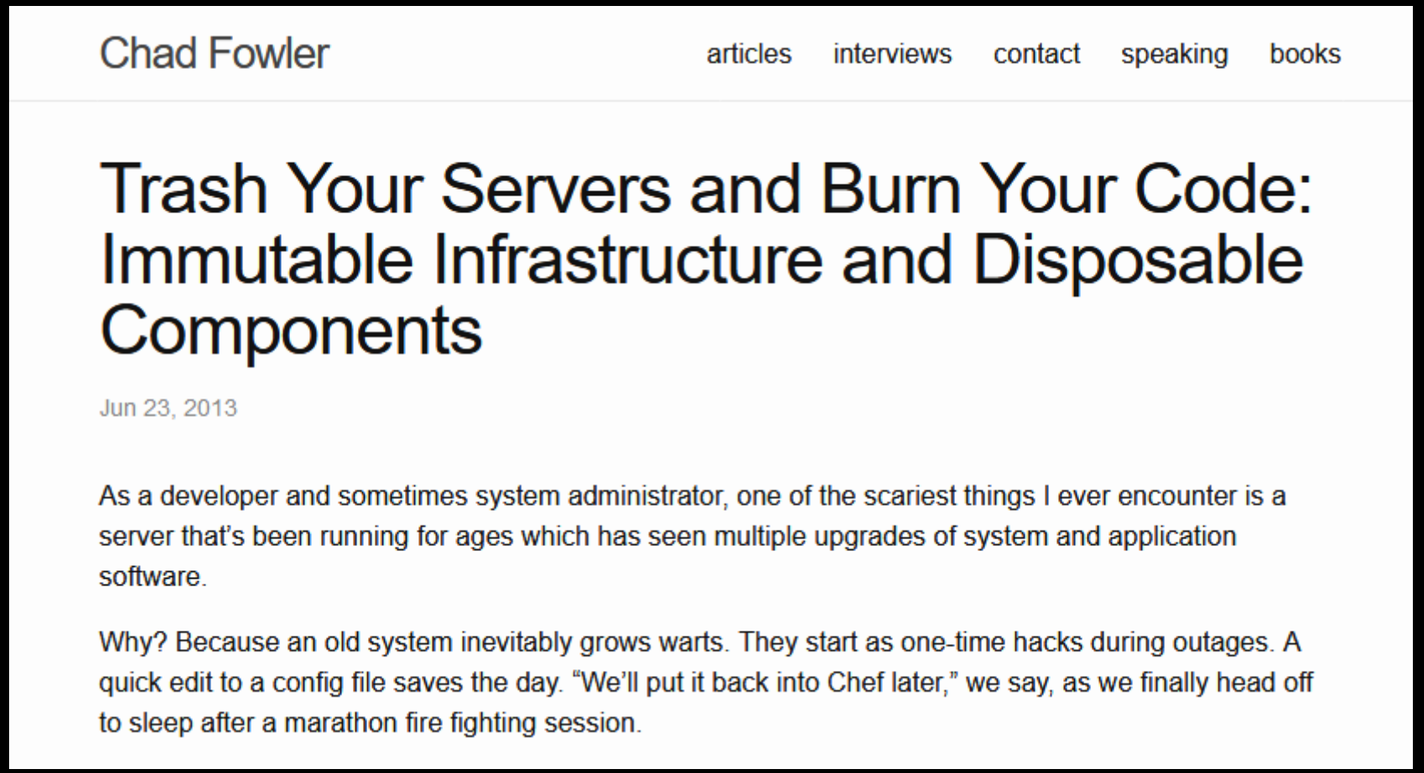
Server

Container

Function

Immutable deployments





http://chadfowler.com/2013/06/23/immutable-deployments.html

Benefits of immutable deployment

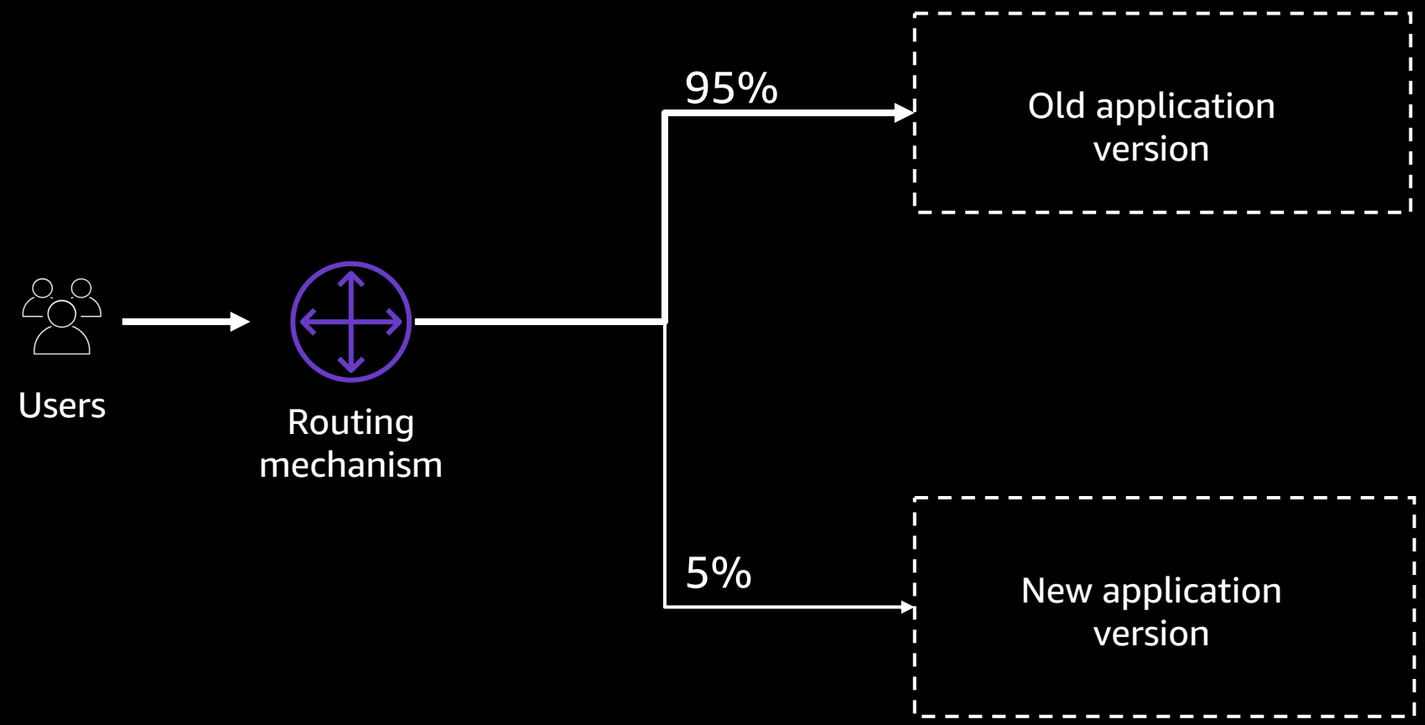
1. Reduction in configuration drifts
2. Simplified deployments
3. Reliable atomic deployments
4. Safer deployments with fast rollback and recovery processes
5. Consistent testing and debugging environments
6. Increased scalability
7. Simplified toolchain
8. Increased security

Security considerations

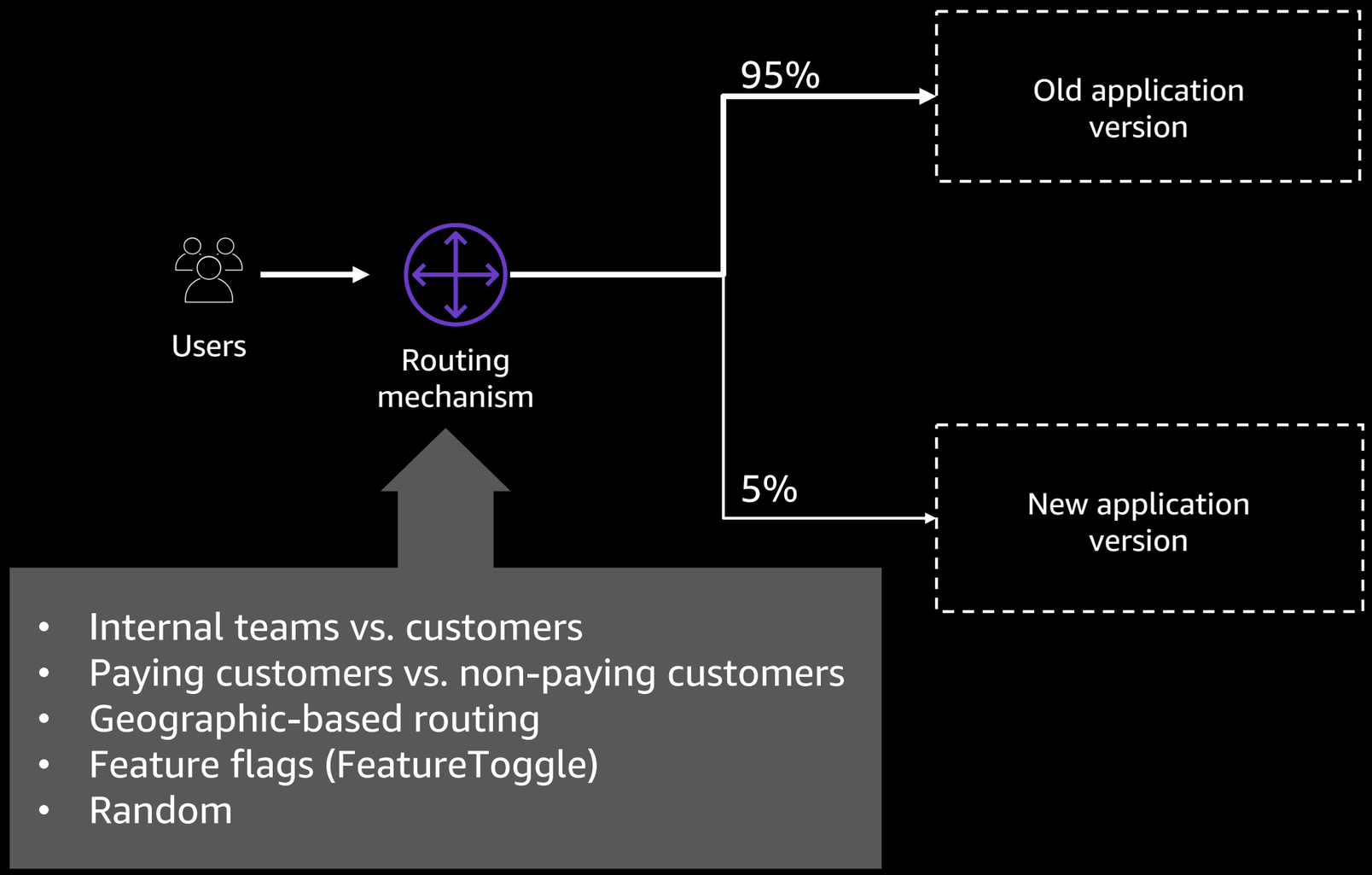
Mutability is one of the most critical attack vectors for cyber crimes.

DETECT. NUKE. REPLACE.

Canary deployment



Canary deployment



Canary deployments on AWS



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

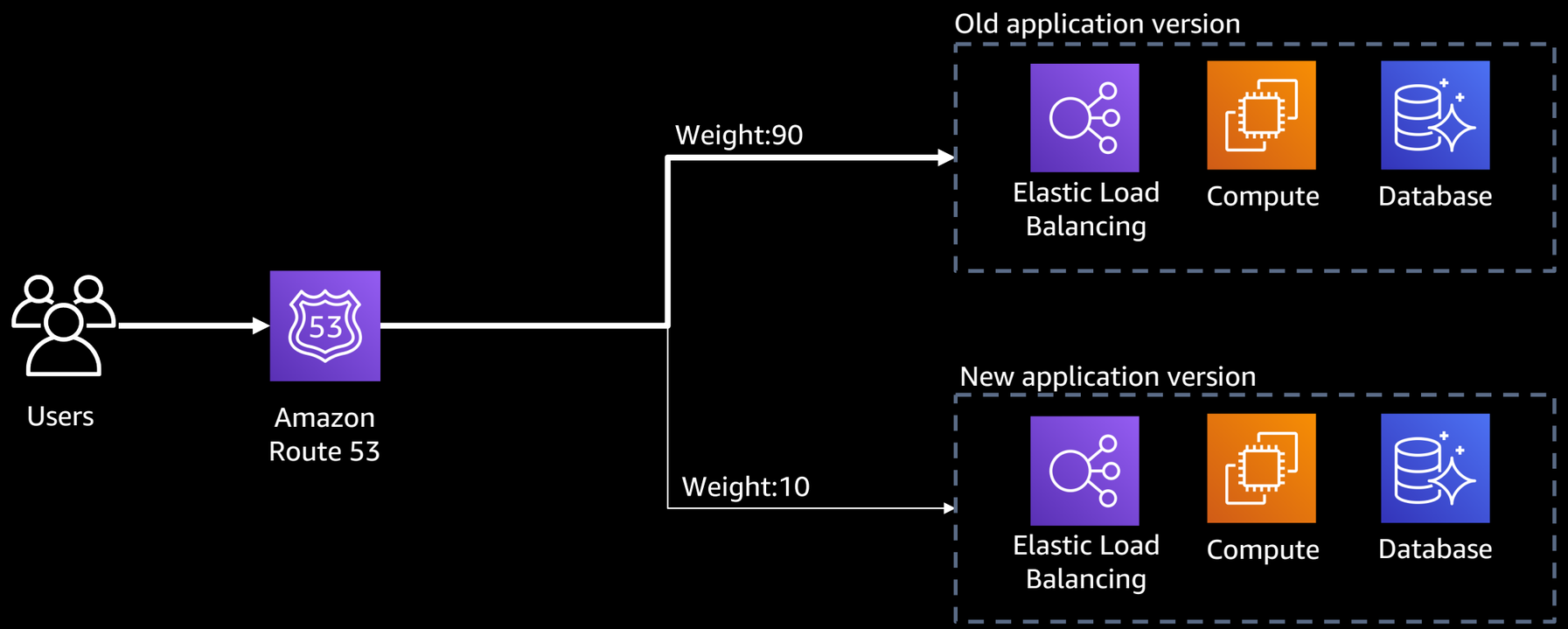
Canary deployment on AWS

1. Route 53 with weighted routing policy
2. Rolling Deployments for Auto Scaling Groups
3. Load balancer with ELB weighted target groups
4. API Gateway release deployments
5. AWS Lambda alias traffic shifting [DEMO]

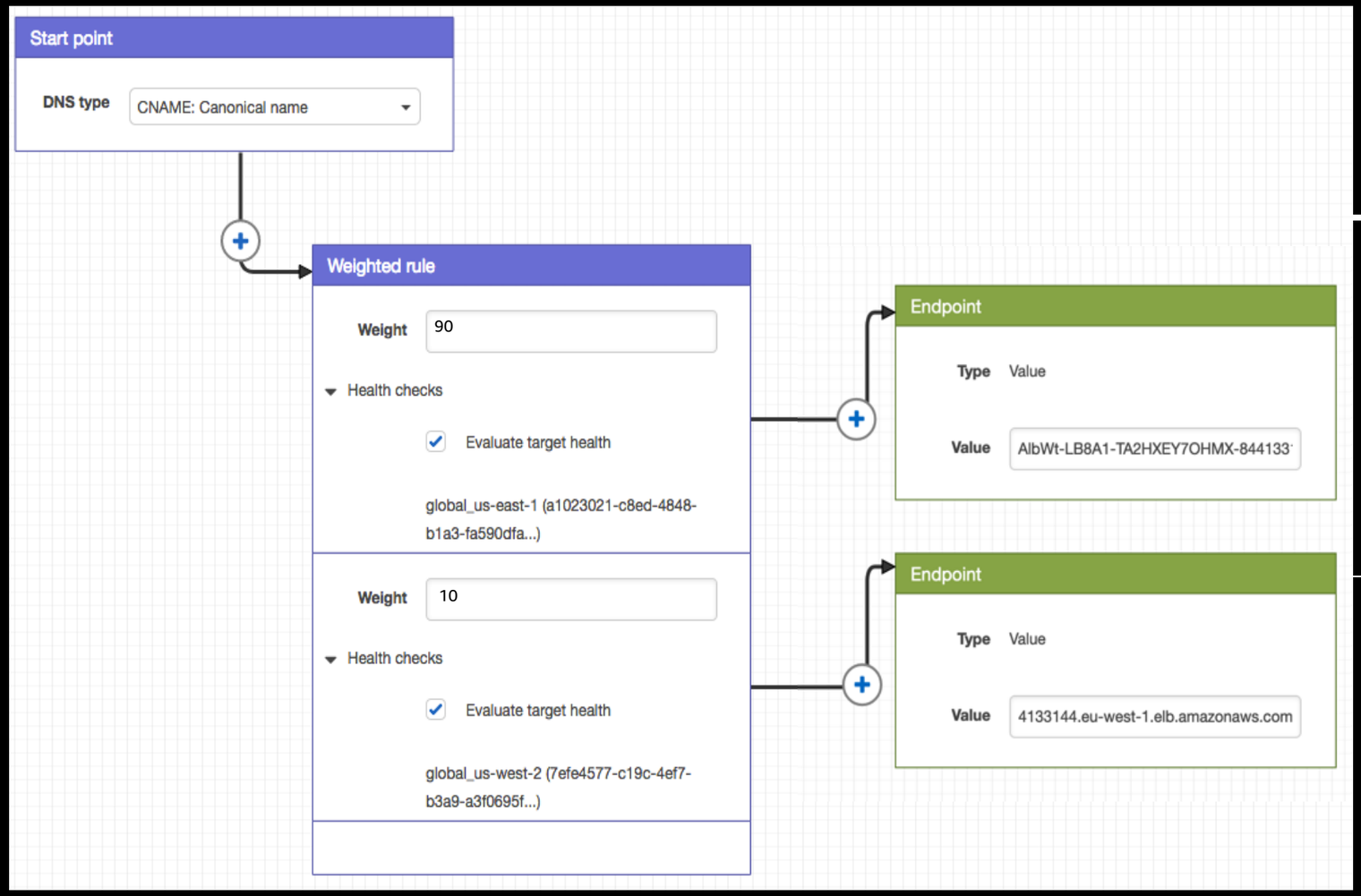


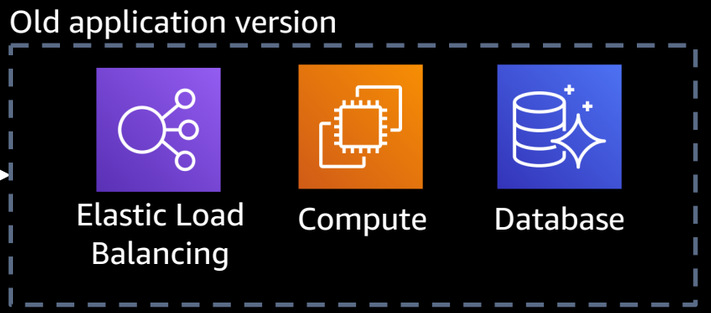
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

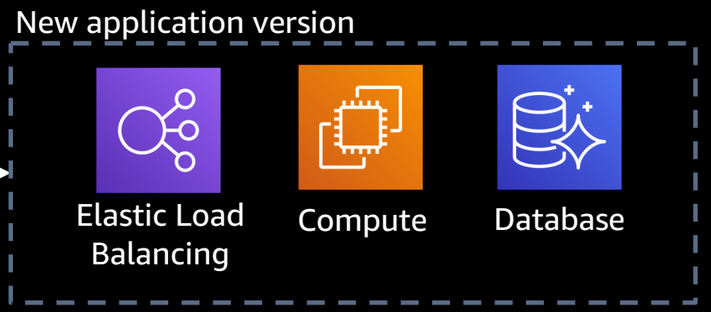
1. Route 53 with weighted routing policy



1. Route 53 with weighted routing policy



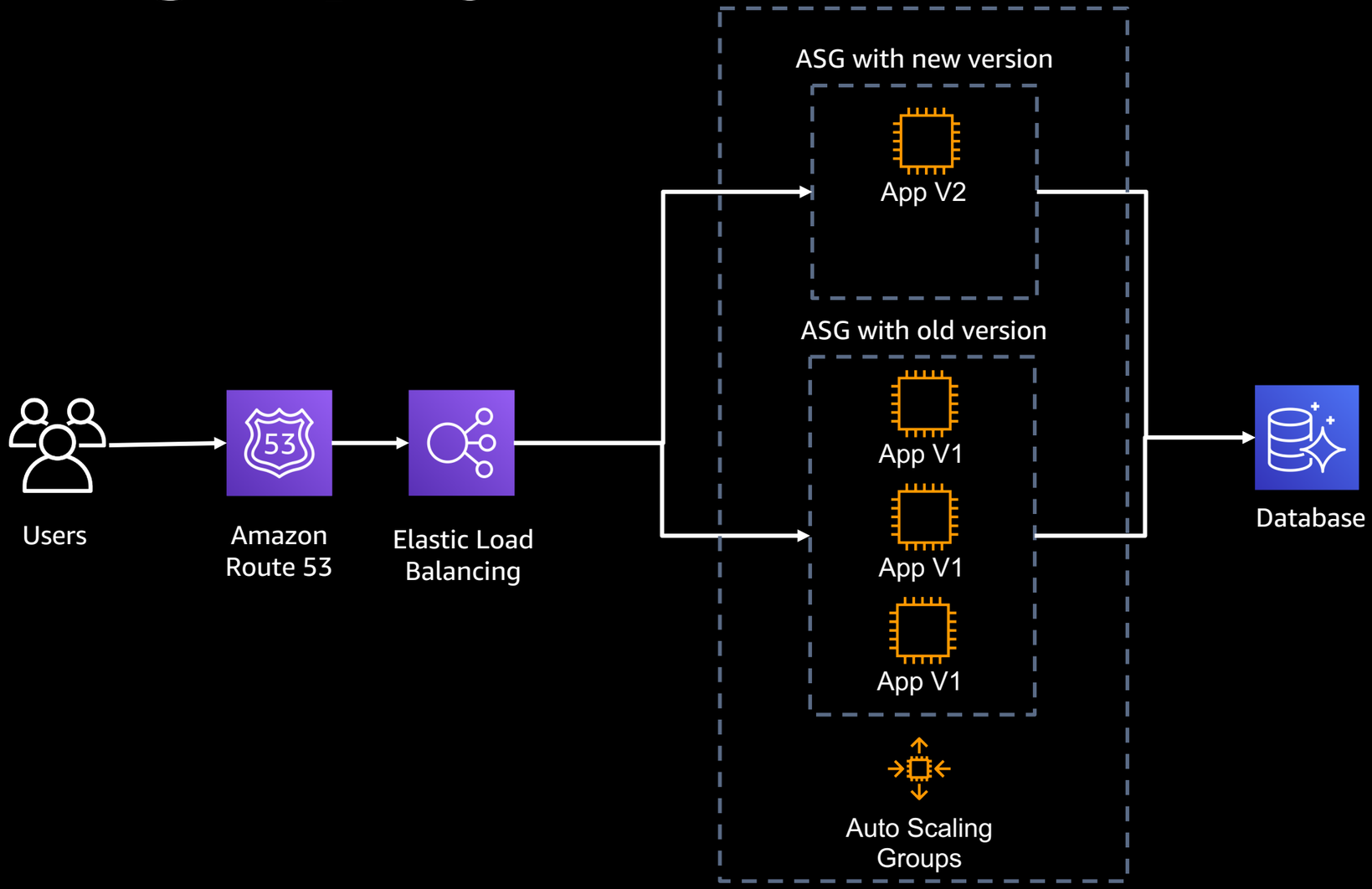






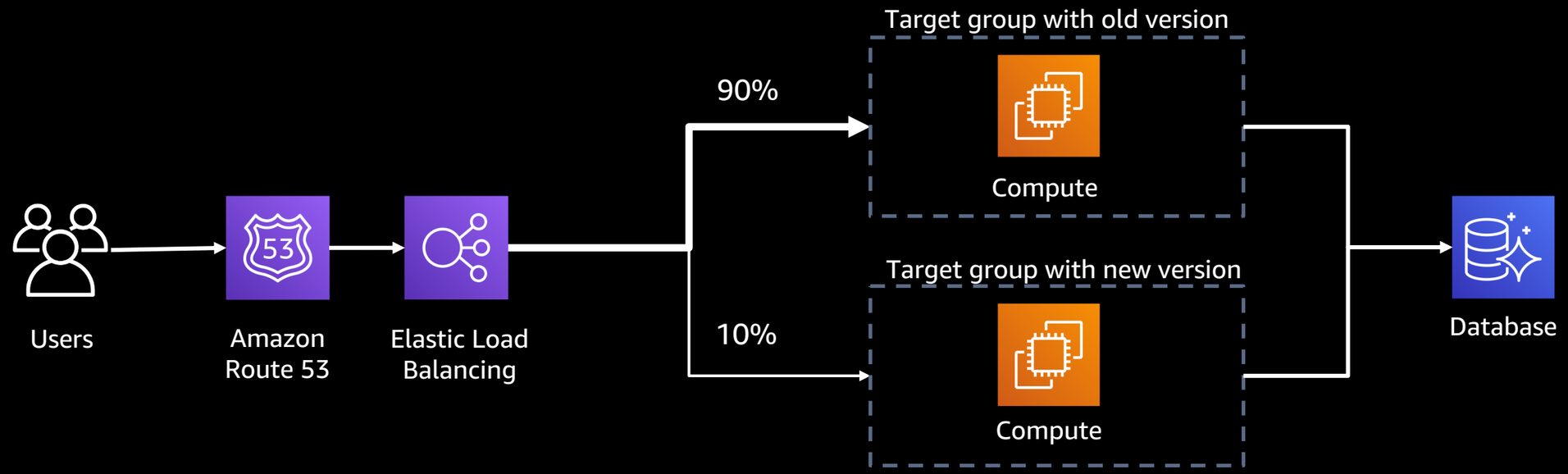
DNS Propagation!

2. Rolling deployments for auto scaling groups



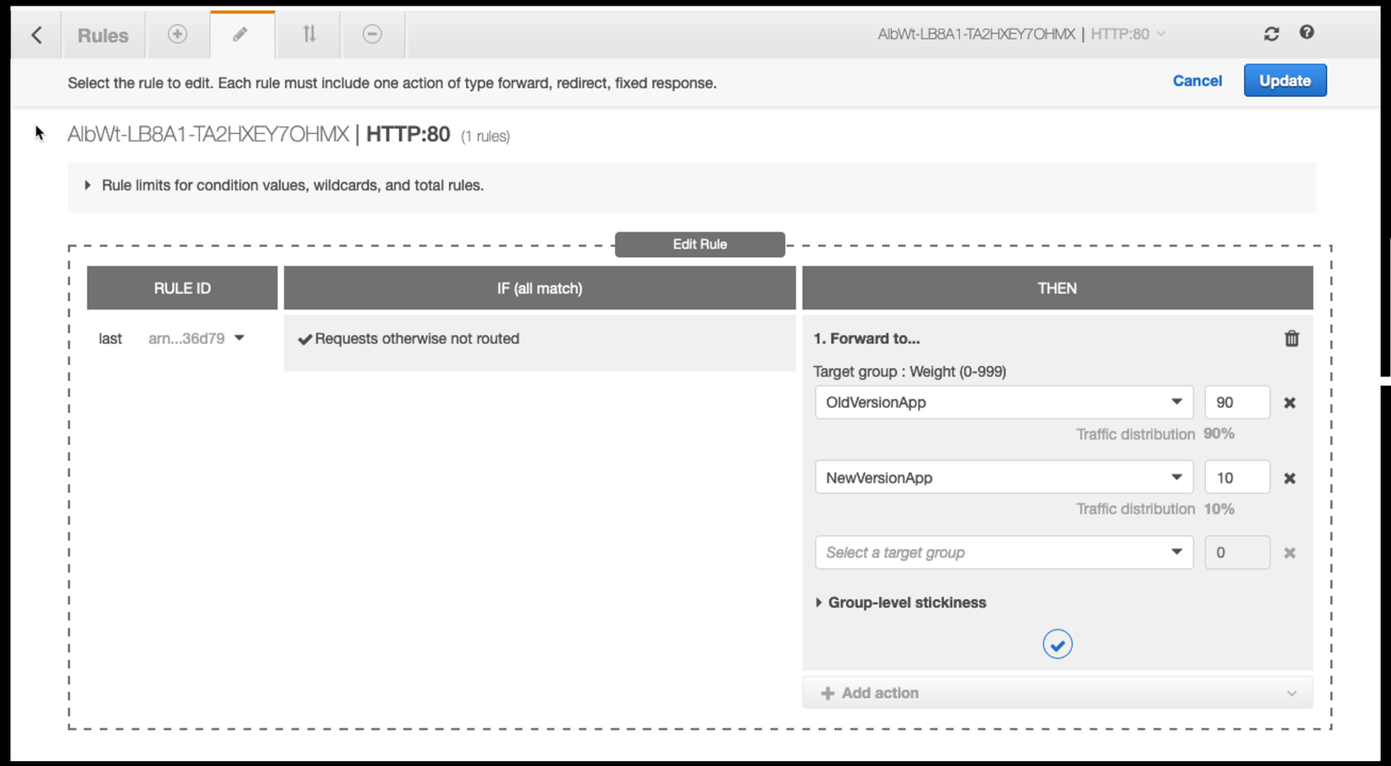
https://aws.amazon.com/blogs/aws/three-new-features-for-aws-cloudformation/
https://engineering.klarna.com/simple-canary-releases-in-aws-how-and-why-bf051a47fb3f

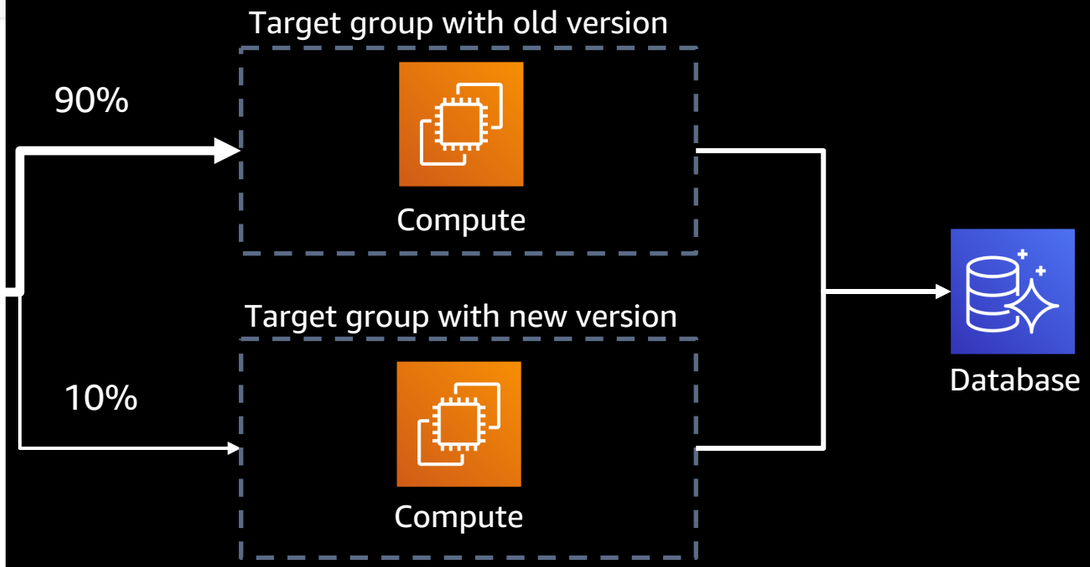
3. Application load balancer and weighted target groups



https://aws.amazon.com/blogs/aws/new-application-load-balancer-simplifies-deployment-with-weighted-target-groups/

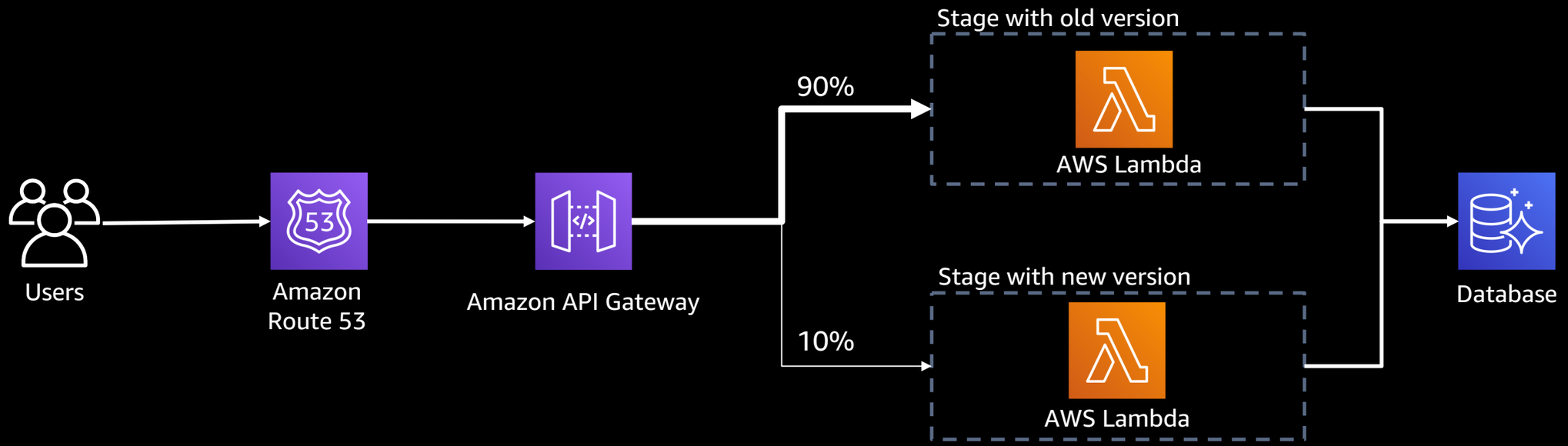
3. Application load balancer and weighted target groups



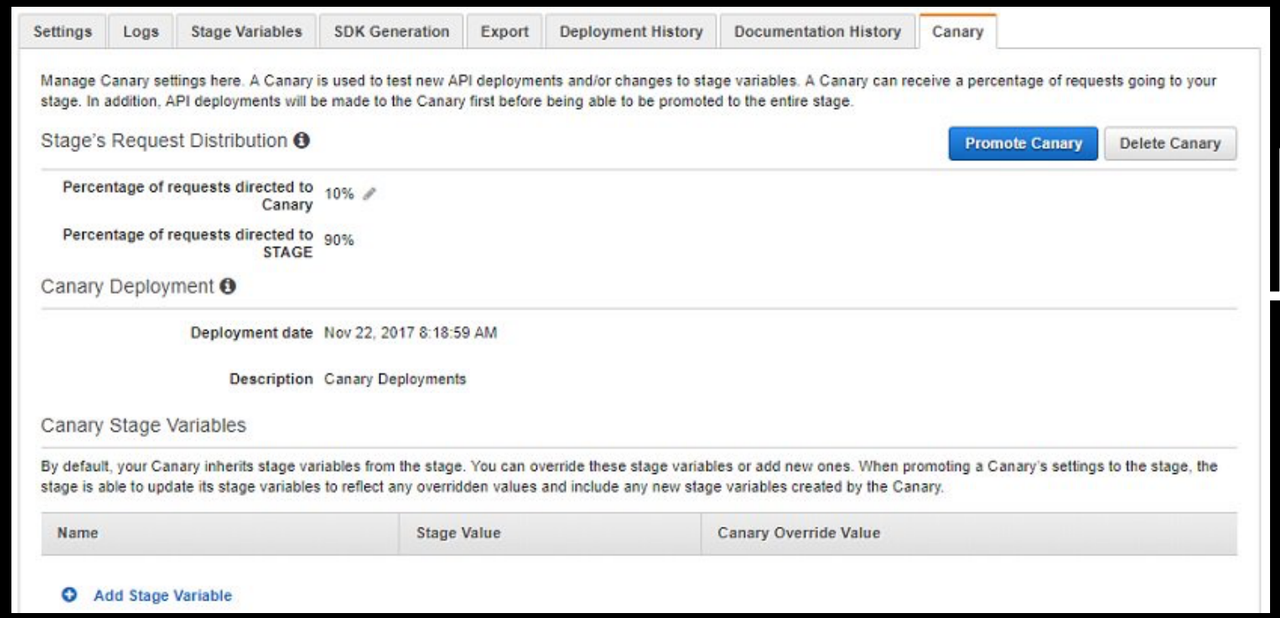


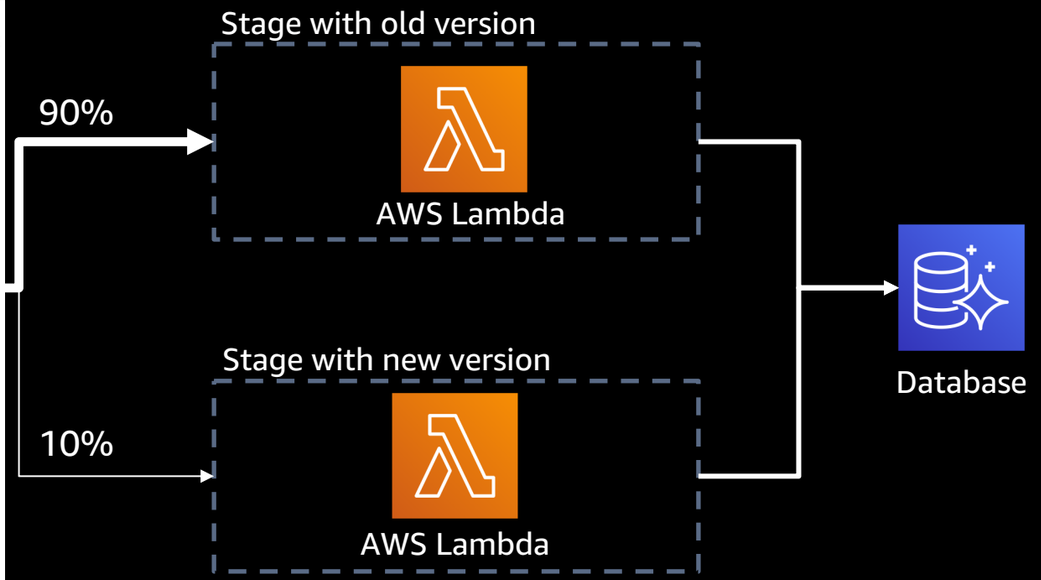
https://aws.amazon.com/blogs/aws/new-application-load-balancer-simplifies-deployment-with-weighted-target-groups/

4. API Gateway release deployments

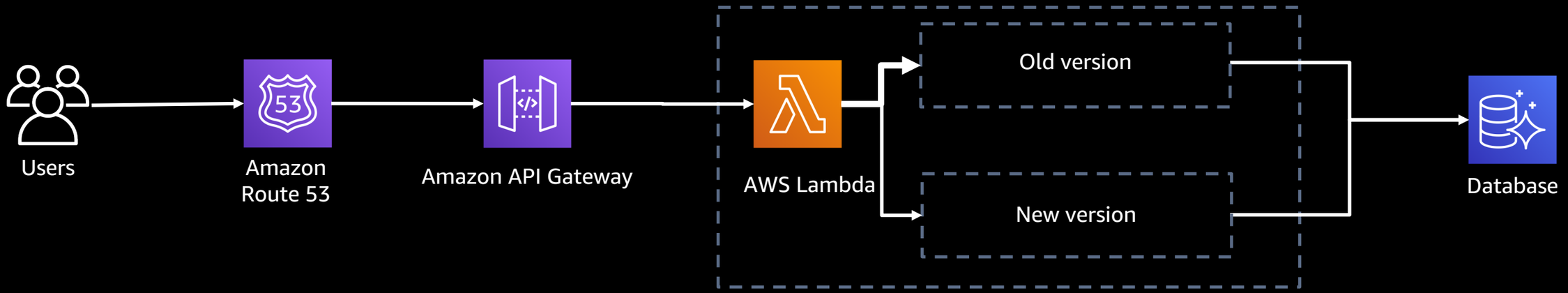


4. API Gateway release deployments

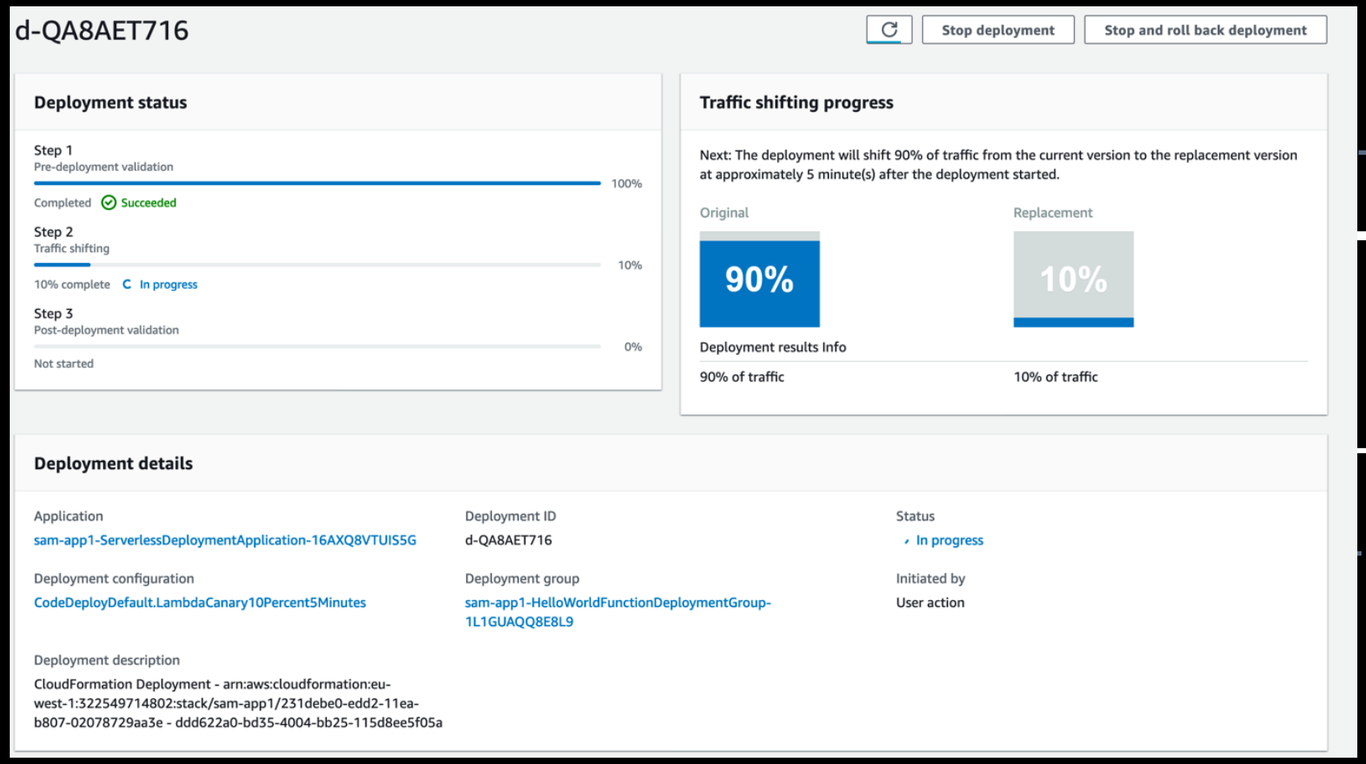


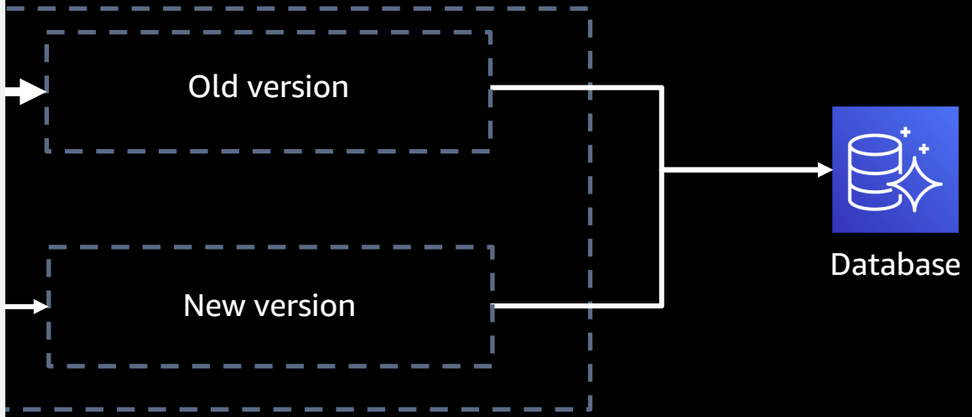


5. AWS Lambda alias traffic shifting



5. AWS Lambda alias traffic shifting





*via CodeDeploy

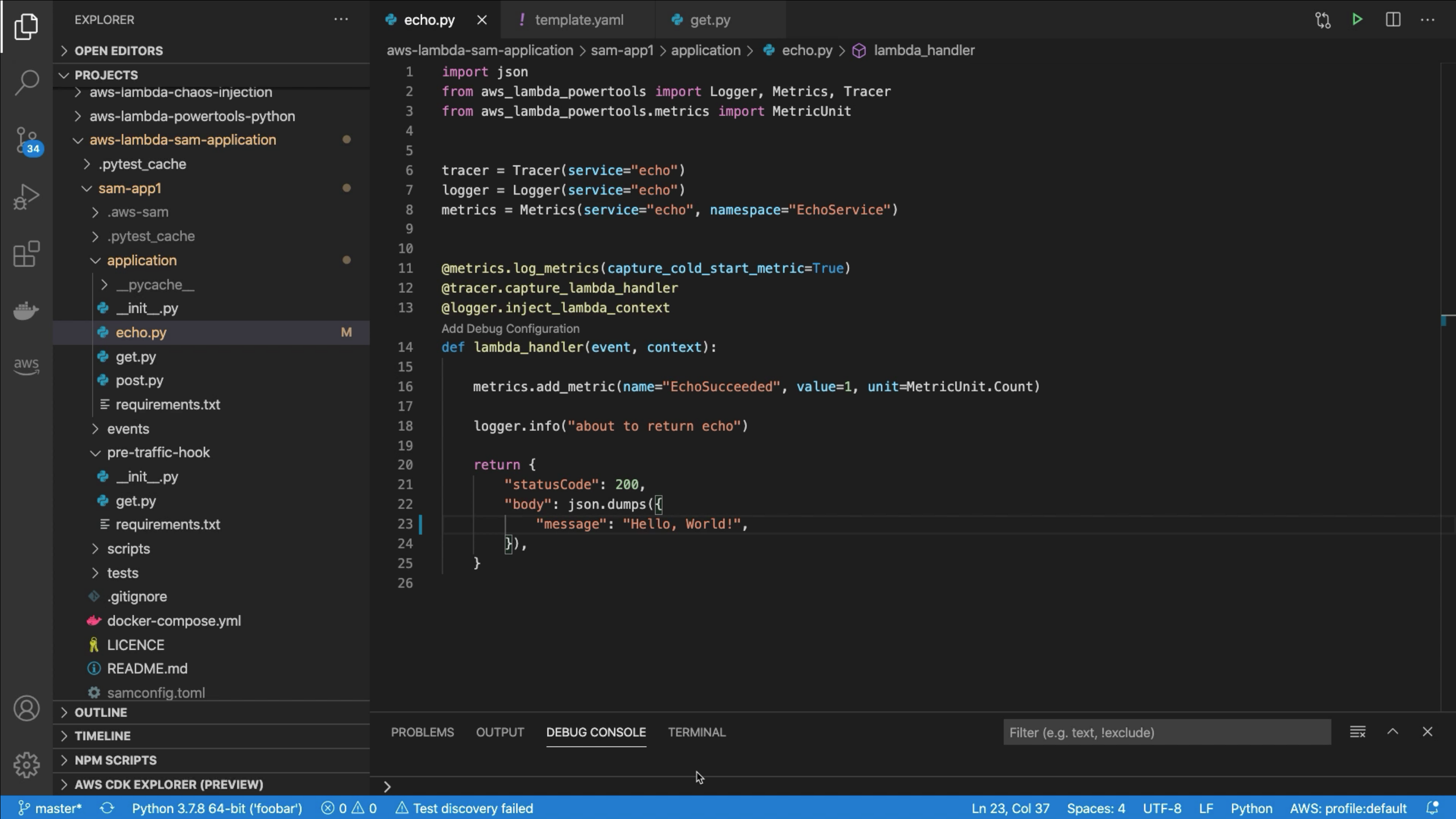
Demo Lambda traffic shifting with CodeDeploy

https://github.com/adhorn/aws-lambda-sam-application



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.


EXPLORER
OPEN EDITORS
PROJECTS
aws-lambda-chaos-injection
aws-lambda-powertools-python
aws-lambda-sam-application
.pytest_cache
sam-app1
.aws-sam
.pytest_cache
application
__pycache__
__init__.py
echo.py
M
get.py
post.py
requirements.txt
events
pre-traffic-hook
__init__.py
get.py
requirements.txt
scripts
tests
.gitignore
docker-compose.yml
LICENCE
README.md
samconfig.toml
OUTLINE
TIMELINE
NPM SCRIPTS
AWS CDK EXPLORER (PREVIEW)
echo.py
template.yaml
get.py
aws-lambda-sam-application > sam-app1 > application > echo.py > lambda_handler
1 import json
2 from aws_lambda_powertools import Logger, Metrics, Tracer
3 from aws_lambda_powertools.metrics import MetricUnit
4
5
6 tracer = Tracer(service="echo")
7 logger = Logger(service="echo")
8 metrics = Metrics(service="echo", namespace="EchoService")
9
10
11 @metrics.log_metrics(capture_cold_start_metric=True)
12 @tracer.capture_lambda_handler
13 @logger.inject_lambda_context
Add Debug Configuration
14 def lambda_handler(event, context):
15
16 metrics.add_metric(name="EchoSucceeded", value=1, unit=MetricUnit.Count)
17
18 logger.info("about to return echo")
19
20 return {
21 "statusCode": 200,
22 "body": json.dumps({
23 "message": "Hello, World!",
24 }),
25 }
26
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Filter (e.g. text, !exclude)
master*
Python 3.7.8 64-bit ('foobar')
0 0
Test discovery failed
Ln 23, Col 37
Spaces: 4
UTF-8
LF
Python
AWS: profile:default

Architecture resources

AWS Well-Architected Framework

The official best practices for Architecting in the AWS Cloud

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

AWS Well-Architected Labs

Hands-on labs to help you learn, measure, and build using architectural best practices

https://wellarchitectedlabs.com/

AWS Architecture Center

Official AWS repository for all Architecture resources

https://aws.amazon.com/architecture

AWS Solutions Library

Vetted reference implementations and Well-Architected patterns

https://aws.amazon.com/solutions/



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Adrian Hornsby

Principal Developer Advocate
Amazon Web Services

@adhorn

https://medium.com/@adhorn

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.