
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

ADM302

Running high-throughput, real-time ad platforms in the cloud

Victor Gershkovich
Data Platform Team Lead, Real-Time Infrastructure
AppsFlyer

Gerry Louw
Head of Technology, Advertising and Marketing
Amazon Web Services

Akhil Aendapally
Sr. Solutions Architect, Advertising and Marketing
Amazon Web Services

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Metrics

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Running ad platforms at ultra-low latency and high throughput in the cloud

Victor Gershkovich

Data Platform Team Lead, Real-Time Infrastructure
AppsFlyer

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Victor Gershkovich

- Data Platform Team Lead at AppsFlyer
- 10+ years in AWS infrastructure
- Loves working at scale and designing robust systems
- Strives for a good night’s sleep 24/7



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

1. AppsFlyer in a nutshell
2. From Amazon EC2 to Amazon EKS
3. Embracing AWS Graviton – Storage-optimized instances
4. Empowering our developers



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AppsFlyer



$294M
in funding

1,500+
employees

19
offices globally

65%
global MMP market share*

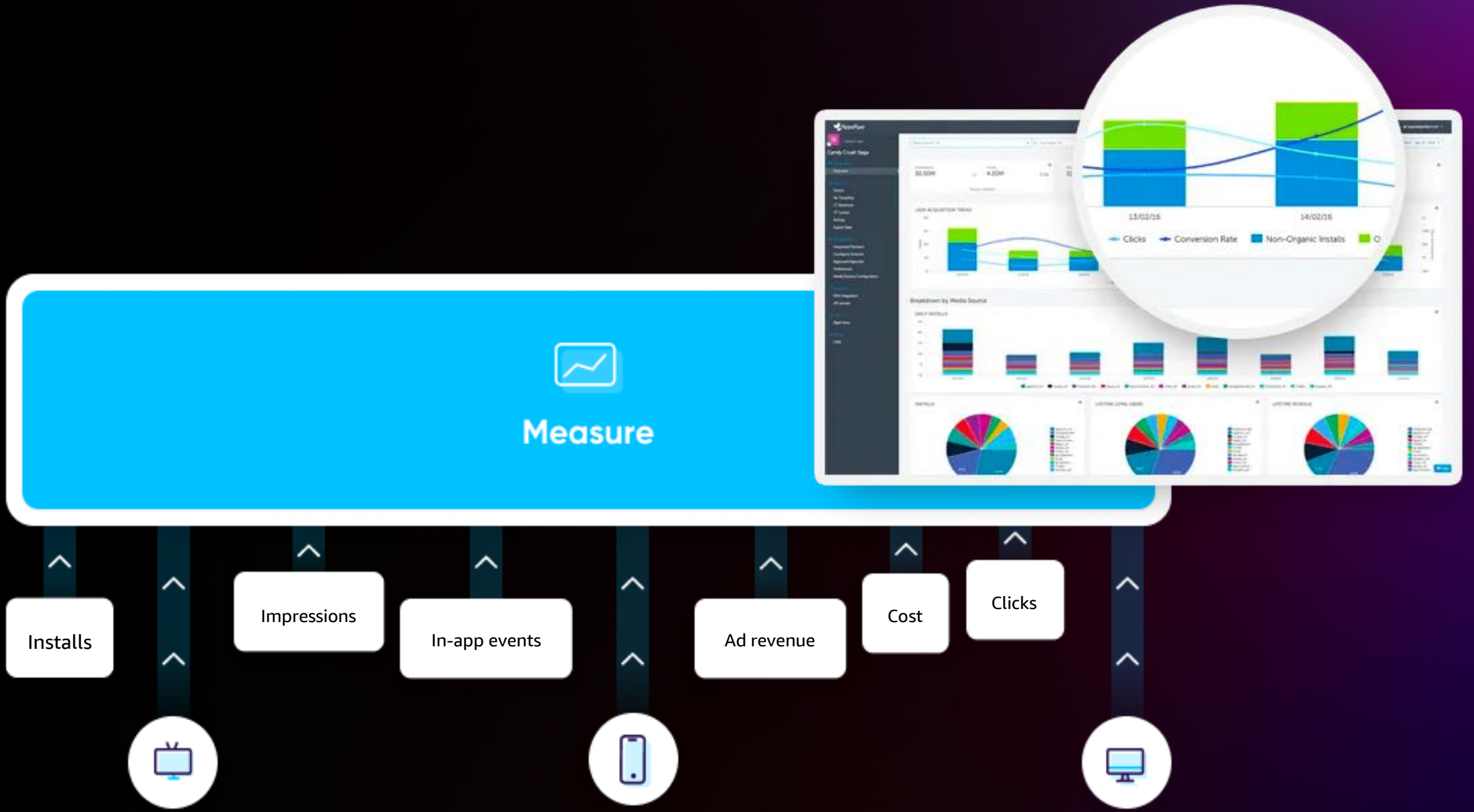
magma · 8VC EIGHT ROADS · T CAPITAL PARTNERS · Goldman Sachs · pitango · QUMRA CAPITAL · GENERAL ATLANTIC · salesforce ventures

*Source: App Annie, Mobile Measurement Provider (MMP) SDKs implemented globally on iOS & Android (May 2020)

aws

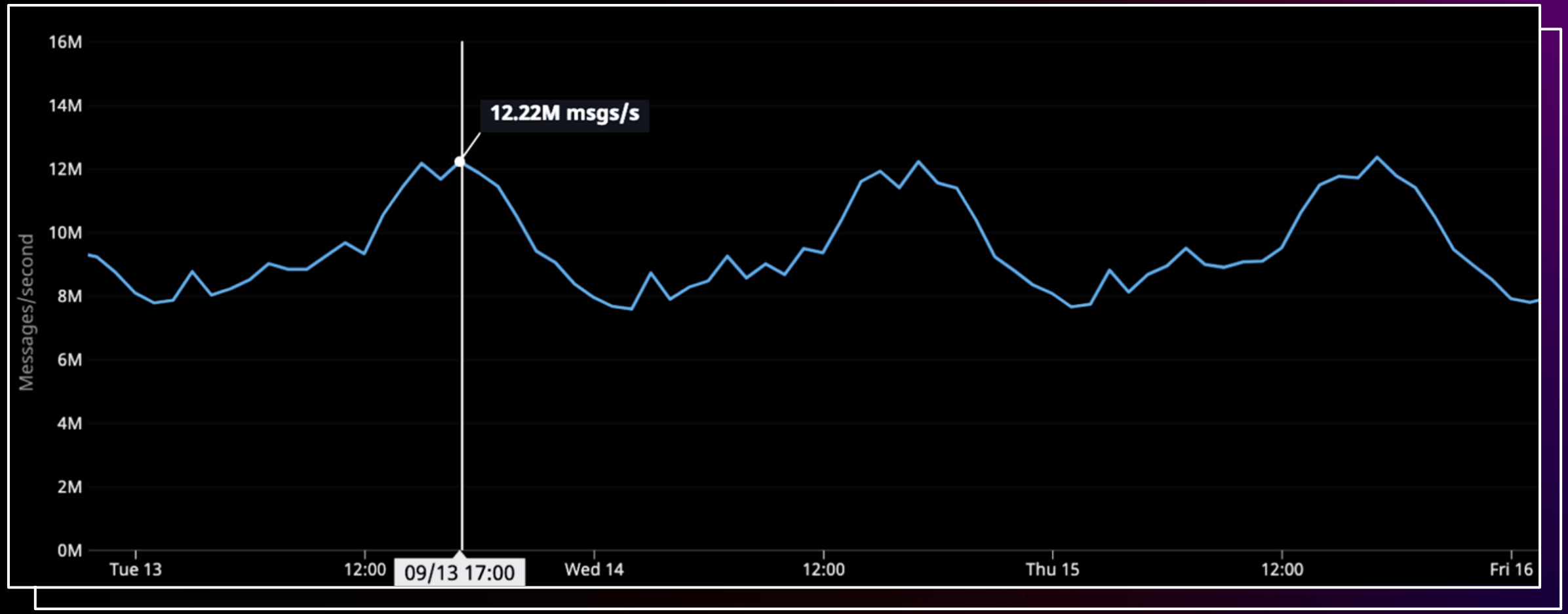
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AppsFlyer



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Real-time infrastructure



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Real-time infrastructure

+400
AppsFlyer engineers

+1,000
Microservices

~250K
Cloud resources and
dozens of SaaS integrations

60+ clusters
Clusters

1,000+ brokers
Instances/
pods

~800 B/d
Messages
per day

~800 TB
Disk
usage

~10 M/s
Messages
per second

~65 d/d
~65 deployments a day

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

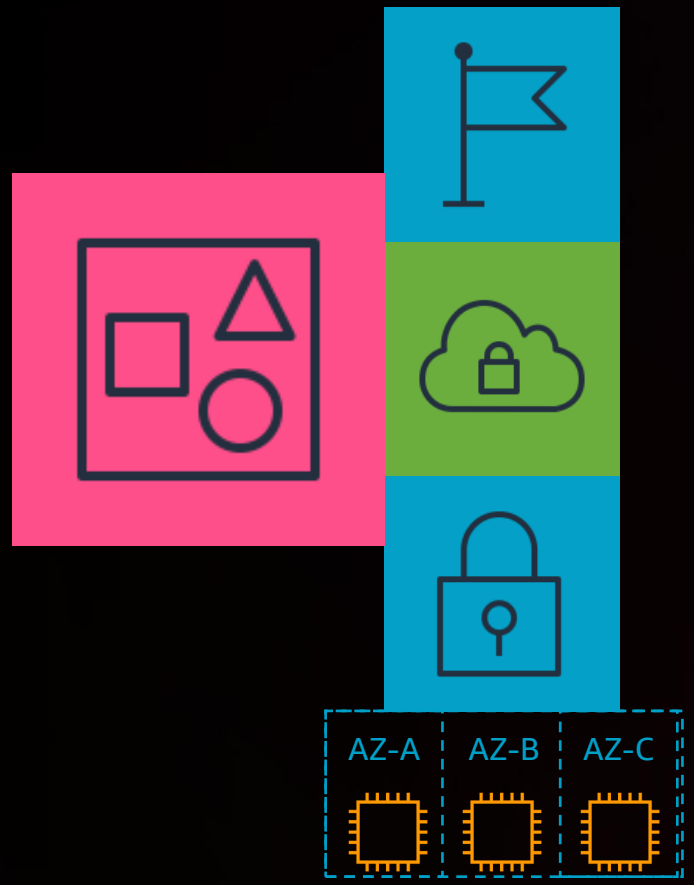
Agenda

- AppsFlyer in a nutshell
- 2 From Amazon EC2 to Amazon EKS
- 3 Embracing AWS Graviton – Storage-optimized instances
- 4 Empowering our developers



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

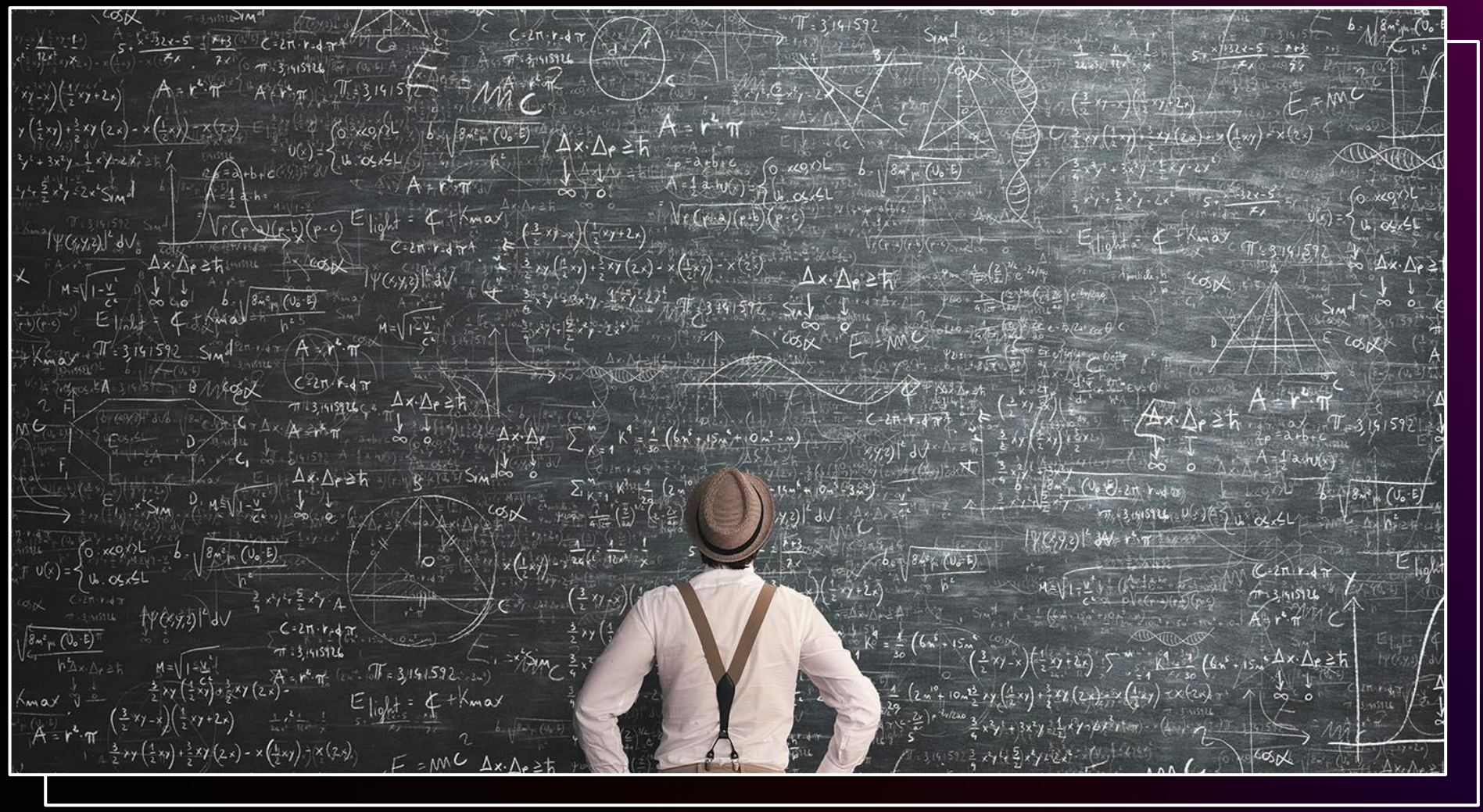
A classic Amazon EC2 stateful database



- Infrastructure as code
- Configuration management
- DNS/Service discovery
- CI/CD pipelines
- Secret management
- Observability stack
- Tooling

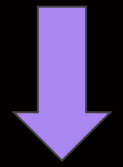
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Complexity



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why Amazon EKS?





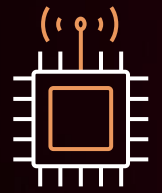
Tooling





Velocity





Instances\AMI





Efficiency

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AppsFlyer

Force multiplier



IaC/GitOps

Self-serve

Community

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

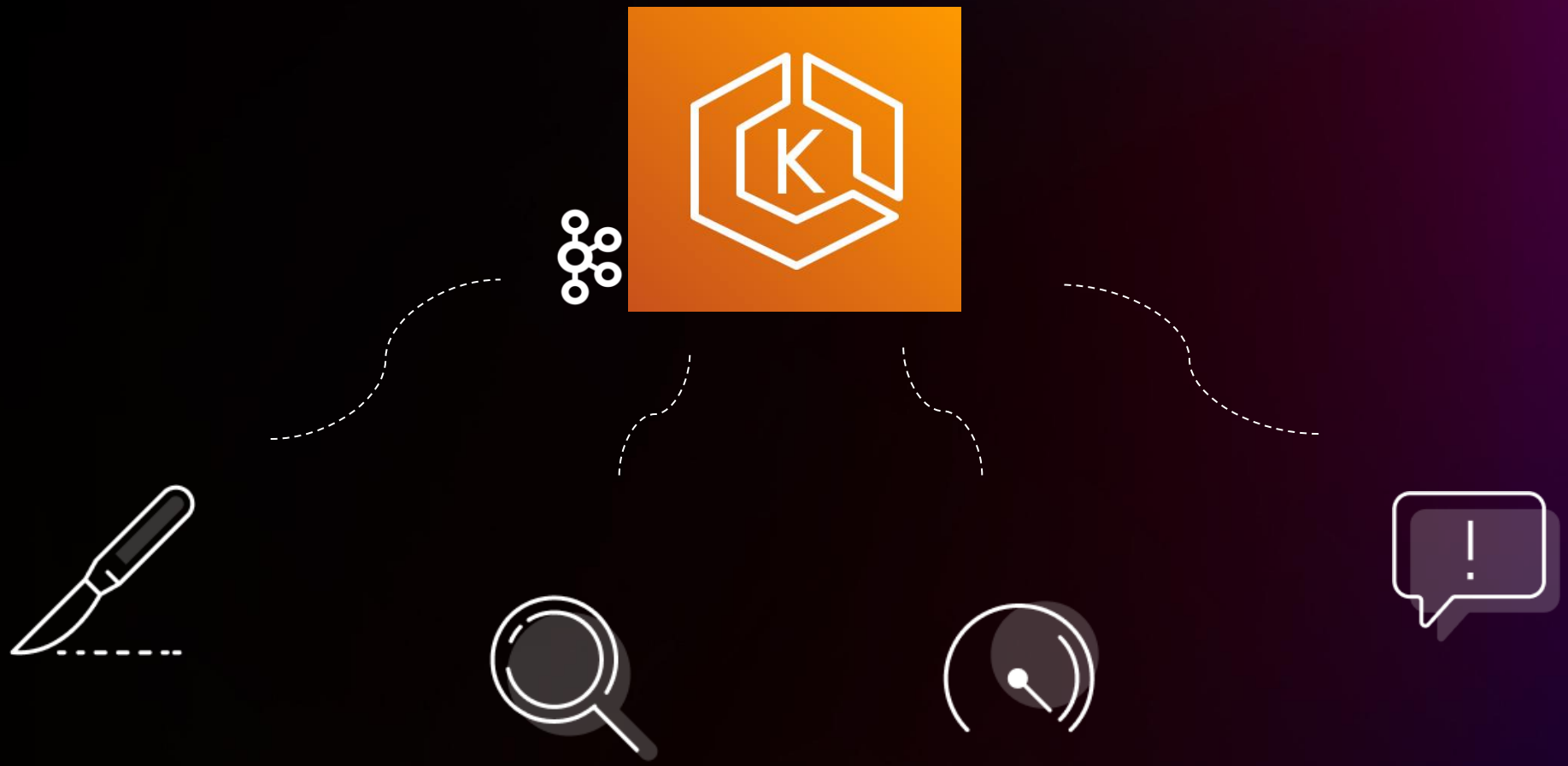
AppsFlyer

What, why, how



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS architecture



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS architecture

Dev Stg Prod

AWS



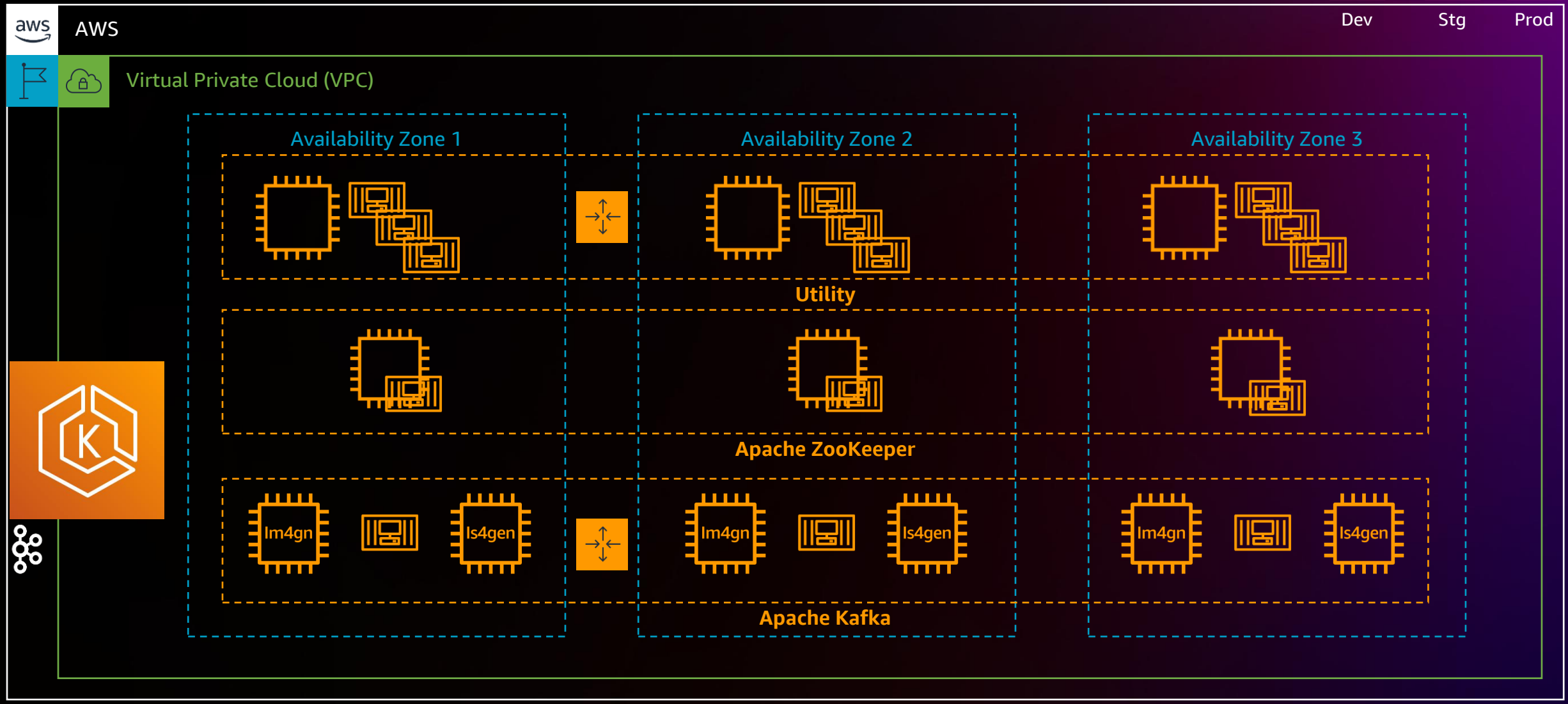
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS architecture



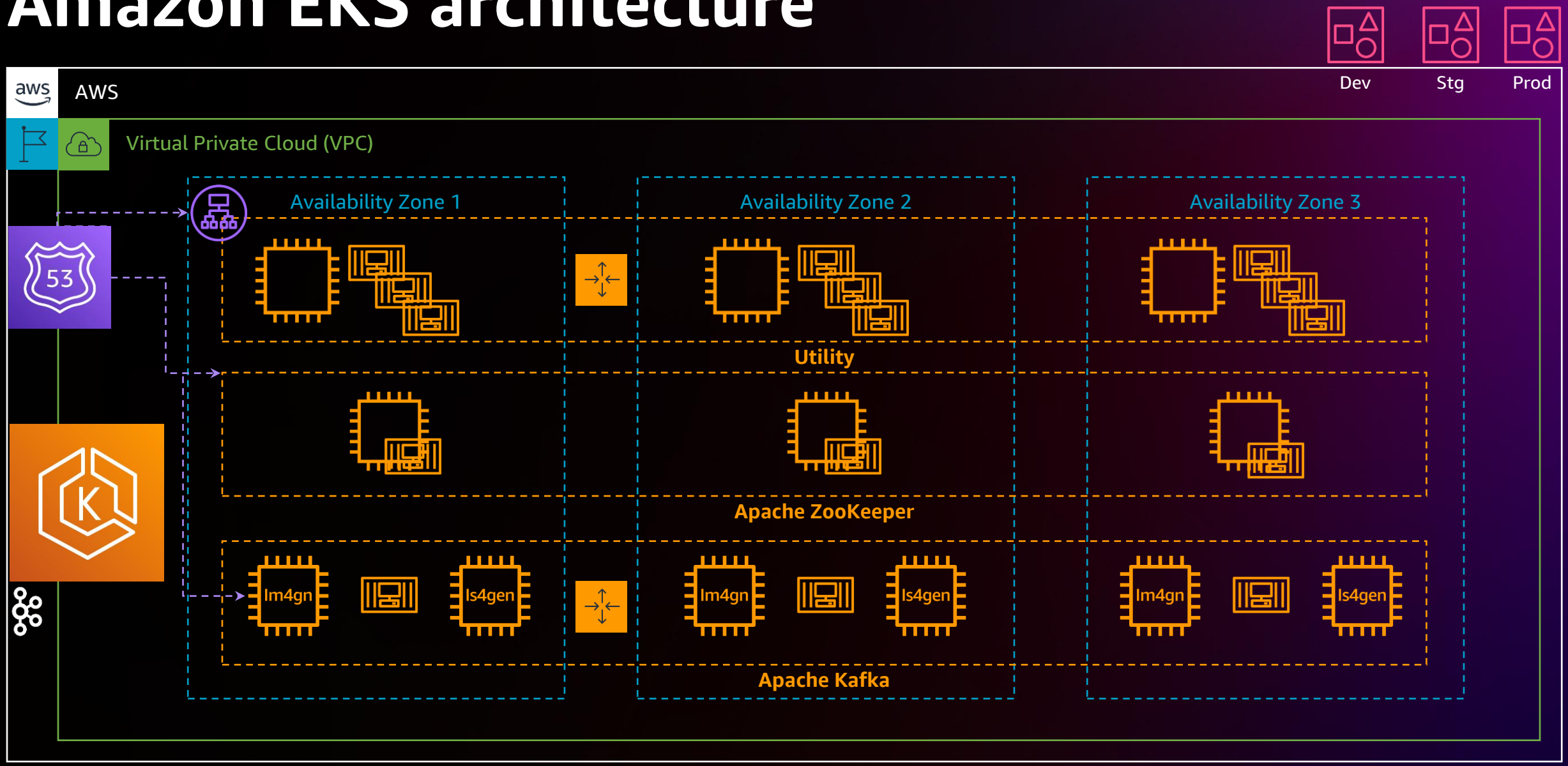
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS architecture



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS architecture

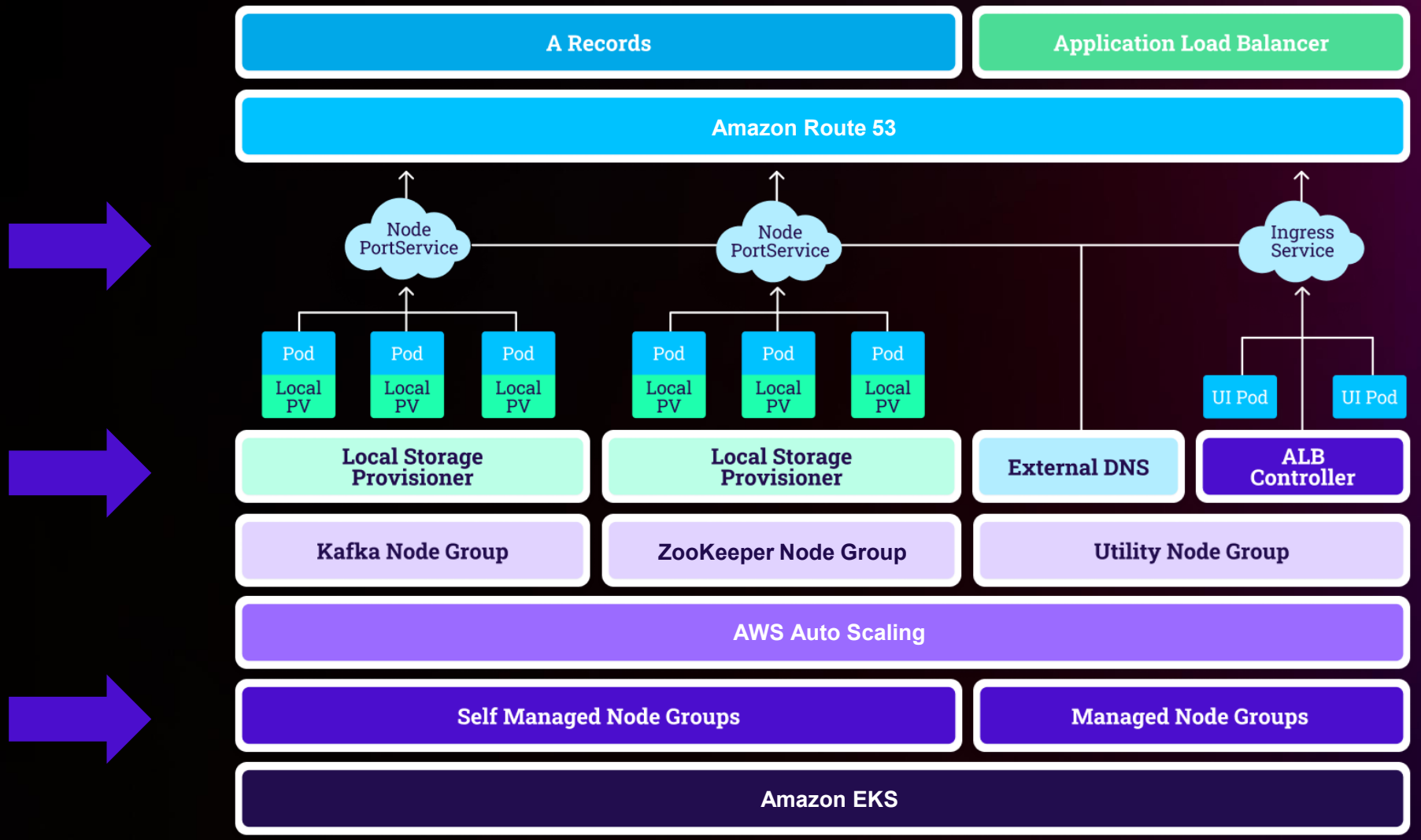


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS internal



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

1

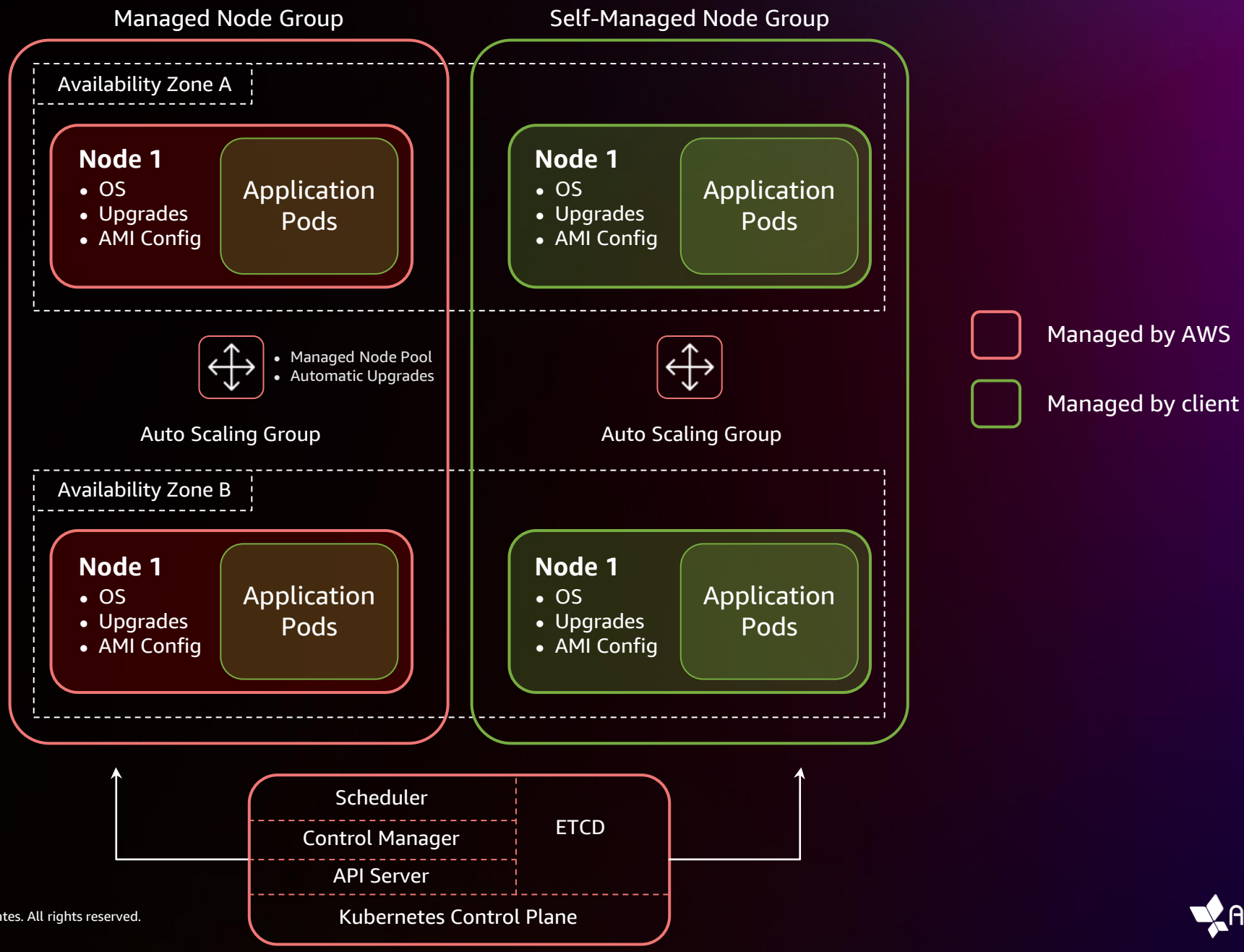
Managed

vs.

self-managed

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS



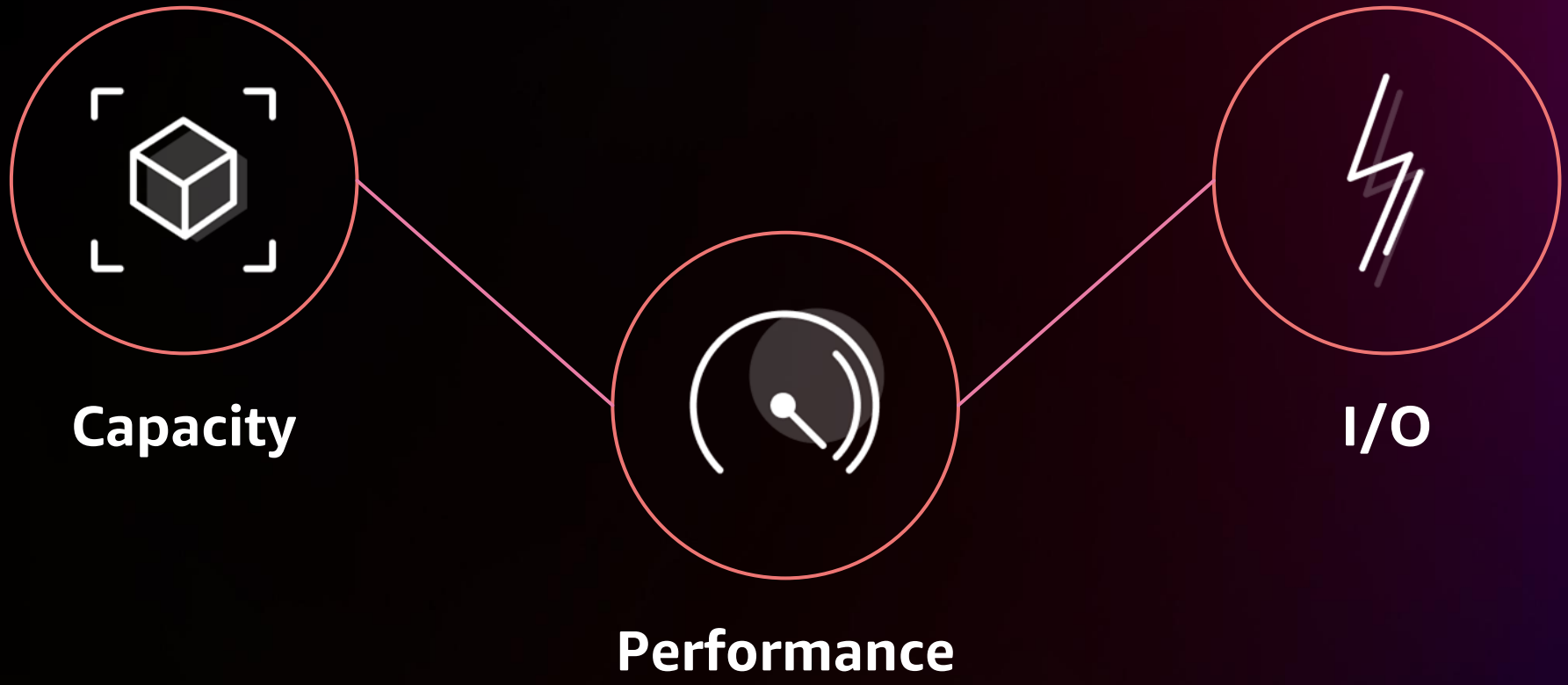
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2

Local instance storage

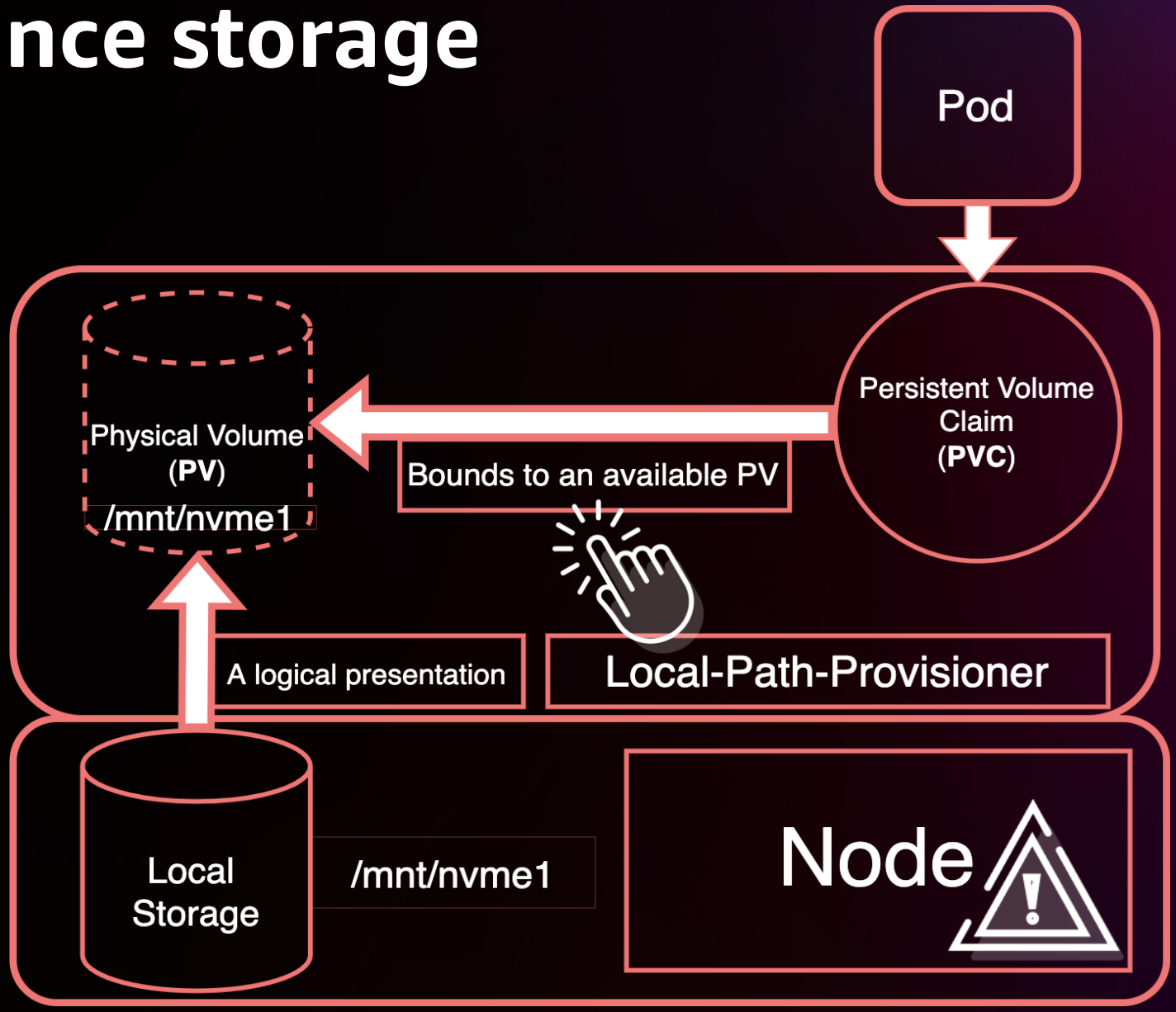
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Local instance storage



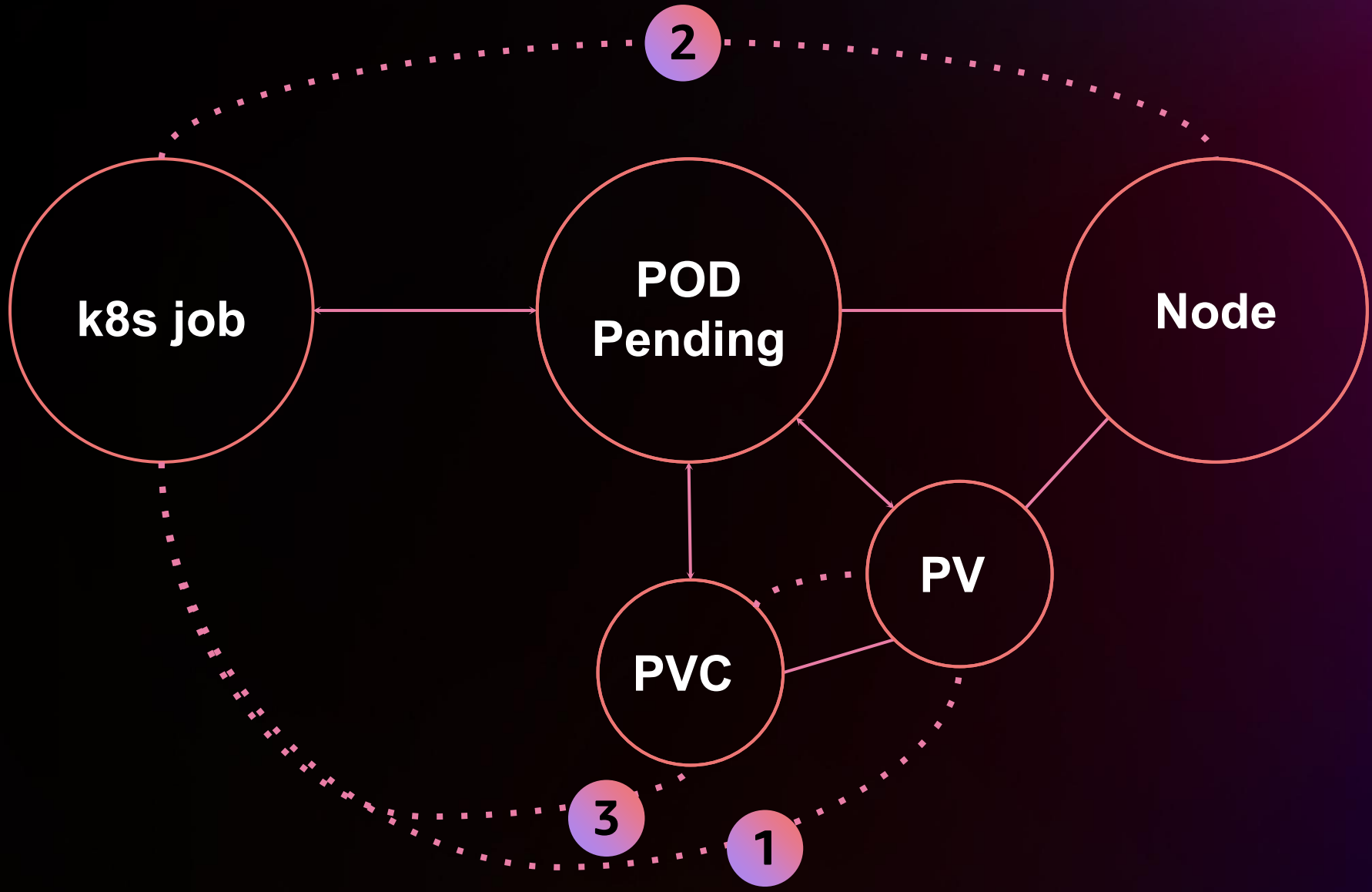
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Local instance storage



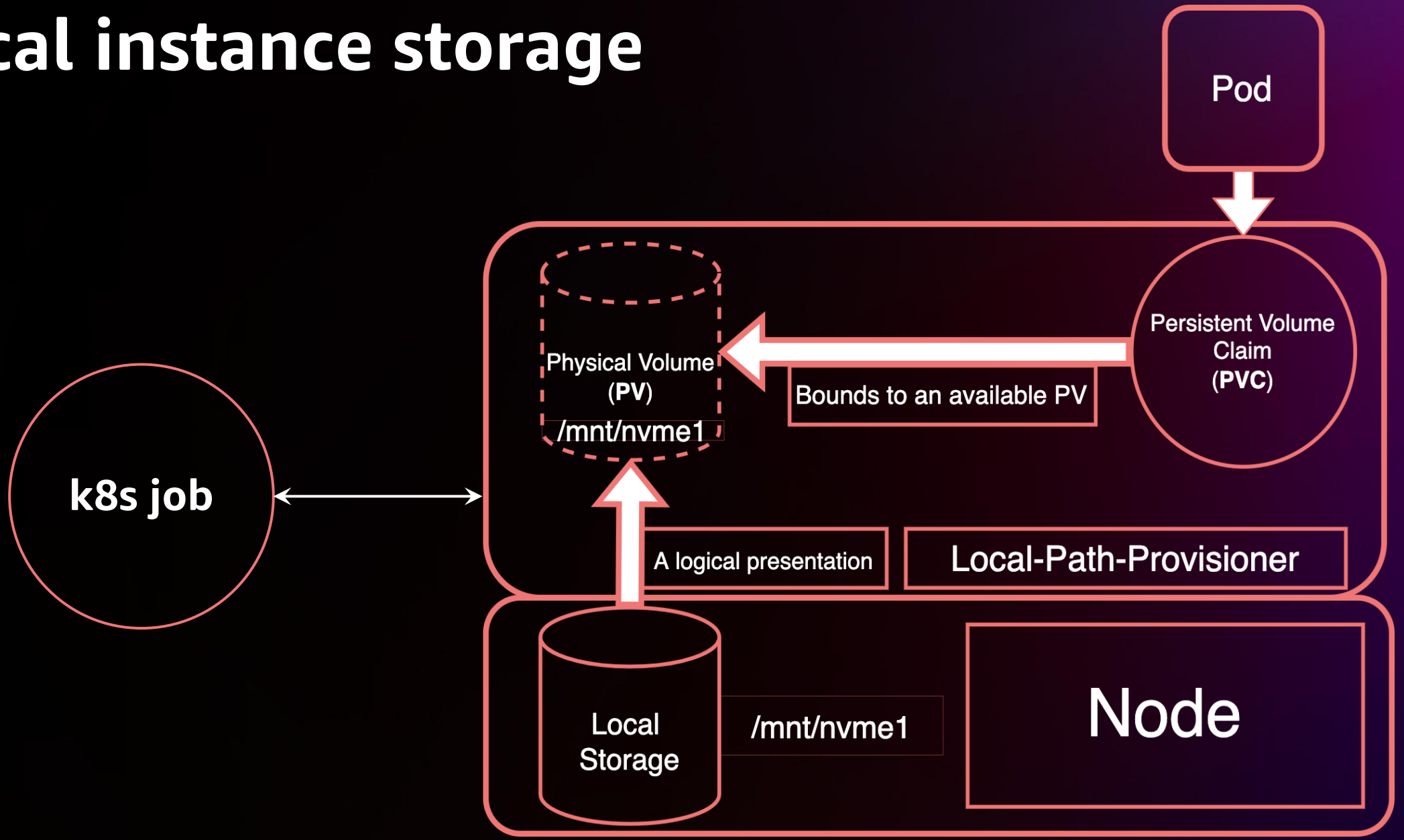
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Local instance storage – Job



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Local instance storage



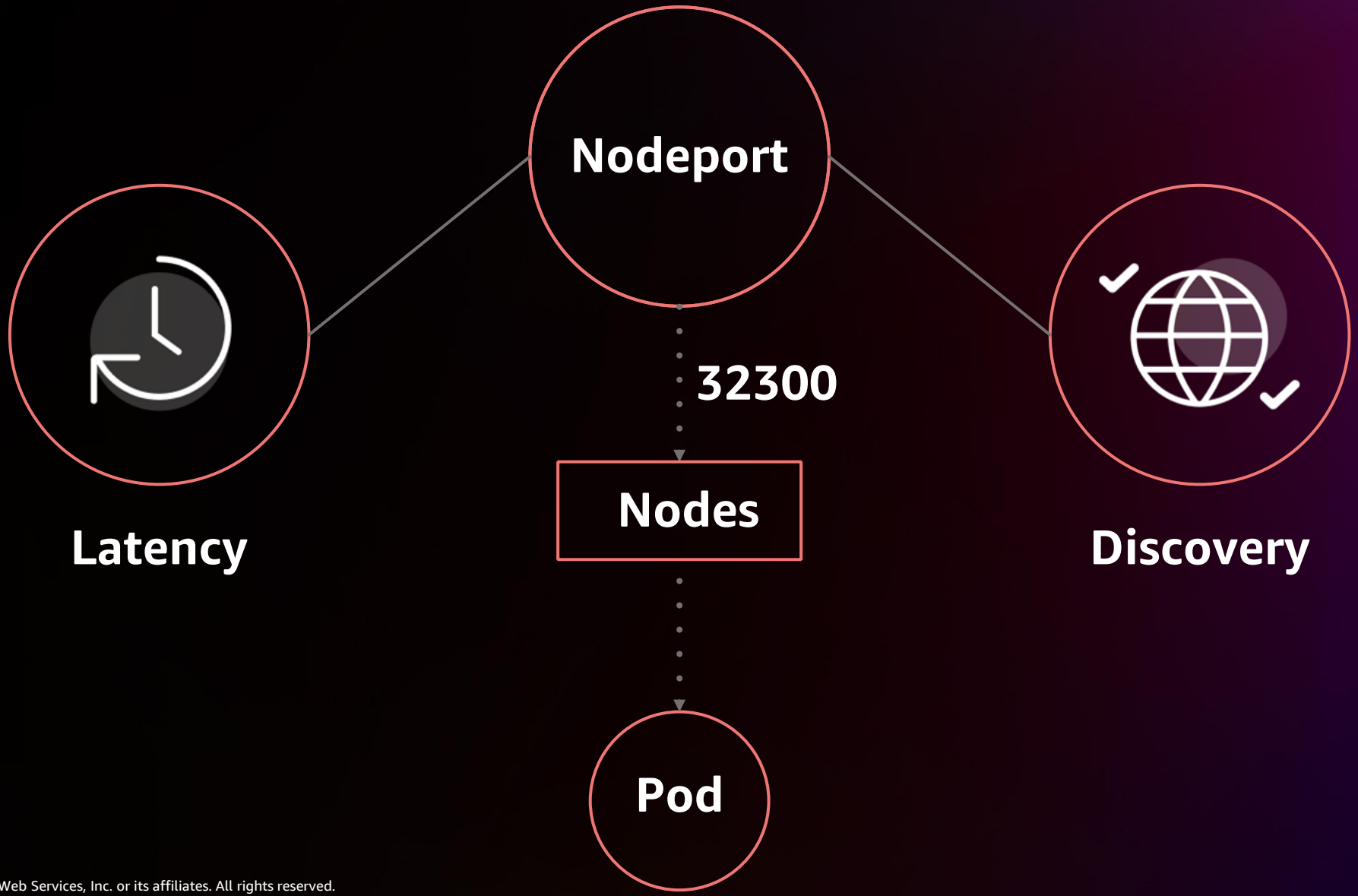
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3

Networking

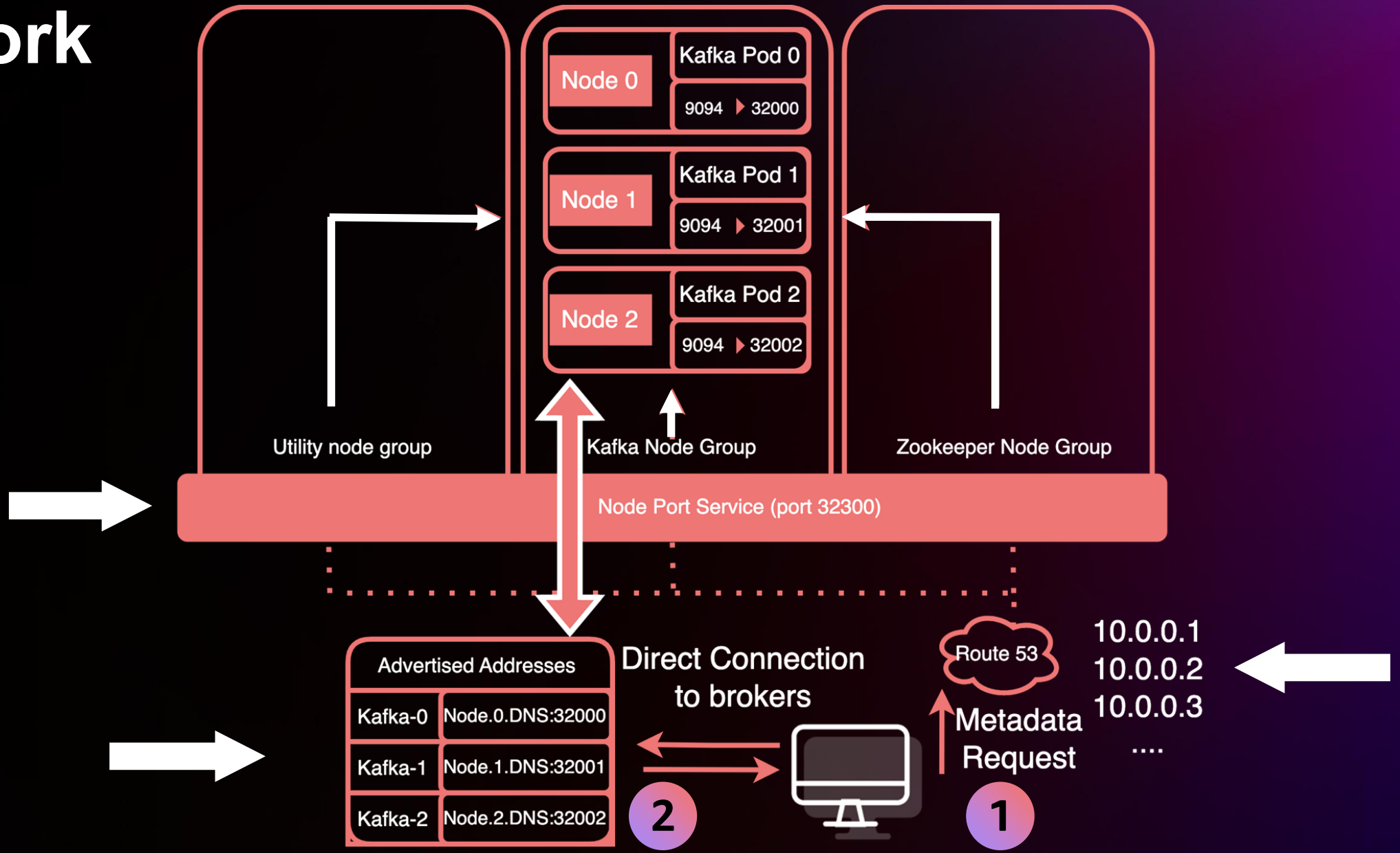
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Network



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.
AppsFlyer

Network



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

- AppsFlyer in a nutshell
- From Amazon EC2 to Amazon EKS
3. Embracing AWS Graviton – Storage-optimized instances
4. Empowering our developers



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

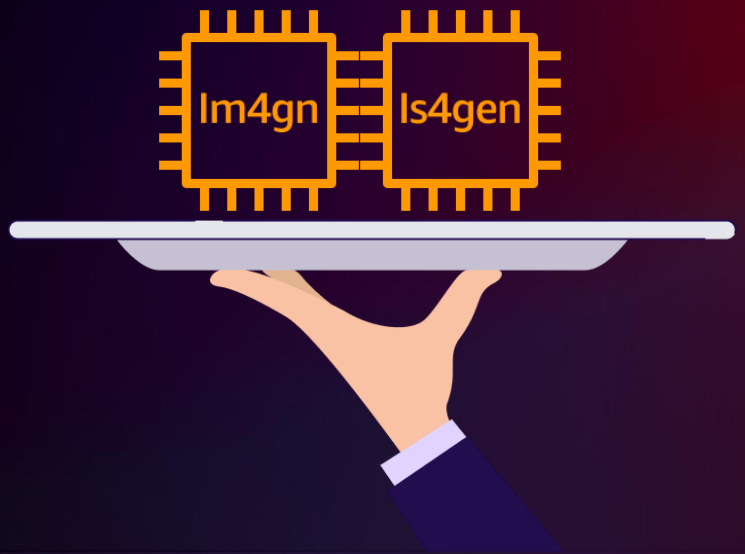
AWS Graviton2 storage-optimized instances

AWS News Blog

New Storage-Optimized Amazon EC2 Instances (Im4gn and Is4gen) Powered by AWS Graviton2 Processors

by Jeff Barr | on 30 NOV 2021 | in Amazon EC2, Announcements, AWS Re:Invent, Graviton, Launch, News | Permalink |

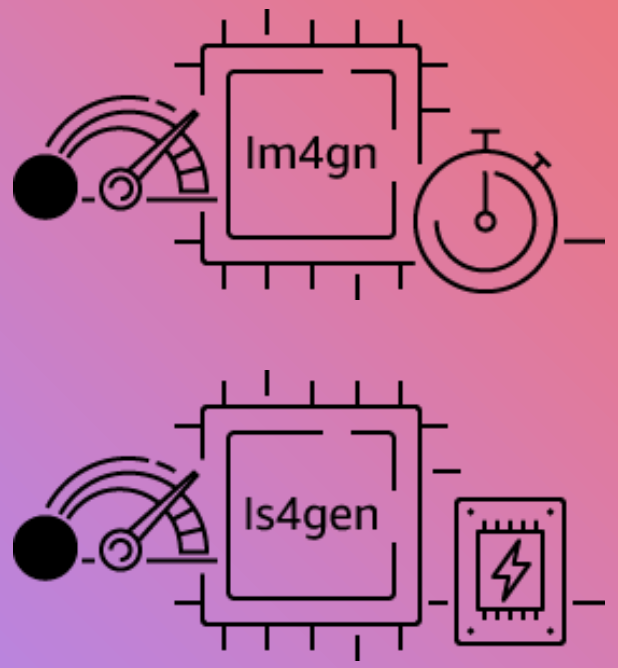
Share



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Graviton2 storage-optimized instances

- Maximized resource efficiency
- Flexibility and choice
- High-performance storage and networking



High-performance, storage-intensive workloads

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.
AppsFlyer

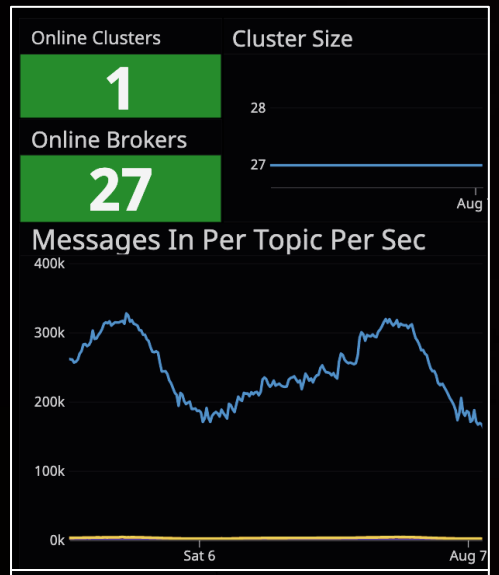
Maximized resource efficiency

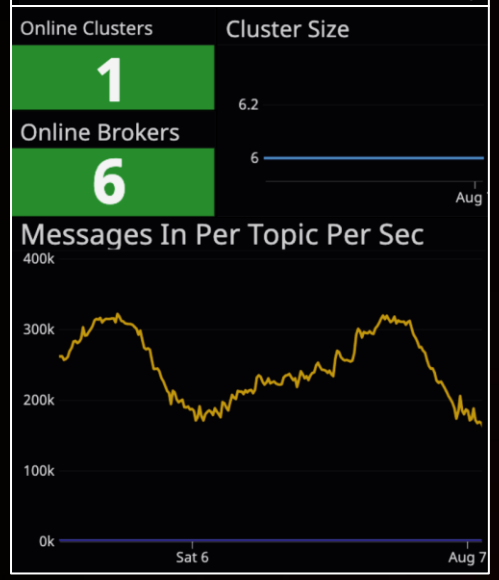
350K msg/s
1.8 KB

Traffic

27 → 6
I3.2xl
is4gen.2xl

Instance





0.563 ms

Produce latency

-93%

0.022 ms

Produce latency

52.64 ms

Consume time average

-31%

36.52 ms

Consume time average

63%

CPU

-16%

50%

CPU

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Maximized resource efficiency



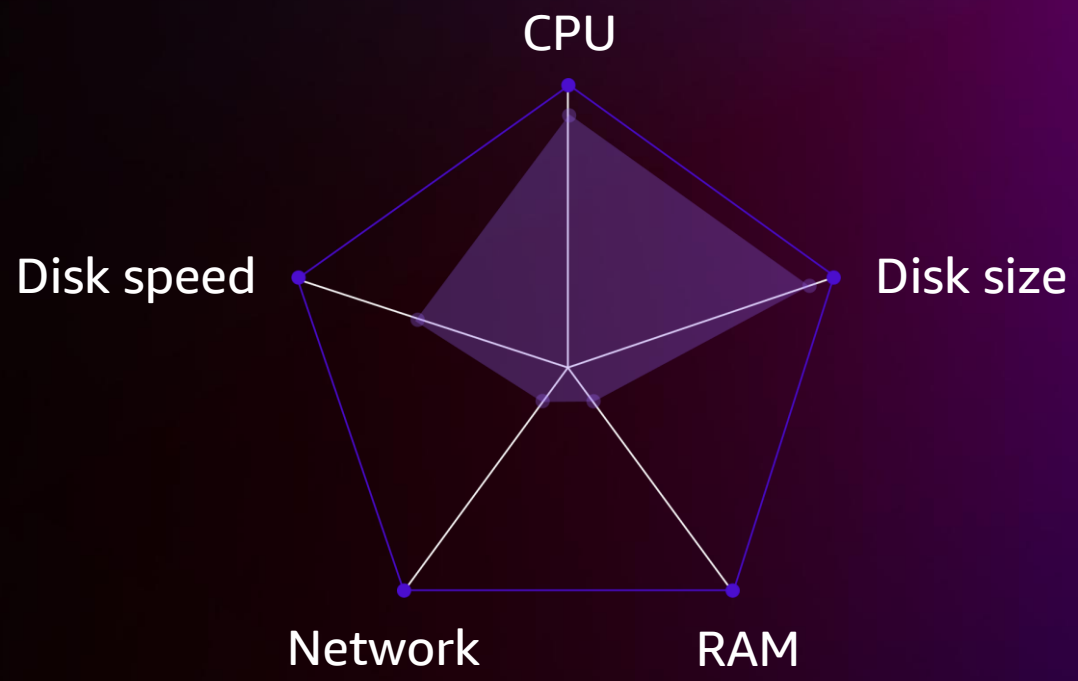
-60%

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Flexibility and choice

Type	Instance	Cost reduction
Im4gn.4xlarge	9	47%
Im4gn.xlarge	9	60%
is4gen.2xlarge	15	66%
is4gen.4xlarge	9	67%
im4gn.2xlarge	9	70%
is4gen.2xlarge	12	72%

65%

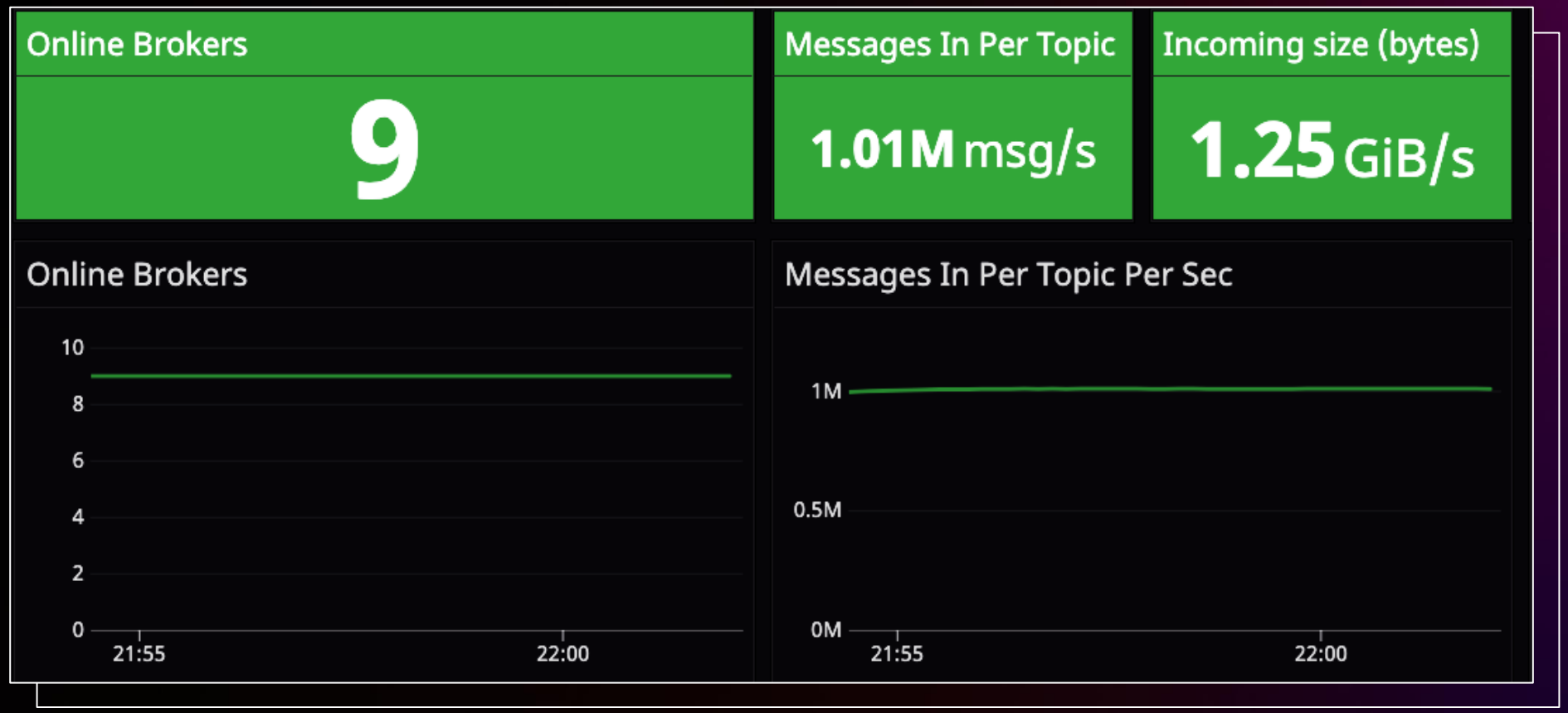


Instance name	vCPUs	Memory	Local NVMe storage (AWS Nitro SSD)	Network bandwidth	Price
im4gn.2xlarge	8	32 GiB	3.75 TB	Up to 25 Gbps	40%
is4gen.2xlarge	8	48 GiB	7.5 TB	Up to 25 Gbps	

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

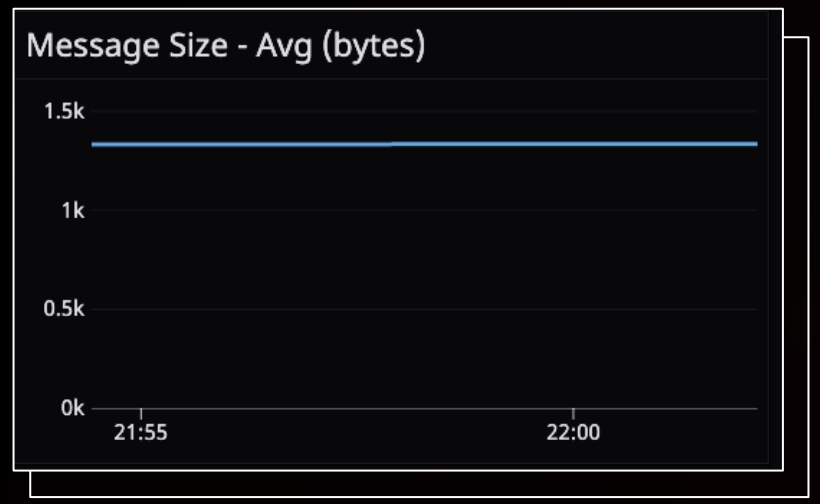
High performance

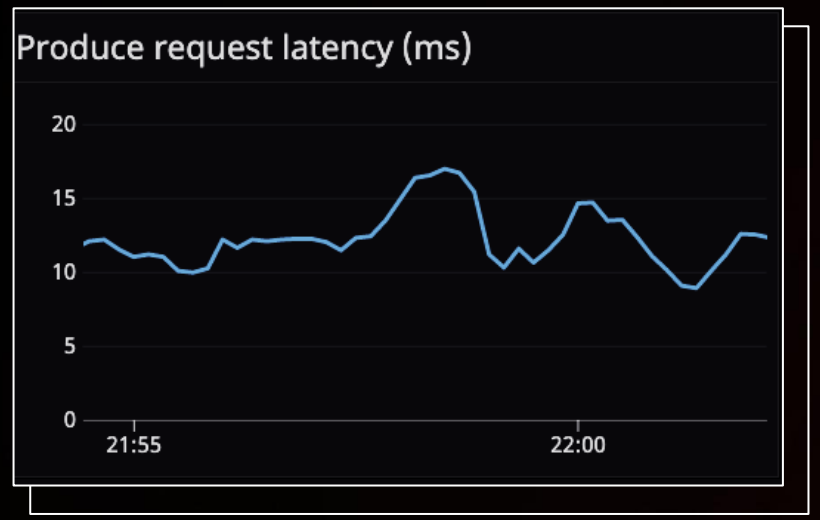
Im4gn.2xlarge

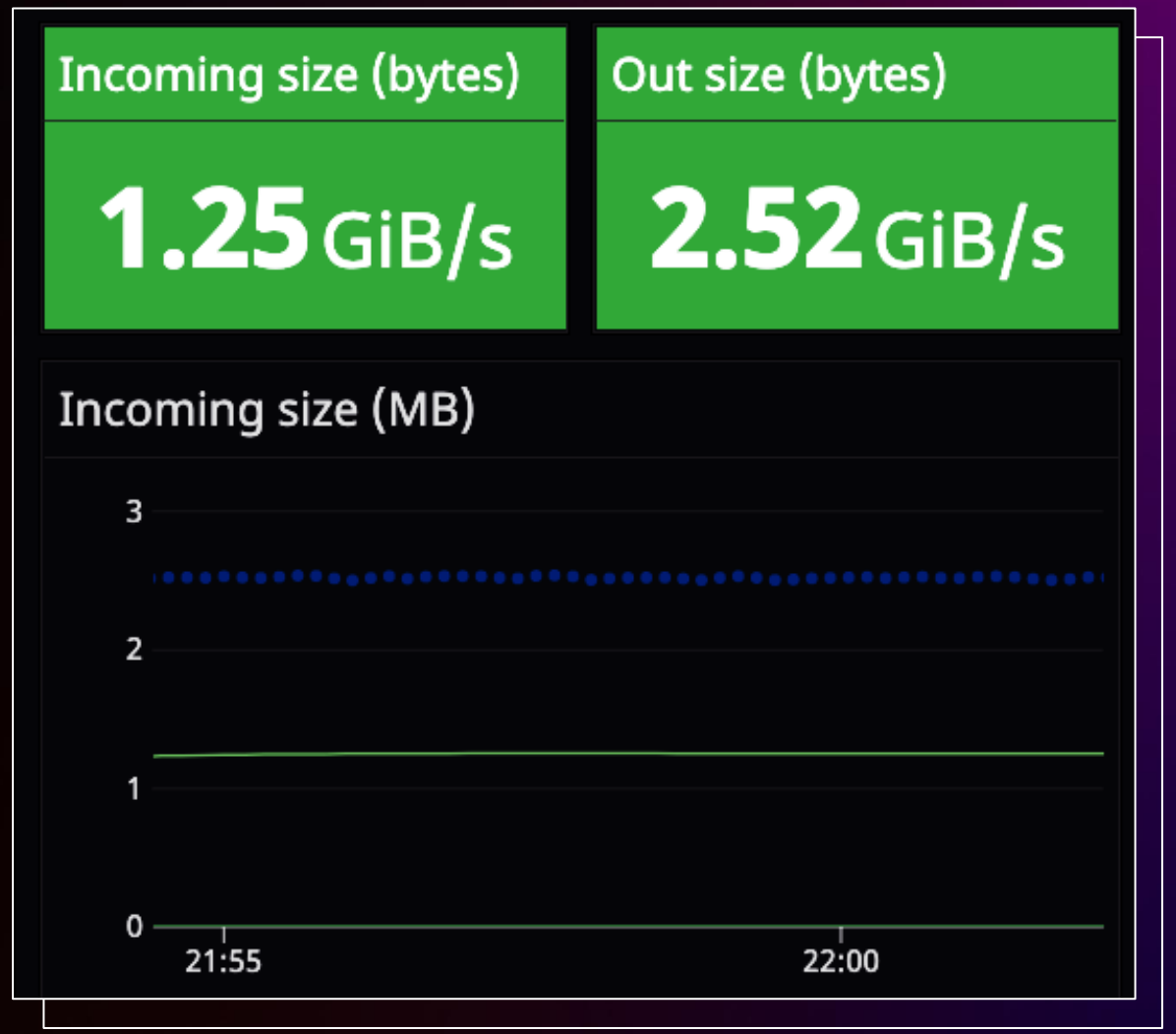


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

High performance

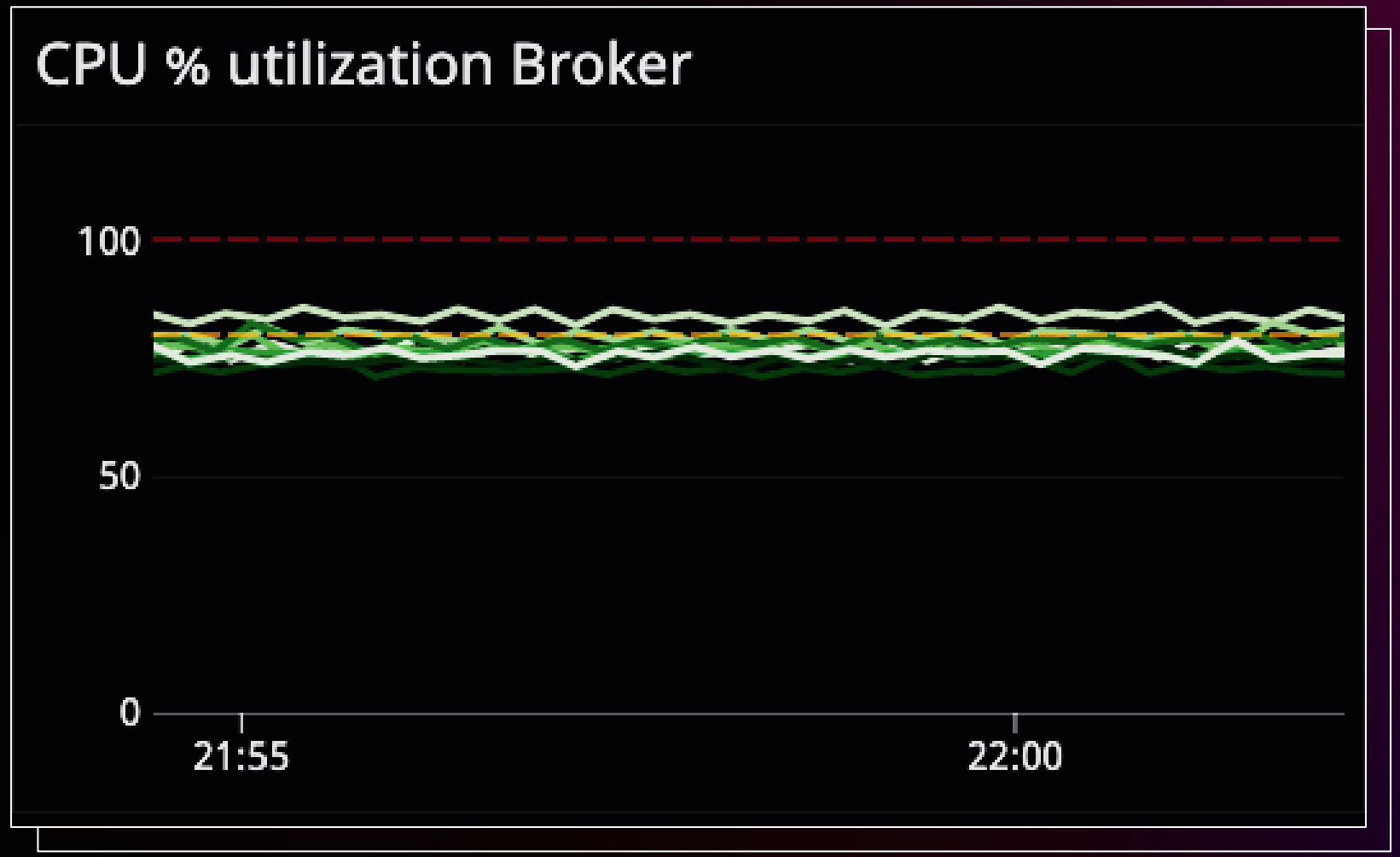






© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

High performance



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

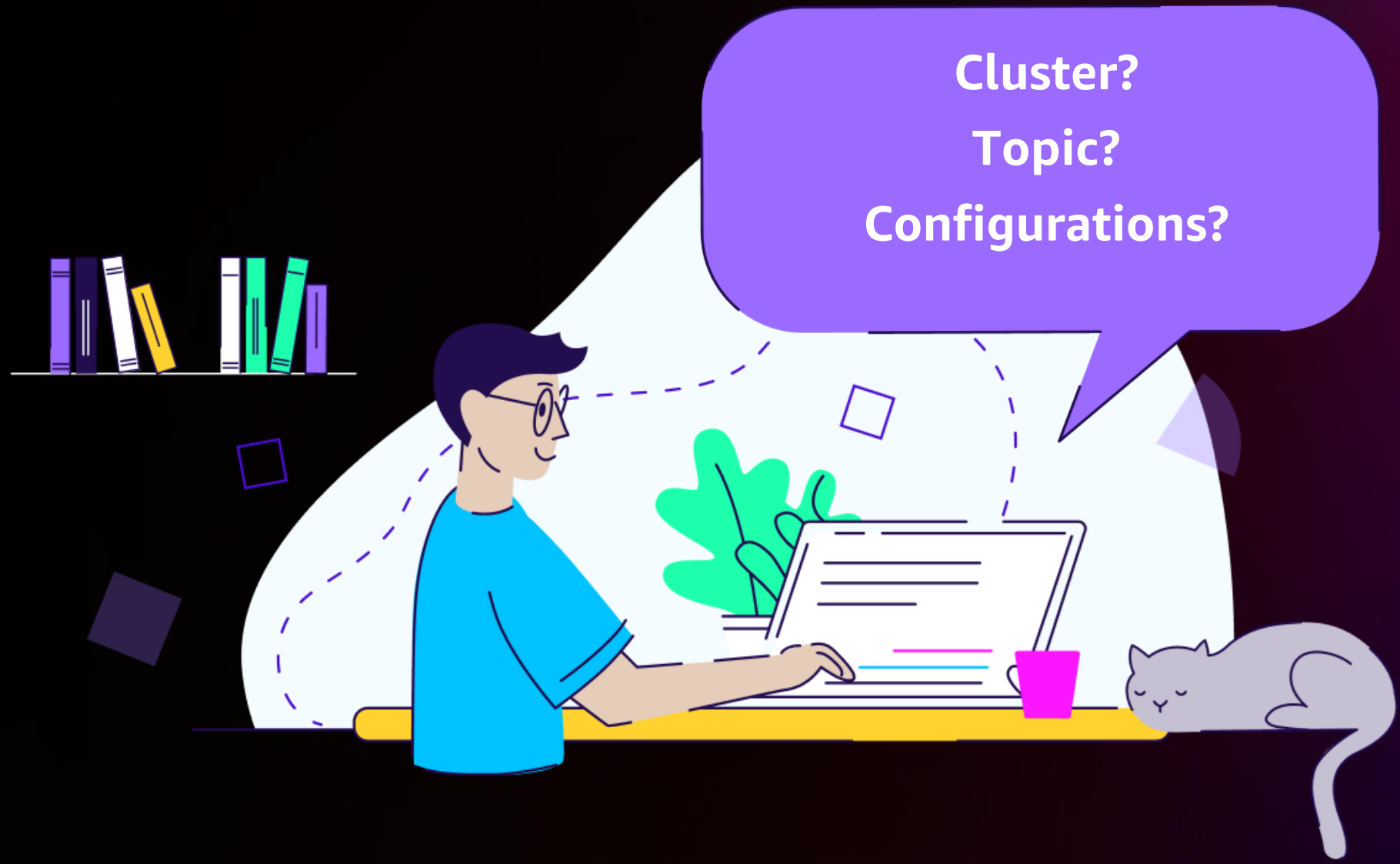
Agenda

- AppsFlyer in a nutshell
- From Amazon EC2 to Amazon EKS
- Embracing AWS Graviton – Storage-optimized instances
- 4 Empowering our developers



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Self-serve flow



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Self-serve flow

Terra Crust

AppsFlyer



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AppsFlyer

Self-serve configuration

```
module "strimzi-kafka" {
  source             = "git::https://git.af.internal/real-time-platform/af-rti-iac/modules/strimzi.git//terraform?ref=0.1.3"
  cluster_name       = "river-relay"
  aws_profile        = "Production"
  repo               = "https://git.af.internal/mos/af_river_relay_analytics"
  role               = "mos"
  system             = "river-relay-analytics"
  create_mirror_maker = false
  mirror_maker = {
    source_cluster_bootstrap = "kafka-mos-lg-1.prod.internal:9092,kafka-mos-lg-2.prod.internal:9092,kafka-mos-lg-3.prod.internal:9092"
    sourceconnector_config = {
      "sync.topic.configs.enabled" : "false"
    }
  }
  kafka = {
      brokers = 3
  }
  topic_list = [
  {
    name = "lowp"
    partitions = 360
    replicas = 3
    configuration = {
        "delete.retention.ms": "10800000"
        "retention.ms": "10800000"
        "cleanup.policy": "delete"
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Self-serve configuration

```
module "strimzi-kafka" {
  source             = "git::https://git.af.internal/real-time-platform/af-rti-iac/modules/strimzi.git//terraform?ref=0.1.3"
  cluster_name       = "river-relay"
  aws_profile        = "Production"
  repo               = "https://git.af.internal/mos/af_river_relay_analytics"
  role               = "mos"
  system             = "river-relay-analytics"
  create_mirror_maker = false
  mirror_maker = {
    source_cluster_bootstrap = "kafka-mos-lg-1.prod.internal:9092,kafka-mos-lg-2.prod.internal:9092,kafka-mos-lg-3.prod.internal:9092"
    sourceconnector_config = {
      "sync.topic.configs.enabled" : "false"
    }
  }
  kafka = {
      brokers = 3
  }
  topic_list = [
  {
    name = "lowp"
    partitions = 360
    replicas = 3
    configuration = {
        "delete.retention.ms": "10800000"
        "retention.ms": "10800000"
        "cleanup.policy": "delete"
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

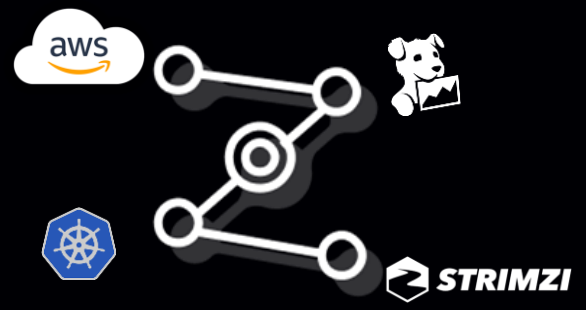
Self-serve configuration

```
module "strimzi-kafka" {
  source             = "git::https://git.af.internal/real-time-platform/af-rti-iac/modules/strimzi.git//terraform?ref=0.1.3"
  cluster_name       = "river-relay"
  aws_profile        = "Production"
  repo               = "https://git.af.internal/mos/af_river_relay_analytics"
  role               = "mos"
  system             = "river-relay-analytics"
  create_mirror_maker = false
  mirror_maker = {
    source_cluster_bootstrap = "kafka-mos-lg-1.prod.internal:9092,kafka-mos-lg-2.prod.internal:9092,kafka-mos-lg-3.prod.internal:9092"
    sourceconnector_config = {
      "sync.topic.configs.enabled" : "false"
    }
  }
  kafka = {
      brokers = 3
  }
  topic_list = [
  {
    name = "lowp"
    partitions = 360
    replicas = 3
    configuration = {
        "delete.retention.ms": "10800000"
        "retention.ms": "10800000"
        "cleanup.policy": "delete"
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Self-serve configuration

```
module "strimzi-kafka" {
  source             = "git::https://git.af.internal/real-time-platform/af-rti-iac/modules/strimzi.git//terraform?ref=0.1.3"
  cluster_name       = "river-relay"
  aws_profile        = "Production"
  repo               = "https://git.af.internal/mos/af_river_relay_analytics"
  role               = "mos"
  system             = "river-relay-analytics"
  create_mirror_maker = false
  mirror_maker = {
    source_cluster_bootstrap = "kafka-mos-lg-1.prod.internal:9092,kafka-mos-lg-2.prod.internal:9092,kafka-mos-lg-3.prod.internal:9092"
    sourceconnector_config = {
      "sync.topic.configs.enabled" : "false"
    }
  }
  kafka = {
      brokers = 3
  }
  topic_list = [
  {
    name = "lowp"
    partitions = 360
    replicas = 3
    configuration = {
        "delete.retention.ms": "10800000"
        "retention.ms": "10800000"
        "cleanup.policy": "delete"
```



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.
AppsFlyer

Self-serve configuration

```
module "strimzi-kafka" {
  source             = "git::https://git.af.internal/real-time-platform/af-rti-iac/modules/strimzi.git//terraform?ref=0.1.3"
  cluster_name       = "river-relay"
  aws_profile        = "Production"
  repo               = "https://git.af.internal/mos/af_river_relay_analytics"
  role               = "mos"
  system             = "river-relay-analytics"
  create_mirror_maker = false
  mirror_maker = {
    source_cluster_bootstrap = "kafka-mos-lg-1.prod.internal:9092,kafka-mos-lg-2.prod.internal:9092,kafka-mos-lg-3.prod.internal:9092"
    sourceconnector_config = {
      "sync.topic.configs.enabled" : "false"
    }
  }
  kafka = {
      brokers = 3
  }
  topic_list = [
  {
    name = "lowp"
    partitions = 360
    replicas = 3
    configuration = {
        "delete.retention.ms": "10800000"
        "retention.ms": "10800000"
        "cleanup.policy": "delete"
```



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Self-serve configuration

```
module "strimzi-kafka" {
  source             = "git::https://git.af.internal/real-time-platform/af-rti-iac/modules/strimzi.git//terraform?ref=0.1.3"
  cluster_name       = "river-relay"
  aws_profile        = "Production"
  repo               = "https://git.af.internal/mos/af_river_relay_analytics"
  role               = "mos"
  system             = "river-relay-analytics"
  create_mirror_maker = false
  mirror_maker = {
    source_cluster_bootstrap = "kafka-mos-lg-1.prod.internal:9092,kafka-mos-lg-2.prod.internal:9092,kafka-mos-lg-3.prod.internal:9092"
    sourceconnector_config = {
      "sync.topic.configs.enabled" : "false"
    }
  }
  kafka = {
      brokers = 3
  }
  topic_list = [
  {
    name = "lowp"
    partitions = 360
    replicas = 3
    configuration = {
        "delete.retention.ms": "10800000"
        "retention.ms": "10800000"
        "cleanup.policy": "delete"
```



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Self-serve flow



Developer

aws © 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AppsFlyer

Self-serve flow



Developer

Terraform

Real-Time Module v0.1.3

Terra Crust

AppsFlyer

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AppsFlyer

Self-serve flow



Developer

GitOps workflow

Terraform

OPA

Real-Time Module v0.1.3

Terra Crust

AppsFlyer

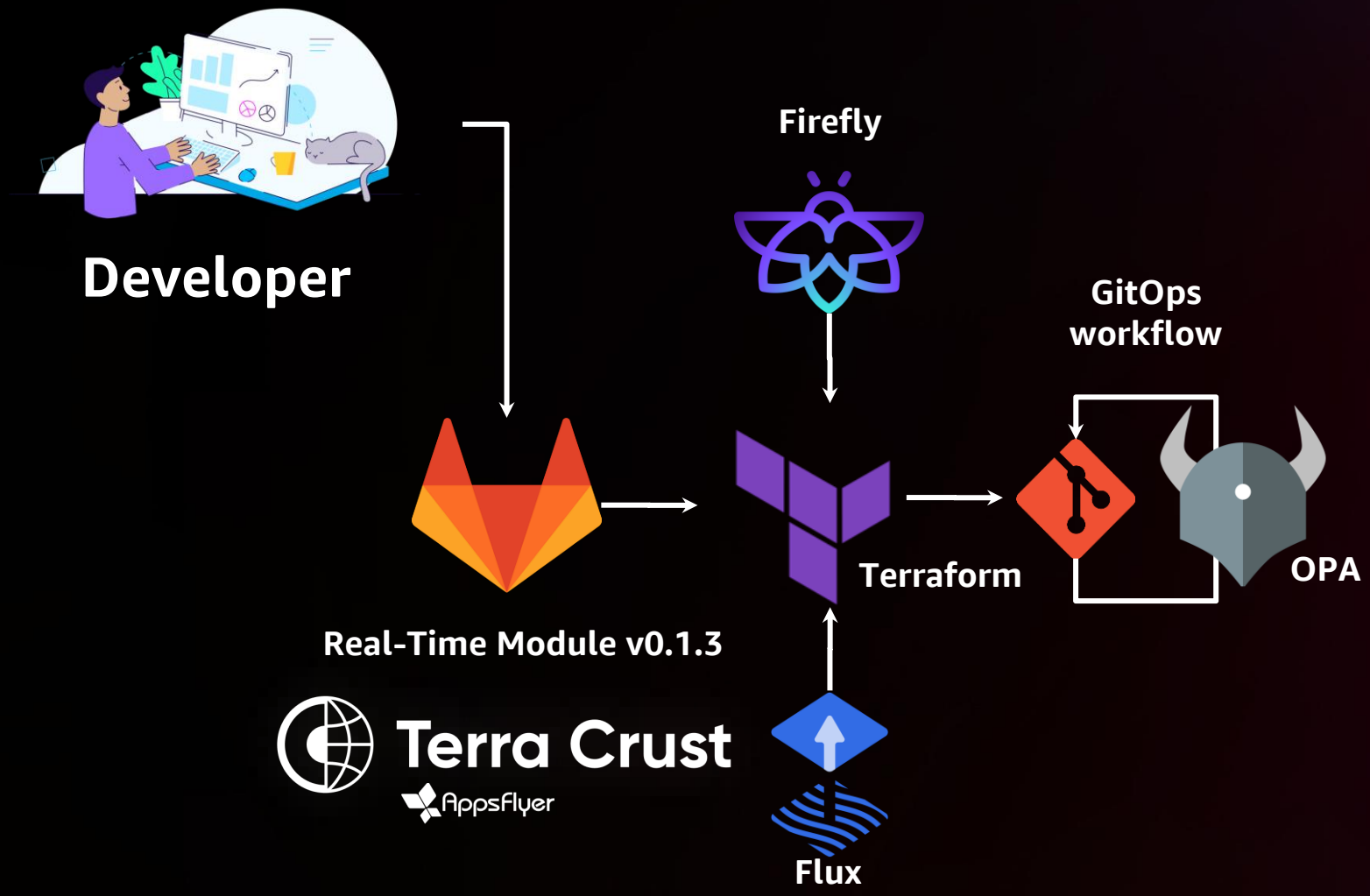
Flux

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AppsFlyer

Self-serve flow



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Self-serve flow



Developer

Real-Time Module v0.1.3

Terra Crust

AppsFlyer

Firefly

Terraform

Flux

GitOps workflow

OPA

Apply changes

STRIMZI

CRUISE CONTROL

aws

DATADOG

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AppsFlyer

Self-serve flow



Developer

Real-Time Module v0.1.3

Terra Crust

AppsFlyer

Firefly

Terraform

Flux

GitOps workflow

OPA

Apply changes

STRIMZI

CRUISE CONTROL

aws

DATADOG

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AppsFlyer

Agenda

- AppsFlyer in a nutshell
- From Amazon EC2 to Amazon EKS
- Embracing AWS Graviton – Storage-optimized instances
- Empowering our developers



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Value

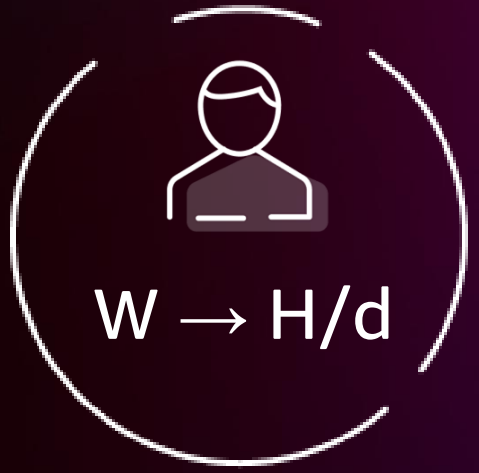


Latency





Cost



Operations

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Victor Gershkovich
Data Platform Team Lead, Real-Time Infrastructure
AppsFlyer
https://www.linkedin.com/in/victor-gershkovich
victor@appsflyer.com



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Running high-throughput, real-time ad platforms in the cloud

Gerry Louw

Head of Technology,
Advertising and Marketing
Amazon Web Services

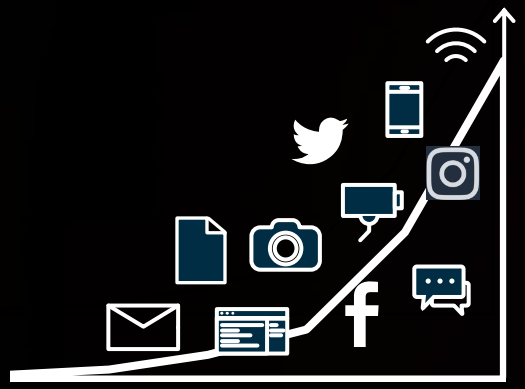
Akhil Aendapally

Sr. Solutions Architect,
Advertising and Marketing
Amazon Web Services

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Programmatic advertising challenges



Margin and
cost pressure



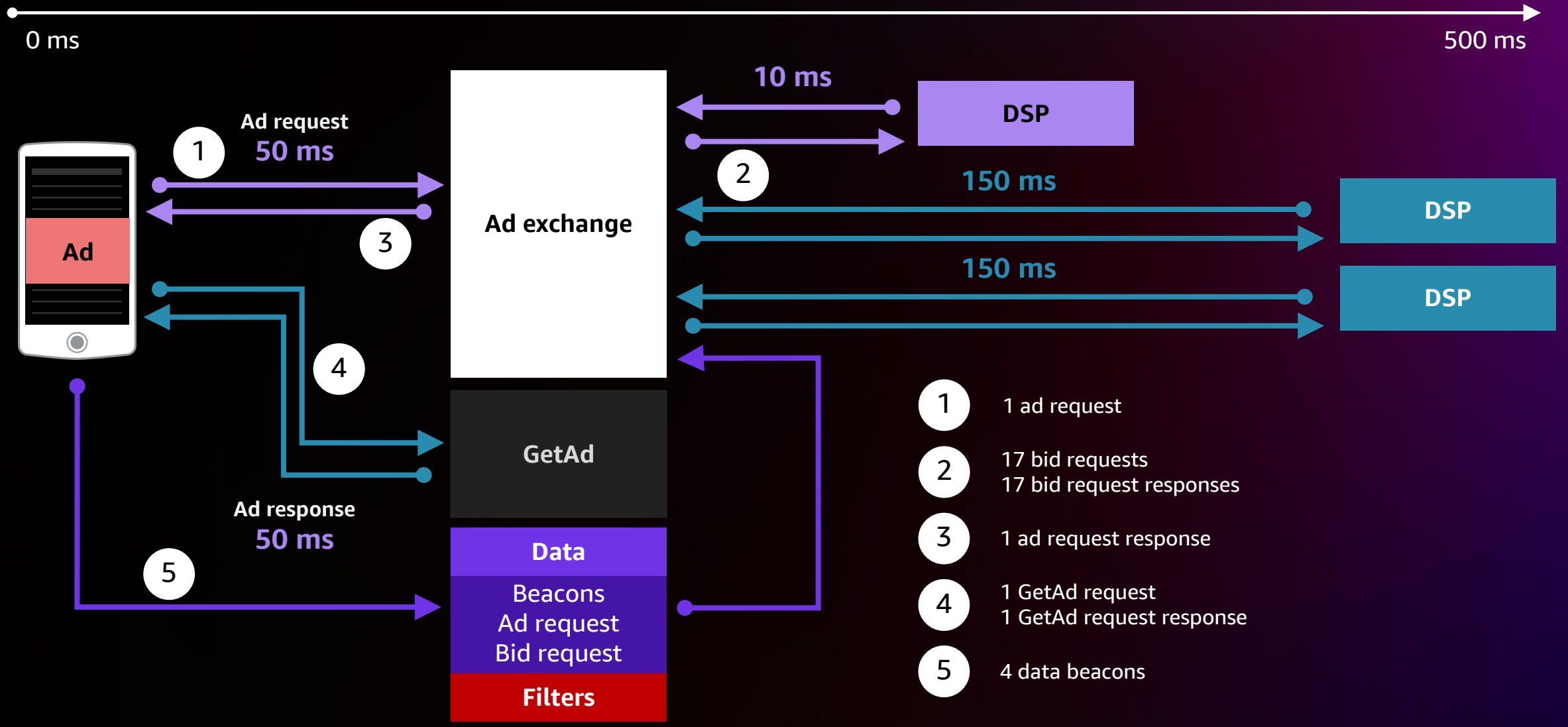
Millisecond latency
response times



Variable global
internet scale
impression volumes

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The blink of an eye



42 events recorded in <500 milliseconds

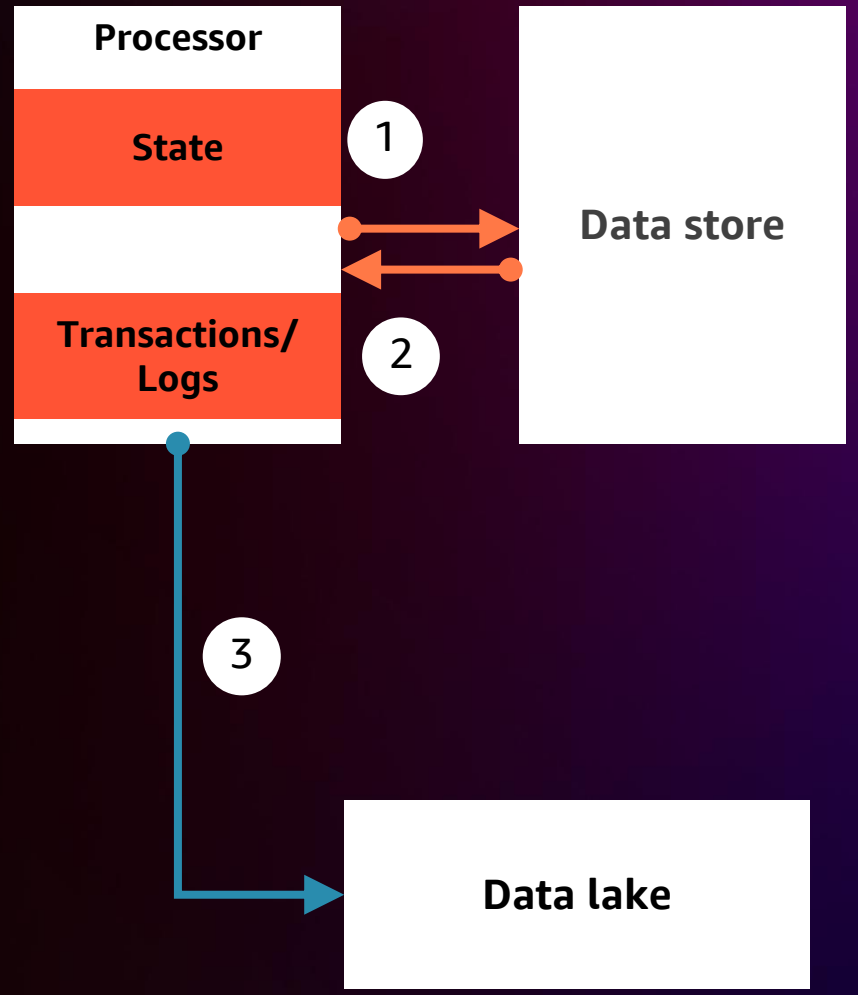
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Best practice 1 – Stateless architecture for auto scaling

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Stateful – Legacy

1. Select tables in memory, periodically refreshed/written to data stores
2. Transactions/Logs – written to local disk and data store; retain large, non-cached tables
3. Batch job – periodic batch job writes transactions/logs to data lake



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

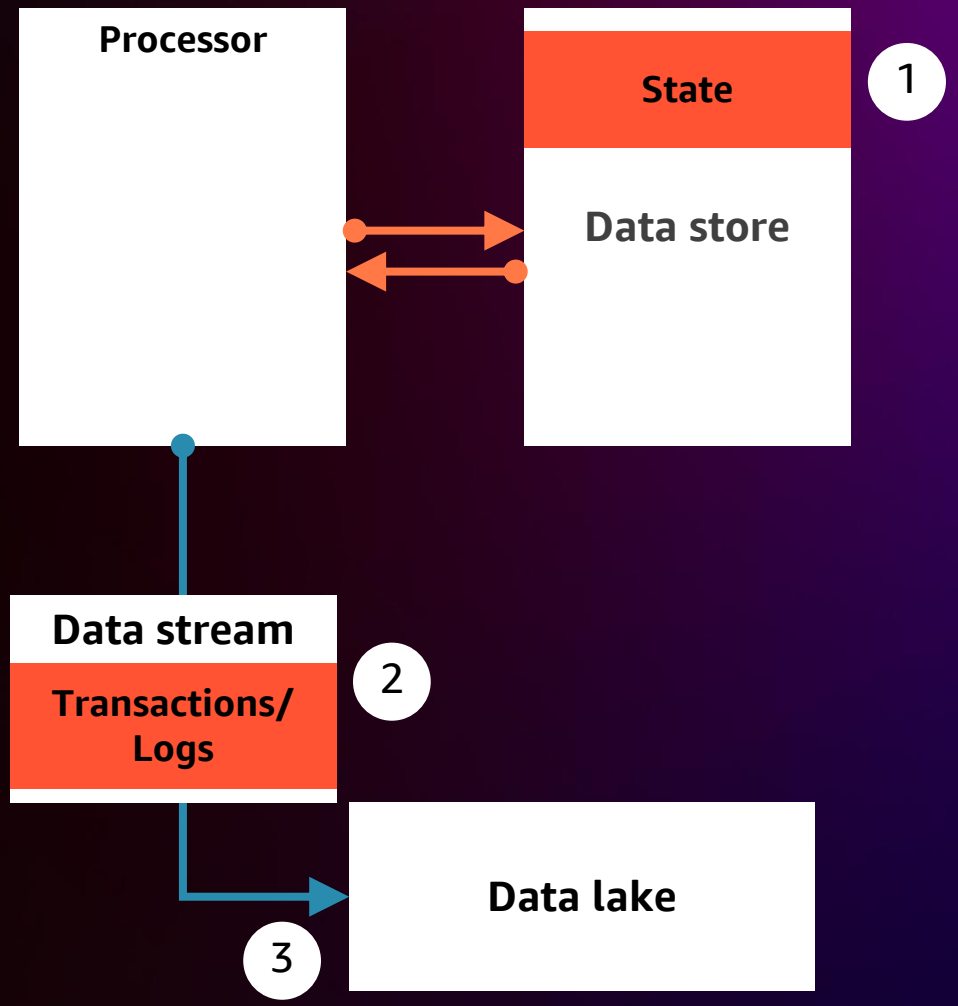
Stateless in 3 steps

Stateful

1. Select tables in memory, periodically refreshed/written to data stores
2. Transactions/Logs – written to local disk and data store; retain large, non-cached tables
3. Batch job – periodic batch job writes transactions/logs to data lake

Stateless

1. Move state to data store
2. Replace batch job with data stream
3. Write data stream into data lake

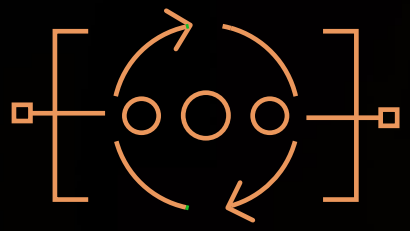


aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Best practice 2 – Spot Instances

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

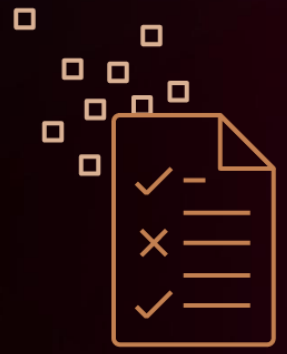
Spot Instances



Minimize boot time



Attribute-based selection and auto scaling



Capacity rebalance



Spot placement score

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Results from using Spot Instances

Quantcast

60%

smaato

61%

gumgum

62%

IPONWEB

80%

AppsFlyer

80%

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Best practice 3 – AWS Graviton instances

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Graviton–based Amazon EC2 instances

Better price performance

Up to

40%

Graviton2 over x86 instances

Up to

25%

Graviton3 over Graviton2 instances

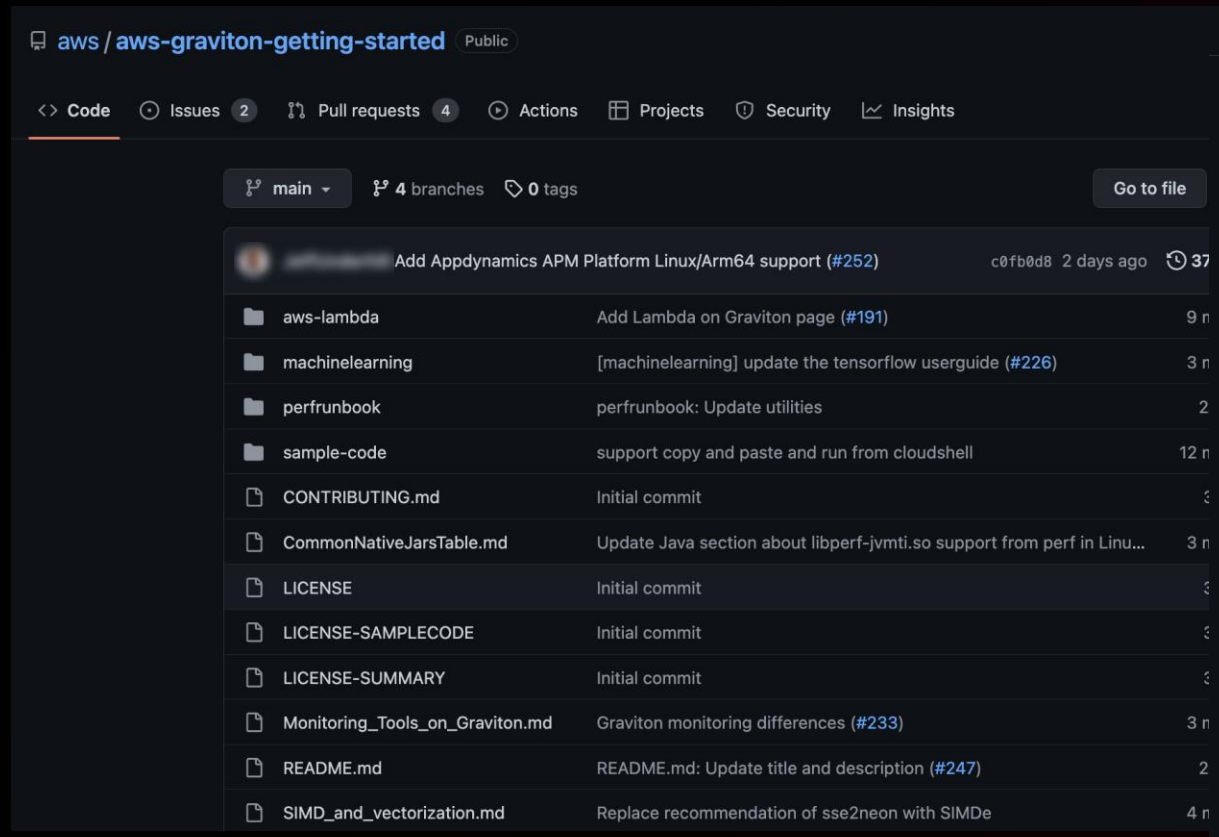
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Graviton2 ease of adoption

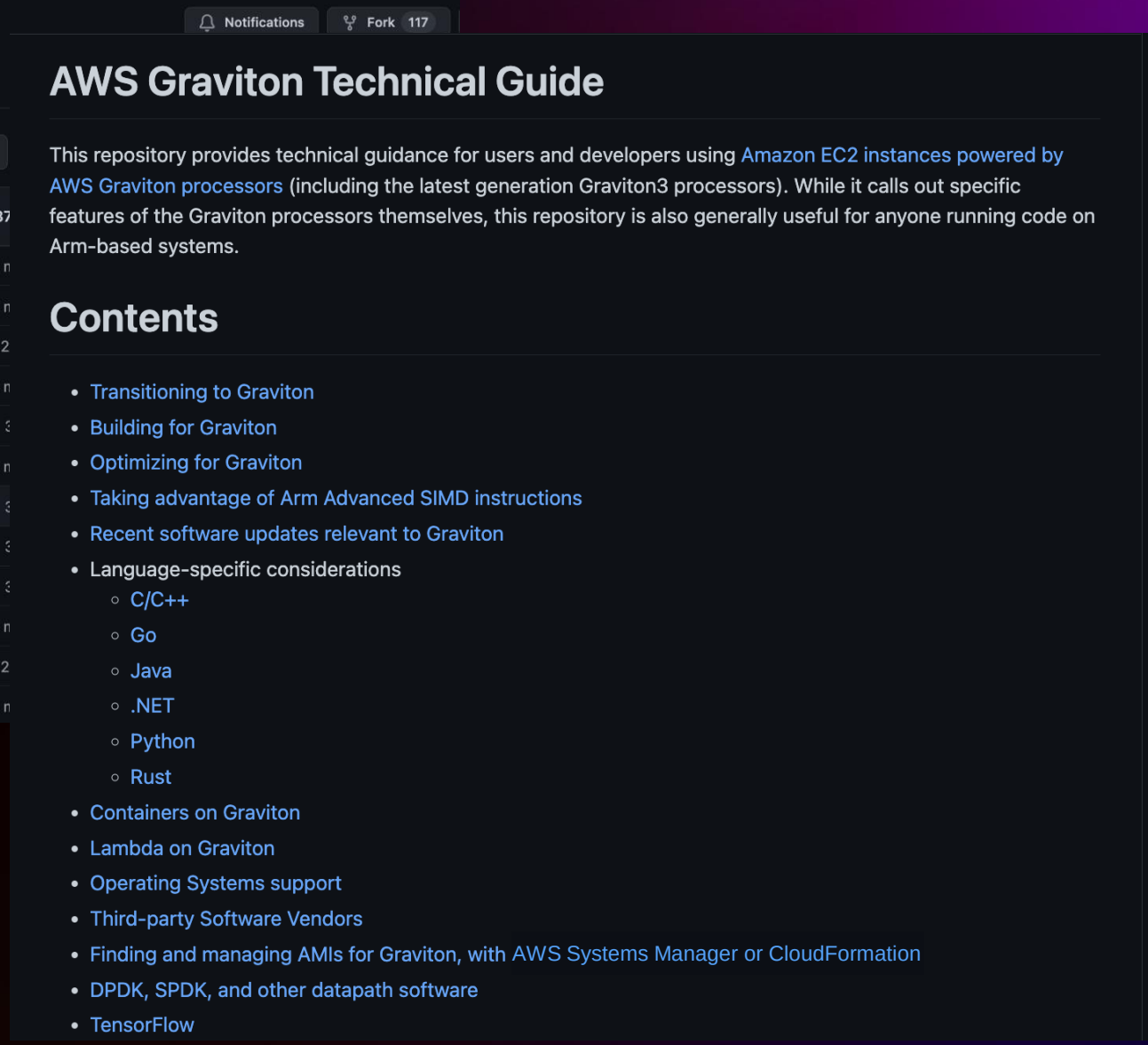
Difficulty	Workload	Actions
Virtually no effort	Amazon RDS, Amazon Aurora, Amazon ElastiCache, and Amazon OpenSearch Service	Upgrade to the latest and enjoy
Super easy	Amazon EMR	Typically just works
Quite easy	Linux – interpreted and JIT'd languages (e.g., Java, PHP, Node.js)	Select the Arm64 AMI and install; bonus if using containerized
More involved	Linux – compiled languages (e.g., C/C++, Python, Go)	Select Arm64 AMI and compile and port any intrinsics or assembly
Some work, high reward	Microsoft Windows – .NET	Migrate to .NET core, run on Linux/Arm64
Sorry, not yet	Microsoft Windows	Microsoft Windows Server not yet available for Arm64

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Graviton migration and optimization



https://github.com/aws/aws-graviton-getting-started



- Graviton performance runbook for benchmarking
- Profiling the code for performance optimization
- Language-specific considerations and work-arounds

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Results from using AWS Graviton2



50% reduction in compute costs for RTB using AWS Graviton2–based Amazon EC2 instances, using **M6g and C6g** instances (Erlang, Go, and Java)



50% performance increase between **M5 to M6g** (Java benchmark)



"We seamlessly transferred this Java application to Amazon EC2 A1 instances powered by the AWS Graviton processor. We've since tested the new Graviton2-based **M6g** instances and it was **able to handle twice the load** of an A1."

Chris Nicotra, SVP Digital, Nielsen



"We wanted to utilize instances which offered us a balance of compute and network throughput. Graviton2 powered instances (M6g/R6g) offered us both, while lowering our EC2 costs by **50 to 70%** over comparable x86based instances."

Jude Pereira, Principal Architect, CleverTap

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Results – 3 easy steps?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Results from using auto scaling, Spot Instances, and AWS Graviton2

Current status	% Cost Reduction	Cost reduction	Total Cost remaining
Application is stateful			$100

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Results from using auto scaling, Spot Instances, and AWS Graviton2

Current status	% Cost Reduction	Cost reduction	Total Cost remaining
Application is stateful			$100
Move to stateless			
Implement auto scaling	32%	$32	$68

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Results from using auto scaling, Spot Instances, and AWS Graviton2

Current status	% Cost Reduction	Cost reduction	Total Cost remaining
Application is stateful			$100
Move to stateless			
Implement auto scaling	32%	$32	$68
On-Demand to Spot Instances	60%	$41	$27

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Results from using auto scaling, Spot Instances, and AWS Graviton2

Current status	% Cost Reduction	Cost reduction	Total Cost remaining
Application is stateful			$100
Move to stateless			
Implement auto scaling	32%	$32	$68
On-Demand to Spot Instances	60%	$41	$27
Port to Graviton2	40%	$10.88	**$16.32**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Results from using auto scaling, Spot Instances, and AWS Graviton2

Current status	% Cost Reduction	Cost reduction	Total Cost remaining
Application is stateful			$100
Move to stateless			
Implement auto scaling	32%	$32	$68
On-Demand to Spot Instances	60%	$41	$27
Port to Graviton2	40%	$10.88	**$16.32**
		Total Cost reduction	**84%**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Best practice 4 – Data and data pipeline

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Data and data pipelines

Data streaming: Amazon Kinesis Data Steams (provisioned Amazon Kinesis)

- PUT payload unit (25 KB)
- PUT payload units (per 1 million)

Databases: Aerospike

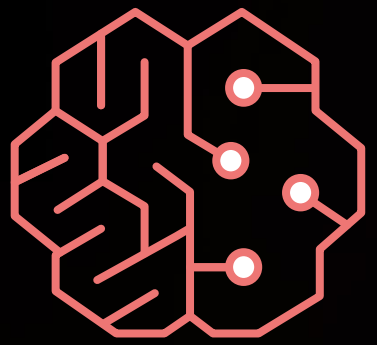
- Configure rack-aware READ
- I3en–i4i
- Aerospike on Graviton

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Best practice 5 – Reducing data transfer costs

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

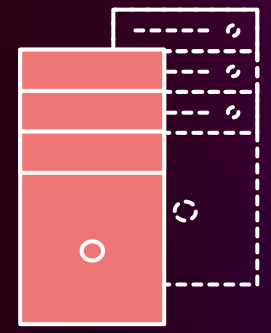
Reducing data transfer costs



Traffic filtering



AWS PrivateLink



Availability Zone

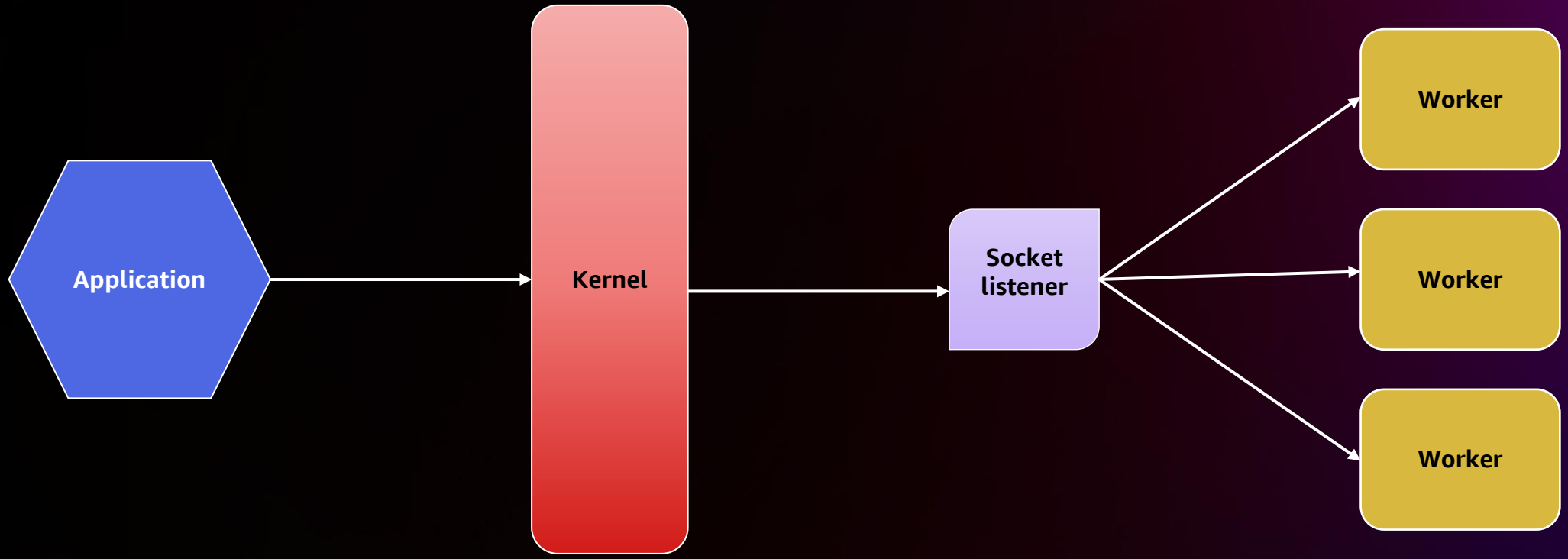
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Real-time bidding on AWS – Latency and throughput

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

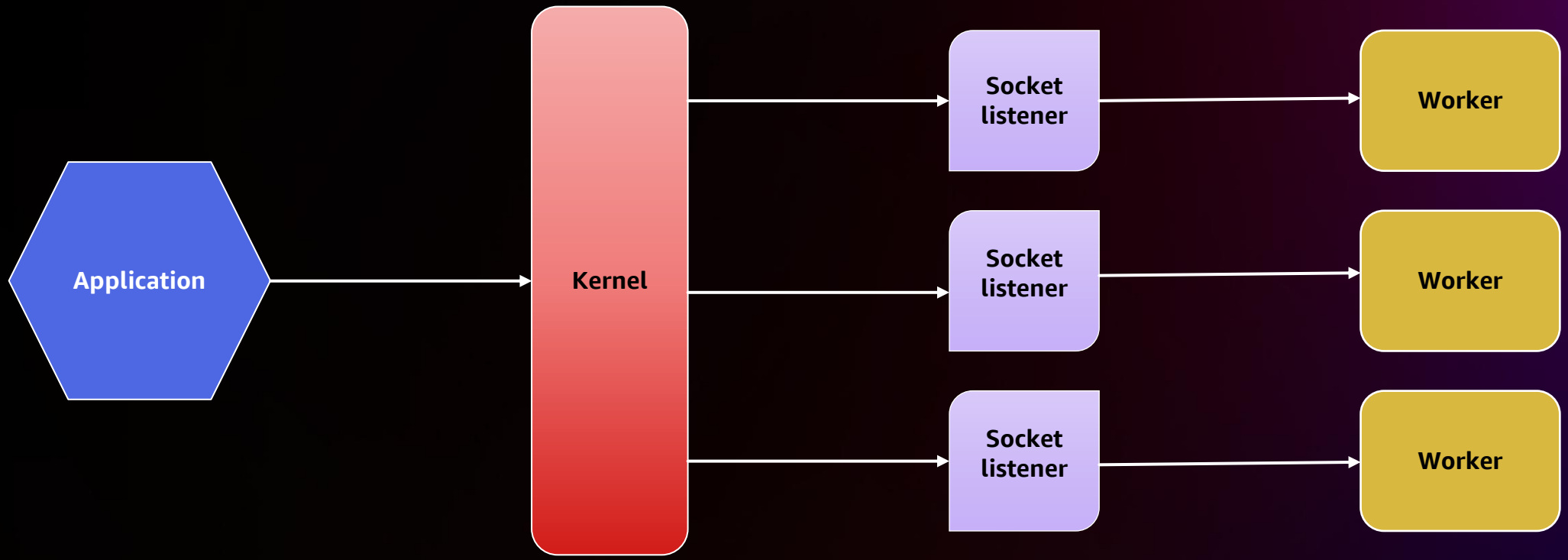
Low latency 1 – Enable SO_REUSEPORT



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Low latency 1 – Enable SO_REUSEPORT

With the SO_REUSEPORT option enabled, there are multiple socket listeners for each IP address and port combination, one for each worker process



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Low latency 2 – Optimize instances per CPU

1 bidder on 64 CPUs – 26.4K / 0.99 = 30 ms



32 bidders, each one on 2 CPUs – 100.27K / 0.99 = 12 ms



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Low latency 3 – Use FastJSON

Parsing a bid request takes 14% of the CPU load

Using standard JSON: 100.27K / 0.99 = 12 ms



Using FastJSON: 124.02K / 0.99 = 6 ms



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Low latency 4 – Run fastHTTP

Using net_HTTP: 147.33K / 0.99 = 5 ms



Using fastHTTP: 176.91K / 0.99 = 990 µs

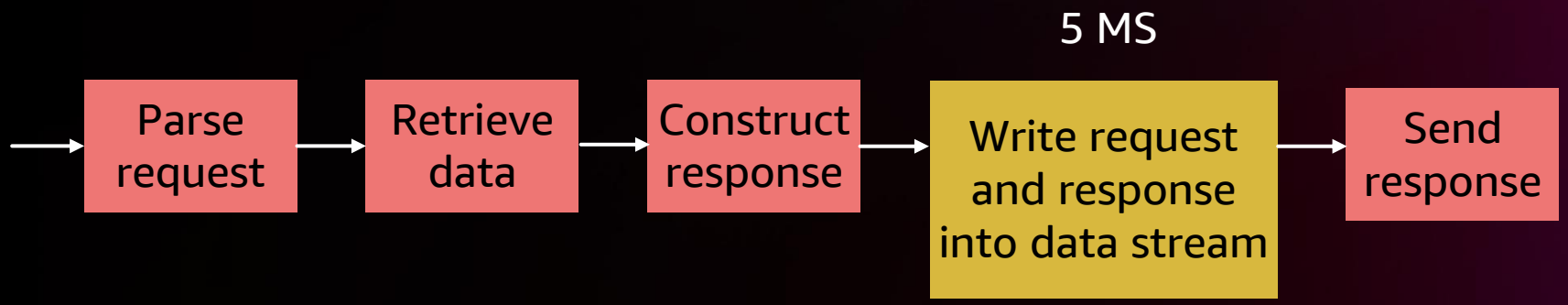


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

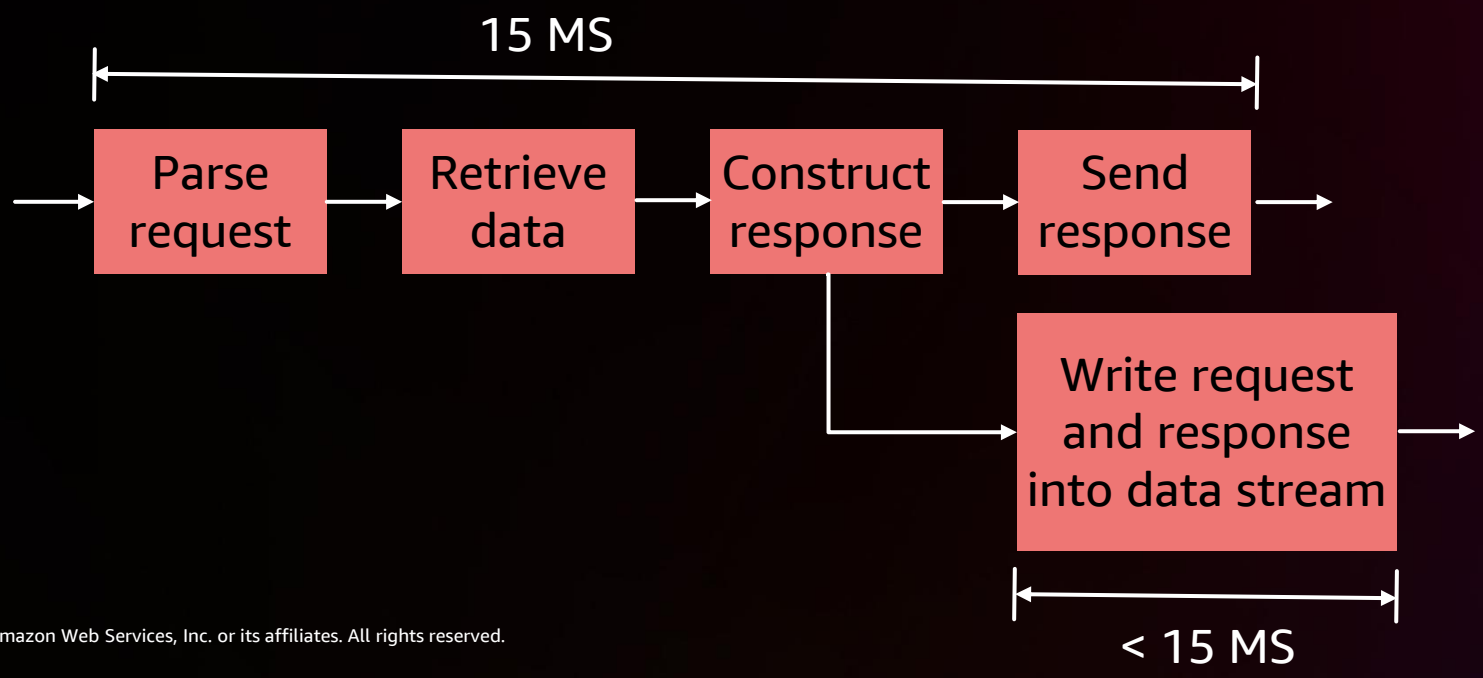
Low latency 5 – Nonblocking data streams

Make data streaming a non-blocking event

BEFORE



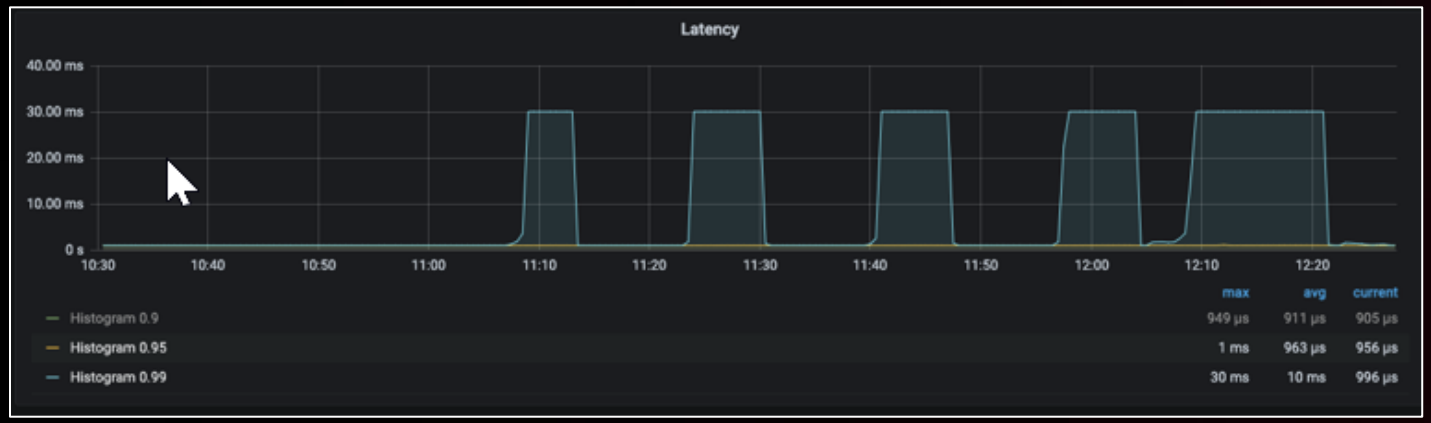
AFTER



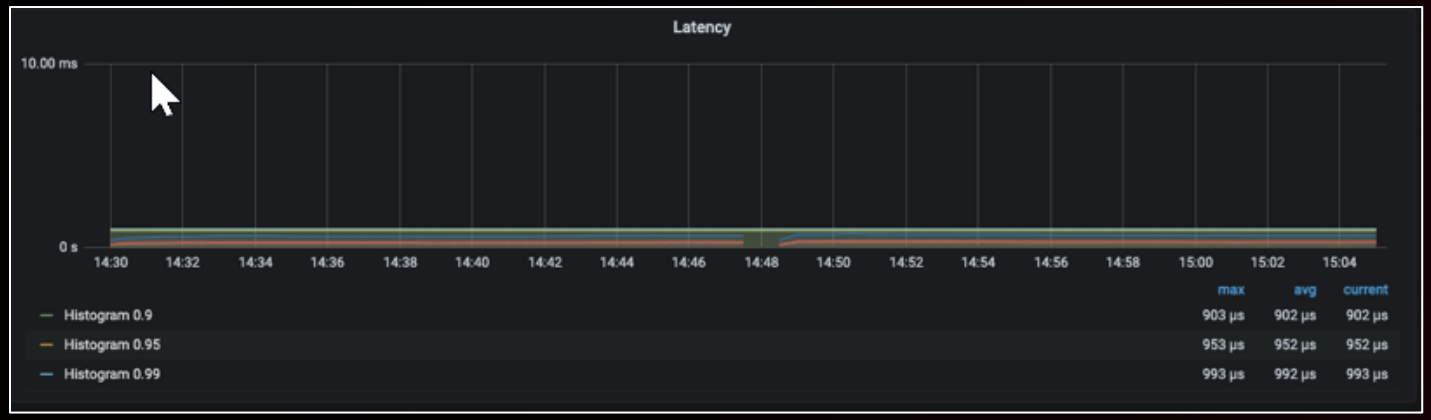
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Low latency 6 – Use ZStandard compression

Using the standard Kinesis Producer Library: 10 ms



Using Kinesis_zstd: 992 µs



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What are the results?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Benchmark results – Bidder QPS and latency

Experiment Results

Baseline Instance: One **m6g.16 xl**, 64 vCPUs
CPU Utilization 70%

BIDDER	MAX QPS (in thousands)	Latency at 99th percentile (in milliseconds)
baseline	**7.48**	**30**
SO_REUSEPORT = ON	29.7	30
optimize instance count (1 Bidder on 64 vCPUs)	29.7	30
optimize instance count (32 bidder on 2 vCPUs)	100.27	12
use standard JSON	100.27	12
use FastJSON	124.02	6
use net_HTTP	147.33	5
use fastHTTP	**176.91**	**0.99**
TOTAL IMPROVEMENT	**169.43**	**29.01**
% Improvement	**2165%**	**2930%**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Solutions Library

ADVERTISING AND MARKETING

Guidance for building a real-time bidder on AWS

aws.amazon.com/solutions/guidance/building-a-real-time-bidder-for-advertising-on-aws



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Victor Gershkovich
victor@appsflyer.com

Gerry Louw
louwgerr@amazon.com

Akhil Aendapally
aaendapa@amazon.com



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.