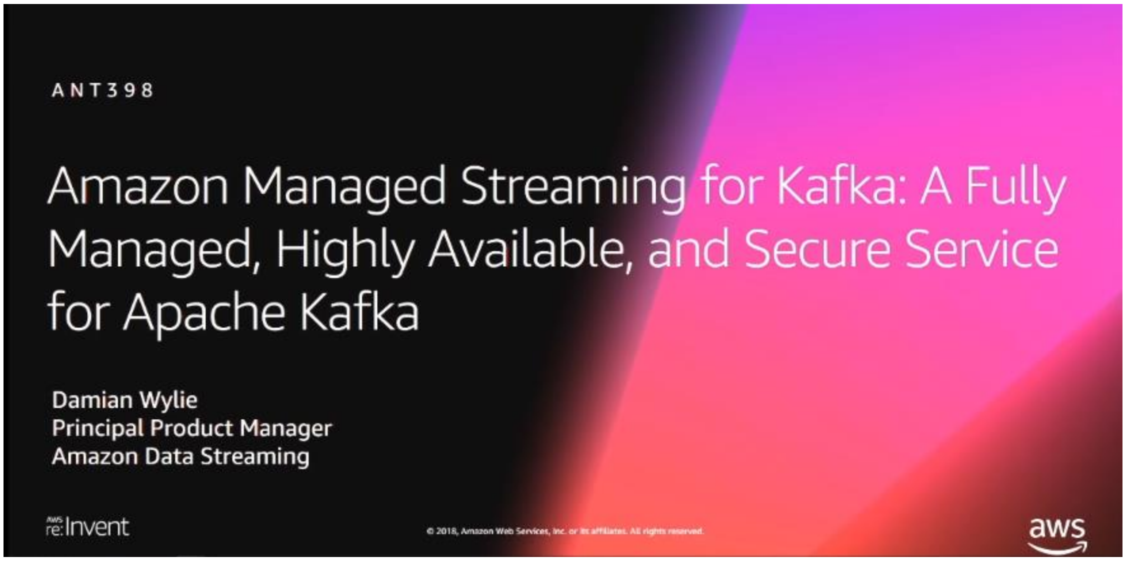


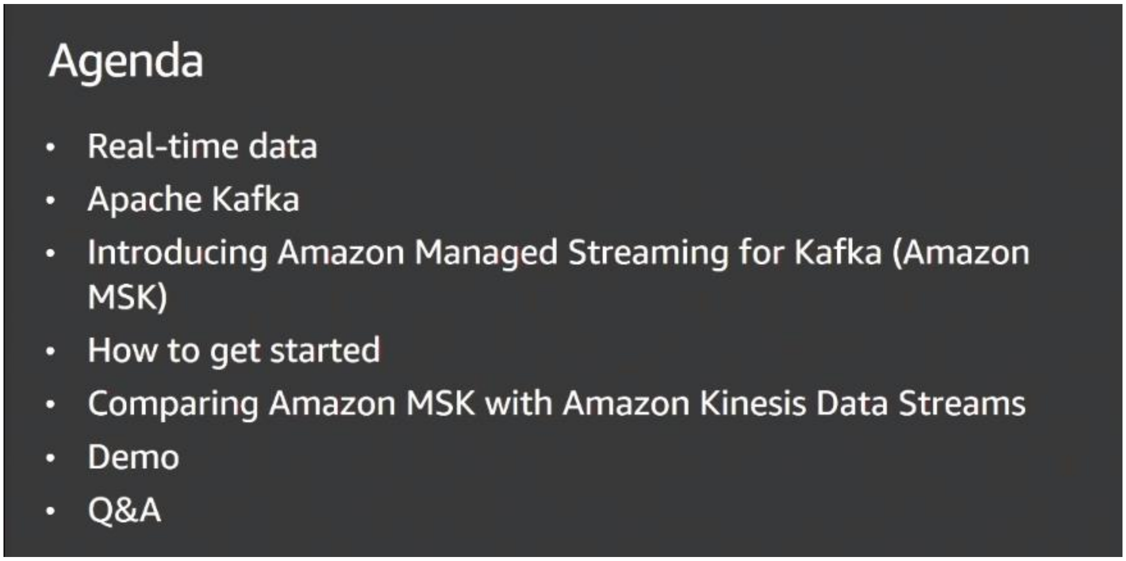
ANT398

Amazon Managed Streaming for Kafka: A Fully Managed, Highly Available, and Secure Service for Apache Kafka

Damian Wylie
Principal Product Manager
Amazon Data Streaming

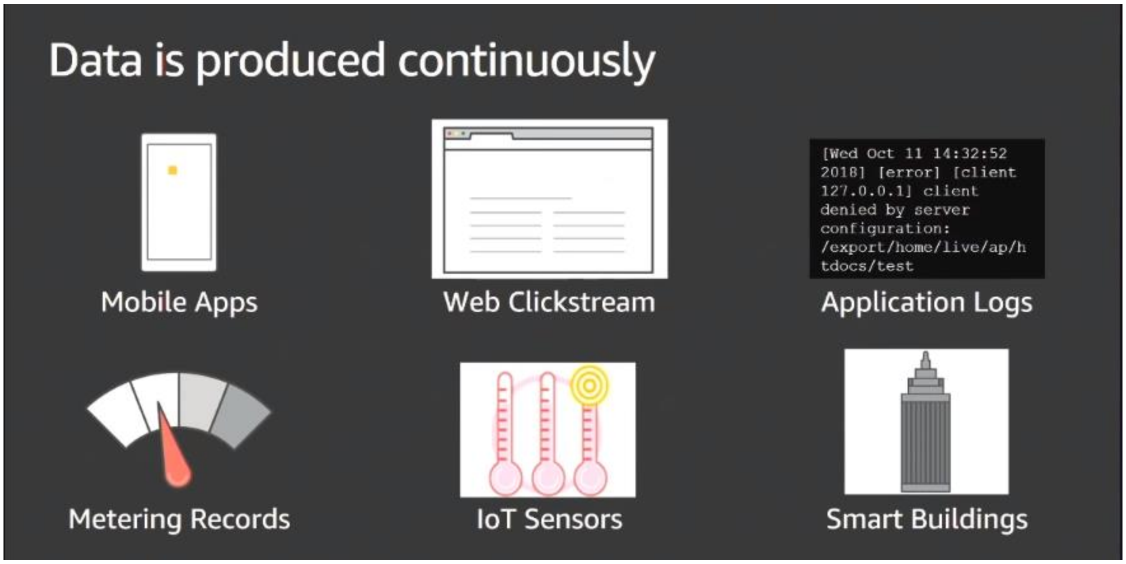
re:Invent
aws

Discover the power of running Apache Kafka on a fully managed AWS service. In this session, we describe how Amazon Managed Streaming for Kafka (Amazon MSK) runs Apache Kafka clusters for you, demo Amazon MSK and a migration, show you how to get started, and walk through other important details about the new service.



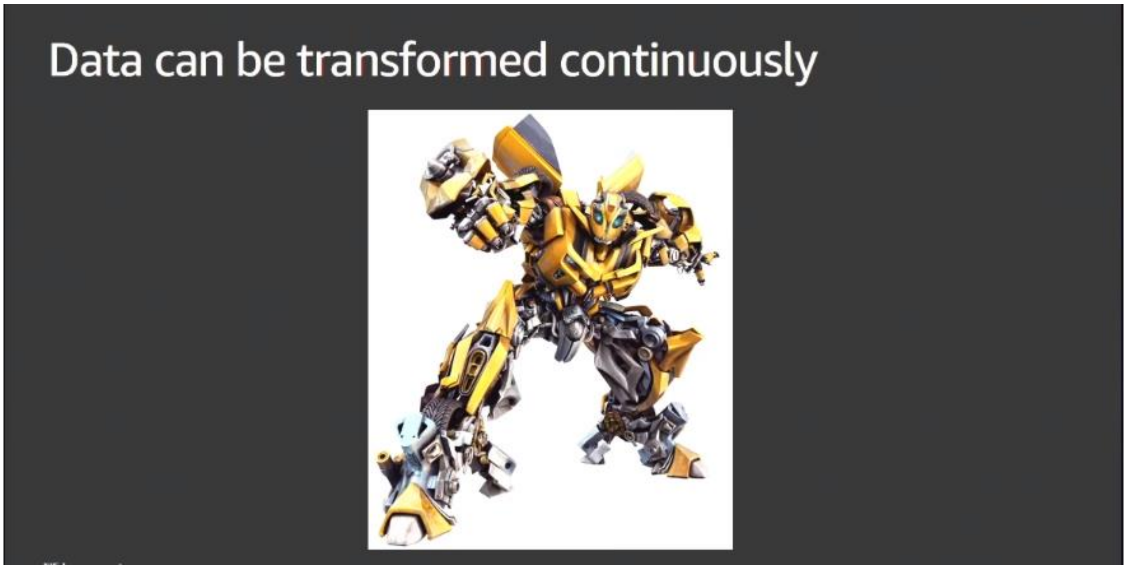
Agenda

- Real-time data
- Apache Kafka
- Introducing Amazon Managed Streaming for Kafka (Amazon MSK)
- How to get started
- Comparing Amazon MSK with Amazon Kinesis Data Streams
- Demo
- Q&A

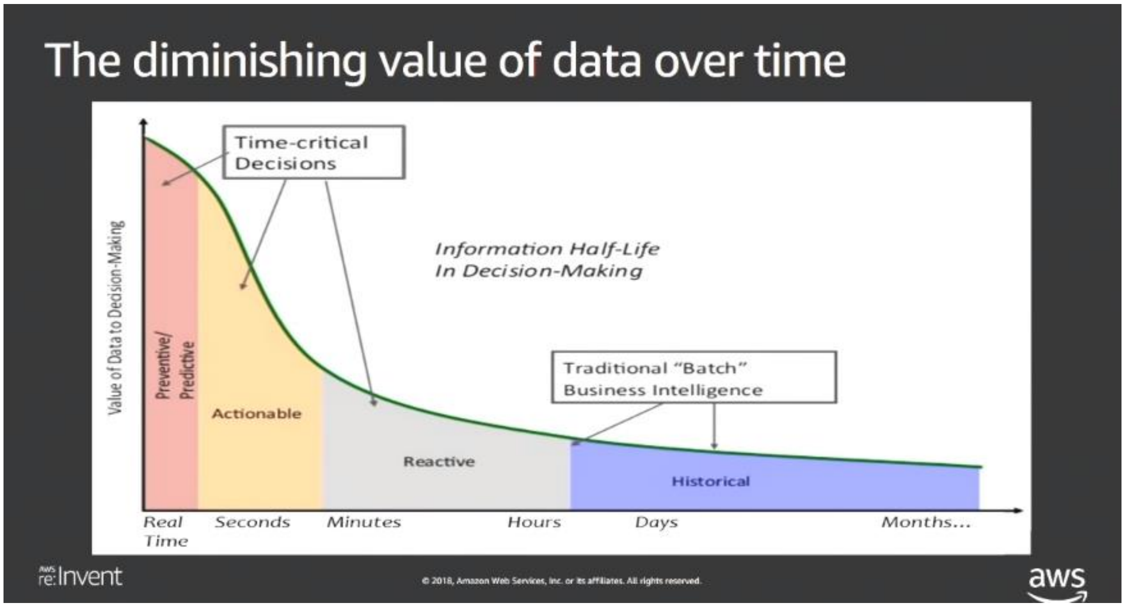


Data is produced continuously

```
[Wed Oct 11 14:32:52
2018] [error] [client
127.0.0.1] client
denied by server
configuration:
/export/home/live/ap/h
tdocs/test
```



Data can be transformed continuously



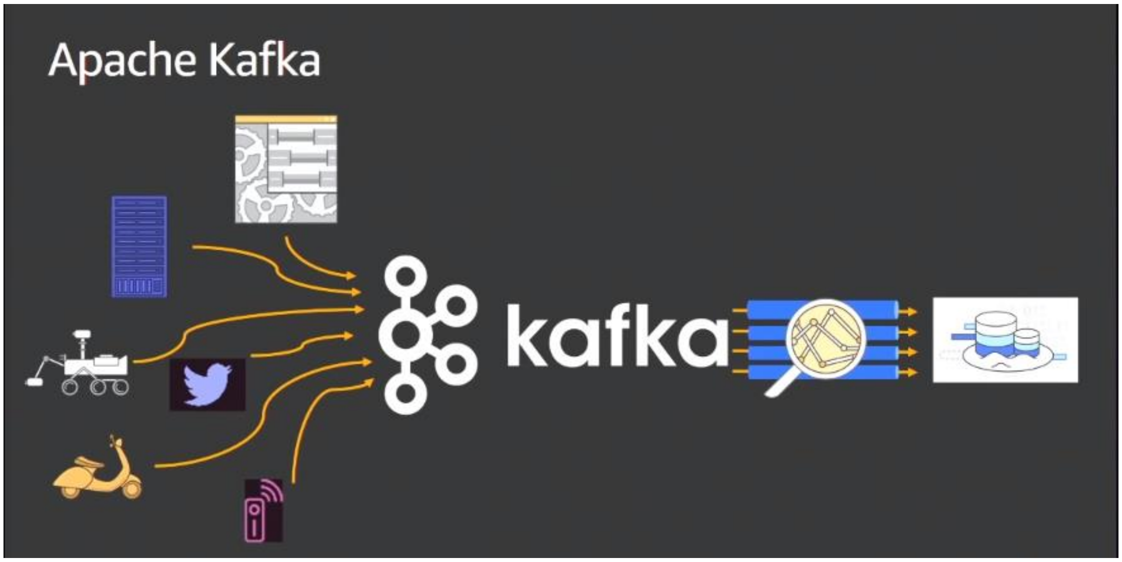
The diminishing value of data over time

re:Invent

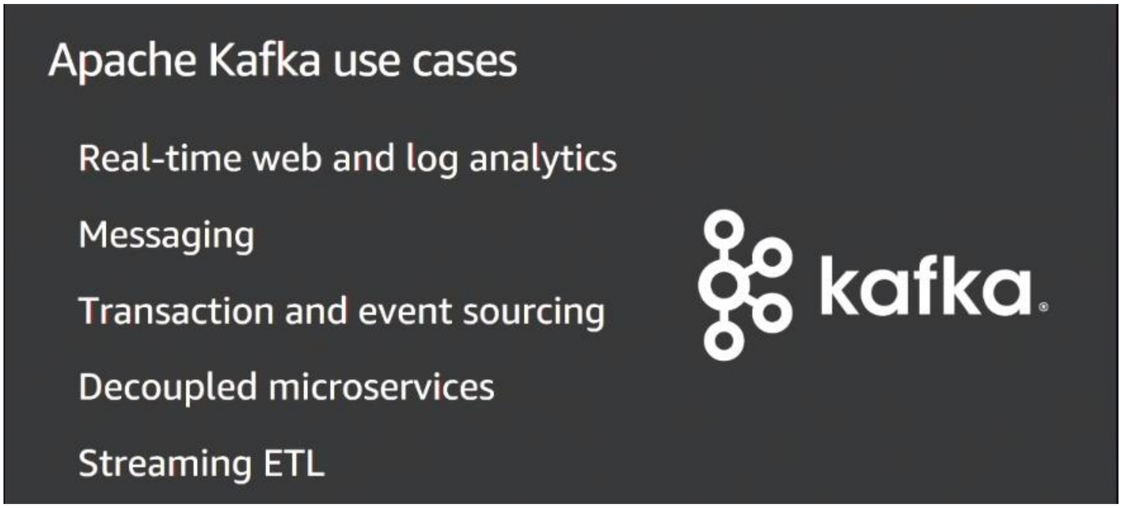
aws

kafka®

Apache Kafka



Apache Kafka use cases



Real-time web and log analytics

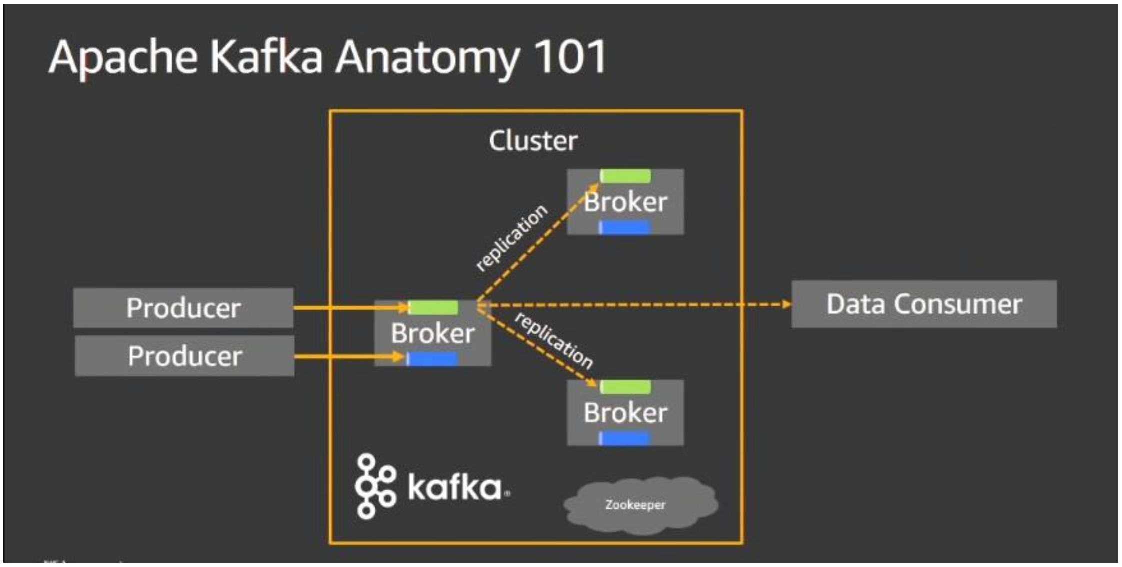
Messaging

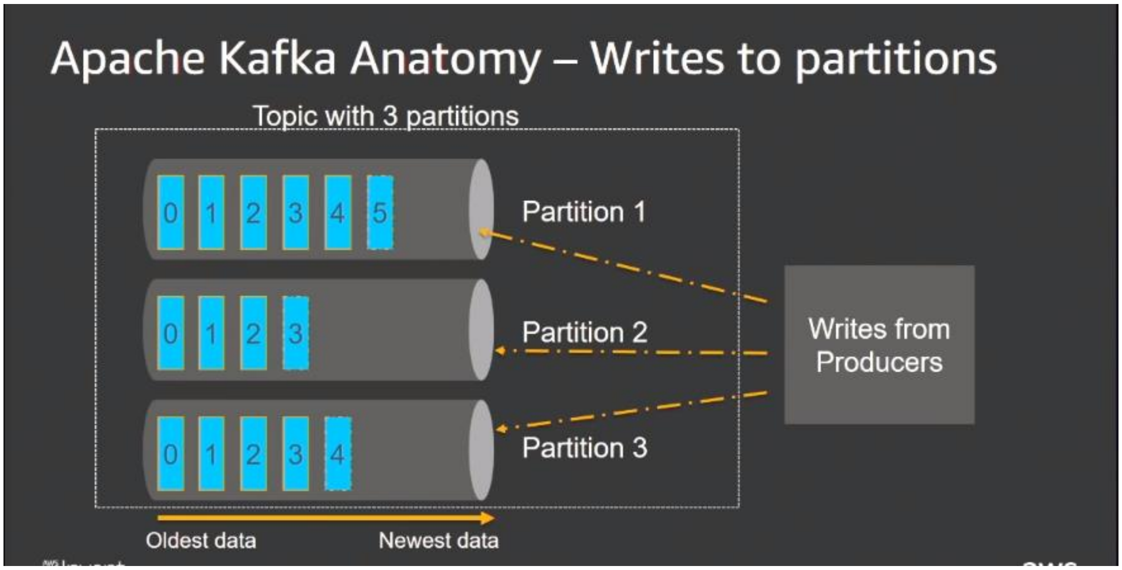
Transaction and event sourcing

Decoupled microservices

Streaming ETL

Apache Kafka Anatomy 101





Apache Kafka Anatomy – Writes to partitions

Topic with 3 partitions

Partition 1

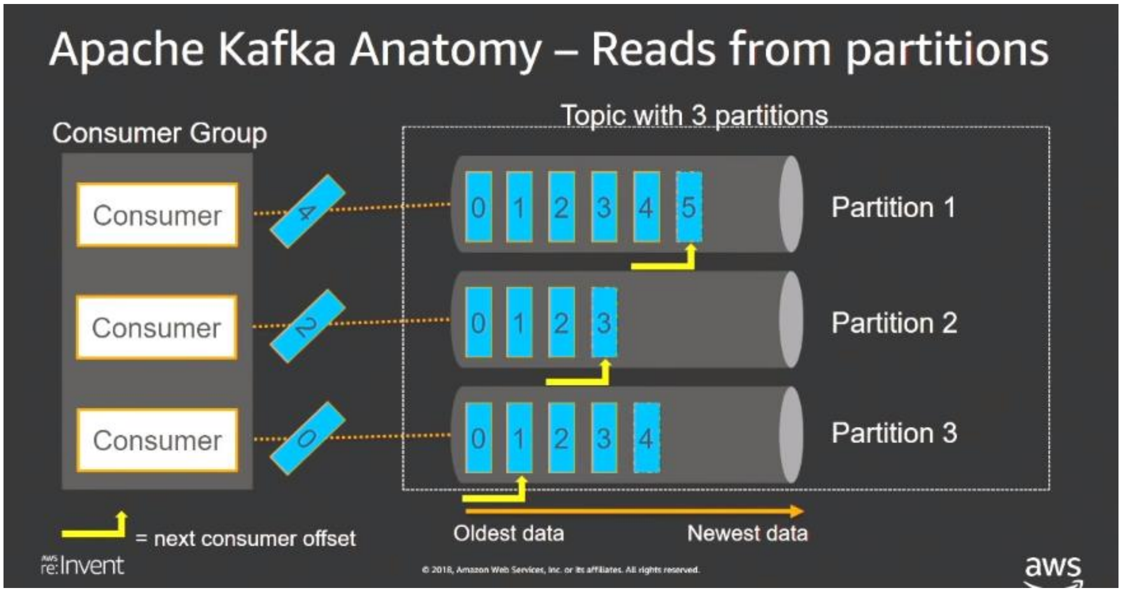
Partition 2

Partition 3

Writes from Producers

Oldest data

Newest data



Apache Kafka Anatomy – Reads from partitions

Consumer Group

Consumer

Consumer

Consumer

Topic with 3 partitions

Partition 1

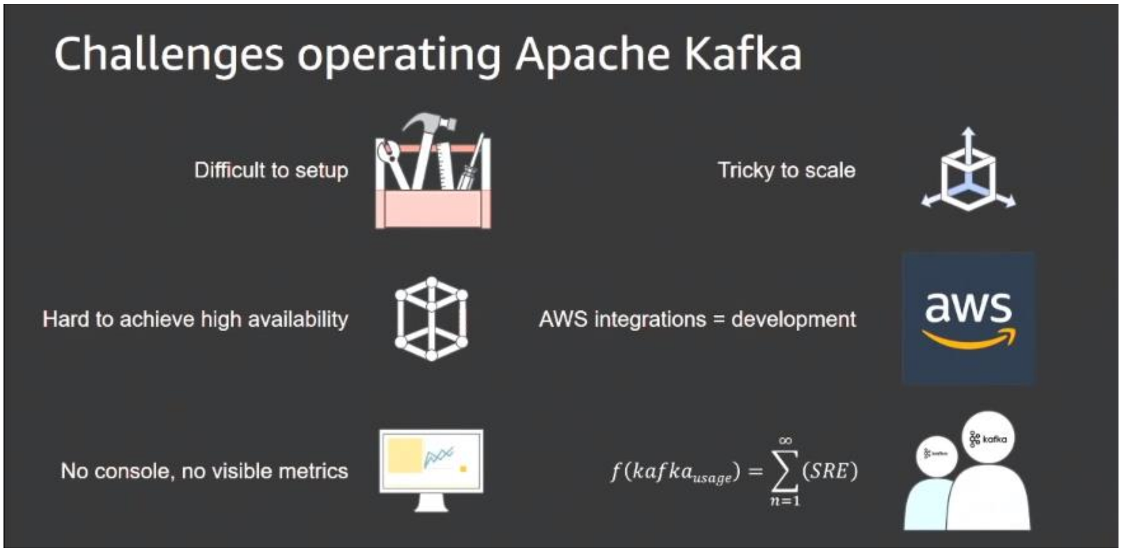
Partition 2

Partition 3

= next consumer offset

Oldest data

Newest data



Challenges operating Apache Kafka

Difficult to setup

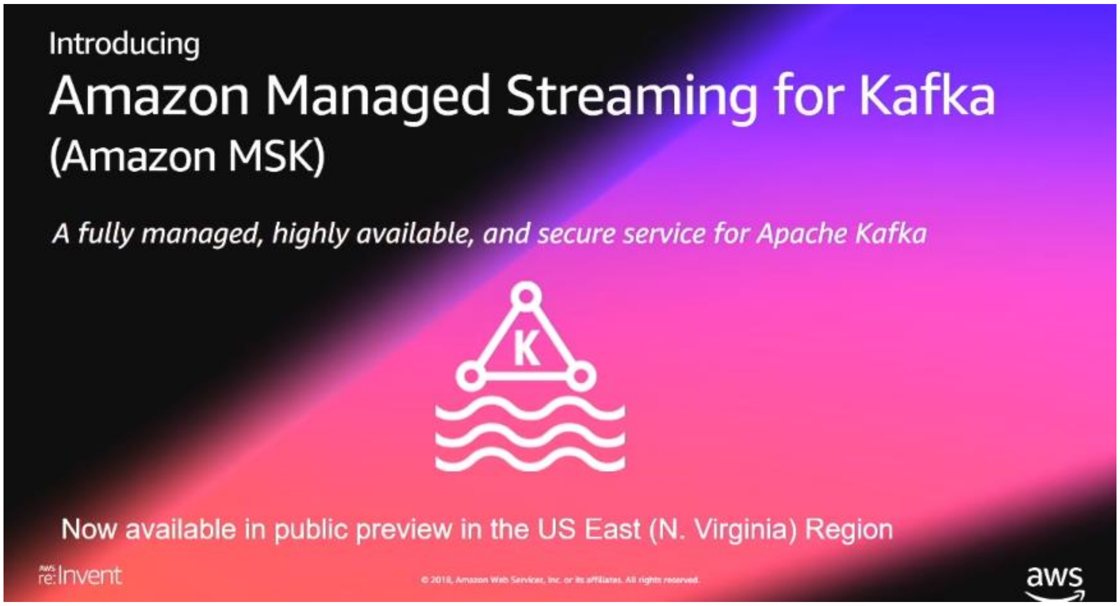
Tricky to scale

Hard to achieve high availability

AWS integrations = development

No console, no visible metrics

$f(kafka_{usage}) = \sum_{n=1}^{\infty}(SRE)$

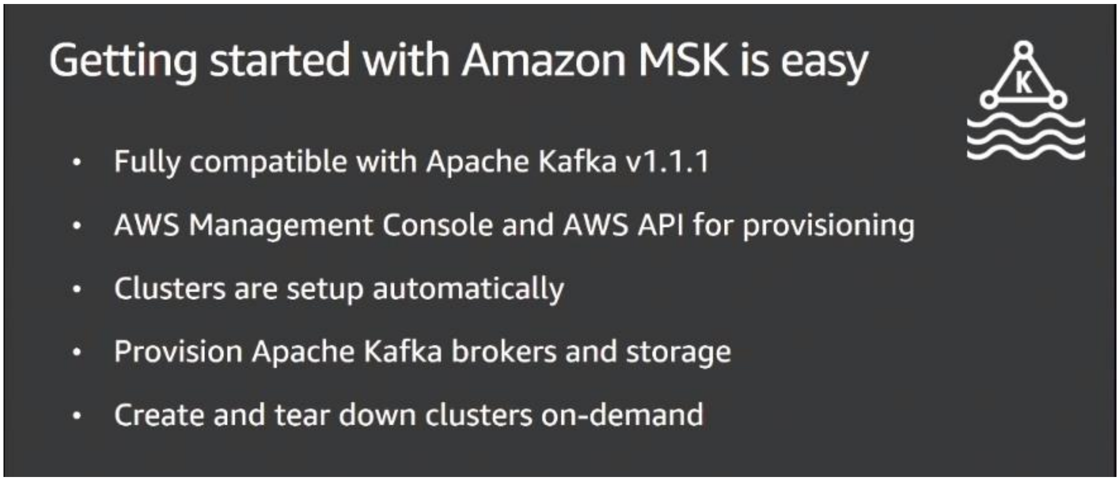


Introducing

Amazon Managed Streaming for Kafka (Amazon MSK)

A fully managed, highly available, and secure service for Apache Kafka

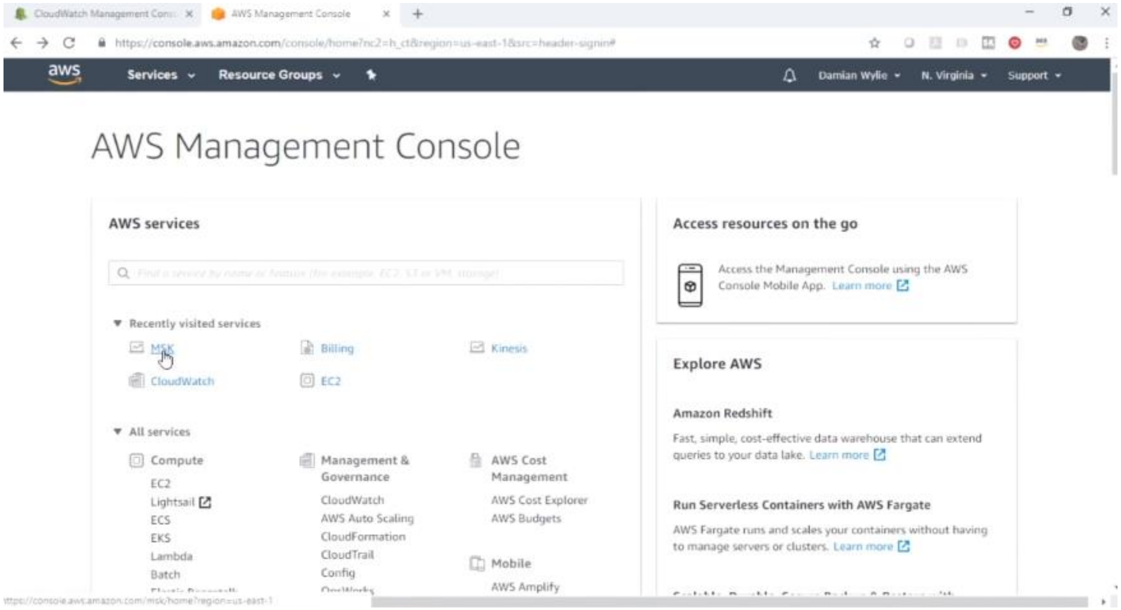
Now available in public preview in the US East (N. Virginia) Region



Getting started with Amazon MSK is easy

- Fully compatible with Apache Kafka v1.1.1
- AWS Management Console and AWS API for provisioning
- Clusters are setup automatically
- Provision Apache Kafka brokers and storage
- Create and tear down clusters on-demand

Create cluster,
producer/consumer
demo



CloudWatch Management Cons
AWS Management Console
https://console.aws.amazon.com/console/home?nc2=h_ct&region=us-east-1&src=header-signin#
aws
Services
Resource Groups
Damian Wylie
N. Virginia
Support

AWS Management Console

AWS services

▼ Recently visited services

MSK | Billing | Kinesis
CloudWatch | EC2

▼ All services

Compute
EC2
Lightsail
ECS
EKS
Lambda
Batch

Management & Governance
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config

AWS Cost Management
AWS Cost Explorer
AWS Budgets

Mobile
AWS Amplify

Access resources on the go

Access the Management Console using the AWS Console Mobile App. Learn more

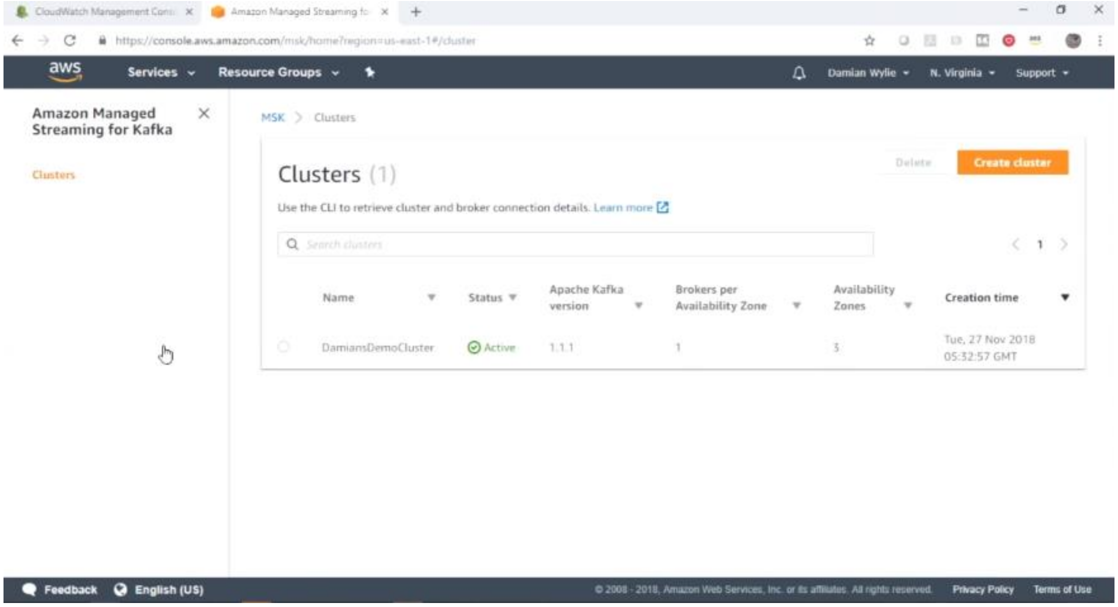
Explore AWS

Amazon Redshift

Fast, simple, cost-effective data warehouse that can extend queries to your data lake. Learn more

Run Serverless Containers with AWS Fargate

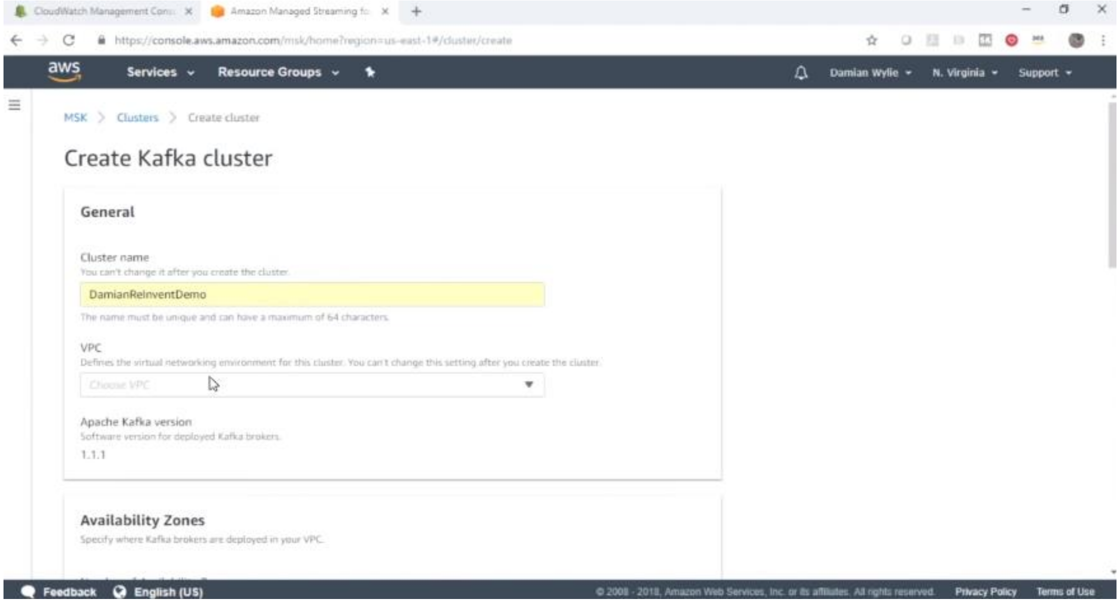
AWS Fargate runs and scales your containers without having to manage servers or clusters. Learn more


CloudWatch Management Cons
Amazon Managed Streaming fo
https://console.aws.amazon.com/msk/home?region=us-east-1#/cluster
aws
Services
Resource Groups
Damian Wylie
N. Virginia
Support
Amazon Managed Streaming for Kafka
Clusters
MSK > Clusters
Delete
Create cluster
Search clusters
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Clusters (1)

Use the CLI to retrieve cluster and broker connection details. Learn more

	Name	Status	Apache Kafka version	Brokers per Availability Zone	Availability Zones	Creation time
○	DamiansDemoCluster	Active	1.1.1	1	3	Tue, 27 Nov 2018 05:32:57 GMT


CloudWatch Management Cons
Amazon Managed Streaming fo
https://console.aws.amazon.com/msk/home?region=us-east-1#/cluster/create
aws
Services
Resource Groups
Damian Wylie
N. Virginia
Support
MSK > Clusters > Create cluster
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Create Kafka cluster

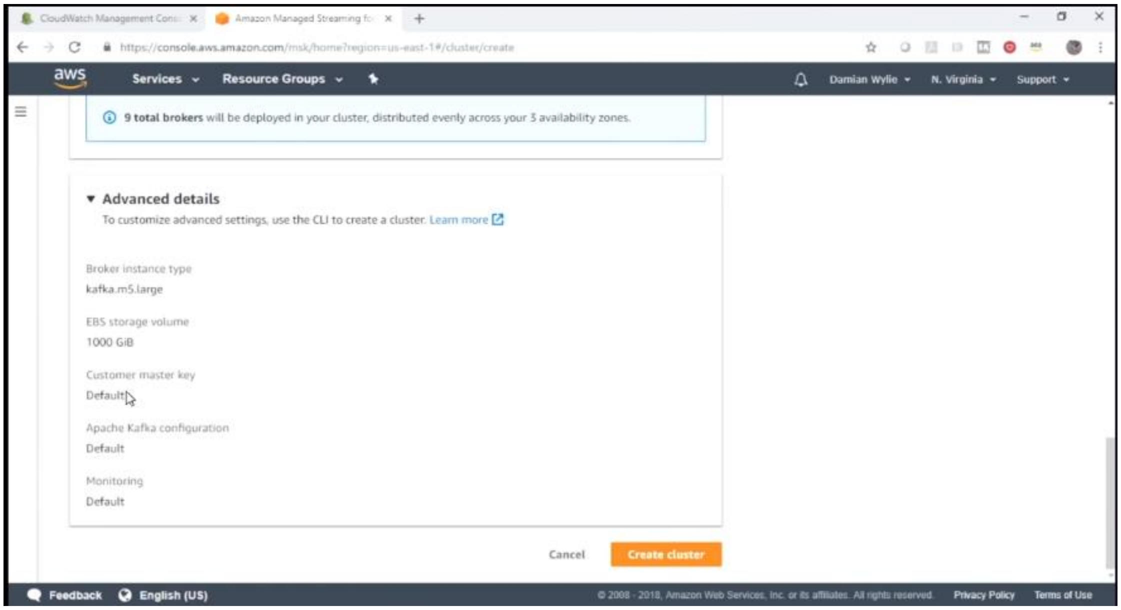
General

Cluster name
You can't change it after you create the cluster.
DamianReInventDemo
The name must be unique and can have a maximum of 64 characters.

VPC
Defines the virtual networking environment for this cluster. You can't change this setting after you create the cluster.
Choose VPC

Apache Kafka version
Software version for deployed Kafka brokers.
1.1.1

Availability Zones
Specify where Kafka brokers are deployed in your VPC.


CloudWatch Management Cons
Amazon Managed Streaming fo
https://console.aws.amazon.com/msk/home?region=us-east-1#/cluster/create
aws
Services
Resource Groups
Damian Wylie
N. Virginia
Support
Cancel
Create cluster
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

9 total brokers will be deployed in your cluster, distributed evenly across your 3 availability zones.

▼ Advanced details
To customize advanced settings, use the CLI to create a cluster. Learn more

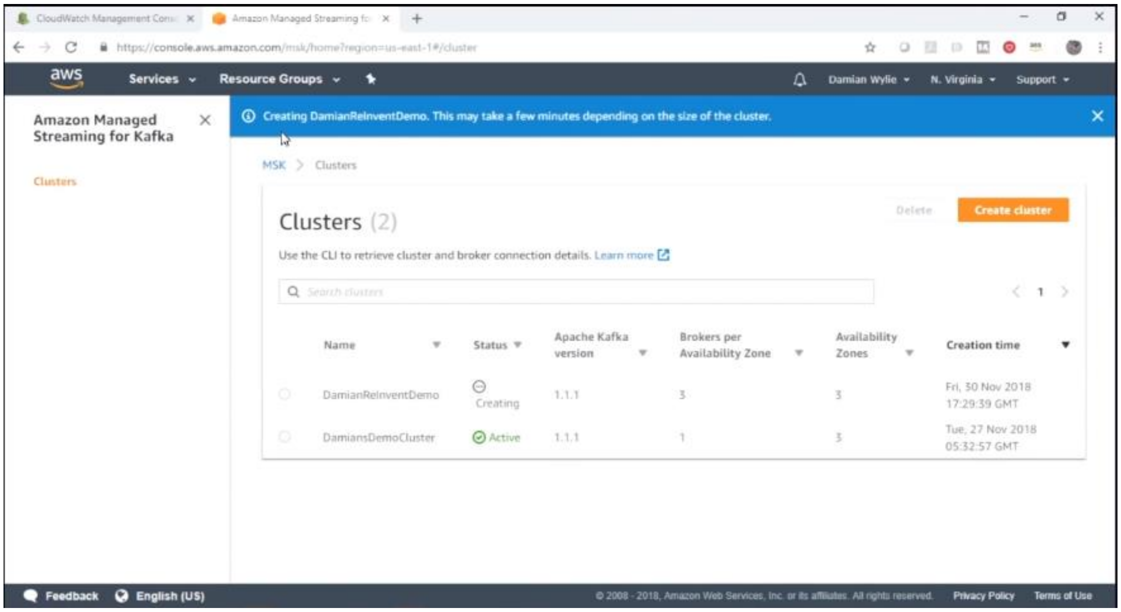
Broker instance type
kafka.m5.large

EBS storage volume
1000 GiB

Customer master key
Default

Apache Kafka configuration
Default

Monitoring
Default



CloudWatch Management Cons
Amazon Managed Streaming fo
https://console.aws.amazon.com/msk/home?region=us-east-1#/cluster
aws
Services
Resource Groups
Damian Wylie
N. Virginia
Support
Amazon Managed Streaming for Kafka
Clusters

Creating DamianReInventDemo. This may take a few minutes depending on the size of the cluster.

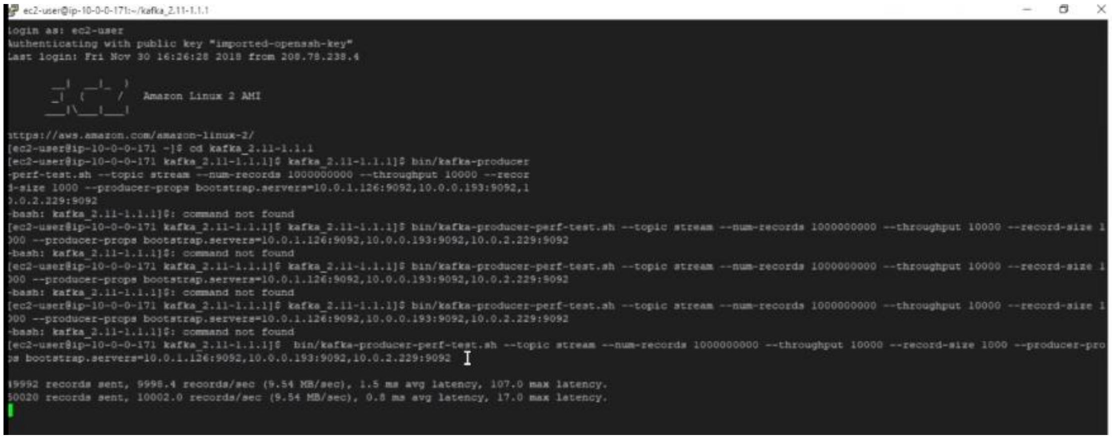
MSK > Clusters
Delete
Create cluster

Clusters (2)

Use the CLI to retrieve cluster and broker connection details. Learn more

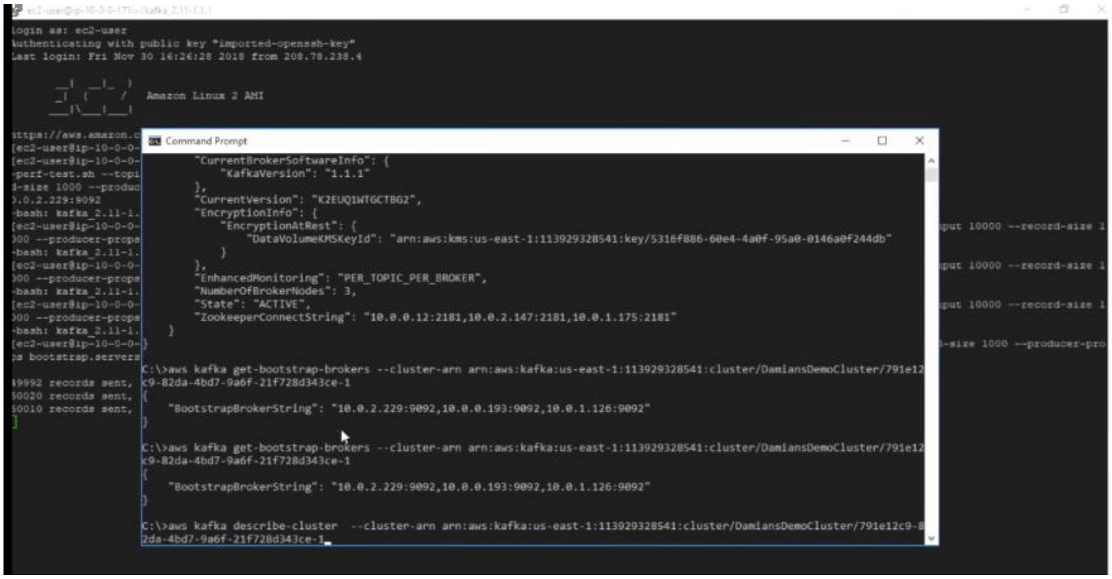
	Name	Status	Apache Kafka version	Brokers per Availability Zone	Availability Zones	Creation time
○	DamianReInventDemo	Creating	1.1.1	3	3	Fri, 30 Nov 2018 17:29:39 GMT
○	DamiansDemoCluster	Active	1.1.1	1	3	Tue, 27 Nov 2018 05:32:57 GMT

Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



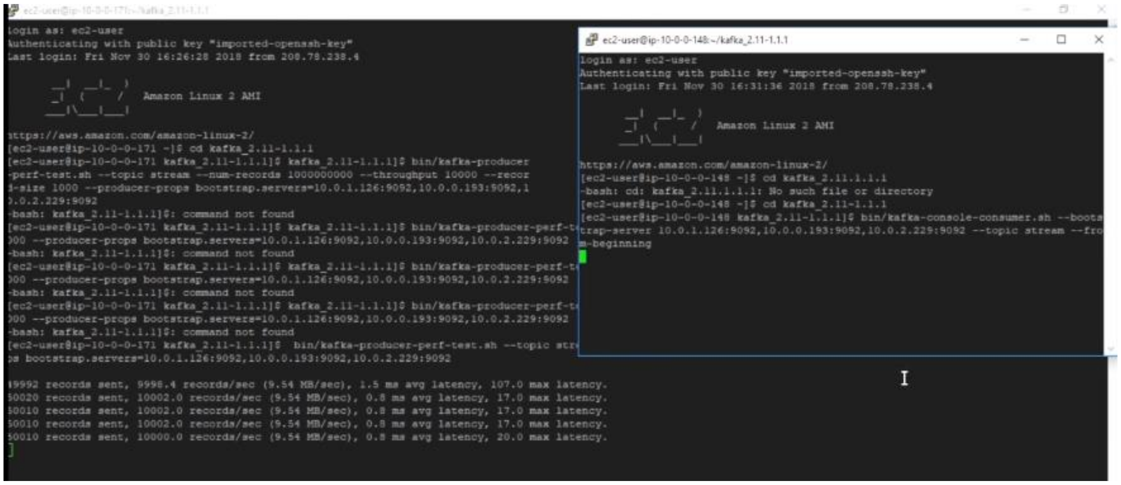
ec2-user@ip-10-0-0-171:~/kafka_2.11-1.1.1

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Fri Nov 30 16:26:28 2018 from 208.78.238.4

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-171 ~]$ cd kafka_2.11-1.1.1
[ec2-user@ip-10-0-0-171 kafka_2.11-1.1.1]$ kafka_2.11-1.1.1]$ bin/kafka-producer
-perf-test.sh --topic stream --num-records 1000000000 --throughput 10000 --recor
d-size 1000 --producer-props bootstrap.servers=10.0.1.126:9092,10.0.0.193:9092,1
0.0.2.229:9092
-bash: kafka_2.11-1.1.1]$: command not found
[ec2-user@ip-10-0-0-171 kafka_2.11-1.1.1]$ kafka_2.11-1.1.1]$ bin/kafka-producer-perf-test.sh --topic stream --num-records 1000000000 --throughput 10000 --record-size 1
000 --producer-props bootstrap.servers=10.0.1.126:9092,10.0.0.193:9092,10.0.2.229:9092
-bash: kafka_2.11-1.1.1]$: command not found
[ec2-user@ip-10-0-0-171 kafka_2.11-1.1.1]$ kafka_2.11-1.1.1]$ bin/kafka-producer-perf-test.sh --topic stream --num-records 1000000000 --throughput 10000 --record-size 1
000 --producer-props bootstrap.servers=10.0.1.126:9092,10.0.0.193:9092,10.0.2.229:9092
-bash: kafka_2.11-1.1.1]$: command not found
[ec2-user@ip-10-0-0-171 kafka_2.11-1.1.1]$ kafka_2.11-1.1.1]$ bin/kafka-producer-perf-test.sh --topic stream --num-records 1000000000 --throughput 10000 --record-size 1
000 --producer-props bootstrap.servers=10.0.1.126:9092,10.0.0.193:9092,10.0.2.229:9092
-bash: kafka_2.11-1.1.1]$: command not found
[ec2-user@ip-10-0-0-171 kafka_2.11-1.1.1]$  bin/kafka-producer-perf-test.sh --topic stream --num-records 1000000000 --throughput 10000 --record-size 1000 --producer-pro
ps bootstrap.servers=10.0.1.126:9092,10.0.0.193:9092,10.0.2.229:9092

49992 records sent, 9998.4 records/sec (9.54 MB/sec), 1.5 ms avg latency, 107.0 max latency.
50020 records sent, 10002.0 records/sec (9.54 MB/sec), 0.8 ms avg latency, 17.0 max latency.
```



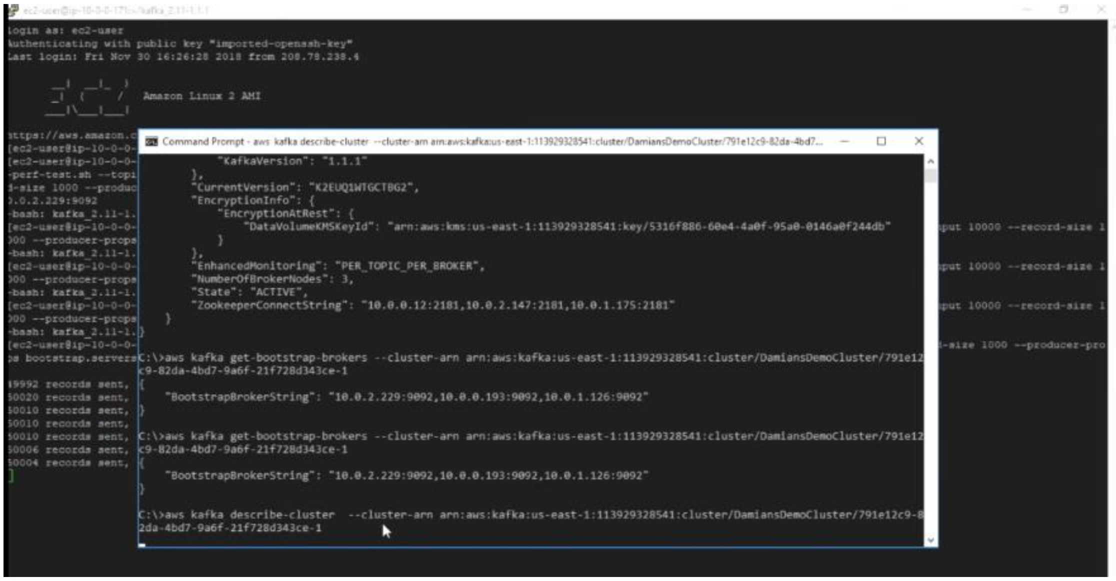
```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Fri Nov 30 16:26:28 2018 from 208.78.238.4

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|
```

```
        "CurrentBrokerSoftwareInfo": {
            "KafkaVersion": "1.1.1"
        },
        "CurrentVersion": "K2EUQ1WTGCTBG2",
        "EncryptionInfo": {
            "EncryptionAtRest": {
                "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:113929328541:key/5316f886-60e4-4a0f-95a0-0146a0f244db"
            }
        },
        "EnhancedMonitoring": "PER_TOPIC_PER_BROKER",
        "NumberOfBrokerNodes": 3,
        "State": "ACTIVE",
        "ZookeeperConnectString": "10.0.0.12:2181,10.0.2.147:2181,10.0.1.175:2181"
    }
}

C:\>aws kafka get-bootstrap-brokers --cluster-arn arn:aws:kafka:us-east-1:113929328541:cluster/DamiansDemoCluster/791e12
c9-82da-4bd7-9a6f-21f728d343ce-1
{
    "BootstrapBrokerString": "10.0.2.229:9092,10.0.0.193:9092,10.0.1.126:9092"
}

C:\>aws kafka get-bootstrap-brokers --cluster-arn arn:aws:kafka:us-east-1:113929328541:cluster/DamiansDemoCluster/791e12
c9-82da-4bd7-9a6f-21f728d343ce-1
{
    "BootstrapBrokerString": "10.0.2.229:9092,10.0.0.193:9092,10.0.1.126:9092"
}

C:\>aws kafka describe-cluster  --cluster-arn arn:aws:kafka:us-east-1:113929328541:cluster/DamiansDemoCluster/791e12c9-8
2da-4bd7-9a6f-21f728d343ce-1
```

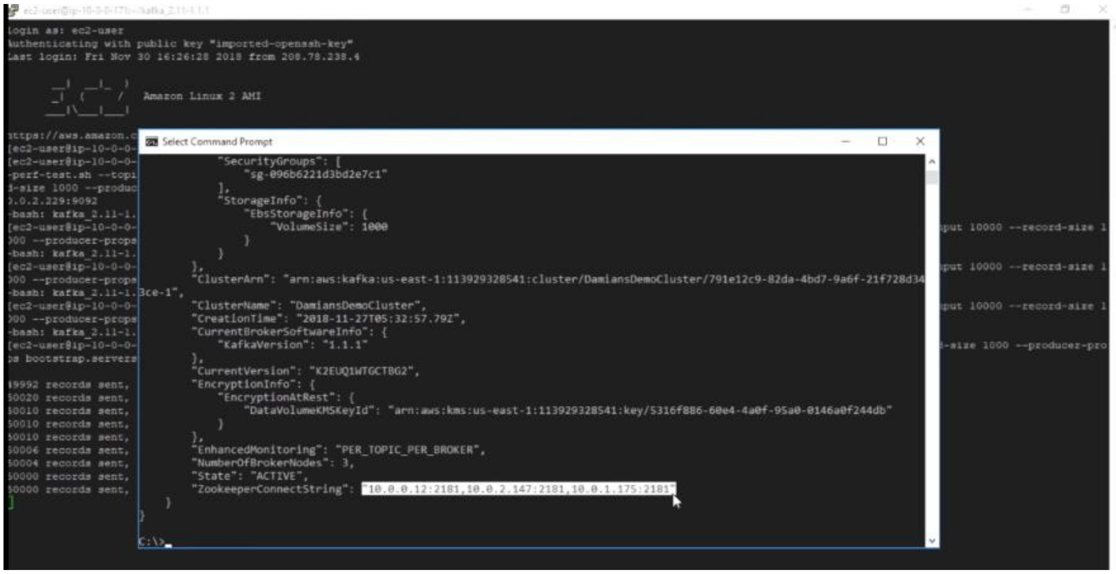


```
ec2-user@ip-10-0-0-171:~/kafka_2.11-1.1.1
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Fri Nov 30 16:26:28 2018 from 208.78.238.4

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-171 ~]$ cd kafka_2.11-1.1.1
[ec2-user@ip-10-0-0-171 kafka_2.11-1.1.1]$ kafka_2.11-1.1.1]$ bin/kafka-producer
-perf-test.sh --topic stream --num-records 1000000000 --throughput 10000 --recor
d-size 1000 --producer-props bootstrap.servers=10.0.1.126:9092,10.0.0.193:9092,1
0.0.2.229:9092
-bash: kafka_2.11-1.1.1]$: command not found
[ec2-user@ip-10-0-0-171 kafka_2.11-1.1.1]$ kafka_2.11-1.1.1]$ bin/kafka-producer-perf-t
000 --producer-props bootstrap.servers=10.0.1.126:9092,10.0.0.193:9092,10.0.2.229:9092
-bash: kafka_2.11-1.1.1]$: command not found
[ec2-user@ip-10-0-0-171 kafka_2.11-1.1.1]$ kafka_2.11-1.1.1]$ bin/kafka-producer-perf-t
000 --producer-props bootstrap.servers=10.0.1.126:9092,10.0.0.193:9092,10.0.2.229:9092
-bash: kafka_2.11-1.1.1]$: command not found
[ec2-user@ip-10-0-0-171 kafka_2.11-1.1.1]$ kafka_2.11-1.1.1]$ bin/kafka-producer-perf-t
000 --producer-props bootstrap.servers=10.0.1.126:9092,10.0.0.193:9092,10.0.2.229:9092
-bash: kafka_2.11-1.1.1]$: command not found
[ec2-user@ip-10-0-0-171 kafka_2.11-1.1.1]$  bin/kafka-producer-perf-test.sh --topic str
ps bootstrap.servers=10.0.1.126:9092,10.0.0.193:9092,10.0.2.229:9092

49992 records sent, 9998.4 records/sec (9.54 MB/sec), 1.5 ms avg latency, 107.0 max latency.
50020 records sent, 10002.0 records/sec (9.54 MB/sec), 0.8 ms avg latency, 17.0 max latency.
50010 records sent, 10002.0 records/sec (9.54 MB/sec), 0.8 ms avg latency, 17.0 max latency.
50010 records sent, 10002.0 records/sec (9.54 MB/sec), 0.8 ms avg latency, 17.0 max latency.
50010 records sent, 10000.0 records/sec (9.54 MB/sec), 0.8 ms avg latency, 20.0 max latency.
```

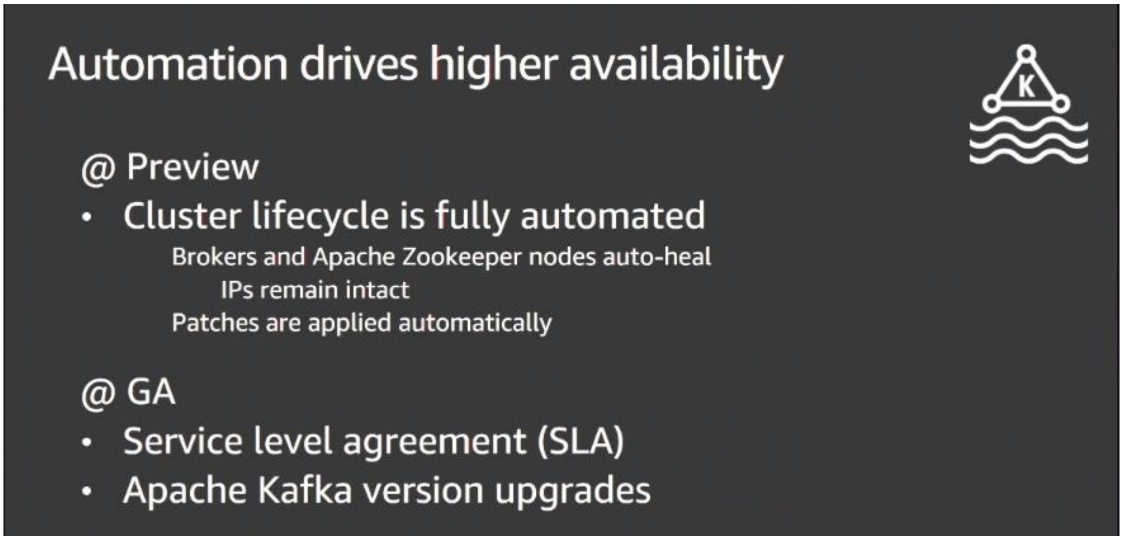
```
ec2-user@ip-10-0-0-148:~/kafka_2.11-1.1.1
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Fri Nov 30 16:31:36 2018 from 208.78.238.4

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-148 ~]$ cd kafka_2.11.1.1.1
-bash: cd: kafka_2.11.1.1.1: No such file or directory
[ec2-user@ip-10-0-0-148 ~]$ cd kafka_2.11-1.1.1
[ec2-user@ip-10-0-0-148 kafka_2.11-1.1.1]$ bin/kafka-console-consumer.sh --boots
trap-server 10.0.1.126:9092,10.0.0.193:9092,10.0.2.229:9092 --topic stream --fro
m-beginning
```



```
Command Prompt - aws  kafka describe-cluster  --cluster-arn arn:aws:kafka:us-east-1:113929328541:cluster/DamiansDemoCluster/791e12c9-82da-4bd7...
            "KafkaVersion": "1.1.1"
        },
        "CurrentVersion": "K2EUQ1WTGCTBG2",
        "EncryptionInfo": {
            "EncryptionAtRest": {
                "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:113929328541:key/5316f886-60e4-4a0f-95a0-0146a0f244db"
            }
        },
        "EnhancedMonitoring": "PER_TOPIC_PER_BROKER",
        "NumberOfBrokerNodes": 3,
        "State": "ACTIVE",
        "ZookeeperConnectString": "10.0.0.12:2181,10.0.2.147:2181,10.0.1.175:2181"
    }
}

C:\>aws kafka get-bootstrap-brokers --cluster-arn arn:aws:kafka:us-east-1:113929328541:cluster/DamiansDemoCluster/791e12
c9-82da-4bd7-9a6f-21f728d343ce-1
{
    "BootstrapBrokerString": "10.0.2.229:9092,10.0.0.193:9092,10.0.1.126:9092"
}

C:\>aws kafka get-bootstrap-brokers --cluster-arn arn:aws:kafka:us-east-1:113929328541:cluster/DamiansDemoCluster/791e12
c9-82da-4bd7-9a6f-21f728d343ce-1
{
    "BootstrapBrokerString": "10.0.2.229:9092,10.0.0.193:9092,10.0.1.126:9092"
}

C:\>aws kafka describe-cluster  --cluster-arn arn:aws:kafka:us-east-1:113929328541:cluster/DamiansDemoCluster/791e12c9-8
2da-4bd7-9a6f-21f728d343ce-1
```



```
Select Command Prompt
            "SecurityGroups": [
                "sg-096b6221d3bd2e7c1"
            ],
            "StorageInfo": {
                "EbsStorageInfo": {
                    "VolumeSize": 1000
                }
            }
        },
        "ClusterArn": "arn:aws:kafka:us-east-1:113929328541:cluster/DamiansDemoCluster/791e12c9-82da-4bd7-9a6f-21f728d34
3ce-1",
        "ClusterName": "DamiansDemoCluster",
        "CreationTime": "2018-11-27T05:32:57.79Z",
        "CurrentBrokerSoftwareInfo": {
            "KafkaVersion": "1.1.1"
        },
        "CurrentVersion": "K2EUQ1WTGCTBG2",
        "EncryptionInfo": {
            "EncryptionAtRest": {
                "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:113929328541:key/5316f886-60e4-4a0f-95a0-0146a0f244db"
            }
        },
        "EnhancedMonitoring": "PER_TOPIC_PER_BROKER",
        "NumberOfBrokerNodes": 3,
        "State": "ACTIVE",
        "ZookeeperConnectString": "10.0.0.12:2181,10.0.2.147:2181,10.0.1.175:2181"
    }
}

C:\>
```



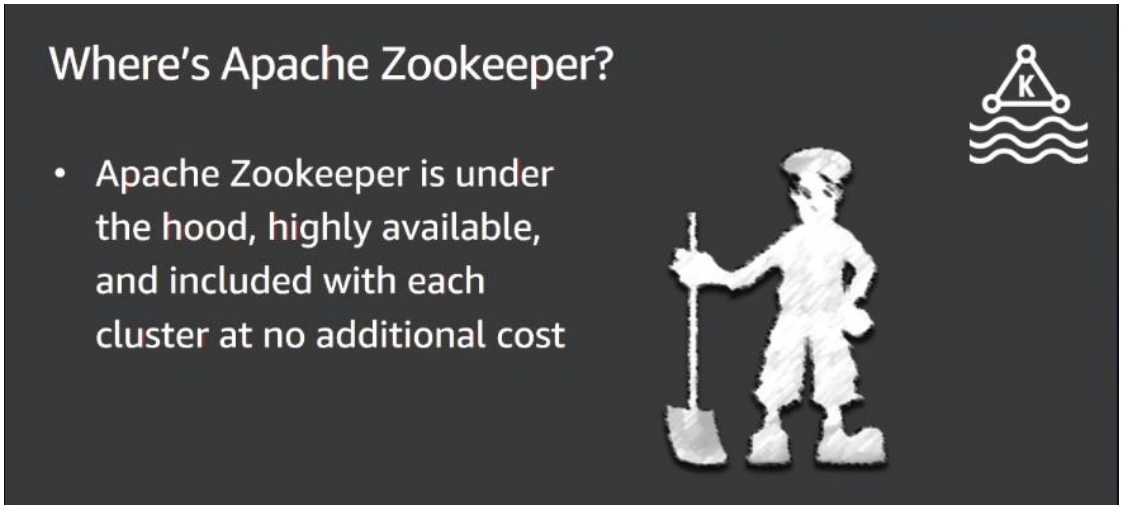
Automation drives higher availability

@ Preview

- Cluster lifecycle is fully automated
  - Brokers and Apache Zookeeper nodes auto-heal
    - IPs remain intact
  - Patches are applied automatically

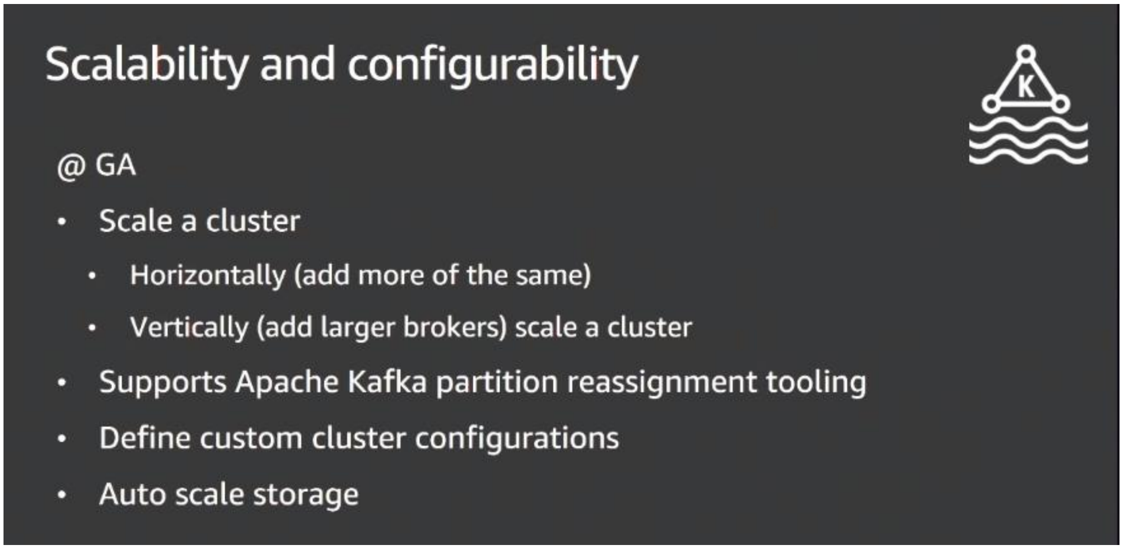
@ GA

- Service level agreement (SLA)
- Apache Kafka version upgrades



Where's Apache Zookeeper?

- Apache Zookeeper is under the hood, highly available, and included with each cluster at no additional cost



Scalability and configurability

@ GA

- Scale a cluster
  - Horizontally (add more of the same)
  - Vertically (add larger brokers) scale a cluster
- Supports Apache Kafka partition reassignment tooling
- Define custom cluster configurations
- Auto scale storage

Deeply integrated with AWS services

@ Preview

- Amazon Virtual Private Cloud (Amazon VPC) for network isolation
- AWS Key Management Service (AWS KMS) for at-rest encryption
- AWS Identity and Access Management (IAM) for control-plane API control
- Amazon CloudWatch for Apache Kafka broker, topic, and ZK metrics
- Amazon Elastic Compute Cloud (Amazon EC2) M5 instances as brokers
- Amazon EBS GP2 broker storage
- Offered in the US-East (N. Virginia) AWS Region

@ GA

- Tagging
- AWS CloudTrail
- AWS CloudFormation
- Offered worldwide

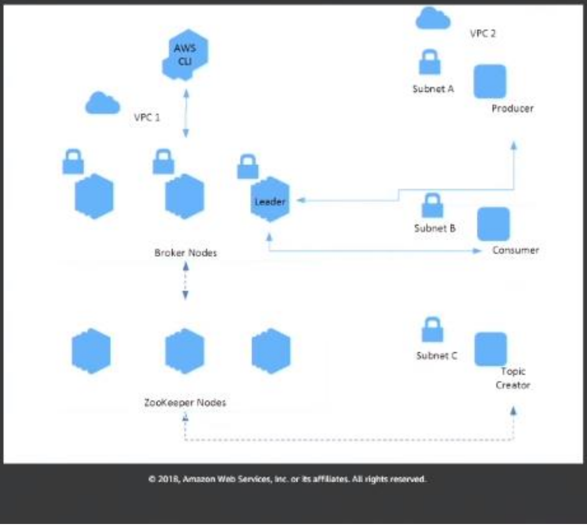
What Amazon MSK does for you

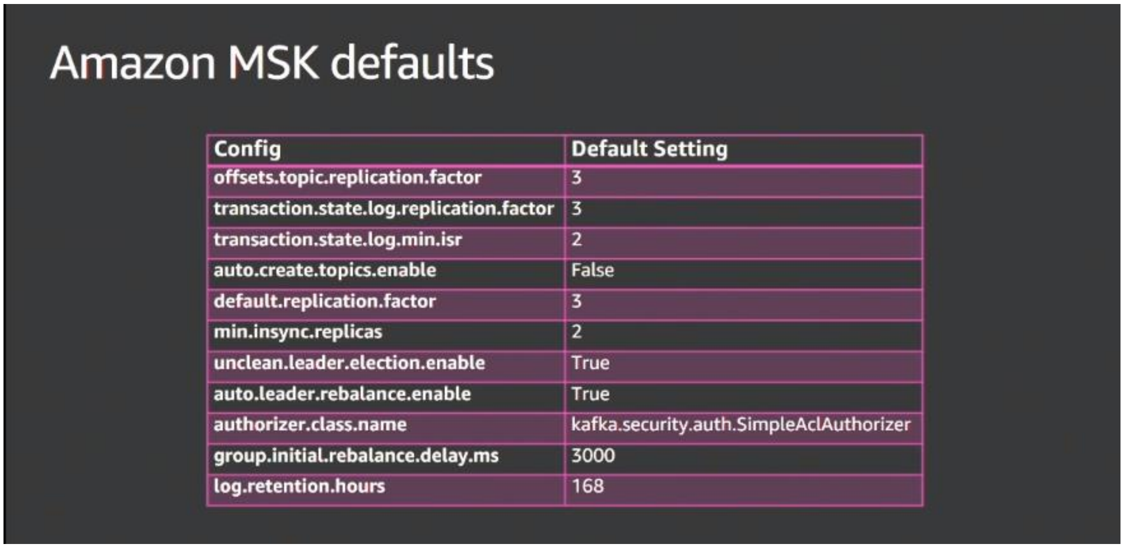
- Makes Apache Kafka more accessible to your organization

$$f(kafka_{usage}) = \sum_{n=1}^{\infty} Streaming\ Apps$$

- Drives best practices through design, defaults, and automation
- Allows developers to focus more on app development, less on infrastructure management
- Amazon MSK is committed to improving open-source Apache Kafka

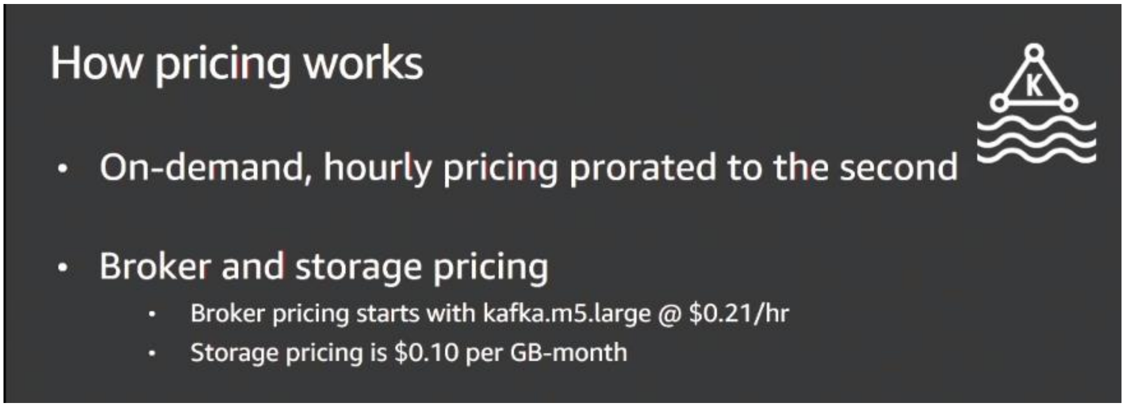
How it works





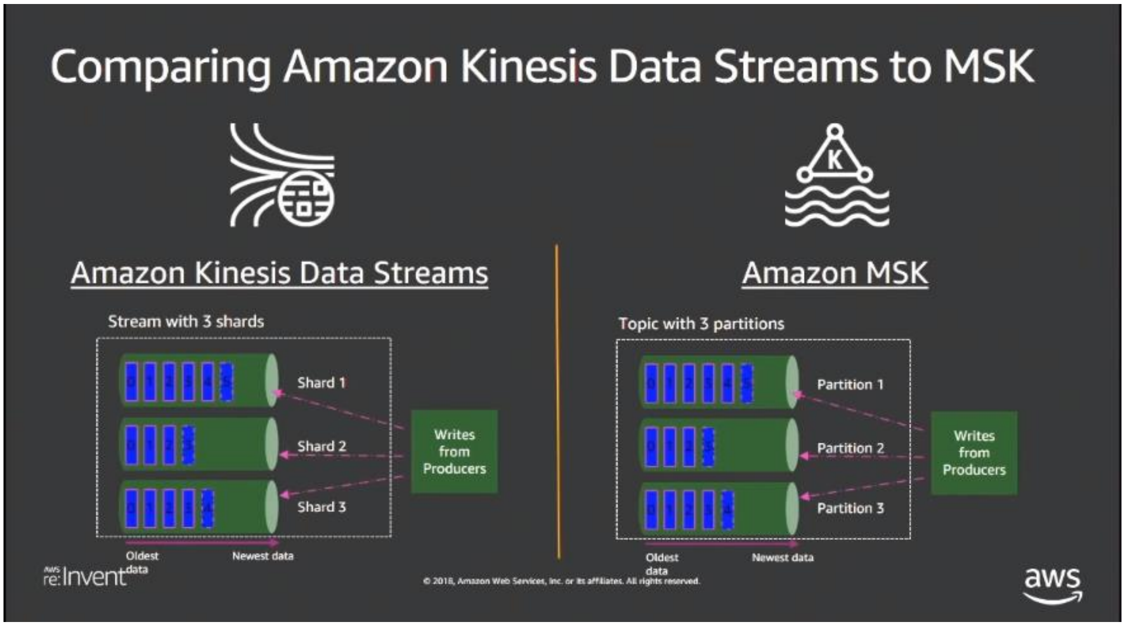
Amazon MSK defaults

Config	Default Setting
offsets.topic.replication.factor	3
transaction.state.log.replication.factor	3
transaction.state.log.min.isr	2
auto.create.topics.enable	False
default.replication.factor	3
min.insync.replicas	2
unclean.leader.election.enable	True
auto.leader.rebalance.enable	True
authorizer.class.name	kafka.security.auth.SimpleAclAuthorizer
group.initial.rebalance.delay.ms	3000
log.retention.hours	168



How pricing works

- On-demand, hourly pricing prorated to the second
- Broker and storage pricing
  - Broker pricing starts with kafka.m5.large @ $0.21/hr
  - Storage pricing is $0.10 per GB-month



Comparing Amazon Kinesis Data Streams to MSK

Amazon Kinesis Data Streams

Stream with 3 shards

Shard 1

Shard 2

Shard 3

Writes from Producers

Oldest data

Newest data

Amazon MSK

Topic with 3 partitions

Partition 1

Partition 2

Partition 3

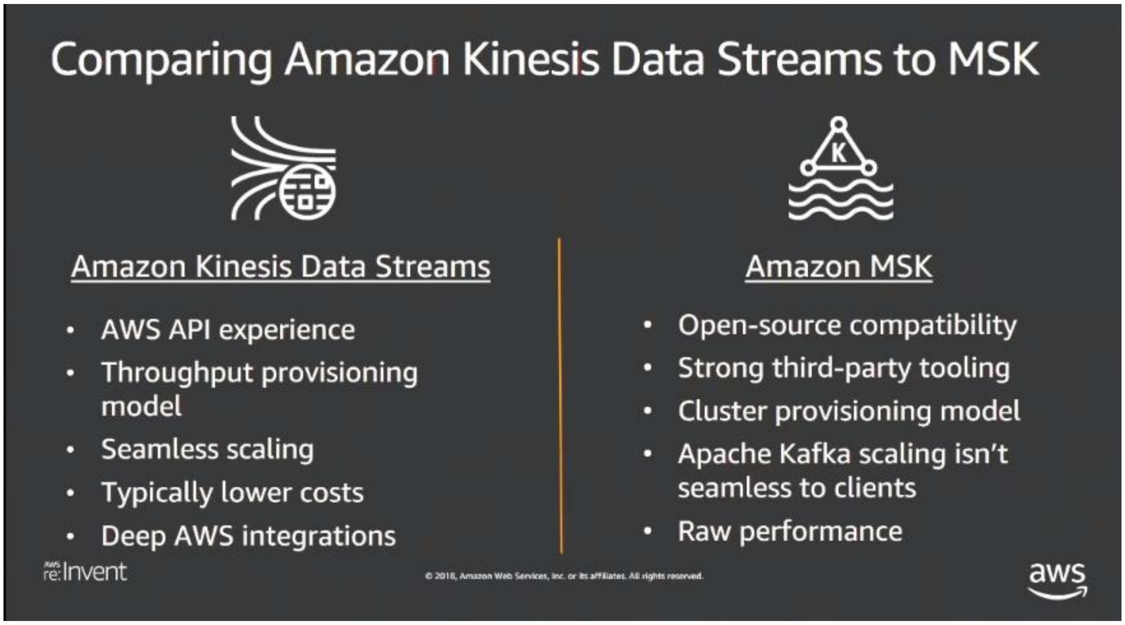
Writes from Producers

Oldest data

Newest data

re:Invent

aws



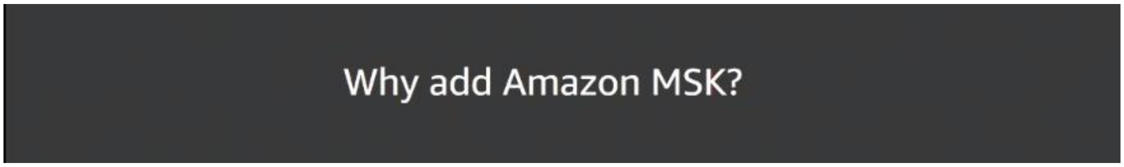
Comparing Amazon Kinesis Data Streams to MSK

Amazon Kinesis Data Streams

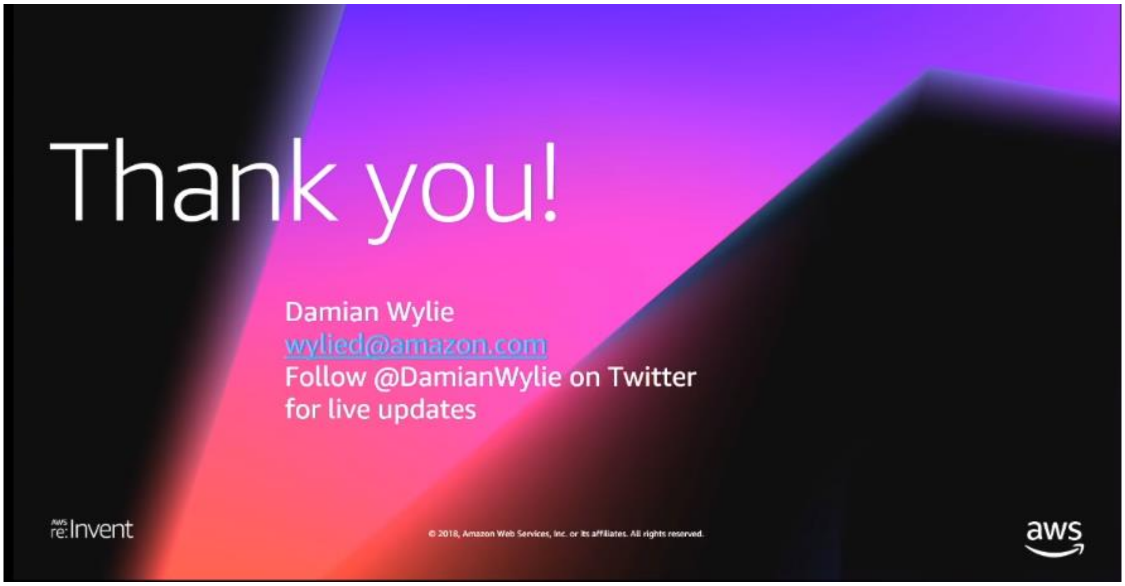
- AWS API experience
- Throughput provisioning model
- Seamless scaling
- Typically lower costs
- Deep AWS integrations

Amazon MSK

- Open-source compatibility
- Strong third-party tooling
- Cluster provisioning model
- Apache Kafka scaling isn't seamless to clients
- Raw performance



Why add Amazon MSK?



Thank you!

Damian Wylie
wylied@amazon.com
Follow @DamianWylie on Twitter
for live updates