



Amazon MSK Overview

**“Amazon MSK: A fully managed,
high available and secure service
for Apache Kafka”**

Amazon MSK is committed to improving open-source Apache Kafka



Amazon MSK key benefits



Makes Apache Kafka more secure, highly available, and accessible to your organization



Drives best practices through design, defaults, and automation

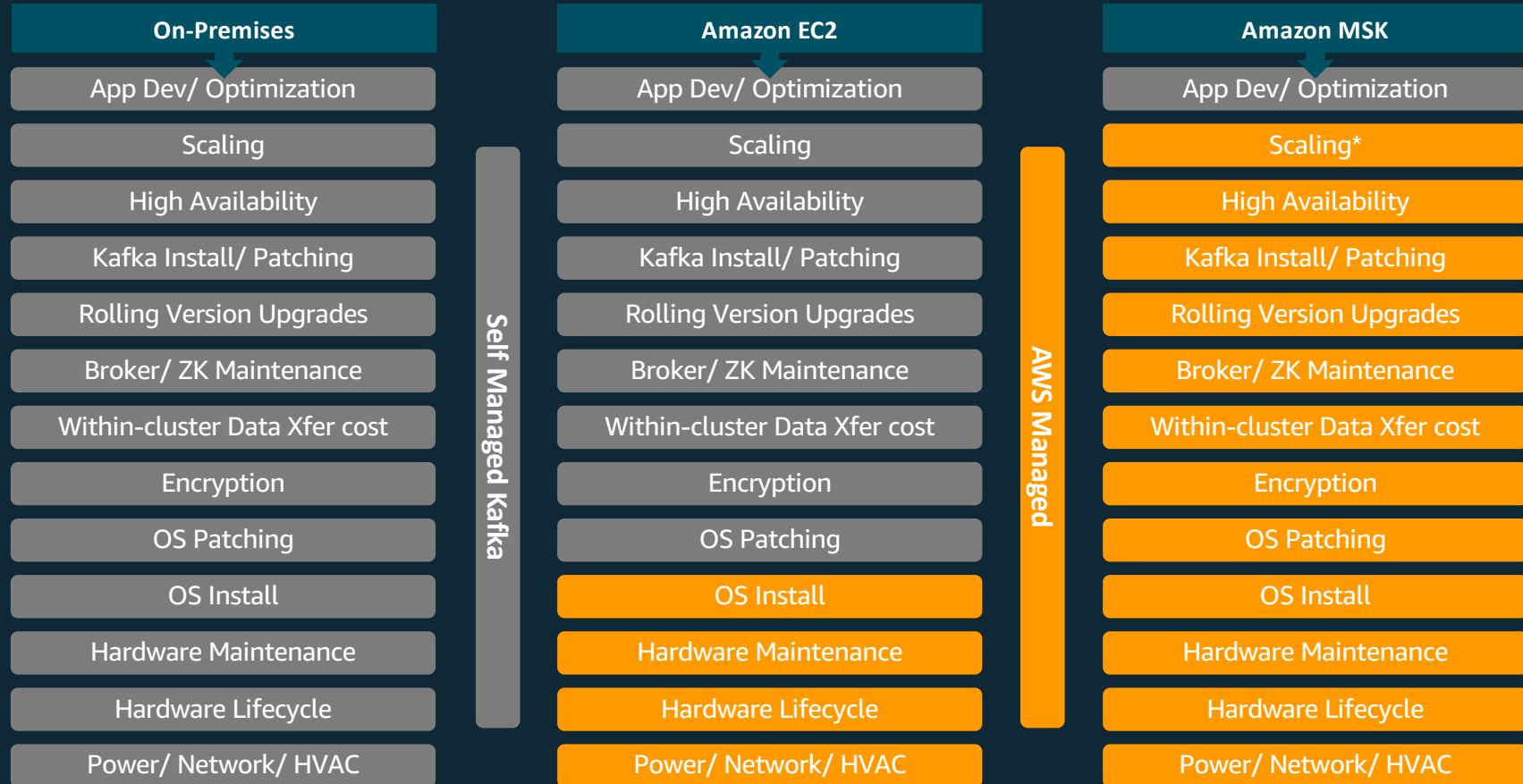


Allows developers to focus more on application development and less on infrastructure management



Eliminates right-sizing, scaling, and partition management with MSK Serverless

Fully Managed



More focus on creating Streaming Applications than managing infrastructure

* Cluster expansion and auto scaling storage



Key Features



Amazon MSK - Key features

Amazon MSK - Provisioned

Amazon MSK - Serverless

MSK Connect

Tiered Storage

Security

Scaling

Upgrades, Partition
rebalancing *(Serverless)*

MSK Connect - Private DNS

Monitoring

Integration with AWS Services

Real Time Analytics

Glue Schema Registry



Amazon MSK cluster types

Amazon MSK Provisioned

- Automate provisioning, configuring, and tuning
- Fully compatible with Opensource Apache Kafka
- Highly configurable
- Highly secure
- Lower cost

Amazon MSK Serverless

- Easily run Apache Kafka clusters without needing to right-size cluster or overprovisioning
- Instantly scale I/O without needing to worry about scaling up and down
- Automated partition reassignment and scaling
- Throughput based pricing



MSK Serverless - key features



On-demand streaming capacity



Fully compatible



Throughput based pricing



Same security as MSK



Auto partition placement



Same high availability

Amazon MSK Connect



Run fully managed Kafka Connect clusters with Amazon MSK

Eliminates the need to provision and maintain cluster infrastructure

Easily deploy, monitor and scale connectors that move data in and out of Apache Kafka and Amazon MSK

Connectors scale automatically in response to increases in usage and you pay only for the resources you use

Fully compatible with Kafka Connect that makes it easy to migrate workloads without code changes

Amazon MSK - Tiered Storage

- A low cost storage tier for Amazon MSK which scales virtually unlimited
- Helps to retain data longer in Apache Kafka topics
- Re-process old data in exact production order with your existing stream processing code
- Data between brokers and the tiered storage moves within the VPC and doesn't travel through the internet
- Faster partition rebalancing because data on secondary storage does not require replication across brokers
- Best serves applications that must retain data for longer than a day, or more than 1-2 TB per broker.

***The Amazon MSK cluster with tiered storage enabled must use Kafka version 2.8.2.tiered.*



Storage in Amazon MSK

1 EBS Volume per
Broker

Size from 1 GiB to
16384 GiB

Enable provisioned
storage throughput
up to 1000 MiB/s per
broker



Scaling Storage in Amazon MSK Provisioned

Storage



MSK data storage is backed by Elastic Block Storage (EBS) volumes. Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud.


EBS storage volume per broker

You can't decrease the storage after the cluster is created.

GiB

Minimum: 1 GiB, maximum: 16384 GiB

 After creating your cluster, you can configure auto scaling to auto-expand broker EBS storage at a specified utilization target. [Learn more](#) 



- Scale storage in 10 GiB increments
- Cluster is available during update
- Configure storage auto-scaling to automatically expand storage
- Deployment follows Apache Kafka best practices for replication

How we handle encryption

Encryption at rest



AWS KMS

Amazon MSK uses Amazon server-side encryption and AWS KMS service and customer managed keys to encrypt storage volumes

Encryption in transit



TLS

TLS encryption for data-in-transit enabled by default

Scalability

Scale out clusters



Scale out existing clusters by adding additional brokers without affecting cluster availability

Auto scale storage



Scale up broker attached EBS storage without affecting broker availability

Vertical scaling



Change the size or family of your brokers without reassigning Apache Kafka partitions

Automated scaling



MSK Serverless automatically scales your cluster's streaming capacity as your application throughput grows



Scaling Amazon MSK

Horizontal Scaling

Add additional Kafka brokers

Must be a multiple of used AZs

Only *scale out* operation supported

Requires reassigning of partitions

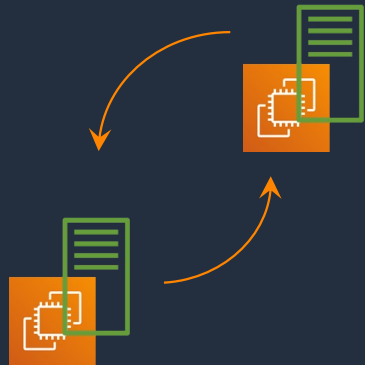
Vertical Scaling

Change the size or family of Kafka brokers

Scale up and down operations

No cluster I/O interruption

Rolling Version Upgrades



Upgrade your Kafka versions
without any downtime for
your clusters*


* Deployment follows Apache Kafka best practices for replication




Upgrading Apache Kafka version

Upgrade the Apache Kafka version for demo-cluster-1

Apache Kafka version

Apache Kafka version
The Apache Kafka version that you want on all brokers. [Learn more](#) 

3.3.1 

Current version: 2.8.1. You can't downgrade the Apache Kafka version.

Update cluster configuration

- Upgrade to subsequent available versions (currently supported version 3.3.1)
- Update runs in background
- Cluster is available during upgrade
- Deployment follows Apache Kafka best practices for replication



Sizing Amazon MSK



Up to 40% lower cost compared
to self managed Kafka

The calculator gets you in the ballpark for most workloads

Your latency, encryption and retention needs will help to define
your MSK cluster sizing

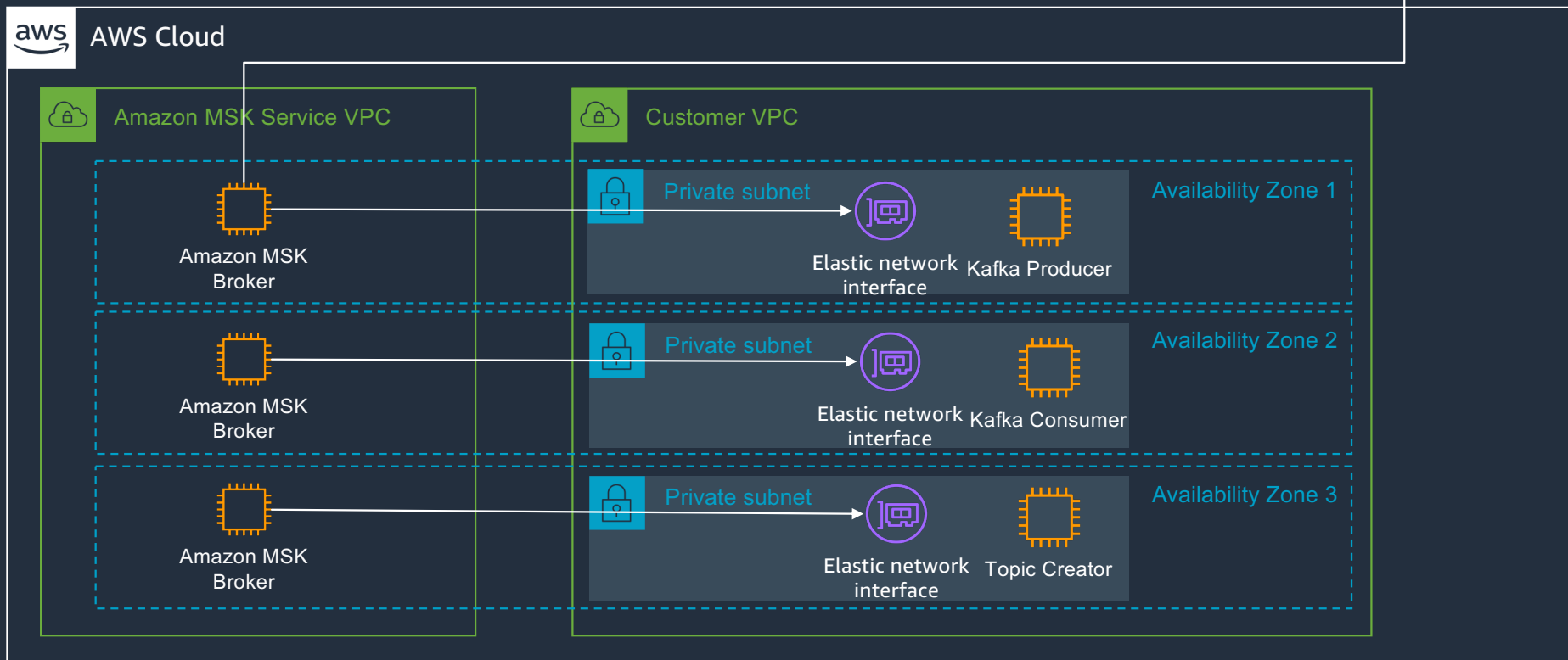
Testing is key after the initial sizing

You can find a link to the calculator in the best practices section of
the Amazon MSK user documentation

<https://docs.aws.amazon.com/msk/latest/developerguide/bestpractices.html>

Amazon MSK Connectivity

Public access



Private access by default

Options to turn on Public access



MSK Monitoring



CloudWatch Metrics

You can set three levels of monitoring with in CloudWatch for MSK, Default (at no cost to you), PER_BROKER and PER_TOPIC_PER_BROKER level.



Open Monitoring with Prometheus

You can enable open monitoring with Prometheus and expand your monitoring capability to third party compatible tools such as Datadog, Lenses, New Relic and Sumo Logic



Broker Logs to CW logs, S3 and AES

Continuously stream Apache Kafka broker logs to Amazon CloudWatch Logs, Amazon S3, or Amazon Opensearch Service via Amazon Kinesis Data Firehose

Deep AWS service integrations



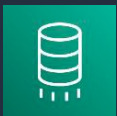
Amazon IAM access control for cluster authentication and API authorization



Amazon Glue Schema Registry for schema evolution



Amazon Kinesis Data Analytics for Flink for complex stream processing



Amazon Data Migration Service for change data capture use cases



AWS Lambda with MSK as an event source



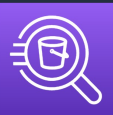
AWS Certificate Manager for Private CAs used for client TLS authentication



Deep AWS service integrations (Contd...)



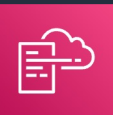
Amazon KMS for storage volume encryption



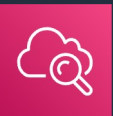
Athena to query streaming data in MSK and self-managed Apache Kafka



Real time stream ingestion from Amazon MSK to Redshift



CloudFormation and Terraform for describing and provisioning Amazon MSK clusters using code



Amazon CloudWatch for metrics

Amazon MSK Compatibility

✓ Open source Apache Kafka

- Kafka Connect
- MirrorMaker
- Kafka Streams

✓ Apache Kafka tooling and frameworks

- AWS Glue Schema Registry or 3rd party schema registries
- REST proxies
- Additional 3rd party tools: Burrow, Kafdrop, CMAK, etc.

⊘ Tools that load .jar files on brokers

- Confluent Control Center
- Confluent Auto Data Balancer

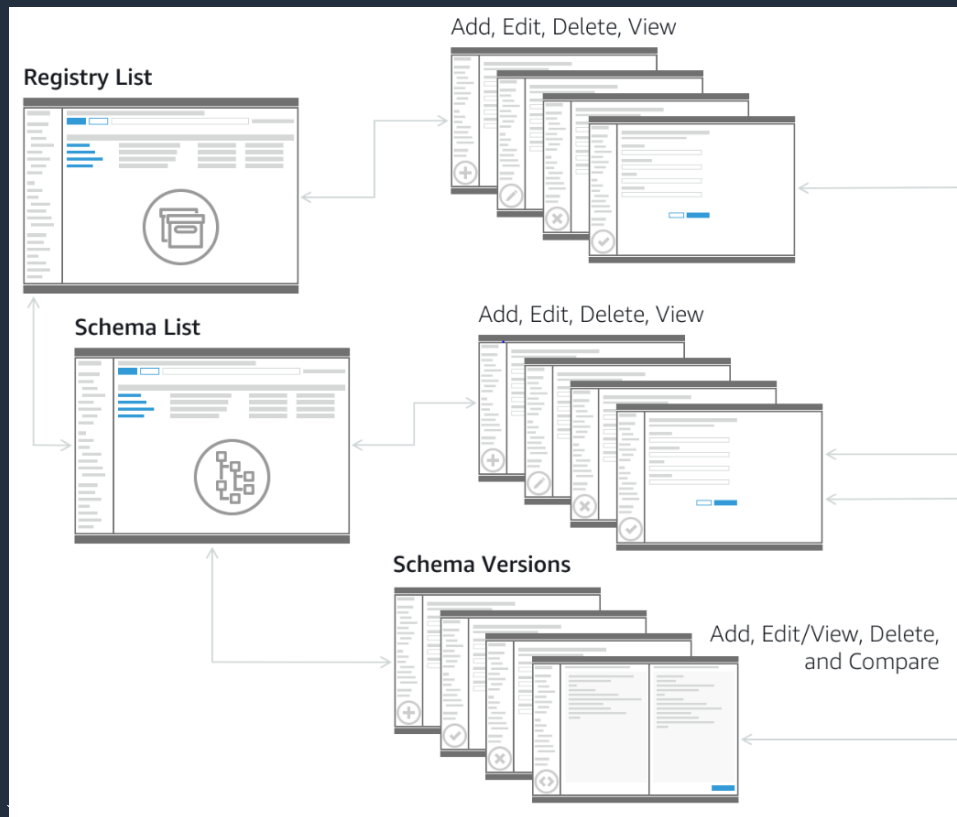


• Uber uReplicator

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

AWS Glue Schema Registry

Centrally discover, control, and evolve your data schemas



Improve data quality for your data streaming applications

Enforce schemas and schema evolution to prevent downstream application failures

Easily integrates with Amazon Managed Streaming for Apache Kafka (MSK), Amazon Kinesis Data Streams, and Amazon Kinesis Data Analytics for Apache Flink for convenient setup

Use provided open source libraries to compress data and save on storage and data transfer costs

Amazon MSK compliance



HIPAA



PCI



ISO



SOC



GDPR
ready

Best Practices



Best Practices

Right-size Your Cluster

- Don't just lift and shift. Use Amazon MSK Best practices as a guideline.
- To determine the right number of brokers for your MSK cluster and understand costs, use the [MSK Sizing and Pricing](#) calculator.

Set the appropriate configuration parameters

- Use latency/throughput, durability and availability requirements to guide tuning parameters
- Test thoroughly

Confine Zookeeper usage to Amazon MSK

- Do not add non-MSK brokers or application configuration related metadata to MSK provisioned Zookeeper

Adjust data retention parameters

- Fine tune your retention time period or the retention log size at both the cluster and topic levels as applicable
- Use tiered storage feature for longer data retention period.



Best Practices (contd...)

Disk Usage

- Monitor the KafkaDataLogsDiskUsed metric in CloudWatch
- Set an alarm at 85% utilization
- When the alarm is triggered, increase broker storage
- Ensure the retention period for topics is ideal

Partitioning

- More partitions generally means higher throughput
- Total # of partitions per broker < 4000 (see [Recommended Partitions for broker types](#))
- Rule of thumb: # of partitions per topic = Max ingest rate per topic in MB per sec / 2
- OR, if you know how much throughput each consumer can support per partition use that in the denominator.
- Keep in mind you need to ensure you have enough capability for catch up



Thank you!