AWS App Mesh

Shubha Rao
Principal Product Manager
Amazon Web Services
Customers have lots of pieces to operate.

What do developers really need to build their applications?
App Mesh Features

Consistency across teams

Failure visibility and isolation

Fine-grained deployment controls

Authentication and Authorization controls
Why did we built AWS App Mesh?

Security, reliability, availability, and scale
Managed by AWS

Application focus
Declarative model for application communication

Choice: picking the best tool for the job
Works with multiple compute options

AWS App Mesh is our first step in building an application-aware network
AWS App Mesh manages your proxy configuration

Users → envoy

Service 1 → Service 2

Service 4 → Service 5

Service 3

AWS App Mesh
High-Level App Mesh Control Plane Architecture

AWS App Mesh

Customer Managed

AWS Managed
App Mesh Configuration

Mesh
Application

Virtual Gateway
Ingress rules

Virtual Service
Logical name/Service discovery

Virtual Router
Routing match conditions

Virtual Node
Group of service endpoints

Diagram:
- Virtual Gateway
- Virtual Service
- Virtual Router
- Virtual Node
App Mesh requires explicitly modeled dependencies
Minimizes config size, mutation and blast radius
Performs fragment parsing to customize config
App Mesh implements Envoy xDS
App Mesh Features and Roadmap

- Consistency across teams
- Failure visibility and isolation
- Fine-grained deployment controls
- Authentication and Authorization controls
## Where we are and near term roadmap

<table>
<thead>
<tr>
<th>Observability</th>
<th>Traffic Management</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container Insights Mesh Dashboard</td>
<td>HTTP/2, gRPC</td>
<td>Encryption – ACM</td>
</tr>
<tr>
<td>--</td>
<td>Ingress Gateway</td>
<td>Encryption – Bring your cert</td>
</tr>
<tr>
<td>Access logs</td>
<td>--</td>
<td>mTLS - Bring your cert</td>
</tr>
<tr>
<td>Metrics</td>
<td>Routing based on explicit dependency mapping</td>
<td>mTLS - ACM</td>
</tr>
<tr>
<td>Tracing</td>
<td>TCP, HTTP1.1</td>
<td>Bring your AuthZ</td>
</tr>
<tr>
<td>AWS solutions and Partner integrations</td>
<td>Path, Header based routing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Weighted Routing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Route Priorities</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Retries</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Health Check &amp; Load</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Balancing at Envoy</td>
<td></td>
</tr>
<tr>
<td>Must-Haves</td>
<td>Usability</td>
<td>Integrations</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>Regions</td>
<td>ECS Console</td>
<td>Cross Account Support using AWS RAM</td>
</tr>
<tr>
<td>CloudFormation</td>
<td>K8S Controllers</td>
<td>On-Prem / Hybrid with AWS Outposts</td>
</tr>
<tr>
<td>Scaling</td>
<td>Helm Charts for K8S controllers</td>
<td>Metadata based routing with AWS Cloud Map</td>
</tr>
<tr>
<td>PrivateLink</td>
<td>Flagger Integration</td>
<td>Cross Cluster Support using AWS Cloud Map</td>
</tr>
<tr>
<td>Backwards compatible API</td>
<td>HIPAA Compliance</td>
<td></td>
</tr>
<tr>
<td>Public Previews</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Envoy versions support</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Where we are and near term roadmap
Community Participation

Upstream Envoy support – SigV4 AuthN for Envoys
Envoy configuration sequencing changes
ACM integration with SPIFFE/SPIRE
X-Ray integration
Open Sourcing App Mesh control plane – Envoy Management Service
Edmunds: Improving modern software development and microservices with App Mesh

Edmunds is a leading car information and shopping platform, helping millions of visitors each month find their perfect car.

Challenge: There are a number of challenges with Microservices, including control over service-to-service communication, visibility into service-to-service communication and ensure trust by automating security and compliance on small Dev-Ops teams.

Solution: "App Mesh provide us a consistent communications management, complete visibility, failure isolation, protection, and fine-grained deployment controls." - Nitin Mahajan, Executive Director, service engineering, Edmunds

https://youtu.be/1UDRGlmbiZA
Use Cases

1. Migrate from EC2 or self-managed Kubernetes to EKS
Use Cases

2. Span a mesh across compute mode, accounts, clusters and VPCs

Shared Mesh: myapp

AWS Cloud Map Namespace

- EKS Cluster – EC2
- VPC
- Account 1

- EKS Cluster
- VPC
- Account 2

Assumes flat network – one of shared VPC via RAM, VPC Peering, PrivateLink, Transit GW

Use AWS Resource Access Manager to share the mesh resource
Use Cases

3. Span a mesh across hybrid deployments
Future Directions - Brainstorming (not on roadmap)

1. Lambda Support
Future Directions - Brainstorming (not on roadmap)

2. Connection management
Future Directions - Brainstorming (not on roadmap)

3. More Managed (Think Fargate)
Give us your input

surveymonkey.com/r/appmesh
Roadmap: https://github.com/aws/aws-app-mesh-roadmap
Engage with us!

• Take the App Mesh Workshop
• Meet our team at the booth
• Check example apps and config on Github: https://github.com/aws/aws-app-mesh-examples/tree/master/walkthroughs
• Read our latest post on Containers blog: https://aws.amazon.com/blogs/containers/cross-amazon-eks-cluster-app-mesh-using-aws-cloud-map/
• Tweet with #appmesh
Engage with us

Meet us at our booth

aws-app-mesh-roadmap

#appmesh or _shubha
Vision

“Our goal is that if you are running applications on AWS, you should not have to worry about managing networking infrastructure. It should be handled by our application-aware network … App Mesh, as it exists today, is the first step in this journey.”

~Werner Vogels

Give us your input: surveymonkey.com/r/appmesh