



Introduction ECS, EKS, Kubernetes Native and re:Invent Recap

Eduardo Miranda,
Cloud Engineer, Globant
19 December, 2018

Gonzalo Vásquez
R&D Director, Waypoint

What is the best of containers?

Everything about containers is great!



They're light



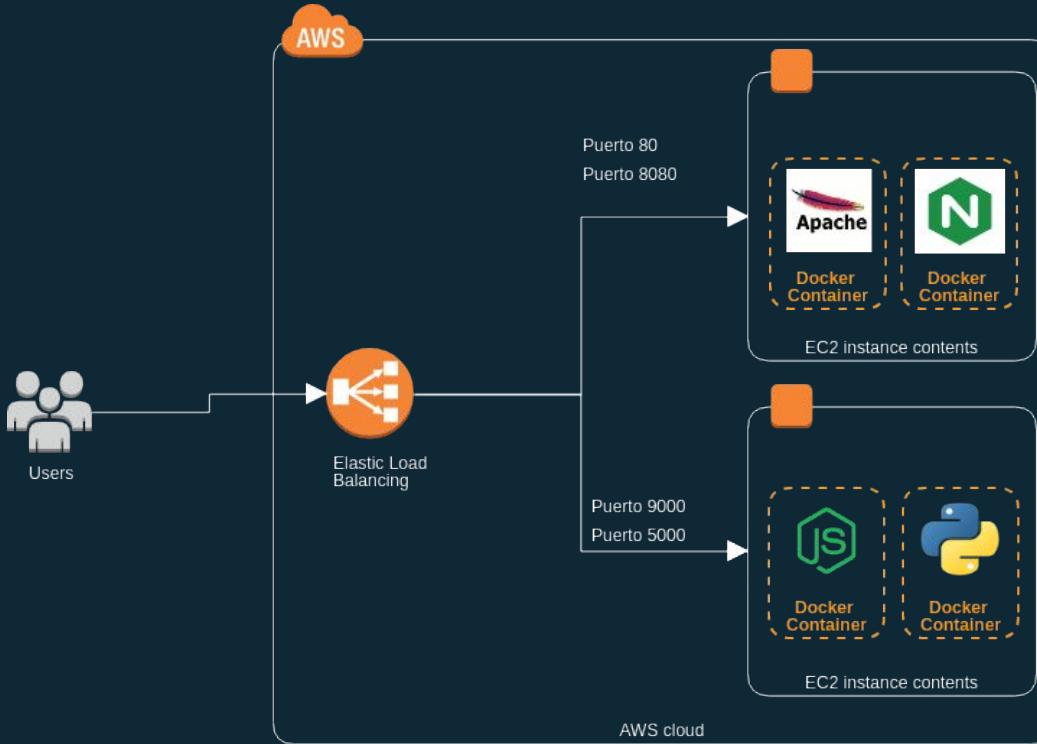
Adaptables



Portables

Containers and microservices go hand in hand

So, if I have a container running in my instance ¿How could it scale?



... and ¿What alternatives do we have for the containers administration in AWS?

How can I deploy my containers without affecting the service?

- Without downtime

How can I maintain my containers always available?

- Scheduled, Reset

How can the containers communicated between them?

- Linking service, discovery by network

How can I configure my containers in execution time?

- Access via console and terminal

How can I optimize the use and amount of my containers?

- Adjustment of resource association through continuous monitoring, escalation of containers on demand



kubernetes

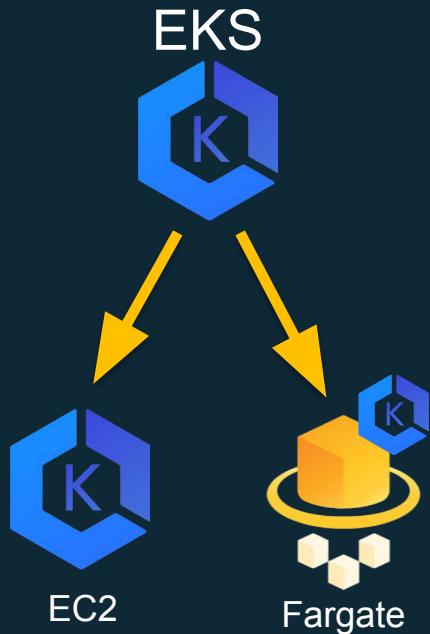
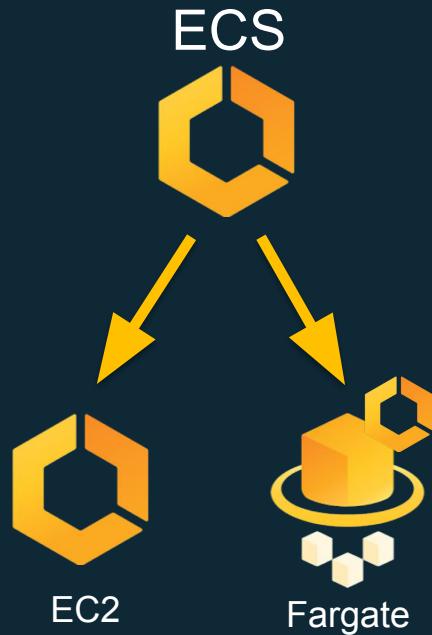


AWS ECS



AWS EKS

We can choose:



Kubernetes Native



Amazon EC2 Container Service (ECS)



Amazon EC2 Container Service (ECS)

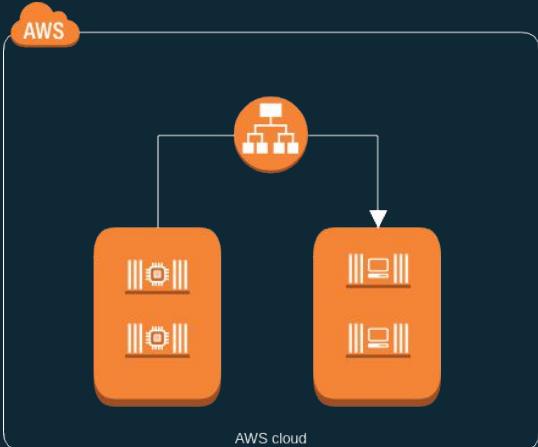


Instances: each one registers to the cluster and runs its task here

Services: this layer manages and assigns tasks

Task: container configures and wraps the processes running within the instance

¿Cómo trabaja ECS?



Load Balancer: (ALB o ELB Classic)
distribute traffic to clusters

Cluster: is made of one or more EC2
instances

Each cluster instance runs one or more
services

What advantages does ALB have over ELB?

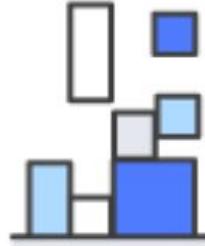
What advantages does ALB have over ELB?

You can define basic rules for the routing of traffic: When you say send traffic to different services based on endpoint, don't you think it's great?

Bonus: ALB allows you to assign the ports dynamically for the use of multi services



Some features...



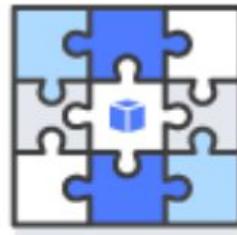
Amazon ECS Task Placement



IAM Roles for Tasks



Flexible scaling for performance



Amazon ECS Event Stream for Cloudwatch Logs

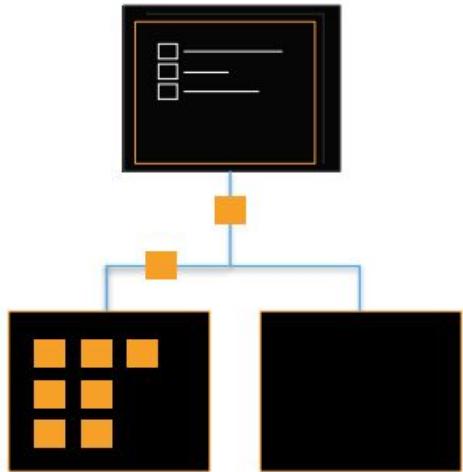


Fast, hassle-free deployments

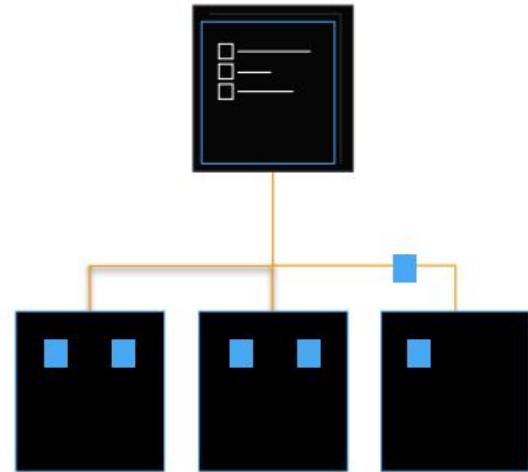
Amazon ECS Task Placement

- A task placement strategy is an algorithm to select the instances and the placement of the tasks o to finish them.
- A task placement constraint is a rule that is taken into consideration during the placement of the task.
- A task placement strategy and a task placement constraint can be used together.

Multiple Strategies are supported

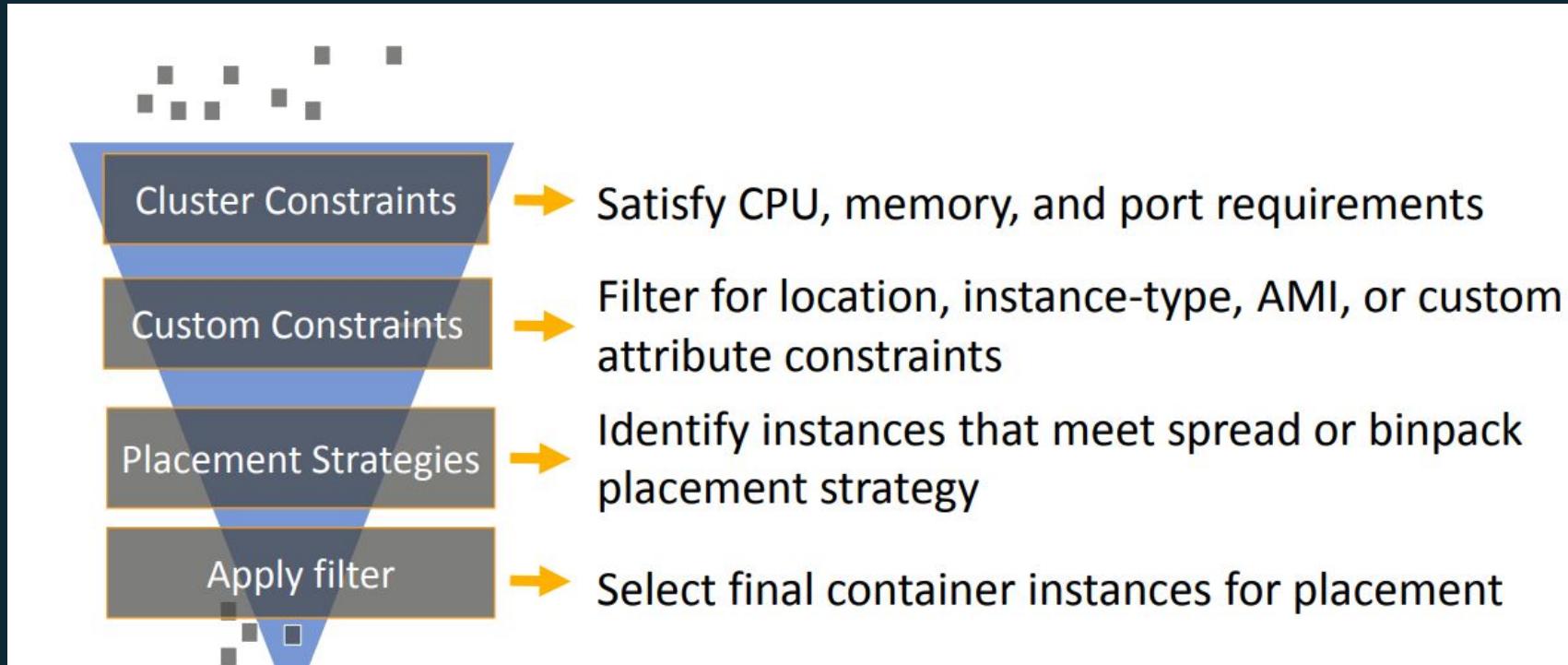


Binpacking



Spread

How it works...



Amazon ECS Event Stream for Cloudwatch Logs

- Receive in real time the updates about the current state of the containers
- You can program actions based on the status of the cluster generating events directly in Lambda.



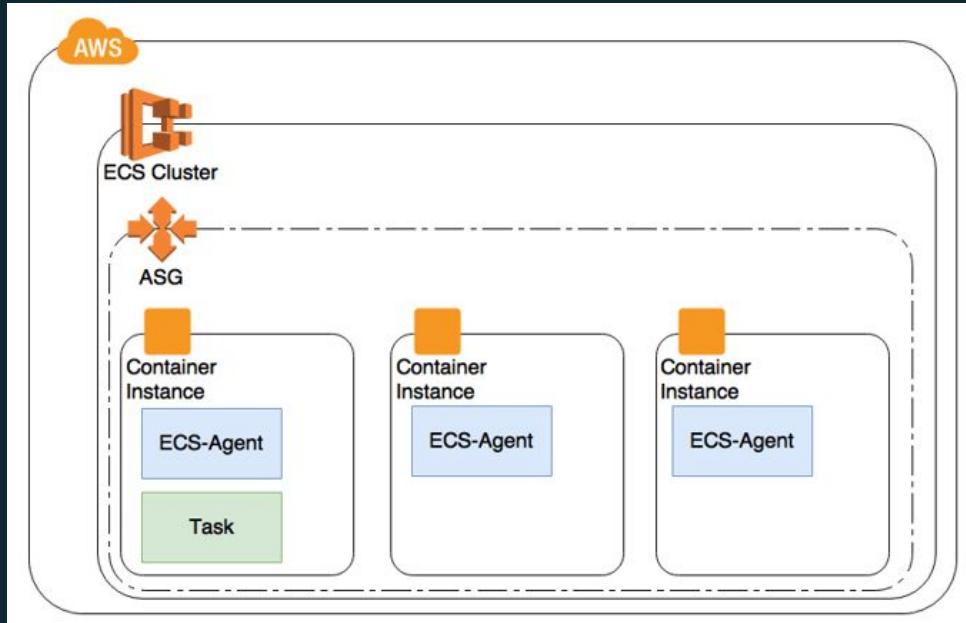
IAM Roles for ECS Tasks

- The roles are specified by containers in the tasks.
- Containers can only access the role assigned to the task.
- **Authorization:** containers can not access other credentials defined for other tasks.
- **Auditability:** Cloudtrail does the work for us.



Scaling flexibility

- The escalation of the services is based on the alarms detected by cloudwatch
- EC2 Autoscaling Group help us to scale our cluster without major inconvenience



Kubernetes on AWS



Statistics...



57%

of the workloads in
Kubernetes are executed
in AWS.

— Cloud Native Computing Foundation

```
$ kubectl get tips –n meetup
```

GET api/v1/namespaces/meetup/tips/{1}

Never build a cluster of kubernetes the hard way!

Options to configure a cluster of Kubernetes in AWS.

Community

- **Kops** – kubernetes-aws.io
- **Kubeadm** – Kit de Herramienta para arrancar los clusters
- **Kubespray** – set de herramientas para deployar K8s clusters.

Enterprise

- Elastic Container Service for Kubernetes (**EKS**)

Other options

- CloudFormation, Terraform, Ansible, Puppet

Kops

You can create it with Kops, but the management is your responsibility.



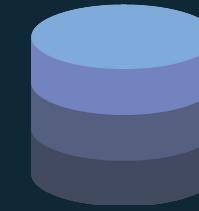
Your containers



Slave nodes



Master nodes

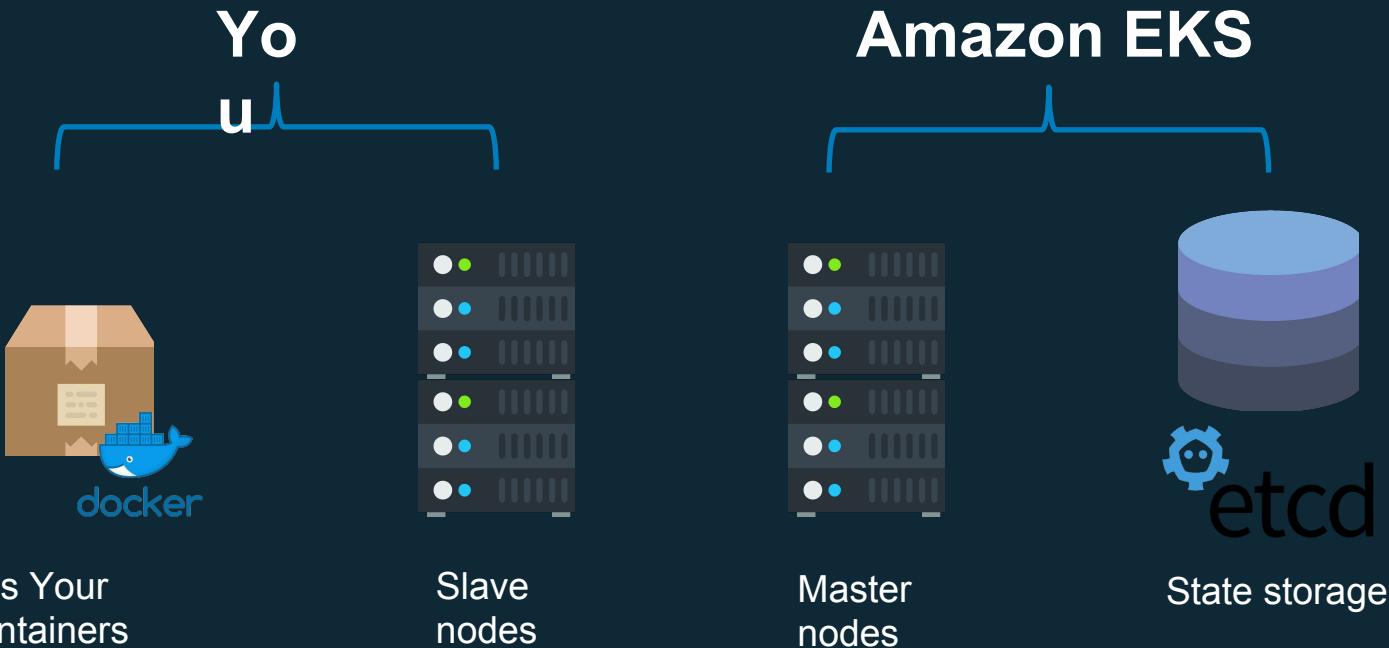


etcd

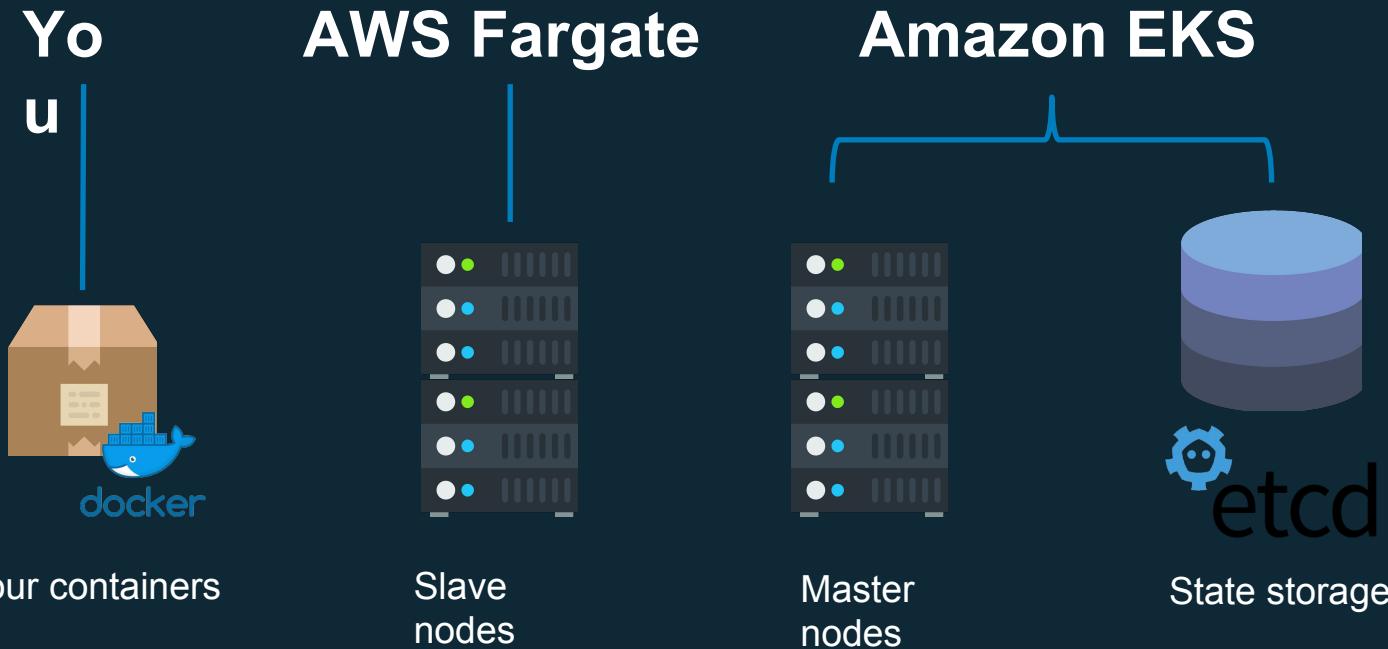
State storage



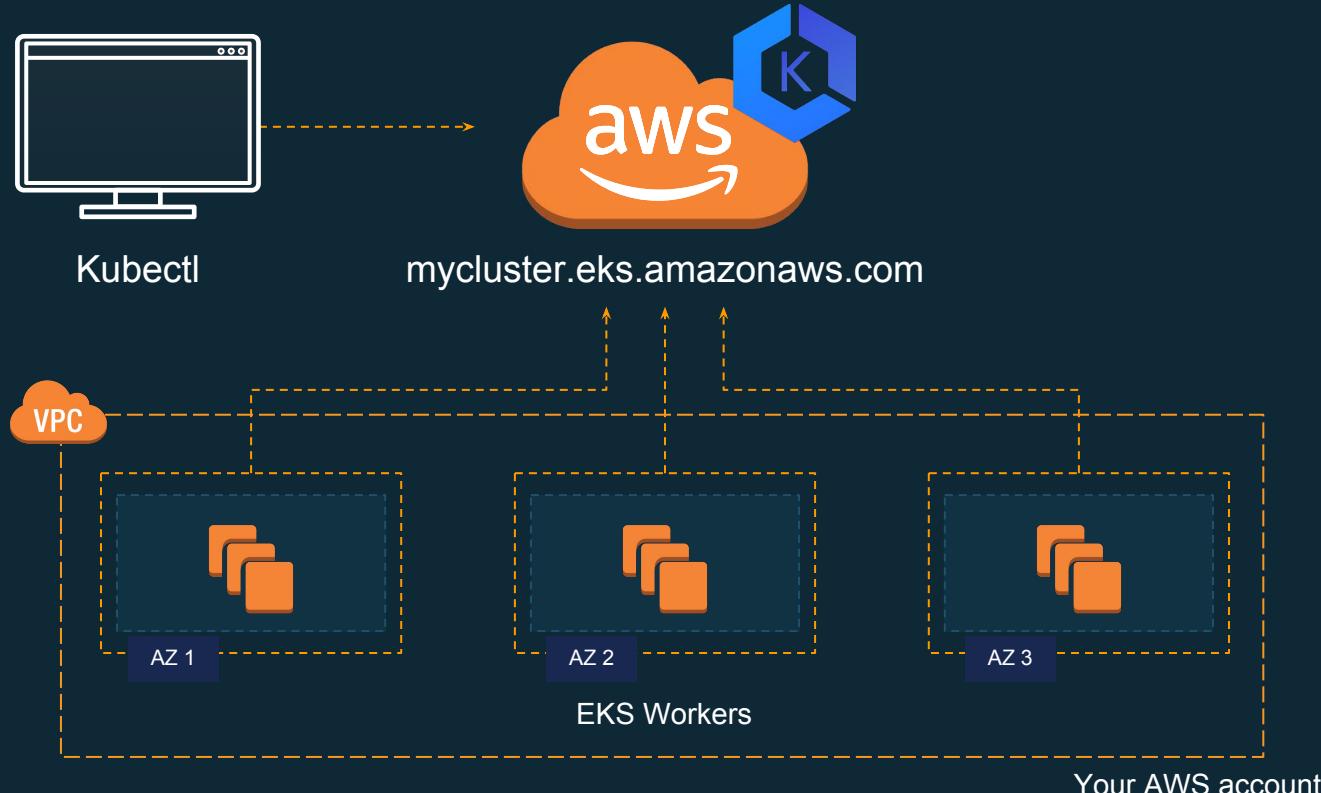
Responsibility based in EKS with EC2



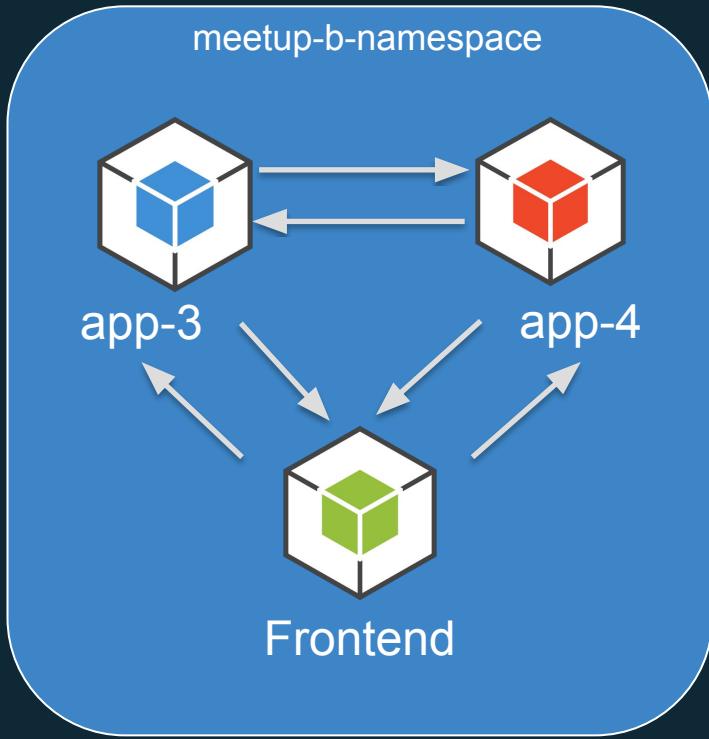
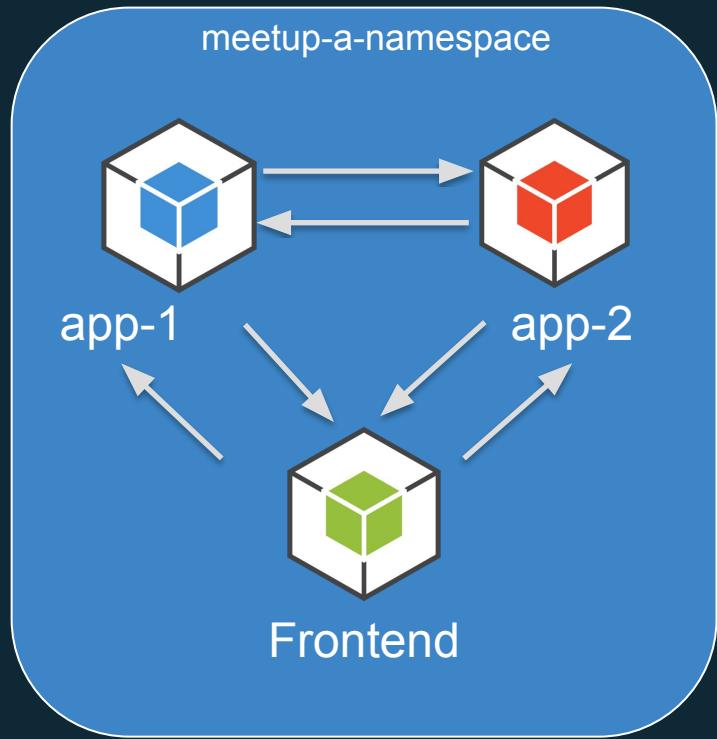
Total management of EKS



Amazon EKS



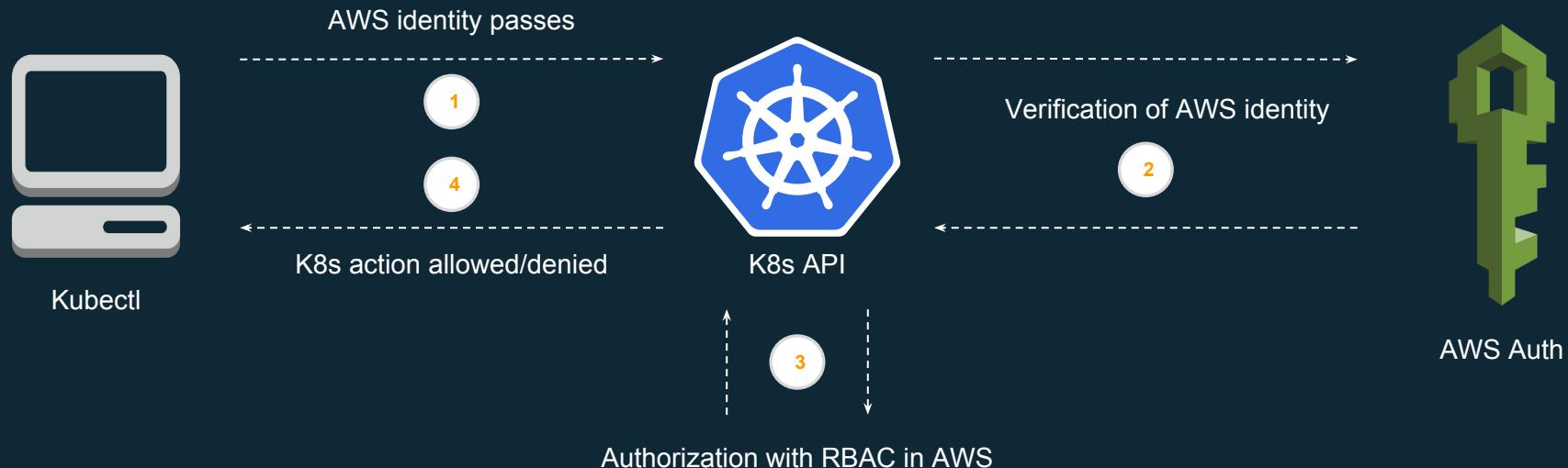
kubectl create namespace meetup



GET api/v1/namespaces/meetup/tips/{2}

Ensure RBAC is enabled in your Cluster

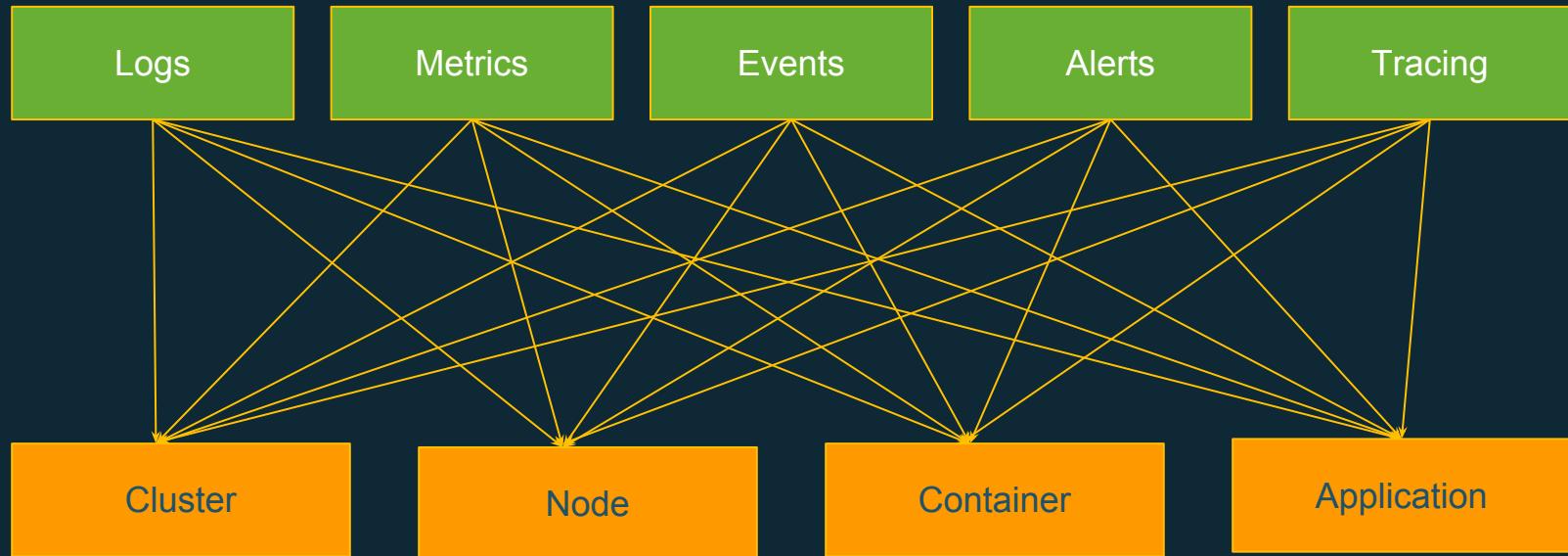
EKS: IAM Authentication + Kubectl



GET api/v1/namespaces/meetup/tips/{3}

Log and Monitoring of all resources!

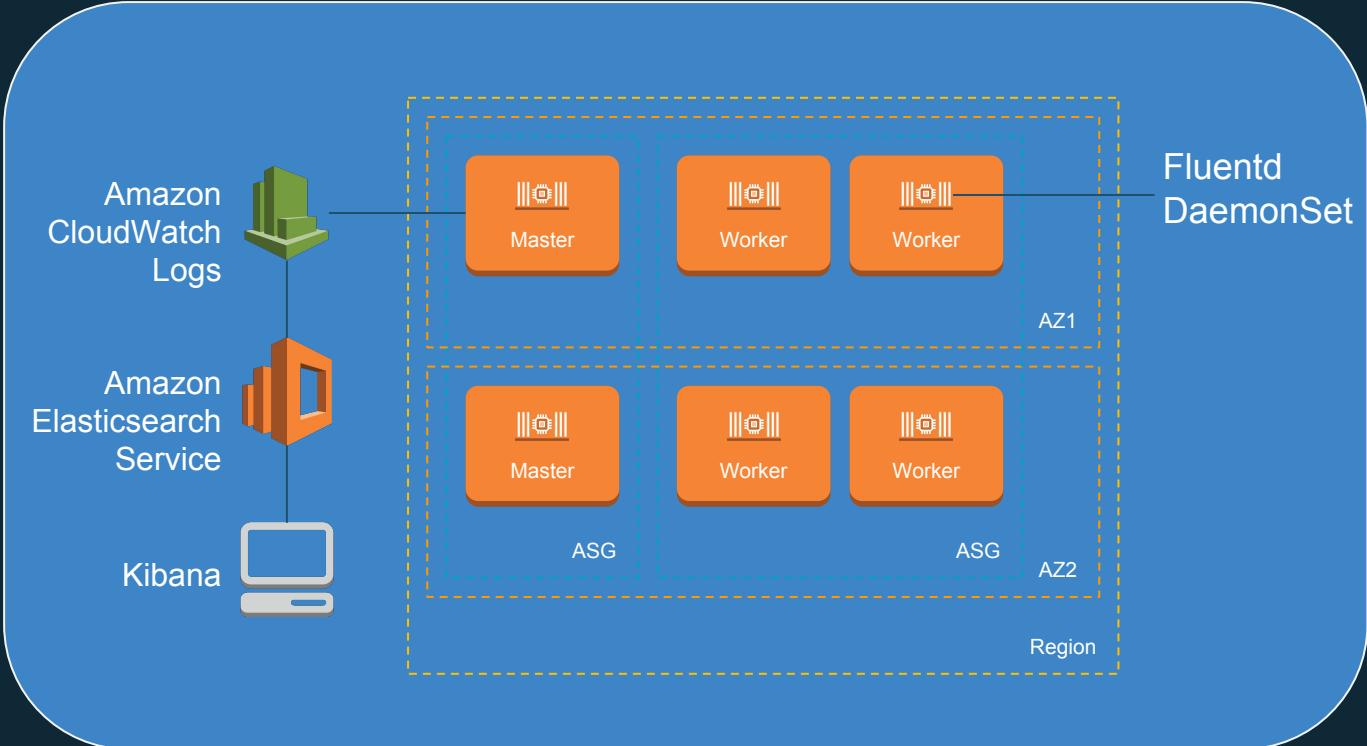
Visibility of your Kubernetes Cluster



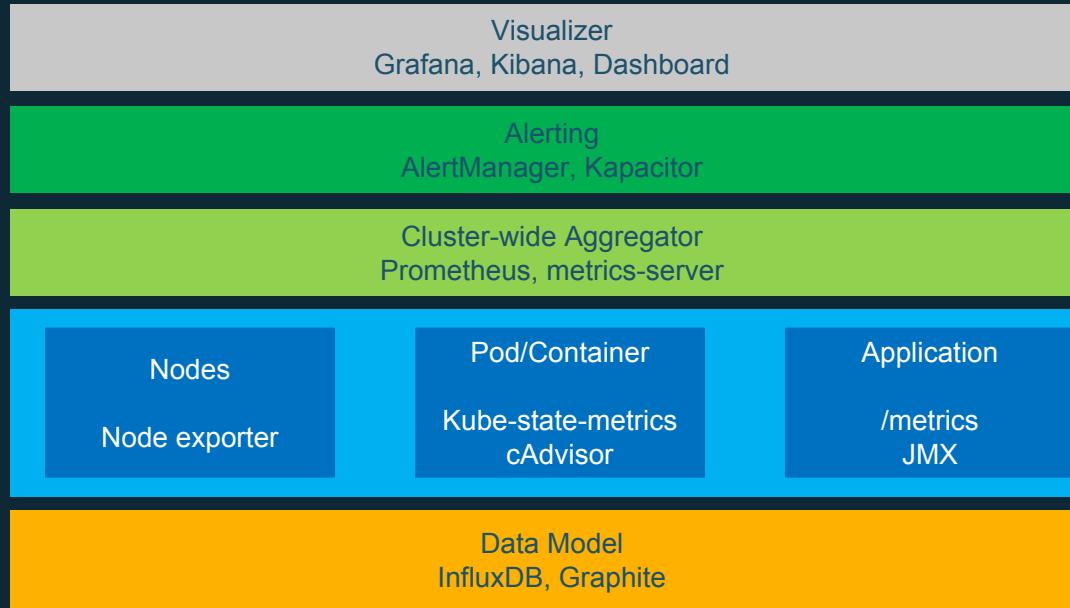
Logs

Kubectl logs

Elasticsearch (index),
Fluentd (store), and
Kibana (visualize)



Metrics

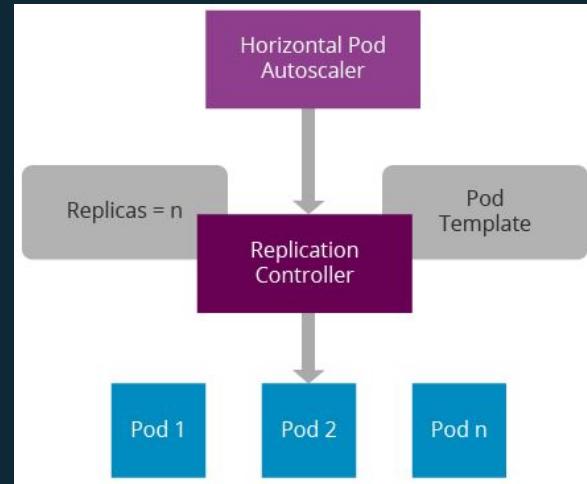
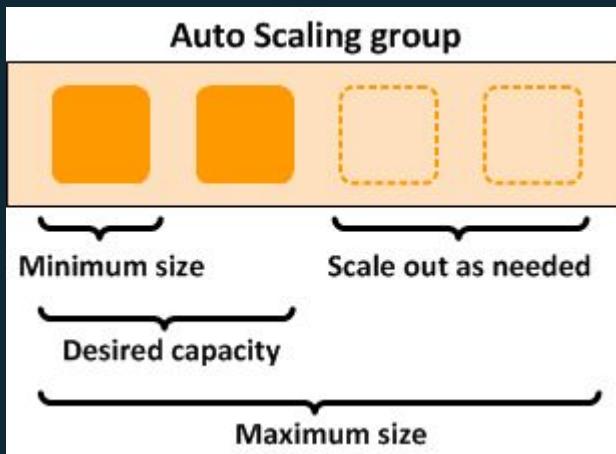


GET api/v1/namespaces/meetup/tips/{4}

*Escalation options on demand with
Autoscaling*

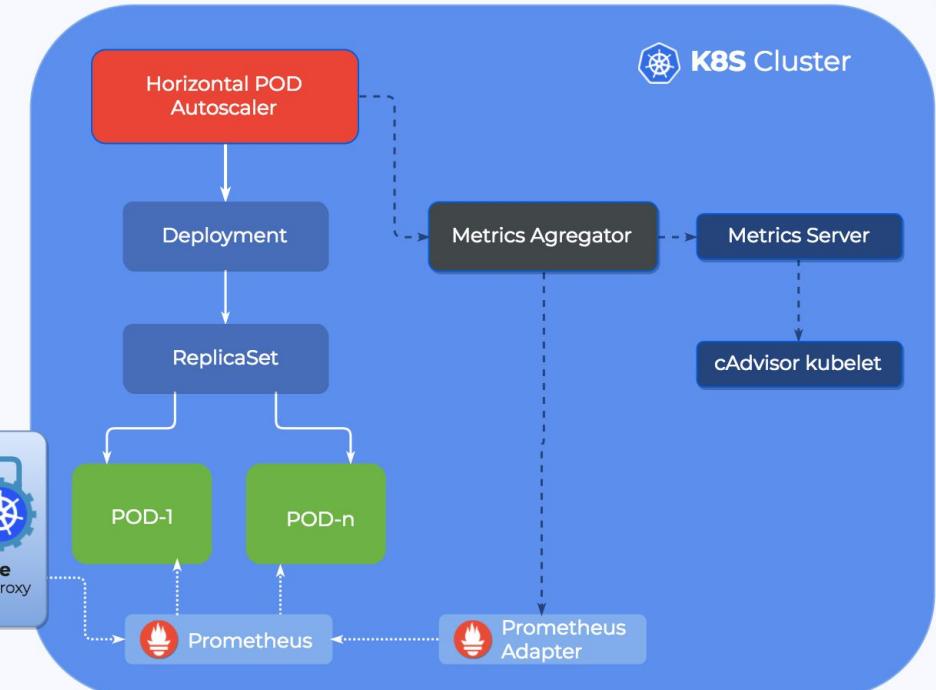
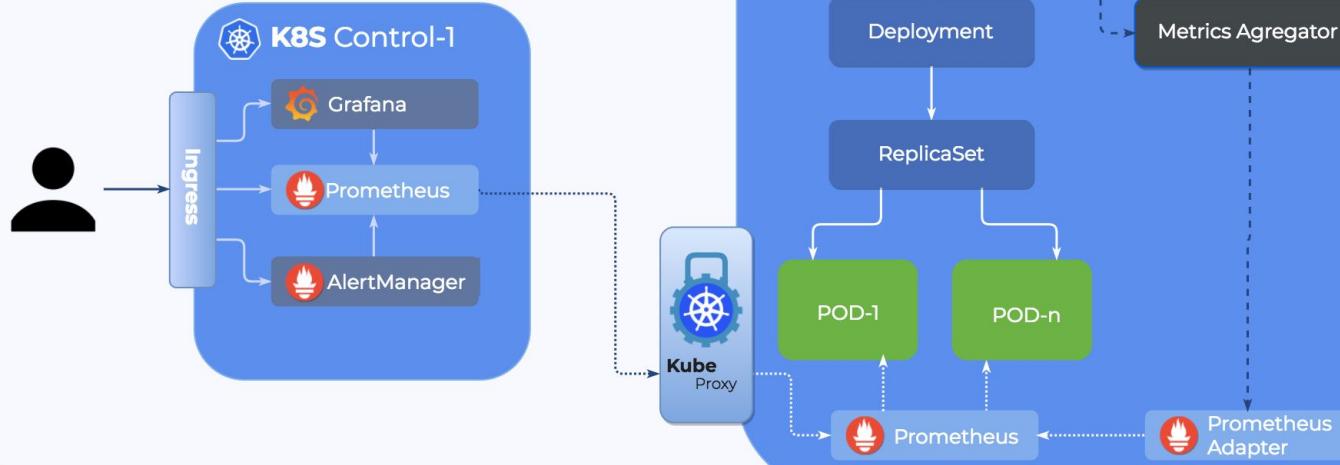
Alternatives of Autoscaling

- Native Autoscaling of AWS
- Autoscaler of k8s Cluster



HPA k8s Cluster

Horizontal POD Autoscaling with **Pipeline**



GET api/v1/namespaces/meetup/tips/{5}

Build, Send, Deploy...

CI/CD applications within Kubernetes

Jenkins

AWS CodePipeline, AWS CodeCommit, AWS CodeBuild

OSS: Spinnaker, Skaffold

AWS partners

- GitLab
- Shippable
- CircleCI
- Codeship

It's time to...

reward our participants!!

Problem N°1 - ECS

1.- If I want to specify the instance where my container should be deployed, What type of **Placement Strategy** should I use to achieve it?

¿?

Problema N°2 - Kubernetes

2.- I want to obtain the logs of my applications running in my cluster and also the logs generated by the same cluster with the aim of visualizing them later in Kibana, based on the logs section presented, how is the Data Collector called that runs as a DaemonSet inside of Kubernetes for the collection of logs?

¿?

re:Invent Recap



Heating up motors



Midnight Madness



Launches, Preview & Pre-Announcements

<https://aws.amazon.com/es/new/reinvent/>

- AWS Robomaker
- AWS DataSync
- AWS Transfer for SFTP
- S3 Intelligent-Tiering
- S3 Batch Operations
- Snowball Edge Compute Optimized
- AWS Amplify Console
- AWS Global Accelerator
- AWS Transit Gateway
- Amazon EC2 A1 Instances
- Amazon EC2 C5n Instances
- Amazon EC2 P3dn instances
- Elastic Fabric Adapter
- Dynamic Training with Apache MXNet on AWS
- Amazon Sagemaker Neo
- Firecracker
- KMS Custom Key Store
- AWS IoT SiteWise
- Amazon EBS Doubles Max

Launches, Preview & Pre-Announcements

<https://aws.amazon.com/es/new/reinvent/>

- Performance of io1 Volumes
- AWS IoT Events
- AWS IoT Things Graph
- S3 Object Lock
- Private Marketplace
- AWS Marketplace for Containers
- Amazon Comprehend Medical
- AWS Ground Station
- ML Insights for QuickSight
- Dashboard embedding in Amazon QuickSight
- DynamoDB transactions
- AWS Elemental MediaConnect
- S3 PUT to Glacier
- CodeDeploy Blue/Green Deployment Support for ECS
- CodePipeline: ECR as a source action
- Amazon Kinesis Data Analytics for Java applications

Launches, Preview & Pre-Announcements

<https://aws.amazon.com/es/new/reinvent/>

- CloudWatch Logs Insights
- Amazon S3 Glacier Deep Archive
- Amazon FSx for Windows File Server
- Amazon FSx for Lustre
- AWS Control Tower
- AWS Security Hub
- Amazon Lake Formation
- DynamoDB On-Demand
- Amazon Timestream
- Amazon Quantum Ledger Database
- Amazon Managed Blockchain
- Amazon Elastic Inference
- AWS Inferentia
- Amazon SageMaker Ground Truth
- AWS Marketplace for ML
- Amazon SageMaker RL
- AWS DeepRacer
- Amazon Textract
- Amazon Personalize

Launches, Preview & Pre-Announcements

<https://aws.amazon.com/es/new/reinvent/>

- AWS Well-Architected Tool
- Amazon Forecast
- AWS Outposts
- EFS Multi-VPC Access
- Amazon RDS on VMware
- On-Demand Hibernated
- AWS Cloud Map
- AWS App Mesh
- Pre-built scikit-learn container
- AWS License Manager
- Custom Runtimes for Lambda
- Ruby support for Lambda
- Lambda Layers
- Nested Applications using Serverless Application Repository
- Step Functions service integrations
- WebSocket support for API Gateway
- ALB Support for Lambda
- Amazon Managed Streaming for Kafka

AWS Transfer for SFTP

Fully managed SFTP service for Amazon S3



Seamless migration
of existing workflows



Fully managed
in AWS



Secure and Compliant



Native integration
with AWS services



Cost-effective



Simple
to use

Amazon S3 Object Lock—What is It?

- New S3 feature that protects data from being overwritten or deleted in order to support regulatory compliance requirements (“WORM”)
- Two modes—Governance Mode & Compliance Mode
 - For Compliance Mode, objects cannot be modified or deleted by any user, including the root account. Once this is set, it cannot be modified
 - For Governance Mode, objects can only be deleted by the AWS accounts that have explicitly been granted permission through IAM policies
- Set at the bucket level—works with lifecycle (to S3-IA/Glacier)

Amazon S3 Intelligent Tiering—What is It?

- New S3 Storage Class that automatically optimizes storage costs for data with changing access patterns
- Stores objects in two access tiers
 - One optimized for frequent access priced the same as S3-Standard
 - One optimized for infrequent access priced the same as S3-Infrequent Access
- Automatically determines which tier to use based on access patterns—no lifecycle policies needed
- Small monthly per object management fee / no per-GB access charges

S3 Glacier Deep Archive

- A pre-announcement
- Offline storage class for preservation, compliance, and DR where likelihood of recall is low and not sensitive to SLA
- Full functionality: 11 9's (3-AZ), Compliance/WORM, lifecycle, direct PUT, etc
- 8-15 hour retrieval
- Pricing targeted to be less than half of Glacier
- Targeting offsite vaulting services
- Offline tape migrations supported today via partners

S3 Glacier Direct PUT to Glacier Class

- Previously to upload to Glacier via S3...
- lifecycle policies including “0-day”
- Nondeterministic window where read method changes
- 2-layers of transaction fees

x-amz-storage-class

If you don't specify, Standard is the default storage class. Amazon S3 supports other storage classes. For more information, see [Storage Classes](#) in the *Amazon Simple Storage Service Developer Guide*.

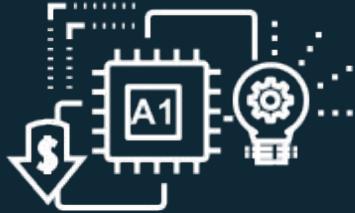
Type: Enum

Default: STANDARD

Valid Values: STANDARD | STANDARD_IA | ONEZONE_IA | REDUCED_REDUNDANCY

Constraints: You cannot specify GLACIER as the storage class. To transition objects to the GLACIER storage class, use lifecycle configuration. For more information, see [Object Lifecycle Management](#) in the *Amazon Simple Storage Service Developer Guide*.

New EC2 Instances



- Lower cost for scale-out workloads
- ARM-based development platform
- AWS Graviton Processor



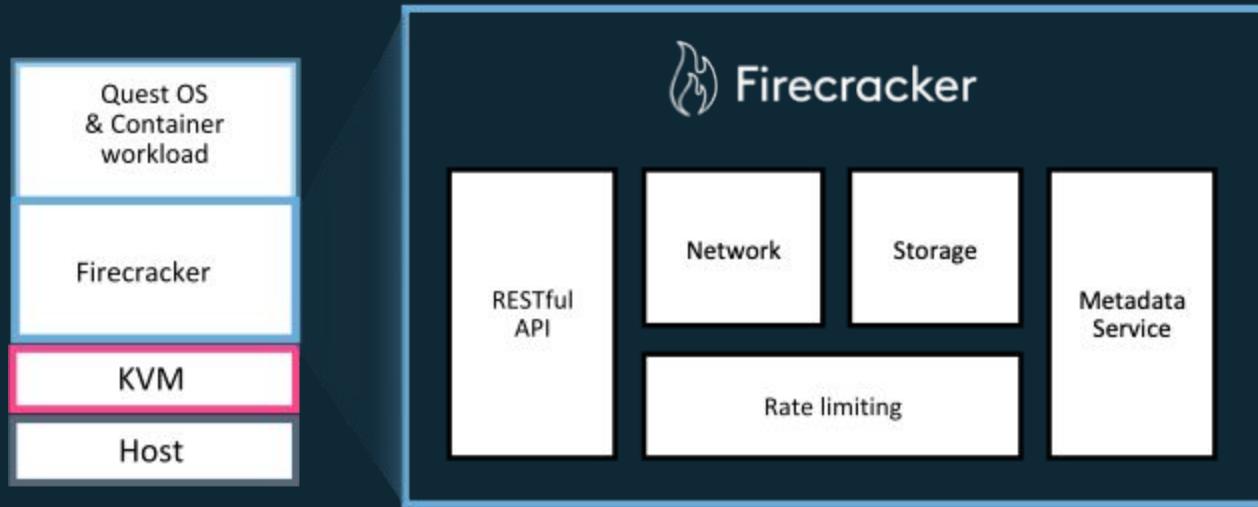
- Faster analytics and big data workloads
- Lower costs for network-bound workloads
- 100 Gbps network throughput



- 100 Gbps network throughput
- NVMe-based SSDs
- 96 vCPUs, 768 GB RAM, 8 x Tesla V100 GPU

Firecracker

(What powers Lambda and Fargate)



Firecracker microVMs have the same security as KVM VMs

Designed for low overhead, high density, and fast start times

Built-in fair sharing

Apache 2.0 OSS
<https://github.com/firecracker-microvm/firecracker>



AWS Global Accelerator

Improve availability and performance for internet applications used by a global audience

- **Static Anycast IP addresses**
 - Never change over the lifetime of an accelerator
 - Safe to put in DNS/firewall/whitelist etc.
- Client traffic ingresses via closest available Edge location
- Route client to closest healthy Endpoint (ELB or Elastic IP)
- Traffic uses AWS Global Network from Edge to Region

AWS Control Tower

- The easiest way to set up and govern a secure, compliant, multi-account environment (Landing Zone)
 - Multi-account structure with AWS Organizations
 - Centrally-managed identities through AWS SSO or Microsoft AD
 - Centralized logging with AWS CloudTrail and AWS Config
 - Account factory through Service Catalog

Automated Landing Zone
with best-practices
blueprints

Guardrails for policy enforcement

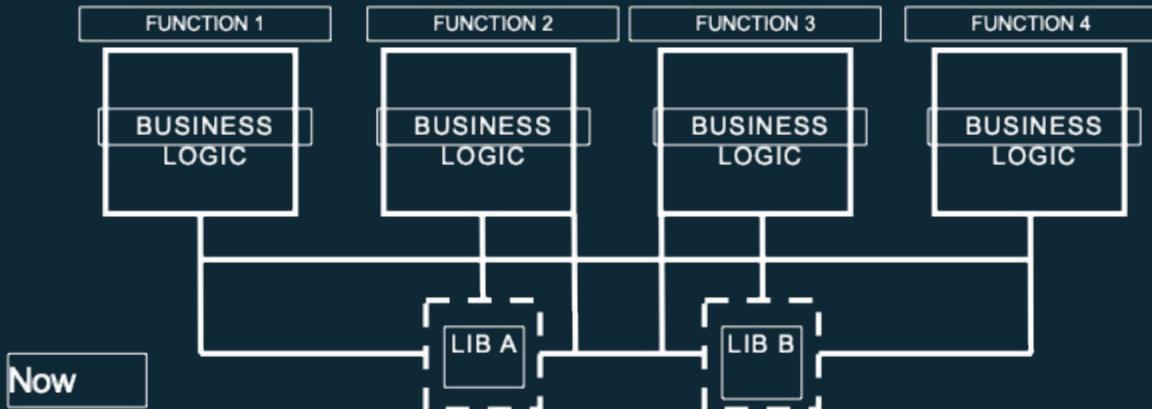
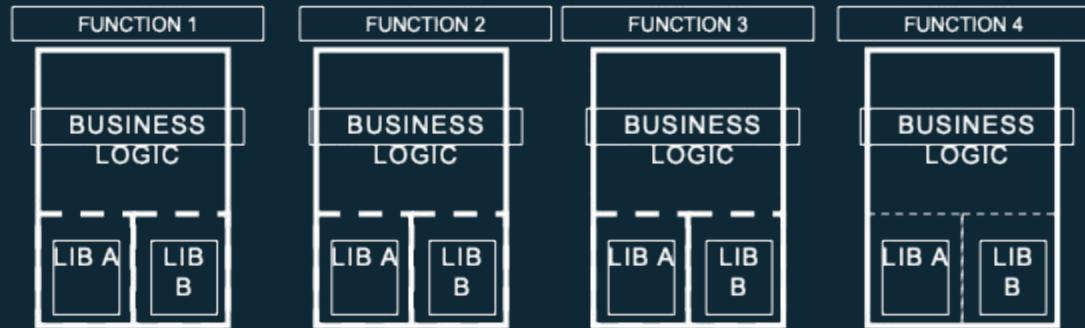
Dashboard for
continuous visibility



Lambda Layers

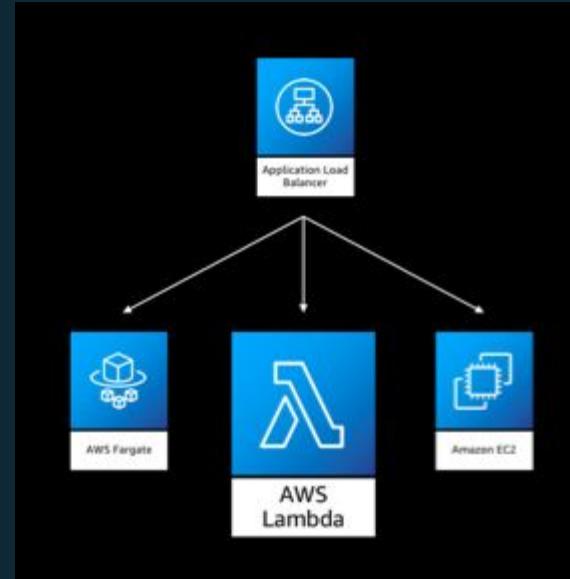
Extend the Lambda execution environment with any binaries, dependencies, or runtimes

Before



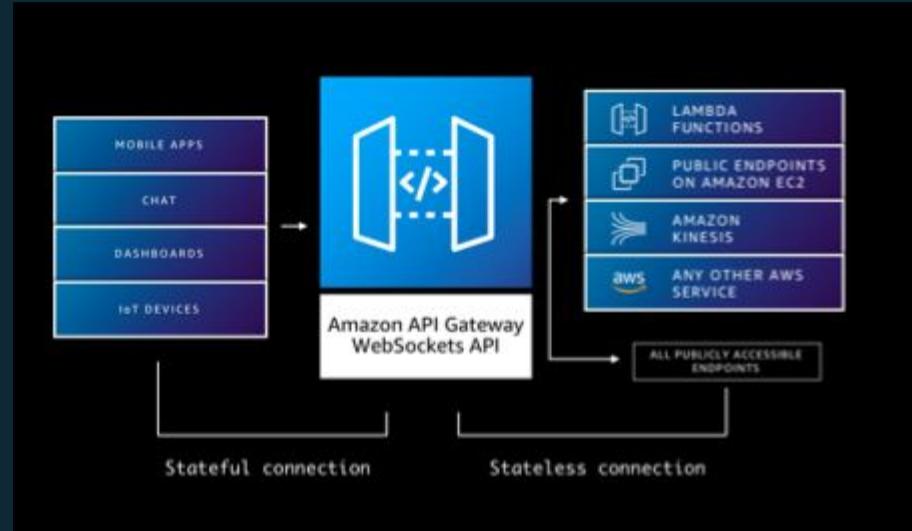
ALB Support for Lambda

Integrate Lambda functions
into existing web
architectures



WebSocket support for API Gateway

Build real-time two way communication applications

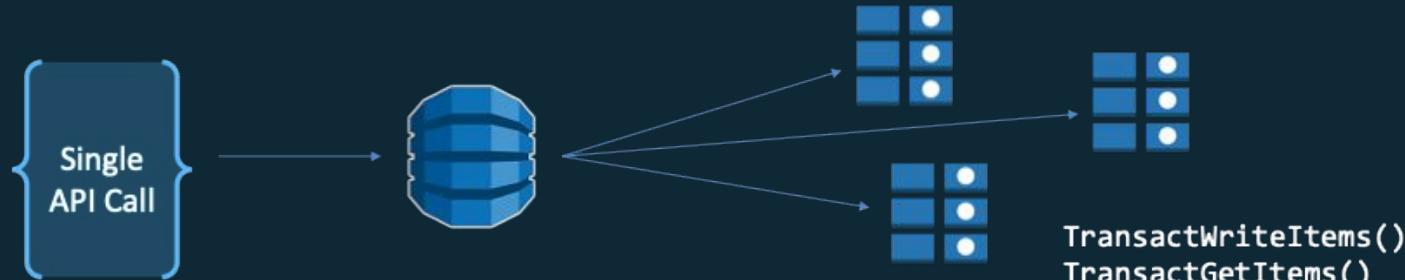


Introducing AWS RoboMaker

<https://twitter.com/awsreinvent/status/1066966596546060289>

DynamoDB transactions

Simplify your code by executing multiple, all-or-nothing actions within and across tables with a single API call.



- Provides atomicity, consistency, isolation, and durability (ACID) in DynamoDB.
- You can perform transactions both within and across multiple DynamoDB tables.
- Native, server-side solution that provides better performance and lower costs than client-side libraries.

DynamoDB: Capacity managed for you

Provisioned

Govern Max Consumption

Auto Scaling

Set a Minimum

On-Demand

No Limit

Start at Zero



Amazon Timestream (Preview)

- Fast, scalable, and fully managed time series database

Typical Use Cases: Application events, IoT Sensor Readings and DevOps data

1,000x faster at 1/10th the cost of relational databases



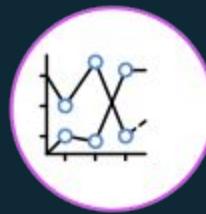
Collect fast moving time-series data from multiple sources at the rate of millions of inserts per second

Trillions of daily events



Capable of processing trillions of events daily; the adaptive query processing engine maintains steady, predictable performance

Analytics optimized for time series data



Built-in analytics for interpolation, smoothing, and approximation to identify trends, patterns, and anomalies

Serverless



No servers to manage; time-consuming tasks such as hardware provisioning, software patching, setup, & configuration done for you

Amazon Managed Blockchain features

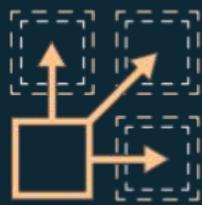


Fully managed
Create a blockchain network in minutes

HYPERLEDGER
FABRIC
ethereum



Decentralized
Democratically govern the network



Reliable & scalable
Backed with Amazon QLDB technology



Low cost
Only pay for resources used



Integrated
Send data to Amazon QLDB
for secure analytics

QLDB (Preview)

- Fully managed ledger database
- Track and verify history of all changes made to your application's data

Immutable and transparent



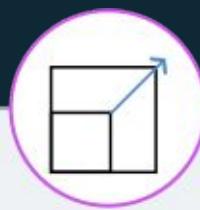
Append-only, immutable journal tracks history of all changes which cannot be deleted or modified. Get full visibility into entire data lineage

Cryptographically verifiable



All changes are cryptographically chained and verifiable

Highly scalable



Executes 2 – 3X as many transactions than ledgers in common blockchain frameworks

Easy to use



Flexible document model, query with familiar SQL-like interface

Purpose-built databases



Relational



Key-value



Document



In-memory



Graph



Time-series



Ledger



Amazon
RDS

Aurora Community Commercial



DynamoDB



ElastiCache



Neptune



Timestream



Quantum

To talk

- Amazon Lake Formation
- AWS DeepRacer
- AWS Well-Architected Tool

Multiple events



Characters



A Cloud Guru Founder
@KroonenburgRyan



AWS Serverless Hero
@theburningmonk

Highlights



Highlights



Thank you!

e.miranda@globant.com

gvasquez@waypoint.cl

