CON217

Roadmap for Containers, Application Networking, and Amazon Linux

Abby Fuller

Containers and Linux @ AWS

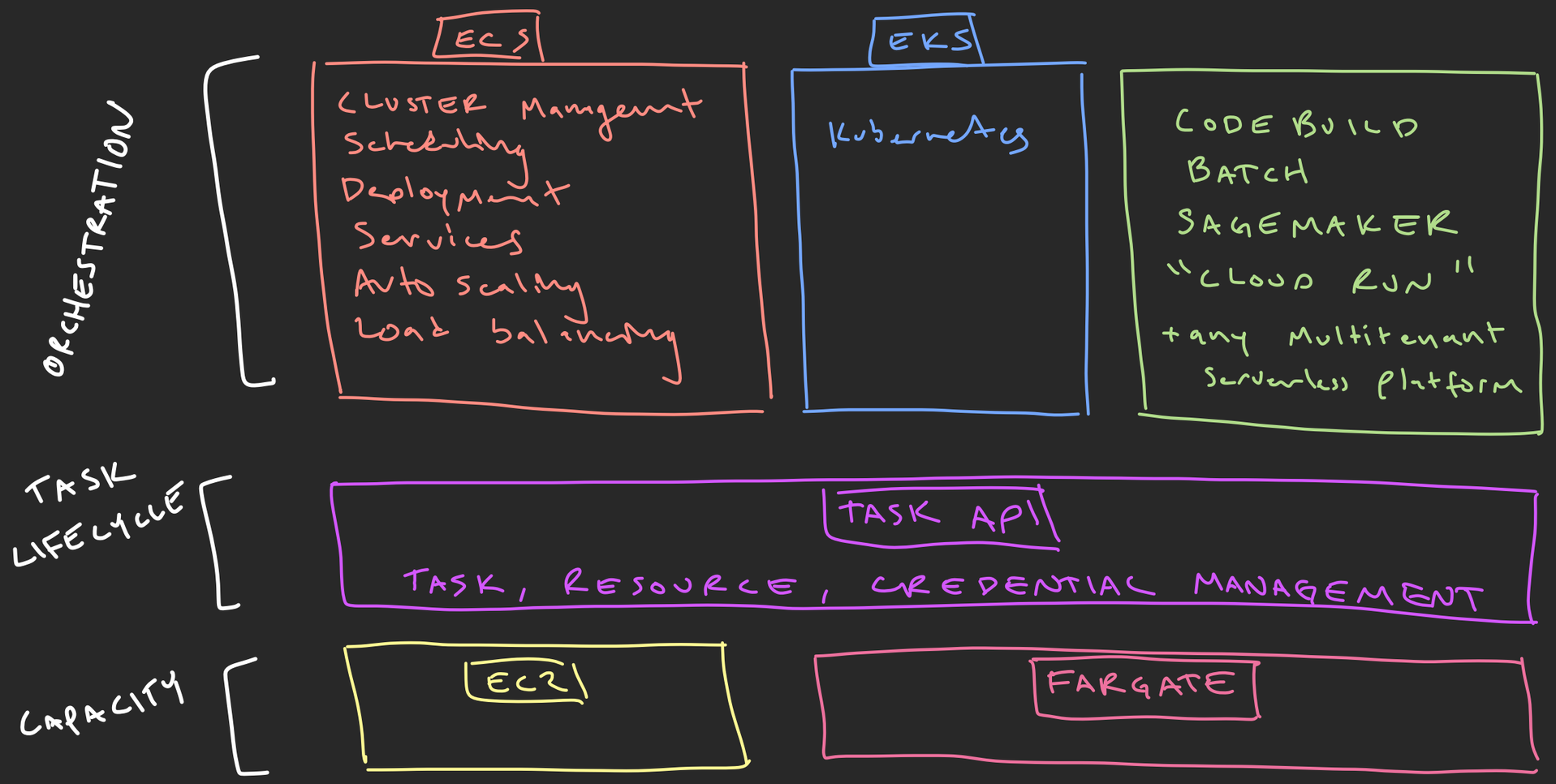
AWS re:Invent

aws

Is this a state of the union?

Kind of.

What’s the big picture?


ORCHESTRATION
ECS
CLUSTER Mangement
Scheduling
Deployment
Services
Auto scaling
Load balancing
EKS
Kubernetes
CODE BUILD
BATCH
SAGEMAKER
"CLOUD RUN"
+any Multitenant
Serverless platform
TASK LIFECYCLE
TASK API
TASK, RESOURCE, CREDENTIAL MANAGEMENT
CAPACITY
EC2
FARGATE

Ø. STORE YOUR IMAGES:

ECR

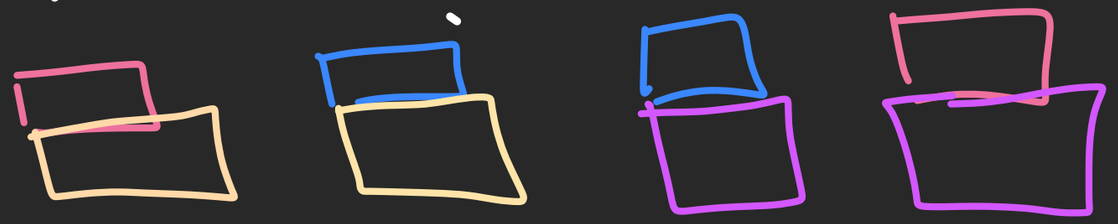
1. CHOOSE YOUR SCHEDULER:

ECS EKS

2. CHOOSE WHERE/HOW TO RUN:

FARGATE EC2

3. BUILD!



What have we been thinking about since last year?

Patterns and abstractions make applications repeatable

With distributed systems comes complexity; abstractions can mitigate this. How can we make building and running applications streamlined, repeatable, and modular? [ecs-cli 2.0]

Control over what you care about (and sensible defaults for the rest)

The other face of abstractions is the ability to customize where it matters to you, whether that means control over how much you want to pay [Fargate Spot, Savings Plan], or how many knobs you want to tweak [EKS on Fargate, EKS IAM roles for service accounts/pods].

This also means control over the tools you use: We aim to provide sensible default tools built off of OSS tools and standards [FireLens], but it also means the ability to use a totally different set of tools if you want [eksctl with Weave, upstream Kubernetes for EKS, Argo/Flux]. We contribute back to projects that our customers use to help them work better with AWS [atlassian/escalator, spinnaker]

Always-evolving compute

It’s not just about reimplementing EC2 at the container level - we are fundamentally changing how developers think [like Lambda]. With Fargate [ECS on Fargate, EKS on Fargate], developers no longer have to manage their infrastructure at the cluster level, but it can’t stop there.

How can we remove the concepts of clusters entirely? How can we eliminate the need for things like CRDs?

Abstractions, patterns, integrations, sensible defaults, and the ability to tweak just the settings you care about. Everything you write should be be business logic, but how can we get there?

Stronger integrations with other AWS services

Integrations between AWS services should feel seamless. Not just container native, but AWS native.

Whether it's autoscaling, networking, or monitoring, we've focused on having tight, first class integrations between all parts of AWS [ENI trunking, CloudWatch Container Insights, Cluster Autoscaling].

More community transparency, more feedback from our users

This year, we've opened up our roadmap for all of the AWS container and application networking services [roadmaps for ECS, Fargate, EKS, ECR, App Mesh]. Developers can see what we're working on, what's shipping soon, and what we're thinking about building next, plus join developer previews and comment on RFCs.

Let’s talk about what’s new

We’ve launched a lot over the last year

Here’s a few highlights

ECS (Elastic Container Service)

Compute Savings Plan 🆕

Support for additional log drivers (SumoLogic, FluentD)

Container Insights integration

Additional CloudWatch events

Additional CloudFormation support

Multiple target groups per load balancer

Run task definitions locally

FireLens support 🆕

ENI density improvements

EKS (Elastic Kubernetes Service)

Support for Windows nodes 🆕
Managed worker node groups 🆕
Instance draining with Spot
IAM Roles for service accounts (pods)
Container insights integration
New CNI versions
Deep learning benchmarking utility
Control plane metrics endpoint
Argo/Flux GitOps 🆕

Fargate

FireLens support 🆕

Support for additional log drivers

More CloudFormation support

More CloudWatch events support

More regions

Support for Compute Savings Plan 🆕

Price reduction (1/7/2019)

ECR (Elastic Container Registry)

FIPS compliance

VPC private endpoint policies (PrivateLink support)

Immutable image tags

Support for additional CloudWatch events

Image vulnerability scanning 🆕

EventBridge support 🆕

App Mesh

GA! + Preview Channel

HTTP2/gRPC support 🆕

Cookie-based, HTTP and TCP based routing

Weave Flagger integration

EKS and CloudMap integration

App Mesh Controllers for Kubernetes

In Preview Channel: end-to-end encryption with ACM and customer-managed certificates, cross account support

What about re:Invent launches?

New and exciting at re:Invent

EKS on Fargate

Fargate Spot

Cluster Autoscaling

ECS Capacity Providers

ecs-cli v2 preview

ECS, EKS, and App Mesh support for Outposts

AWS EKS support for AWS Fargate

Use EKS to run Kubernetes pods on AWS Fargate.

In other words, run Kubernetes-based applications without managing or provisioning infrastructure. With Fargate, define and pay for resources at the pod level. Pods run with a VM-level isolation boundary.

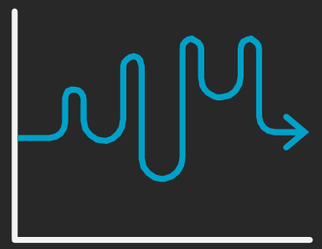
With Fargate, customers don’t need to be Kubernetes operations experts to run a secure, available, cost-optimized cluster.

Fargate Spot

Now run your Fargate-based tasks on Spot capacity. Fargate Spot is spare Fargate capacity at a savings of up to 70%.

Fargate

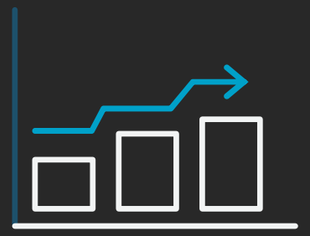
Pay for containers per-second with no long-term commitment



Capacity needs can change rapidly

Fargate Savings Plan

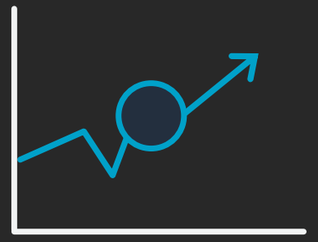
Make a 1- or 3-year commitment and receive a significant discount



Baseline compute needs known in advance

Fargate Spot

Spare capacity with savings up to 70% off Fargate standard pricing



Fault-tolerant, flexible workloads

EC2 Spot vs Fargate Spot

Amazon EC2 Spot

Unused EC2 Capacity

Save up to 90% over On-Demand

Can be *reclaimed* by EC2 (with two-minute warning)

You choose instance pools, recommend flexibility across multiple instance types and use Capacity optimized allocation strategy

AWS Fargate Spot

Unused Fargate Capacity

Save up to 70% over standard Fargate

Can be *reclaimed*
(with two minute warning)

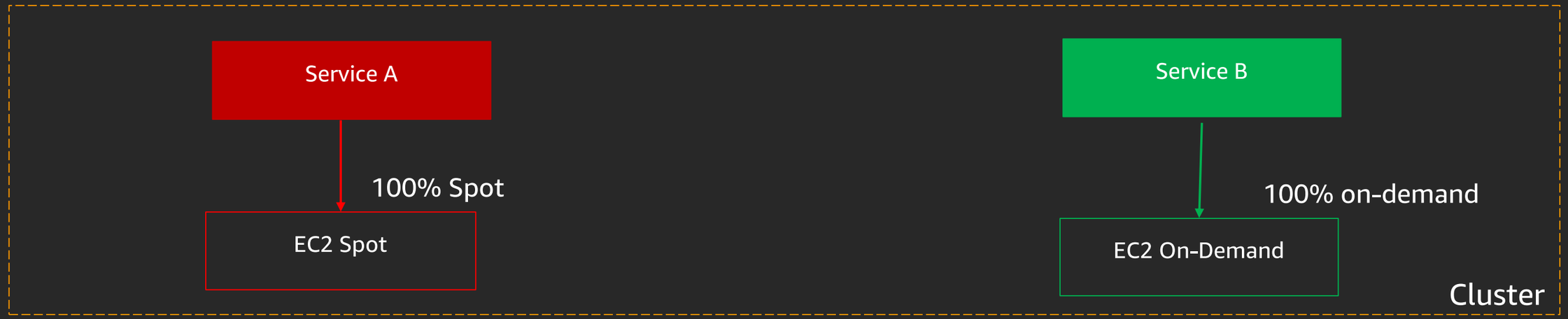
Automatic diversification

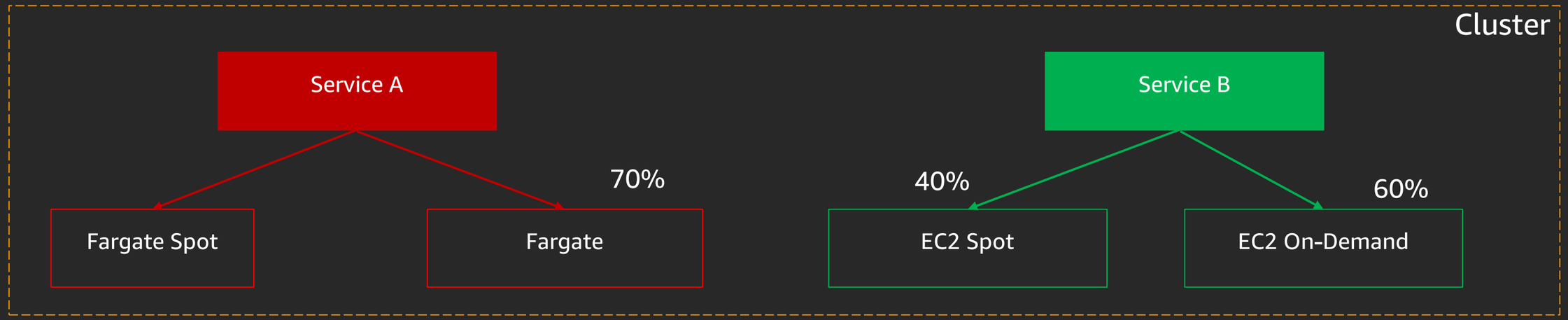
ECS Cluster Capacity Providers

With ECS Capacity Providers, customers will be able to define multiple Auto Scaling Groups in a single cluster; each ASG is associated with its own Capacity Provider.

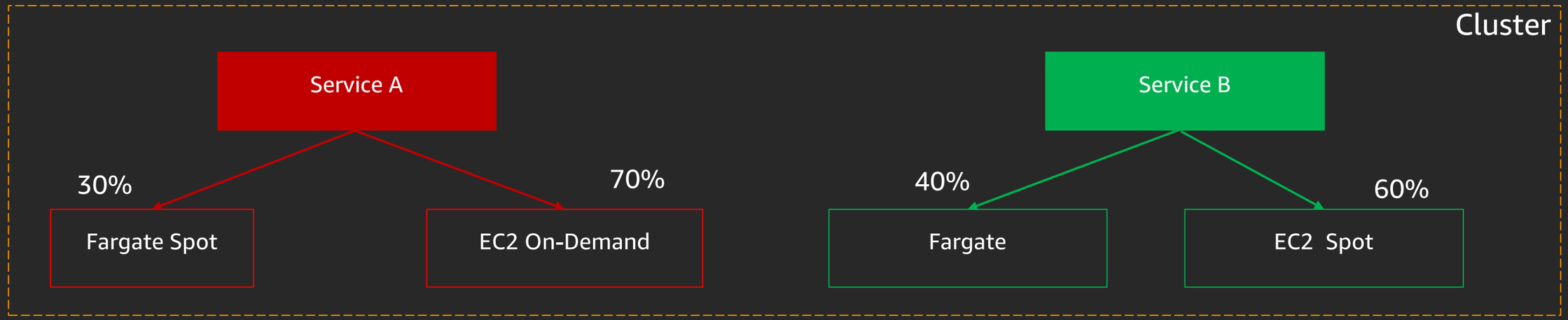
A Capacity Provider can be EC2 Spot, EC2 On-Demand, Fargate Spot, or Fargate On-Demand.

Capacity Providers





Future: Mixing Capacity Providers



ECS Cluster Autoscaling

Two pieces: a new ECS cluster scaling metric, and container-aware instance termination management.

The new metric, called the task reservation, measures the total percentage of cluster resources needed by all ECS workloads in the cluster. This metric enables the scaling policy to scale out quicker and more reliably than it could when using CPU or memory reservation metrics. Customers can also use this metric to reserve spare capacity in their clusters.

ECS Cluster Autoscaling

Part two: Instance termination management

With instance termination management, ECS controls which instances the scaling policy is allowed to terminate on scale in, with the objective of minimizing disruptions of running containers. These improvements help customers achieve lower operational costs and higher availability of their container workloads running on ECS.

ecs-cli v2

Create, release and manage production ready containerized applications on ECS. Applications built with the ecs-cli are modern and serverless by default, and include operations (like debugging and deployments) as part of the workflow.

Once you've built something you're excited to deploy, let the ecs-cli set up a CI/CD pipeline for you, with built-in testing hooks and manual gates. Tail your logs, monitor your key metrics and push updates all from the comfort of your terminal.

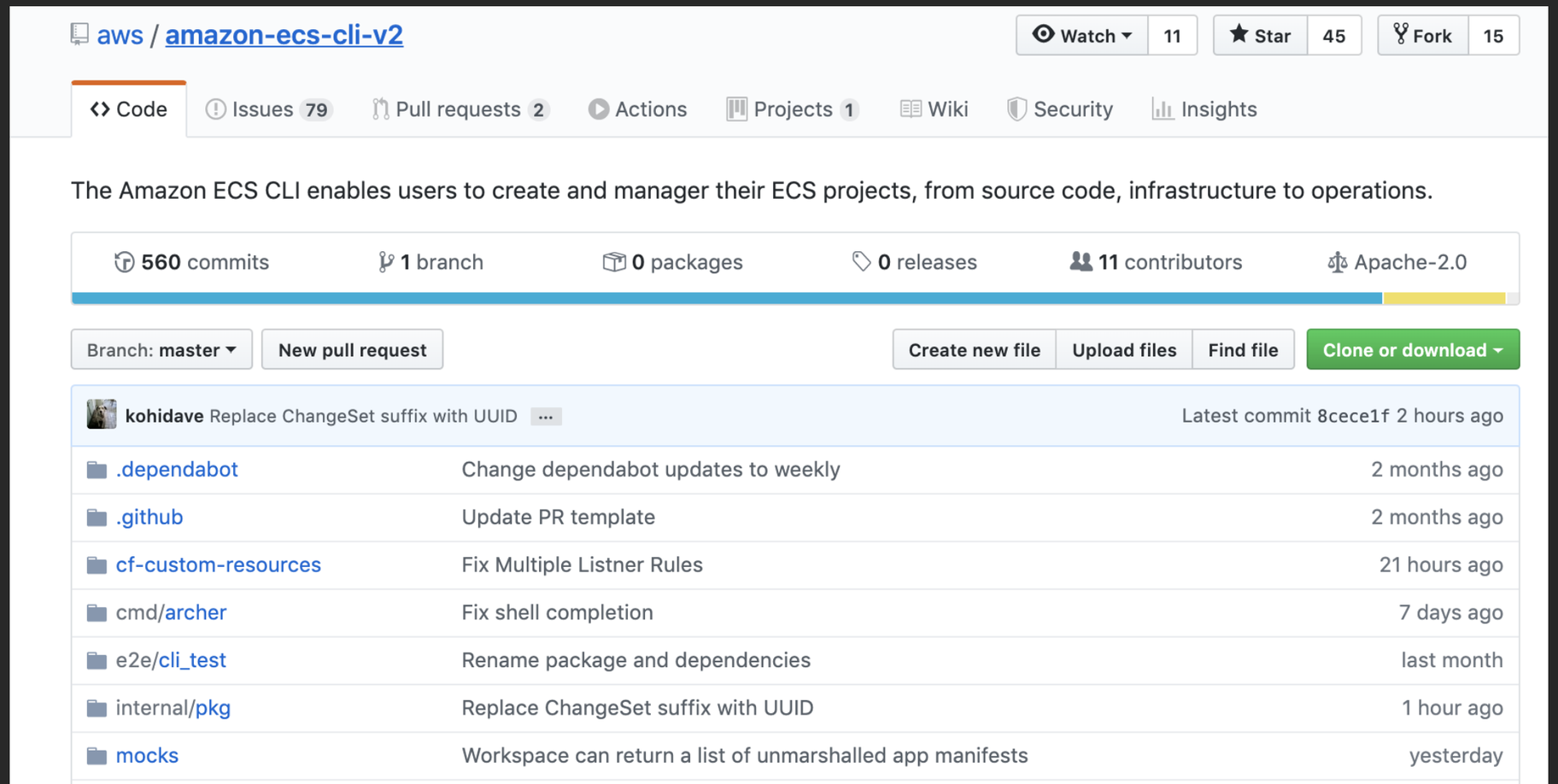
ecs-cli v2

Use the ecs-cli to:

- Bring your existing Docker apps
- Set up staging and production environments, cross region and cross account
- Set up production-ready, battle-tested ECS Clusters, Services and infrastructure
- Set up CI/CD Pipelines for all of the micro-services that make up your Project
- Monitor and debug your applications

How can you find out more?

The ecs-cli is open source, and published on GitHub. Ask questions, file issues, or download the source here.



How can you learn more?

Some breakouts and other options (that you can still go to)

- **CON333-R2 - Best practices for CI/CD using AWS Fargate and Amazon ECS (Clare Liguori, Hsing-Hui Hsu)**
- **CON208-R2 - [REPEAT 2] Build your microservices application on AWS Fargate (Nathan Peck, Adam Keller)**
- **CON328-R1 - [REPEAT 1] Improving observability of your containers (Akshay Ram, Shubha Rao, Sharanya Devaraj)**
- **CON324/325 - Container Cost Optimization, ECS Capacity Providers (both Nick Coult)**
- **CON312 - chalk talk on Cluster Autoscaling**
- **Containers coverage on twitch.tv/aws**

After re:Invent

Containers and **App Mesh** GitHub roadmaps (more on that in a bit)

Container blog: https://aws.amazon.com/blogs/containers/welcome-to-the-aws-containers-blog/

Breakout sessions from re:Invent are posted on YouTube

Workshops: **ecsworkshop.com**, **eksworkshop.com**

App Mesh Examples: https://github.com/aws/aws-app-mesh-examples

Let’s talk about Amazon Linux

AWS re:Invent

aws

What’s Amazon Linux?

Linux distro designed to provide a stable, secure, and high performance execution environment for applications running on EC2. Support for the latest EC2 instance type features plus packages that enable easy integration with AWS. We provide security and maintenance updates to all instances running the Amazon Linux AMI. Packages available via *yum*.

What are we focusing on?

Upcoming AL1 deprecation (December 30, 2020)

- Extended maintenance support through June 30, 2023
- Specific packages/CVE classes that will be covered by EMS coming soon

What's new with AL2? Let's talk about Extras.

Iterating on the container space, both as a host OS for containers (like the ecs-optimized AMI), and as an OS inside containers (*FROM: amazon-linux-2*)

Getting more frequent package and AMI updates (newer software faster)

Most importantly: We want to hear from you!

abbyfull@amazon.com
trawets@amazon.com

What’s next?

What do YOU want to see?

Here’s what we’re thinking about

tl;dr: more focus on abstractions and developer experience, more ability to tweaks knobs when you need them.

How can you get your voice heard?

Public GitHub roadmaps

aws / containers-roadmap

Unwatch 487 | Star 2k | Fork 89

Code | Issues 373 | Pull requests 1 | Actions | Projects 1 | Security | Insights | Settings

containers-roadmap

Updated 2 hours ago

Filter cards | + Add cards | Fullscreen | Menu

26 Researching

- [Fargate] [Feature]: Support for UDP + NLB with Fargate services
  #445 opened by danrubenstein
  ECS Fargate Proposed
- [EKS] [request]: Definable Service ClusterIP CIDR range
  #216 opened by dewjam
  EKS Proposed
- ecs UI makes it hard to see image name
  #432 opened by ORESoftware
  Console ECS
- [ECS] Pre-install AWS SSM Agent on the ECS-optimized Amazon Linux 2 AMI
  #426 opened by coultn
  ECS
- [ECS] Increase maximum number of containers in a task definition from 10 to 30
  #368 opened by coultn
  ECS
- EKS - Cost Options on Control Plane (developer friendly)
  #45 opened by jpoley
  EKS Proposed

31 We're Working On It

- [ECS]: ECS CLI v2
  #513 opened by kohidave
  ECS
- AWS Service Operator
  #456 opened by mhausenblas
  EKS
- [EKS] [request]: A reliable EKS AMI release process
  #319 opened by max-rocket-internet
  EKS Proposed
- [EKS]: Next Generation AWS VPC CNI Plugin
  #398 opened by tabern
  EKS Proposed
- [EKS] [CRI]: Support for Containerd CRI
  #313 opened by paavan98pm
  EKS Proposed
- Fargate ephemeral volume encryption
  #314 opened by akshayram-wolverine
  Fargate
- New EKS Region: GovCloud West
  #253 opened by joemccall86
  EKS Proposed

15 Coming Soon

- [ECR][request] Image security scanning return 'undefined'
  #553 opened by zanhsieh
  ECR
- EKS on Fargate
  #32 opened by outofcoffee
  EKS Fargate
- EKS: Envelope Encryption of Kubernetes Secrets
  #530 opened by tabern
  EKS
- ECS CodeDeploy canary deployments
  #229 opened by clareliguori
  ECS Fargate
- [ECS] [CodeDeploy] [CloudFormation]: CloudFormation support for BLUE/GREEN deployments on ECS
  #130 opened by ilyasotkov
  ECS Proposed
- [EKS]: Kubernetes v1.11 End of Support (Nov 4, 2019)
  #464 opened by mikestef9
  EKS
- [ECS] How to programmatically get event "unable to place a task because

1 Developer Preview

- [EKS]: Support for Arm Nodes - EC2 A1 Instances
  #264 opened by tabern
  Developer Preview EKS

107 Just Shipped

- New EKS Region: Canada Central
  #113 opened by tabern
  EKS
- [Fargate/ECS] [request]: Fargate in South America (São Paulo)
  #545 opened by pgarbe
  Proposed
- [fargate]: pricing for persistent applications
  #109 opened by mwarkentin
  Fargate Proposed
- Fargate Log Driver Support - Sumo Logic
  #39 opened by nikovirtala
  Fargate Proposed
- Fargate Log Driver Support v2
  #10 opened by abby-fuller
  Fargate
- [Fargate] [request]: Support for FluentD log driver
  #383 opened by chungath
  Fargate Proposed
- [EKS] Managed worker nodes
  #139 opened by tabern
  EKS

Public GitHub roadmaps

aws / aws-app-mesh-roadmap

Unwatch 95 | Star 128 | Fork 7

Code | Issues 65 | Pull requests 0 | Actions | Projects 1 | Security | Insights | Settings

aws-app-mesh-roadmap

Updated 3 days ago

Filter cards | + Add cards | Fullscreen | Menu

6 Researching

- Open Source the App Mesh Envoy Image build, release, and validation tools
  #5 opened by dastbe
  Priority: Medium | Roadmap: Accepted
- HIPAA eligibility
  #66 opened by shubharao
  Phase: Researching | Roadmap: Accepted
- Circuit Breaker Policy
  #6 opened by jamsajones
  Phase: Researching | Roadmap: Accepted
- Integration with AWS Lambda
  #33 opened by akahen
  Roadmap: Accepted
- API Gateway as ingress to App Mesh
  #111 opened by shubharao
  Phase: Researching | Roadmap: Accepted
- Open-source App Mesh Envoy Management Service (EMS)
  #42 opened by jamsajones
  Phase: Researching | Priority: High | Roadmap: Accepted

8 We're Working On It

- Support App Mesh Across Multiple Accounts
  #64 opened by dastbe
  Phase: Working on it | Roadmap: Accepted
- authN based on mTLS
  #34 opened by jamsajones
  Phase: Working on it | Roadmap: Accepted
- Support SPIFFE/SPIRE for mTLS
  #68 opened by paavan98pm
  Phase: Working on it | Roadmap: Accepted
- Use App Mesh for ingress routing
  #37 opened by jamsajones
  Phase: Working on it | Roadmap: Accepted
- Region expansion
  14 of 23
  #1 opened by jamsajones
  Phase: Working on it | Roadmap: Accepted
- Commit X-Ray tracer plugin to Envoy upstream
  #21 opened by jamsajones
  Phase: Working on it | Roadmap: Accepted

1 Coming Soon

- End to end encryption of traffic with customer provided certs
  #38 opened by jamsajones
  Phase: Coming Soon | Roadmap: Accepted

1 Available in Preview Channel

- End to end encryption of traffic with ACM managed certs
  #39 opened by jamsajones
  Phase: In Preview | Roadmap: Accepted

32 Just Shipped

- Increase service limits
  #110 opened by shubharao
  Roadmap: Shipped
- Feature Request: Release Envoy 1.12.0
  #132 opened by lavignes
  Envoy Docker Image | Roadmap: Shipped
- Http2/gRPC support
  #96 opened by skiyani
  Roadmap: Shipped
- [BUG] X-Ray extension causing seg-fault in app-mesh-envoy
  #99 opened by kiranmeduri
  Bug
- Support Envoy 1.11
  #67 opened by shubharao
- Cookie based routing
  #14 opened by jamsajones
- ECS integration with App Mesh in the ECS console
  #8 opened by jamsajones
- Bring Envoy from official release
  #10 opened by jamsajones
  Roadmap: Shipped
- VPC Endpoint/Private Link for App Mesh Envoy xDS API

Developer previews

aws / containers-roadmap

Unwatch 487 | Star 2k | Fork 89

Code | Issues 373 | Pull requests 1 | Actions | Projects 1 | Security | Insights | Settings

[EKS]: Support for Arm Nodes - EC2 A1 Instances #264

Edit | New issue

Open tabern opened this issue on Apr 23 · 20 comments

tabern commented on Apr 23 • edited — Member

Amazon EKS now supports Arm processor EC2 A1 instances as a developer preview. You can now run containers using EC2 A1 instances on a Kubernetes cluster that is managed by Amazon EKS.

Learn more and get started here: https://github.com/aws/containers-roadmap/tree/master/preview-programs/

Learn more about Amazon A1 instances: https://aws.amazon.com/ec2/instance-types/a1/

Please leave feedback and comments on the preview using this ticket.

5

autarchprinceps, thomashusa, binoculars, zhouziyang, and lyndon160 reacted with thumbs up emoji

view labels on Apr 23

tabern mentioned this issue on Apr 23

EC2 A1 for EKS Preview Program #265 — Merged

Assignees

No one—assign yourself

Labels

Developer Preview

EKS

Projects

containers-roadmap

Developer Preview

Milestone

No milestone

Notifications — Customize

Unsubscribe

Developer previews

aws / aws-app-mesh-roadmap

Unwatch 95 | Star 128 | Fork 7

Code | Issues 65 | Pull requests 0 | Actions | Projects 1 | Security | Insights | Settings

End to end encryption of traffic with ACM managed certs #39

Edit | New issue

Open **jamsajones** opened this issue on Nov 28, 2018 · 32 comments

jamsajones commented on Nov 28, 2018 — Member

No description provided.

55

mumoshu commented on Dec 5, 2018

This is very interesting indeed!

I was planning to use Istio to enable end-to-end encryption between microservices, so that we have no chances to connect wrong services due to reused VPC IPs. But maintaining Istio CA/Auth/Citadel and the other parts of Istio's control-plane and its foundation, K8S cluster, just for a service mesh seemed an overkill.

I'd expect App Mesh integrated with ACM to provide the same benefit, without the operational burden.

Assignees

bcelenza

Labels

Phase: In Preview

Roadmap: Accepted

Projects

aws-app-mesh-roadmap

Available in Preview Cha...

Milestone

No milestone

Notifications Customize

Thank you!

Abby Fuller

@abbyfuller
abbyfull@amazon.com

AWS re:Invent

aws

Please complete the session survey in the mobile app.

AWS re:Invent

aws