
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

CON322

Building an enterprise-scale Kubernetes platform at Expedia Group

Dan Gorst

Principal Systems Engineer
Expedia Group

Ben Fields

Solutions Architect
Amazon Web Services

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Scene setting: From brands to a global platform

Expedia Group runtime platform overview

Tech deep(er) dives

- Cluster(s) per team or multi-tenancy?
- Control plane and Cluster Federation
- Providing more than compute with Crossplane
- Handling compute needs with Karpenter
- Migration support is a first-class feature

Overview and key takeaways

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

From brand tech stacks to a global platform



Expedia

Hotels.com

Vrbo

travelocity

hotwire

ORBITZ

ebookers

CheapTickets

CarRentals.com

Expedia Cruises

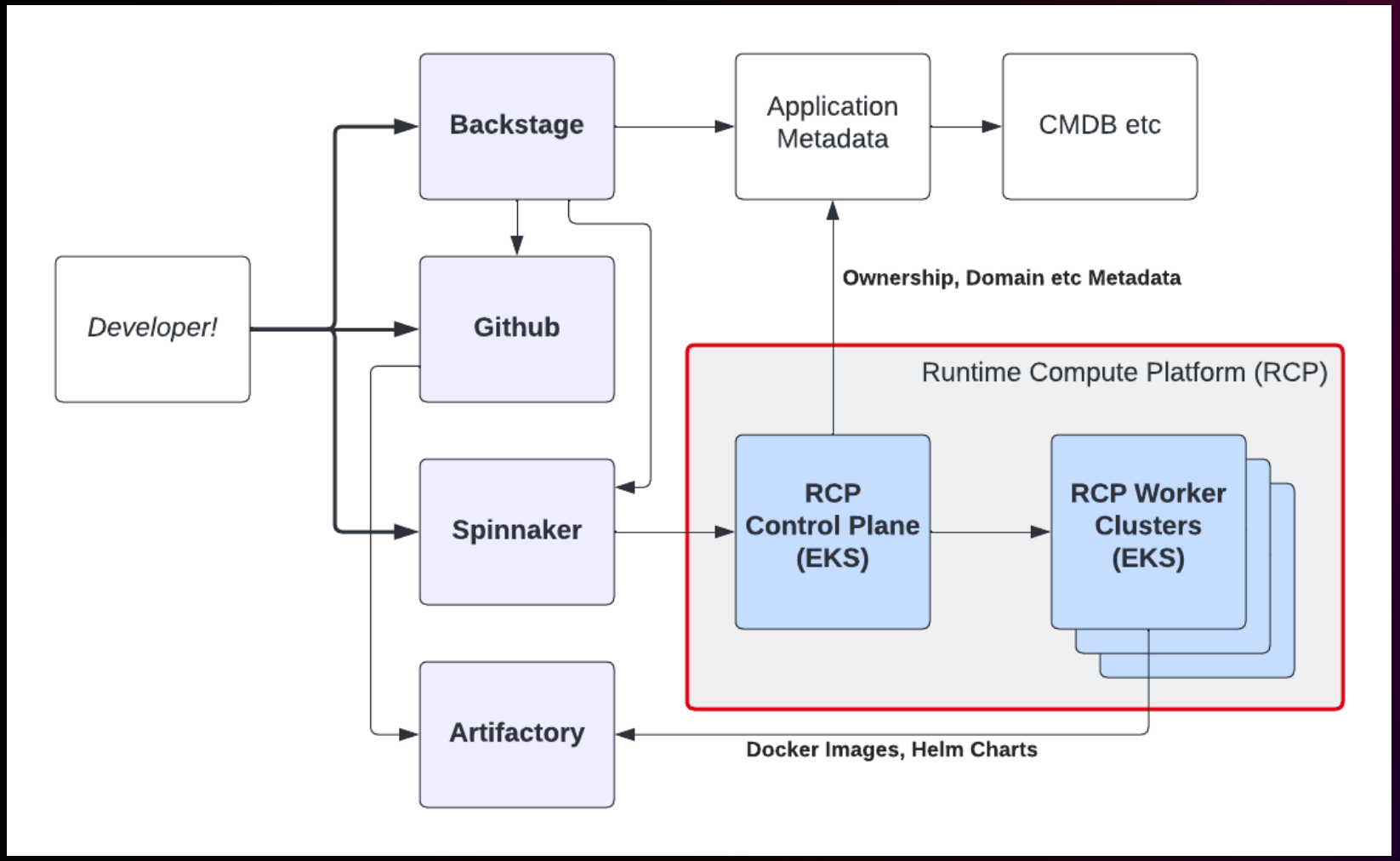
wotif

trivago

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Expedia runtime platform overview



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Expedia runtime platform overview

What didn’t work out before?

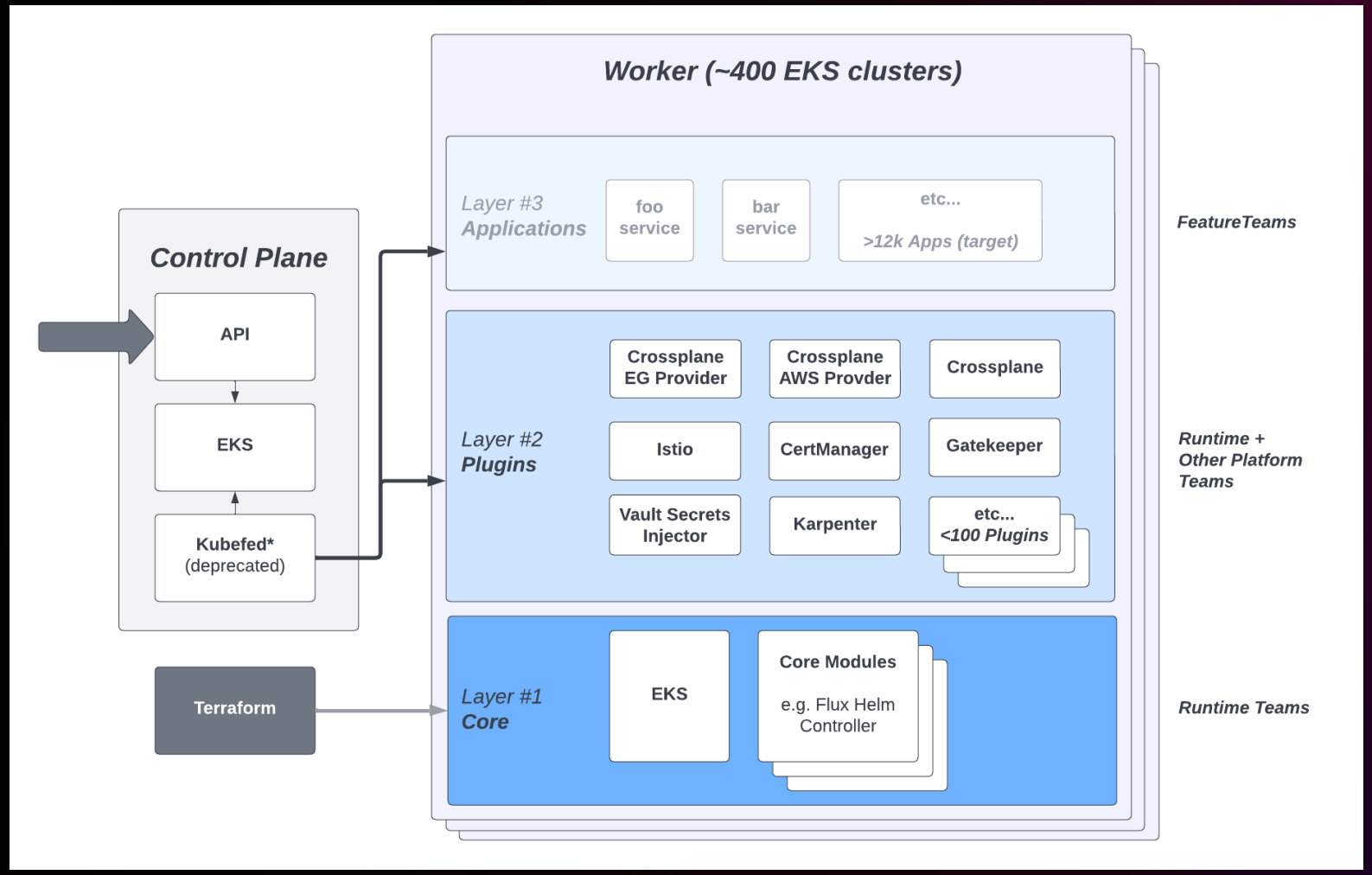
- Self-managed Kubernetes
- Bundling cluster infrastructure and common services (e.g., service mesh, logging, metrics, etc.) as a platform monolith
- Cookie-cutter Kubernetes clusters and team-owned AWS accounts
- One-size-fits-all approaches (e.g., devs must know Kubernetes or devs must know custom abstraction layer)

What we did this time . . .

- Switched to Amazon EKS
- Split core infrastructure Terraform, and created a plugin mechanism to independently deliver common services to all clusters
- Fully managed, multi-tenant AWS accounts and clusters
- Provided complimentary simplified and advanced experiences to meet a broader developer base

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Expedia runtime platform overview



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Expedia runtime platform overview

In numbers (excluding on-premises and legacy environments):

- ~200 AWS accounts
- ~400 Amazon EKS clusters
- ~10000+ applications in scope for migration
  - Phased migration approach with ~10% in progress and completed
  - ~100 RCP plugins
  - Coming from Amazon ECS, legacy Kubernetes, Mesos + Marathon, Amazon EC2 environments

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How about you?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Tech deep(er) dives

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cluster(s) per team or multi-tenancy?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cluster(s) per team or multi-tenancy?

What didn't work out?

- Immature access controls and Murphy's Law
- Only solving for K8s RBAC
- Contributors considered a single, trusted platform tenant
- Not catering for the 10–20% of particularly demanding applications

Course correction . . .

- Investment in Gatekeeper policies, improved RBACs, exception approval process
- ABAC in the account, consistent with K8s RBAC configuration
- Trend toward marketplace-like model with additional use-case-specific cluster provisioning (e.g., data)
- Created DAWG and offering alternative compute experiences for Amazon EC2 and AWS Lambda

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Control plane and Cluster Federation

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Control plane and Cluster Federation

What didn't work out?

- Not proactively testing the platform at the projected scale and beyond
- Not allowing for controlled configuration drift
- "Deploy failed" considered good enough feedback from the pipeline!
- KubeFed

Course correction . . .

- Fully understand needs, simulate scaling the platform, identify, and overcome bottlenecks
- Consistency is important, but controlled configuration drift (e.g., CRD versions) is needed for safe updates
- Build for providing high quality, interpreted feedback to users
- Considering options (Karmada, Open Cluster Management, etc.)

aws © 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Providing more than compute with Crossplane

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Providing more than compute with Crossplane

What didn't work out?

- Bundling Crossplane resources into application helm charts
- Hiding Crossplane resource complexity using helm templating
- Inconsistent developer experience for internal resources versus AWS
- Forcing migration to Crossplane

Course correction . . .

- Split the lifecycle for infrastructure and application deployment
- GitOps may be a better fit for infrastructure
- Crossplane already has the answer: Create Compositions to add a layer of opinion and bundling over raw AWS resources (e.g., S3+Role+Policy)
- Create custom Crossplane providers for internal services (e.g., datastores)
- Offer both AWS CloudFormation and Crossplane

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Handling compute needs with Karpenter

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Handling compute needs with Karpenter

What didn’t work out?

- Central team acting as a custodian of node groups in multi-tenant clusters
- Not catering for some applications isolation needs

Course correction . . .

- Use AWS Karpenter to give applications direct access to the compute infrastructure they need
- Allow for dedicated provisioners to be created (with lightweight governance!)

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migration support is a first-class feature

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migration support is a first-class feature

What didn't work out?

- Build it and they ~~will~~ won't come
- Not catering for the safe and incremental migration of traffic from legacy to new compute platform

Course correction . . .

- Identify the most painful migration "Jobs To Be Done" and, where possible, address them as first-class platform features
- Out of all migration jobs, this one is near universal and can be complex – provide a mechanism for safely ramping traffic!

Health warning:
You may need to work your legacy environments too!

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Key takeaways

Building a platform is hard!

- Be prepared to get things wrong, accept feedback, and course-correct

The earlier you understand the scale of what you're building, and account for it in your architecture, the better

- Also applies to people and teams!

Temporary features that assist migrating applications from legacy to your new platform are equally important features of your platform

Carefully consider how you will protect the core platform from tenants and tenants from each other

- It's always easier to add user permissions than take them away later!

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Dan Gorst
Email: dgorst@expediagroup.com
LinkedIn: dangorst
Twitter: dangorst

Ben Fields
Email: Benjaws@amazon.com
LinkedIn: benjaminfields22
Instagram: potatofields22
Twitter: PotatoFields



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.