CLOUD NATIVE
COMPUTING FOUNDATION

Building a Kubernetes-native control plane with Crossplane

Cloud Native Kuala Lumpur, Feb 2026 Meetup

Vasil Shaikh

Senior DevOps Engineer @NTT Data

About Me

Vasil Shaikh
DevOps / Platform Engineer

- Works on Kubernetes-based platforms and cloud infrastructure
- Hands-on with **Crossplane**, **Backstage**, and AWS in real-world environments
- Focused on platform engineering, self-service infrastructure, and reducing operational friction for teams
- Enjoys turning complex infrastructure problems into simple, repeatable workflow
- https://www.linkedin.com/in/vasilshaikh/

Agenda

- The Problem
- The Shift: Platform Engineering
- What is Crossplane
- Core Concepts
- How It Works
- Example: How does a claim look like?
- Live Demo: Provisioning a Database
- What’s Next: Crossplane v2
- Q&A

The Problem

Scenario: “I just need a database”

- Developer needs a PostgreSQL DB for a new service
- Raises a ticket with infra requirements
- Back-and-forth on size, networking, security
- Click Ops or IaC provisioning by ops
- Credentials shared separately
- Drift & inconsistencies appear later

Result:

❌ Slow delivery
❌ Snowflake infrastructure
❌ Ops becomes the bottleneck

The Shift: Platform Engineering

Platform Engineering isn’t a new tool — it’s the discipline of building and maintaining **Internal Developer Platforms (IDPs)** that enable self-service infrastructure.

- **Self-service:** Developers provision infrastructure without relying on Ops or Platform teams
- **Golden paths:** Standardized, well-tested workflows that teams can trust
- **Abstraction:** Developers consume infrastructure through simple APIs instead of learning cloud tools
- **Platform as a product:** Built-in security, faster time-to-market, and a consistent developer experience — where platform teams build the product and developers are the users

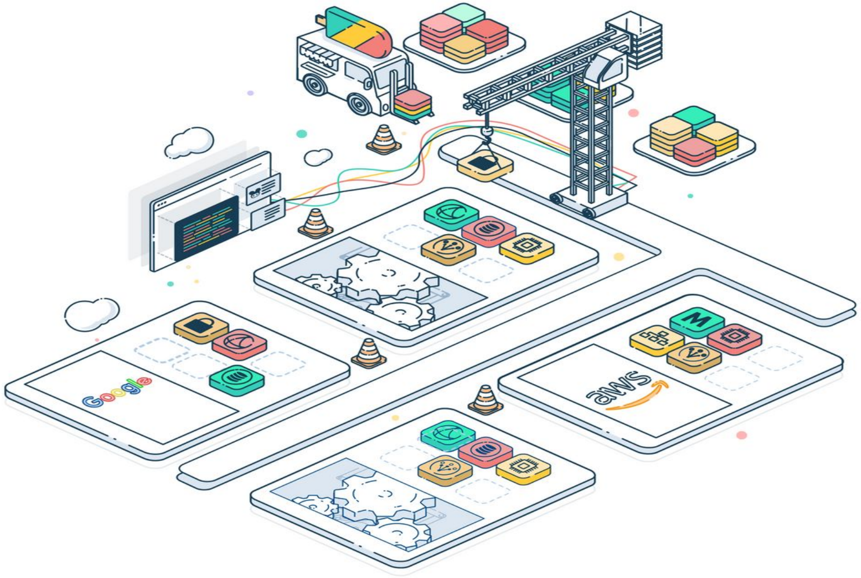
The Developer Experience We Want

Now imagine if instead of raising a ticket, a developer could just say:

“I want a database, in this environment, with these guardrails.”

And the platform takes care of the rest.

This is where Crossplane enters the picture.



What is Crossplane?

Crossplane Is the Cloud-Native Framework for Platform Engineering.

Simply put it’s an open-source K8s add-on that transforms your cluster into a universal control plane to manage infrastructure across cloud providers as well as on-prem environments.

Universal control plane - Single API for managing multi-cloud resources

Declarative approach - Describe desired state, let crossplane handl implementation

Kubernetes-native - Leverages existing K8s skills & tooling

Core Concepts

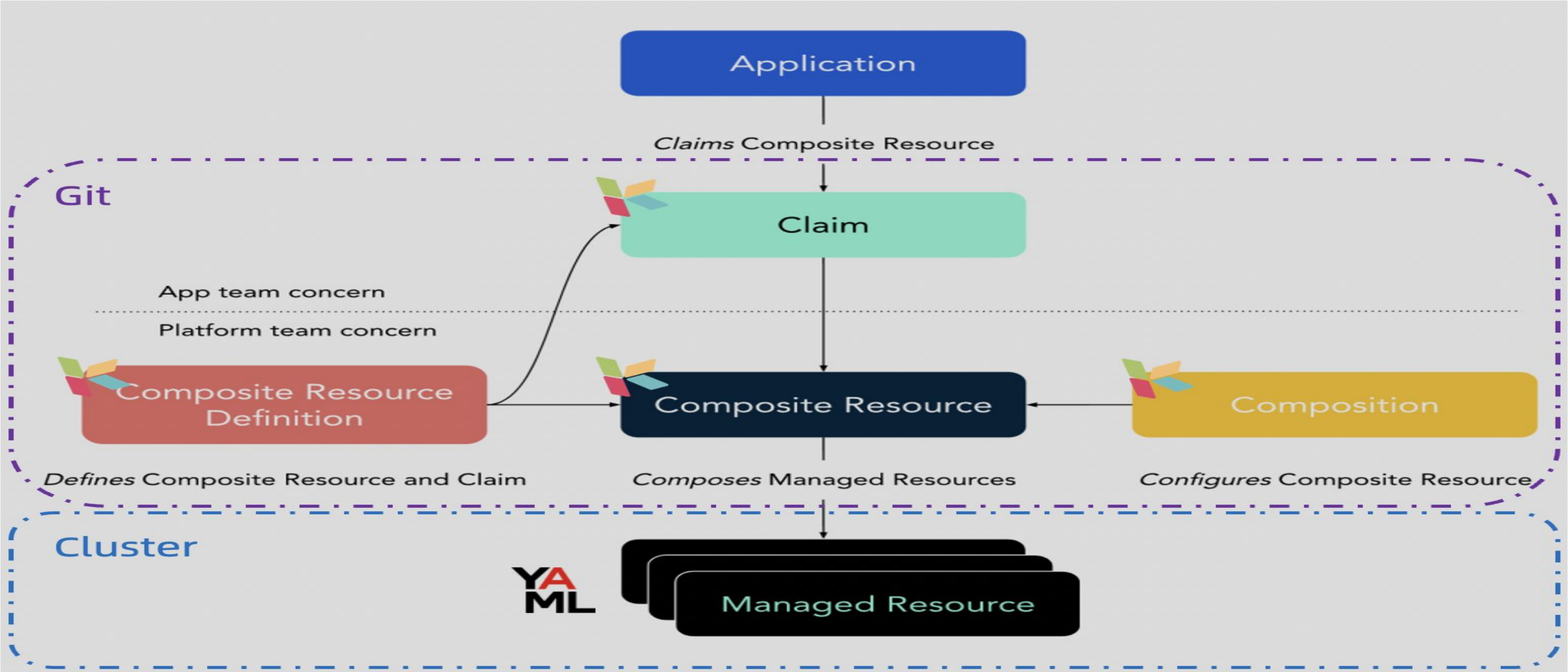
Component	Abbreviation	Scope	Purpose
Provider	-	Cluster	A package (e.g., AWS, Azure, GCP) that installs CRDs and controllers to manage external resources.
ProviderConfig	PC	Cluster	Defines authentication and configuration (credentials, region) for a specific Provider.
Managed Resource	MR	Cluster	A granular Kubernetes object representing a single cloud resource (e.g., an RDS instance or S3 bucket).

Core Concepts (contd.)

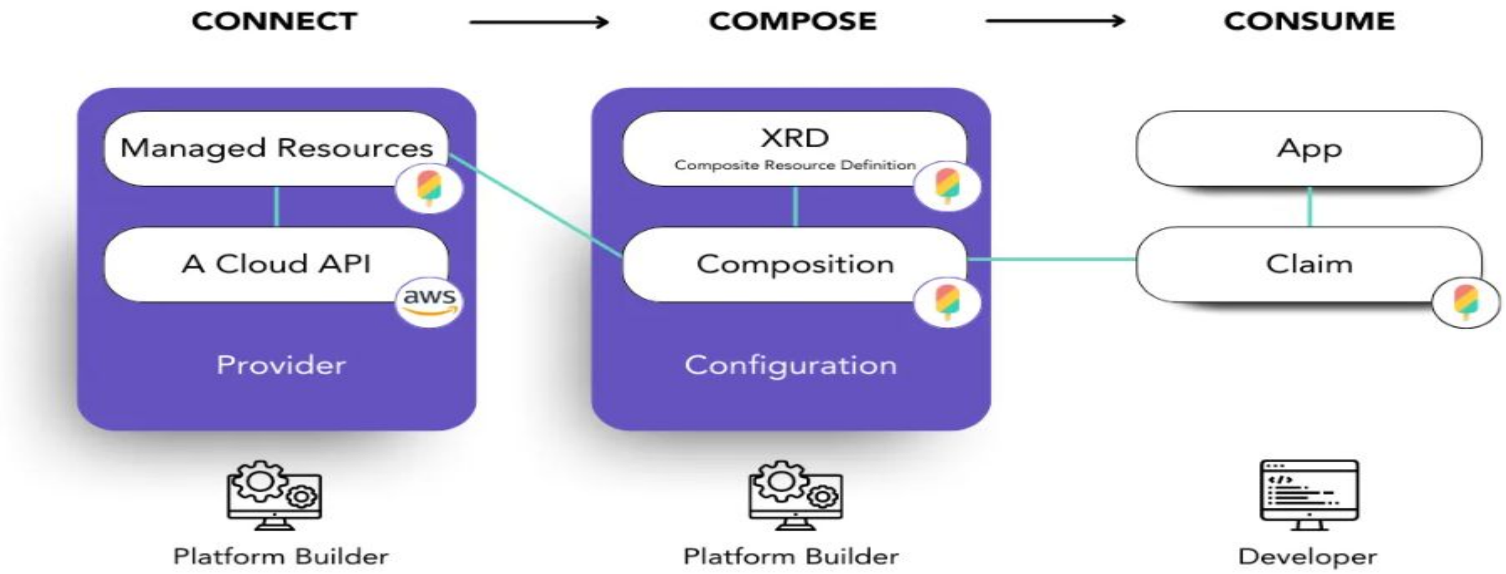
Component	Abbreviation	Scope	Purpose
Composite Resource Definition	XRD	Cluster	The "API Blueprint" that defines the schema for custom, high-level infrastructure types.
Composition	-	Cluster	The "Template" that maps a high-level XR to specific Managed Resources (MRs).
Composite Resource	XR	Cluster	The cluster-scoped instance of a defined infrastructure bundle.

Core Concepts (contd.)

Component	Abbreviation	Scope	Purpose
Composite Resource Claim	XRC / Claim	Namespace	The developer-facing request used to provision infrastructure within a specific namespace.


Application
Claims Composite Resource
Git
Claim
App team concern
Platform team concern
Composite Resource Definition
Composite Resource
Composition
Defines Composite Resource and Claim
Composes Managed Resources
Configures Composite Resource
Cluster
YAML
Managed Resource

How it works



CONNECT → COMPOSE → CONSUME

Managed Resources

A Cloud API

aws

Provider

Platform Builder

XRD

Composite Resource Definition

Composition

Configuration

Platform Builder

App

Claim

Developer

Three steps of building a platform with Crossplane

Example: How does a claim look like?

```
apiVersion: platform.iamvasil.com/v1alpha1
kind: DatabaseClaim
metadata:
  name: dev-app-db
  namespace: dev
spec:
  environment: dev
  size: small
  databaseName: testapp
  writeConnectionSecretsToNamespace: dev
```

Traditional Developer Request

“I need a PostgreSQL database”

Infrastructure Team Response:

- What size and performance requirements?
- Which region and availability zones?
- What networking and security setup?
- Backup and monitoring requirements?
- Connection and credential management?

Timeline:
⏳ 2–3 weeks of back-and-forth conversations

Traditional Developer Request (But now with Crossplane)

“I need a PostgreSQL database”

Platform Response:

```
apiVersion: platform.company.com/v1alpha1
kind: Database
metadata:
  name: my-app-db
spec:
  size: medium
  environment: production
```

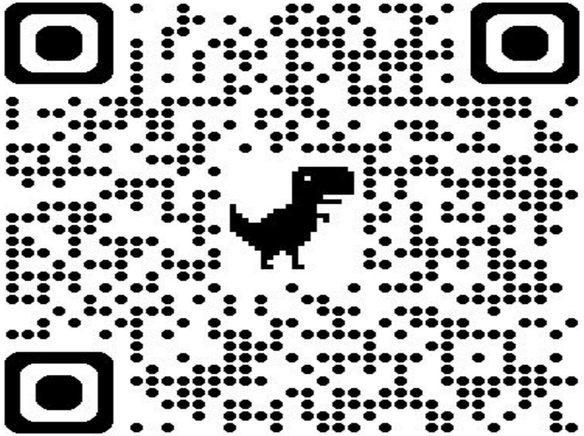
Platform Action:

- Automatically creates VPC, subnets, security groups
- Provisions RDS with approved configurations
- Sets up monitoring and backups
- Delivers credentials securely

Timeline:
⚡ 5–10 minutes, fully automated

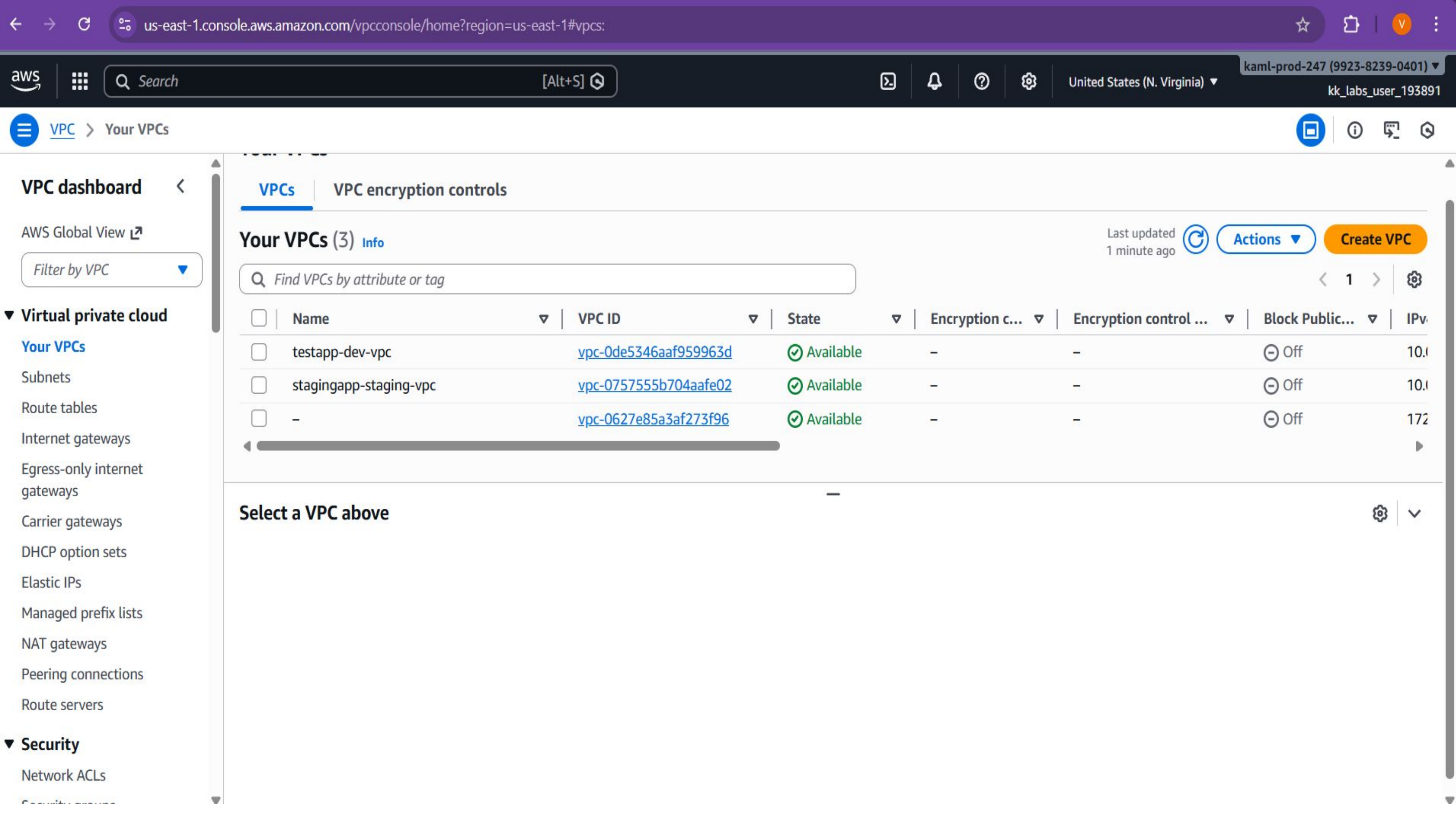
Live Demo: Let the action begin!

Scan the QR for repo link



Live Demo: What could possibly go wrong?

Live Demo (Recording)



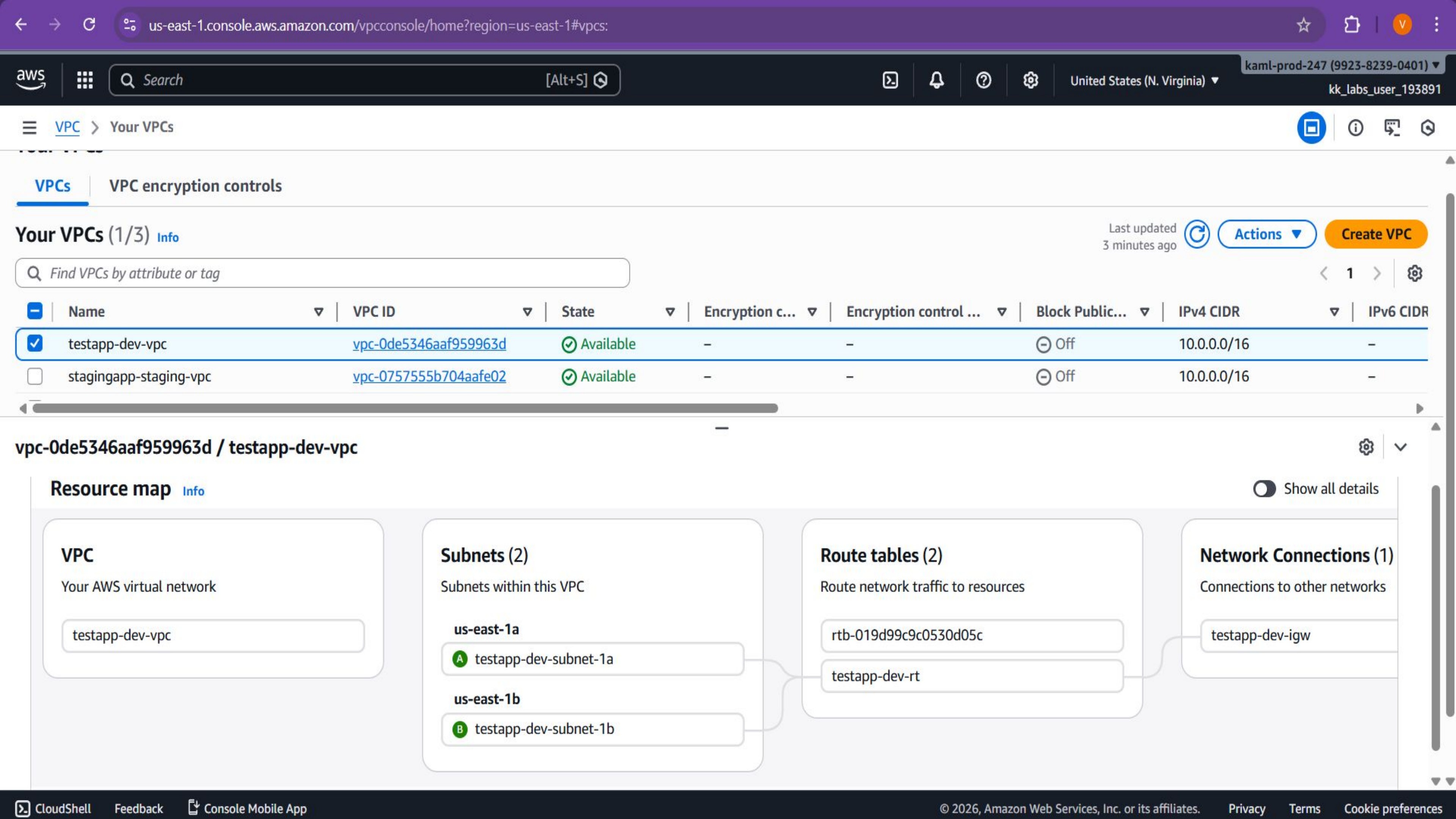
us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#vpcs:

Your VPCs (3) Info

Last updated 1 minute ago

□	Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv
□	testapp-dev-vpc	vpc-0de5346aaf959963d	✓ Available	–	–	⊖ Off	10.(
□	stagingapp-staging-vpc	vpc-0757555b704aafe02	✓ Available	–	–	⊖ Off	10.(
□	–	vpc-0627e85a3af273f96	✓ Available	–	–	⊖ Off	172

Select a VPC above


us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#vpcs:
aws
Search
[Alt+S]
United States (N. Virginia) ▼
kaml-prod-247 (9923-8239-0401) ▼
kk_labs_user_193891
VPC > Your VPCs
VPCs
VPC encryption controls
Last updated 3 minutes ago
Actions ▼
Create VPC
Find VPCs by attribute or tag
1
Show all details
CloudShell
Feedback
Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Your VPCs (1/3) Info

	Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv4 CIDR	IPv6 CIDR
☑	testapp-dev-vpc	vpc-0de5346aaf959963d	✓ Available	–	–	⊖ Off	10.0.0.0/16	–
□	stagingapp-staging-vpc	vpc-0757555b704aafe02	✓ Available	–	–	⊖ Off	10.0.0.0/16	–

vpc-0de5346aaf959963d / testapp-dev-vpc

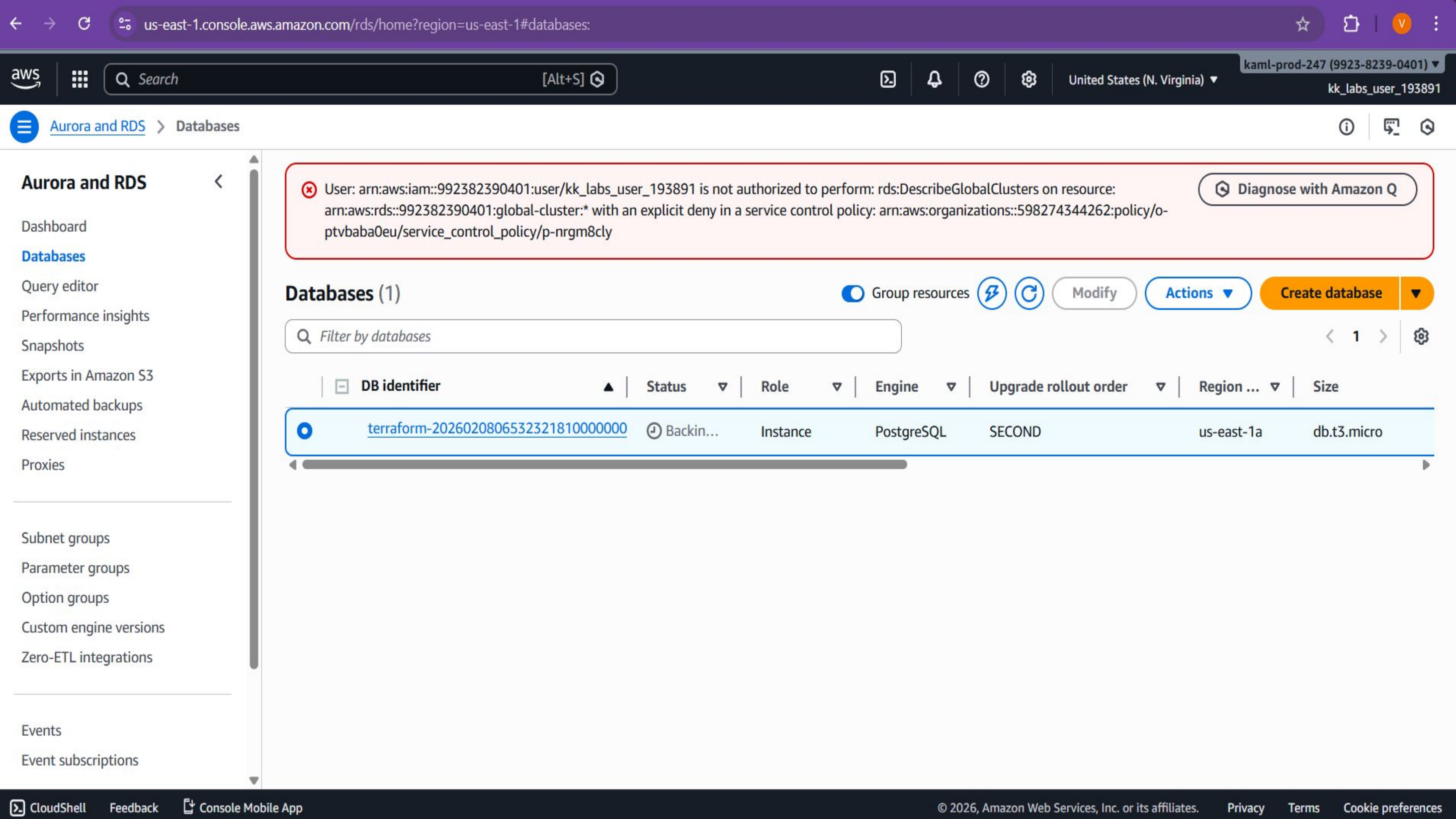
Resource map Info

VPC
Your AWS virtual network
- testapp-dev-vpc

Subnets (2)
Subnets within this VPC
- us-east-1a
  - A testapp-dev-subnet-1a
- us-east-1b
  - B testapp-dev-subnet-1b

Route tables (2)
Route network traffic to resources
- rtb-019d99c9c0530d05c
- testapp-dev-rt

Network Connections (1)
Connections to other networks
- testapp-dev-igw



User: arn:aws:iam::992382390401:user/kk_labs_user_193891 is not authorized to perform: rds:DescribeGlobalClusters on resource: arn:aws:rds::992382390401:global-cluster:* with an explicit deny in a service control policy: arn:aws:organizations::598274344262:policy/o-ptvbaba0eu/service_control_policy/p-nrgm8cly

Databases (1)

	DB identifier ▲	Status	Role	Engine	Upgrade rollout order	Region ...	Size
●	terraform-20260208065323218100000000	Backin...	Instance	PostgreSQL	SECOND	us-east-1a	db.t3.micro

What’s next: Crossplane v2

What’s Changing in Crossplane v2?

Crossplane v2 simplifies how composite resources (XRs) and managed resources (MRs) are structured:

Namespaces by default – Most composite resources (XRs) and all managed resources (MRs) are now namespaced.

Broader composition capabilities – XRs can now compose any Kubernetes resource, not just MRs.

Claims are removed – This reduces complexity and aligns Crossplane more closely with Kubernetes conventions.

These changes make it easier to manage applications and infrastructure in a unified way, without the need for extra cluster-scoped components.

Questions?

Let’s connect!

Resources

- https://www.crossplane.io/
- https://docs.crossplane.io/v2.1/whats-crossplane/
- https://blog.upbound.io/composing-a-platform-by-patching-crossplane-resources
- https://www.kcl-lang.io/blog/2024-03-04-crossplane-marketplace-kcl
- https://blog.crossplane.io/announcing-crossplane-v2-proposal/