TECHNICAL DEEP DIVE

Multi-Cloud Kubernetes Architecture

Architecture your platform team needs to validate. How emma runs one Kubernetes cluster across AWS, GCP, and Azure — and why it works.

- KUBERNETES 1.35.1
- CILIUM CNI / eBPF
- EU CONTROL PLANE
- TERRAFORM PROVIDER

ARCHITECTURE COMPONENTS

Six things your engineers will ask about first

Kubernetes 01

Upstream 1.35.1. No fork. Standard API server, scheduler, etcd. Your existing manifests, Helm charts, and operators work unchanged.

```
kubectl get nodes → nodes across AWS, GCP, Azure
```

Networking 02

Cilium CNI (eBPF) on top of emma's multi-cloud fabric (BGP/VXLAN, Cisco Catalyst 8000v). One network policy layer across all clouds. No VPN stitching.

```
kubectl get cnp → same policies, all clouds
```

Storage 03

Unified emma CSI driver across all clouds. Abstracts EBS, PD, Managed Disks behind one interface. Data stays in-region. No cross-cloud replication unless configured.

```
kubectl get sc → mma-storage (works across AWS, GCP, Azure)
```

Control Plane 04

Hosted in Luxembourg (EU). emma manages etcd, API server, scheduler, monitoring. Full kubeconfig with RBAC. Control plane outage != workload outage.

```
EU jurisdiction · no US hyperscaler in the path
```

IaC / Terraform 05

Official Terraform provider on registry. Cluster and node groups as HCL. CI/CD friendly. emma operates what Terraform creates.

```
resource "emma_kubernetes" "prod" { ... }
```

Node Pools 06

Per-provider node pools in the same cluster. Isolated failure domains. Move workloads between providers with node affinity, not re-architecture.

```
nodeSelector: topology.emma.ms/cloud: aws
```

INFRASTRUCTURE AS CODE

One cluster. Three clouds. Standard Terraform.

```
# One cluster. Three clouds. Standard Terraform.
resource "emma_kubernetes" "production" {
name = "prod-multi-cloud"
worker_nodes = [
{ name = "aws-pool", data_center = "eu-west-1", vcpu = 4, ram_gb = 16 },
{ name = "gcp-pool", data_center = "europe-west1", vcpu = 4, ram_gb = 16 },
{ name = "azure-pool", data_center = "westeurope", vcpu = 4, ram_gb = 16 },
]
}
# terraform apply → 1 cluster, 3 clouds, 3 node pools
# Networking handled by emma. No VPN, no peering.
```

HARD QUESTIONS YOUR ENGINEERS WILL ASK

Honest answers, no marketing.

Latency? 01

Follows cloud-region topology. No forced traffic routing through a central control plane. Cross-cloud latency = typical inter-region latency.

Networking details? 02

Cilium on eBPF. BGP/VXLAN, Cisco Catalyst 8000v. Direct inter-cloud connectivity. No VPN stitching, no provider peering dependencies.

Blast radius? 03

Node pools isolated per cloud provider. Failure in AWS doesn't cascade to GCP or Azure. Control plane outage doesn't affect running workloads.

Debugging? 04

Standard kubectl exec, logs, port-forward. Works identically regardless of which cloud the pod runs on. Prometheus, Grafana, PagerDuty — same endpoints.

K8s version? 05

1.35.1. Upstream. No fork. If it works on EKS/GKE/AKS, it works on emma.

Migration? 06

Namespace-level. kubectl apply your manifests. Keep old clusters live while validating. Typical PoC: 2–4 weeks.

When NOT a fit? 07

Regulations requiring separate control planes per environment, or sub-5ms inter-node latency requirements. We'll tell you honestly on the first call.

READY TO VALIDATE?

Book a 30-min architecture review with our engineers.

We'll review your current setup and show exactly where emma fits — or tell you honestly if it doesn't.

k8s.emma.ms →