# Achieving Hybrid-Cloud Elasticity through Dynamic Cloud Bursting and Federated Kubernetes

IRIS Kubernetes Federation Project

Manu Antony

# To show a successful Proof of Concept for Multi-Cloud Workload Bursting

> This pilot project has to demonstrate a federated Kubernetes architecture bridging on-premise infrastructure and public cloud resources. By leveraging advanced orchestration and multi-cluster networking, the project needs to achieve automated workload distribution and bursting capabilities.
>
> The system needs to prove being capable of maintaining service continuity and scaling dynamically based on real-time resource availability, successfully meeting all required pilot goals.

George Beckett
Amy Krause
Mark Holliman
Manu Antony

# Meeting the Primary Goals of the IRIS Federation Pilot

The pilot successfully delivered on all required project objectives.

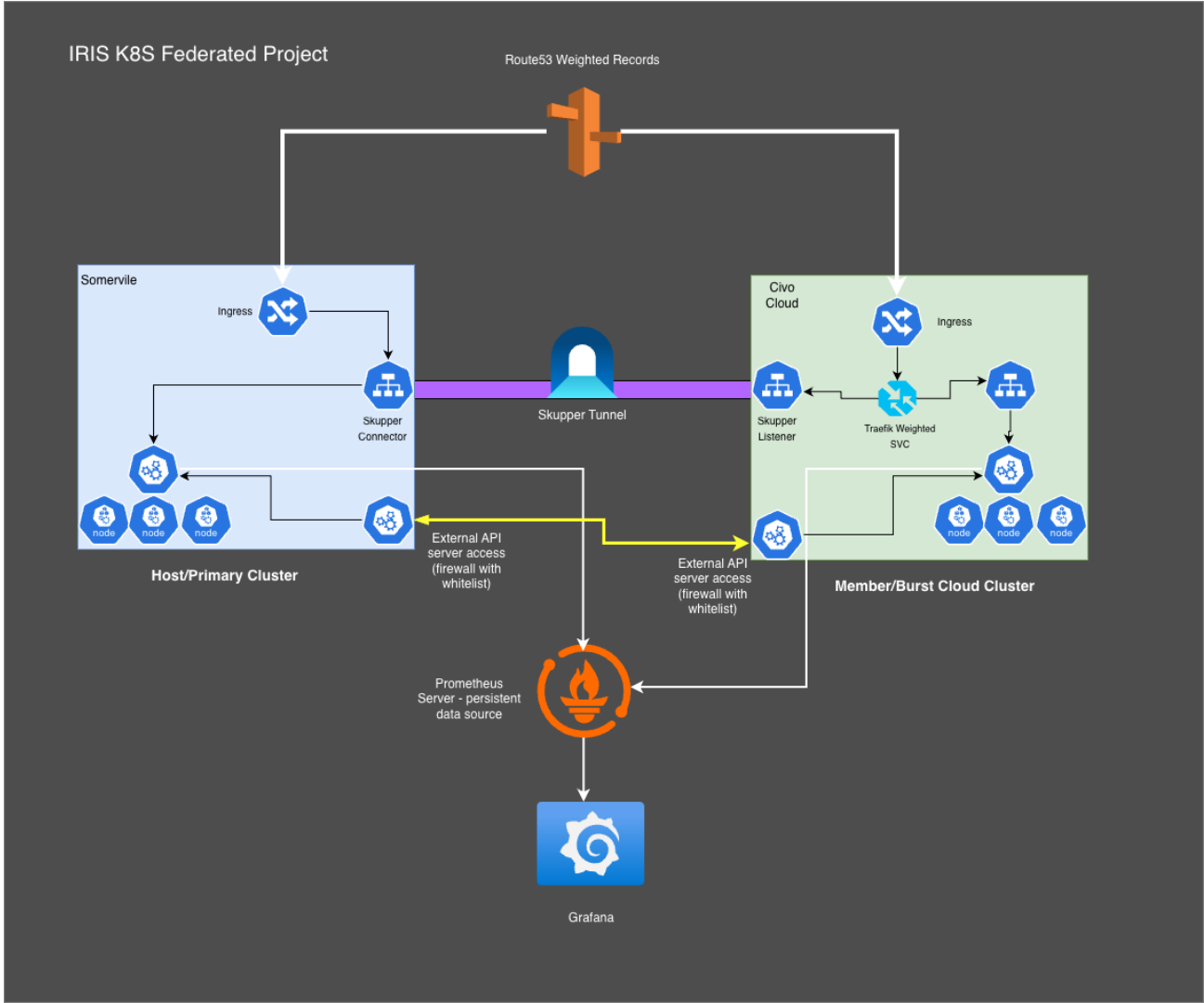
**01** Infrastructure: **Set up a federated Kubernetes cluster using Infrastructure as Code.**

**02** Observability: **Provisioned a complete observability stack utilizing Prometheus and Grafana.**

**03** Deployment: **Successfully deployed and routed a distributed workload across the federated cluster.**

**04** Validation: **Collected actionable data validating the scalability, distribution, and cost-effectiveness of the architecture.**

# Architecture



# Bridging On-Premise OpenStack with Public Civo Cloud

**The core infrastructure was provisioned using Infrastructure as Code via Terraform and managed through GitLab CI.**

## Host Cluster

**On-Premise Environment**

Deployment Platform

**OpenStack Magnum (Somerville)**

Kubernetes Version

**v1.34.1**

Node Configuration

**3 Nodes | 4 vCPUs | 16GB RAM | 100GB Storage**

## Member Cluster

**Public Cloud Environment**

Deployment Platform

**Civo Cloud**

Kubernetes Version

**v1.34.1-k3s**

Node Configuration

**3 Nodes | 4 vCPUs | 16GB RAM | 100GB Storage**

# Civo for Edinburgh University

*Cloud, the way you want it. Sovereign UK infrastructure, built for AI.*

## UK Data Sovereignty & Compliance

- 100% UK-hosted regions (London LON1, Swindon LON2) plus Frankfurt, NYC, Phoenix, Mumbai
- Certified for UK Government workloads: G-Cloud, Crown Commercial Service, Cyber Essentials Plus
- ISO 27001 and SOC 2 accredited platform
- CivoStack Enterprise option keeps data fully on-premises, on your own hardware

## AI & GPU Capabilities

- Available today: NVIDIA A100, H100, B200, B300 and GB300 Blackwell generations
- Coming Q1 2027: NVIDIA Vera Rubin
- GPU acceleration across public cloud, private cloud and on-premises
- relaxAI: Civo's privacy-first, UK-hosted AI platform for open-source LLMs

## Civo Public Cloud (Edinburgh's current service)

- Fully managed, CNCF-certified Kubernetes, compute instances and Cloud GPUs
- NVMe volumes, managed databases and S3-compatible object storage
- Roughly 60% lower cost than AWS, Google and Microsoft for equivalent Kubernetes clusters
- Free, unlimited data transfer and free control plane for predictable billing

## Civo Private Cloud (for when you need total control)

- CivoStack Enterprise: the same platform, deployed on Edinburgh's own hardware or a dedicated region
- 100% feature parity with public cloud, using the same dashboard, API, CLI and Terraform provider
- Fully managed by Civo: remote monitoring, updates and security patching (ZeroOps)
- Ideal for research data residency, custom compliance or air-gapped requirements

Ready for UK Sovereign Workloads: ISO 27001 | HM Government G-Cloud | Crown Commercial Service | Cyber Essentials Plus | SOC 2

# Leveraging Karmada for Dynamic Workload Distribution

*Karmada serves as the central control plane, enabling multi-cluster management without requiring application code changes.*

Resource Modeling: **Utilizes metrics adapters to make scheduling decisions based on actual available replicas.**

Propagation Policy: **Implemented a weighted division preference based on dynamic available replicas.**

Automated Spillover: **As the host cluster reaches saturation, the scheduler automatically pushes excess workloads to the member cluster/s.**

# Overcoming Firewall Constraints with Skupper Layer 7 Service Interconnect

Initial Architecture Constraint: UDP Ports Blocked (VXLAN/IPsec)

**While Submariner was initially considered, strict firewall limitations blocking UDP ports necessitated an alternative connectivity approach.**

## Skupper Layer 7 Interconnect

### Layer 7 Networking

**Skupper was selected to operate over standard port 443 via load balancers, avoiding lower-level routing issues.**

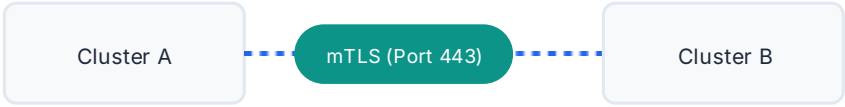
### Secure Connectivity

**Established secure mTLS tunnels to successfully bypass strict campus firewall restrictions.**

## Virtual Routing & Topology

### Service Discovery

**Enabled cross-cluster communication using a connector/listener model without flattening the underlying network.**



# Unified Entry Point via Traefik and Route53

## The ingress architecture provides a unified, secure entry point into the federated environment.

### Global DNS

**AWS Route53 utilizes 50/50 weighted A records to distribute traffic equally across cluster endpoints.**

### Ingress Controller

**Independent Traefik instances deployed on both clusters manage public traffic via dedicated cloud load balancers.**

### Security

**SSL certificates are managed by cert-manager using DNS-01 validation, with wildcard certificates shared between clusters.**

# Compute-Intensive Matrix Multiplication for Scaling Evaluation

To accurately test scaling behaviors, a custom workload was designed to simulate high CPU utilization.

### Application Profile

**A Python Flask application simulating a notebook cell performing CPU-intensive matrix multiplication to trigger rapid auto-scaling actions.**

### Resource Limits

**Configured with** 200m CPU **and** 128Mi Memory **requests. This layout ensures fast, predictable scaling actions under simulated utilization spikes.**

### Tracking Capabilities

**Dynamic environment variables monitor and map physical pod placement during aggressive scaling waves:**

Host Node Mapping

```
ENV: NODE_NAME
```

Pod Replica Identifier

```
ENV: POD_NAME
```

Target Infrastructure Cluster

```
ENV: CLUSTER_NAME
```

# A Four-Phase k6 Performance Profile

Testing Environment

**Performance testing was executed using k6 on a dedicated large testing node (8 vCPU, 32GB RAM) to simulate realistic traffic spikes.**

01

### Warmup Phase

75 Virtual Users for 3 minutes to establish baseline metrics.

02

### Ramp Phase

Scaling from 75 to 150 Virtual Users over 3.5 minutes.

03

### Sustain Phase

Holding at 150 Virtual Users for approximately 2 minutes.

04

### Cooldown Phase

Ramping down to 0 Virtual Users over 2.5 minutes.

# Results

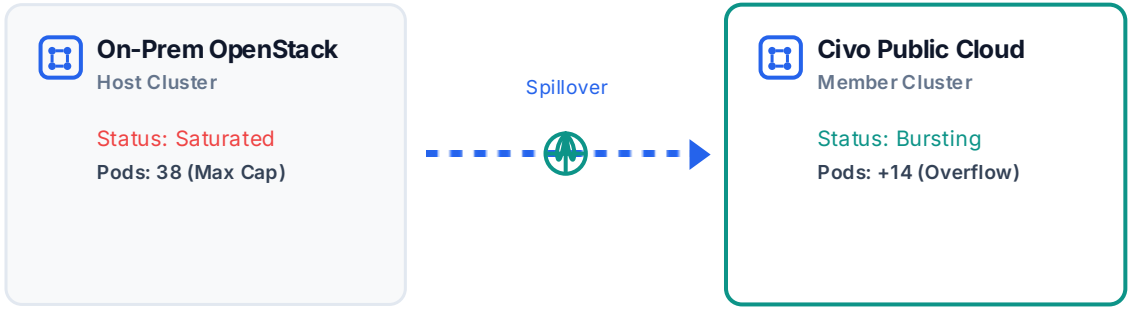

SOMERVILE - number of pods
40
35
30
25
20
15
10
11:33:00
11:34:00
11:35:00
11:36:00
11:37:00
11:38:00
11:39:00
11:40:00
11:41:00
11:42:00
11:43:00
11:44:00
11:45:00
11:46:00
11:47:0
Name
Last *
13
CIVO - number of pods
14
12
10
8
6
4
2
11:33:00
11:34:00
11:35:00
11:36:00
11:37:00
11:38:00
11:39:00
11:40:00
11:41:00
11:42:00
11:43:00
11:44:00
11:45:00
11:46:00
11:47:0
Name
Last *
3

# Automated Workload Spillover during Saturation

The data logs confirmed successful automated workload distribution during peak pressure.

- Baseline: **Started with 10 workload pods on the host and 0 on the member cluster.**
- Saturation Point: **The host cluster reached maximum capacity, peaking at 38 pods.**
- Dynamic Bursting: **The Civo member cluster automatically scaled up to 14 pods to handle the overflow traffic.**
- Recovery: **Both clusters successfully scaled back down to baseline as the simulated load decreased.**

Hybrid-Cloud Workload Spillover




**38 Pods**
Host Peak Saturation

**14 Pods**
Civo Member Burst

# Stable Performance and Low Failures Under Pressure

**The federated architecture maintained strong performance metrics despite the aggressive scaling events.**

## 22,748

Total Requests

Processed 22,748 requests throughout the test duration.

## 106 reqs/s

Throughput

Achieved approximately 106 requests per second.

## 2.84s

Latency

Maintained an average duration of 2.84s, with a 95th percentile latency of 2.6s under maximum pressure.

## 3.47%

Failure Rate

Exhibited a low failure rate of only 3.47% during the extreme bursting phase.

# Cost-Effective Bursting Minimizes Public Cloud Spend

The pilot demonstrated a highly efficient financial model for hybrid cloud operations.

**$163**

TOTAL EXPENDITURE

**The total cost for the 2-week duration was approximately $163.**

**$17/day**

DAILY AVERAGE

**Equates to roughly $17 per day for the entire infrastructure.**

FINANCIAL EFFICIENCY

**Proves the viability of cloud bursting, where expensive public cloud resources are strictly utilized only during peak transient loads.**

# Strategic Recommendations for Production Graduation

To move this architecture into a production environment, several enhancements are recommended.

## Security

**Transition secret management to a centralized solution, such as Hashicorp Vault, to automate certificate renewals.**

## Advanced Scheduling and Autoscaling

**Optimize the Karmada scheduler-estimator and metrics adapter configurations directly within the Helm charts.**

**Also provision autoscaling in cloud**

## Future Capabilities

**Pursue stretch goals including multi-tenancy setups, GPU workload support, and priority-based workload management.**

# Dynamic Cloud Bursting Provides a Viable Path to on Demand Scalability

Pilot Verified

The IRIS Federation pilot confirms that hybrid-cloud elasticity is technically feasible and financially reasonable.

**By successfully integrating OpenStack with Civo Cloud via Karmada and Skupper, we were able to automatically absorb traffic spikes without running into constraints for our internal hardware.**

This architecture ensures high availability, secure networking, and strict cost controls for future distributed workloads.