



Moving to Cloud Run?

Leave your Kubernetes Luggage Behind



Olga Mirensky

Lead Platform Engineer

GDG DevFest

Brisbane

2025

There is a Page for That!

<https://cloud.google.com/run/docs/migrate/from-kubernetes>

Migrate

An existing web service

From App Engine

From AWS Lambda

From Heroku

From Cloud Foundry

From VMWare Tanzu

From a VM using Migrate to Containers

From Kubernetes

To GKE

Troubleshoot

Introduction

Troubleshoot errors

Local troubleshooting tutorial

Known issues

Samples

All Cloud Run code samples

All Cloud Run functions code samples

Kubernetes deployment

Cloud Run service

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
  namespace: default
  labels:
    app: my-app
spec:
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
        - image: gcr.io/cloudrun/hello
          env:
            - name: HELLO
              value: world
replicas: 3
selector:
matchLabels:
app: my-app
```

```
apiVersion: serving.knative.dev/v1
kind: Service
metadata:
  name: my-app
  namespace: 'PROJECT\_NUMBER'
  labels:
    app: my-app
spec:
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
        - image: gcr.io/cloudrun/hello
          env:
            - name: HELLO
              value: world
```

To Set the Stage

Focus on Cloud Run Services,
not Functions or Jobs

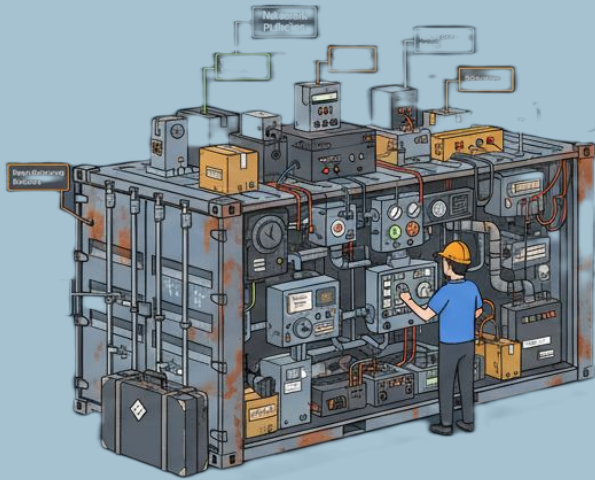
Focus on security and networking

Large scale enterprise migration

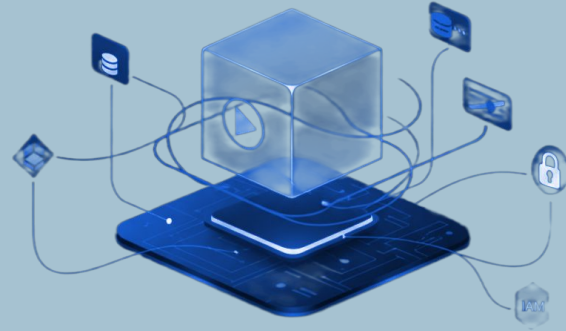
Technical deep-dive and design
principles



Why Cloud Run

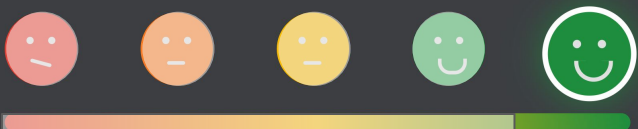


Container as a Service (CaaS)



Platform as a Service (PaaS)

Developer Experience



"Devs love it - drop an image, get the URL back"

"I don't know... just feels more managed"

```
$ gcloud run deploy <SVC_NAME> \  
  --image <IMAGE> \  
  --region <REGION>
```

https://SVC_NAME-PROJ_NUM.REGION.run.app

No Network Environment Segregation

`https://ONE_SVC-dev-proj-num.REGION.run.app`



`https://ANOTHER_SVC-prod-proj-num.REGION.run.app`

Default `.run.app` URL

```
$ dig <SVC>.run.app
```

```
<SVC>.run.app. CNAME v2.run.app.
```

```
v2.run.app.    ... A      34.143.75.2
v2.run.app.    ... A      34.143.72.2
v2.run.app.    ... A      34.143.76.2
v2.run.app.    ... A      34.143.78.2
...
```

[IP ranges that Google makes available to users on the internet](#)

...

```
"ipv4Prefix": "34.32.0.0/11"
```

```
}, {
```

```
"ipv4Prefix": "34.64.0.0/10"
```

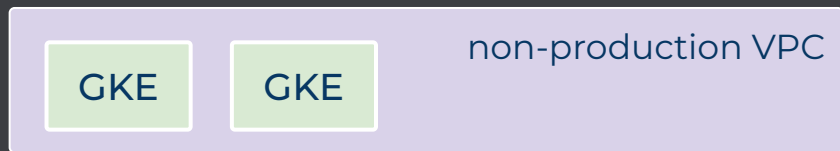
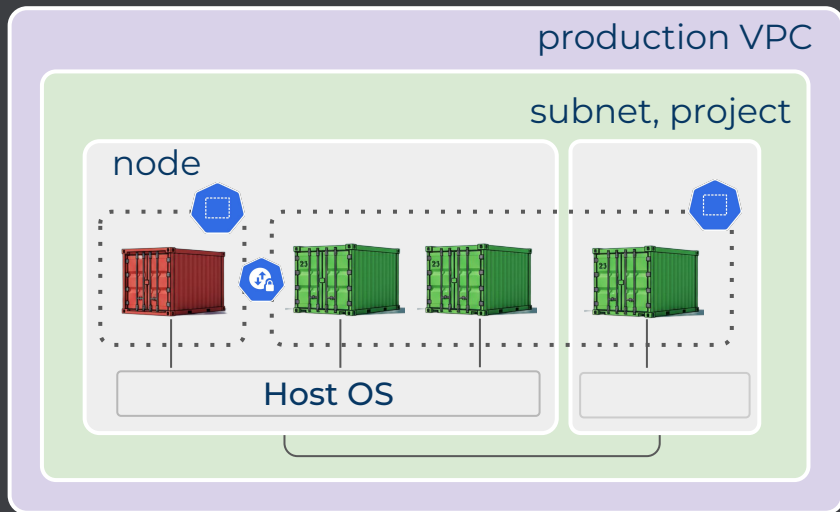
```
}, {
```

```
"ipv4Prefix": "34.128.0.0/10"
```

```
},
```

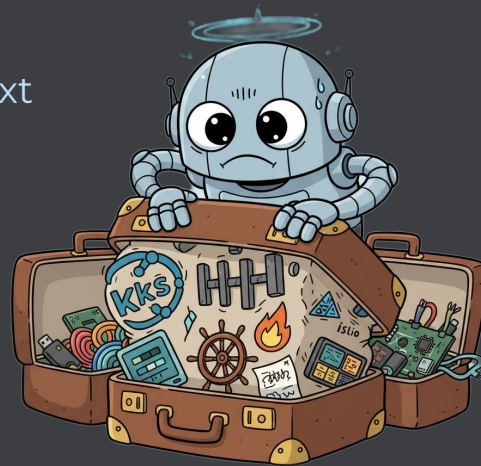
... many, many huge ranges

Kubernetes



-  Namespaces
-  Network policies
-  Nodes
-  RBAC
- Security Context
- Service Mesh
- OPA Policies

...



GKE

Deployment

Namespace

Network Policies

Cloud Run

Service

Project?

Network Tags?

Subnet?

Labels?

I'm Feeling Lucky 

Exercise: Translate Netpols to Firewalls



Even with Internal Ingress

Ingress

☒ Internal
Allow traffic from your project, shared VPC, and VPC service controls perimeter. Traffic from another Cloud Run service must be routed through a VPC. Limitations apply. [Learn more](#)

☐ Allow traffic from external Application Load Balancers

☐ All
Allow direct access to your service from the internet

VPC

Sending traffic directly to the VPC

Network: [cloud-run](#)

Subnet: [subnet-cloud-run-main](#)

Network tags: [allow-egress](#) [all-cloudruns](#)

Traffic routing: route all traffic to the VPC

Even if we could...

The following limits apply to firewall rules:

Item	Limit	Notes
Maximum number of source secure tags per ingress firewall policy rule	256	Applicable only to ingress firewall policy rule—the maximum number of secure tags that you can use as source tags in the firewall rule. This limit cannot be increased.
Maximum number of target secure tags per firewall policy rule	256	Applicable only to firewall policy rule—the maximum number of secure tags that you can use as target tags in the firewall rule. This limit cannot be increased.
Maximum number of source network tags per ingress VPC firewall rule	30	Applicable only to ingress VPC firewall rules—the maximum number of network tags that you can use as source tags in the firewall rule. This limit cannot be increased.
Maximum number of target network tags per VPC firewall rule	70	Applicable only to VPC firewall rules—the maximum number of network tags that you can use as target tags in the firewall rule. This limit cannot be increased.
Maximum number of source service accounts per ingress VPC firewall rule	10	Applicable only to ingress VPC firewall rules—the maximum number of service accounts that you can use as sources in the firewall rule. This limit cannot be increased.
Maximum number of target service accounts per firewall rule	10	The maximum number of service accounts that you can use as targets in a VPC firewall rule or rule in a firewall policy. This limit cannot be increased.

Hard limits due to performance impact

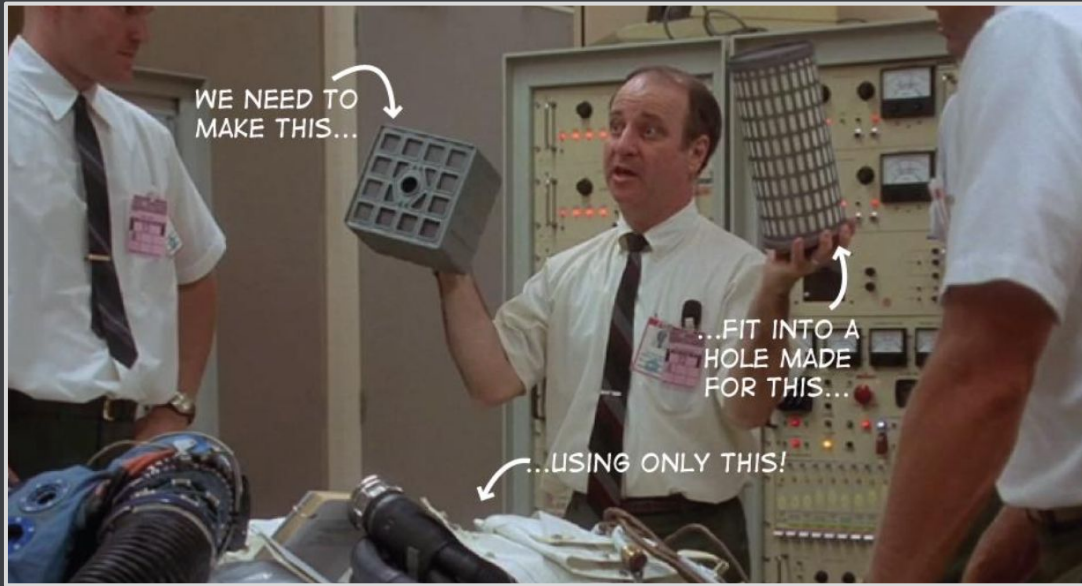
IP-based policies: CIDR only

No Ingress

Permission on shared VPC

Complex at scale

Square Pegs in Round Holes



Granular firewalls:

Don't do it unless
lives depend on it

ROADS?



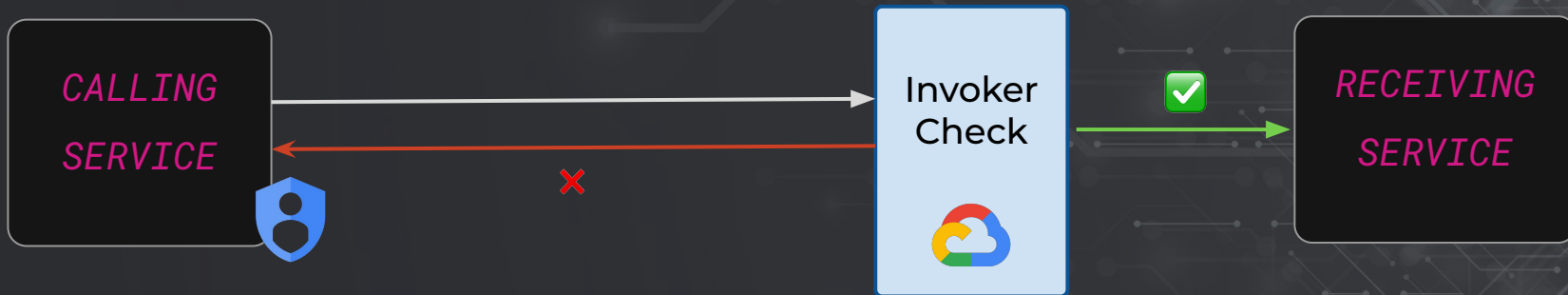
WHERE WE'RE GOING WE DON'T NEED ROADS

We need Identity



Invoker Check

```
$ gcloud run services add-iam-policy-binding RECEIVING_SERVICE \  
  --member='serviceAccount:CALLING_SERVICE_IDENTITY' \  
  --role='roles/run.invoker'
```



Invoker Check



"Bad" Request

Invoker Check

4xx



What We've Learned so Far

Cloud Run ingress can't be secured with network firewall

“Reaching the service” (network)

!=

“Accessing the container” (application)

Are we on the right path?

Security Design Principles

Zero Trust

Adopt "**never trust, always verify**".
Every single request must be authenticated and authorized, regardless of its origin.

Identity as the New Perimeter

Shift focus from "**where**" a request comes from (network location) to "**who**" is making it (verifiable identity).

Identity Aware Proxy (IAP)

Secure Web Proxy (SWP)

Defence in Depth

Establish multiple, independent layers of security controls. Secure the code, edge, runtime, service boundary.

Org Policy

VPC-SC

Cloud Armor

...



Kubernetes is ~~Complicated~~

Amazing

Supports implementation of the above principles

While achieving the main goal - orchestration

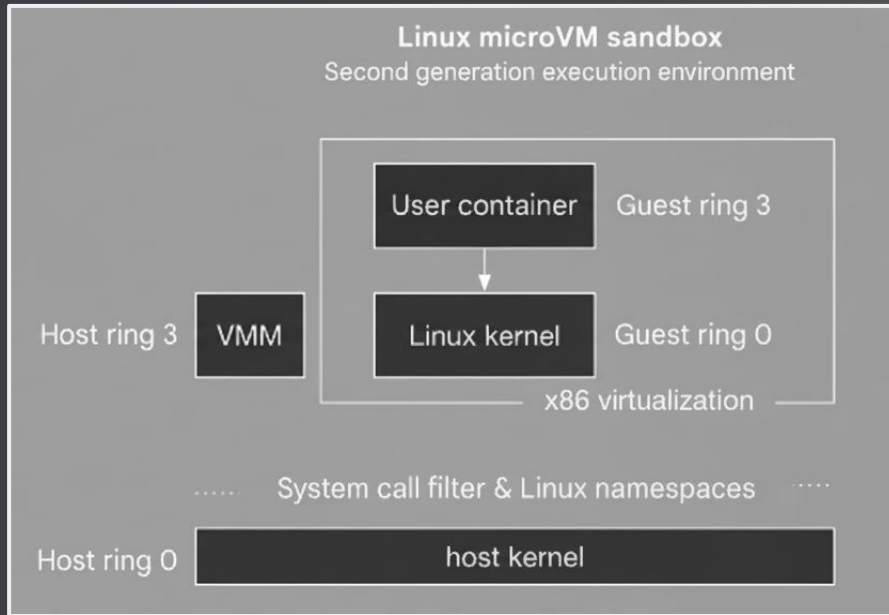
Multi tenanted workloads on shared infra

Same platform to run clusters of 180,000 nodes or on a tiny IoT device

Cloud Run is built on Borg (k8s predecessor)



Execution Environment



Each container in Linux microVM.
Two layers of sandboxing

Syscall filtering

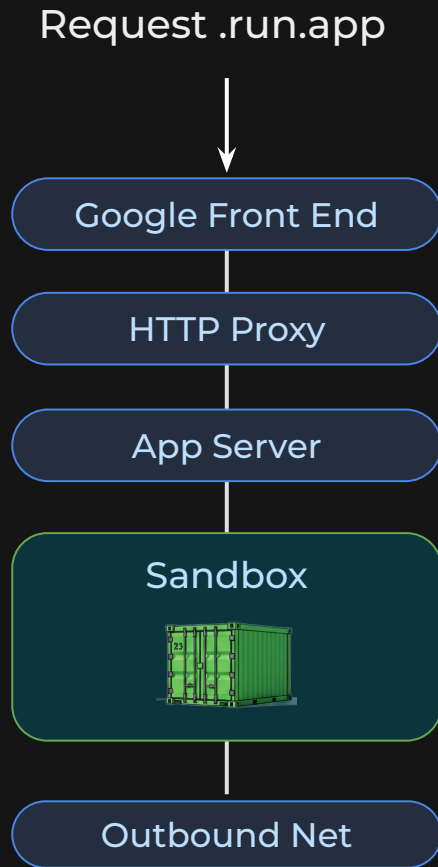
No shared node to escape to

Blast radius is contained to a single,
ephemeral request

Ingress Infrastructure

- Connections are always terminated before reaching container
- GFE mitigates DDoS and other network attacks

```
< HTTP/1.1 403 Forbidden  
< date: Mon, 13 Oct 2025 09:41:43 GMT  
< server: Google Frontend
```



Externally Facing Services

Custom domain

External ALB

Cloud Armor

Disable default `.run.app`

Control DNS name

Use with custom
domain

SSL Policy and
Certificates

WAF at the edge of
Google network

OWASP Top 10

Bot control



Demo

GKE



Cloud Run

So how do we translate?

We don't!

Migration

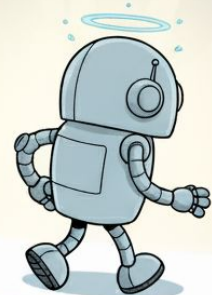
`for_each`
`concept(A) → concept(B)`



Target platform discovery

System design principles

Requirements



Let's Connect!



Olga Mirensky



`olga-mir.github.io`



`github.com/olga-mir`