GCP Server Environments in 40 minutes

Yurii Serhiichuk

Lead SE, TeamDev

TeamDev

Yurii Serhiichuk

Lead Engineer

GCP certified Architect and Data Engineer.

More than 7 years in software development.

Lead engineer at TeamDev.

Wisdom Generator

Generates proverbs with AI.

Built on auto-scalable infrastructure.

Scales to zero.

The problem

Automatic management

You have already coded it,
why do you have to manage it?

Automatic scaling

We have more users,
why isn’t it just working?

Fast deployment

One more day just to deploy it
to our shiny server?

Google Cloud for the rescue

Compute Engine IaaS

Manage virtual servers, not real.

App Engine PaaS

Manage application and services, not the platform.

Cloud Run CaaS

Manage containers.

Cloud Functions FaaS

Manage functions.

Google Compute Engine

Google Compute Engine

Zero-downscale

Any runtime

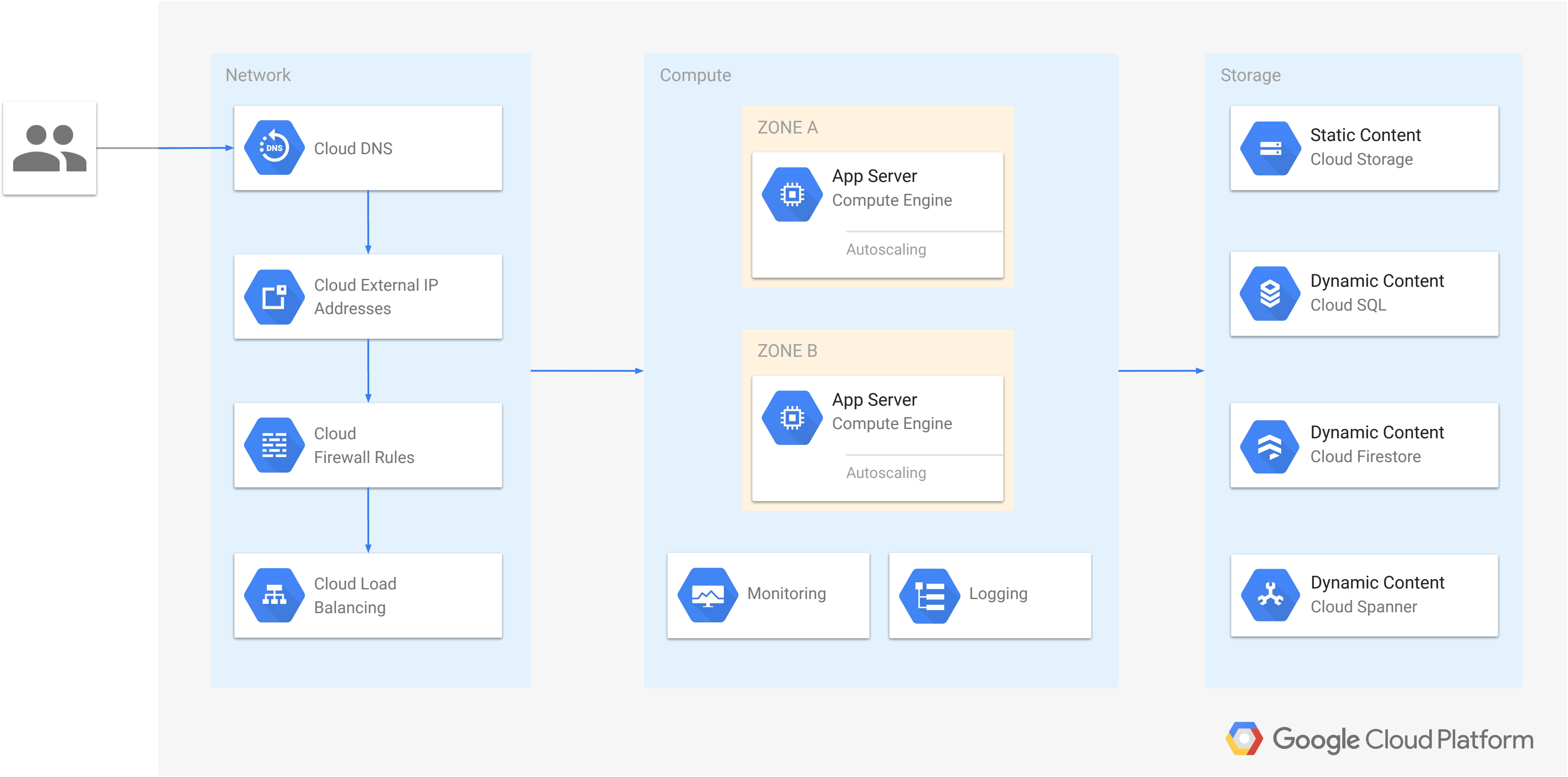
VM Image

Configurable DNS

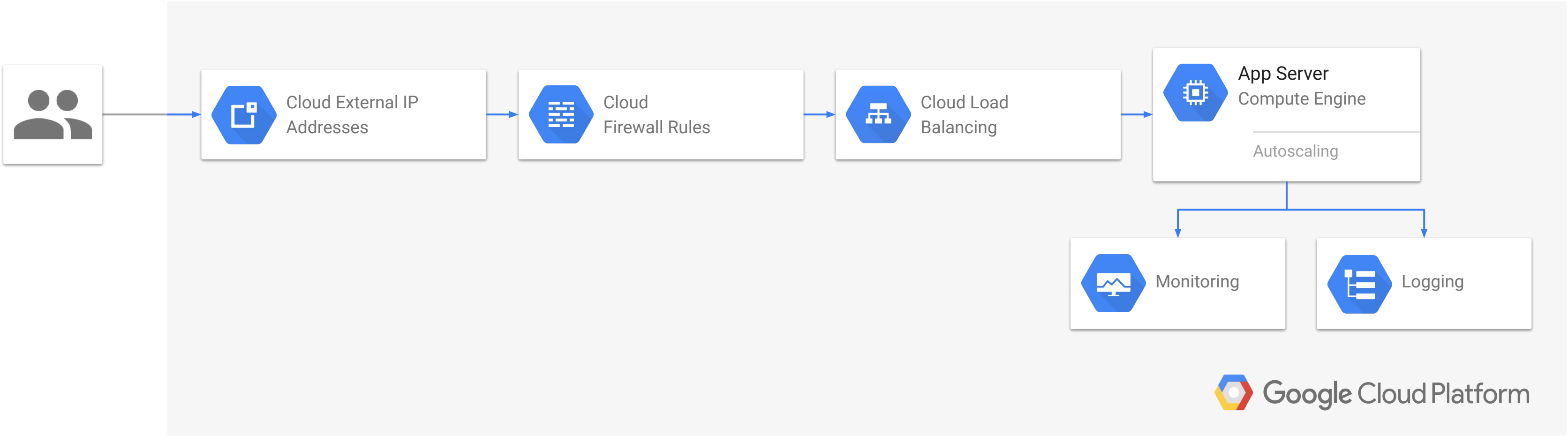
Custom VPC

GPU/TPU

Google Compute Engine



Google Compute Engine



Well Done!

Pros

Most agile and configurable solution

Suitable for custom or resource-intensive workloads.

Any custom resources and networking

GPU/TPU, custom networks, interconnection between on-prem and cloud resources.

Cons

Difficult to set up

Requires a dedicated DevOps or skilled dev to perform the setup.

Prone to errors and misconfigurations

Easy to over- or under-allocate resources or mess up with the security.

Google App Engine

Google App Engine

Zero-downscale

Restricted runtime

Application

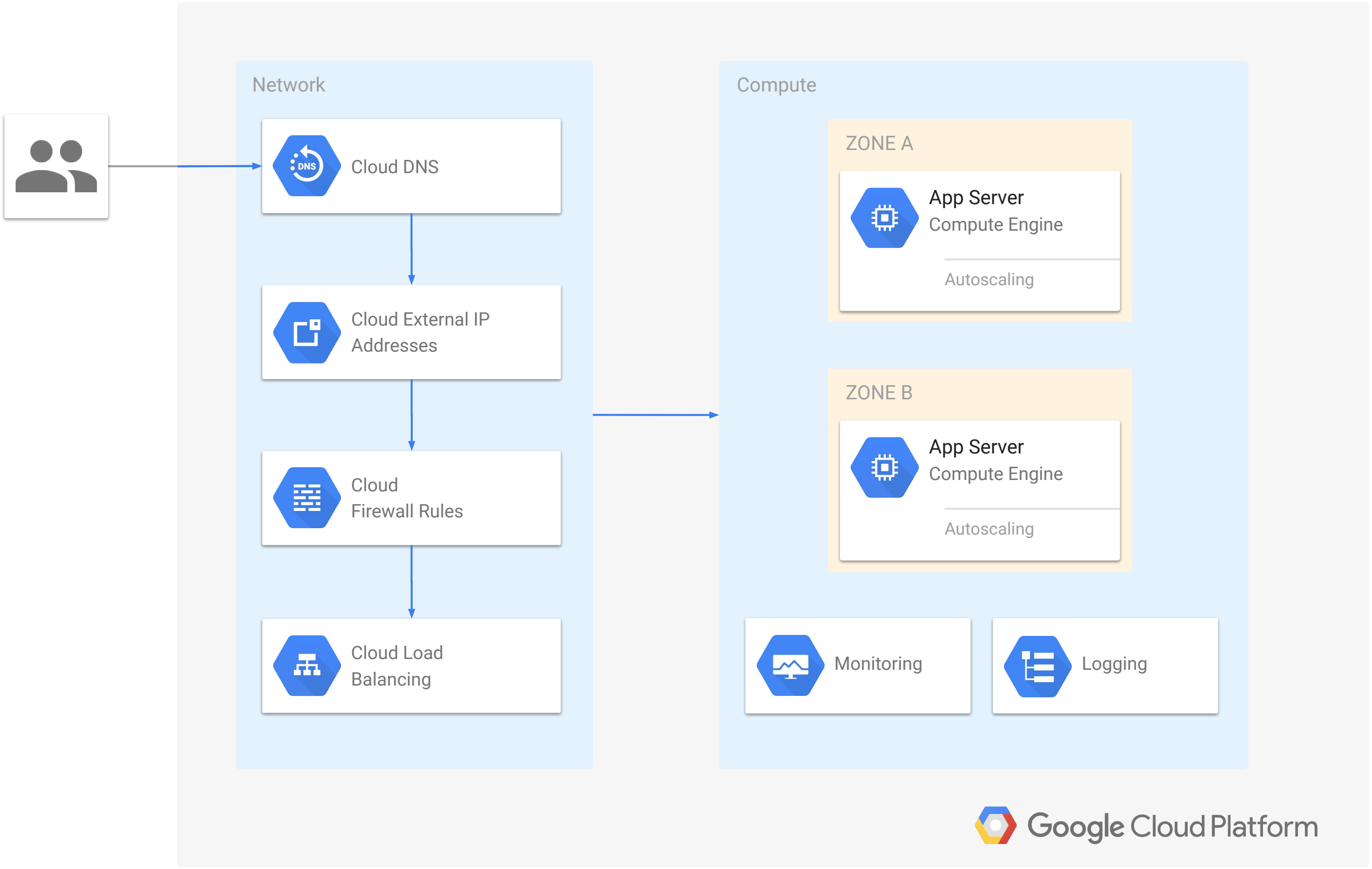
Configurable DNS

Configurable VPC

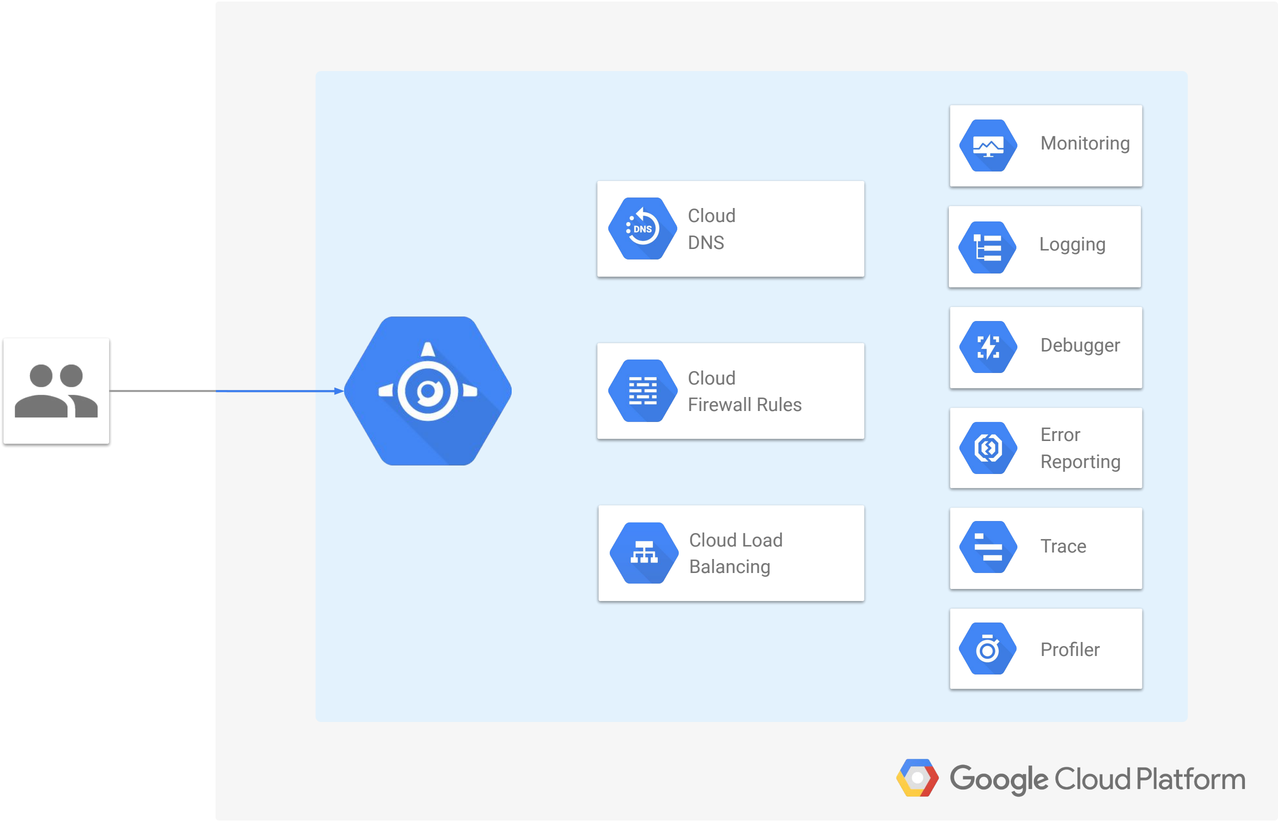
No GPU/TPU

```
gcloud app deploy
```

Google App Engine



Google App Engine



Pros

Batteries included

A platform with a lot of useful services available out-of-the-box.

Easy to manage

Single-command deployment, literally.
Scales up and down automagically.

Cons

Limited resources and restrictions

Up to 2 GB RAM per instance.
Request handling time is tighten up.

May be costly

The autoscaling may go crazy and you’ll have to pay for your errors.

SERVERLESSSSS

Cloud Run

Cloud Run

Zero-downscale

Any runtime

Container

Configurable DNS

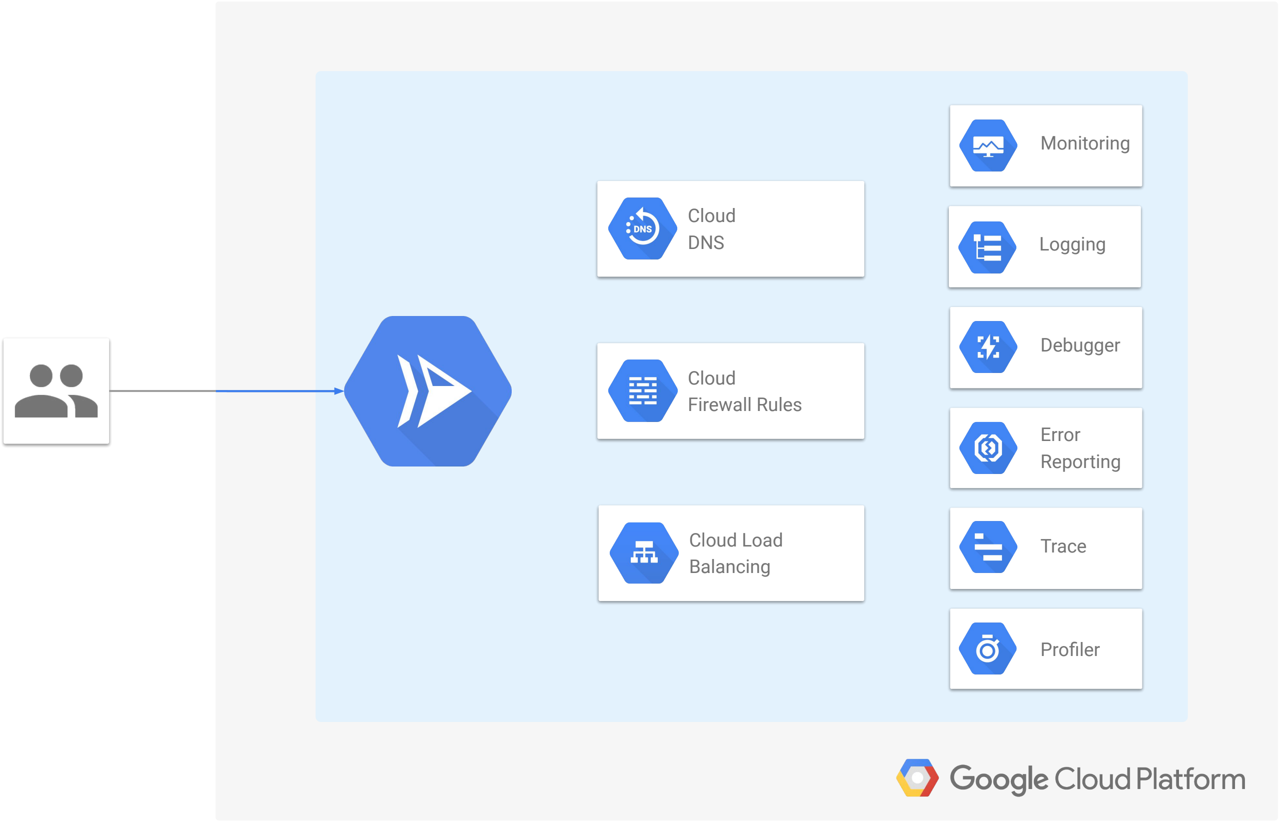
Restricted VPC

No GPU/TPU

```
gcloud builds submit
    --tag="gcr.io/<project-id>/<image>"
```

```
gcloud ~~beta~~ run deploy
    <service-name>
    --image gcr.io/<project-id>/<image>
    --platform managed
    --allow-unauthenticated
```

Cloud Run


Cloud DNS
Cloud Firewall Rules
Cloud Load Balancing
Monitoring
Logging
Debugger
Error Reporting
Trace
Profiler
Google Cloud Platform

Pros

Serverless

No need to manage even platform configs.

Containerized

Reproducible, secure environments.

Cons

Limited resources

2 GB of RAM and 1 vCPU.

Limited capabilities

Only HTTP services.

Cloud Functions

Cloud Functions

Zero-downscale

Restricted runtime

Function

No custom DNS

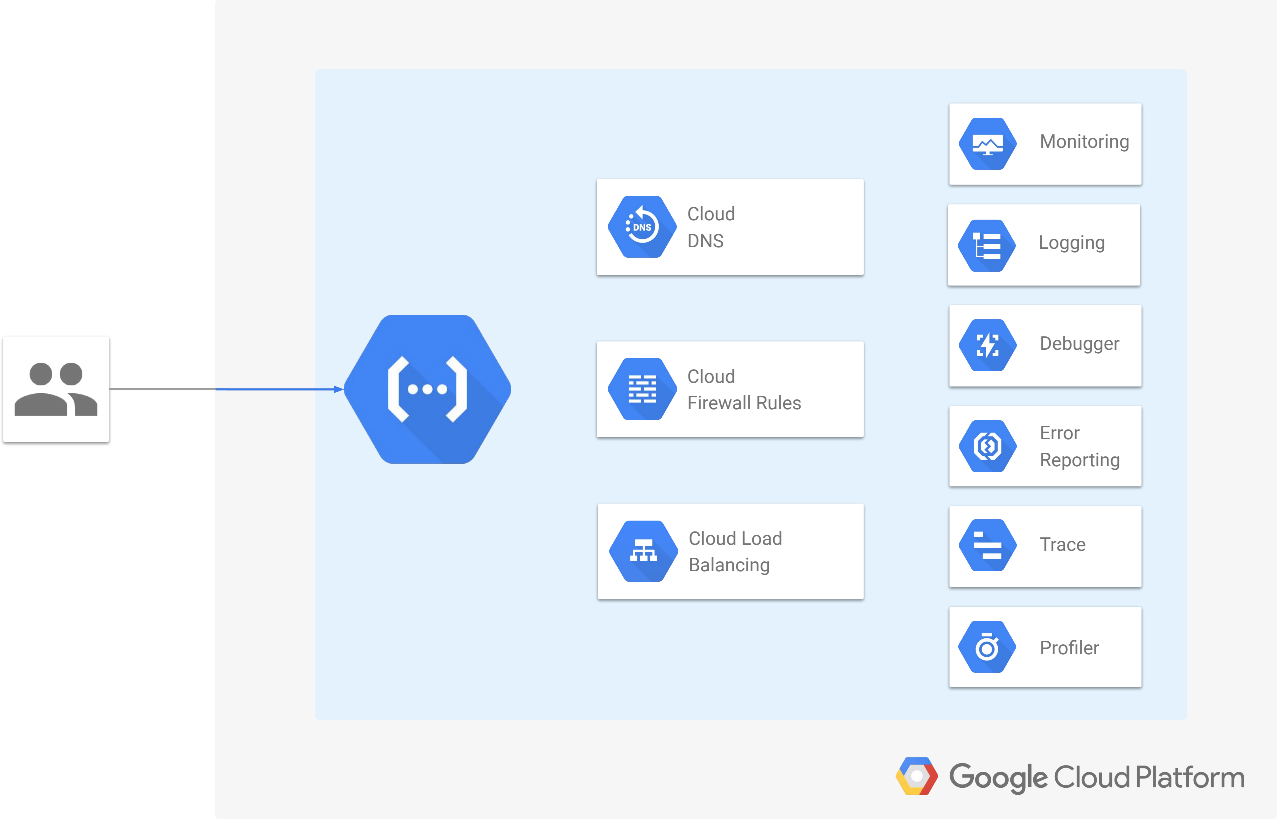
Configurable VPC

No GPU/TPU

Cloud Functions

```
gcloud functions deploy
    <service-name>
    --runtime <runtime>
    --entry-point=<function-name>
    --trigger-http
```

Cloud Functions



Pros

Best time-to-market

New services could be deployed in minutes.

Best inter-service integrations

Firebase, Cloud Storage, PubSub,
Google Analytics, etc.

Cons

Most limited runtime

Only NodeJS, Python and Go.
Java 8/11 in Alpha.

Limited resources

1 vCPU, 2 GB of RAM.
Single request per function.

Conclusion

DNS
文→A

github.com/xSAVIKx/gcp-server-environments

@xSAVIKx

/in/yuriiserhiichuk

yuri.sergiichuk@teamdev.com

Thank You!

TeamDev

	Compute Engine	Cloud Run	App Engine Flex	App Engine Standard	Cloud Functions
Deployment Format	VM image	Container	App or Container	App	Function
Custom URLs	●	●	●	●	
Scale-to-zero	⚙	●		●	●
Free tier	●	●		●	●
Disk persistence	●				
Websockets	●		⚙		
Any runtime	●	●	●		
Request timeout		15 min	60 min	1 min	9 min
Background processes	●		●	⚙	
TPU/GPU	●				
VPC connectivity	●		●	●	●