



## Climbing up the Scaling Ladder

## About me

- I'm Bastian
- Enthusiast about auto scaling  
and automation

```
$ whoami
```

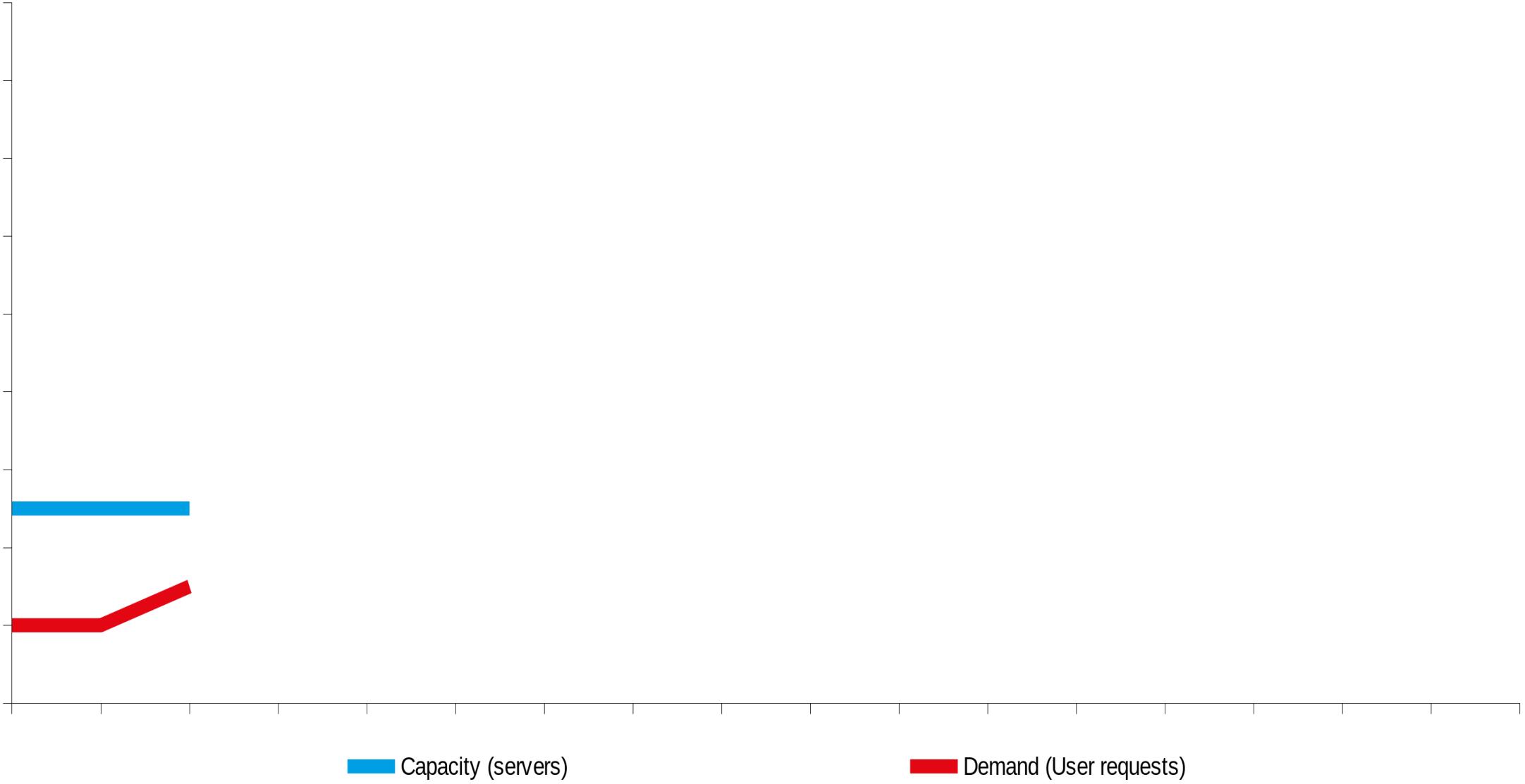
```
I am groot
```

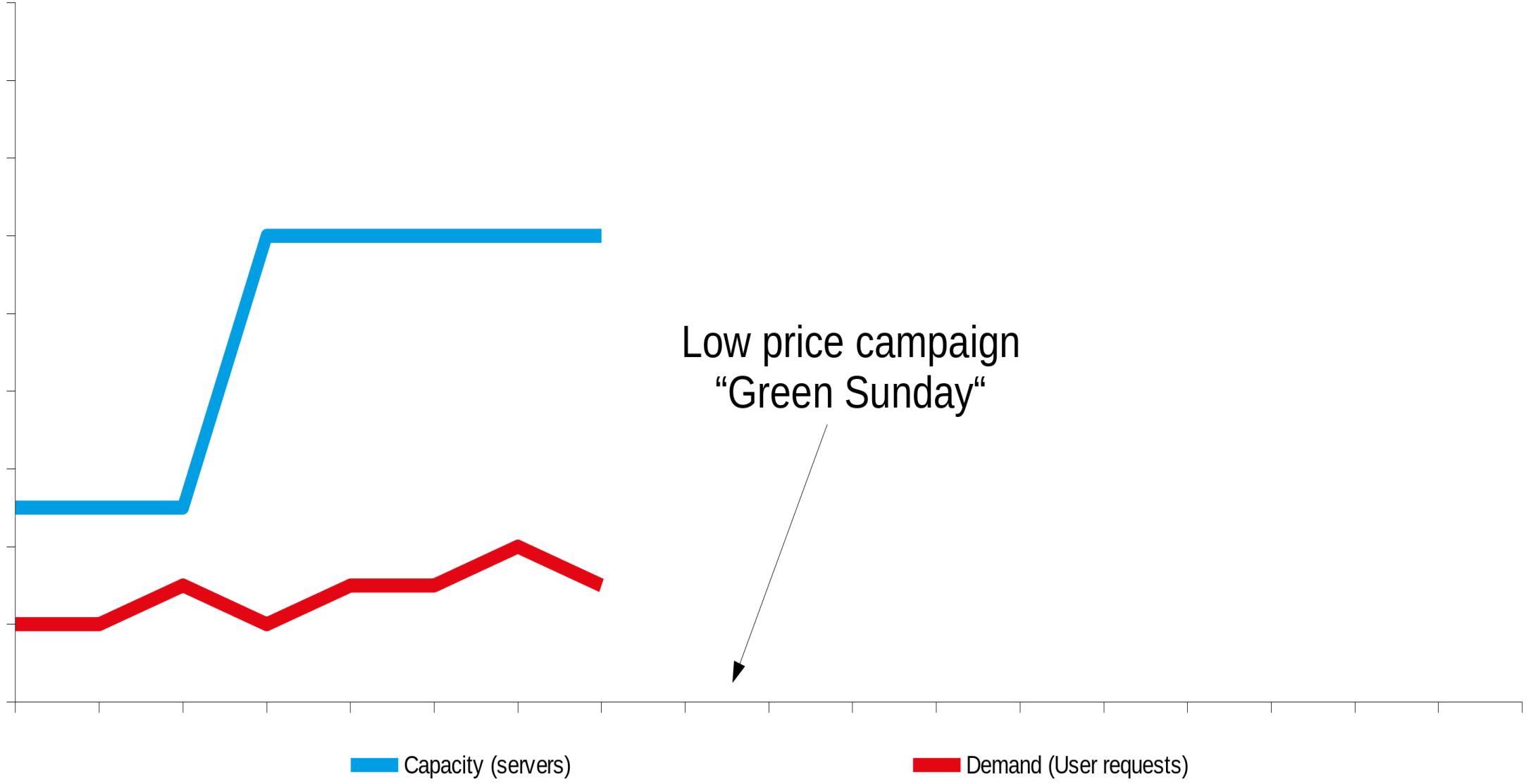


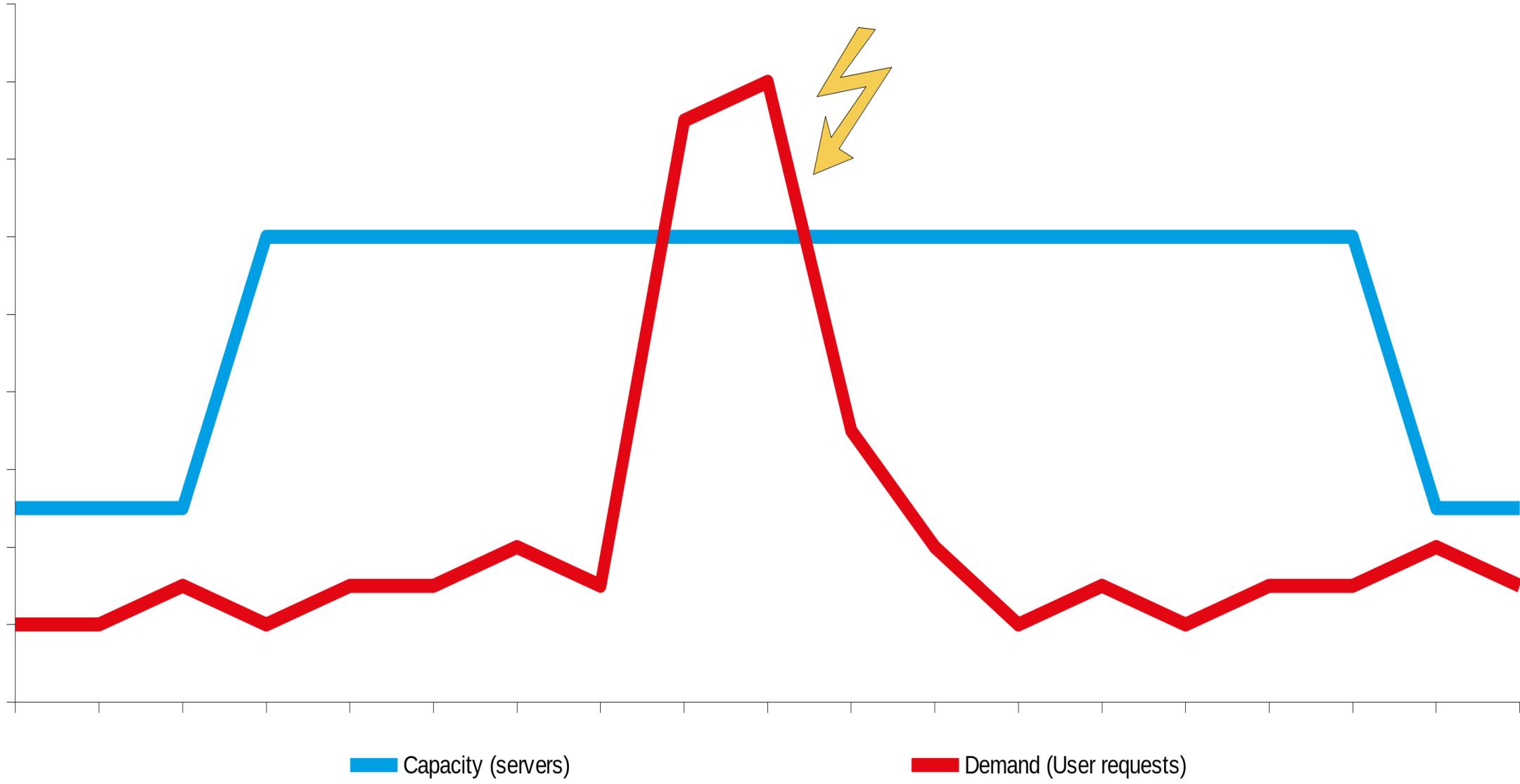
# Problem Statement

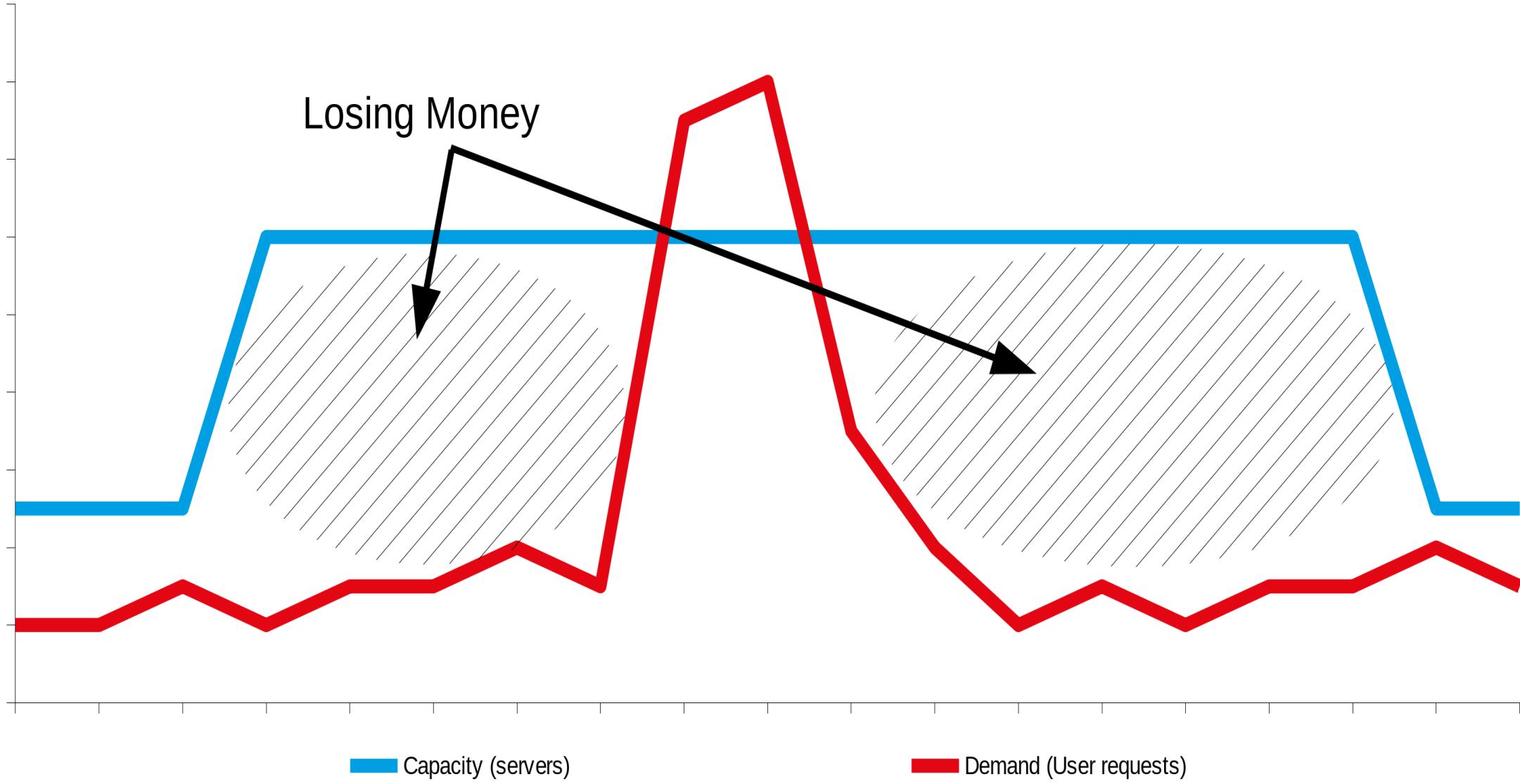
With inappropriate scaling you are losing

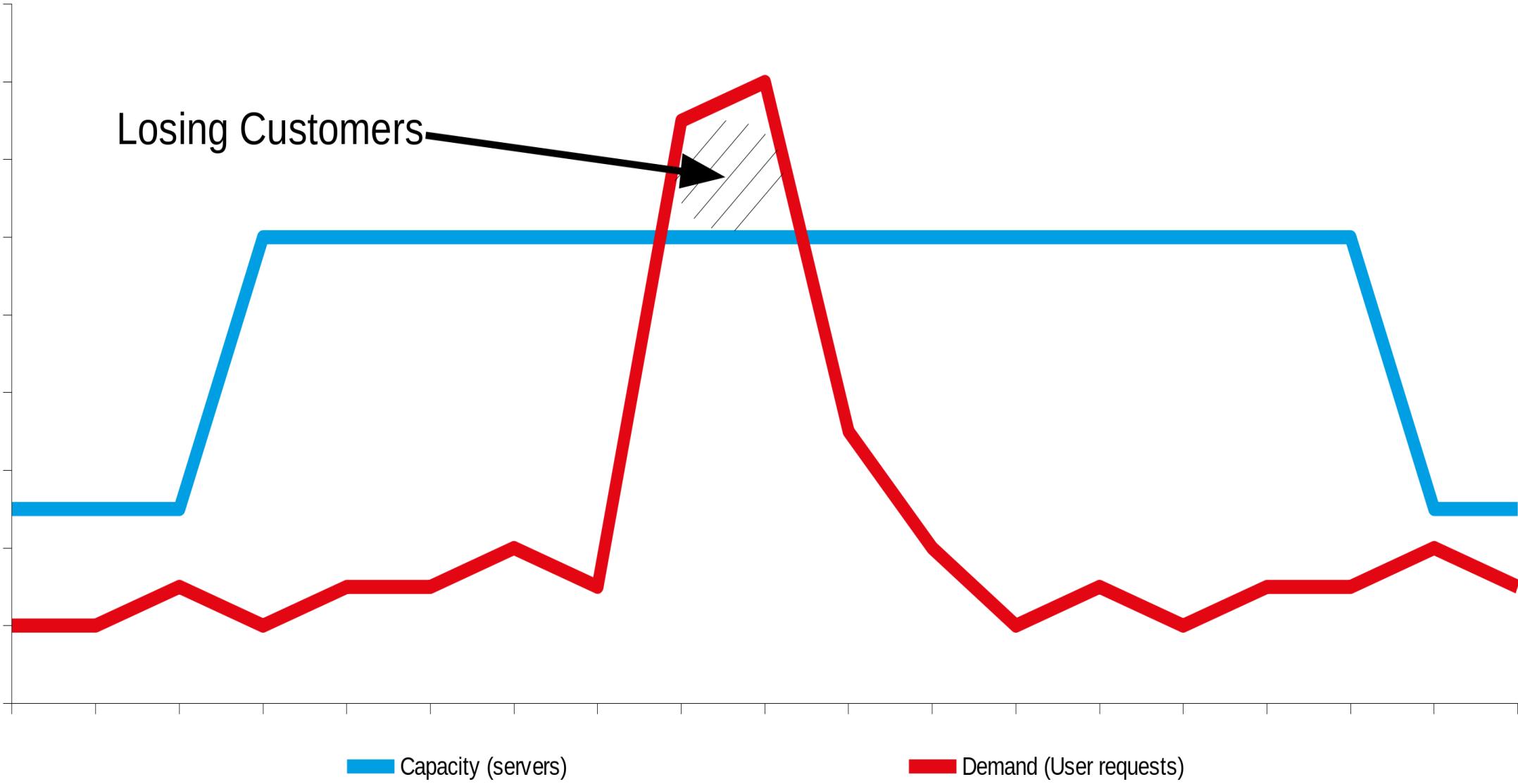
- customers 
- and money 





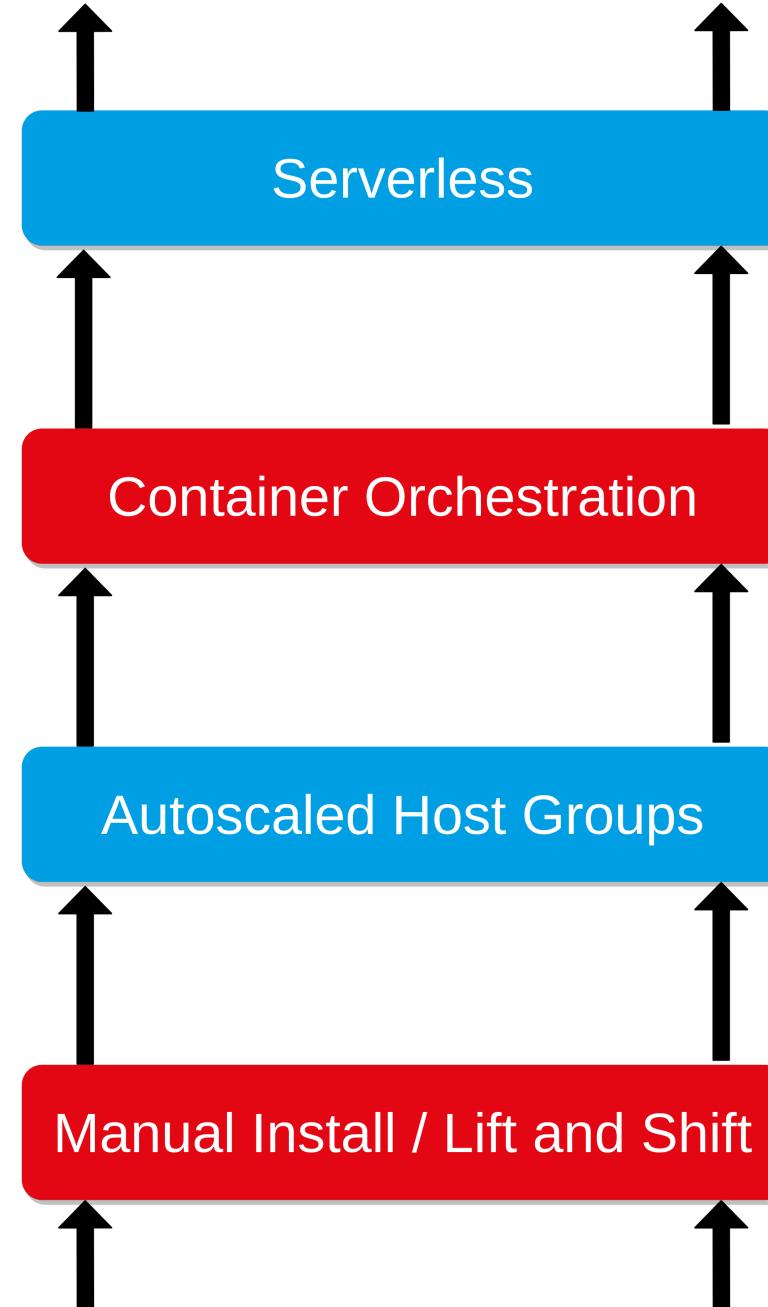








# Scaling in the Cloud





## Lift and Shift

- Migrate legacy VMs with no modification
- Doesn't scale very well
- Your mess for less
- Conclusion: don't do it

# Autoscaled Host Groups

- Lift and Shift done right

Three Step process:

1. Persist data outside VM
2. Automate VM setup
3. Create VM image



# Example Autoscaled Host Groups



- VM Image

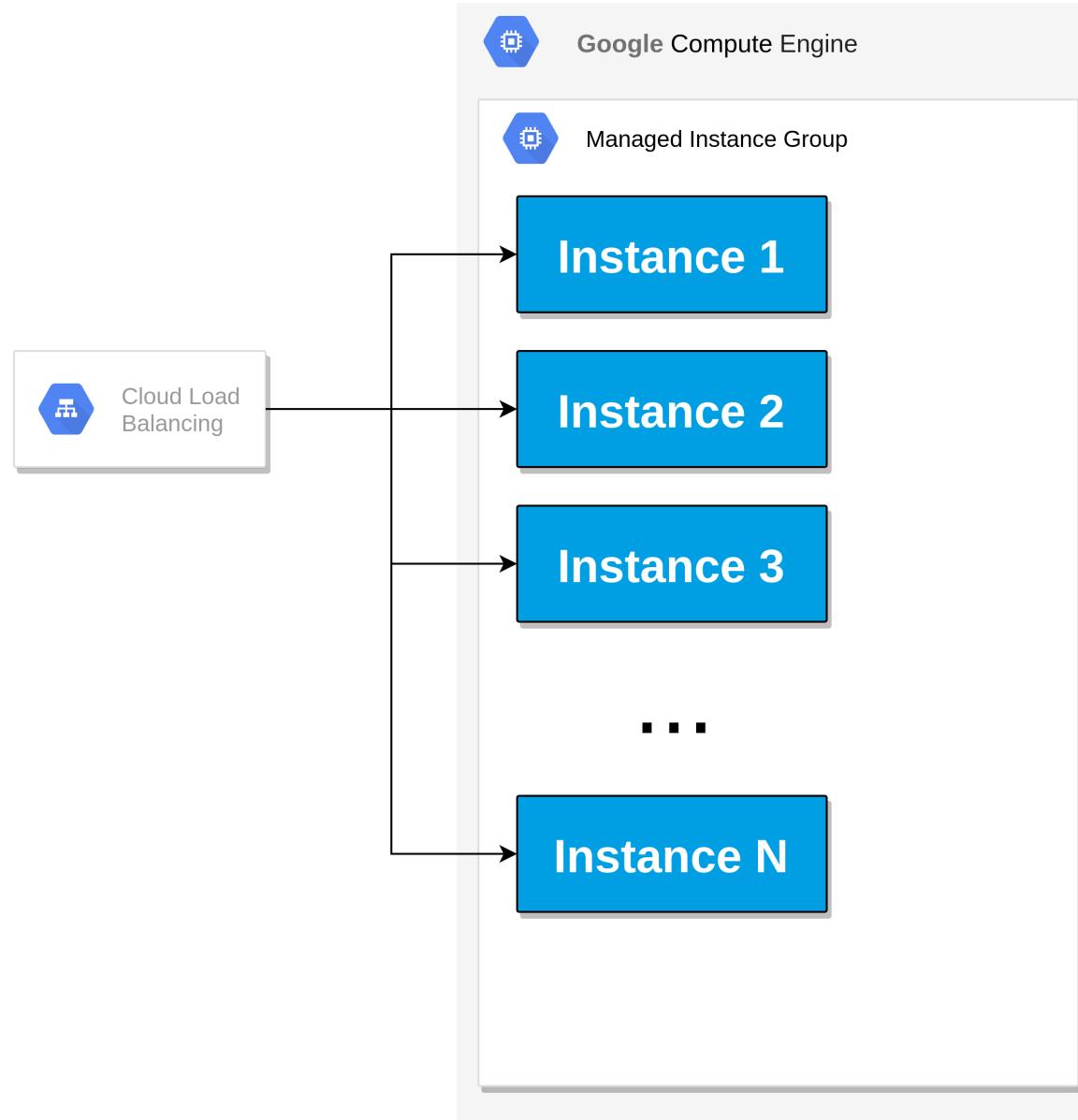


- Managed Instance Group -> VM Autoscaler



- VM Template
- Managed instance group
- Loadbalancer

# Managed Instance Group in GCP



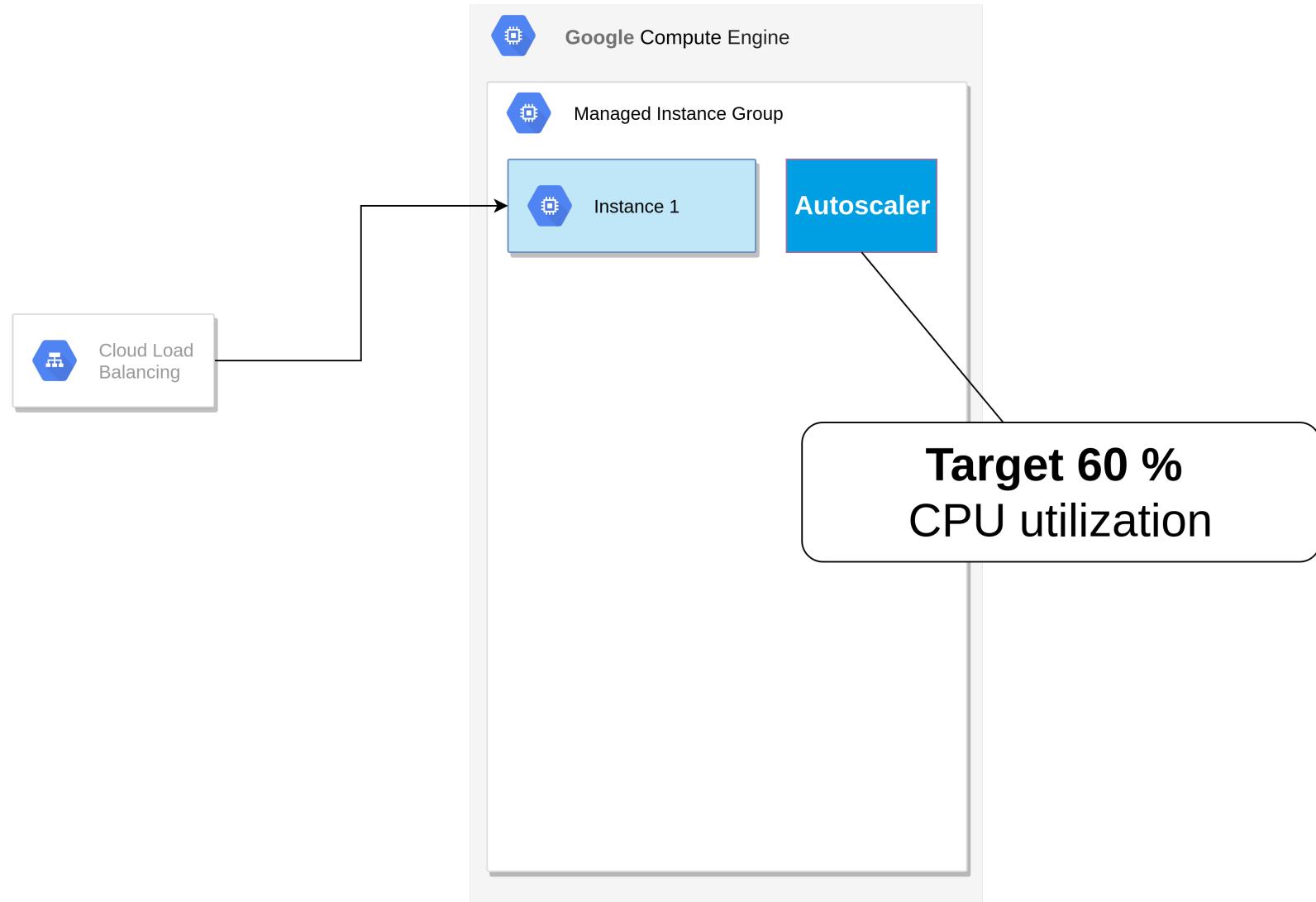
# Example Application

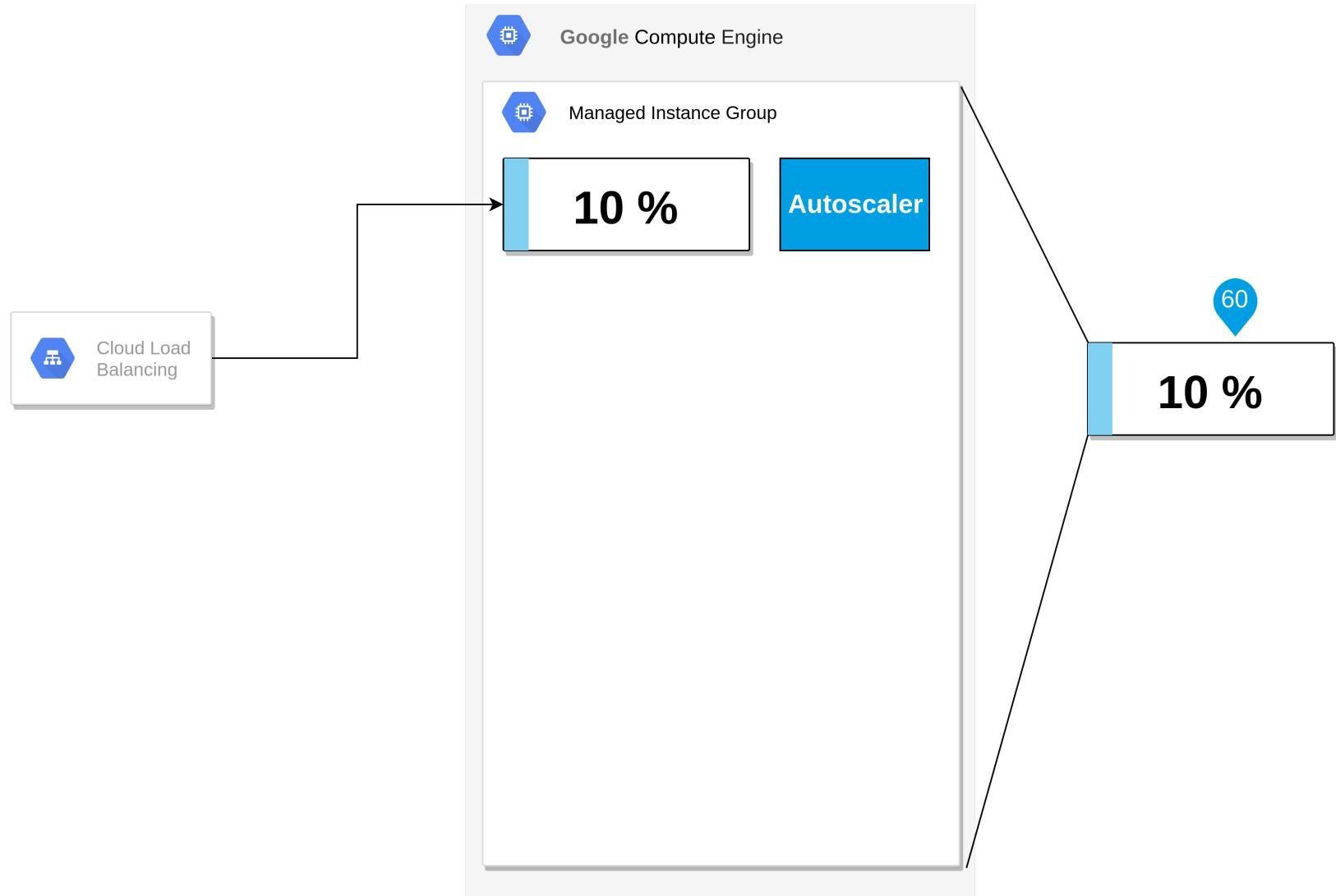
- Hashy

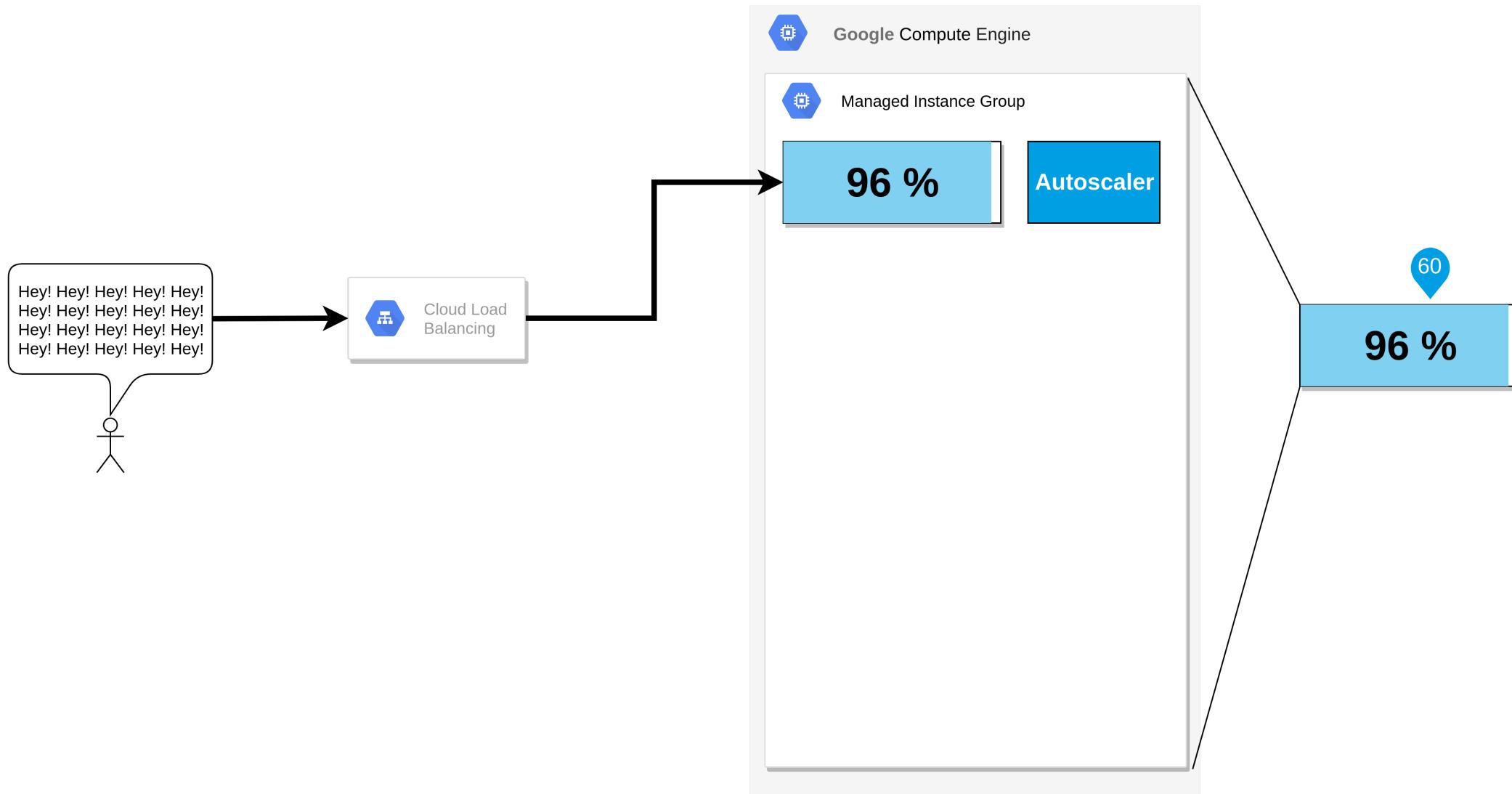
```
bcrypt( time.now() )
```

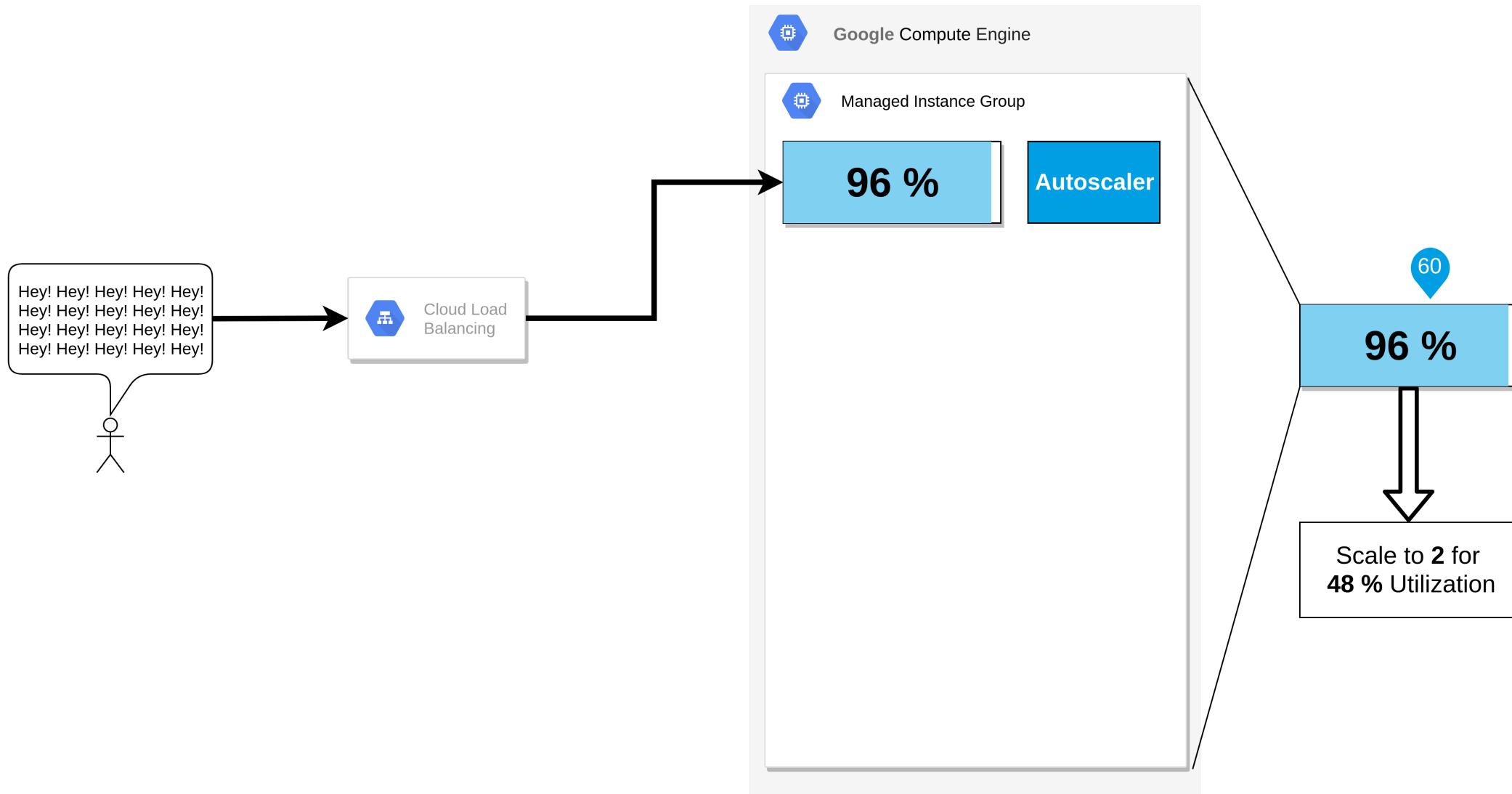
# **Demo Managed Instance Group Part 1**

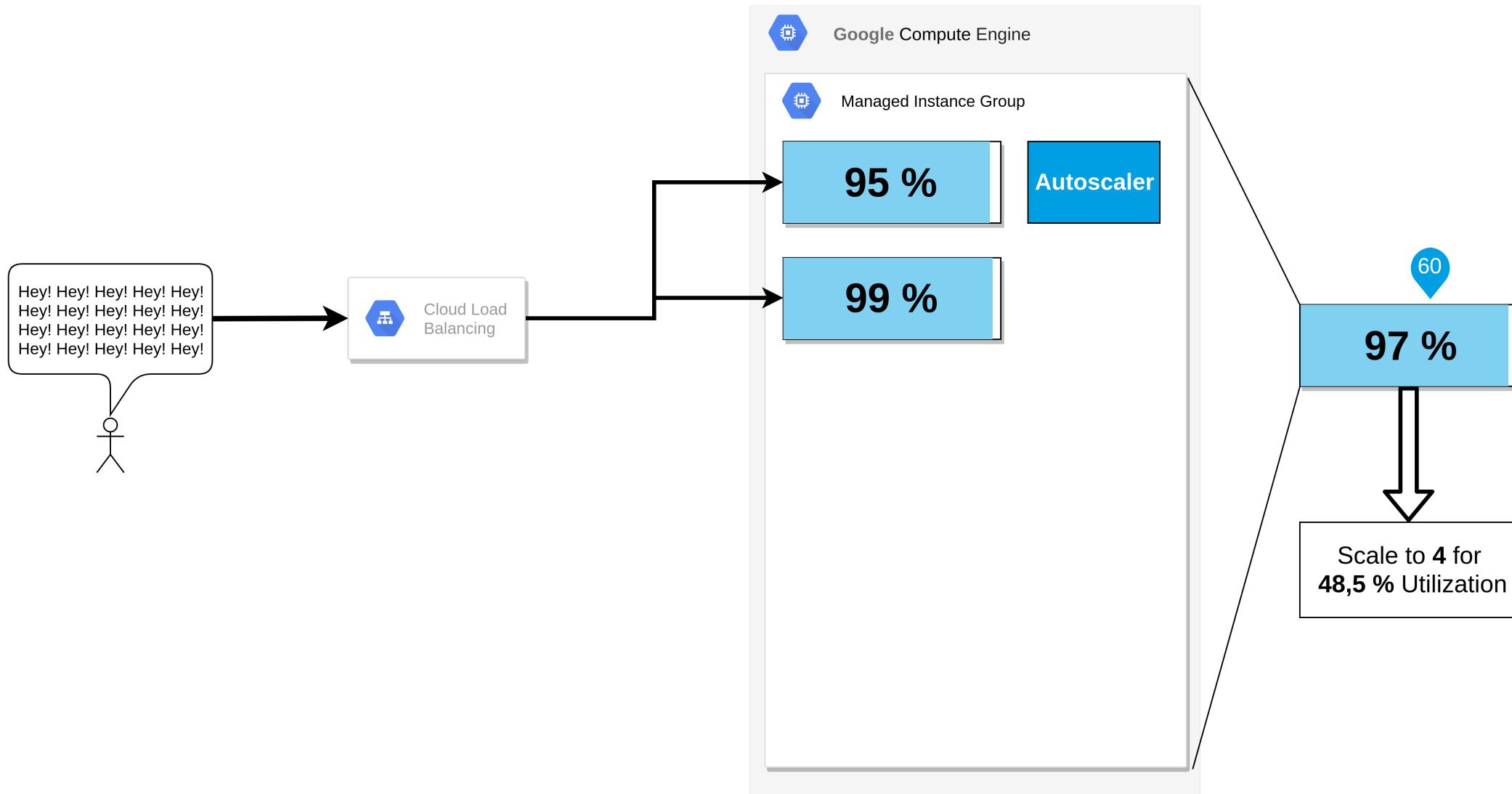


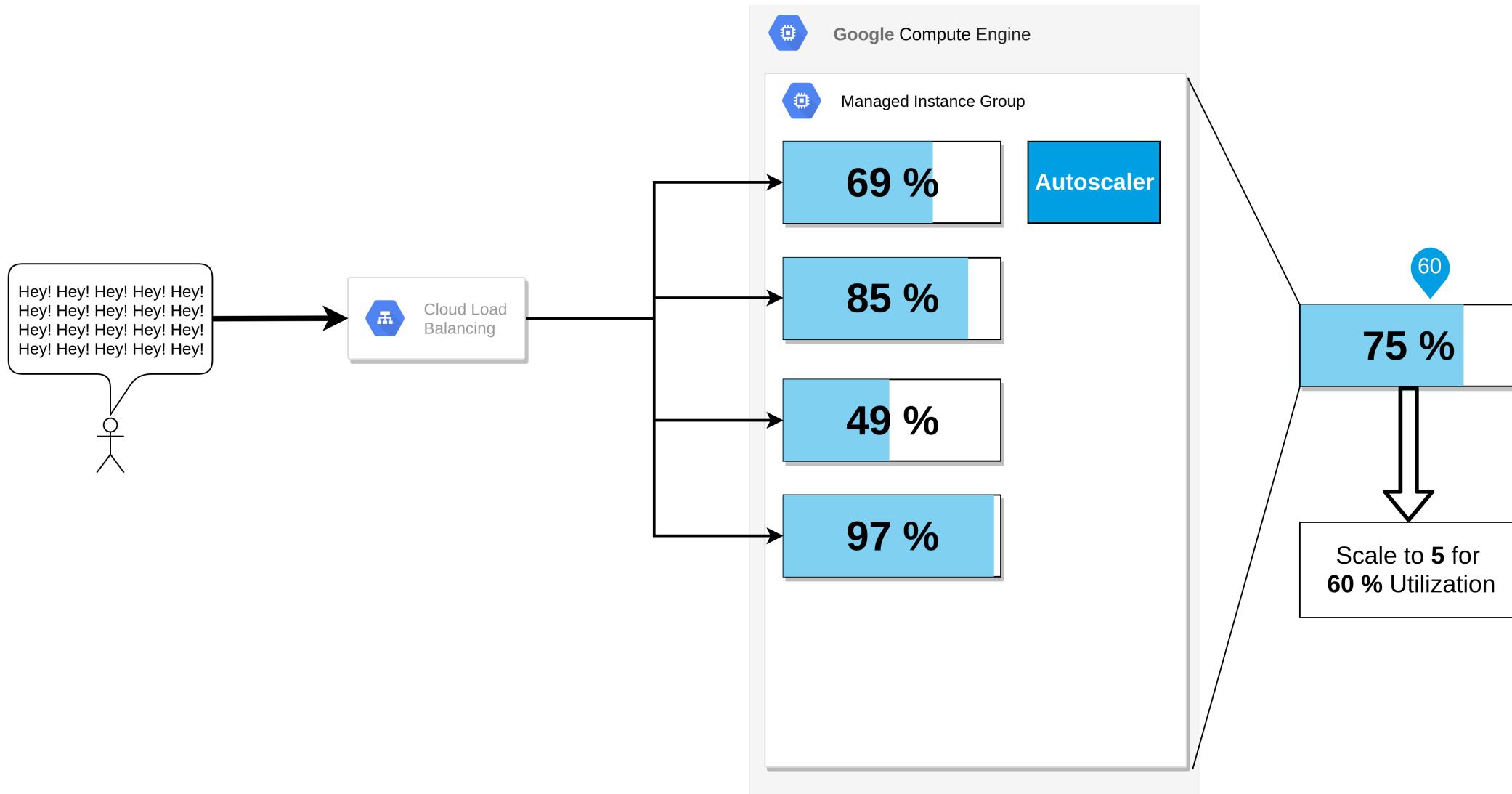


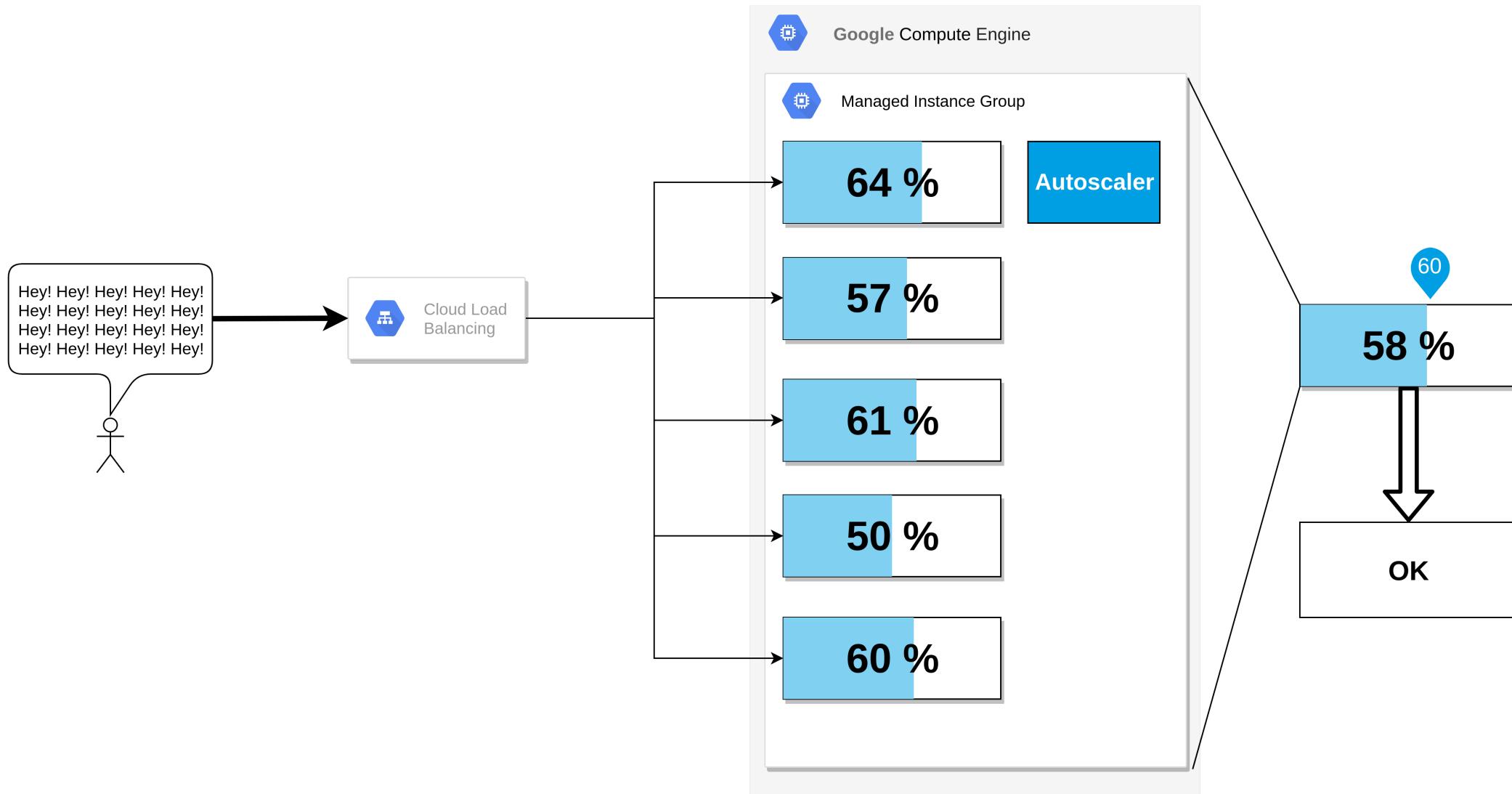












# **Demo Managed Instance Group Part 2**

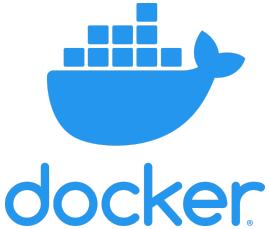




# Container Orchestration

- Paradigm shift: There are no VMs
- Pool of resources
- Services containerized
  - > strip out OS
  - > runtime image instead VM image

# Example Container Orchestration



- Container Image

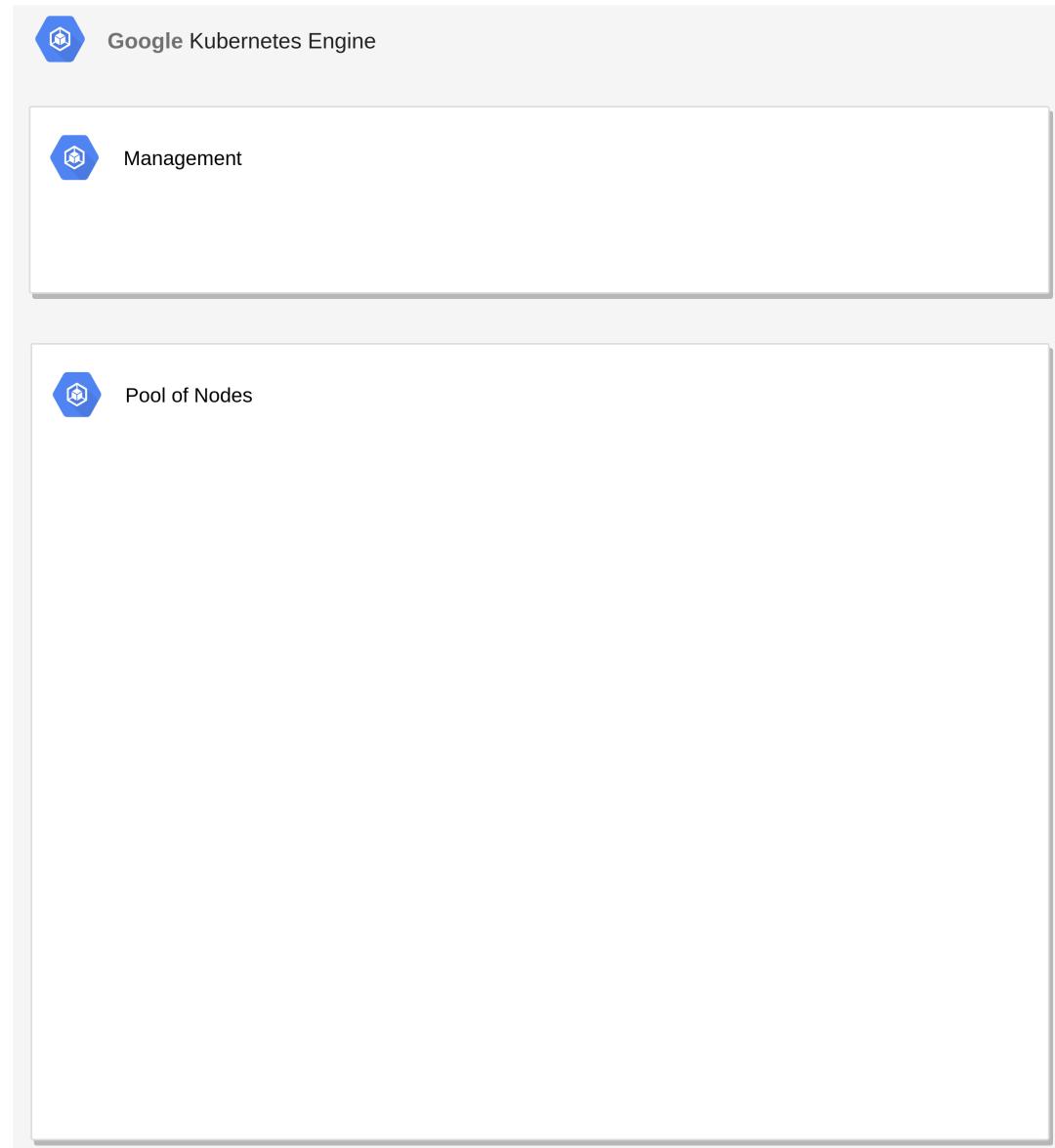


- Deployment
- Horizontal Pod Autoscaler
- Service

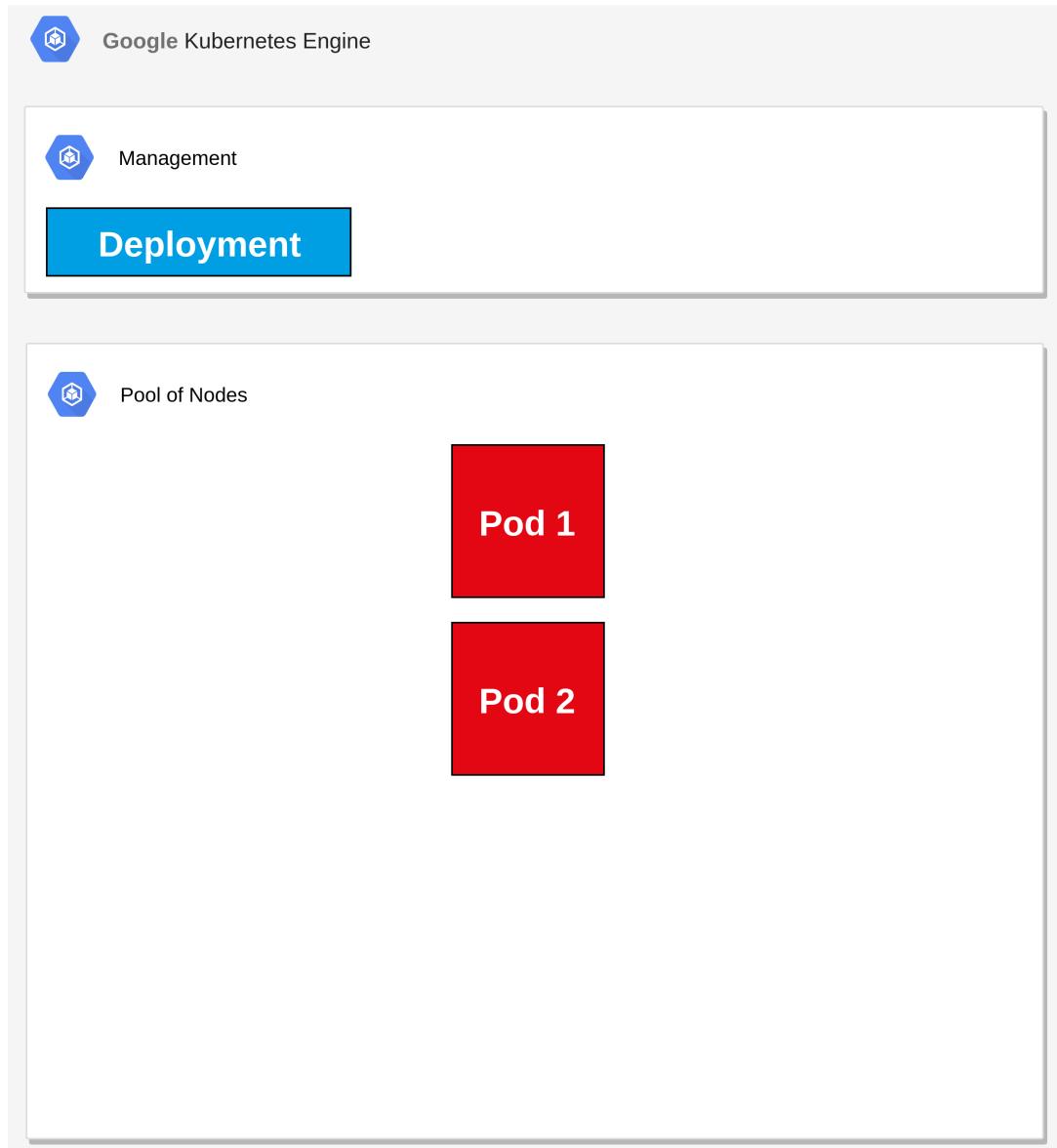


- Kubernetes Cluster
- Node Autoscaler

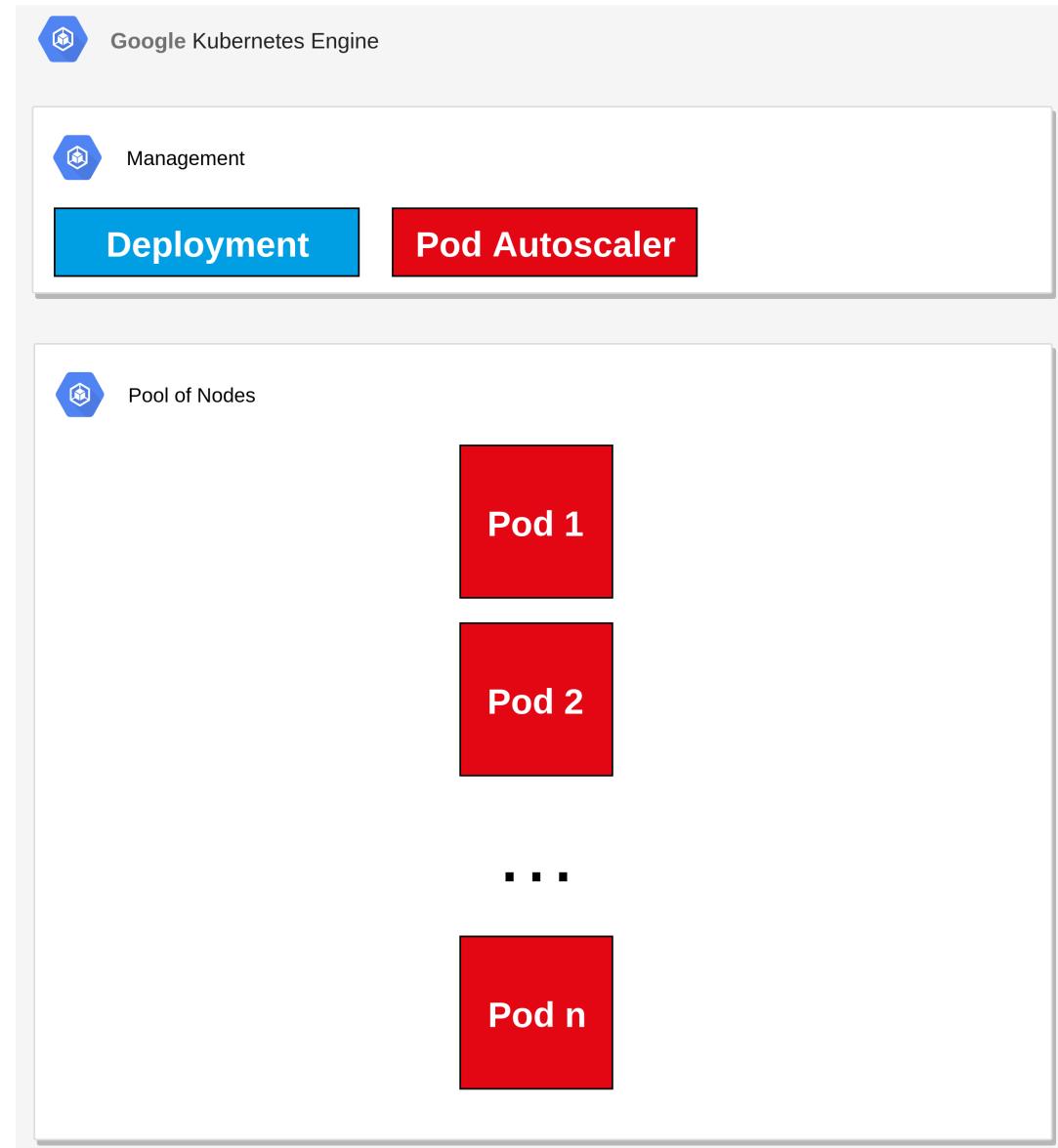
# Kubernetes Cluster



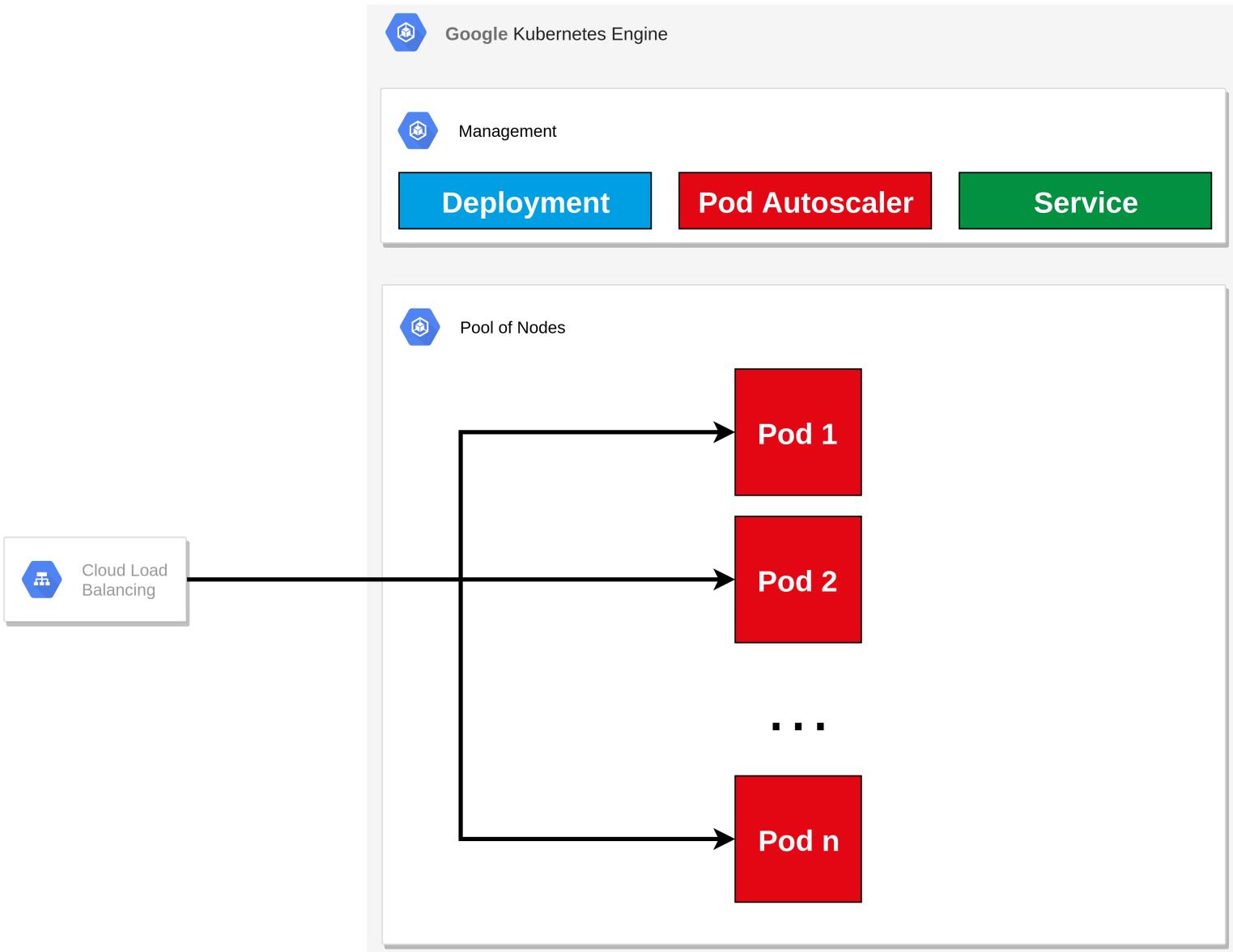
# Kubernetes Deployment



# Kubernetes Pod Autoscaler



# Kubernetes Service

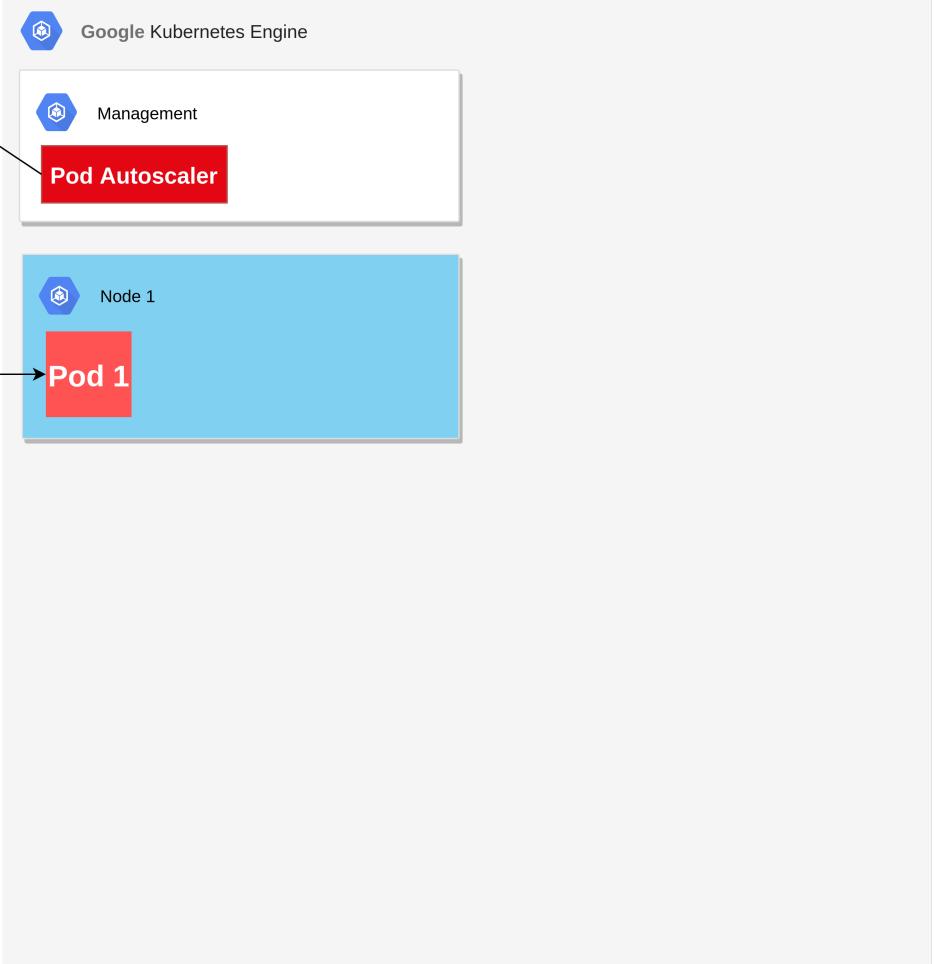


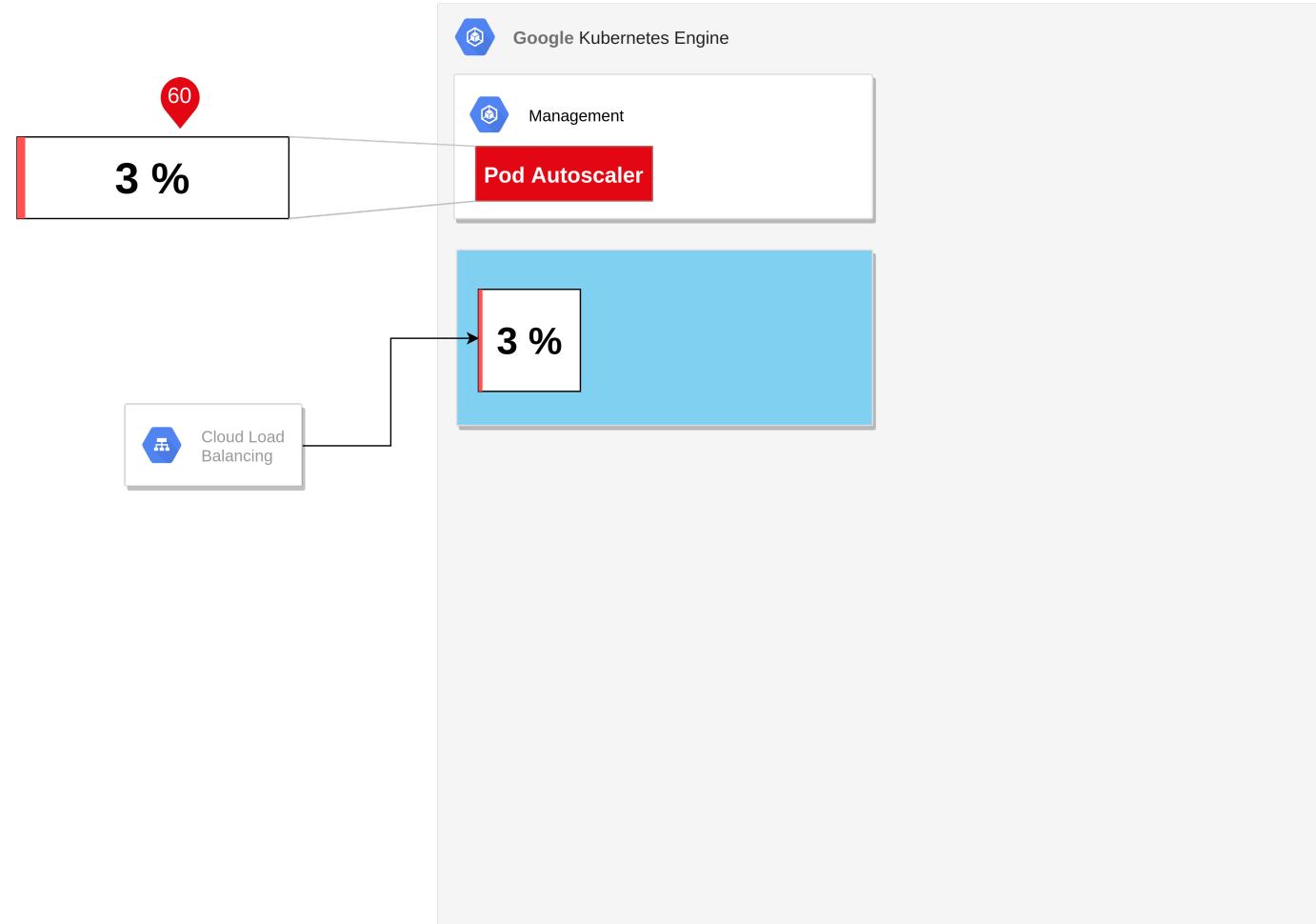
# **Demo Kubernetes**

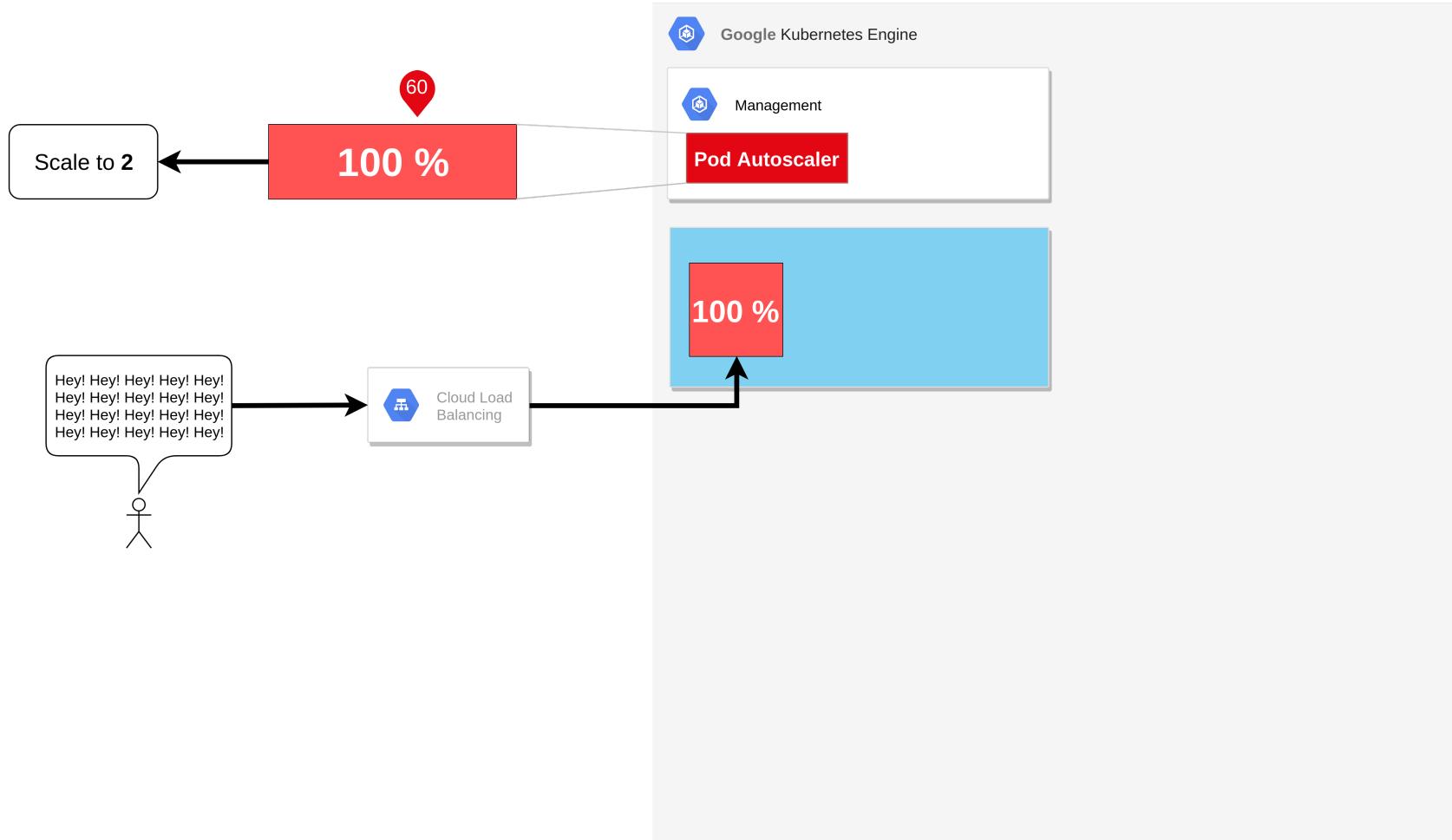
## **Part 1**

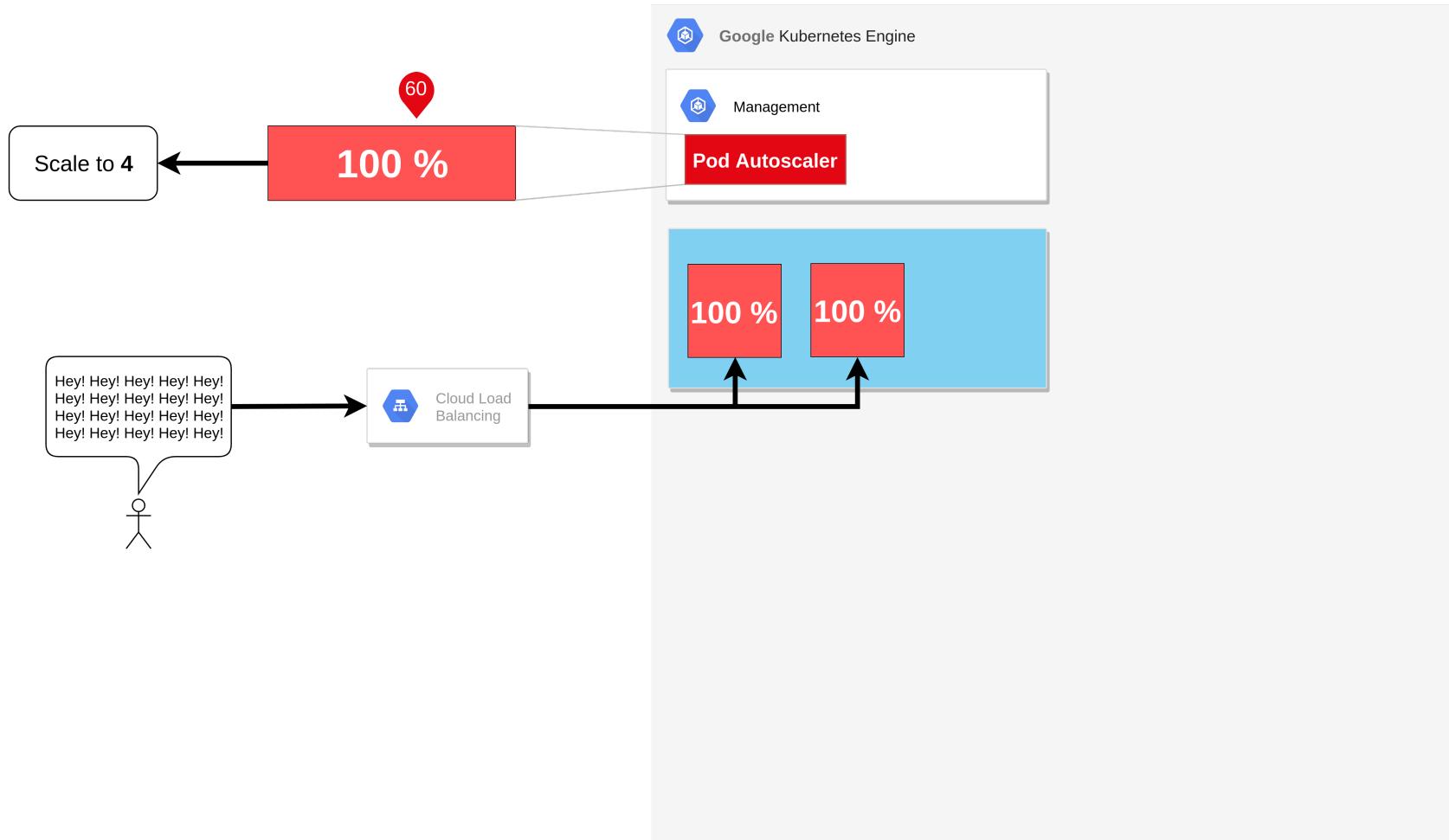


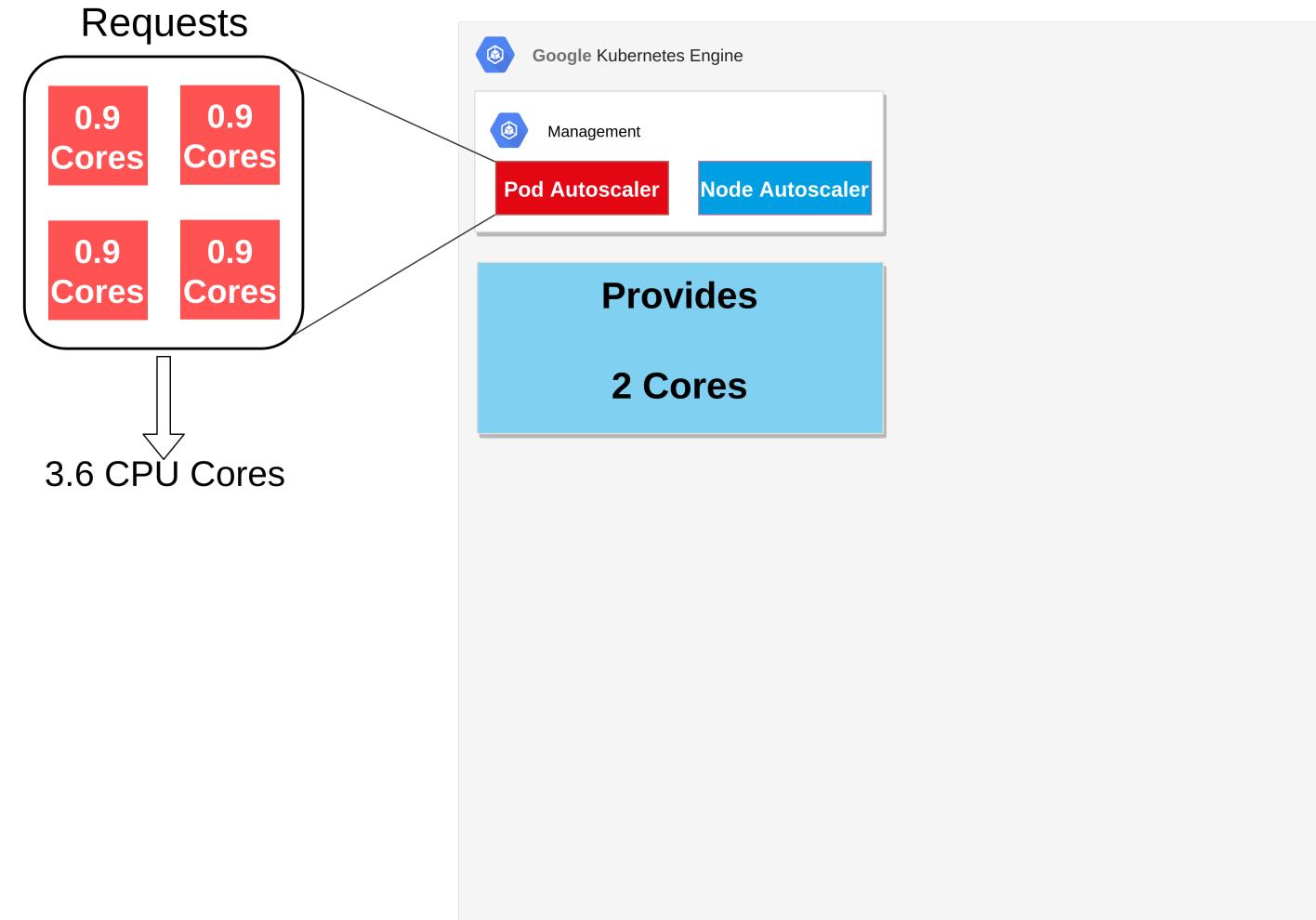
**Target 60 %  
CPU utilization**

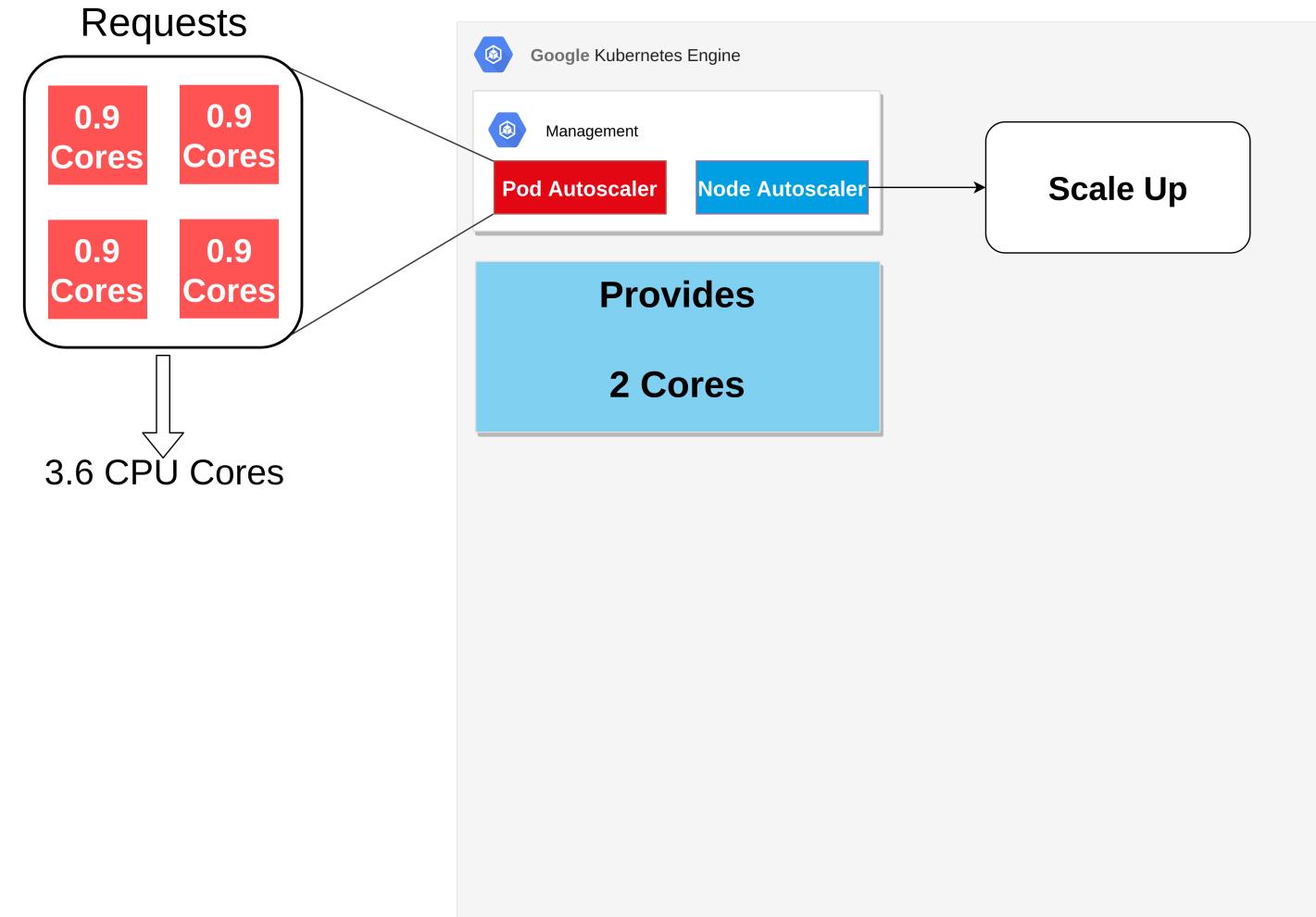


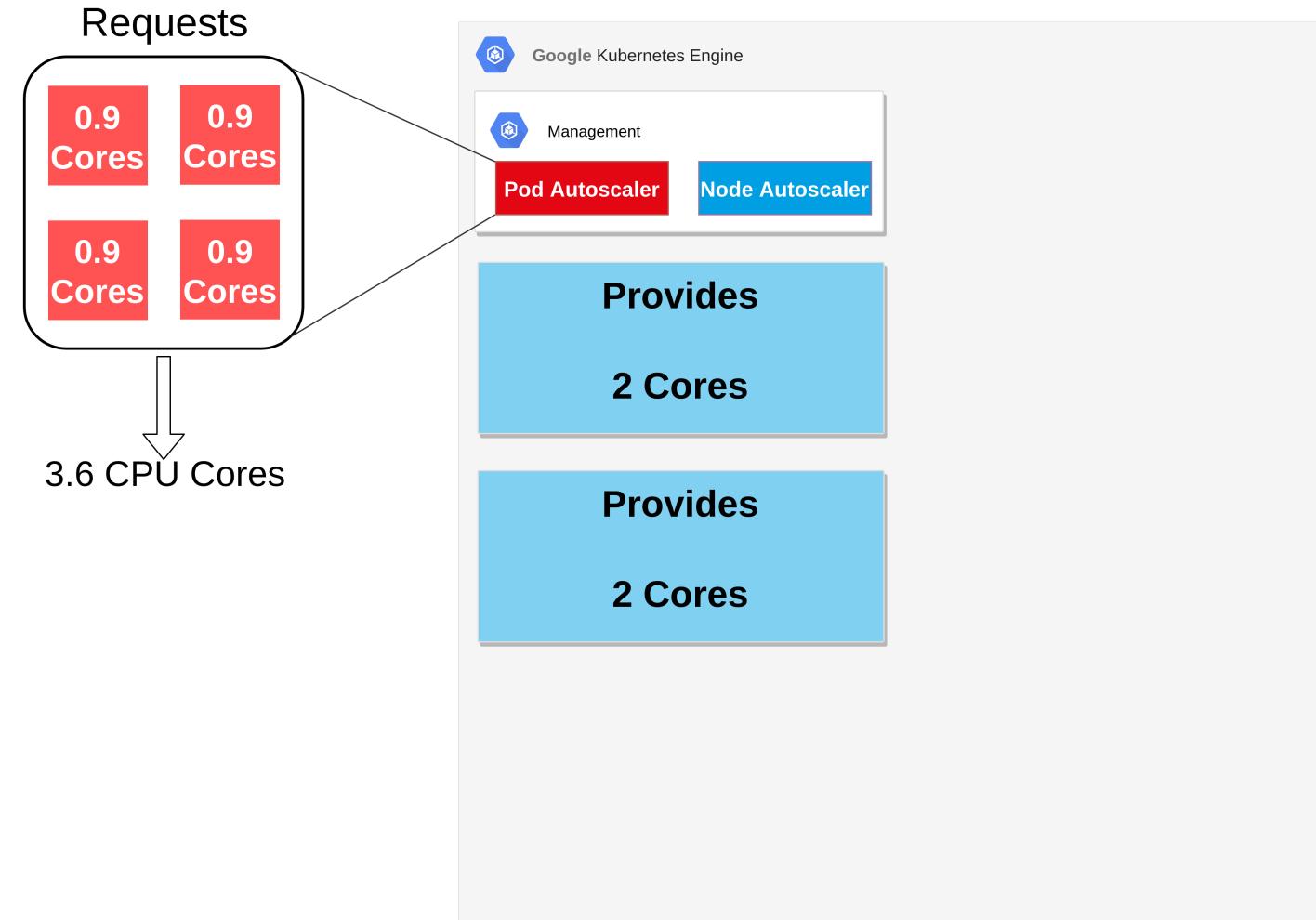


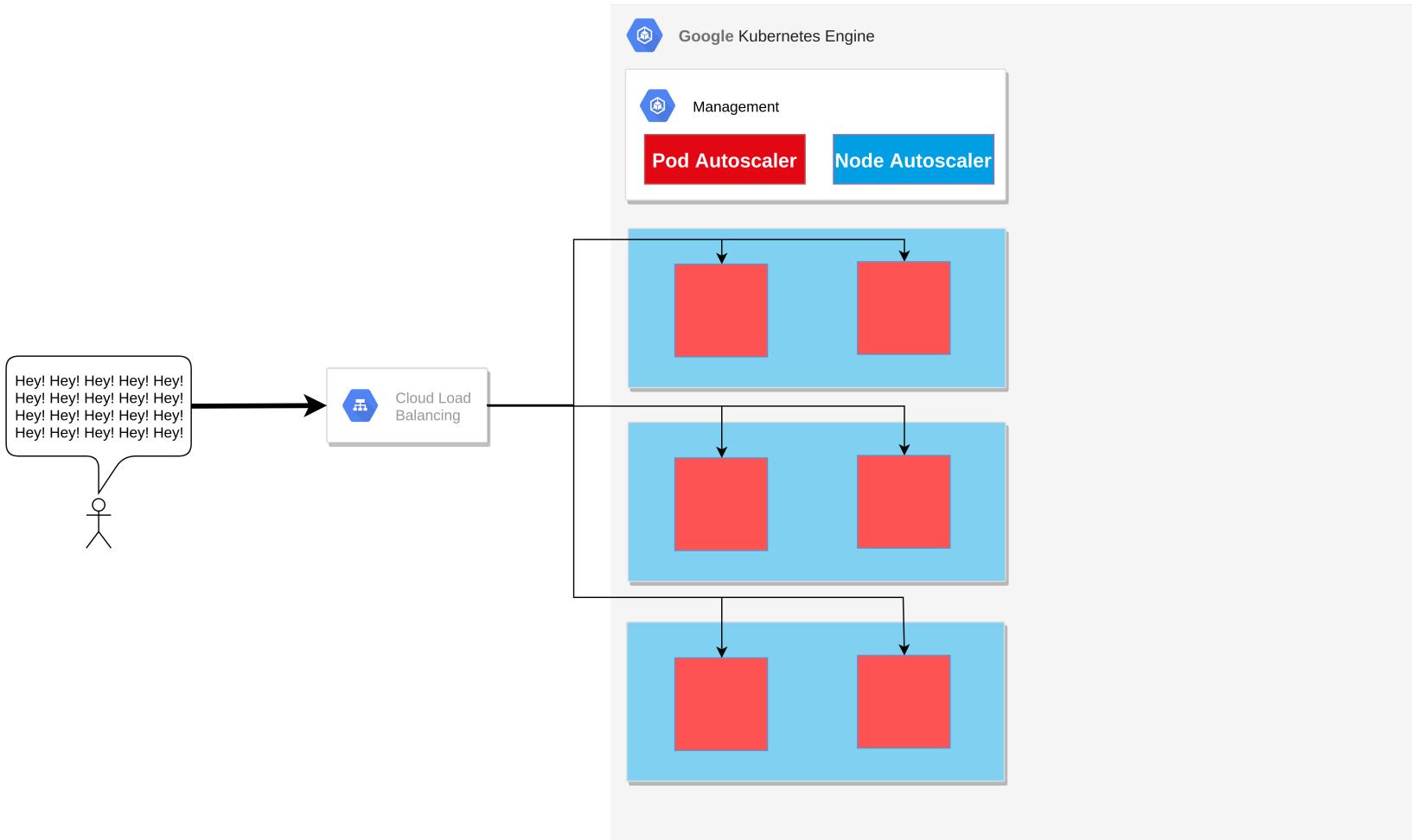












# **Demo Kubernetes**

## **Part 2**

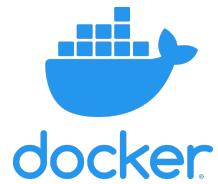




# Serverless

- Paradigm shift: There is no infrastructure
- No infrastructure management
- Pay per use
- Stateless

# Example Serverless



Container Image



Deployment

# Demo Serverless





## Takeaways

- Scalability is a fundamental requirement
- Cloud providers make scaling easy
- Don't do lift and shift

# Thanks for listening

- Demos can be found at [scale.8c.at](https://scale.8c.at)  8 
- Questions?