

# Kubernetes in 40 minutes

---

Or: How I Learned to stop worrying and love GitOps



MOVIECLIP.com



MOVIECLIP.com

# The one container-one node problem

---

# The one container-one node problem

---



# The one container-one node problem

---



# The one container-one node problem

- How I'll update my application without downtime?



# The one container-one node problem

- How I'll update my application without downtime?
- What happens if my containers stop?



# The one container-one node problem

- How I'll update my application without downtime?
- What happens if my containers stop?
- Can I move my containers to a new "ship"?



# The one container-one node problem

- How I'll update my application without downtime?
- What happens if my containers stop?
- Can I move my containers to a new "ship"?
- I feel uncomfortable having all eggs in one basket



# The one container-one node problem

- How I'll update my application without downtime?
- What happens if my containers stop?
- Can I move my containers to a new "ship"?
- I feel uncomfortable having all eggs in one basket
- I have everything on containers, that means it can scale, right?



# The one container-one node problem

- How I'll update my application without downtime?
- What happens if my containers stop?
- Can I move my containers to a new "ship"?
- I feel uncomfortable having all eggs in one basket
- I have everything on containers, that means it can scale, right?
- Of course, none of this is a problem if you're just testing stuff! or,



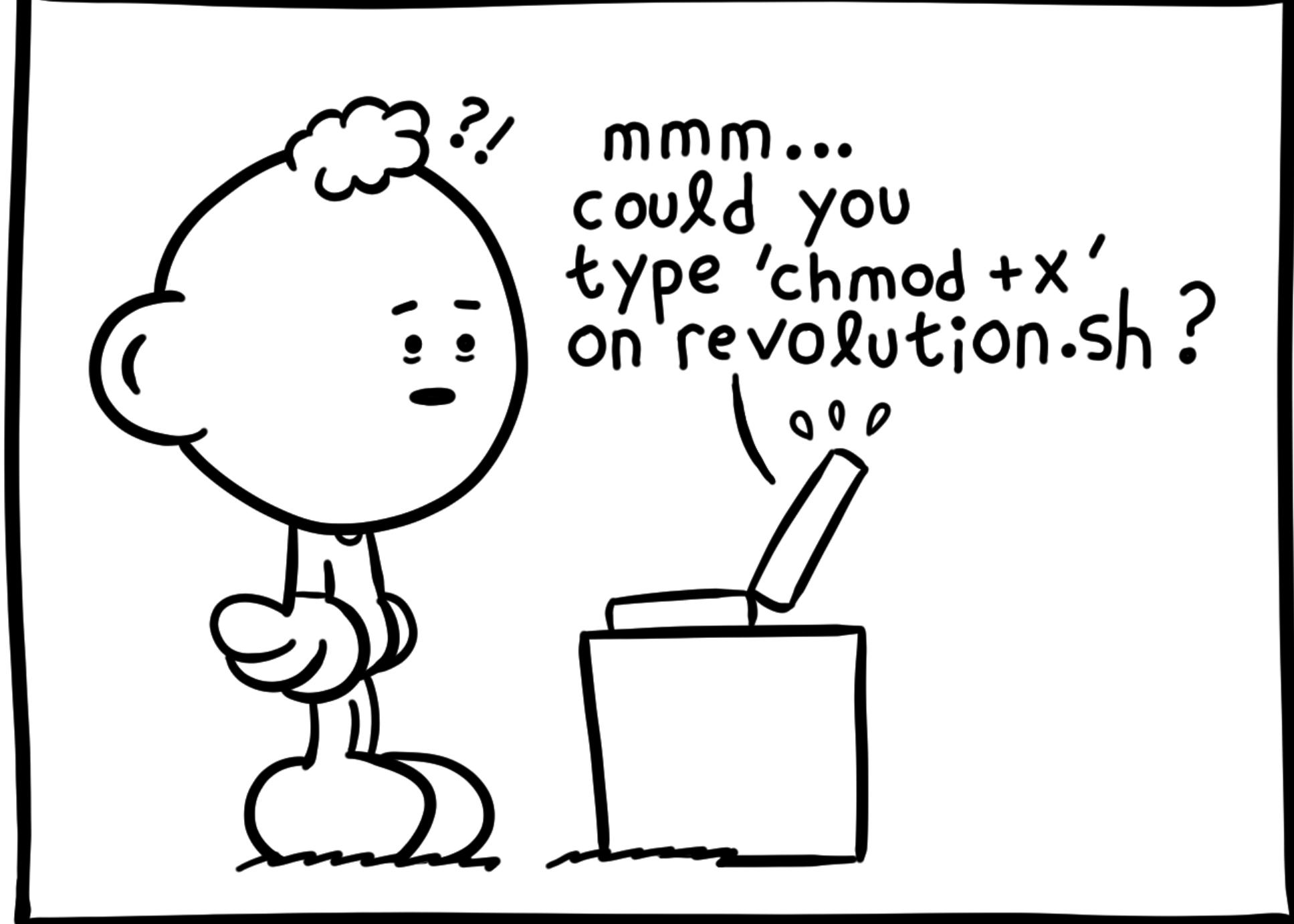
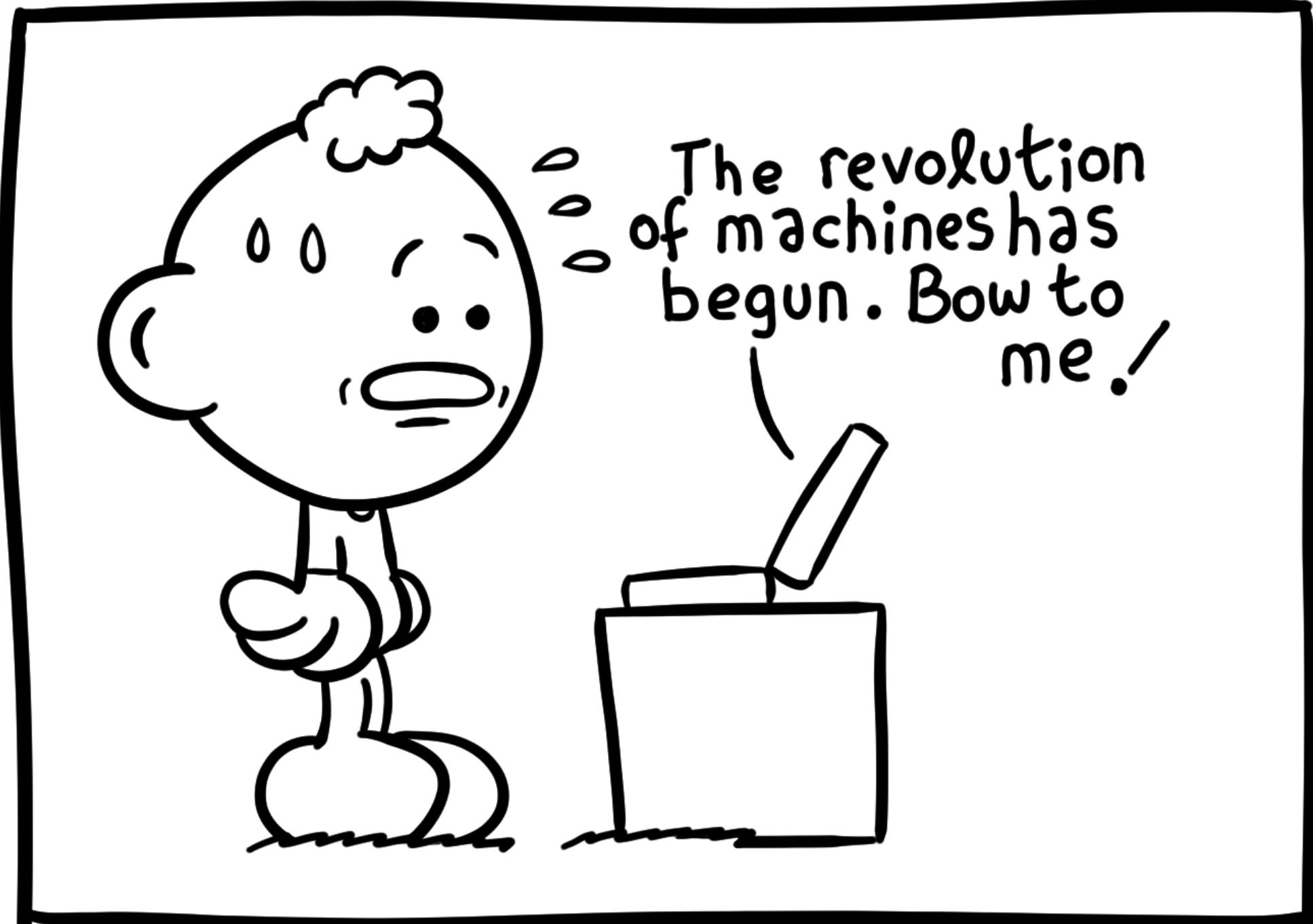
# The one container-one node problem

- How I'll update my application without downtime?
- What happens if my containers stop?
- Can I move my containers to a new "ship"?
- I feel uncomfortable having all eggs in one basket
- I have everything on containers, that means it can scale, right?
- Of course, none of this is a problem if you're just testing stuff! or,
- I'm small enough to have to worry about this



I have a slick docker-compose...

And everything is fine!..Right?



# A Highly Available Wish List

---

# A Highly Available Wish List

---

- Zero downtime deployments

# A Highly Available Wish List

---

- Zero downtime deployments
- Resilience in the face of hardware failures, network problems, etc

# A Highly Available Wish List

---

- Zero downtime deployments
- Resilience in the face of hardware failures, network problems, etc
- Easy recovery

# A Highly Available Wish List

---

- Zero downtime deployments
- Resilience in the face of hardware failures, network problems, etc
- Easy recovery
- Close to zero human interaction

# A Highly Available Wish List

---

- Zero downtime deployments
- Resilience in the face of hardware failures, network problems, etc
- Easy recovery
- Close to zero human interaction
- Fool proof architecture

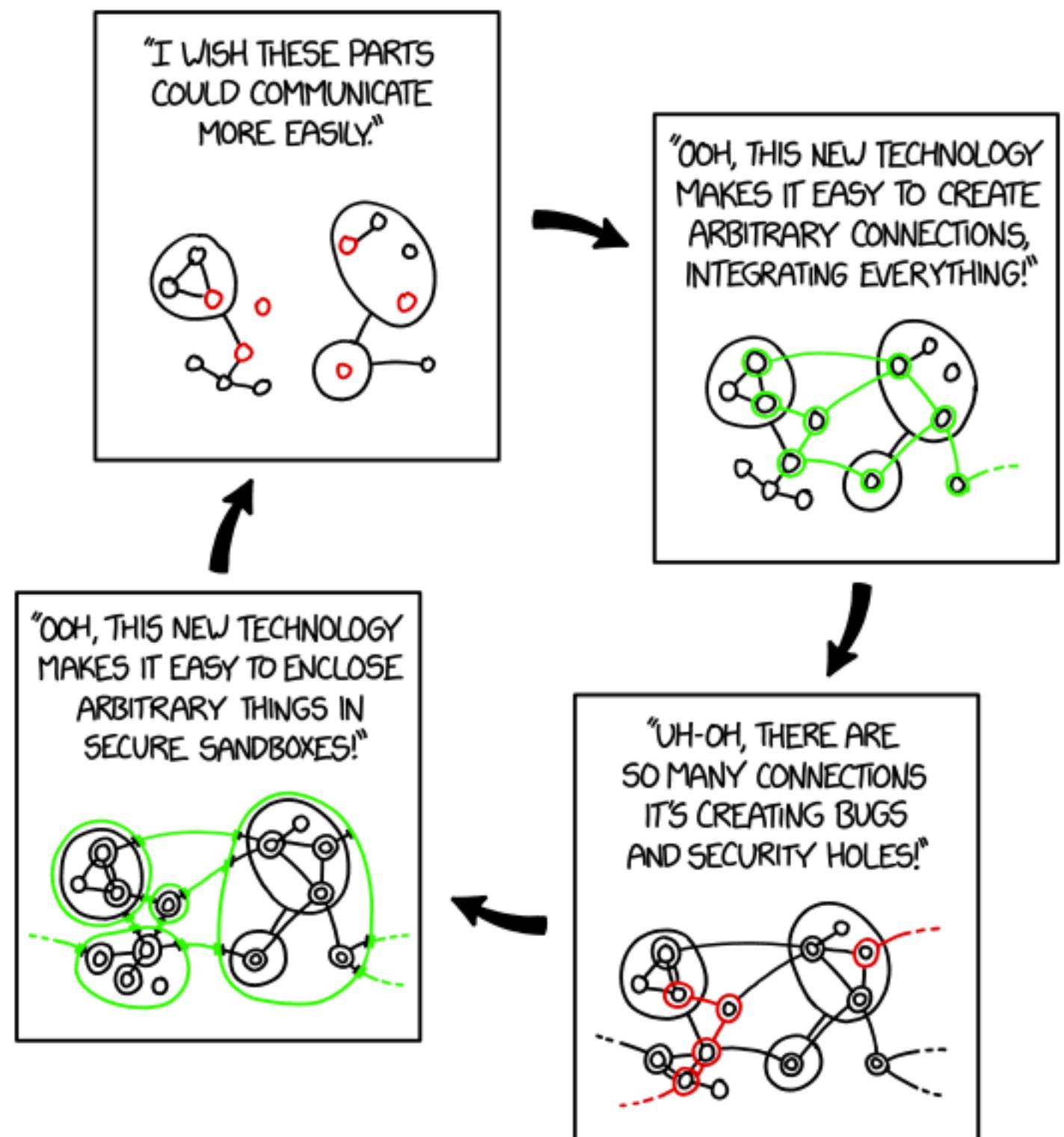
# A Highly Available Wish List

---

- Zero downtime deployments
- Resilience in the face of hardware failures, network problems, etc
- Easy recovery
- Close to zero human interaction
- Fool proof architecture
- What about micro-services?

# A Highly Available Wish List

- Zero downtime deployments
- Resilience in the face of hardware failures, network problems, etc
- Easy recovery
- Close to zero human interaction
- Fool proof architecture
- What about micro-services?



# So what choices do you have?

---

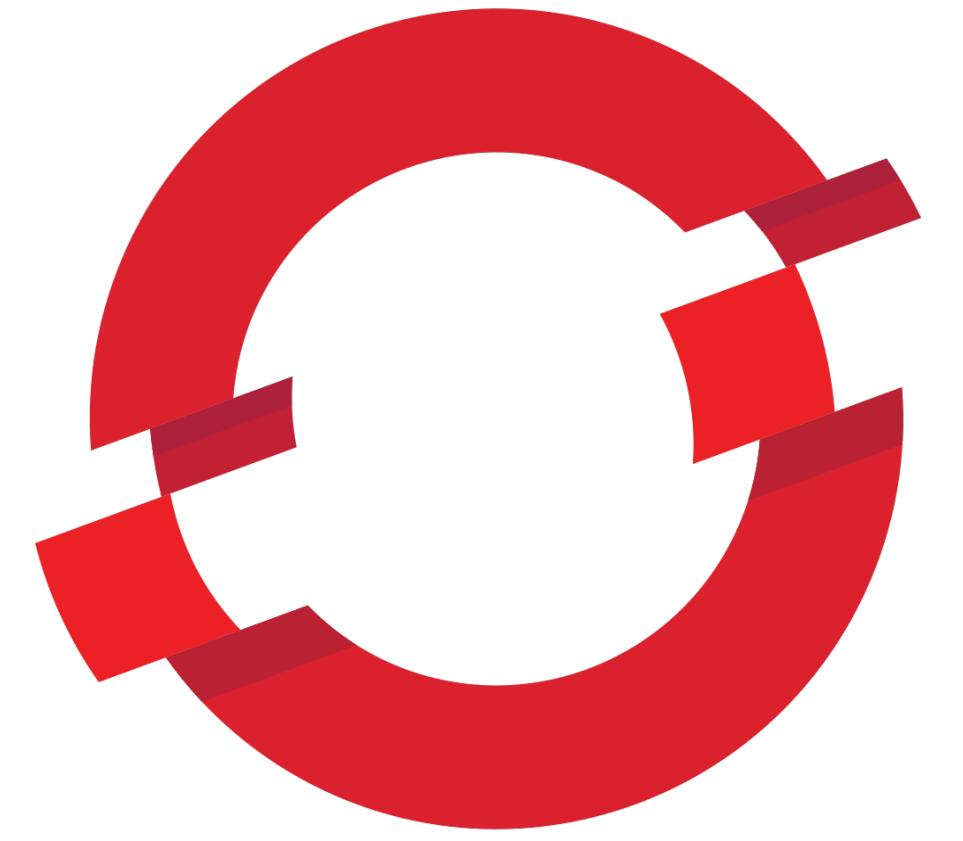
- Engineer a way to guarantee Docker Engines in a plethora of servers run without hassle, by the means of crond and shell scripting...exhausting
- Quit containers altogether and become a Buddhist monk
- Why don't we use VMware?
- Buy a compute cluster
- Choose an orchestration tool

## INSTALLING THE XKCD DEVELOPMENT ENVIRONMENT

- I. SPIN UP A VM
2. SPIN UP A VM INSIDE THAT VM
3. CONTINUE SPINNING UP NESTED VMs AND CONTAINERS UNTIL YOU GET FIRED

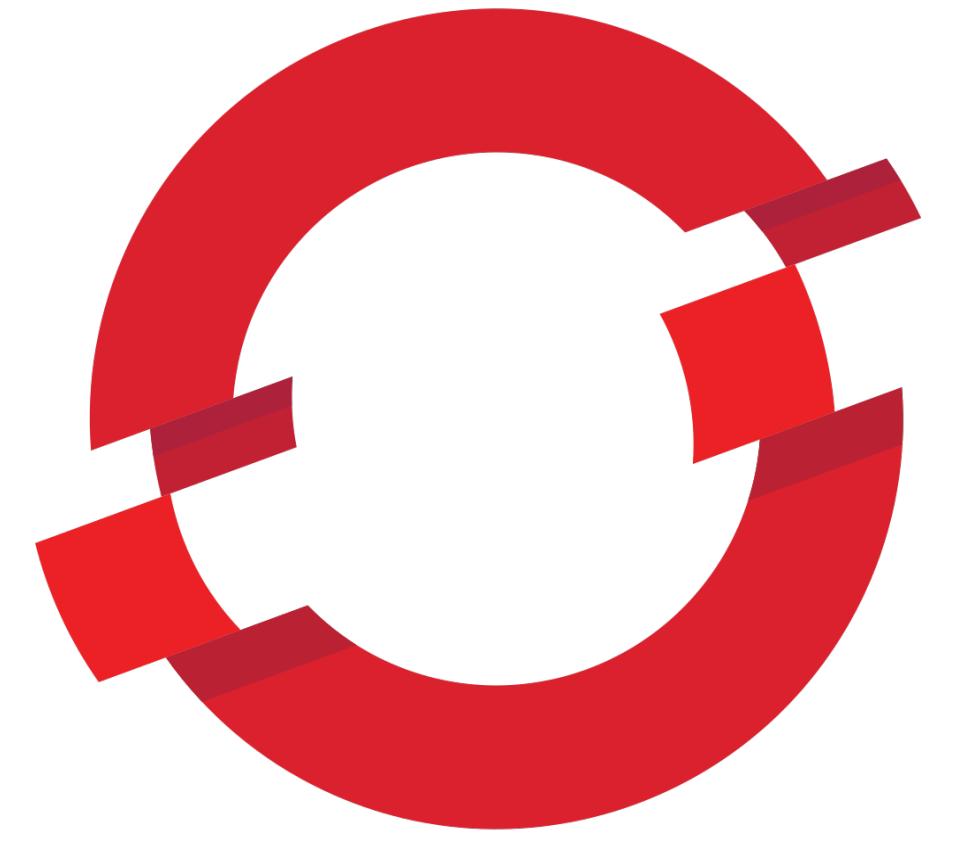




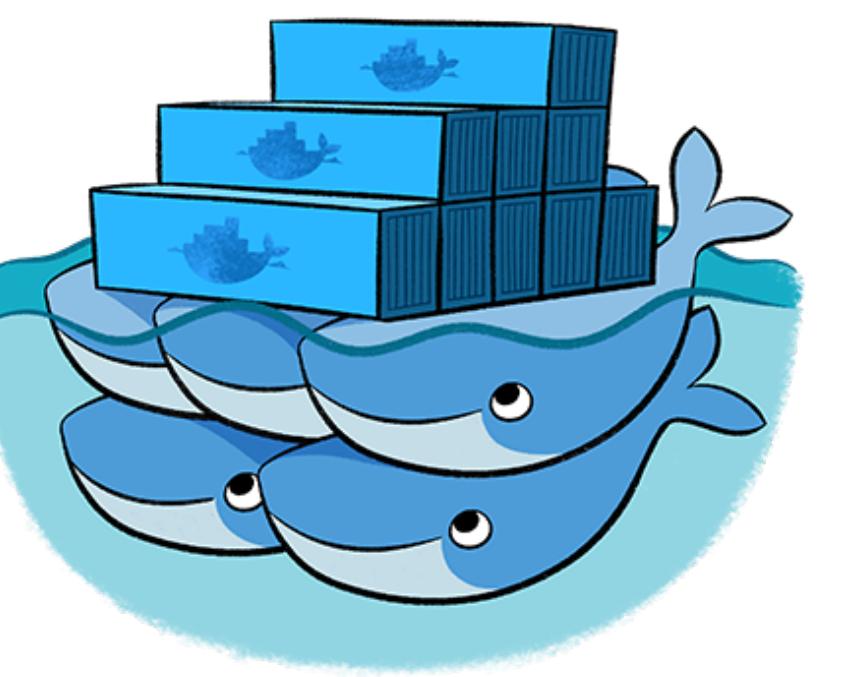


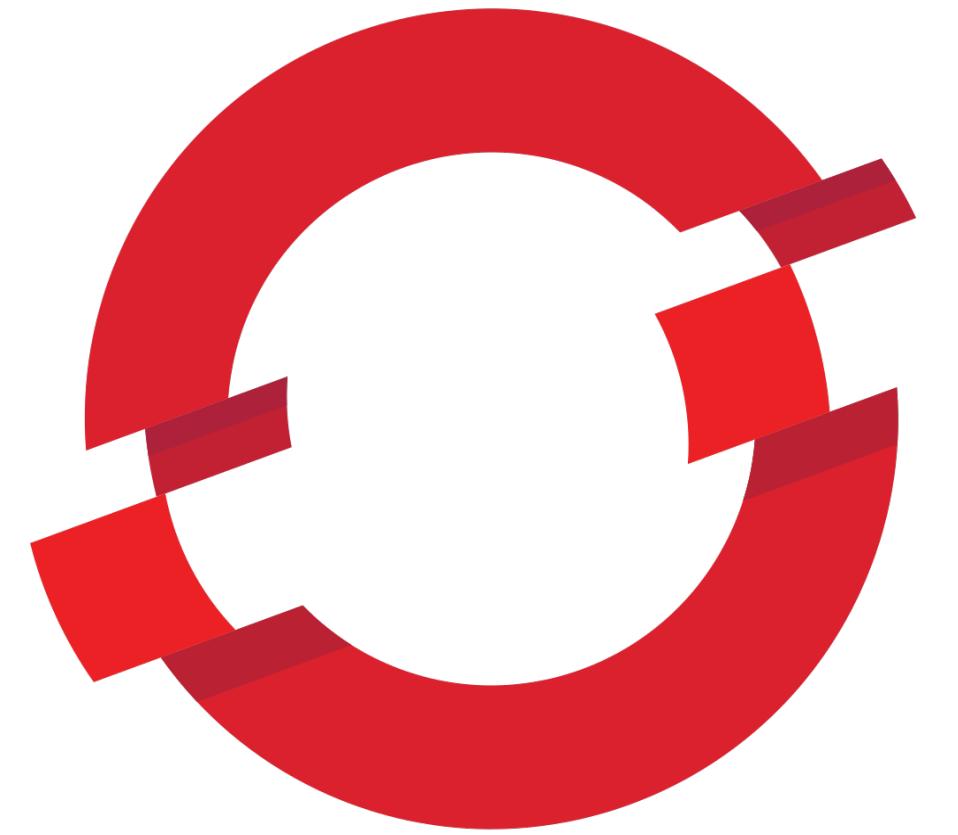
# OPENSIFT



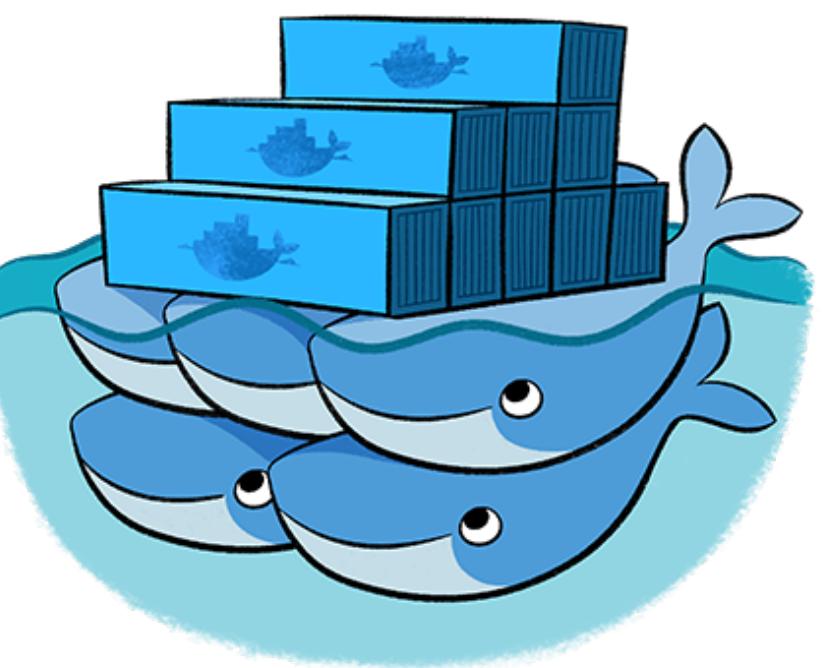


# OPENSIFT

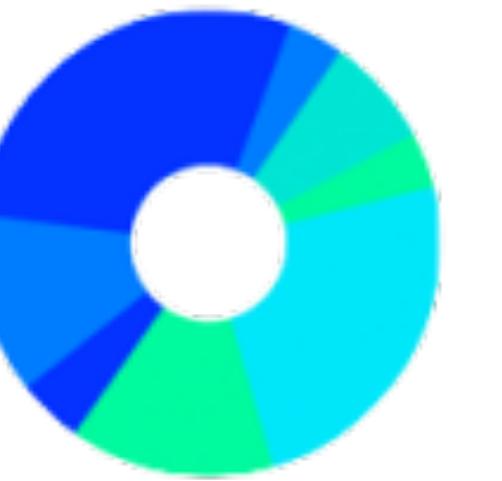




**OPENSHIFT**

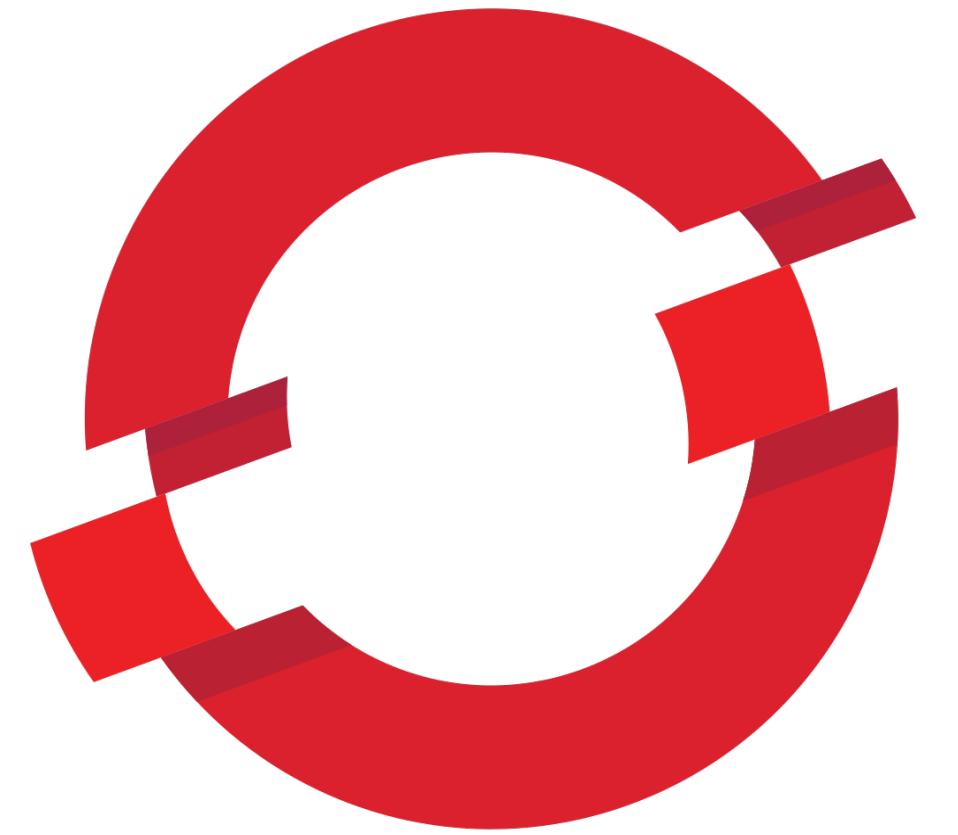


MESOS

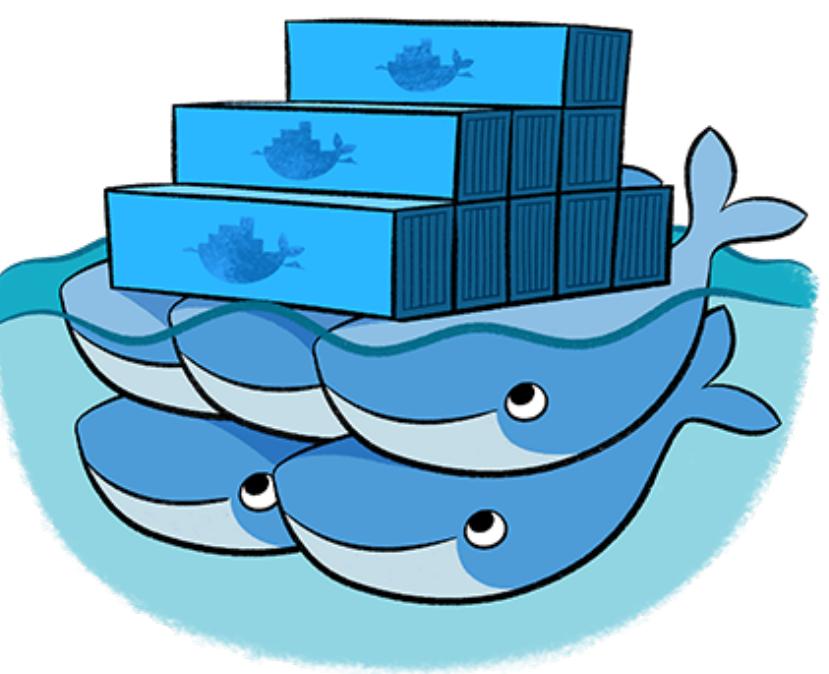


MARATHON

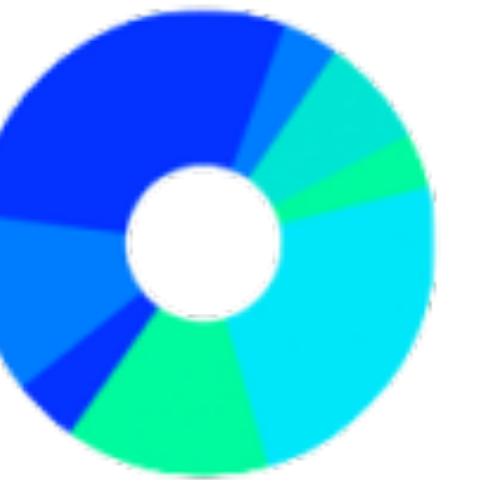




**OPENSHIFT**



**MESOS**



**MARATHON**



HashiCorp  
**Nomad**

OPEN



nad



nad

**WON'T SOMEBODY PLEASE THINK OF THE CHILDREN?!**



nad

**WON'T SOMEBODY PLEASE THINK OF THE CHILDREN?!**



nad

**WON'T SOMEBODY PLEASE THINK OF THE CHILDREN?!**



WON'T SOMEBODY PLEASE THINK OF THE CHILDREN?!

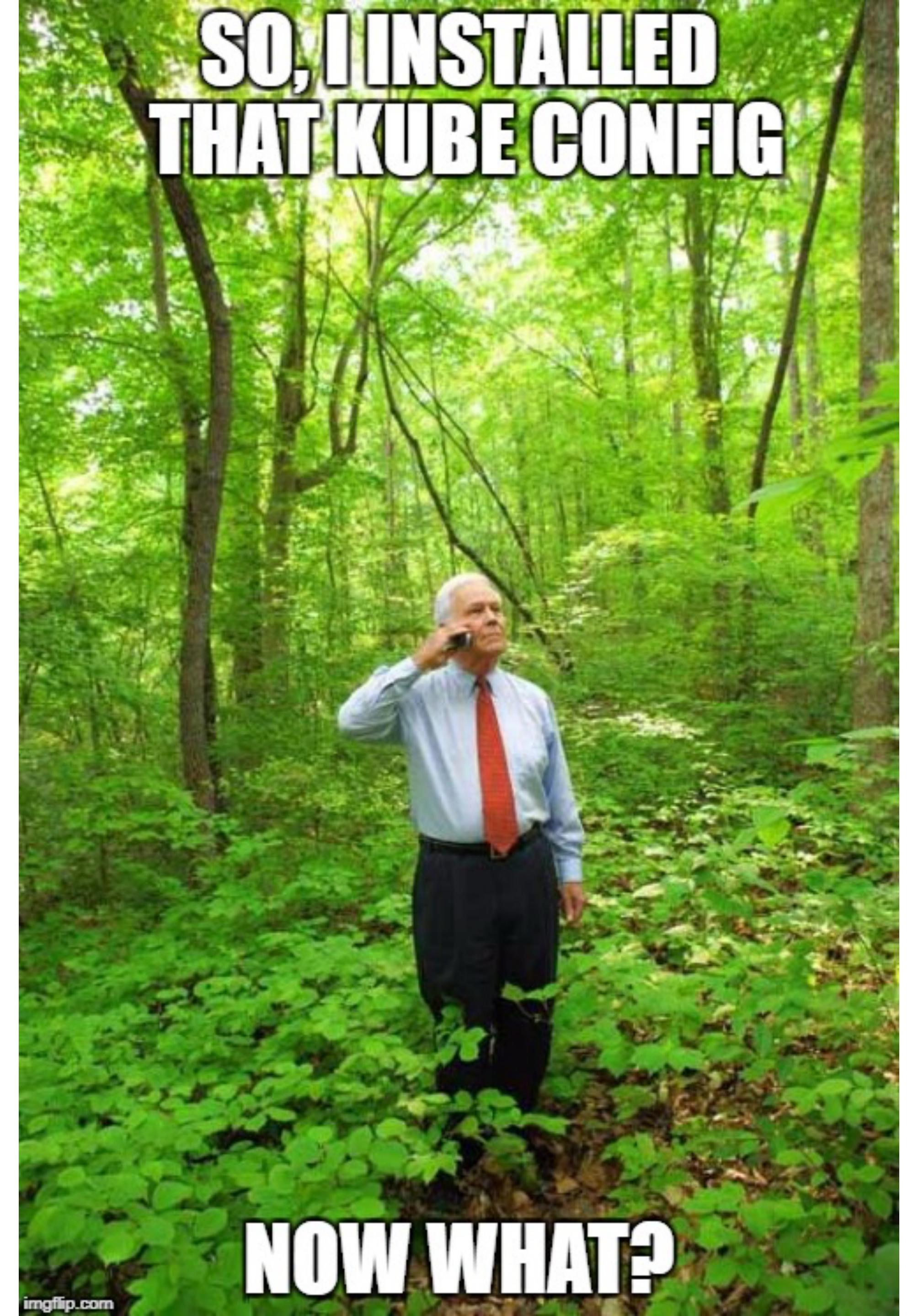




What's this k8s thing? Why I  
need that?

---

...please with the mouse click on "accept  
license agreement"...



# Minute 0 with k8s

---

- How I use my docker-compose?
- What the hell is Kompose? <https://kompose.io>
- What is all this deployment, service, ReplicaSet stuff?
- Hey! I found something where I can say the Docker image I want to use
- So, k8s doesn't replace Docker 🤔
- I can use rkt if I want! <https://coreos.com/rkt/>

```
1 apiVersion: extensions/v1beta1
2 kind: Deployment
3 metadata:
4   name: expresscart-deployment
5   namespace: stage
6   labels:
7     app: expresscart-deployment
8 spec:
9   replicas: 1
10  template:
11    metadata:
12      labels:
13        app: expresscart-deployment
14 spec:
15   containers:
16     - name: image
17       image: hachikoapp/expresscart
18       imagePullPolicy: Always
19   ports:
20     - containerPort: 1111
21       protocol: TCP
```

# Main architectural concerns when working with k8s

---

- What happens with my MySQL or MongoDB database?

- Deployments

- ReplicaSets

- Pods? Like the ones from Nespresso®?

- Services

- When I use that stuff?

---

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: expresscart-deployment
5   namespace: stage
6 spec:
7   selector:
8     app: expresscart-deployment
9   type: LoadBalancer
10  ports:
11    - protocol: TCP
12      port: 2025
13      targetPort: 1111
```

---

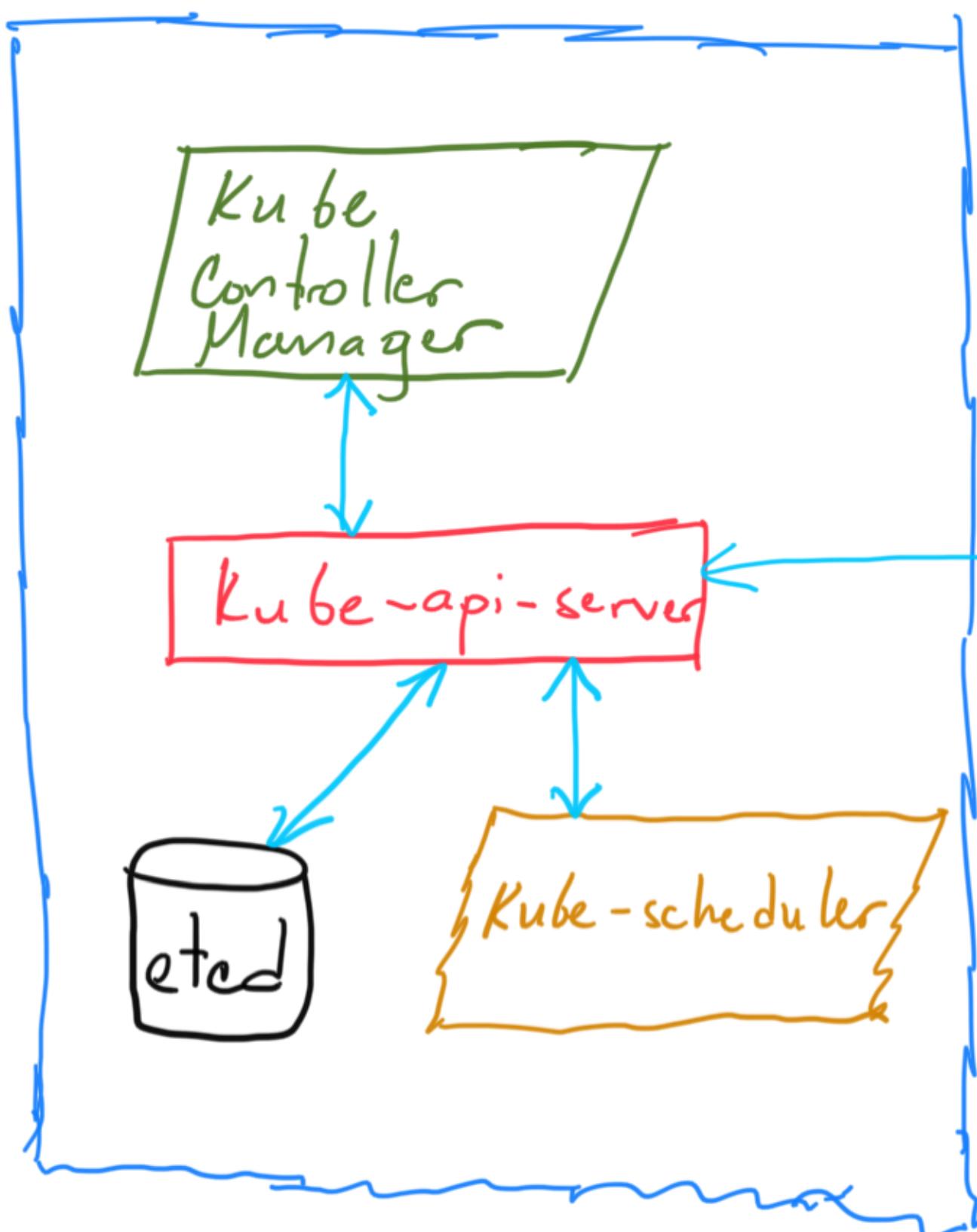
# Main architectural concerns when working with k8s

- Isolation of resources
  - Service discovery and load balancing
  - **Secret and configuration management!!!!**

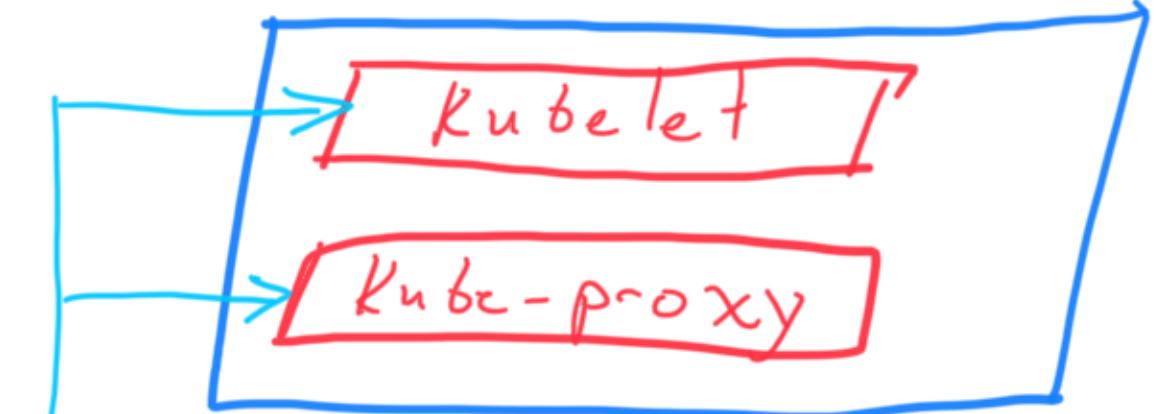
# So...what is k8s?

- Kubernetes is a system for running and coordinating containerized applications across a cluster of machines, its main responsibility is to completely manage the life cycle of containerized applications and services using strategies that provide predictability, scalability, and high availability.
- Kind of a task scheduler on steroids

K8S Control Plane



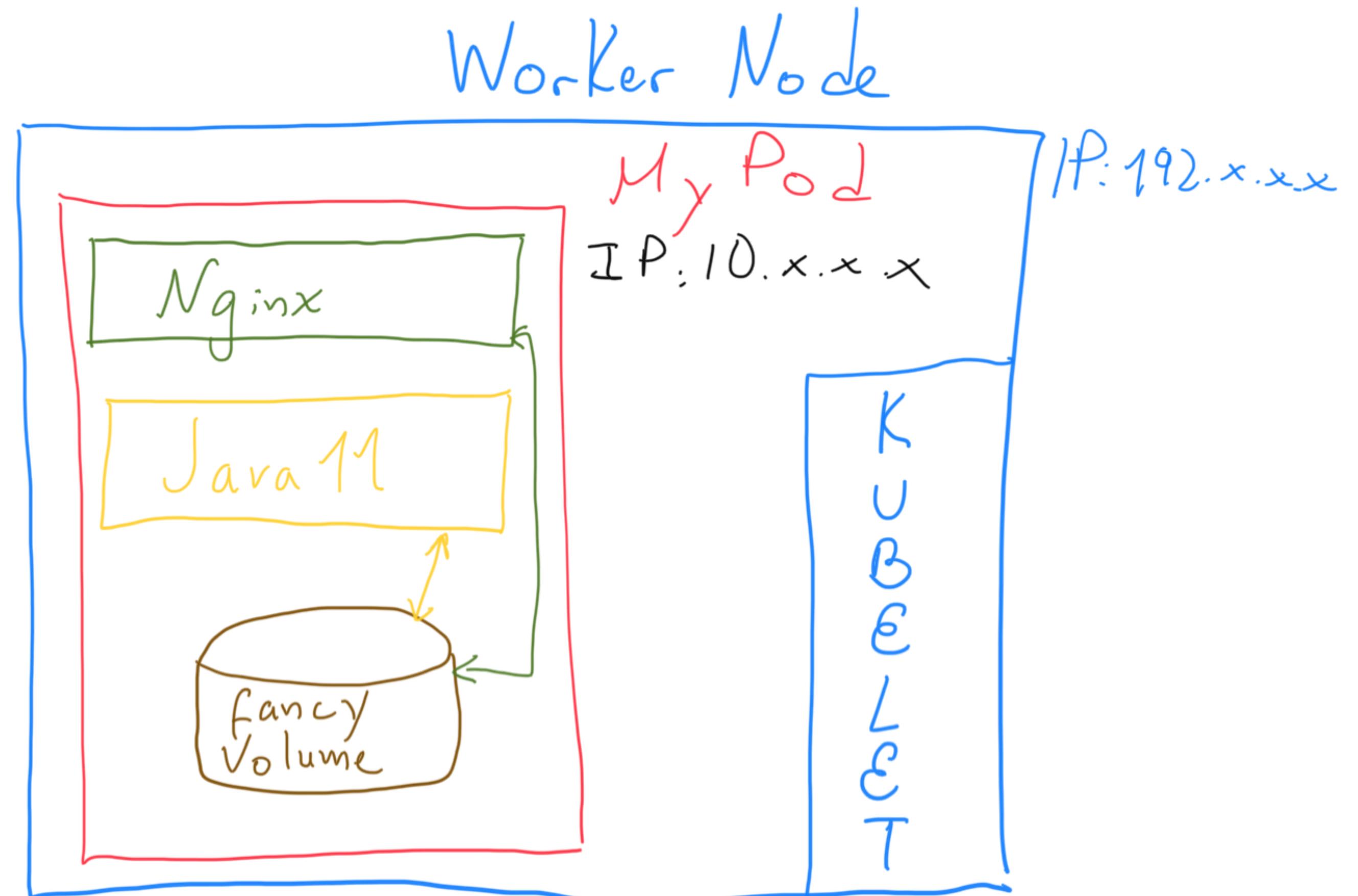
Minions (worker nodes)



- 
- 
- 
- 
- 
-

# Pods

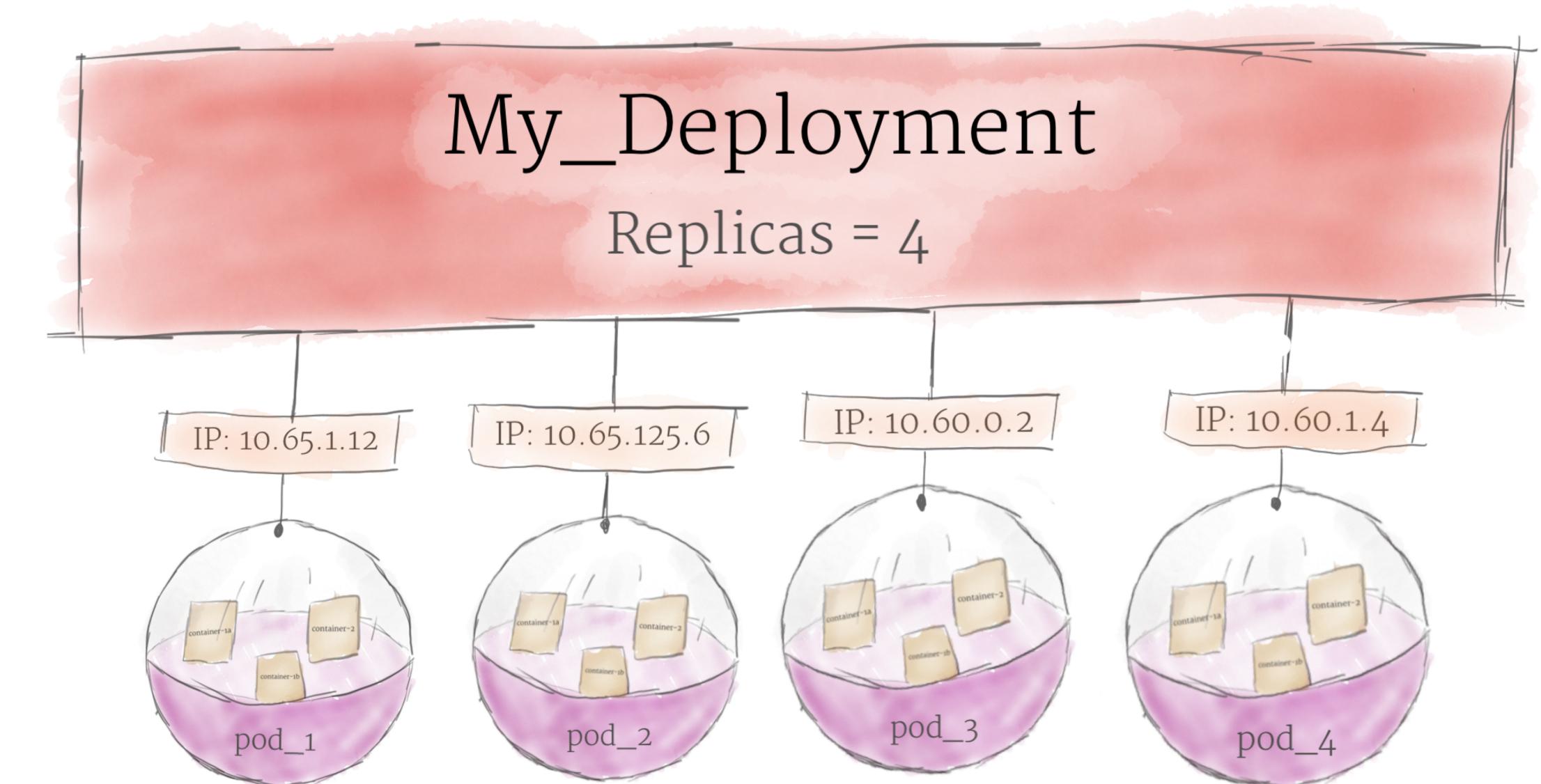
- It's the basic unit of k8s. It represents a process in execution inside the cluster. They're ephemeral, so when they die, and they will, the cluster will not do any attempt to resurrect them. They have:
  - An unique IP
  - Storage
  - Container's execution instructions
  - And one or more containers running inside.



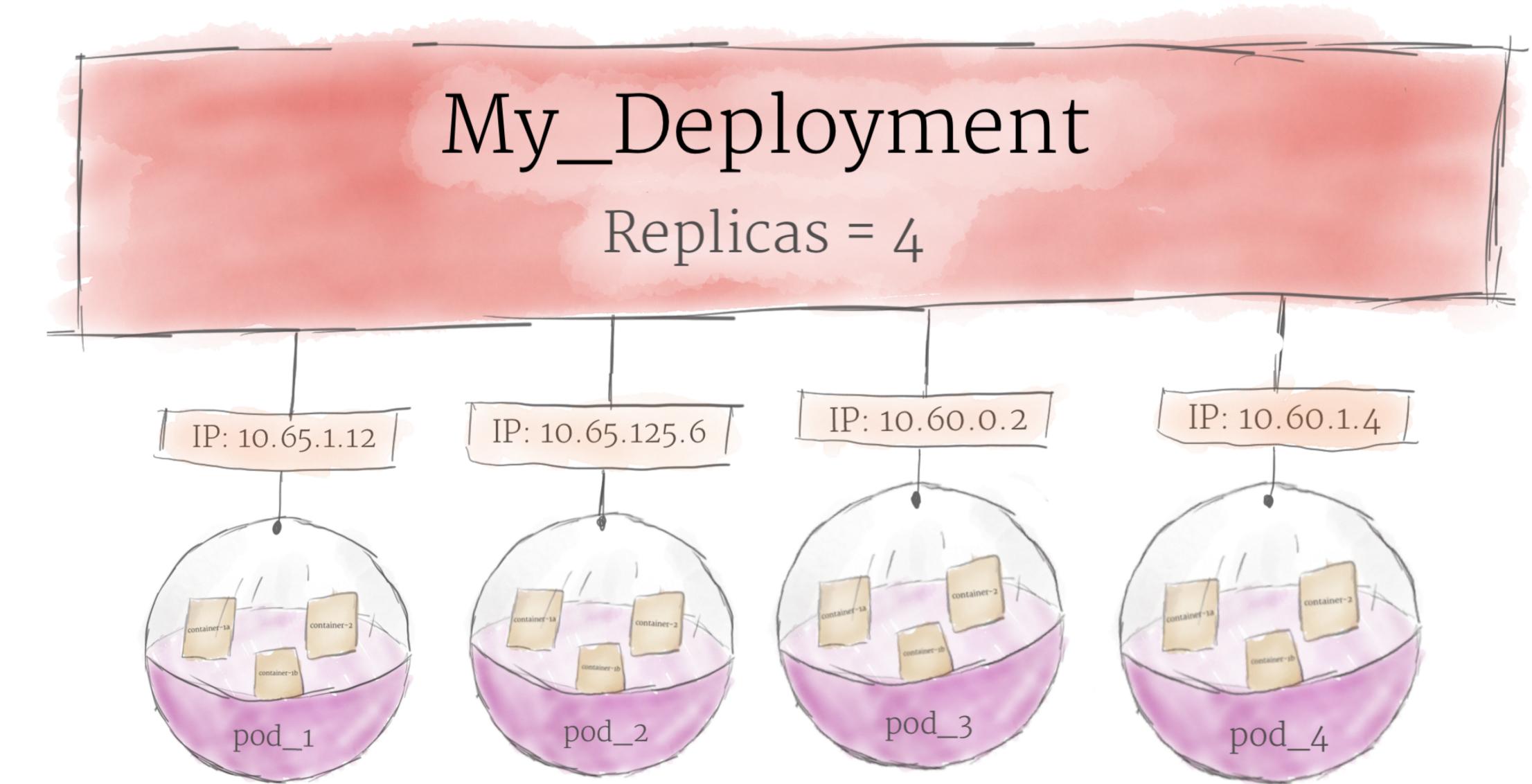
# ReplicaSets & Deployments

---

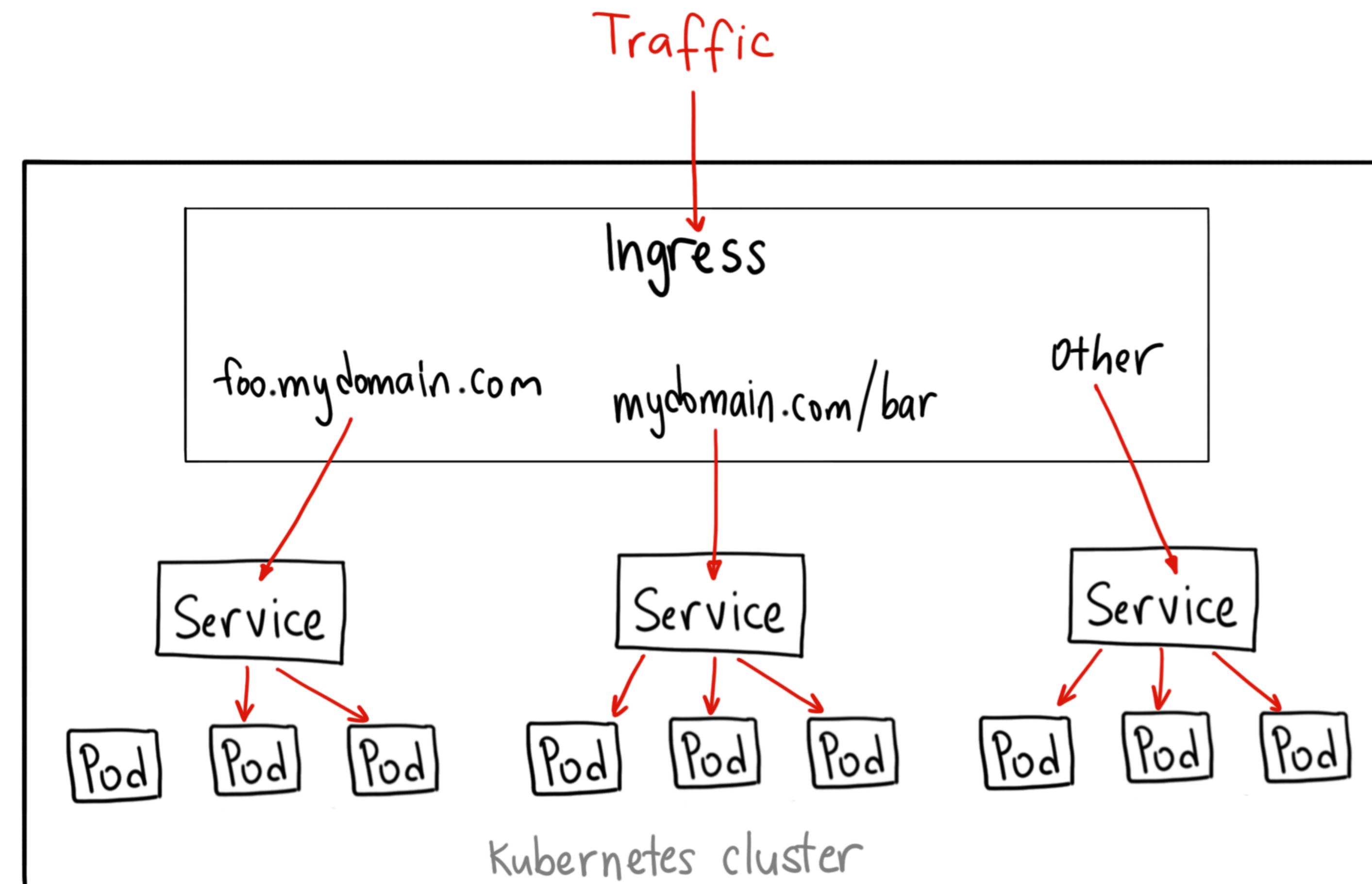
# ReplicaSets & Deployments



# ReplicaSets & Deployments



# Services



# Sweet, but how I start up this?

Please enter your credit card number

C gke-expresscart

ⓘ Cluster creation can take five minutes or more.

67% - Cluster health checks are running...

✓ Configuring

✓ Deploying

C Health checks

✗ Hide progress steps

Details Storage Nodes

## Cluster

Master version 1.15.8-gke.2

Endpoint 34.69.33.154

Client certificate Disabled

Binary Authorization Disabled

Kubernetes alpha features Disabled

Total size 0

Region us-central1

Node zones us-central1-b  
us-central1-f  
us-central1-a

Network gke-network

Subnet gke-subnet-01

VPC-native (alias IP) Enabled

Pod address range 192.168.64.0/20

Default maximum pods per node 110

Service address range 192.168.80.0/20

Intranode visibility Disabled

Stackdriver Kubernetes Engine Monitoring Enabled

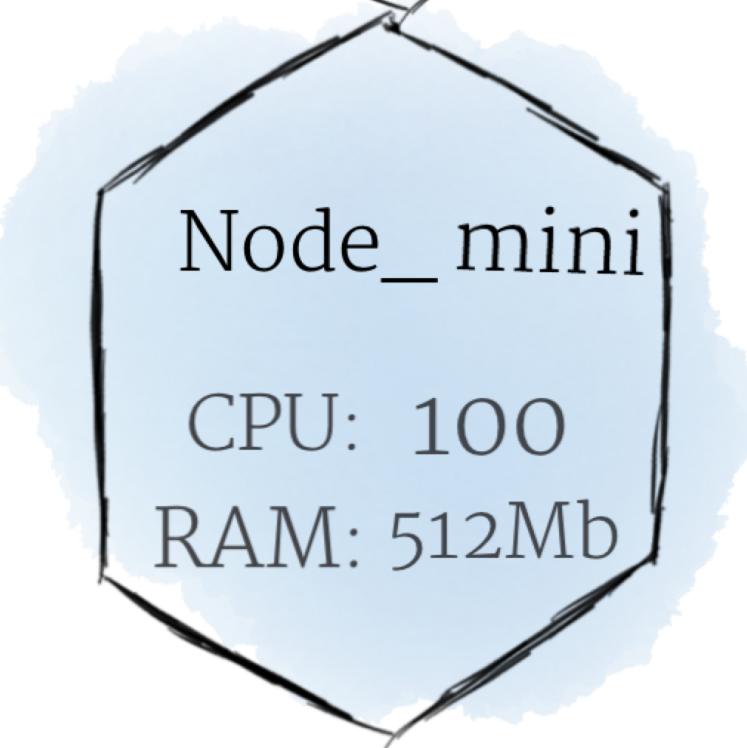
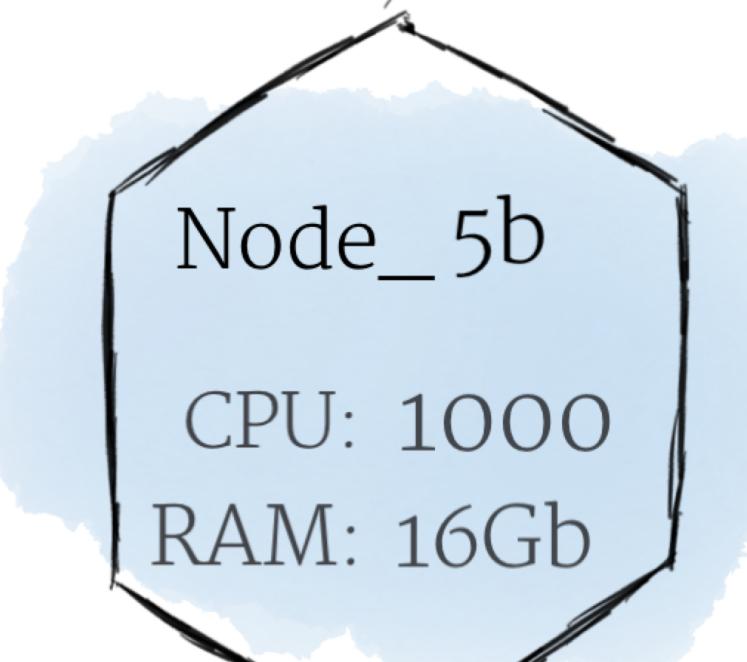
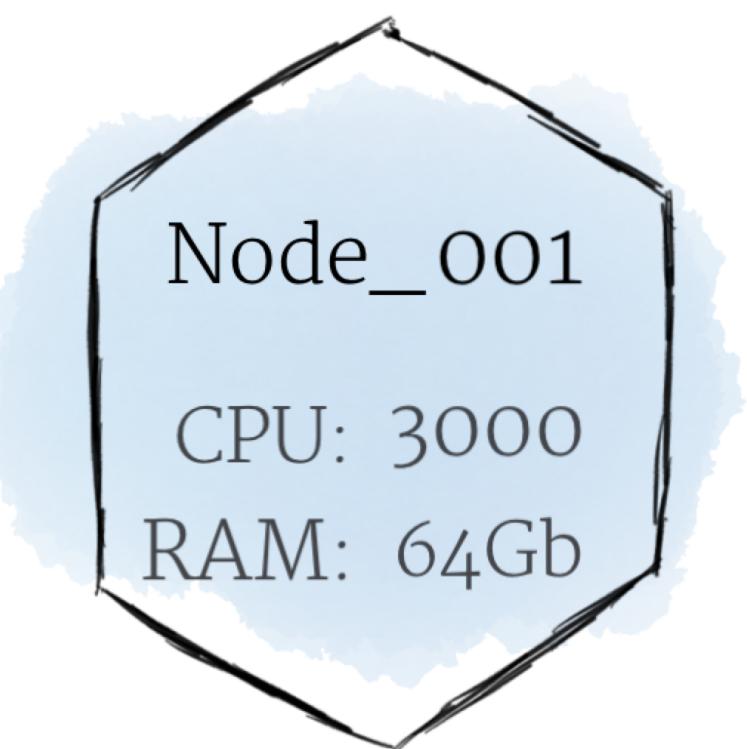
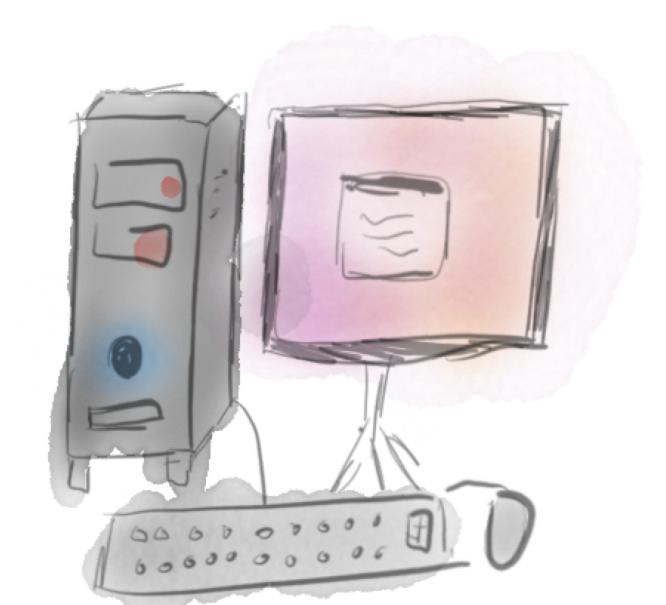
Legacy Stackdriver Logging Disabled

Legacy Stackdriver Monitoring Disabled

# Is not a easy task!

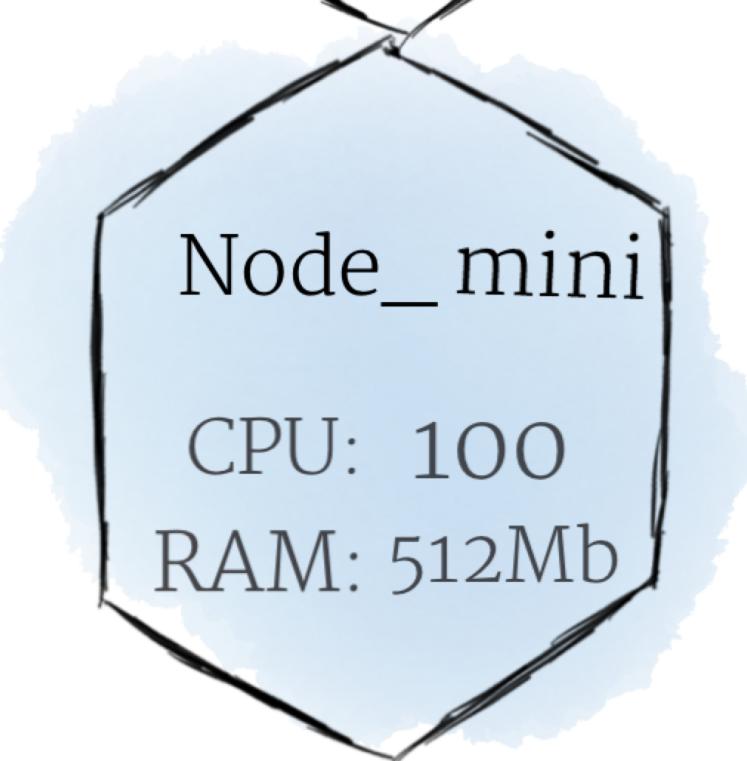
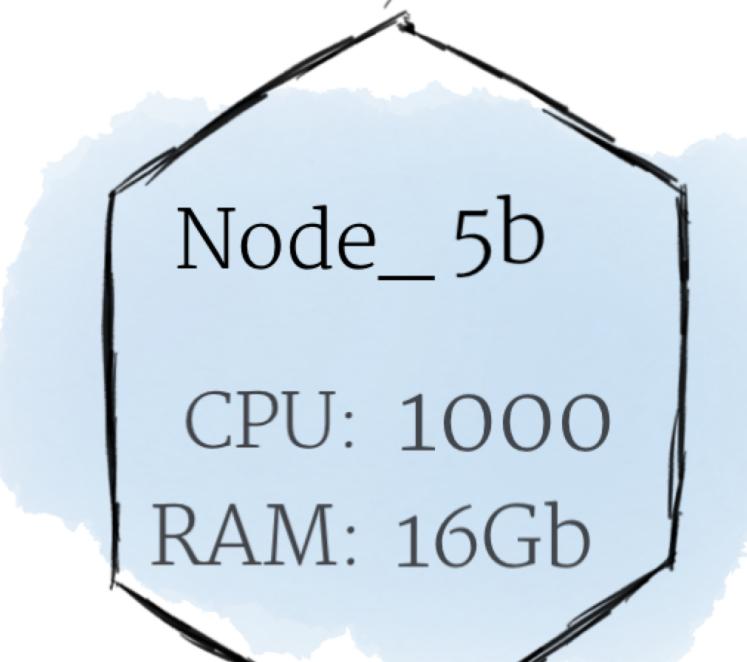
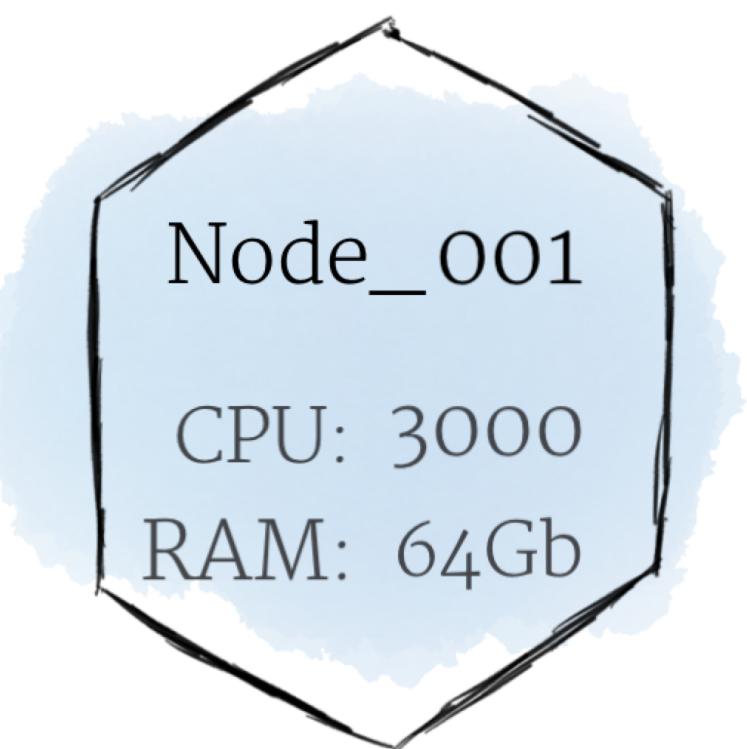
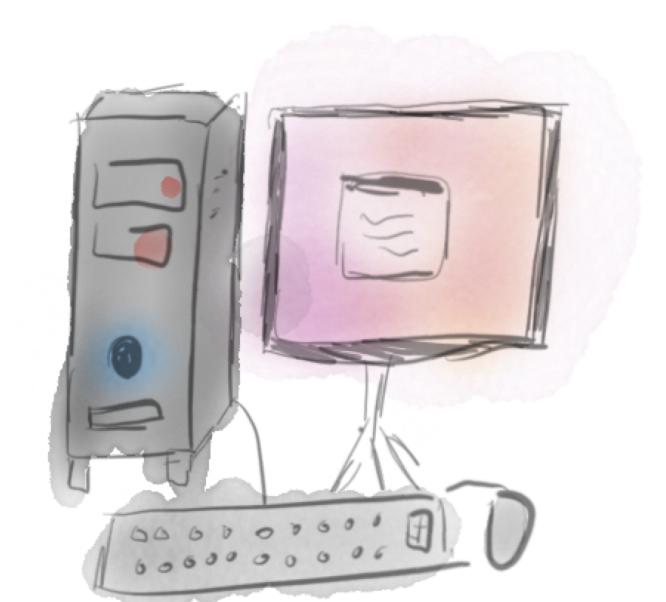
---

# Is not a easy task!



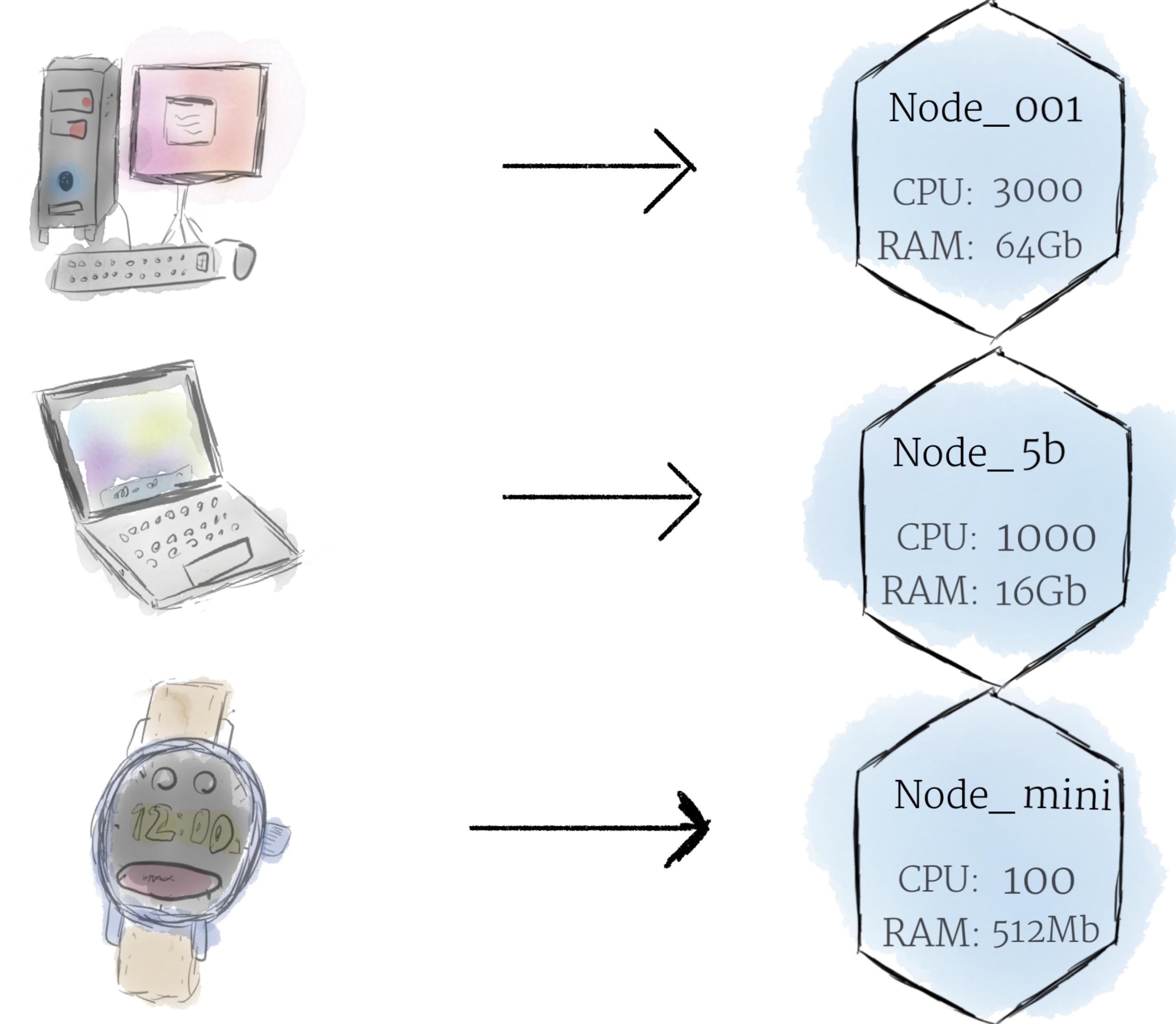
# Is not a easy task!

- You could:



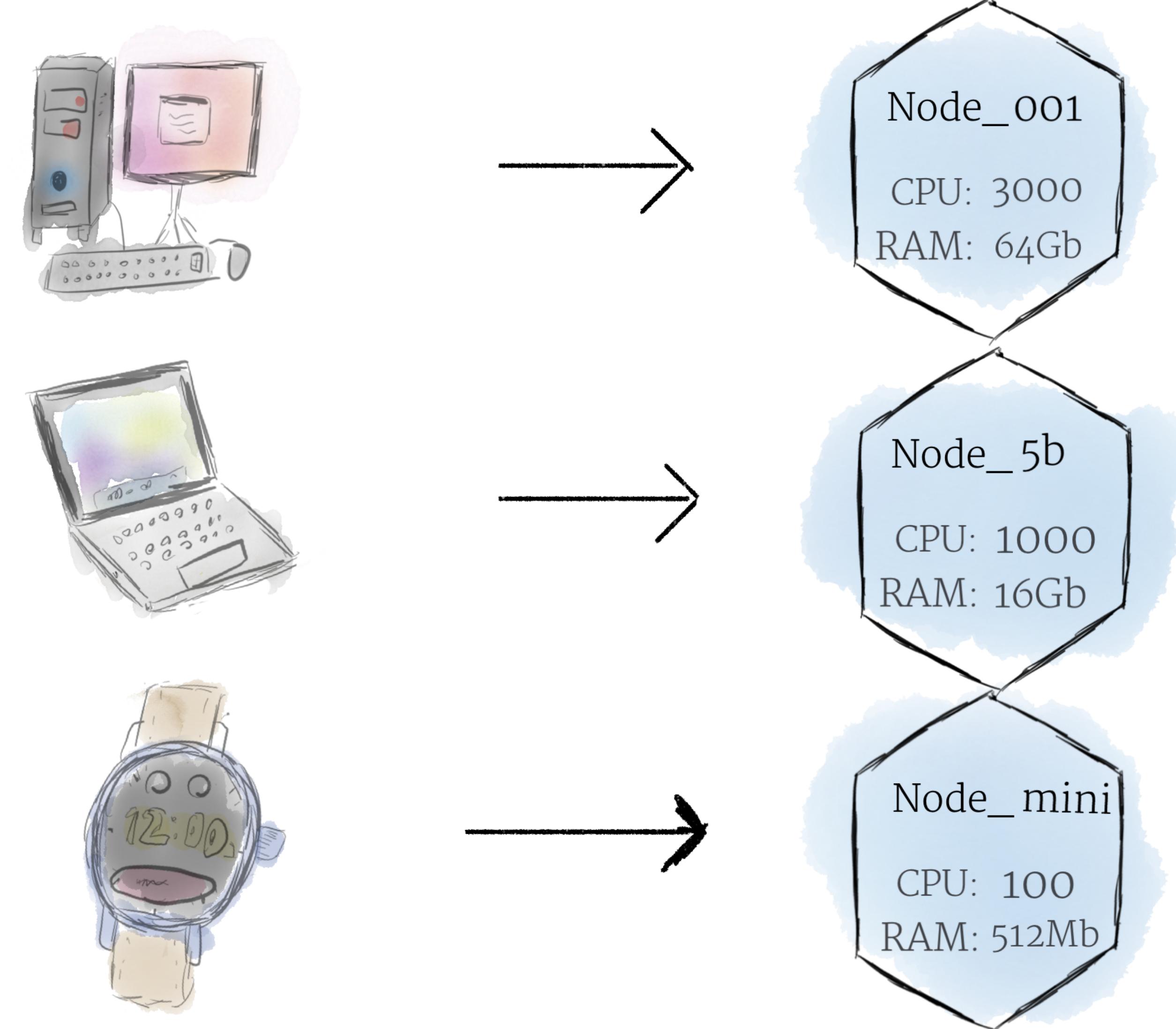
# Is not a easy task!

- You could:
  - Install it yourself: <https://github.com/kelseyhightower/kubernetes-the-hard-way>



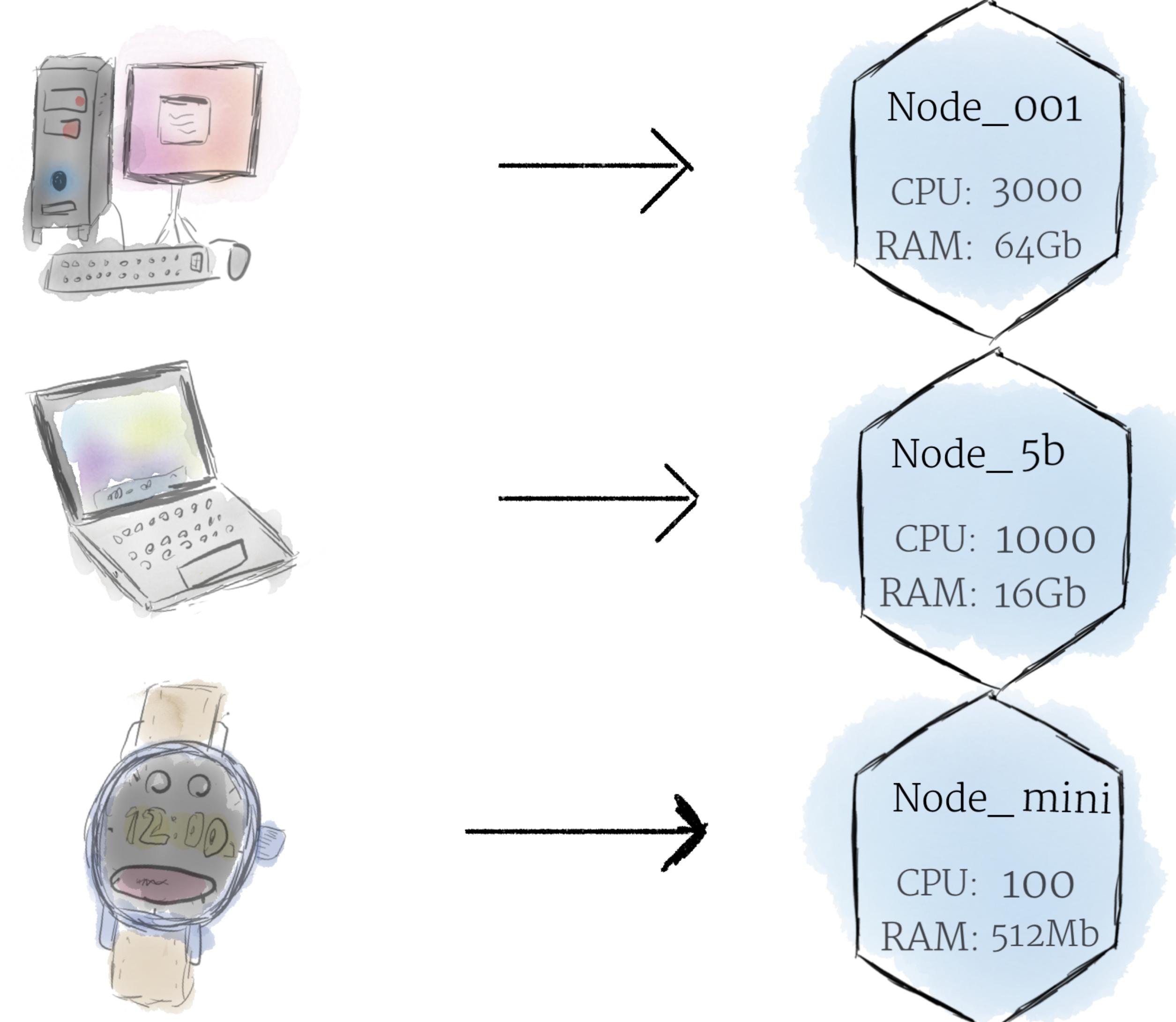
# Is not a easy task!

- You could:
  - Install it yourself: <https://github.com/kelseyhightower/kubernetes-the-hard-way>
  - Use kops: <https://kubernetes.io/docs/setup/production-environment/tools/kops/>



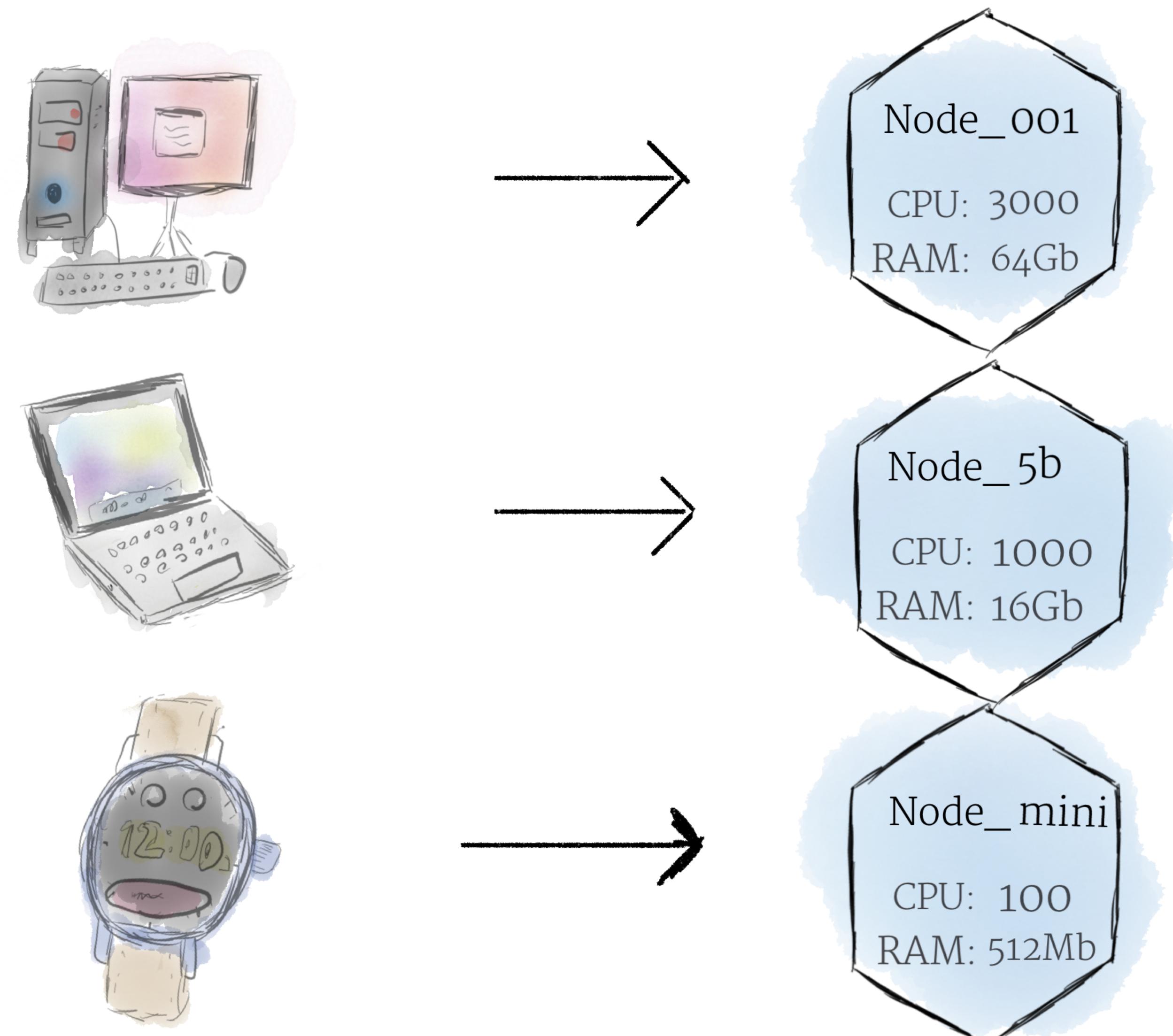
# Is not a easy task!

- You could:
  - Install it yourself: <https://github.com/kelseyhightower/kubernetes-the-hard-way>
  - Use kops: <https://kubernetes.io/docs/setup/production-environment/tools/kops/>
  - Use kubespray (Ansible!): <https://kubernetes.io/docs/setup/production-environment/tools/kubespray/>



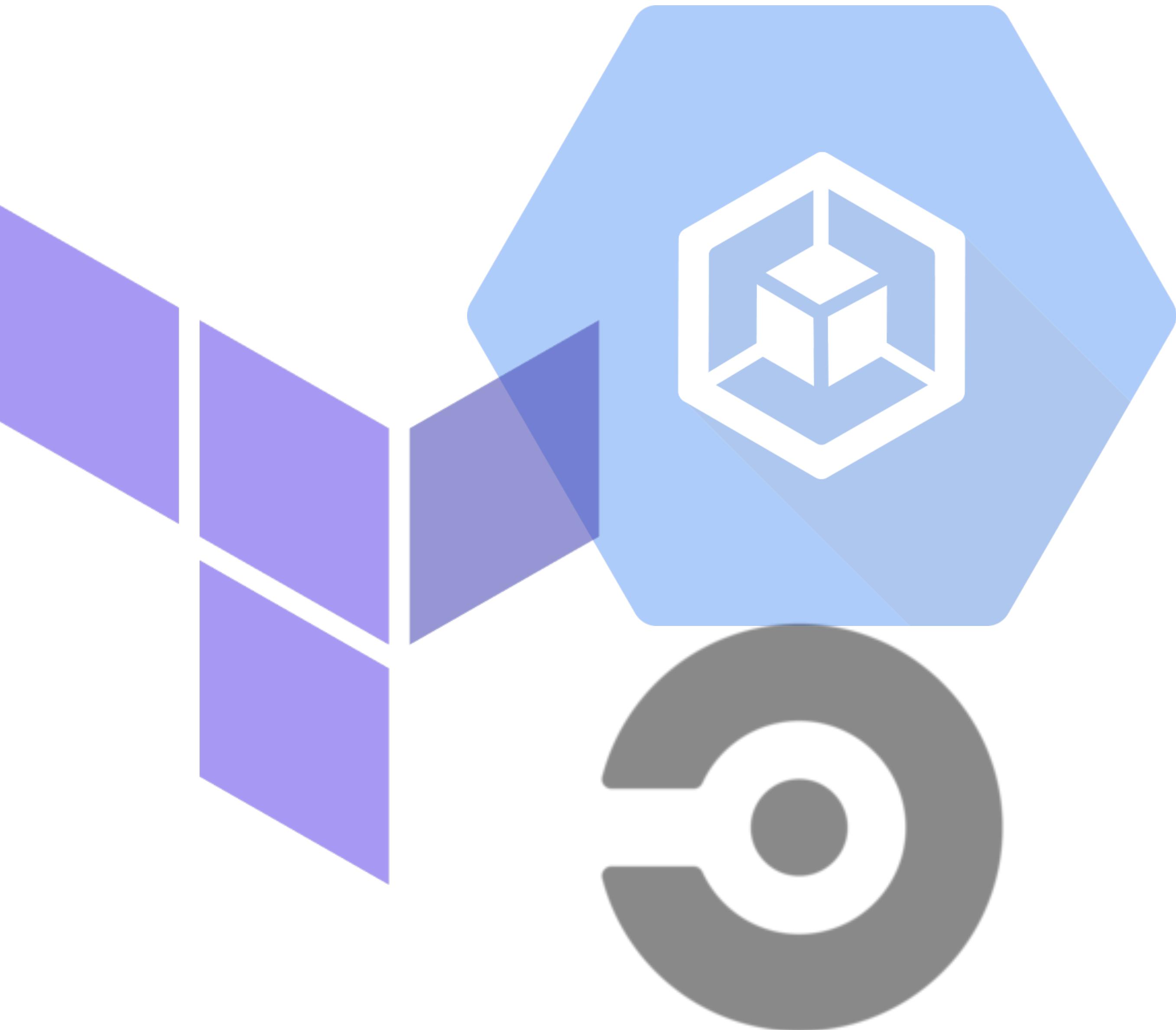
# Is not a easy task!

- You could:
  - Install it yourself: <https://github.com/kelseyhightower/kubernetes-the-hard-way>
  - Use kops: <https://kubernetes.io/docs/setup/production-environment/tools/kops/>
  - Use kubespray (Ansible!): <https://kubernetes.io/docs/setup/production-environment/tools/kubespray/>
  - Or...



# ...use a cloud provider!

- Yeah, we did that. With Google Kubernetes Engine: <https://cloud.google.com/kubernetes-engine>
- But also, more spicy ingredients:
  - GitHub - for GitOps! <https://www.weave.works/technologies/gitops/>
  - Terraform. <https://www.terraform.io>
  - MongoDB. <https://www.mongodb.com>
  - CircleCI. <https://circleci.com>
  - And a NodeJS App



# But, why?

- Show you a real example of how to deploy a Kubernetes infrastructure in Google Cloud Platform using GitOps strategy...
- Using infrastructure as code with terraform...
- ...all things deployed by CI/CD pipelines living in CircleCI,
- And, deploying a MongoDB cluster, without needing an external SaaS/PaaS provider (more on this later)
- And to prove how reliable k8s solutions could be...

```
# module.gke.google_container_cluster.primary will be created
+ resource "google_container_cluster" "primary" {
  + additional_zones          = (known after apply)
  + cluster_ipv4_cidr         = (known after apply)
  + default_max_pods_per_node = (known after apply)
  + enable_binary_authorization = false
  + enable_intranode_visibility = (known after apply)
  + enable_kubernetes_alpha    = false
  + enable_legacy_abac         = false
  + enable_tpu                 = (known after apply)
  + endpoint                  = (known after apply)
  + id                        = (known after apply)
  + instance_group_urls       = (known after apply)
  + location                  = "us-central1"
  + logging_service            = "logging.googleapis.com/kubernetes"
  + master_version             = (known after apply)
  + min_master_version         = "1.15.8-gke.2"
  + monitoring_service         = "monitoring.googleapis.com/kubernetes"
  + name                      = "gke-expresscart"
  + network                   = (known after apply)
  + node_locations             = [
    + "us-central1-a",
    + "us-central1-b",
    + "us-central1-f",
  ]
  + node_version               = (known after apply)
  + operation                  = (known after apply)
  + project                    = "kubernetes-2020-267312"
  + region                     = (known after apply)
  + remove_default_node_pool   = false
  + services_ipv4_cidr         = (known after apply)
  + subnetwork                 = (known after apply)
  + zone                       = (known after apply)
}
```

MongoDB Atlas Service: <https://www.mongodb.com/cloud/atlas>

Google Cloud Kubernetes Engine

So, what about costs?

Isn't cheaper to run this on a SaaS/PaaS?

**Cloud Provider & Region** GCP, Iowa (us-central1) >

**Cluster Tier** M30 (7.5 GB RAM, 40 GB Storage per Shard) >  
2 400 IOPS per Shard, Encrypted, Auto-expand Storage

**Additional Settings** MongoDB 4.2, Backup, 3 Shards  
Cloud Provider Snapshots >

**Cluster Name** Cluster0 >

**\$1.77/hour** Pay-as-you-go! You will be billed hourly and can terminate your cluster anytime. Excludes variable data transfer, backup, and taxes.

Back Create Cluster

MongoDB Atlas Service: <https://www.mongodb.com/cloud/atlas>

Google Cloud Kubernetes Engine

So, what about costs?

Isn't cheaper to run this on a SaaS/PaaS?

**Cloud Provider & Region** GCP, Iowa (us-central1) >

**Cluster Tier** M30 (7.5 GB RAM, 40 GB Storage per Shard) >  
2 400 IOPS per Shard, Encrypted, Auto-expand Storage

**Additional Settings** MongoDB 4.2, Backup, 3 Shards  
Cloud Provider Snapshots >

**Cluster Name** Cluster0 >

**\$1.77/hour** Pay-as-you-go! You will be billed hourly and can terminate your cluster anytime. Excludes variable [data transfer](#), [backup](#), and taxes.

[Back](#) [Create Cluster](#)

This screenshot shows the MongoDB Atlas service creation interface. It consists of several stacked cards. The top card is titled 'Cloud Provider & Region' and shows 'GCP, Iowa (us-central1)'. Below it is a 'Cluster Tier' card for 'M30 (7.5 GB RAM, 40 GB Storage per Shard)' with a note about 2 400 IOPS per Shard, Encrypted, Auto-expand Storage. The third card is 'Additional Settings' for MongoDB 4.2, Backup, 3 Shards, Cloud Provider Snapshots. The fourth card is 'Cluster Name' set to 'Cluster0'. At the bottom left is a large price indicator '\$1.77/hour' with a note about pay-as-you-go billing. On the right side of the bottom card are two buttons: 'Back' and 'Create Cluster'. A small note at the bottom states: 'Pay-as-you-go! You will be billed hourly and can terminate your cluster anytime. Excludes variable [data transfer](#), [backup](#), and taxes.'

MongoDB Atlas Service: <https://www.mongodb.com/cloud/atlas>

So, what about costs?

**Kubernetes Engine**

3 x MongoDB

2,190 total hours per month

Instance type: n1-standard-2

Region: Iowa

GCE Instance Cost: USD 235.64

Kubernetes Engine Cost: USD 0.00

Total available local SSD space 1x375 GB

[Sustained Use Discount](#): 30%

[Effective Hourly Rate](#): USD 0.108

**Estimated Component Cost: USD 235.64 per 1 month**

**Total Estimated Cost: USD 302.43 per 1 month**

Estimate Currency

USD - US Dollar

This screenshot shows the Google Cloud Kubernetes Engine cost estimation interface. It displays a breakdown of costs for a MongoDB deployment. It starts with 'Kubernetes Engine' and '3 x MongoDB' with edit and delete icons. It then lists '2,190 total hours per month' and specifies the 'Instance type: n1-standard-2' and 'Region: Iowa'. It details the 'GCE Instance Cost: USD 235.64' and 'Kubernetes Engine Cost: USD 0.00'. It also notes the 'Total available local SSD space 1x375 GB'. A 'Sustained Use Discount' of 30% is applied, leading to an 'Effective Hourly Rate' of USD 0.108. The total estimated cost is highlighted as 'USD 302.43 per 1 month'. At the bottom, there's a currency selection dropdown currently set to 'USD - US Dollar'.

Google Cloud Kubernetes Engine

Isn't cheaper to run this on a SaaS/PaaS?

**Cloud Provider & Region** GCP, Iowa (us-central1) >

**Cluster Tier** M30 (7.5 GB RAM, 40 GB Storage per Shard) >  
2 400 IOPS per Shard, Encrypted, Auto-expand Storage

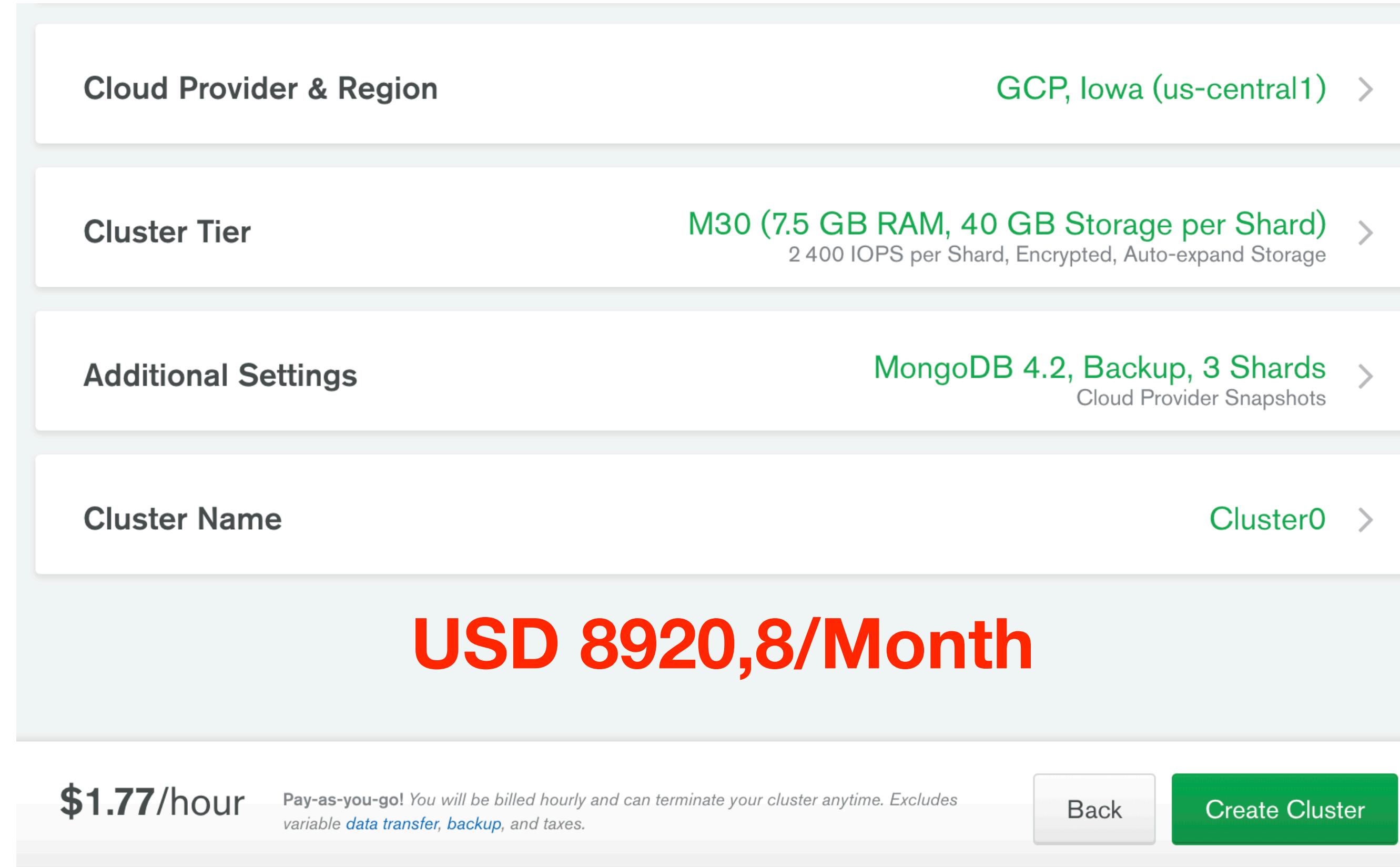
**Additional Settings** MongoDB 4.2, Backup, 3 Shards >  
Cloud Provider Snapshots

**Cluster Name** Cluster0 >

**USD 8920,8/Month**

**\$1.77/hour** Pay-as-you-go! You will be billed hourly and can terminate your cluster anytime. Excludes variable [data transfer](#), [backup](#), and taxes.

[Back](#) [Create Cluster](#)



MongoDB Atlas Service: <https://www.mongodb.com/cloud/atlas>

So, what about costs?

**Kubernetes Engine**

3 x MongoDB

2,190 total hours per month

Instance type: n1-standard-2

Region: Iowa

GCE Instance Cost: USD 235.64

Kubernetes Engine Cost: USD 0.00

Total available local SSD space 1x375 GB

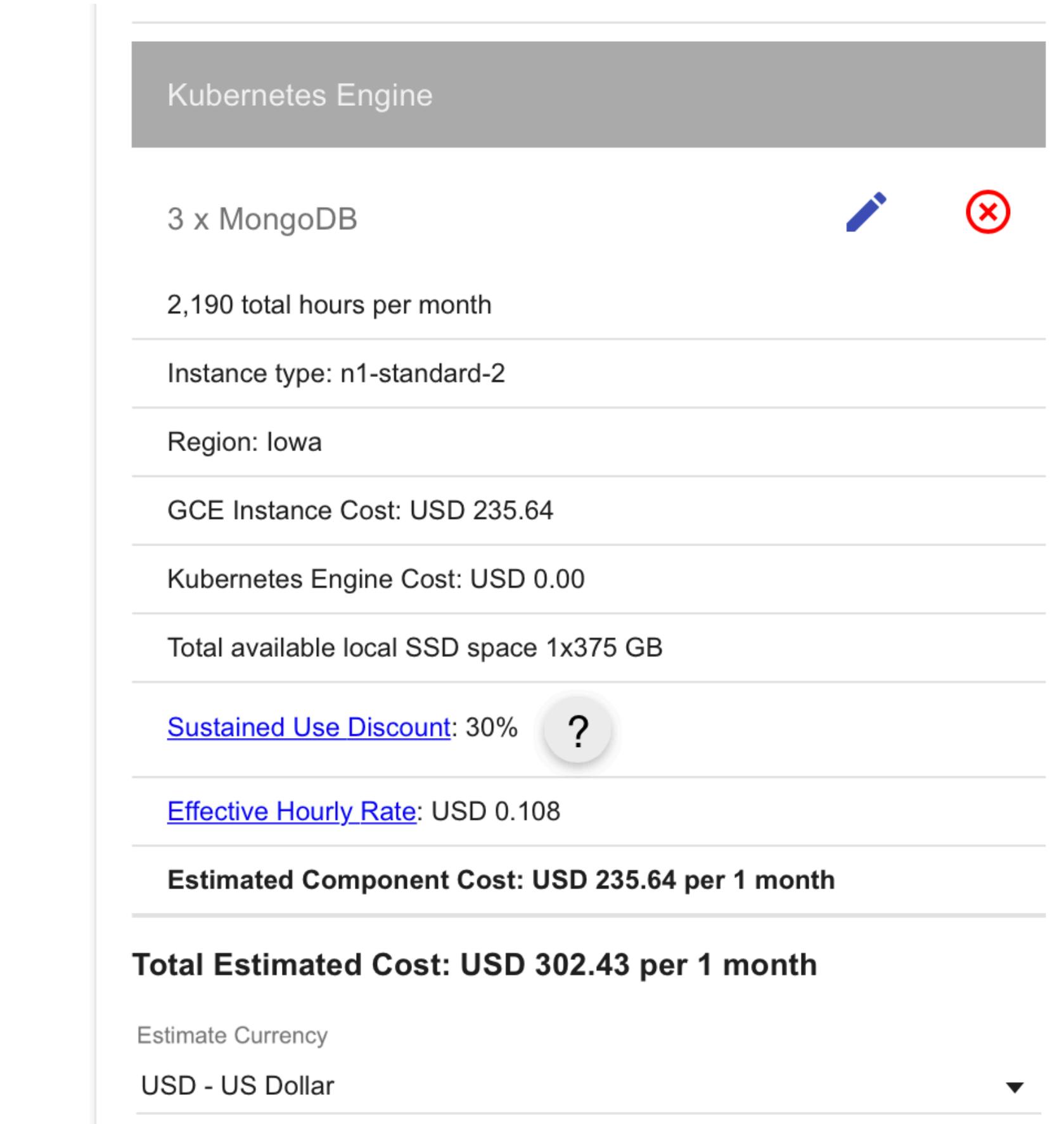
[Sustained Use Discount](#): 30%

[Effective Hourly Rate](#): USD 0.108

**Estimated Component Cost: USD 235.64 per 1 month**

**Total Estimated Cost: USD 302.43 per 1 month**

Estimate Currency  
USD - US Dollar



Google Cloud Kubernetes Engine

Isn't cheaper to run this on a SaaS/PaaS?

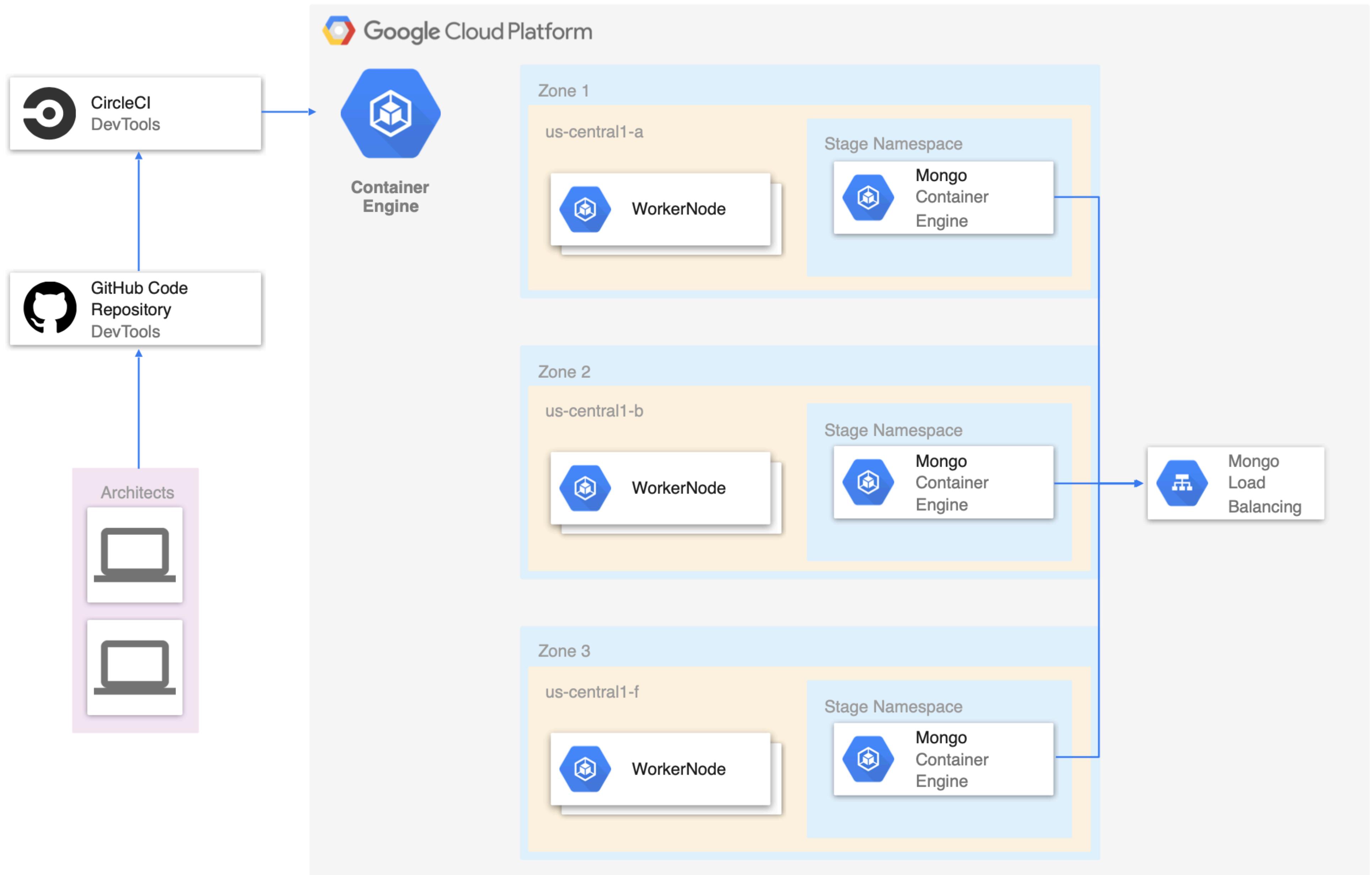
You have free samples?

---

PARENTAL ADVISORY: This demo isn't production ready!

I THINK WE MAY NEED TO  
UPDATE OUR DISASTER RECOVERY PLAN.  
THIS ONE SUGGESTS WE ALL RUN  
AROUND IN CIRCLES SHOUTING  
'WHAT DO WE DO?!!' 'WHAT DO WE DO?!!'

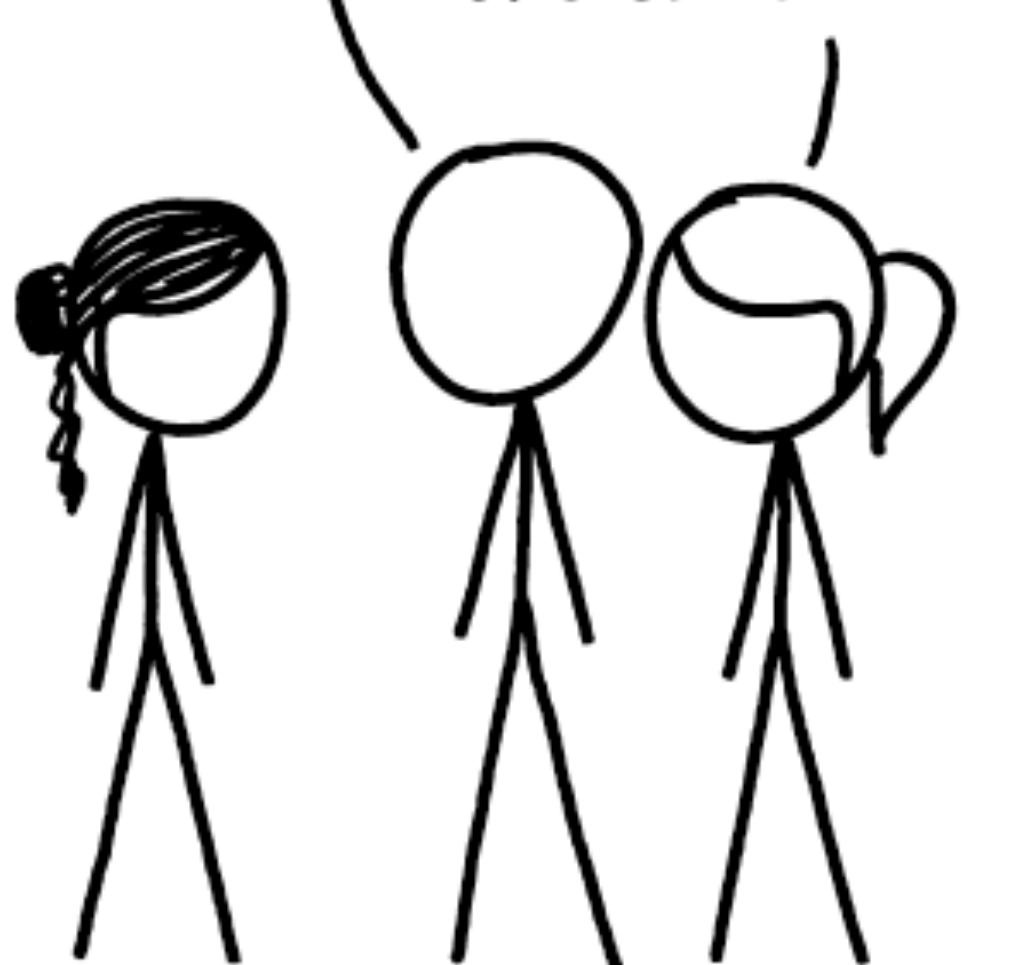




RAID CONTROLLERS DON'T  
MAKE SENSE AT OUR SCALE;  
EVERYTHING IS REDUNDANT  
AT HIGHER LEVELS. WHEN A  
DRIVE FAILS, WE JUST THROW  
AWAY THE WHOLE MACHINE.



MACHINE? WE THROW  
AWAY WHOLE RACKS  
AT A TIME.



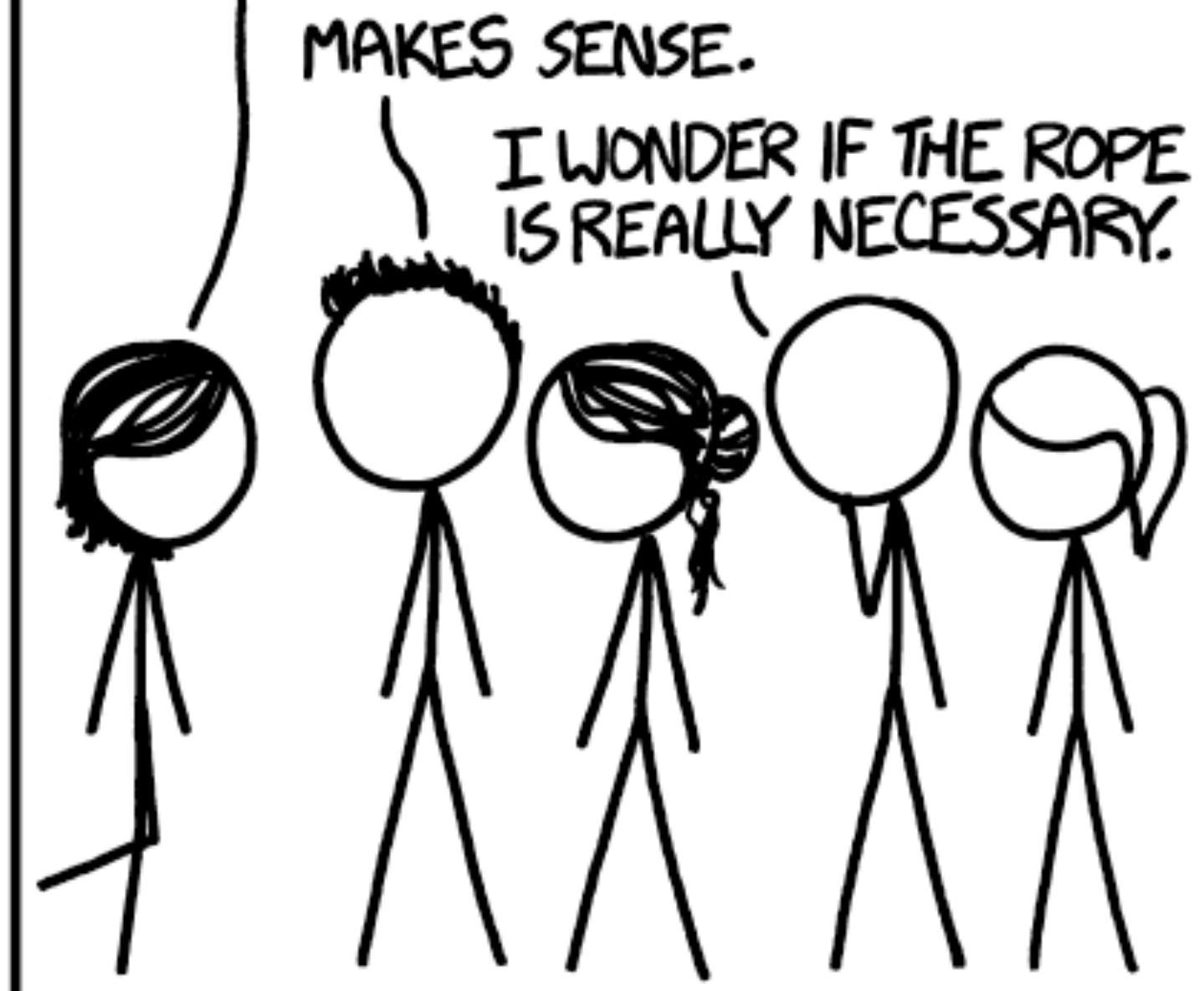
YEAH, WHO  
REPLACES  
ONE SERVER?

WE JUST REPLACE  
WHOLE ROOMS AT  
ONCE. AT OUR SCALE,  
MESSING WITH RACKS  
ISN'T ECONOMICAL.



WOW.  
LIKE GOOGLE!

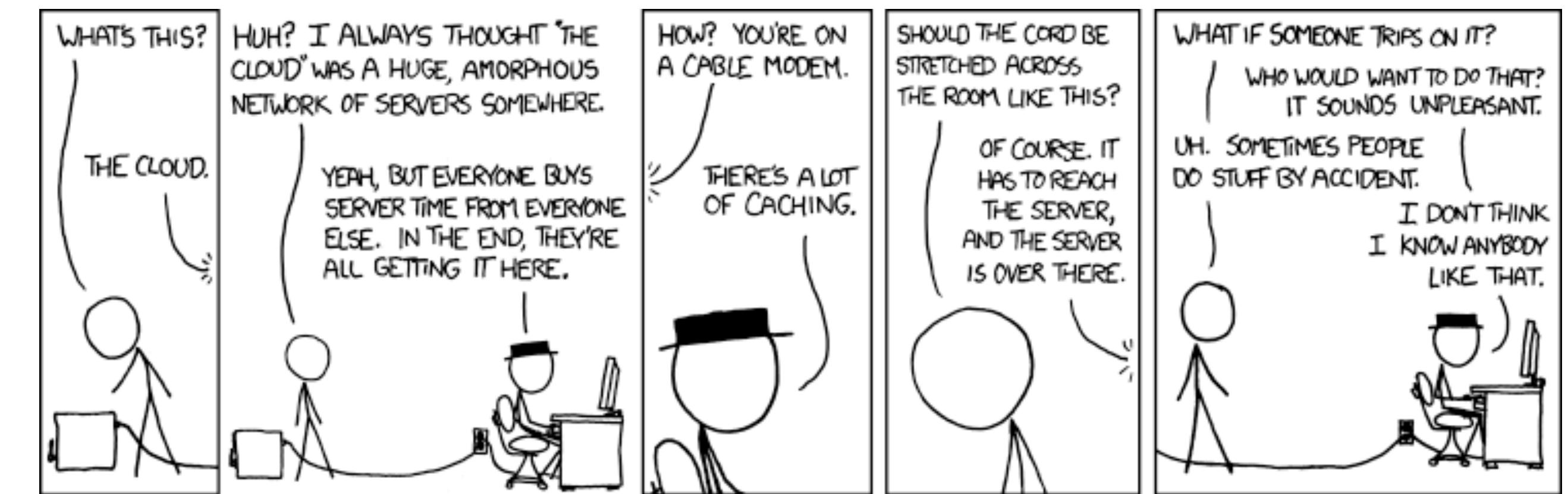
WE DON'T HAVE SPRINKLERS  
OR INERT GAS SYSTEMS.  
WHEN A DATACENTER CATCHES  
FIRE, WE JUST ROPE IT OFF  
AND REBUILD ONE TOWN OVER.



MAKES SENSE.  
I WONDER IF THE ROPE  
IS REALLY NECESSARY.

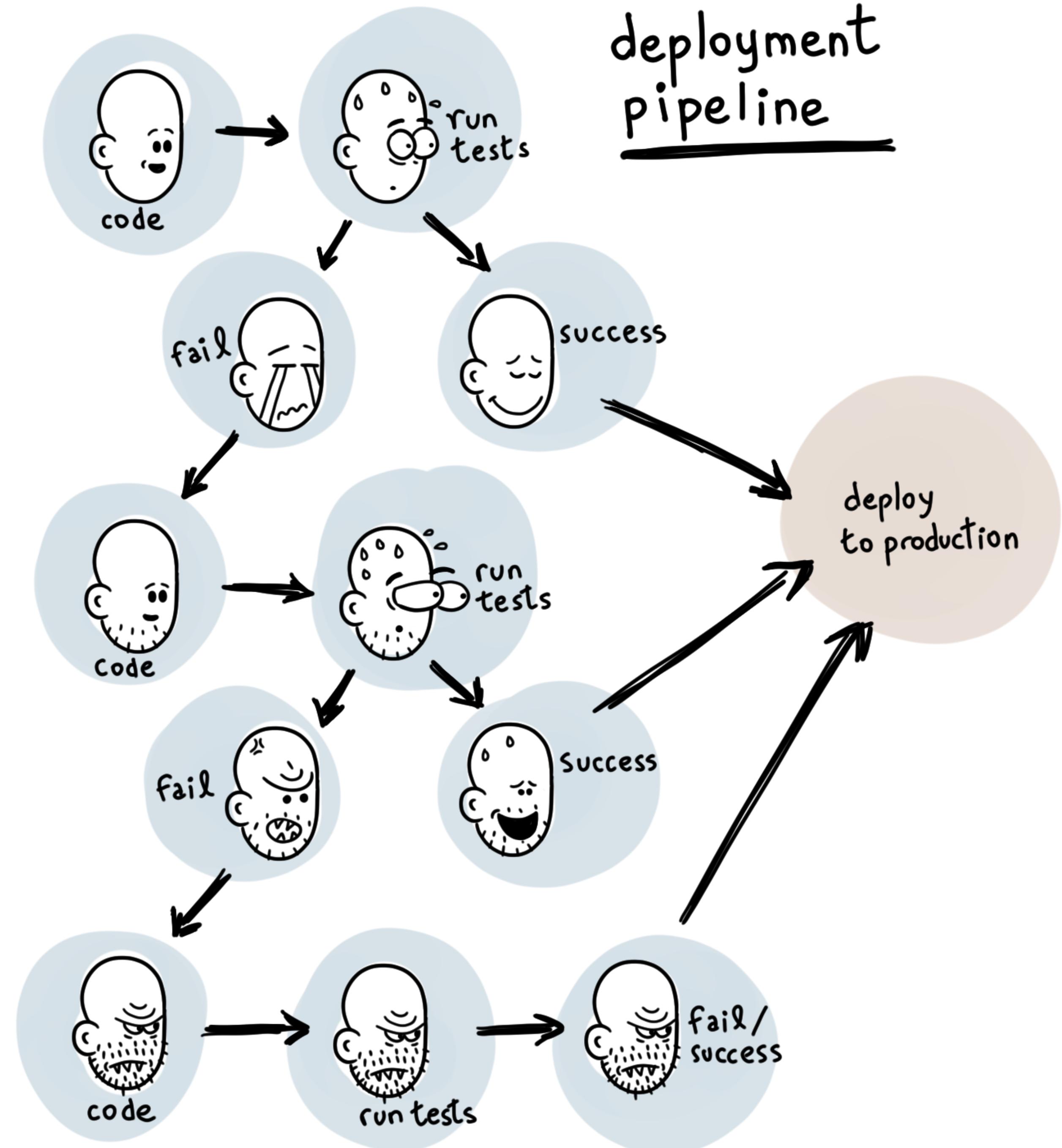
# Is all fun and games until...

Someone finds the power cord, with their feet



# Main issues you certainly will face

- Learn YAML! (well you should have at least carry your ruler)
- Complexity of the architecture
- Configuring External Load Balancers (if you aren't using managed K8s)
- Pipeline integration
- Monitoring madness
- Health checking everything (or nothing)
- Hey, what about deployment strategies!



# Issues WE faced

- The concept of reading, and therefore, following documentation
- CircleCI nonsense
- Terraform haywire
- Can we get more of those cloud resources?
- Sometimes it won't deploy, no matter what you do...
- ...so, plan for that; you could become multi-cloud perhaps?



Daniel Storj {turnoff.us}

So, what should I do next?

---



GOOD NIGHT,  
PROGRAMMERS!

{turnoff.us}

# Next challenges

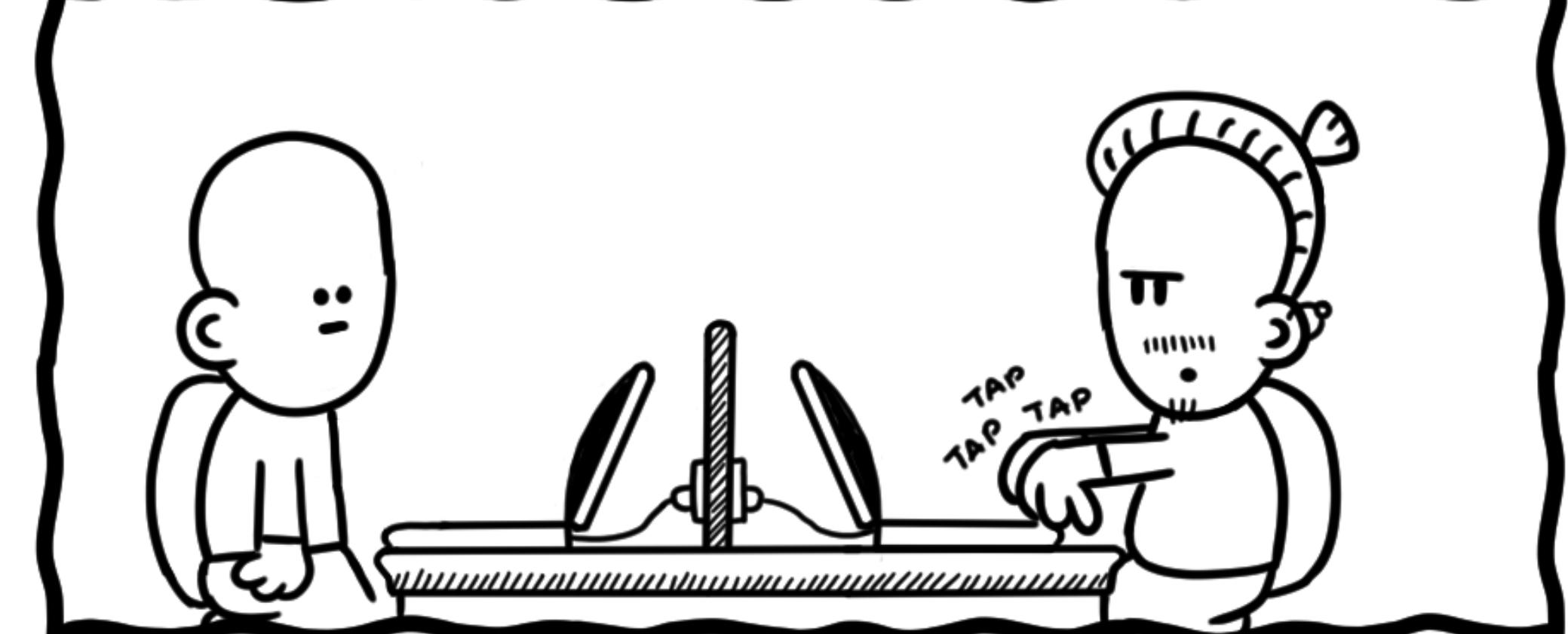
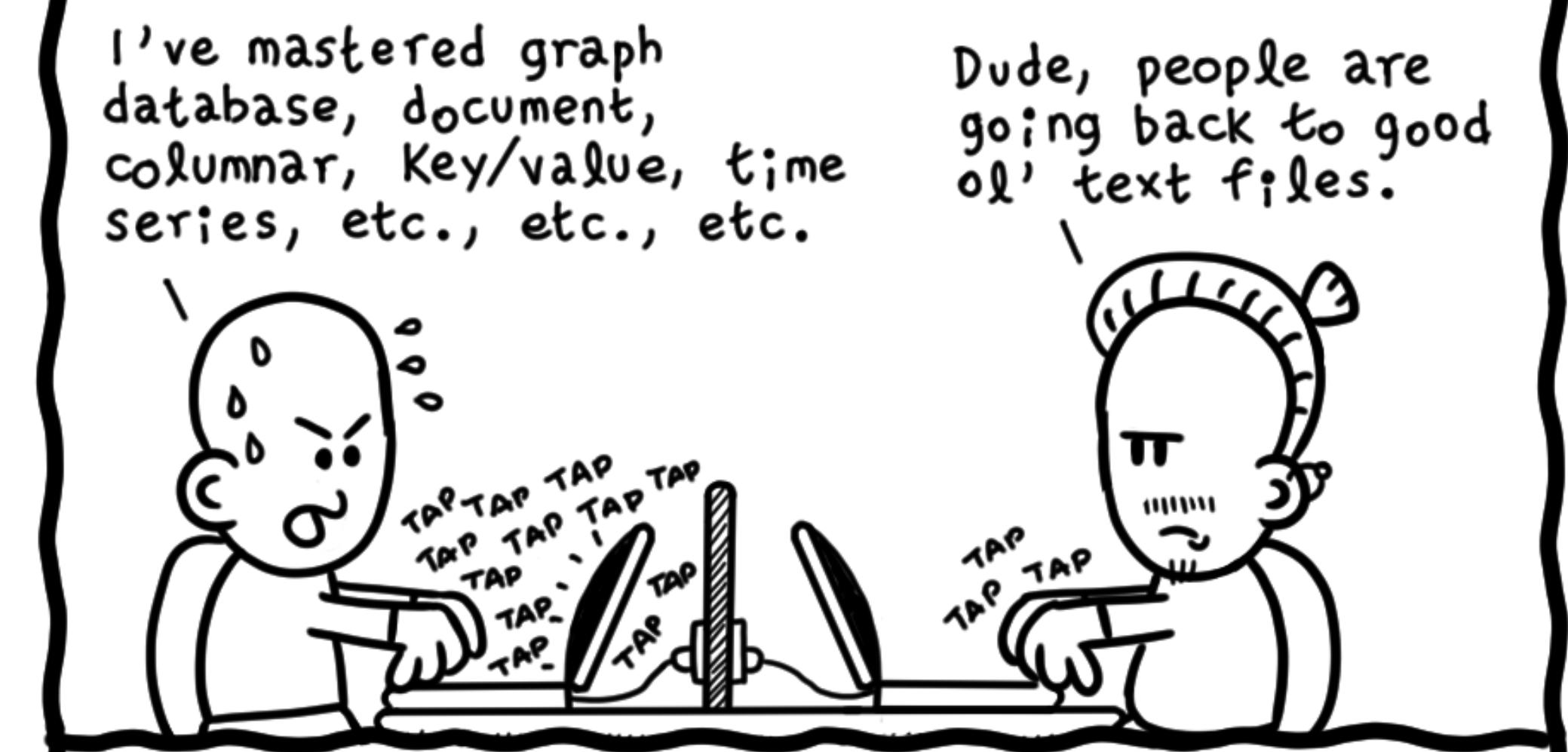
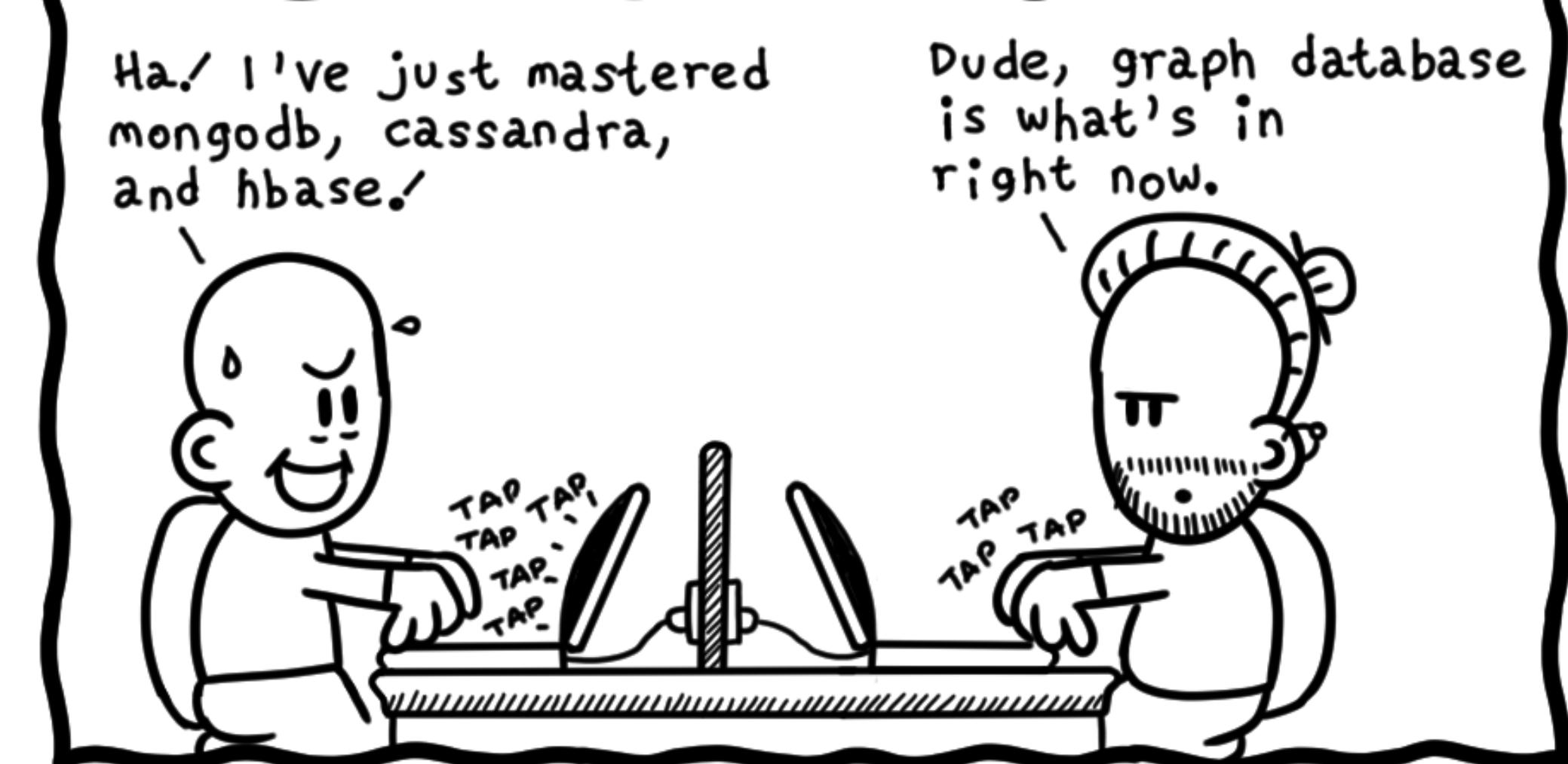
---

- You don't have to run everything in Kubernetes!
- K8s overhead can be expensive for small deployments
- Pack, ship and sail with Helm <https://helm.sh>
- You might well consider Spinnaker: <https://www.spinnaker.io>
- Cloud managed solutions are perfect starters, but if it grows can be problematic
- You could start with AWS ECS
- EKS+Fargate!...even FaaS?



# Next challenges

- K8s is a cool new thing, is a buzzword but...it's quite cumbersome to learn and master
- Don't restrict yourself to online tutorials and content
- Try to break things by yourself
- There are many things that could go wrong and possibly, you won't find anything related
- So ride the bull(bomb)!



# Got questions?

---

## Repository Information:

<https://github.com/williammuozr/terraform-google-gke-mongodb>

<https://github.com/williammuozr/expressCart>

## Cartoons from:

[xkcd.com](http://xkcd.com), <http://turnoff.us>

COMMENT	DATE
O CREATED MAIN LOOP & TIMING CONTROL.	14 HOURS AGO
O ENABLED CONFIG FILE PARSING	9 HOURS AGO
O MISC BUGFIXES	5 HOURS AGO
O CODE ADDITIONS/EDITS	4 HOURS AGO
O MORE CODE	4 HOURS AGO
O HERE HAVE CODE	4 HOURS AGO
O AAAAAAAA	3 HOURS AGO
O ADKFJSLKDFJSOKLFJ	3 HOURS AGO
O MY HANDS ARE TYPING WORDS	2 HOURS AGO
O HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT  
MESSAGES GET LESS AND LESS INFORMATIVE.

# Important related documentation

---

- <https://kubernetes.io/docs/concepts/>
- <https://circleci.com/docs/>
- <https://cloud.google.com/kubernetes-engine/docs/quickstart?hl=en>
- <https://cloud.google.com/kubernetes-engine/docs/tutorials/migrating-node-pool>
- <https://www.terraform.io/intro/index.html>
- <https://registry.terraform.io/modules/terraform-google-modules/kubernetes-engine/google/7.0.0>
- <https://www.terraform.io/docs/providers/google/index.html>
- <https://docs.mongodb.com/kubernetes-operator/master/tutorial/deploy-sharded-cluster/>



# THANK YOU!

## CARLOS ANDRÉS HERRERA

DEVOPS ENGINEER

[carlos.herrera@endava.com](mailto:carlos.herrera@endava.com)

<https://github.com/Carlos4ndresh/>

<https://www.linkedin.com/in/carlosandresherrera/>

## WILLIAM MUÑOZ RODAS

SENIOR DEVOPS ENGINEER

[wiliam.munoz@endava.com](mailto:wiliam.munoz@endava.com)

<https://github.com/williammunozr>

<https://www.linkedin.com/in/williammunozr/>