CLOUD NATIVE CON
Europe 2017

KubeCon
A CNCF EVENT

Autoscaling in Kubernetes

Marcin Wielgus, Senior Software Engineer, Google

CLOUD NATIVE CON
Europe 2017

KubeCon
A CNCF EVENT

Autoscaling in Kubernetes

Marcin Wielgus, Senior Software Engineer, Google

KubeCon
A CNCF EVENT

“ How many? „

“ I don’t know „

“ I think I need... „

Avg. utilization

How big is it?

15% utilization

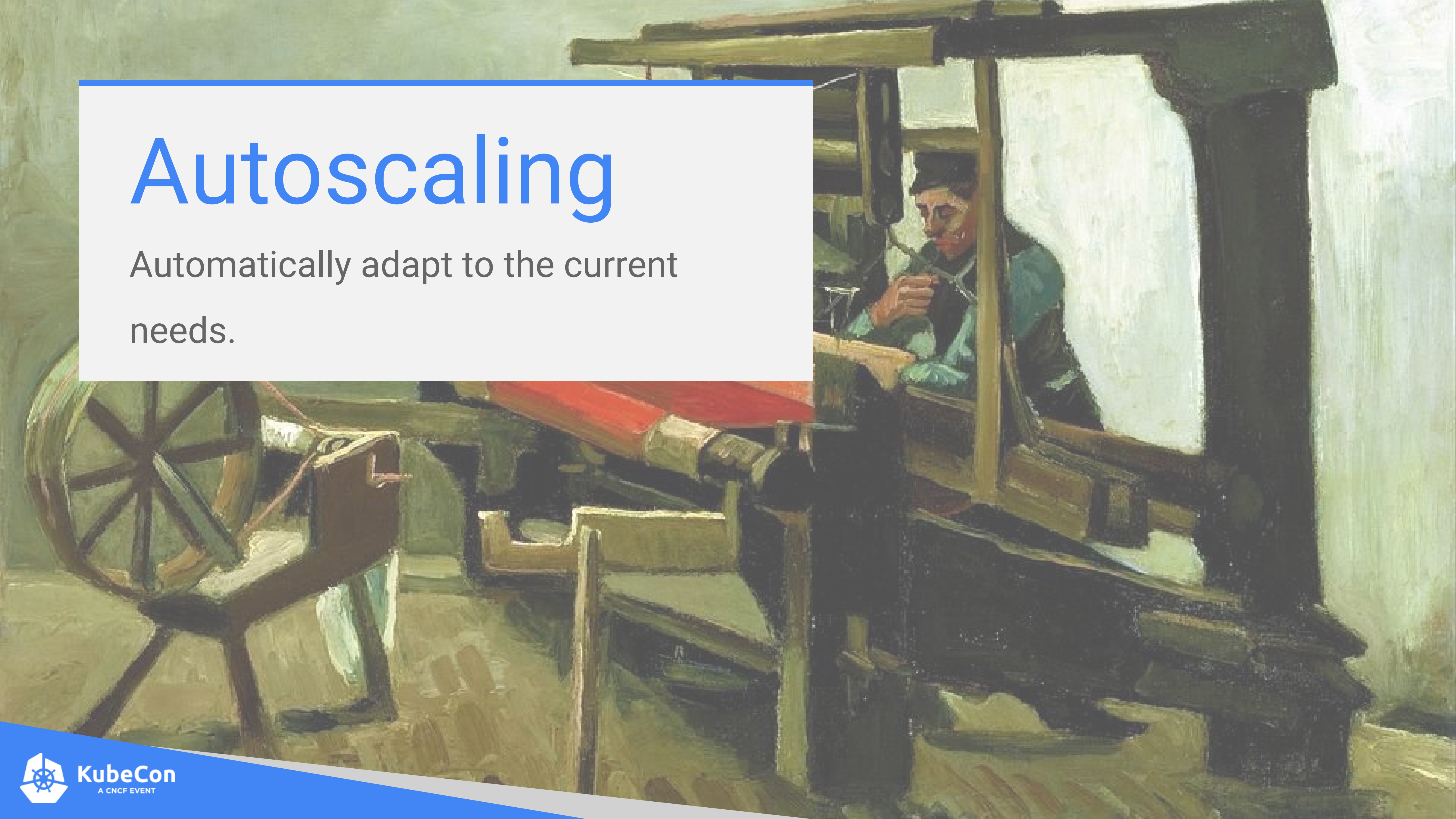
Are we so rich?

Why to overprovision?

- Lack of the knowledge of the real use.
- Hard to change the deployment.
- Lack of automation.

Autoscaling

Automatically adapt to the current needs.



Autoscaling in Kubernetes

Horizontal Pod Autoscaler

Controls the number of replicas in deployments.

Cluster Autoscaler

Controls the number of nodes in the cluster.

Vertical Pod Autoscaler

Controls the amount of requested CPU and Memory for a Pod.

Replica Count

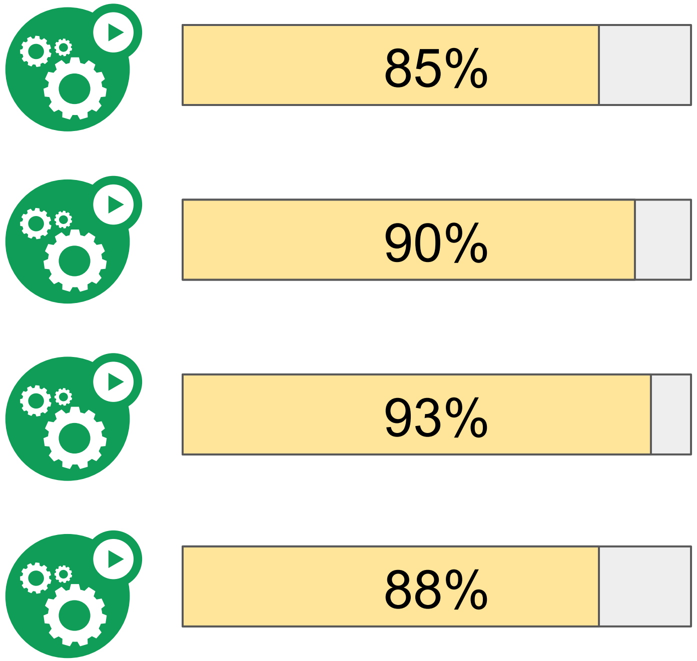
And Horizontal Pod Autoscaler

Autoscaling replica count

- Maintain a decent load.
- Ensure needed redundancy.
- Operate within your quota.

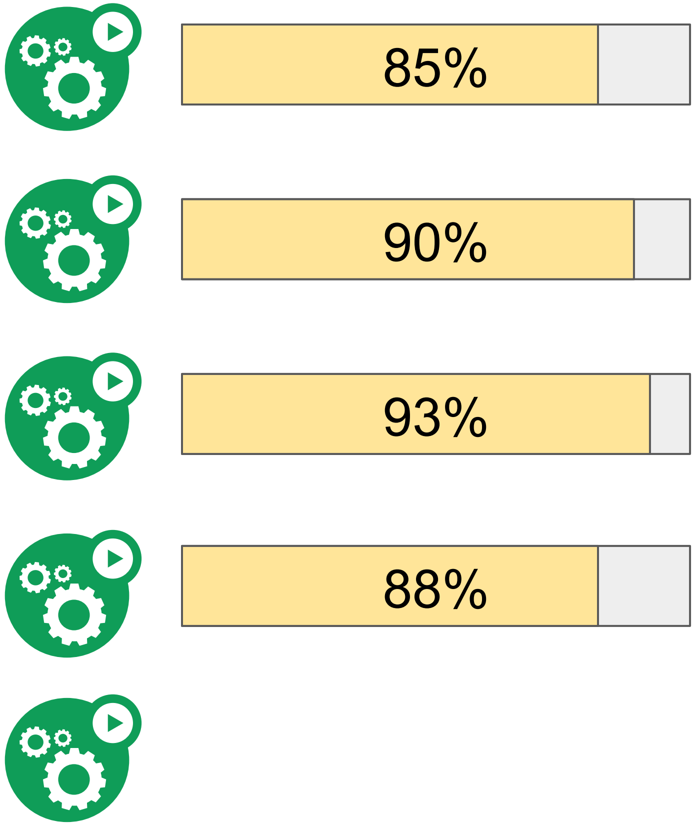
Maintaining the decent load

- If pods are heavily loaded then starting new pods may bring average load down.



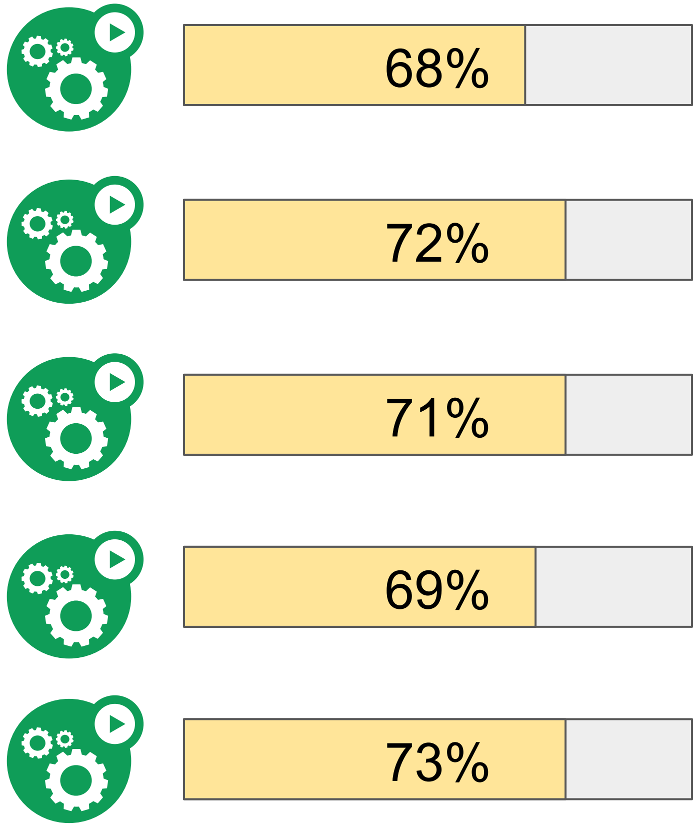
Maintaining the decent load

- If pods are heavily loaded then starting new pods may bring average load down.



Maintaining the decent load

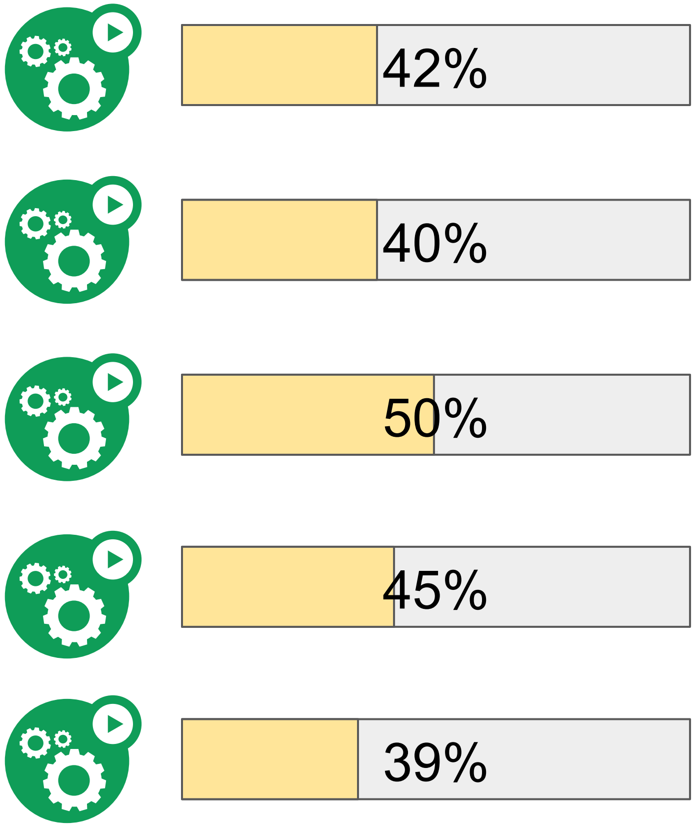
- If pods are heavily loaded then starting new pods may bring average load down.



Maintaining the decent load

- If pods are heavily loaded then starting new pods may bring average load down.

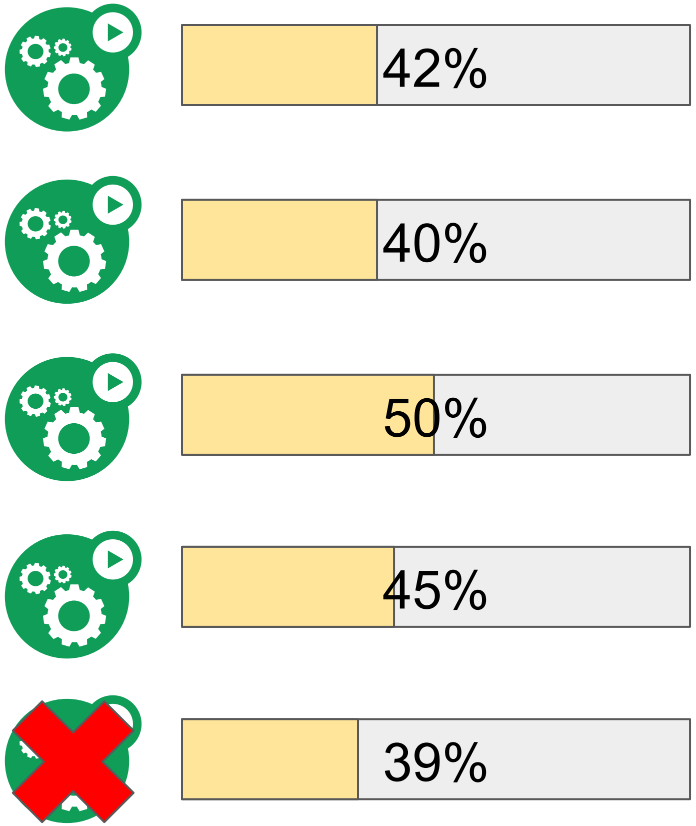
- If pods are barely loaded then stopping pods will free some resources and the deployment should still be ok..



Maintaining the decent load

- If pods are heavily loaded then starting new pods may bring average load down.

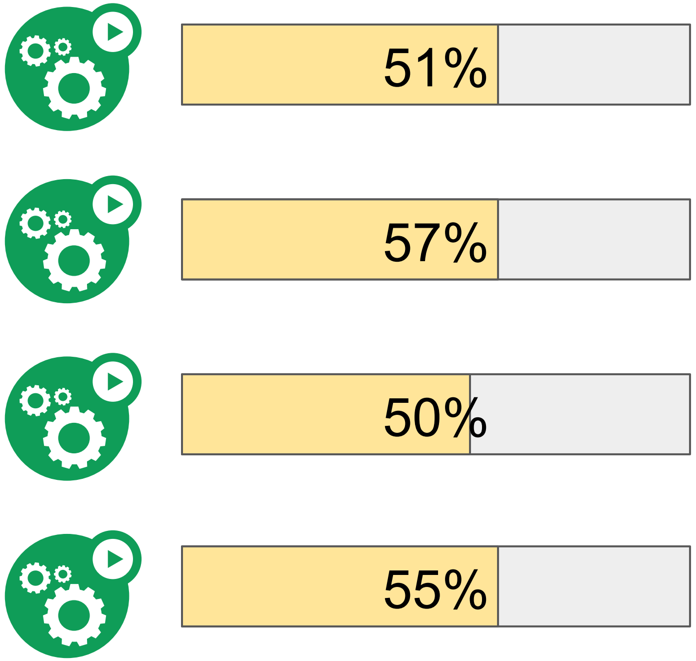
- If pods are barely loaded then stopping pods will free some resources and the deployment should still be ok.



Maintaining the decent load

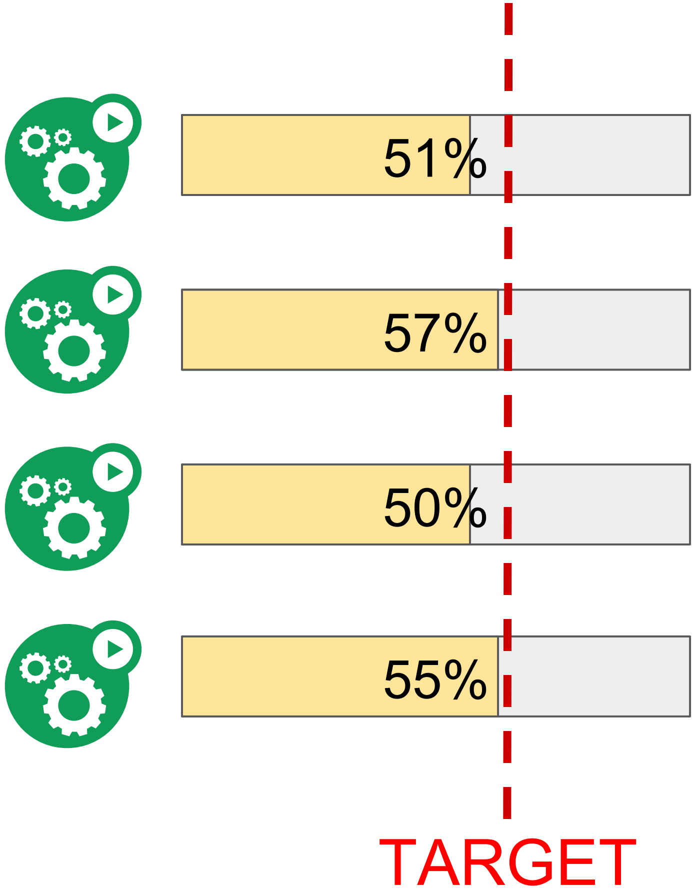
- If pods are heavily loaded then starting new pods may bring average load down.

- If pods are barely loaded then stopping pods will free some resources and the deployment should still be ok.



Maintaining the decent load

- If pods are heavily loaded then starting new pods may bring average load down.
- If pods are barely loaded then stopping pods will free some resources and the deployment should still be ok.
- Specify the target for the load and try to be as close as possible to it.



Replica Count

$$\left\lceil \frac{\sum_{i \in \mathrm{Pods}} \mathtt{Usage}_i}{\mathtt{Target}} \right\rceil$$

Replica Count

- Pod 1 = 70%
- Pod 2 = 80%
- Target = 50%

Replica Count

- Pod 1 = 70%
- Pod 2 = 80%
- Target = 50%
- Sum = 150%
- Replica Count => 3.

What is usage?

$$\frac{\texttt{CurrentCpuConsumption}}{\texttt{PodCpuRequest}}$$

Other Details

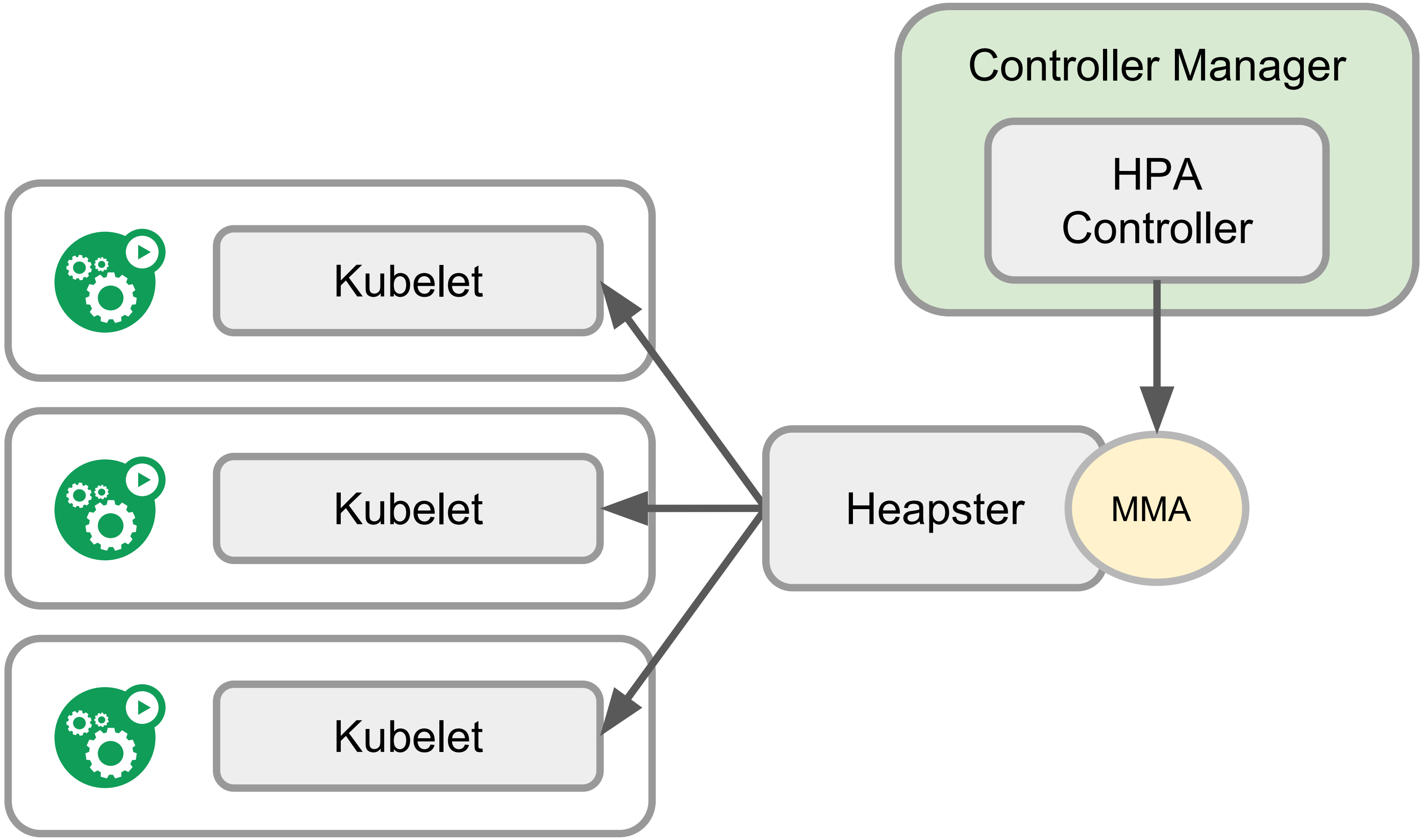
- Margins
- Ready/unready pods
- Missing or broken metrics
- Spikes

HPA - how to enable

```
$ kubectl autoscale
   deployment foo-app
   --min=2 --max=10
   --cpu-percent=70

deployment "foo-app" autoscaled
```

HPA Architecture



HPA Best Practices

- **Declare requests for Pods.**

HPA Best Practices

- Declare requests for Pods
- **Set target well below 100%.**

HPA Best Practices

- Declare requests for Pods
- Set target well below 100%.
- **Target 70% gives you:**

HPA Best Practices

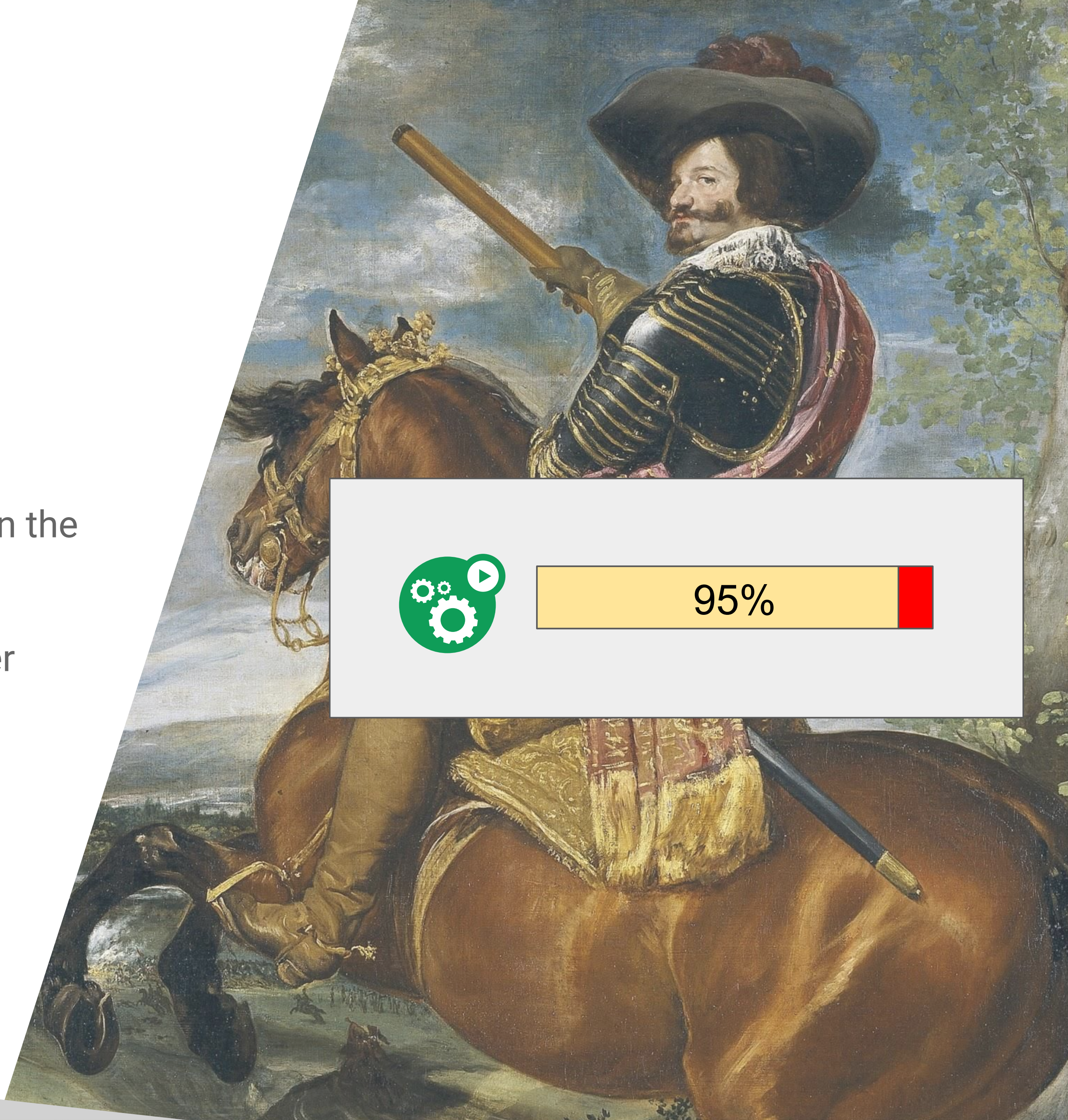
- Declare requests for Pods
- Set target well below 100%.
- Target 70% gives you:
  - **Large window for traffic increase within the currently running pods**

HPA Best Practices

- Declare requests for Pods
- Set target well below 100%.
- Target 70% gives you:
  - Large window for traffic increase within the currently running pods
  - **Ability to have >30% more replicas after the first HPA iteration**

HPA Best Practices

- Declare requests for Pods
- Set target well below 100%.
- Target 70% gives you:
    - Large window for traffic increase within the currently running pods
    - Ability to have >30% more replicas after the first HPA iteration

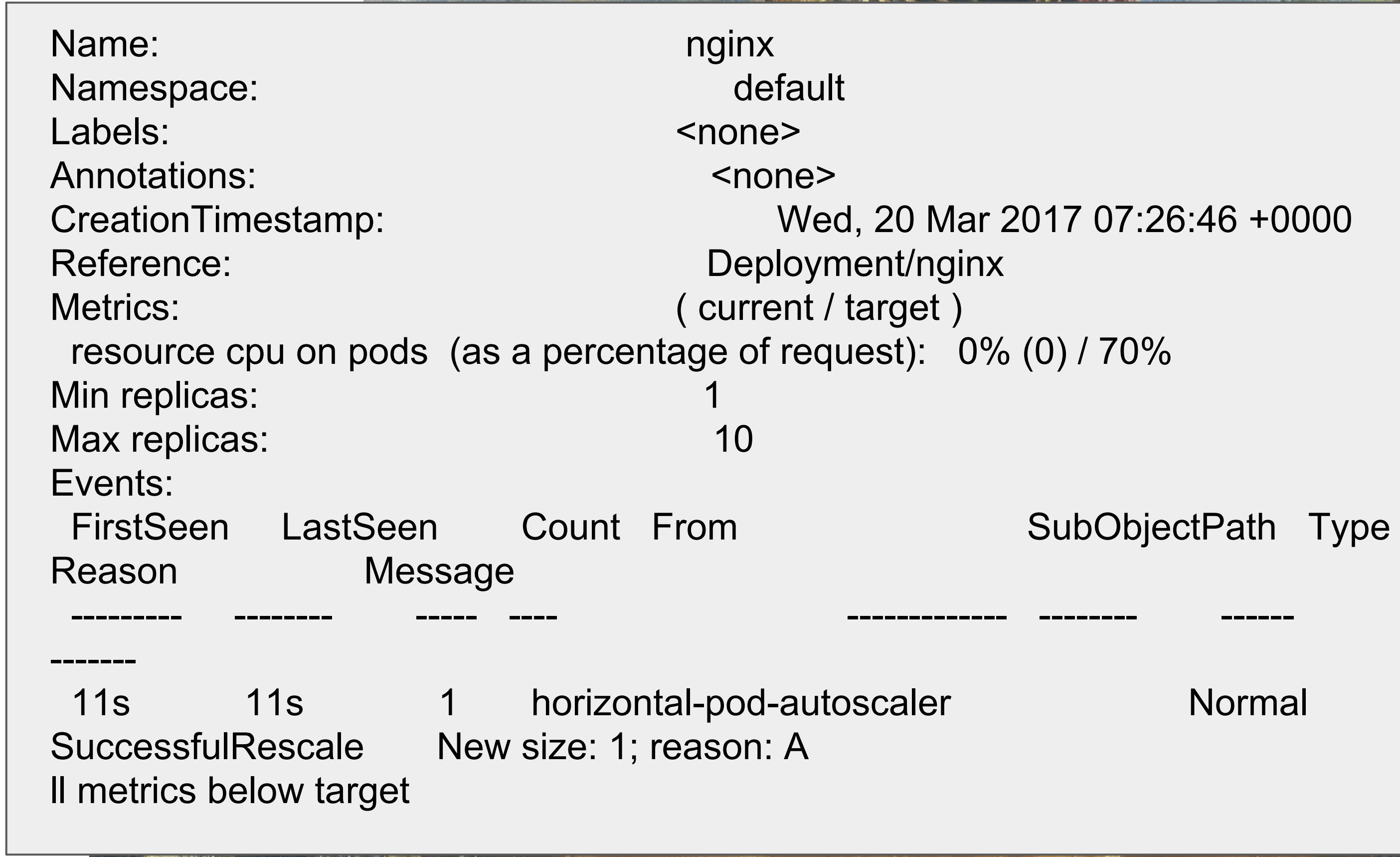


HPA Best Practices

- **Keep you pods and nodes healthy.**

HPA Best Practices

- Keep you pods and nodes healthy.
- **kubectl top**
- **kubectl describe hpa**



```
Name:                                   nginx
Namespace:                                default
Labels:                                 <none>
Annotations:                              <none>
CreationTimestamp:                          Wed, 20 Mar 2017 07:26:46 +0000
Reference:                                Deployment/nginx
Metrics:                                ( current / target )
  resource cpu on pods  (as a percentage of request):   0% (0) / 70%
Min replicas:                             1
Max replicas:                              10
Events:
  FirstSeen     LastSeen        Count   From                    SubObjectPath   Type
Reason                  Message
  ---------     --------        -----   ----                    -------------   --------        ------
-------
  11s           11s             1       horizontal-pod-autoscaler                       Normal
SuccessfulRescale       New size: 1; reason: A
ll metrics below target
```

HPA Best Practices

- Keep you pods and nodes healthy.
- kubectl top
- kubectl describe hpa
- **Custom metrics (like Queries Per Second)**

HPA Best Practices

- Make sure that your requests are short and well load balanced between pods

Node Count

and Cluster Autoscaler

KubeCon
A CNCF EVENT

Philosophy of Node Count

Philosophy of Node Count

- **All pods should have a place to live.**

Philosophy of Node Count

- All pods should have a place to live.
- **Pods are created and deleted.**

Philosophy of Node Count

- All pods should have a place to live.
- Pods are created and deleted.
- **There is Horizontal Pod Autoscaler.**

Philosophy of Node Count

- All pods should have a place to live.
- Pods are created and deleted.
- There is Horizontal Pod Autoscaler.
- **Node count good for today may be bad tomorrow.**

Philosophy of Node Count

- All pods should have a place to live.
- Pods are created and deleted.
- There is Horizontal Pod Autoscaler.
- Node count good for today may be bad tomorrow.
- **Nodes are expensive. Spendthrift is bad.**

Philosophy of Node Count

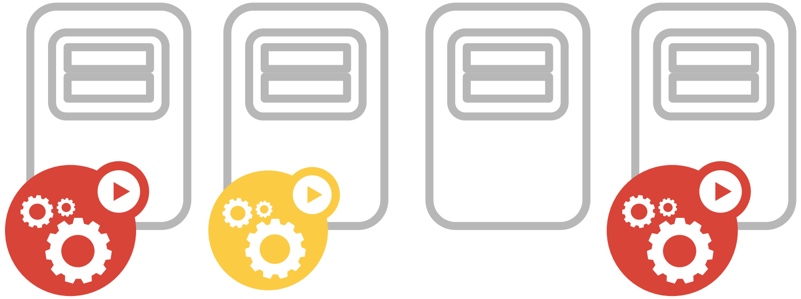
- All pods should have a place to live.
- Pods are created and deleted.
- There is Horizontal Pod Autoscaler.
- Node count good for today may be bad tomorrow.
- Nodes are expensive. Spendthrift is bad.
- **Pods are important. Stinginess is bad.**

Automation

is needed!

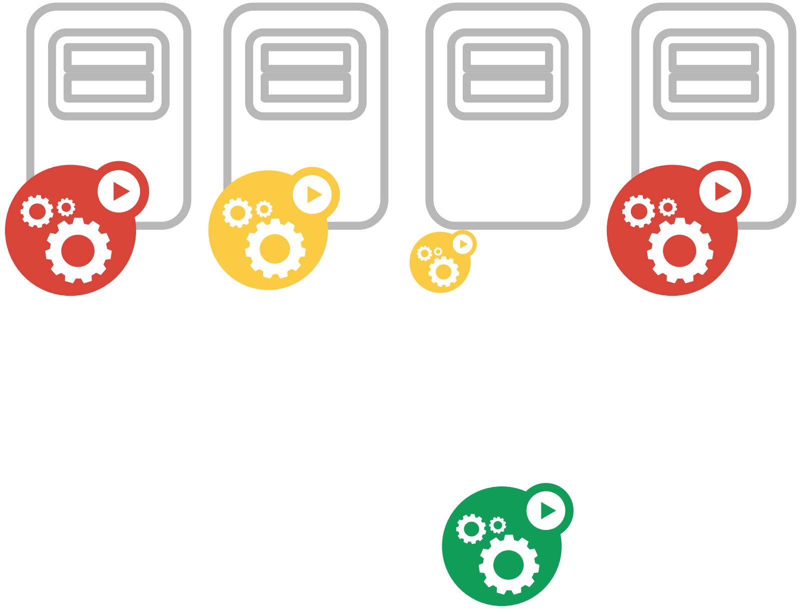
Basic Idea of Automation

- Pods are scheduled based on their declared resource requests.



Basic Idea of Automation

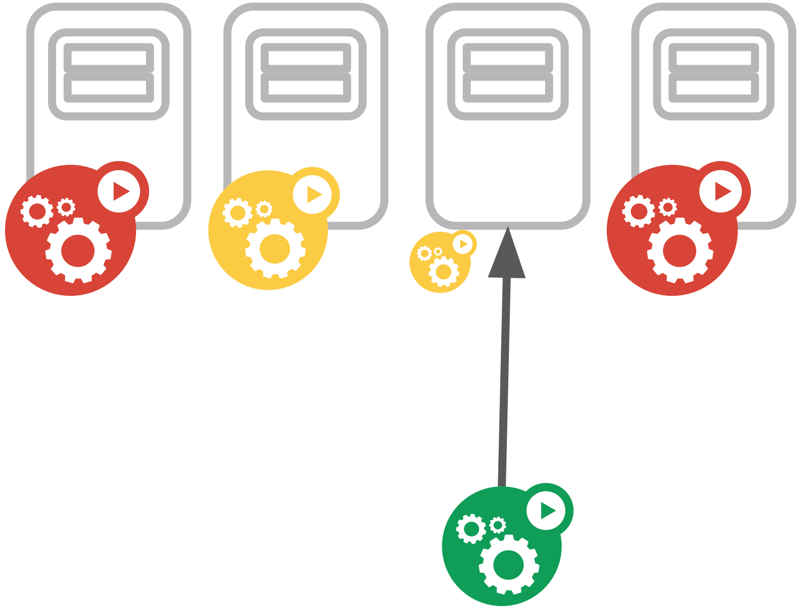
- Pods are scheduled based on their declared resource requests.
- If there is enough resources the pod is scheduled.



Basic Idea of Automation

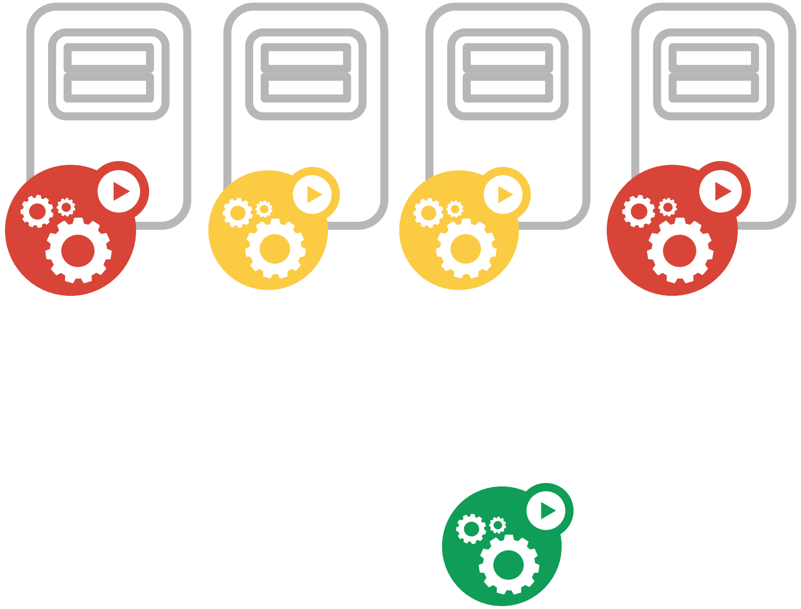
- Pods are scheduled based on their declared resource requests.

- If there is enough resources the pod is scheduled.



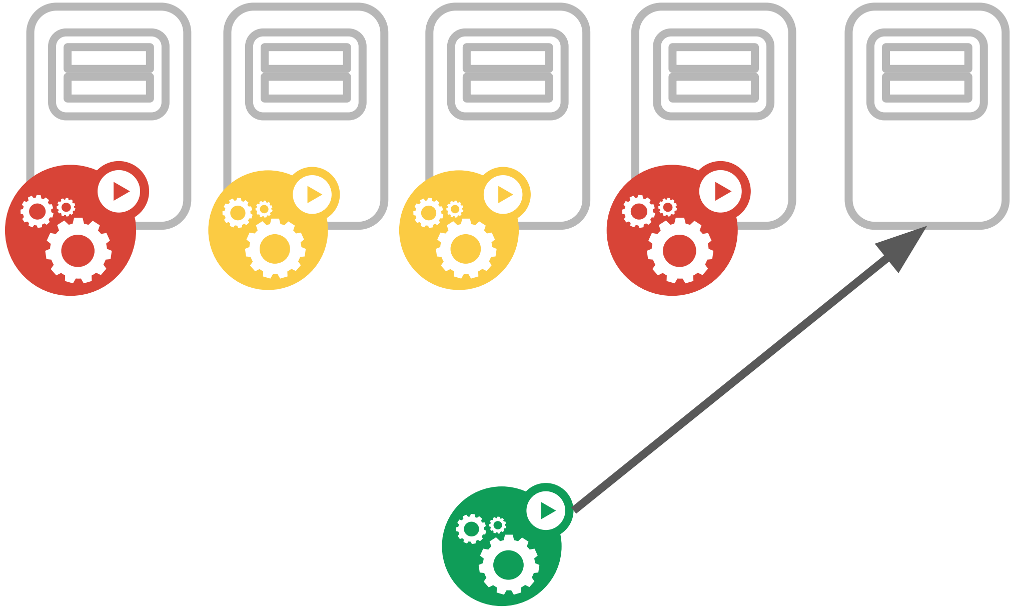
Basic Idea of Automation

- Pods are scheduled based on their declared resource requests.
- If there is enough resources the pod is scheduled.
- If there is no enough resources then a new node has to be added.



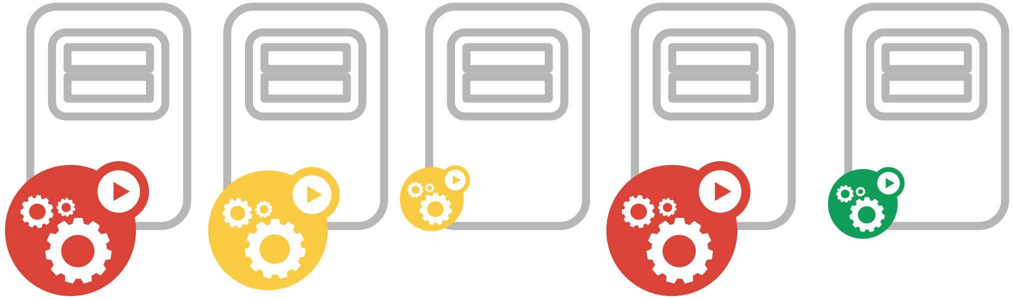
Basic Idea of Automation

- Pods are scheduled based on their declared resource requests.
- If there is enough resources the pod is scheduled.
- If there is no enough resources then a new node has to be added.



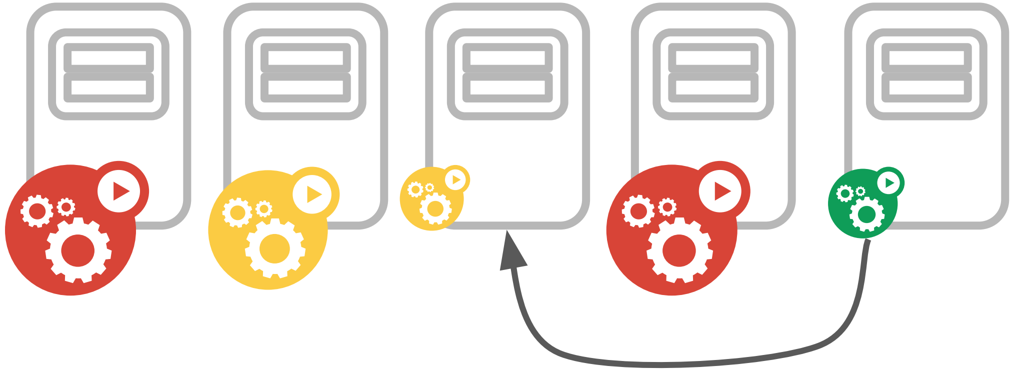
Basic Idea of Automation

- Pods are scheduled based on their declared resource requests.
- If there is enough resources the pod is scheduled.
- If there is no enough resources then a new node has to be added.
- If there are too many resources in the cluster then some nodes should be removed.



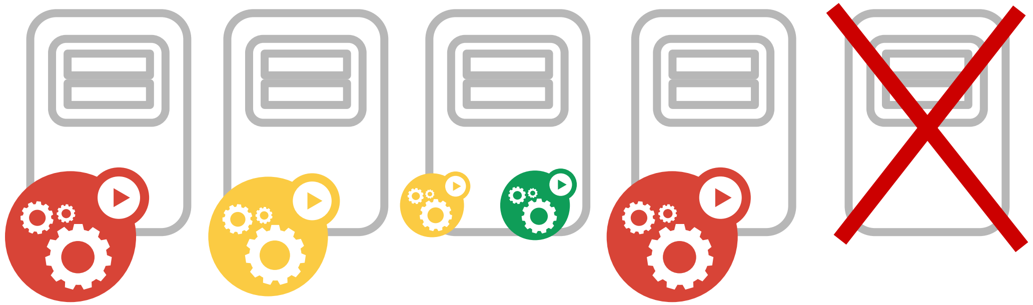
Basic Idea of Automation

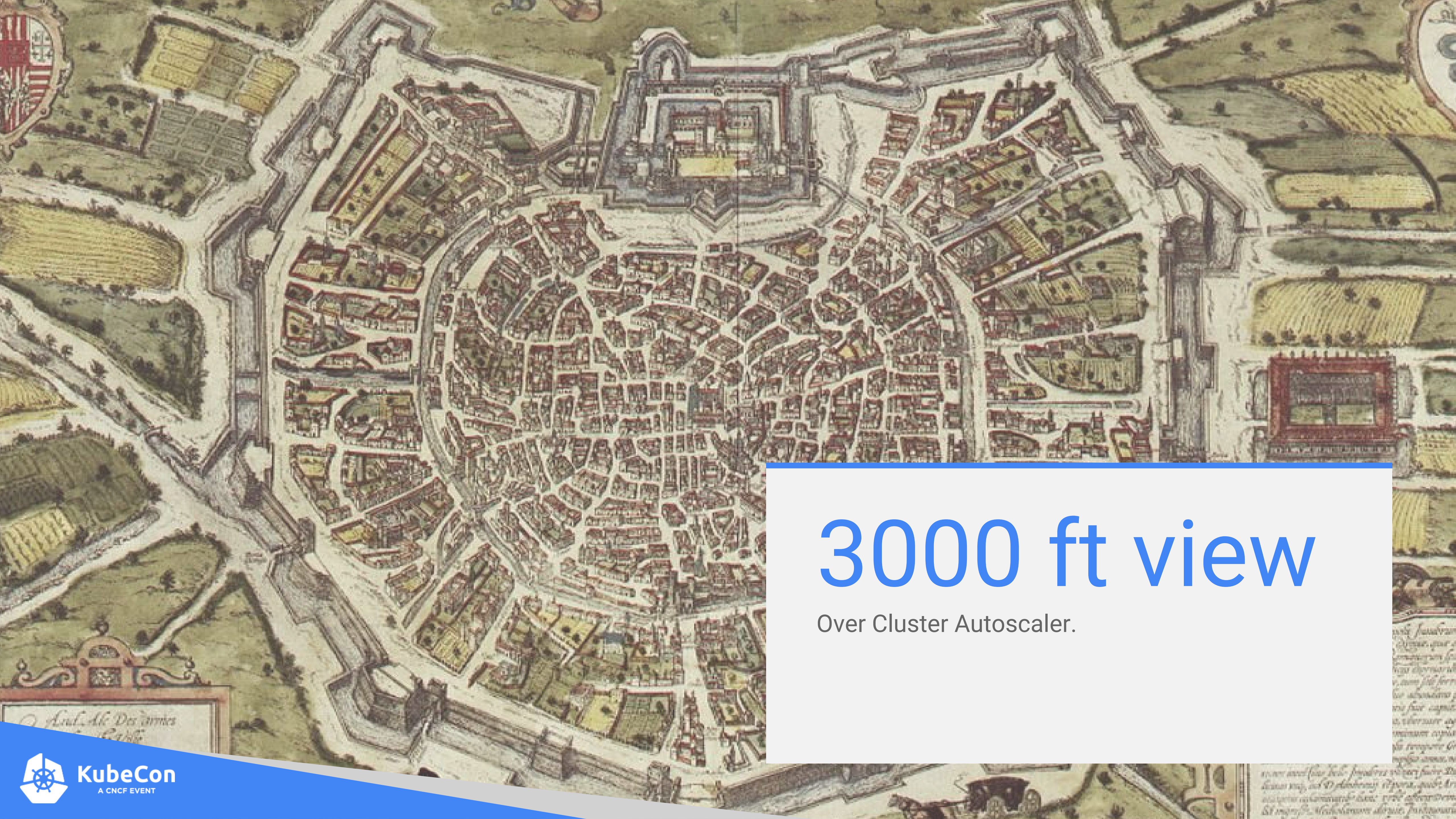
- Pods are scheduled based on their declared resource requests.
- If there is enough resources the pod is scheduled.
- If there is no enough resources then a new node has to be added.
- If there are too many resources in the cluster then some nodes should be removed.



Basic Idea of Automation

- Pods are scheduled based on their declared resource requests.
- If there is enough resources the pod is scheduled.
- If there is no enough resources then a new node has to be added.
- If there are too many resources in the cluster then some nodes should be removed.





3000 ft view

Over Cluster Autoscaler.

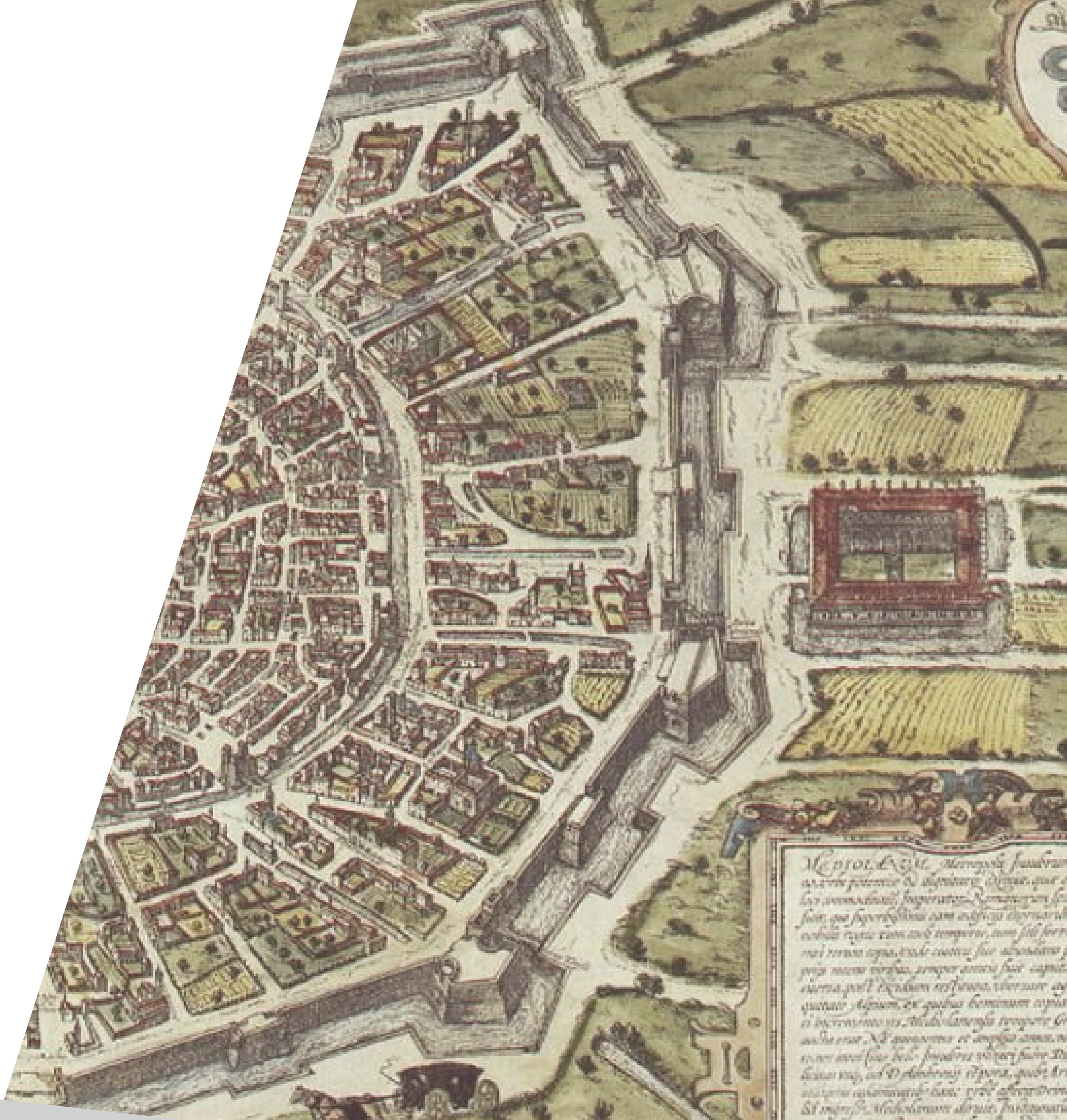
Cluster Autoscaler

- Runs on the master node in a separate pod.
- Maintains API server watches on all nodes and pods in the cluster.
- Doesn’t use any node or pod-level metrics.

Nodes in Cluster Autoscaler

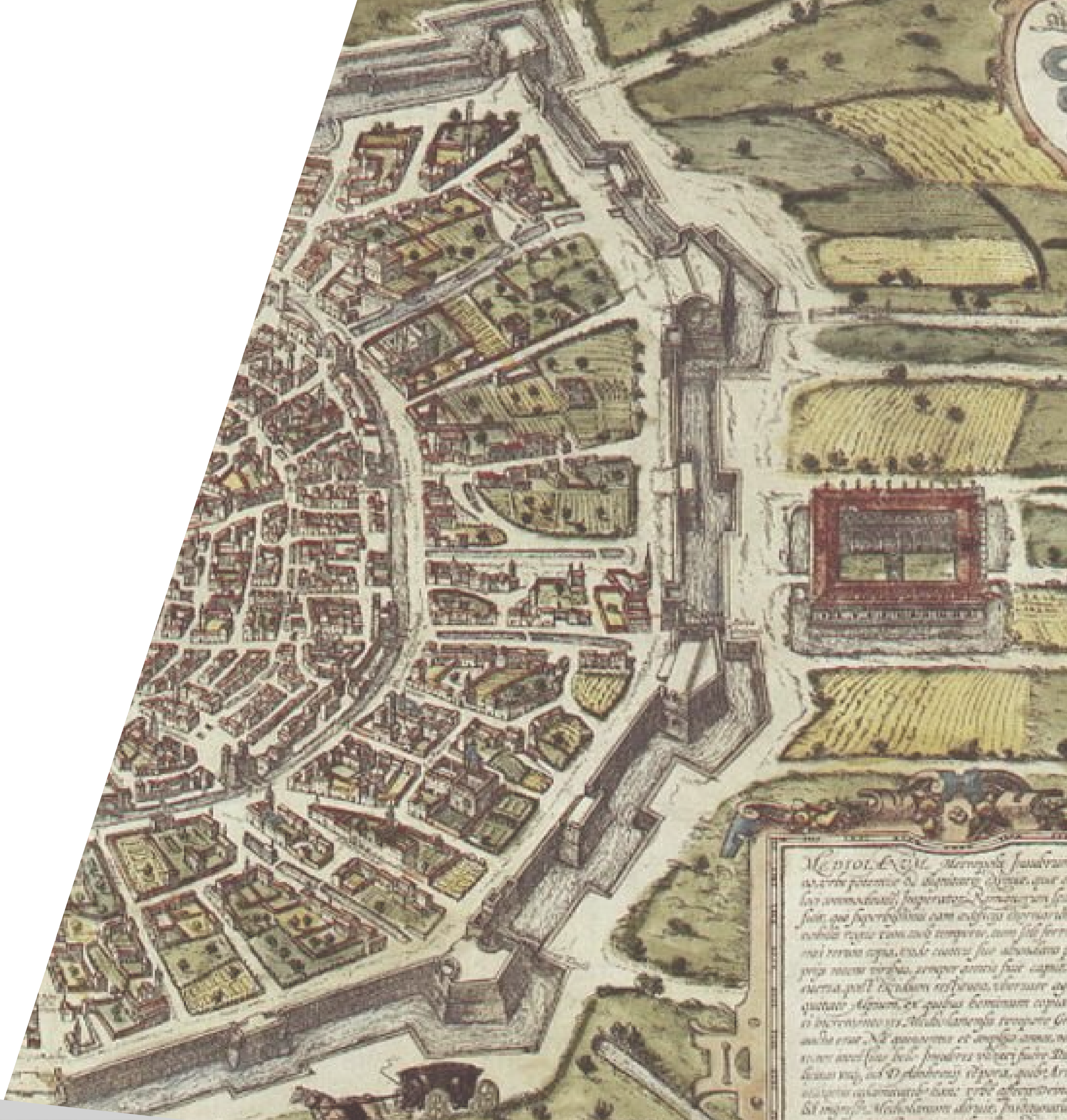
- Node groups:
  - MIGs (GCE/GKE)
  - Autoscaling Groups (AWS)
  - ScaleSets (Azure)

Main Loop Checks



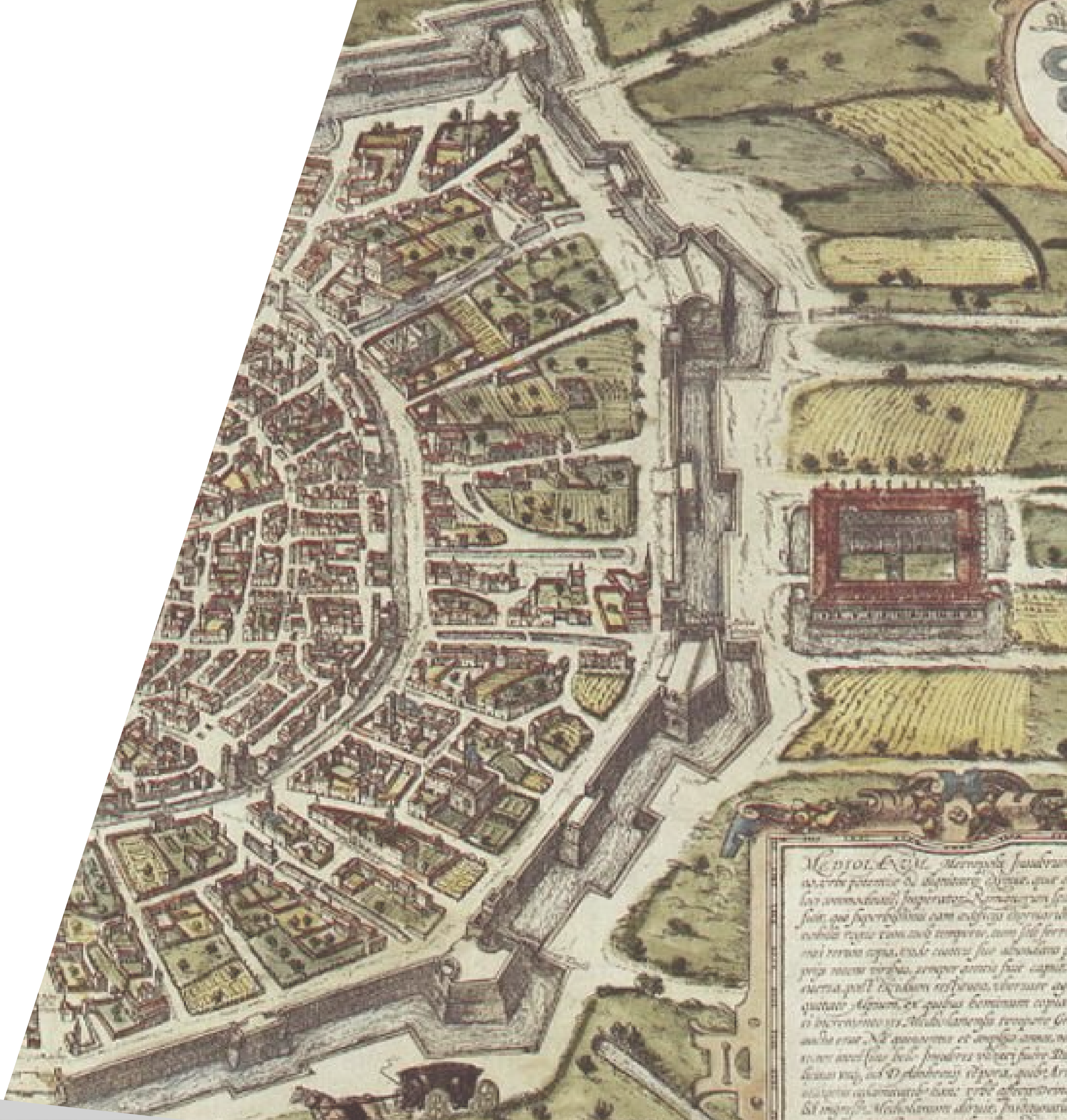
Main Loop Checks

- **If the cluster is in a good shape.**



Main Loop Checks

- If the cluster is in a good shape.
- **If there are unschedulable pods.**



Main Loop Checks

- If the cluster is in a good shape.
- If there are unschedulable pods.
- **Which of the node groups can be expanded to accommodate these pods and expands one of them.**

Main Loop Checks

- If the cluster is in a good shape.
- If there are unschedulable pods.
- Which of the node groups can be expanded to accommodate these pods and expands one of them.
- **How much the nodes are utilized and which can be removed.**

Main Loop Checks

- If the cluster is in a good shape.
- If there are unschedulable pods.
- Which of the node groups can be expanded to accommodate these pods and expands one of them.
- How much the nodes are utilized and which can be removed.
- **Which nodes could be removed for long enough and removes one of them.**

Unneeded nodes

According to current heuristic, a node can be considered unneeded if:

Unneeded nodes

According to current heuristic, a node can be considered unneeded if:

- **Its utilization is below 50%.**

Unneeded nodes

According to current heuristic, a node can be considered unneeded if:

- Its utilization is below 50%.
- **When all of the pods running on the node can be moved elsewhere.**

Unneeded nodes

According to current heuristic, a node can be considered unneeded if:

- Its utilization is below 50%.
- When all of the pods running on the node can be moved elsewhere.
- **There are no kube-system pods**

Unneeded nodes

According to current heuristic, a node can be considered unneeded if:

- Its utilization is below 50%.
- When all of the pods running on the node can be moved elsewhere.
- There are no kube-system pods
- **There are no pods with local storage.**

When to kill a node?

- Node was unneeded for 10 minutes.
- There was no scale up in the last 10 minutes.

Node killing process

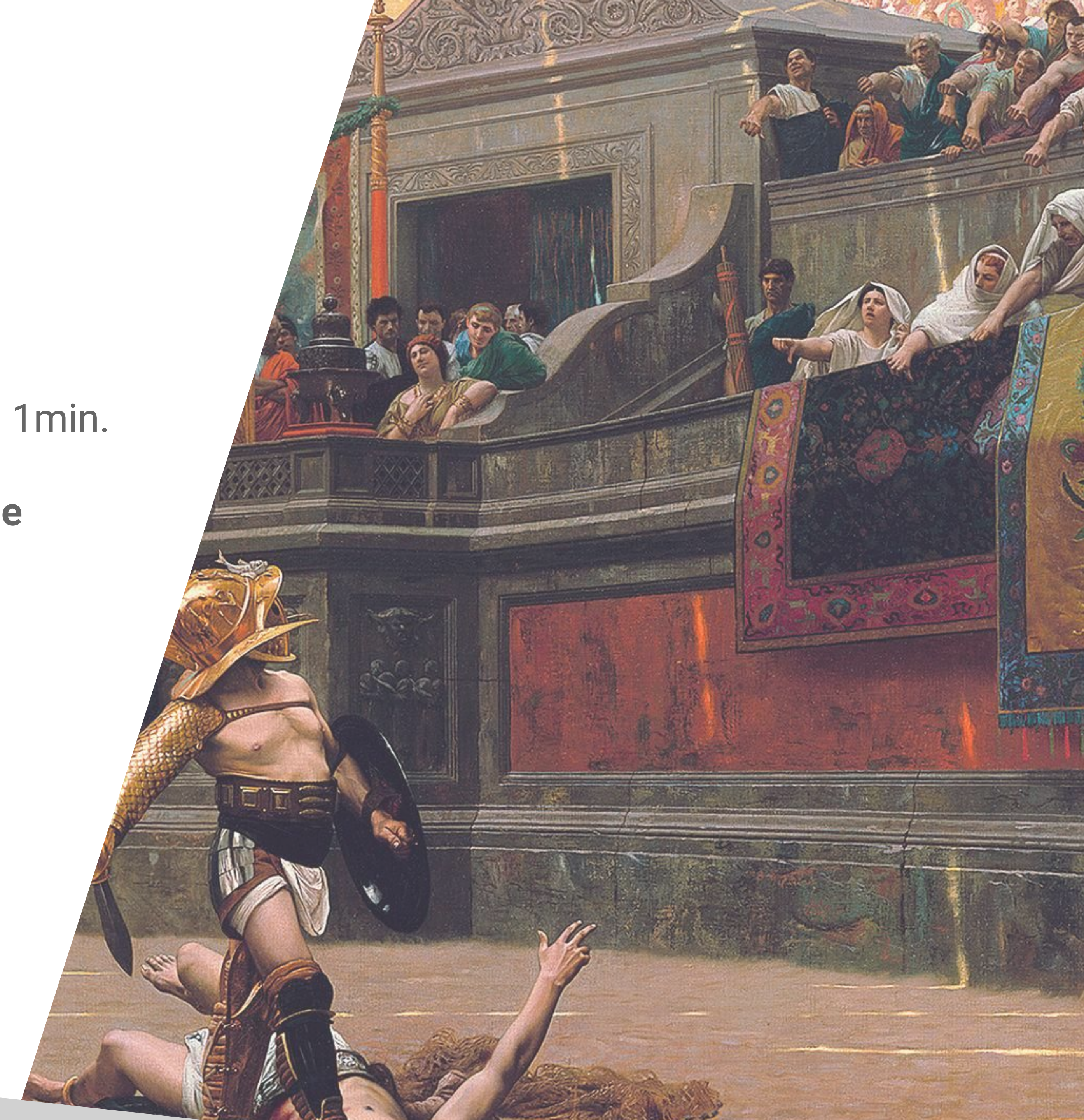
- **Pod Disruption Budget is used.**

Node killing process

- Pod Disruption Budget is used.
- **Graceful termination is honoured up to 1min.**

Node killing process

- Pod Disruption Budget is used.
- Graceful termination is honoured up to 1min.
- **VM running the node is removed by the cloud provider.**



Node killing process

- Pod Disruption Budget is used.
- Graceful termination is honoured up to 1min.
- VM running the node is removed by the cloud provider.
- **Empty nodes are killed in bulk**

Node killing process

- Pod Disruption Budget is used.
- Graceful termination is honoured up to 1min.
- VM running the node is removed by the cloud provider.
- Empty nodes are killed in bulk
- **Non-empty - 1 at a time**

CA Best Practices

- **Do not manually modify single nodes within a node group (e.g. DO NOT add extra labels)**

CA Best Practices

- Do not manually modify single nodes within a node group (e.g. DO NOT add extra labels)
- **Declare requests for Pods.**

CA Best Practices

- Do not manually modify single nodes within a node group (e.g. DO NOT add extra labels)
- Declare requests for Pods.
- **Use Pod Disruption Budgets.**

CA Best Practices

- Do not manually modify single nodes within a node group (e.g. DO NOT add extra labels)
- Declare requests for Pods.
- Use Pod Disruption Budgets.
- **CA works best with homogenous clusters.**

CA Best Practices

- kubectl describe configmap
- kubectl get events

```
$ kubectl describe configmap
    cluster-autoscaler-status
    --namespace=kube-system
[...]
Cluster-autoscaler status at 2017-03-27 14:08:11.175840061 +0000 UT
Cluster-wide:
  Health:      Healthy (ready=3 unready=0 notStarted=0 longNotStart
registered=3)
               LastProbeTime:      2017-03-27 14:08:10.731267279 +0
               LastTransitionTime: 2017-03-27 13:57:17.347440444 +0
  ScaleUp:     InProgress (ready=3 registered=3)
               LastProbeTime:      2017-03-27 14:08:10.731267279 +0
               LastTransitionTime: 2017-03-27 14:07:28.866558907 +0
  ScaleDown:   NoCandidates (candidates=0)
               LastProbeTime:      2017-03-27 14:08:11.175630989 +0
               LastTransitionTime: 2017-03-27 13:57:17.665322299 +0
NodeGroups:
  Name:        https://content.googleapis.com/compute/v1/projects/
  Health:      Healthy (ready=2 unready=0 notStarted=0 longNotStart
cloudProviderTarget=4)
               LastProbeTime:      2017-03-27 14:08:10.731267279 +0
               LastTransitionTime: 2017-03-27 13:57:17.347440444 +0
  ScaleUp:     InProgress (ready=2 cloudProviderTarget=4)
               LastProbeTime:      2017-03-27 14:08:10.731267279 +0
               LastTransitionTime: 2017-03-27 14:07:28.866558907 +0
  ScaleDown:   NoCandidates (candidates=0)
               LastProbeTime:      2017-03-27 14:08:11.175630989 +0
               LastTransitionTime: 2017-03-27 13:57:17.665322299 +0
```

Still BETA?

What is missing?

What is missing to reach GA?

CA-friendly scheduler

The current one tries to spread pods and increases the number of reschedulings.

Easier configuration

Especially for non-GKE users.

More tests

Especially non trivial failure scenarios.

Stable status info

Switch to ComponentStatus.

+ User Feedback

What is missing to reach GA?

CA-friendly scheduler

The current one tries to spread pods and increases the number of reschedulings.

Easier configuration

Especially for non-GKE users.

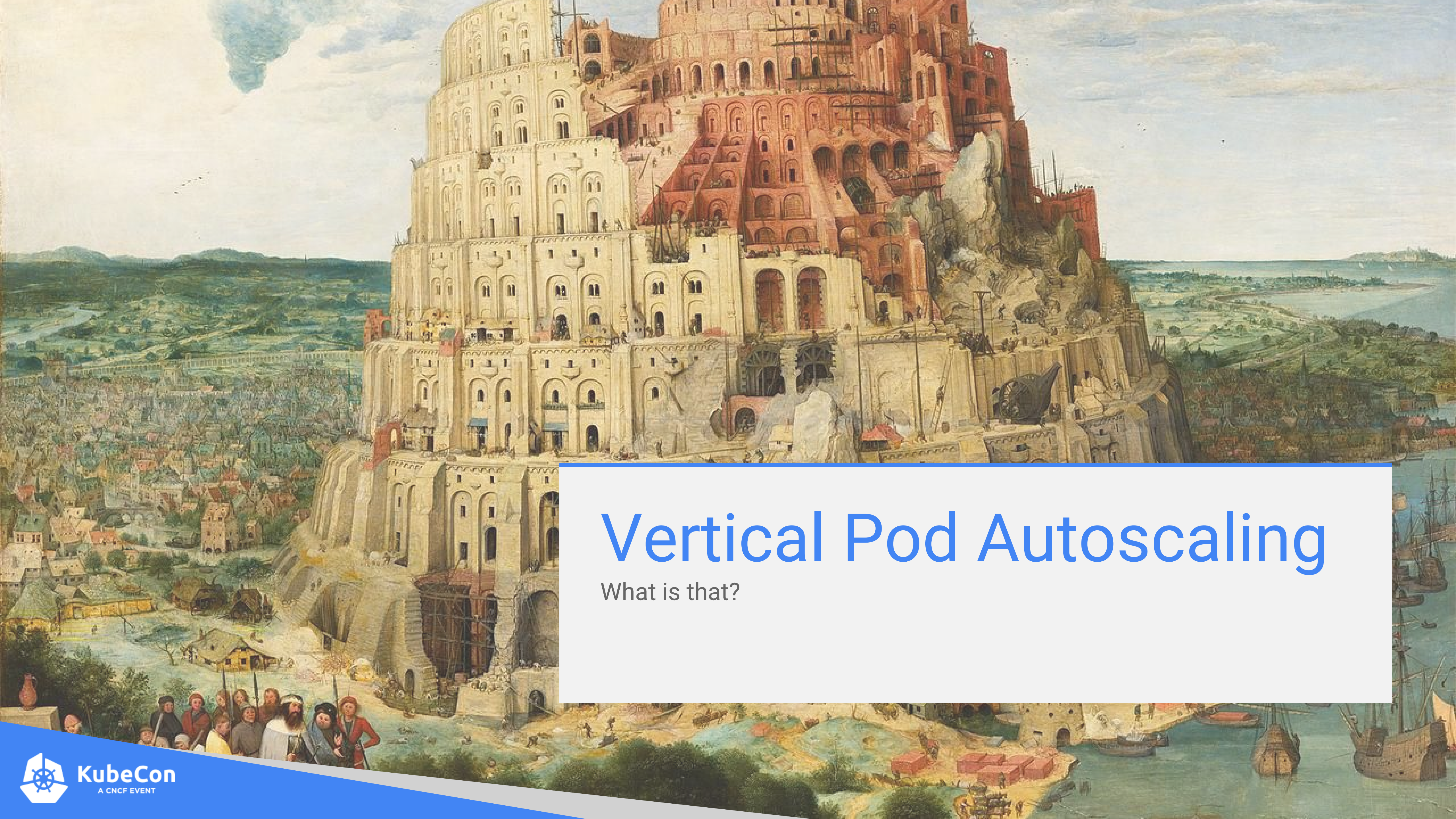
More tests

Especially non trivial failure scenarios.

Stable status info

Switch to ComponentStatus.

+ *User Feedback*



Vertical Pod Autoscaling

What is that?

Vertical Pod autoscaler

- Goal - automatically set container requests.
- Design almost completed.
- Alpha Proof Of Concept expected in June 2017.

SIG-Autoscaling

Every Thursday 17:30 Berlin time

Questions?

There must be some...