We wrote a Kubernetes operator

Here is why you should too

[arthurb@padok.fr ~]$ whoami

Site Reliability Engineer

Certified Kubernetes Administrator

What is a Kubernetes operator?

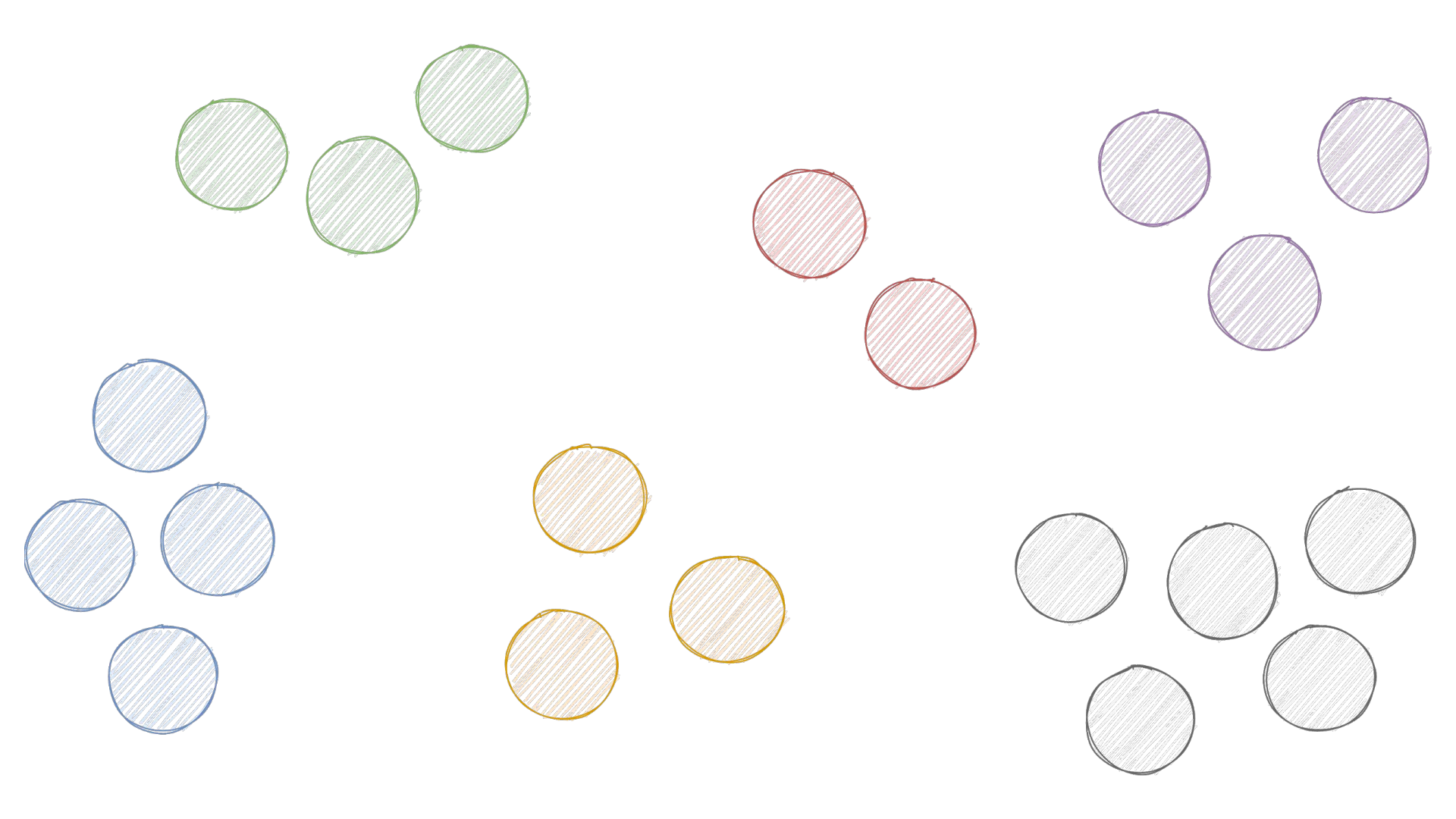
An Operator is an application-specific controller that extends the Kubernetes API to create, configure, and manage instances of complex stateful applications on behalf of a Kubernetes user.

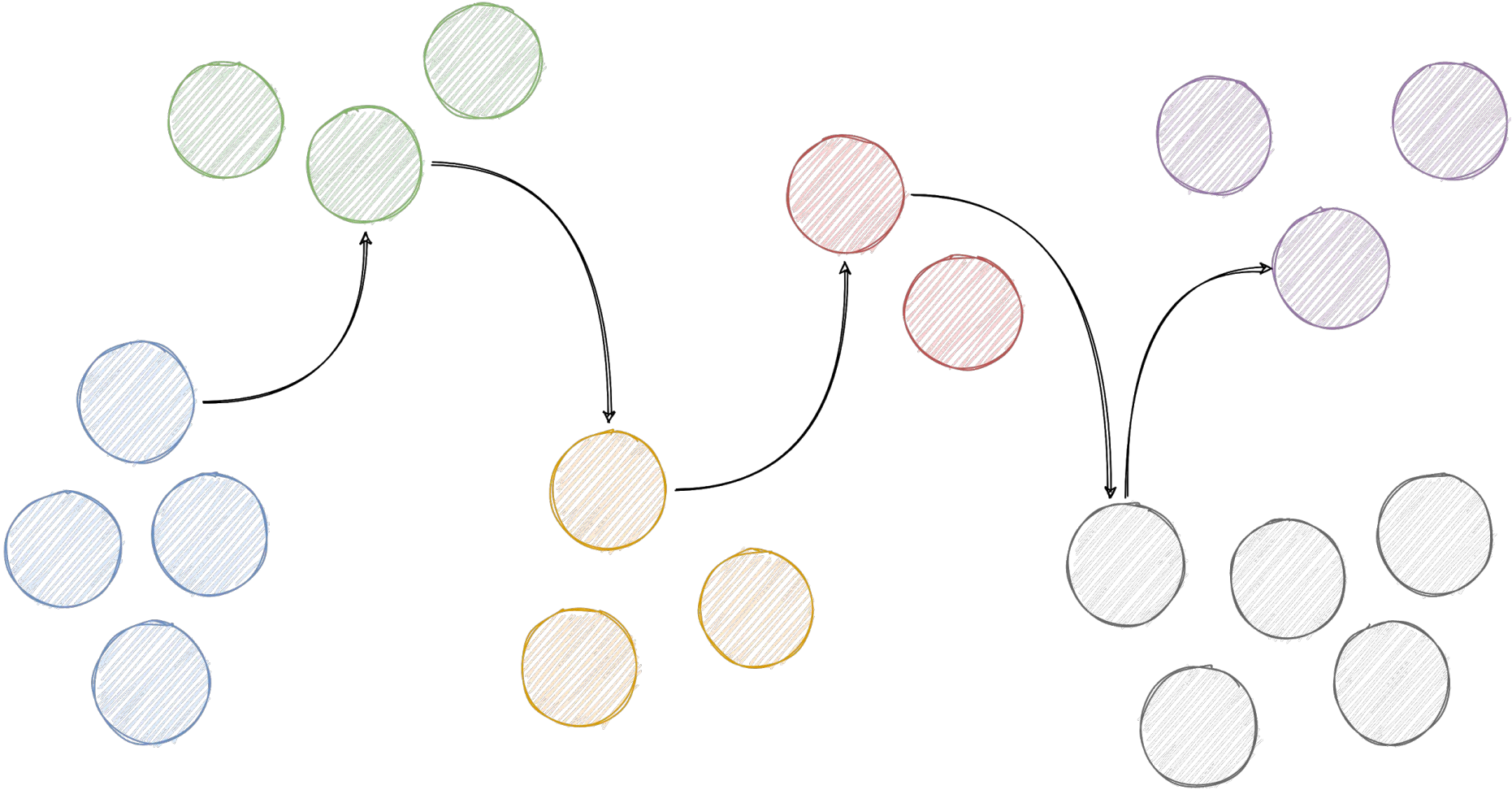
https://coreos.com/blog/introducing-operators.html

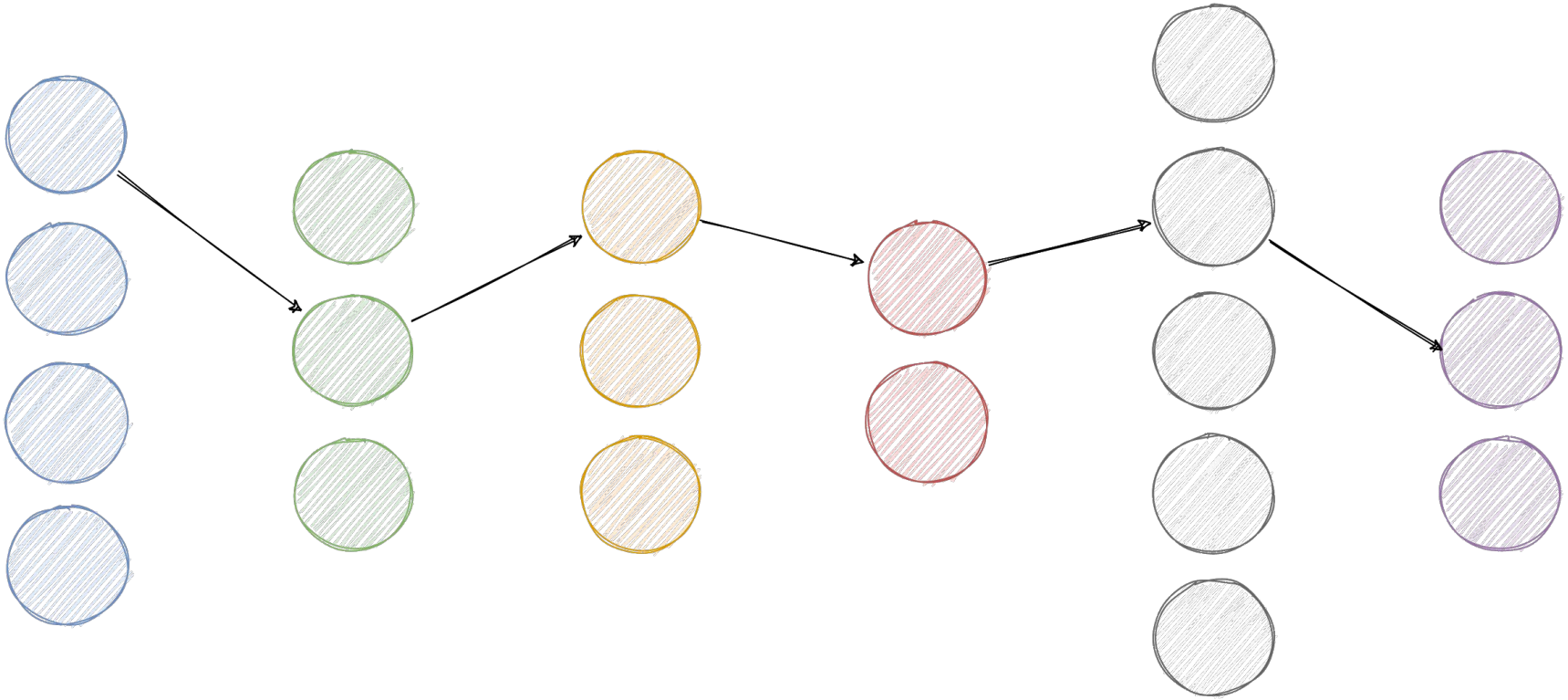
An operator consists of multiple controllers

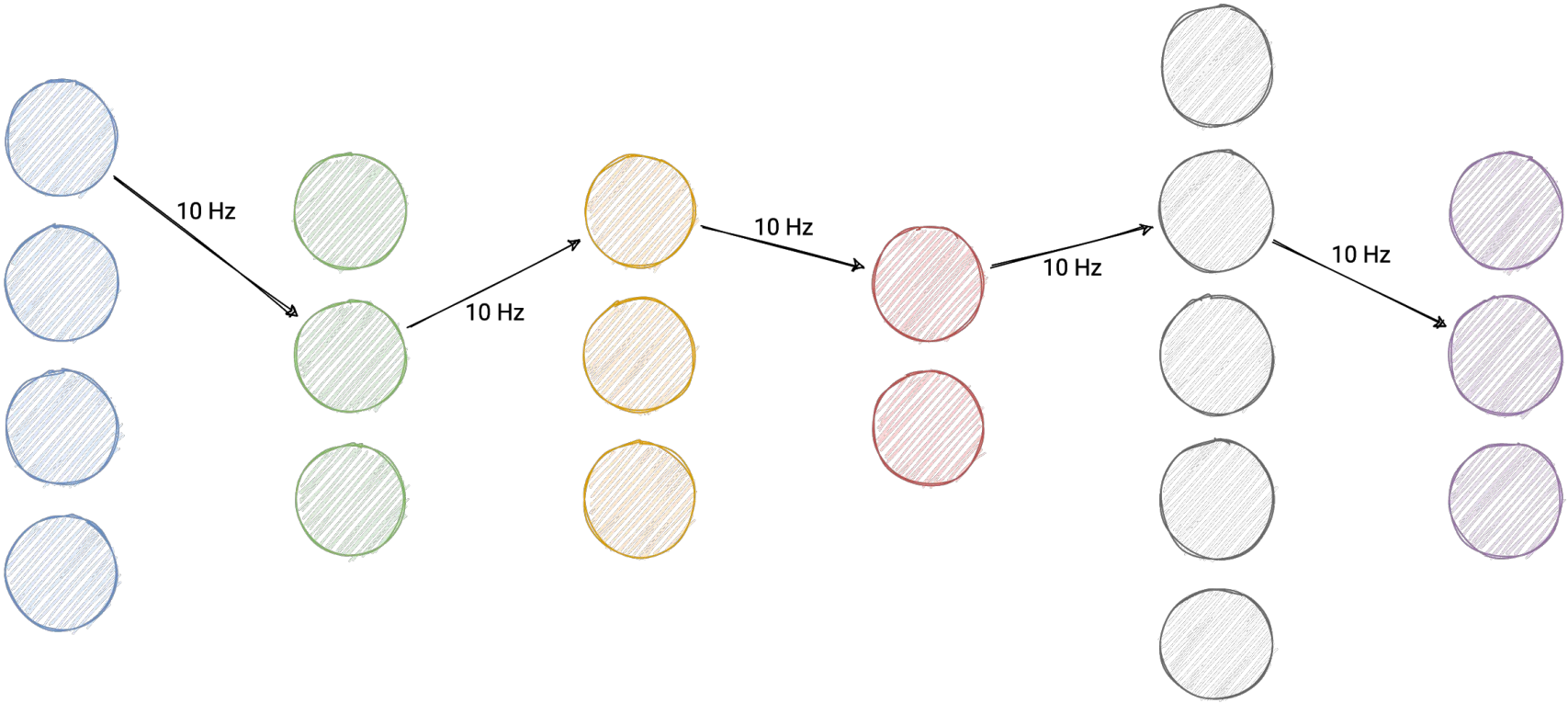
A Kubernetes controller is a finite state machine with a reconciliation loop that converges on desired state step by step

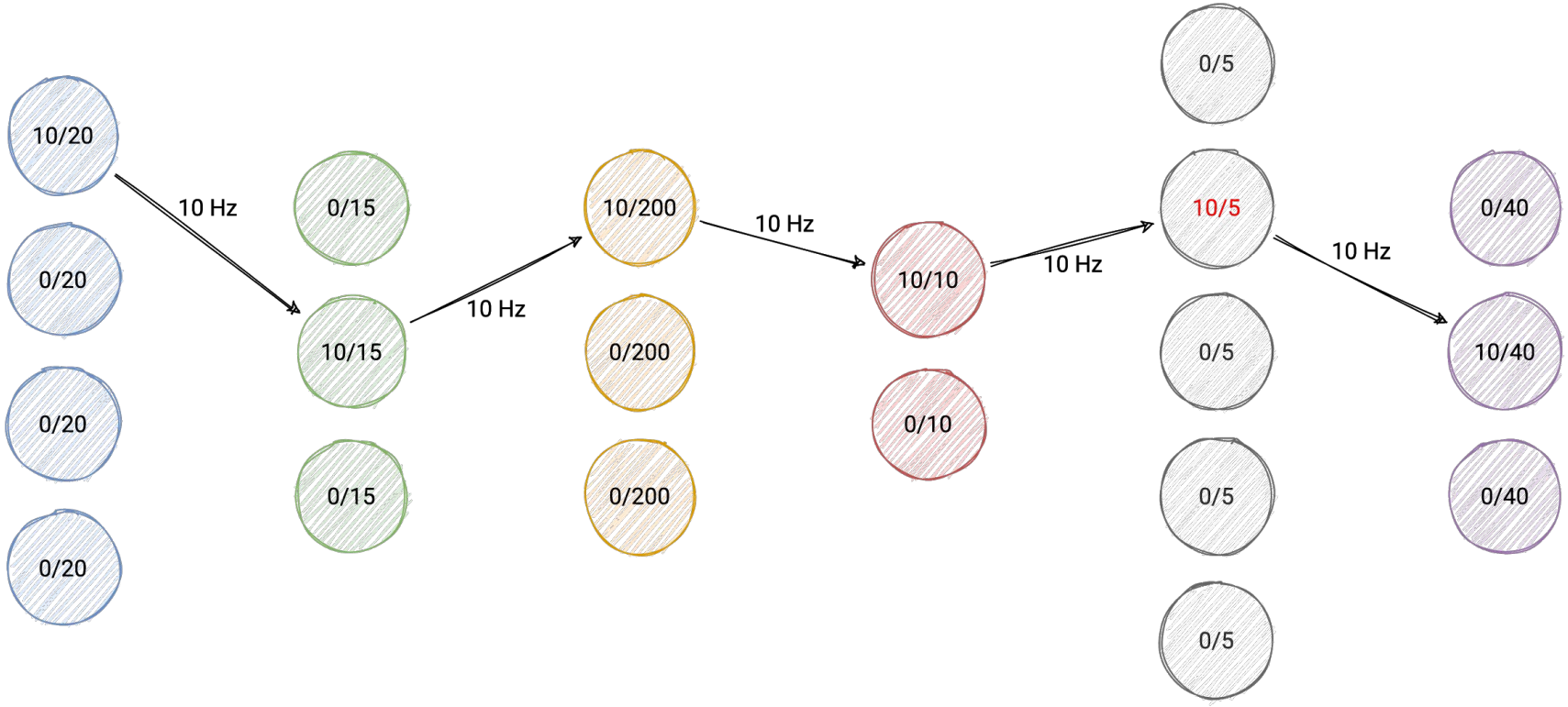
- Why we built it
- What we built it with
- How it works
- Lessons learned
- Overall experience

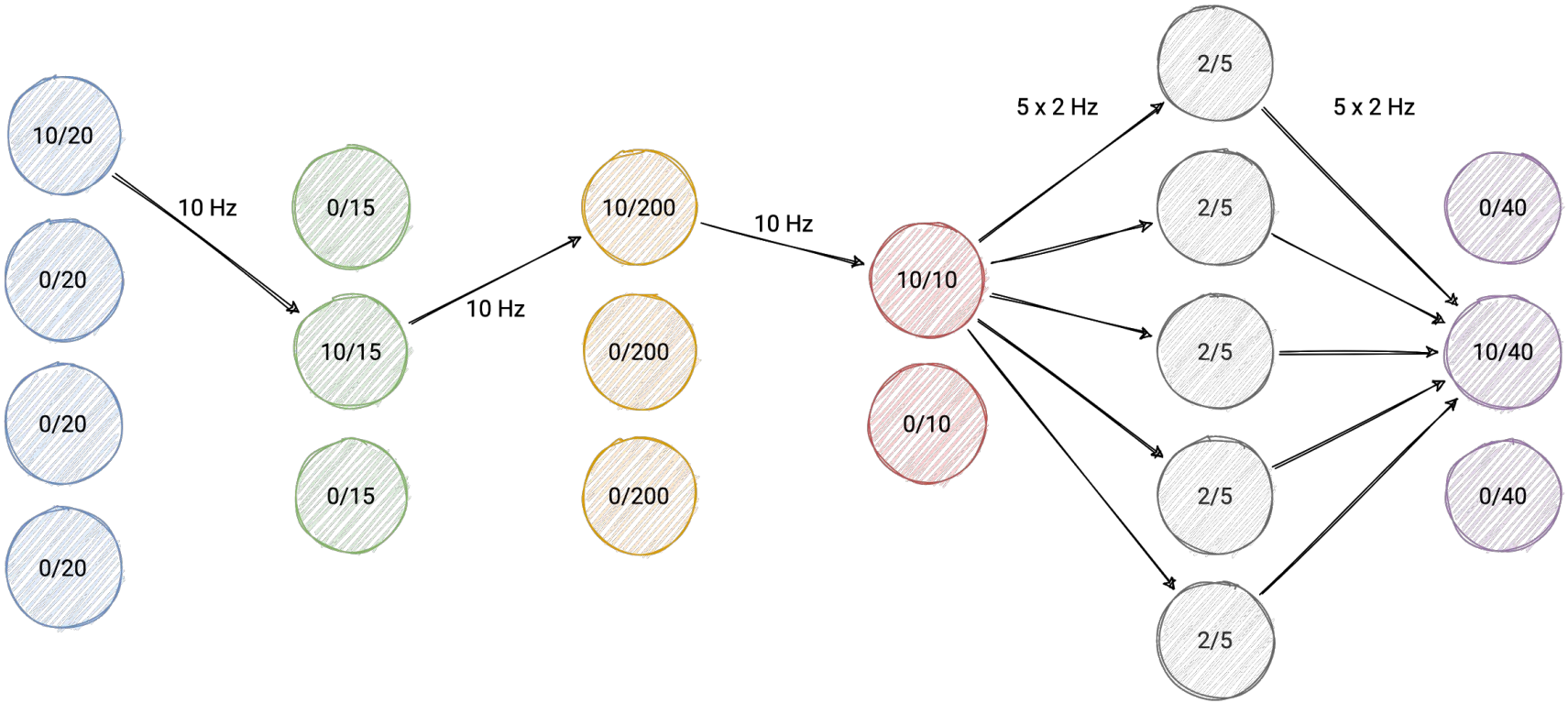


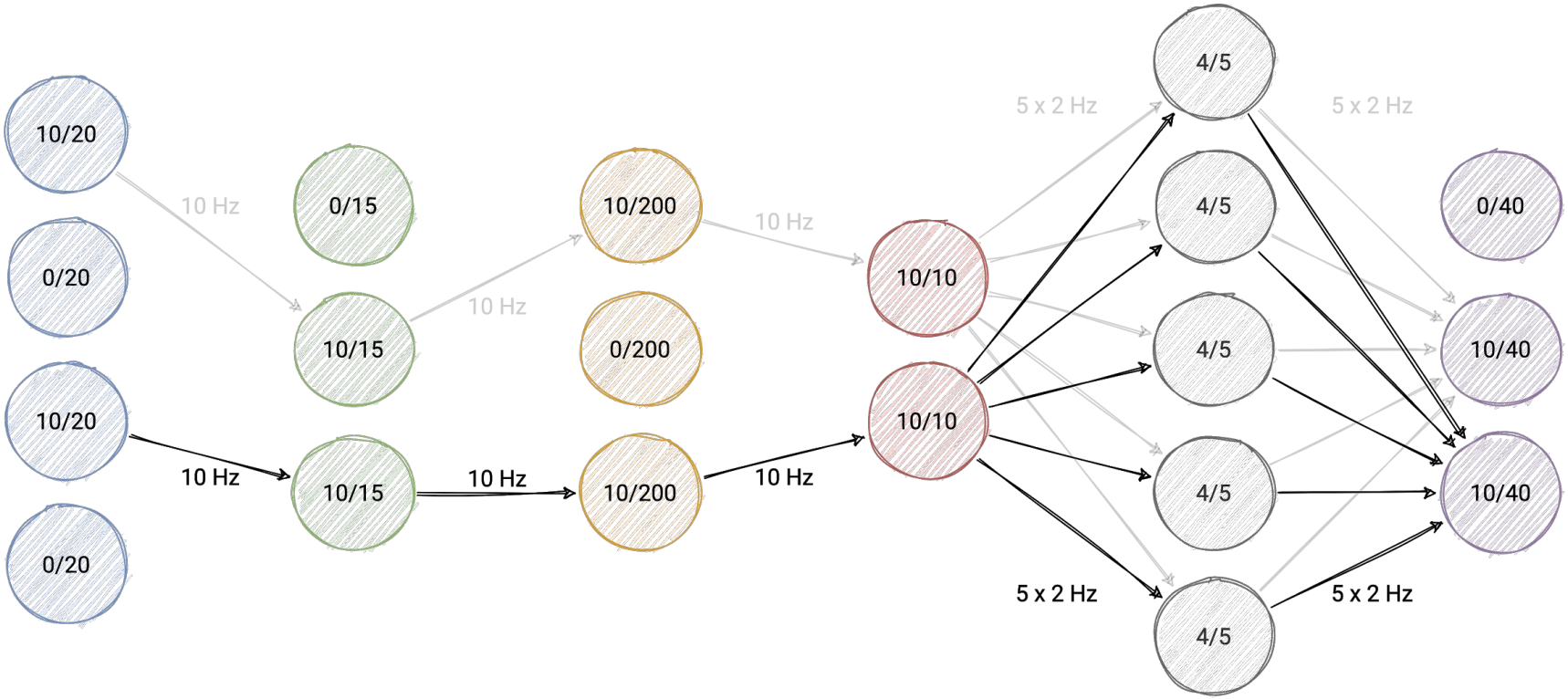


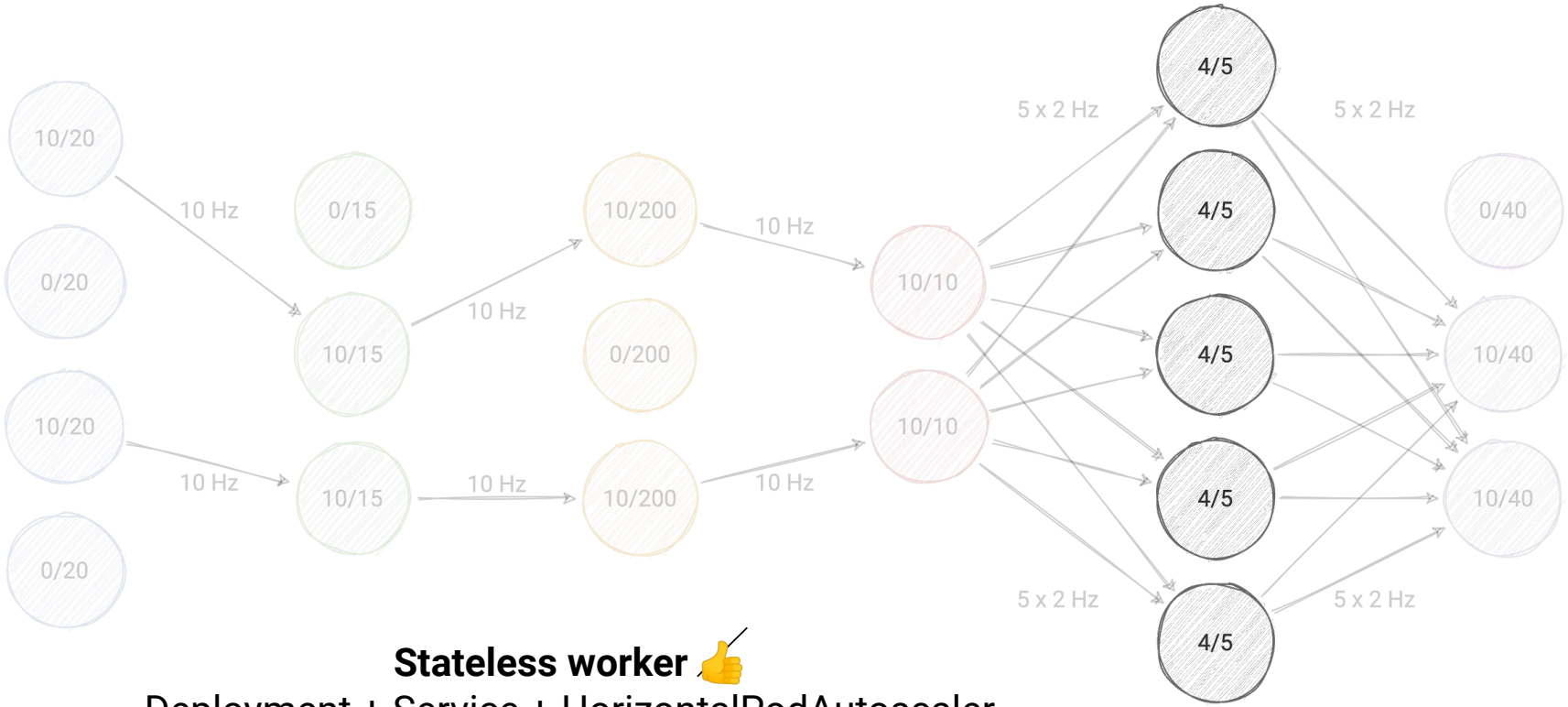



10 Hz
10 Hz
10 Hz
10 Hz
10 Hz


10/20
0/20
0/20
0/20
10 Hz
0/15
10/15
0/15
10 Hz
10/200
0/200
0/200
10 Hz
10/10
0/10
10 Hz
0/5
10/5
0/5
0/5
0/5
10 Hz
0/40
10/40
0/40

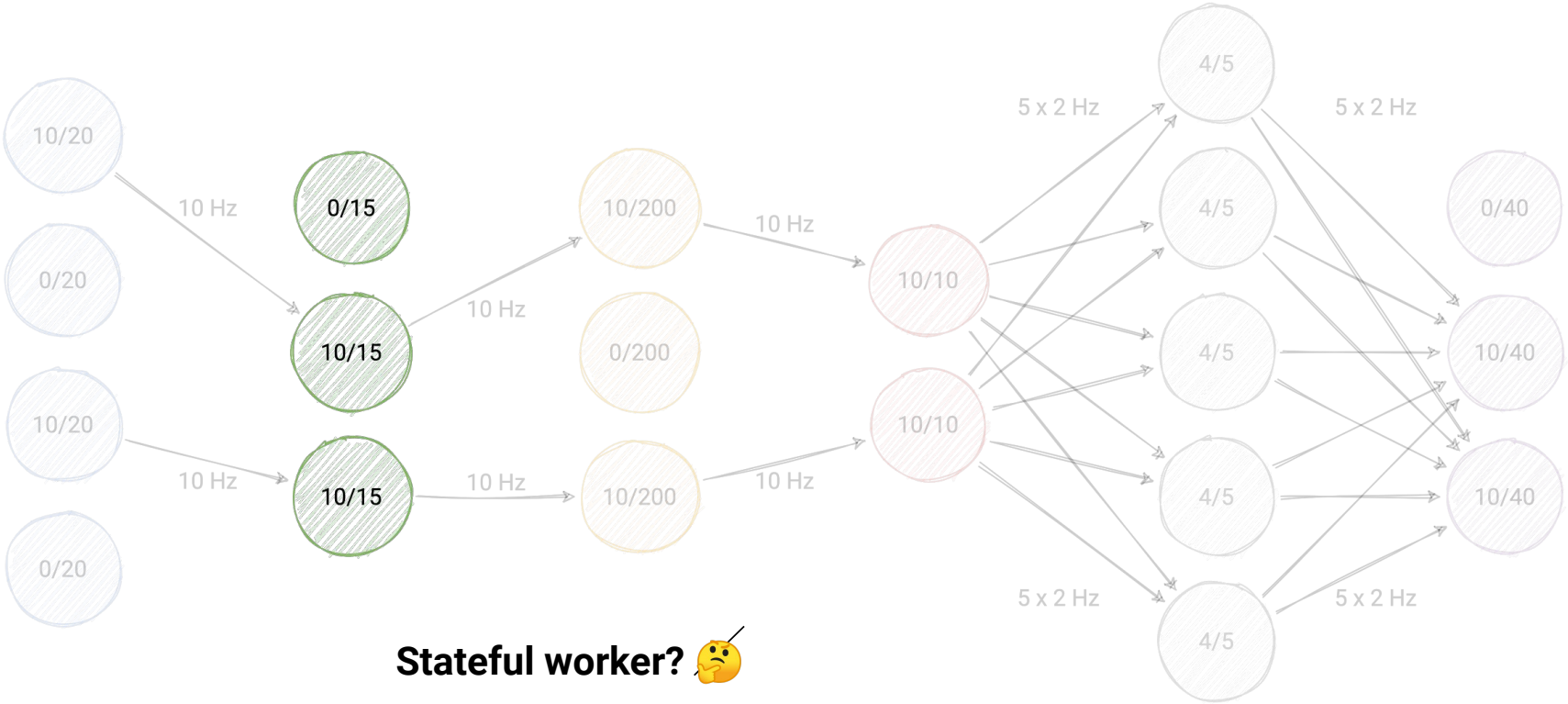

10/20
0/20
0/20
0/20
10 Hz
0/15
10/15
0/15
10 Hz
10/200
0/200
0/200
10 Hz
10/10
0/10
5 x 2 Hz
2/5
2/5
2/5
2/5
2/5
5 x 2 Hz
0/40
10/40
0/40


10/20
0/20
10/20
0/20
10 Hz
10 Hz
0/15
10/15
10/15
10 Hz
10 Hz
10/200
0/200
10/200
10 Hz
10 Hz
10/10
10/10
5 x 2 Hz
5 x 2 Hz
4/5
4/5
4/5
4/5
4/5
5 x 2 Hz
5 x 2 Hz
0/40
10/40
10/40



Stateless worker 👍

Deployment + Service + HorizontalPodAutoscaler



Stateful worker? 🤔


0/30


0/30

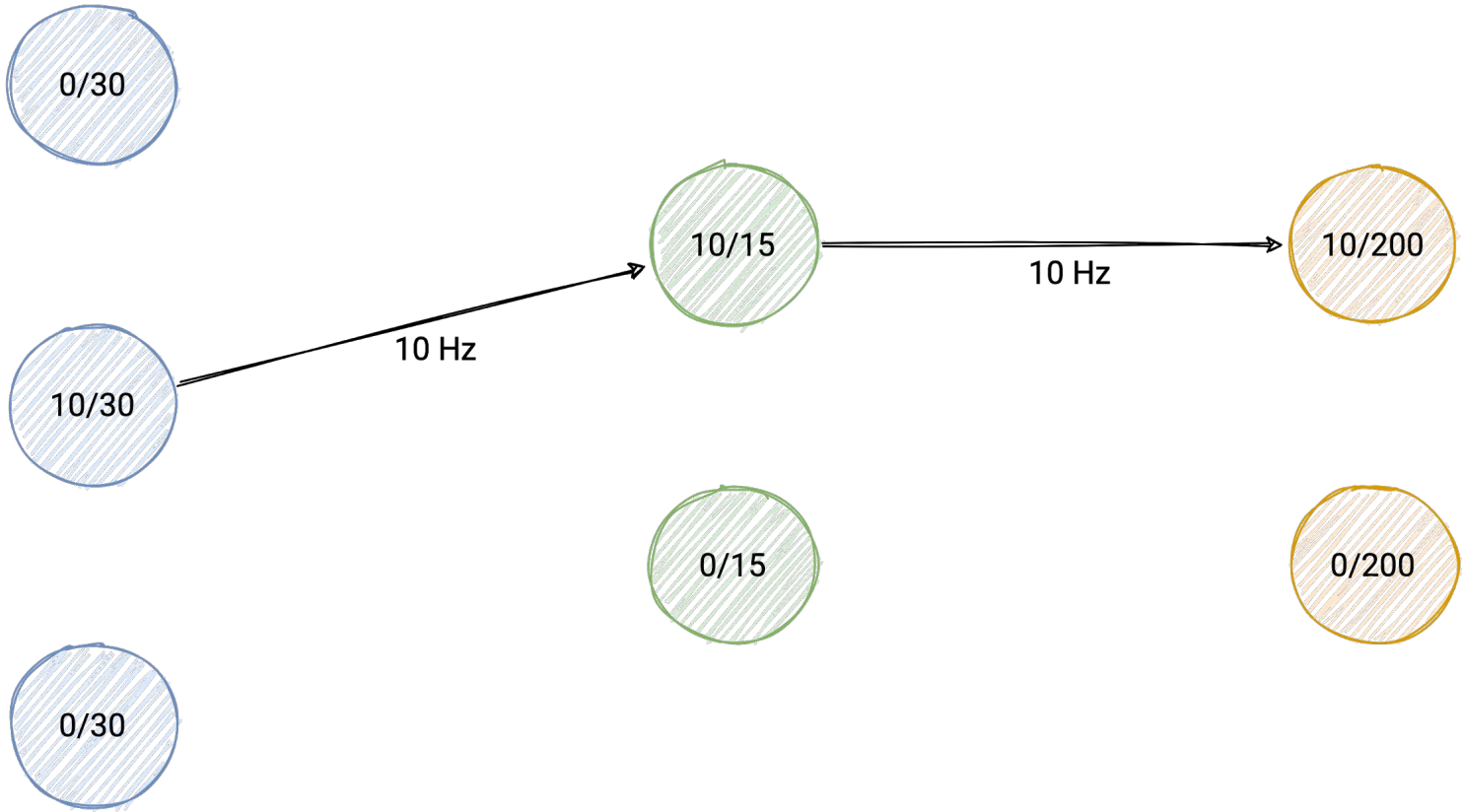

0/30

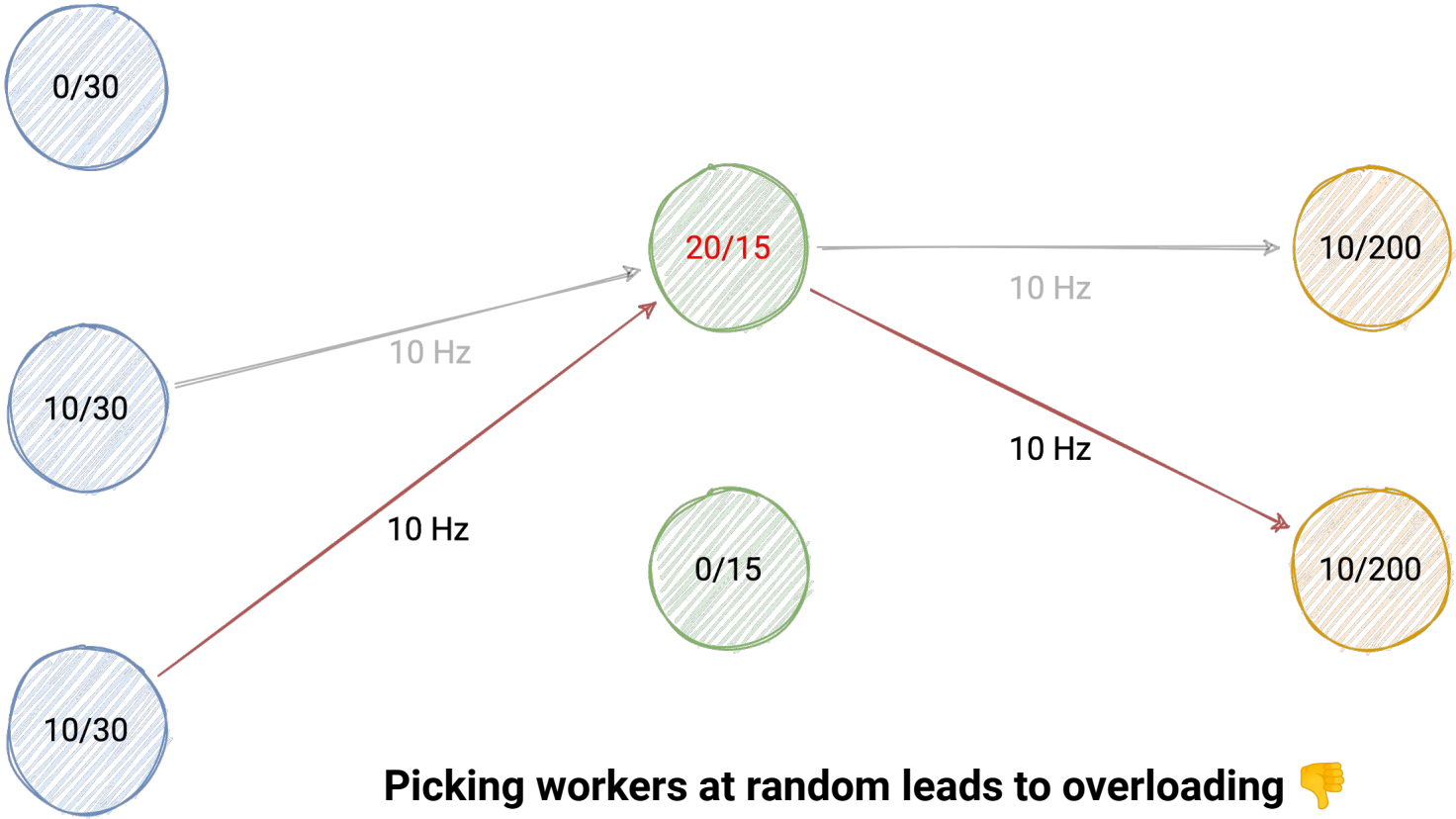

0/15


0/15

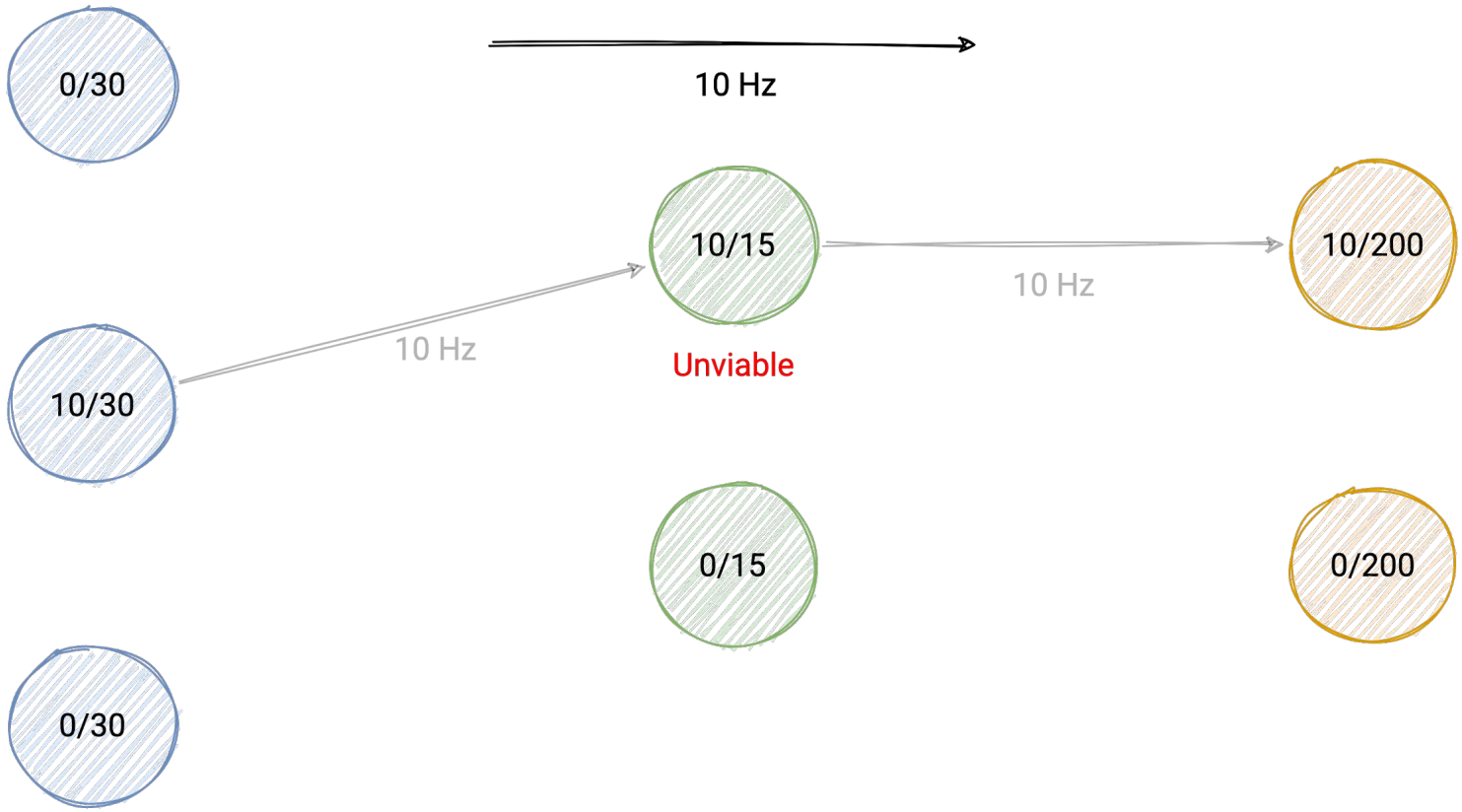

0/200

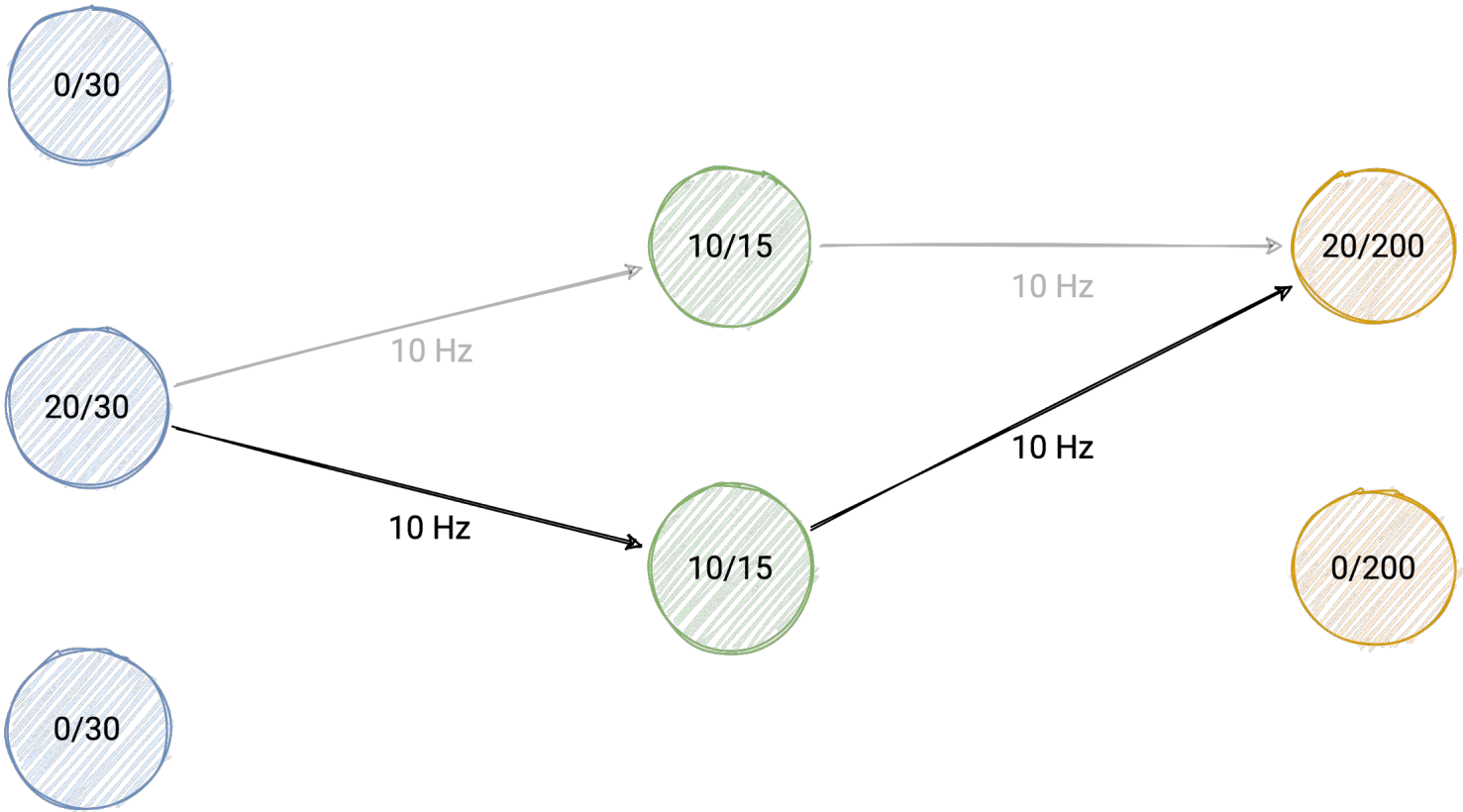

0/200

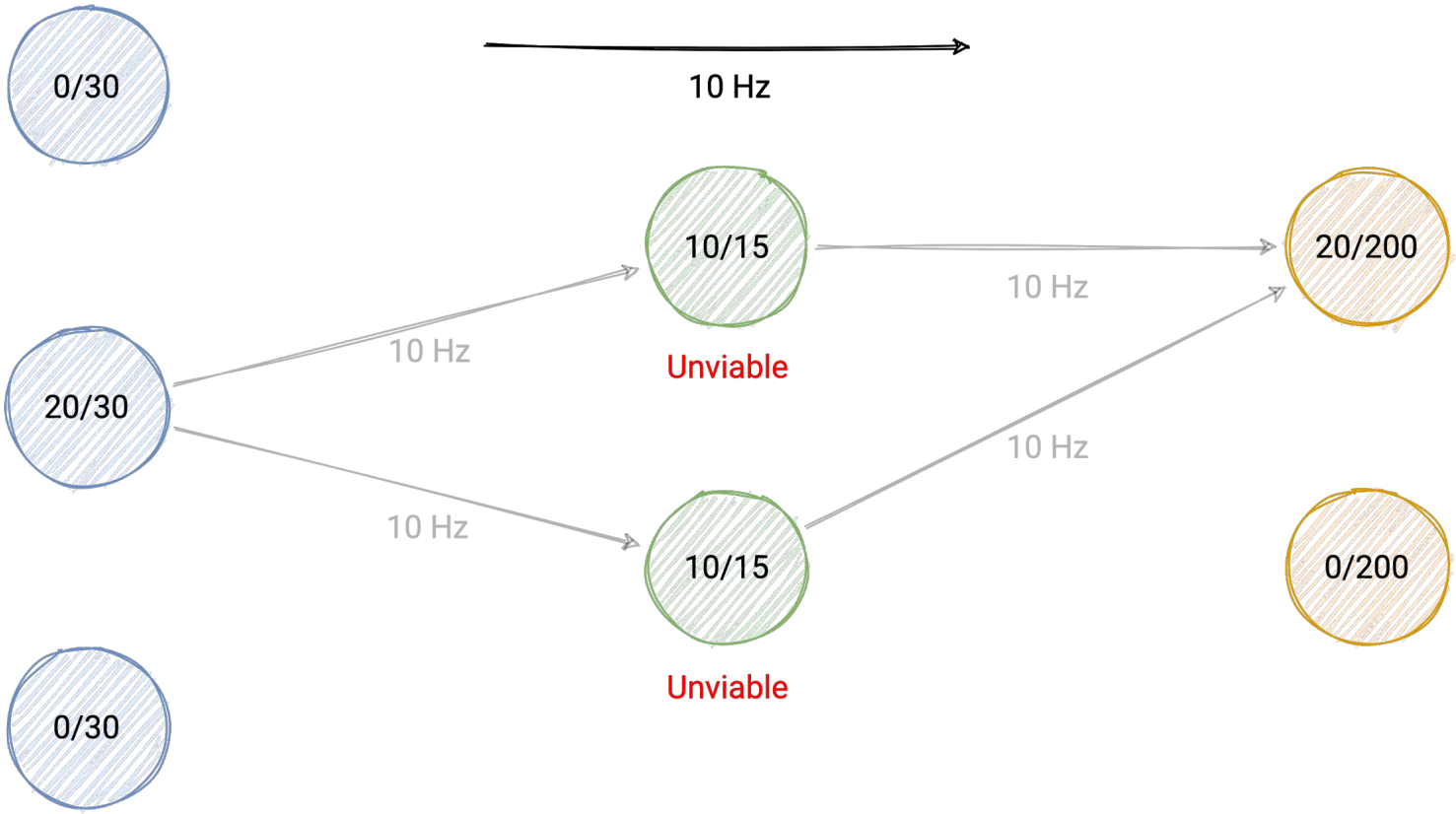

0/30
10/30
0/30
10 Hz
10/15
0/15
10 Hz
10/200
0/200

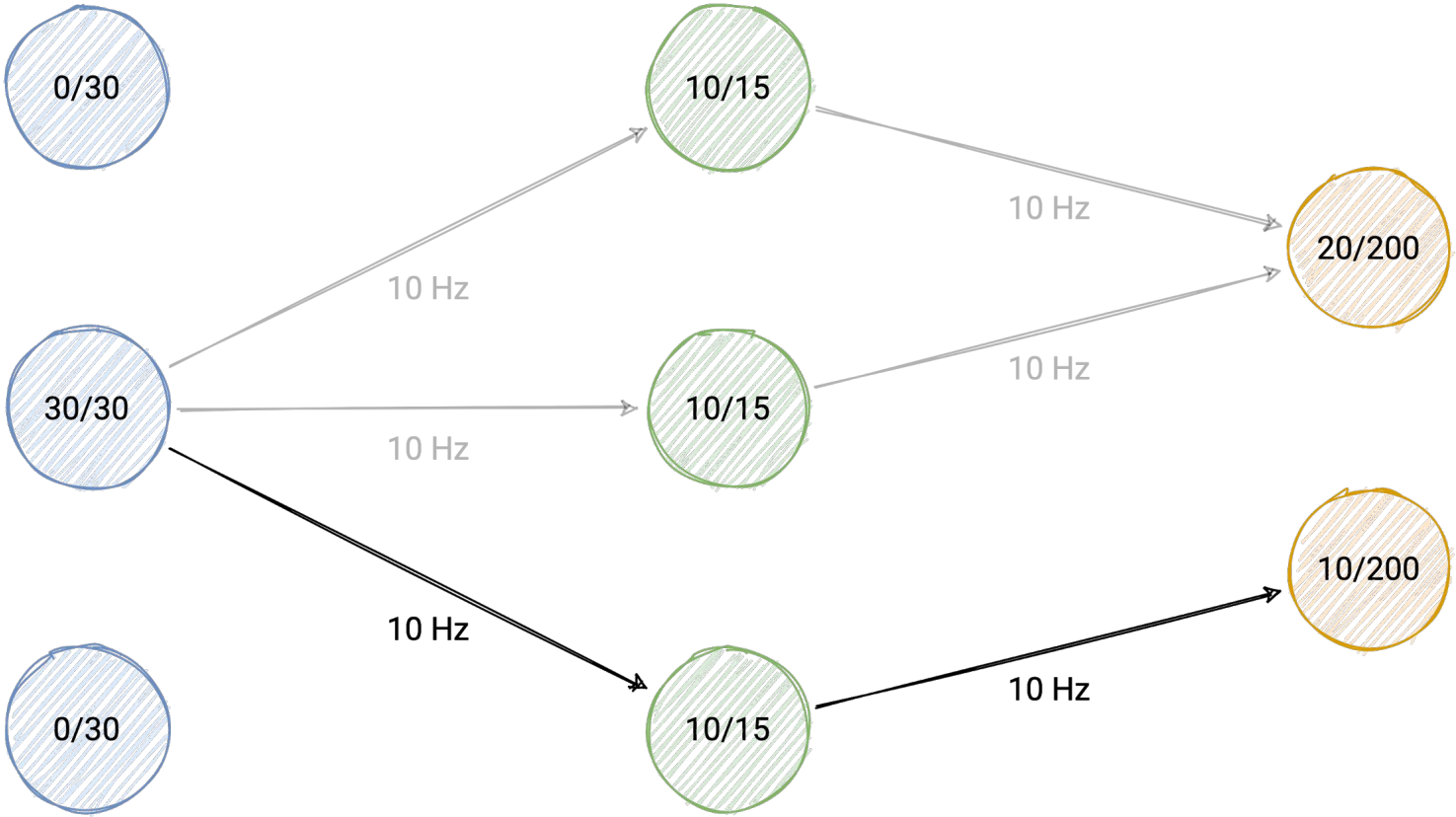


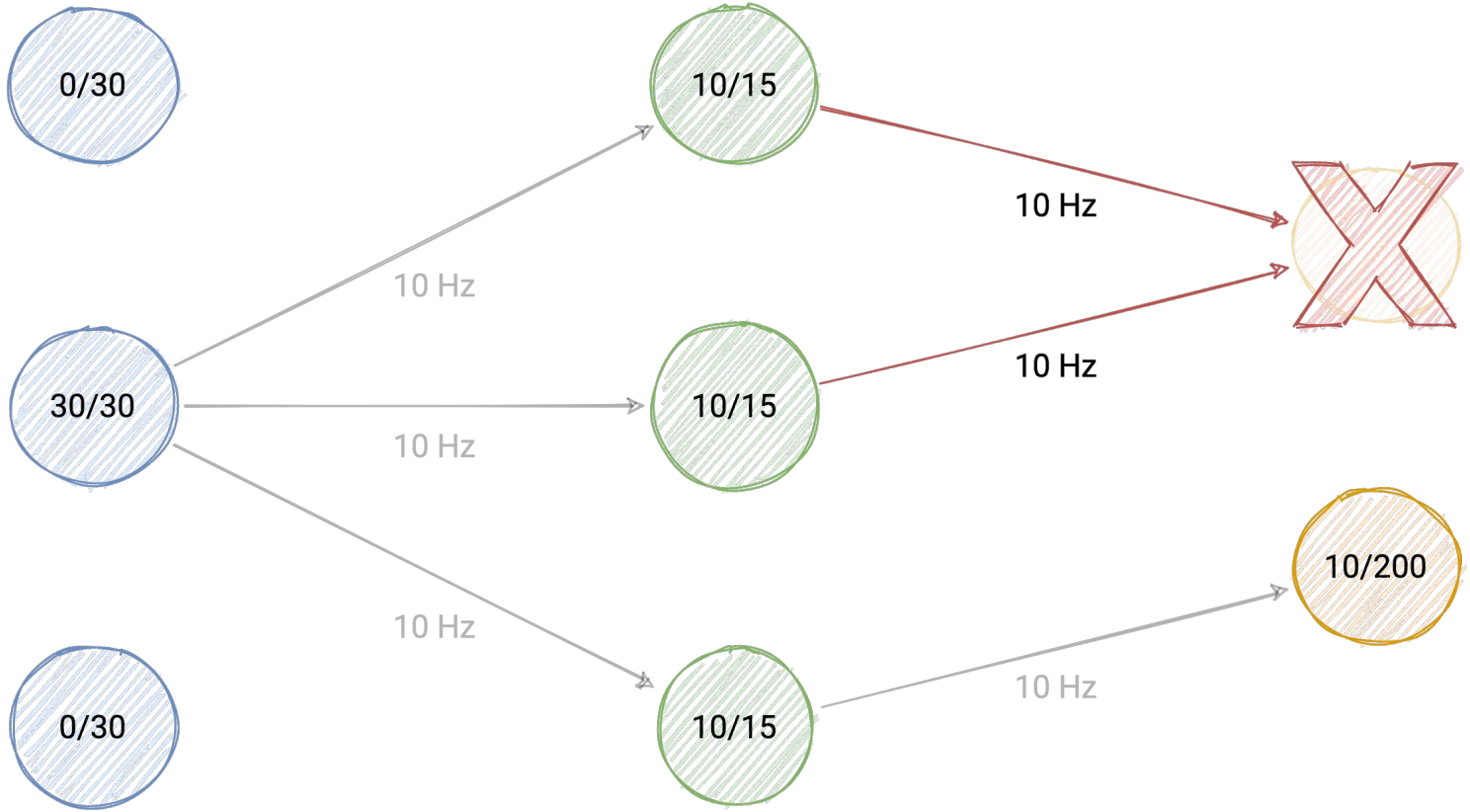
Picking workers at random leads to overloading 👎

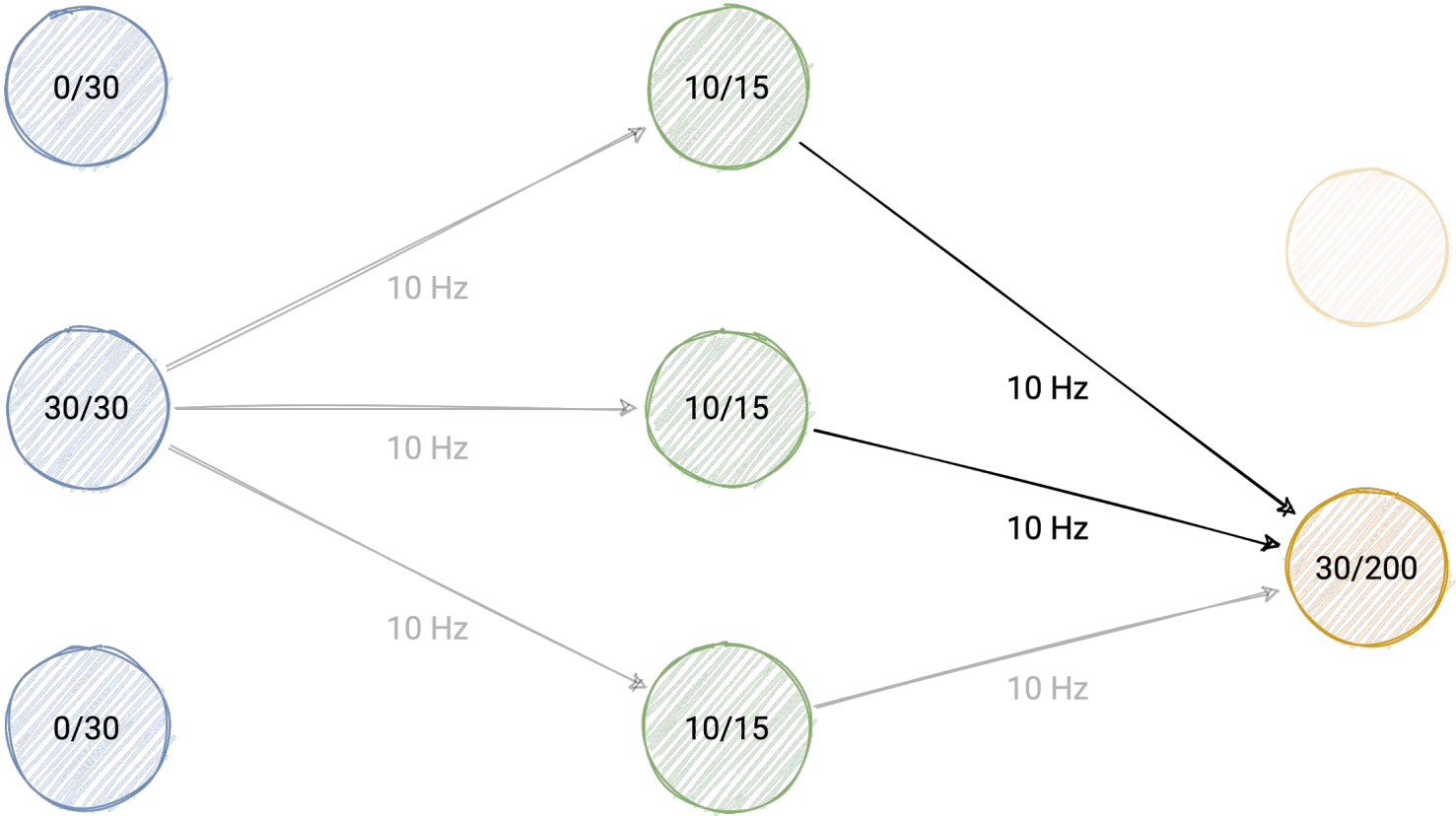

0/30
10 Hz
10/15
10/200
10 Hz
10/30
10 Hz
Unviable
0/15
0/200
0/30


0/30
10/15
20/200
10 Hz
10 Hz
20/30
10 Hz
10 Hz
10/15
0/200
0/30


0/30
10 Hz
10/15
20/200
10 Hz
20/30
10 Hz
Unviable
10 Hz
10 Hz
10/15
0/200
0/30
Unviable


0/30
10/15
20/200
10 Hz
10 Hz
30/30
10/15
10 Hz
10 Hz
10/200
10 Hz
0/30
10/15
10 Hz


0/30
30/30
0/30
10/15
10/15
10/15
10/200
10 Hz
10 Hz
10 Hz
10 Hz
10 Hz
10 Hz


0/30
30/30
0/30
10/15
10/15
10/15
10 Hz
10 Hz
10 Hz
10 Hz
10 Hz
10 Hz
30/200

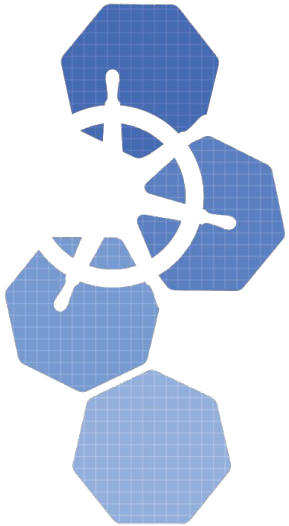
Functional Requirements:

Select viable workers for each pipeline

Scale number of workers based on load

React to changes in worker status

- Why we built it
- **What we built it with**
- How it works
- Lessons learned
- Overall experience



kubebuilder

```
kubebuilder init --domain=k8s.xxii.io
```

Custom Resource Definitions

Pipeline

PipelineBinding

WorkerClass

```
kubebuilder create api --group=scheduling --version=v1alpha1 --kind=Pipeline

kubebuilder create api --group=scheduling --version=v1alpha1 --kind=PipelineBinding

kubebuilder create api --group=scheduling --version=v1alpha1 --kind=WorkerClass
```

```go
// PipelineSpec defines the desired state of Route
type PipelineSpec struct {
    // Number of documents that will be sent along this Pipeline every second.
    // +kubebuilder:validation:Minimum=0
    DocumentsPerSecond int `json:"documentsPerSecond"`

    // The directed acyclic graph of workers to use in this Pipeline.
    Workers map[string]Worker `json:"workers"`
}

// Worker is a node in a Pipeline's graph of workers.
type Worker struct {
    // Name of the WorkerClass to get the worker's specification from.
    // +kubebuilder:validation:MinLength=1
    Class string `json:"class"`

    // Next workers in the the Pipelines's graph.
    Next []string `json:"next"`
}
```

```
make manifests
make install
```

```yaml
# Sample Pipeline
apiVersion: scheduling.k8s.xxii.io/v1alpha1
kind: Pipeline
metadata:
  name: my-pipeline
spec:
  documentsPerSecond: 10
  workers:
    first:
      class: blue
      next:
        - second
    second:
      class: green
      next:
        - third
    third:
      class: yellow
      next: []
status: {}
```

```
// Reconcile the cluster state based on a Pipeline's spec.
func (r *PipelineReconciler) Reconcile(req ctrl.Request) (ctrl.Result, error) {
    var pipeline scheduling.Pipeline
    r.get(req, &pipeline)

    r.updateStatus(&pipeline)

    ok := r.selectWorkers(&pipeline)
    if !ok {
        r.unsetBindings(&pipeline)
        return ctrl.Result{RequeueAfter: time.Second}, nil
    }

    r.createBindings(&pipeline)

    return ctrl.Result{}, nil
}
```

```
# HELP workerclass_request Load the WorkerClass needs to process
# TYPE workerclass_request gauge
workerclass_request{workerclass_name="blue"  ,workerclass_namespace="default"} 30
workerclass_request{workerclass_name="green" ,workerclass_namespace="default"} 75
workerclass_request{workerclass_name="yellow",workerclass_namespace="default"} 110
workerclass_request{workerclass_name="red"   ,workerclass_namespace="default"} 0
workerclass_request{workerclass_name="grey"  ,workerclass_namespace="default"} 0
workerclass_request{workerclass_name="purple",workerclass_namespace="default"} 60
```

```yaml
# Prometheus Adapter Rules
rules:
  custom:
    - seriesQuery: '{__name__=~"^workerclass_.*"}'
      name:
        matches: ^(workerclass_.*)$
        as: ${1}
      resources:
        overrides:
          workerclass_namespace:
            resource: namespace
          workerclass_name:
            group: scheduling.k8s.xxii.io
            resource: workerclass
      metricsQuery: <<.Series>>{<<.LabelMatchers>>}
```

Horizontal Pod Autoscaler

Core Kubernetes feature

```yaml
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata: ...
spec:
  minReplicas: 0
  maxReplicas: 10
  scaleTargetRef:
    apiVersion: apps/v1
    kind: StatefulSet
    name: blue-workers
  metrics:
    - type: Object
      object:
        metric:
          name: workerclass_request
        describedObject:
          apiVersion: scheduling.k8s.xxii.io/v1alpha1
          kind: WorkerClass
          name: blue
        target:
          type: Value
          averageValue: '20'
```

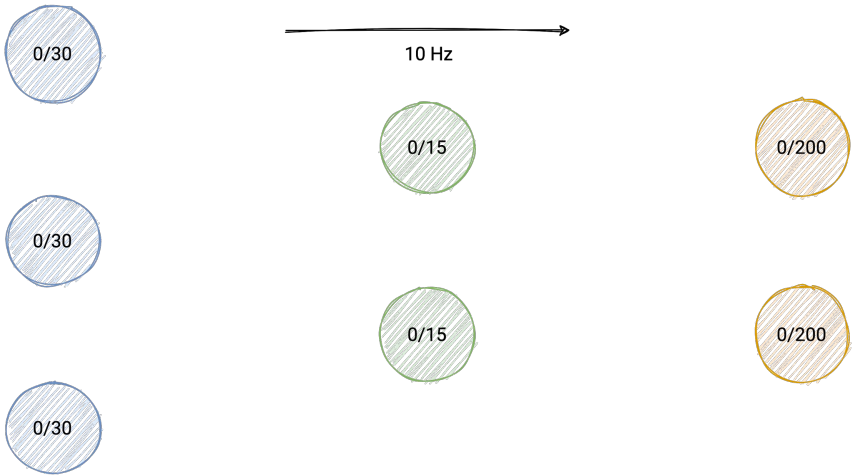
- Why we built it
- What we built it with
- How it works
- Development
- Lessons learned
- Overall experience

Select viable workers for each pipeline

Scale number of workers based on load

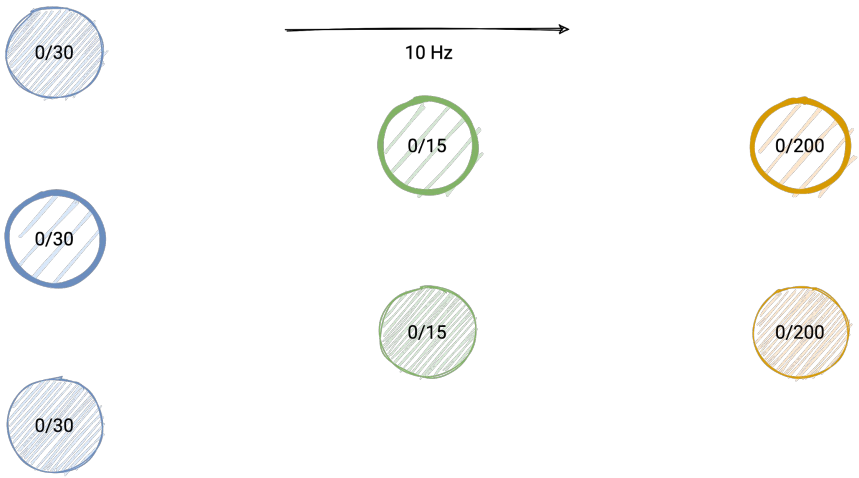
React to changes in worker status


```
WorkerClass     NAME      CAPACITY
                blue      30
                green     15
                yellow    200
```

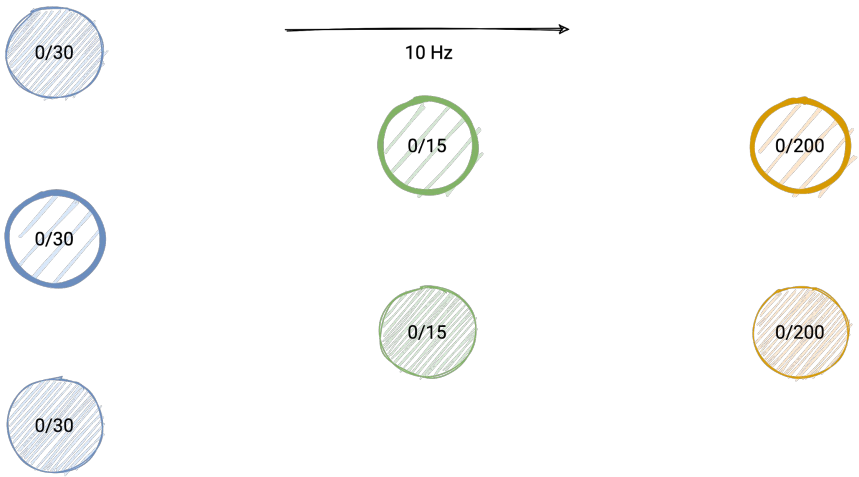



```
WorkerClass   NAME      CAPACITY
              blue      30
              green     15
              yellow    200
```

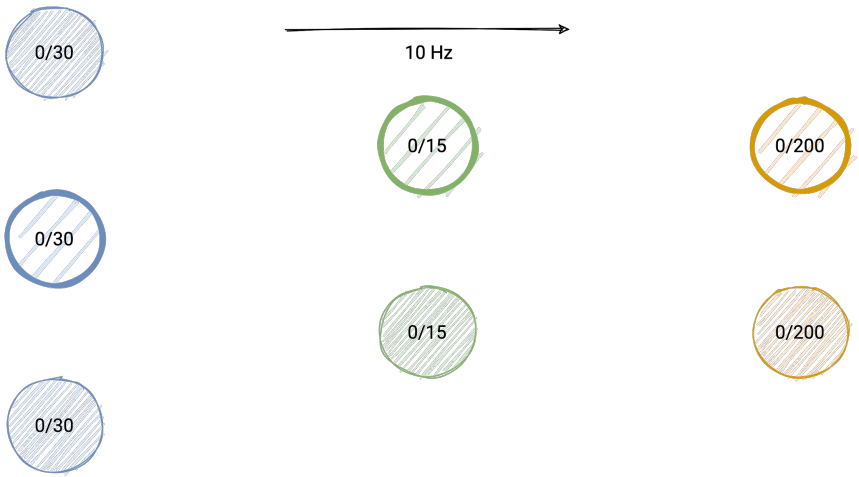
Pipeline	NAME	LOAD	STATUS
	first	10	Pending



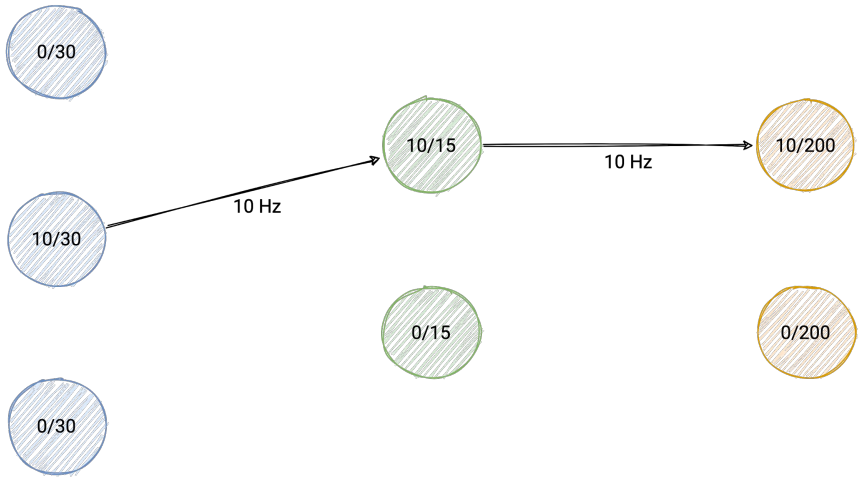

```
WorkerClass   NAME      CAPACITY
              blue      30
              green     15
              yellow    200

Pipeline      NAME      LOAD    STATUS
              first     10      Plotted

PipelineBinding   NAME              STATUS
                  first-blue        Pending
                  first-green       Pending
                  first-yellow      Pending
```



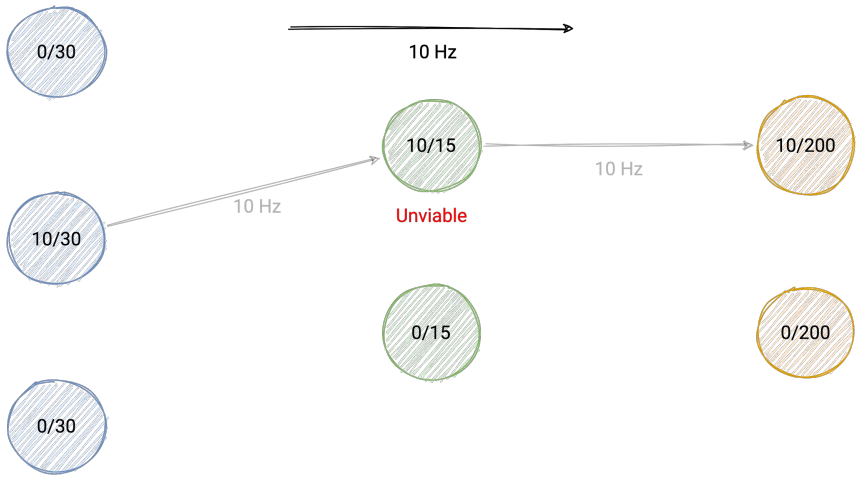

```
WorkerClass   NAME      CAPACITY
              blue      30
              green     15
              yellow    200

Pipeline      NAME      LOAD    STATUS
              first     10      Plotted

PipelineBinding   NAME              STATUS
                  first-blue        Pending
                  first-green       Pending
                  first-yellow      Pending

Service       NAME              ENDPOINTS
              first-blue        blue-1
              first-green       green-0
              first-yellow      yellow-0
```



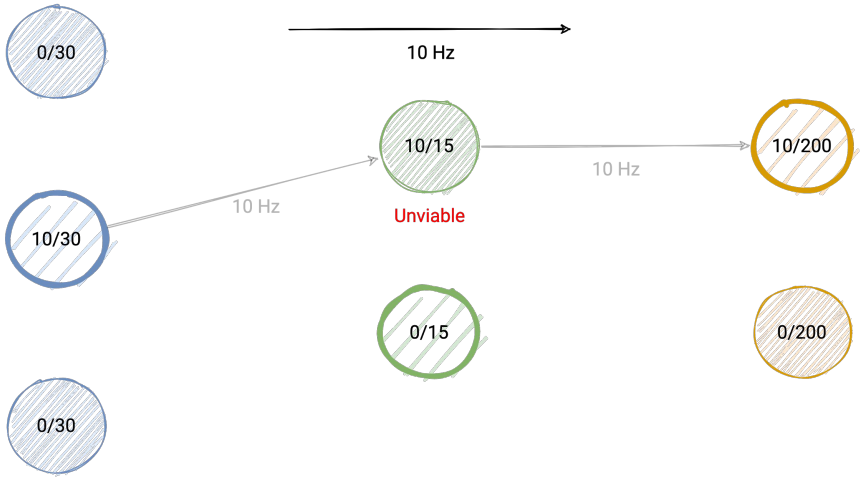

```
WorkerClass   NAME      CAPACITY
              blue      30
              green     15
              yellow    200

Pipeline      NAME      LOAD    STATUS
              first     10      Plotted

PipelineBinding   NAME              STATUS
                  first-blue        Ready
                  first-green       Ready
                  first-yellow      Ready

Service       NAME              ENDPOINTS
              first-blue        blue-1
              first-green       green-0
              first-yellow      yellow-0
```



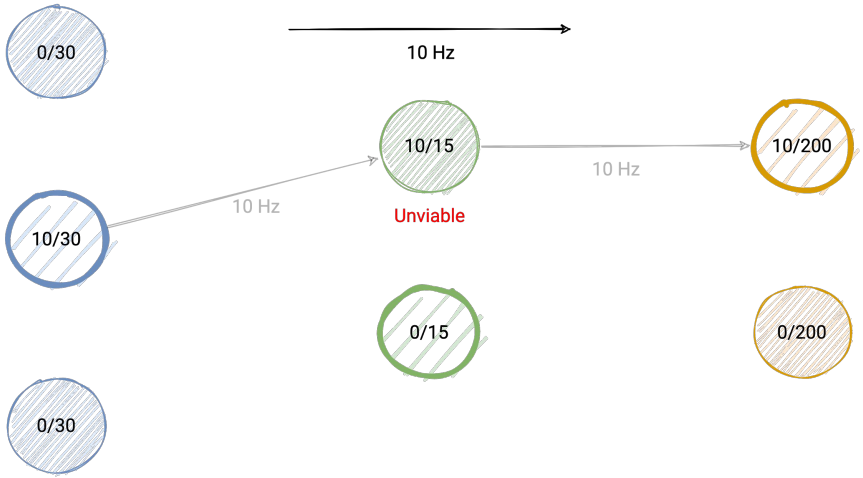

```
WorkerClass   NAME      CAPACITY
              blue      30
              green     15
              yellow    200

Pipeline      NAME      LOAD    STATUS
              first     10      Ready

PipelineBinding   NAME              STATUS
                  first-blue        Ready
                  first-green       Ready
                  first-yellow      Ready

Service       NAME              ENDPOINTS
              first-blue        blue-1
              first-green       green-0
              first-yellow      yellow-0
```



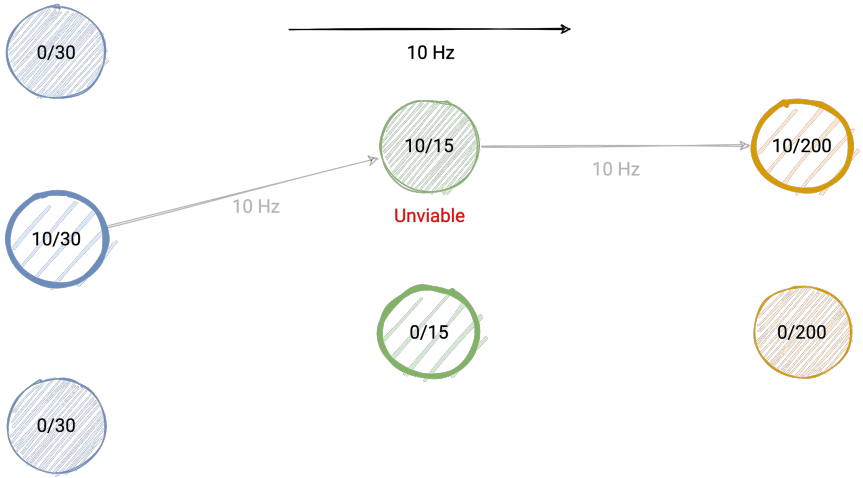

```
WorkerClass   NAME      CAPACITY
              blue      30
              green     15
              yellow    200

Pipeline      NAME      LOAD    STATUS
              first     10      Ready
              second    10      Pending

PipelineBinding   NAME              STATUS
                  first-blue        Ready
                  first-green       Ready
                  first-yellow      Ready

Service       NAME              ENDPOINTS
              first-blue        blue-1
              first-green       green-0
              first-yellow      yellow-0
```



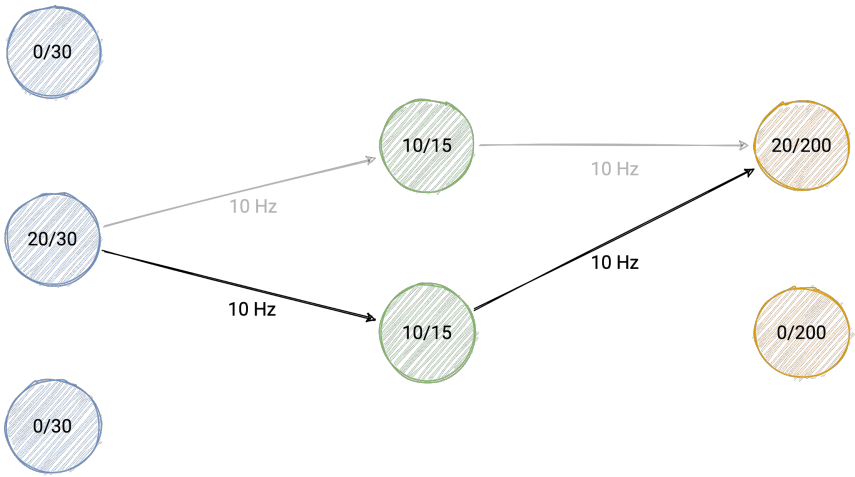

```
WorkerClass   NAME      CAPACITY
              blue      30
              green     15
              yellow    200

Pipeline      NAME      LOAD    STATUS
              first     10      Ready
              second    10      Plotted

PipelineBinding   NAME              STATUS
                  first-blue        Ready
                  first-green       Ready
                  first-yellow      Ready
                  second-blue       Pending
                  second-green      Pending
                  second-yellow     Pending

Service       NAME              ENDPOINTS
              first-blue        blue-1
              first-green       green-0
              first-yellow      yellow-0
```



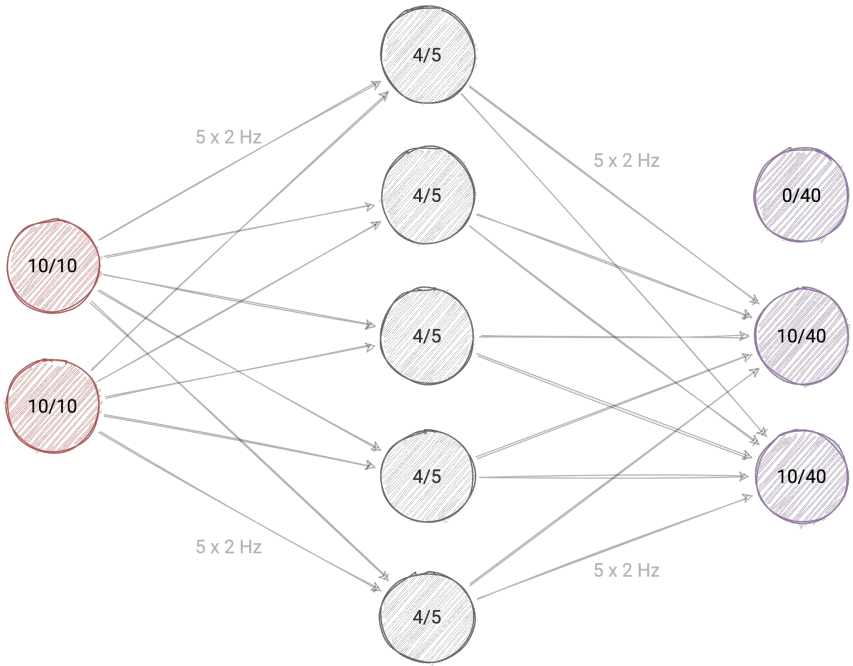

```
WorkerClass   NAME      CAPACITY
              blue      30
              green     15
              yellow    200

Pipeline      NAME      LOAD    STATUS
              first     10      Ready
              second    10      Plotted

PipelineBinding    NAME              STATUS
                   first-blue        Ready
                   first-green       Ready
                   first-yellow      Ready
                   second-blue       Pending
                   second-green      Pending
                   second-yellow     Pending

Service       NAME              ENDPOINTS
              first-blue        blue-1
              first-green       green-0
              first-yellow      yellow-0
              second-blue       blue-1
              second-green      green-1
              second-yellow     yellow-0
```




```
WorkerClass    NAME      CAPACITY
               blue      30
               green     15
               yellow    200

Pipeline    NAME      LOAD    STATUS
            first     10      Ready
            second    10      Plotted

PipelineBinding    NAME              STATUS
                   first-blue        Ready
                   first-green       Ready
                   first-yellow      Ready
                   second-blue       Ready
                   second-green      Ready
                   second-yellow     Ready

Service    NAME              ENDPOINTS
           first-blue        blue-1
           first-green       green-0
           first-yellow      yellow-0
           second-blue       blue-1
           second-green      green-1
           second-yellow     yellow-0
```




```
WorkerClass    NAME      CAPACITY
               blue      30
               green     15
               yellow    200

Pipeline    NAME      LOAD    STATUS
            first     10      Ready
            second    10      Ready

PipelineBinding    NAME              STATUS
                   first-blue        Ready
                   first-green       Ready
                   first-yellow      Ready
                   second-blue       Ready
                   second-green      Ready
                   second-yellow     Ready

Service    NAME              ENDPOINTS
           first-blue        blue-1
           first-green       green-0
           first-yellow      yellow-0
           second-blue       blue-1
           second-green      green-1
           second-yellow     yellow-0
```

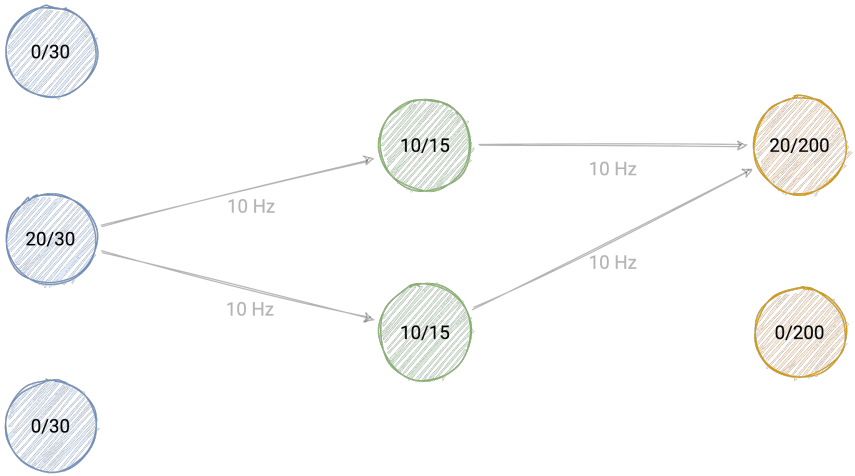



```
WorkerClass   NAME      CAPACITY
              red       10
              grey      5
              purple    40

Pipeline      NAME      LOAD    STATUS
              first     10      Ready
              second    10      Ready

PipelineBinding   NAME              STATUS
                  first-red         Ready
                  first-grey        Ready
                  first-purple      Ready
                  second-red        Ready
                  second-grey       Ready
                  second-purple     Ready

Service       NAME              ENDPOINTS
              first-red         red-0
              first-grey        grey-0,1,2,3,4
              first-purple      purple-1
              second-red        red-1
              second-grey       grey-0,1,2,3,4
              second-purple     purple-2
```

Select viable workers for each pipeline

Scale number of workers based on load

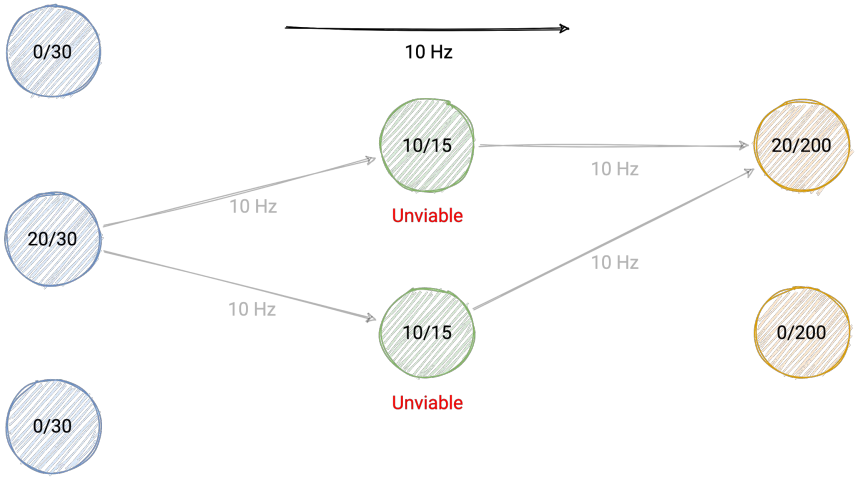
React to changes in worker status




```
WorkerClass   NAME      CAPACITY
              blue      30
              green     15
              yellow    200
```

HPA	NAME	TARGET	ACTUAL	REPLICAS
	blue	**70%**	**22%**	**3**
	green	**70%**	**66%**	**2**
	yellow	**70%**	**5%**	**2**

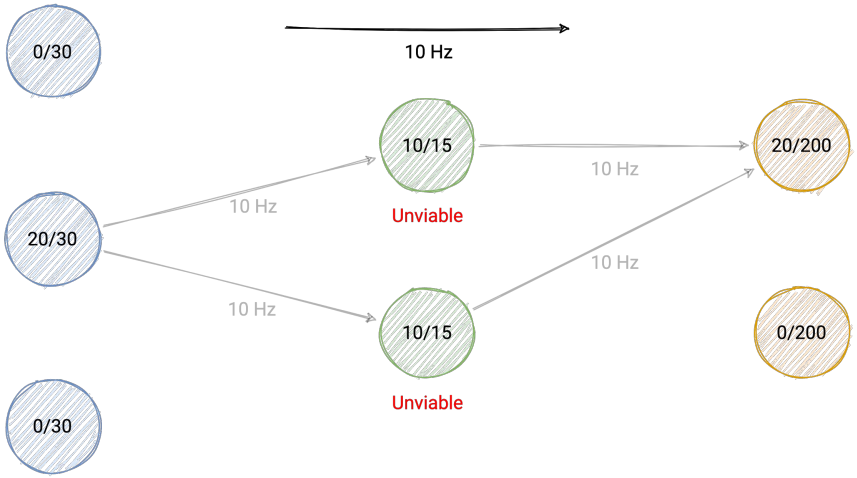
```
Pipeline   NAME      LOAD   STATUS
           first     10     Ready
           second    10     Ready
```



WorkerClass	NAME	CAPACITY
	blue	30
	green	15
	yellow	200

HPA	NAME	TARGET	ACTUAL	REPLICAS
	blue	70%	22%	3
	green	70%	66%	2
	yellow	70%	5%	2

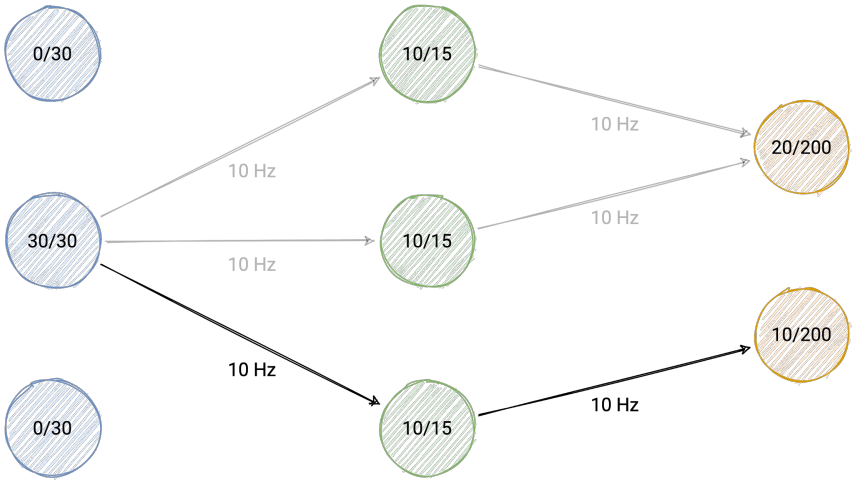
Pipeline	NAME	LOAD	STATUS
	first	10	Ready
	second	10	Ready
	third	**10**	**Pending**



WorkerClass	NAME	CAPACITY
	blue	30
	green	15
	yellow	200

HPA	NAME	TARGET	ACTUAL	REPLICAS
	blue	70%	**33%**	3
	green	70%	**100%**	2
	yellow	70%	**7%**	2

Pipeline	NAME	LOAD	STATUS
	first	10	Ready
	second	10	Ready
	third	10	Pending



WorkerClass	NAME	CAPACITY
	blue	30
	green	15
	yellow	200

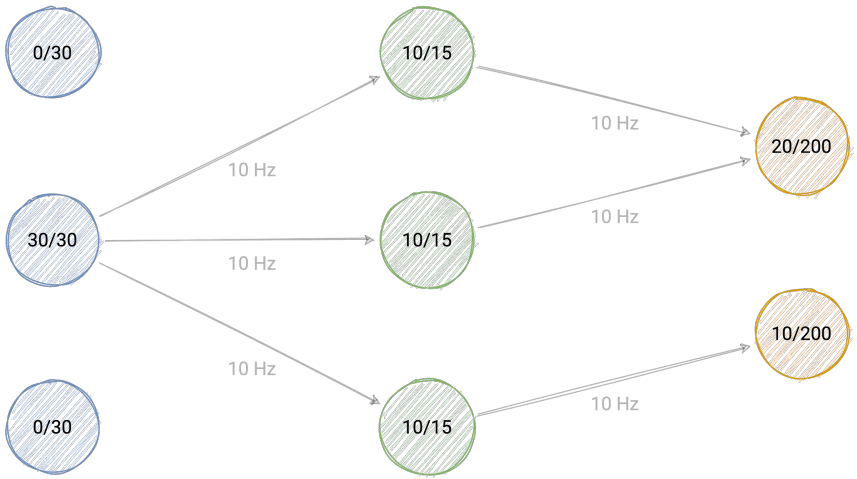
HPA	NAME	TARGET	ACTUAL	REPLICAS
	blue	70%	33%	3
	green	70%	**66%**	**3**
	yellow	70%	7%	2

Pipeline	NAME	LOAD	STATUS
	first	10	Ready
	second	10	Ready
	third	10	**Ready**

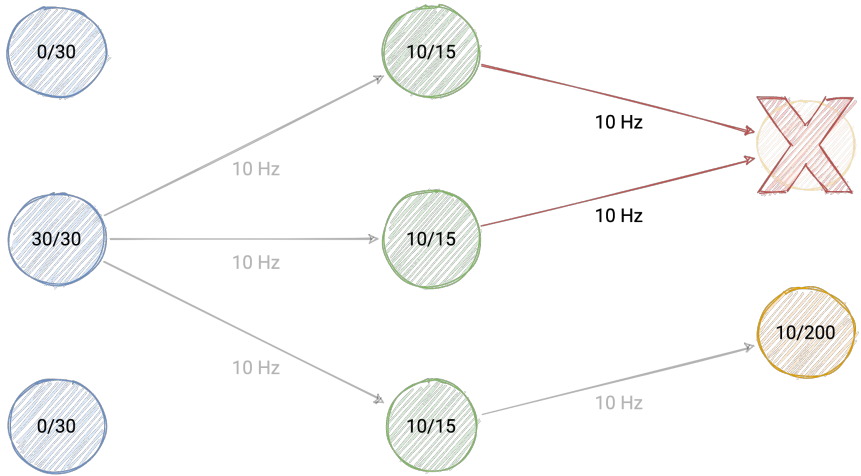
Select viable workers for each pipeline

Scale number of workers based on load

React to changes in worker status



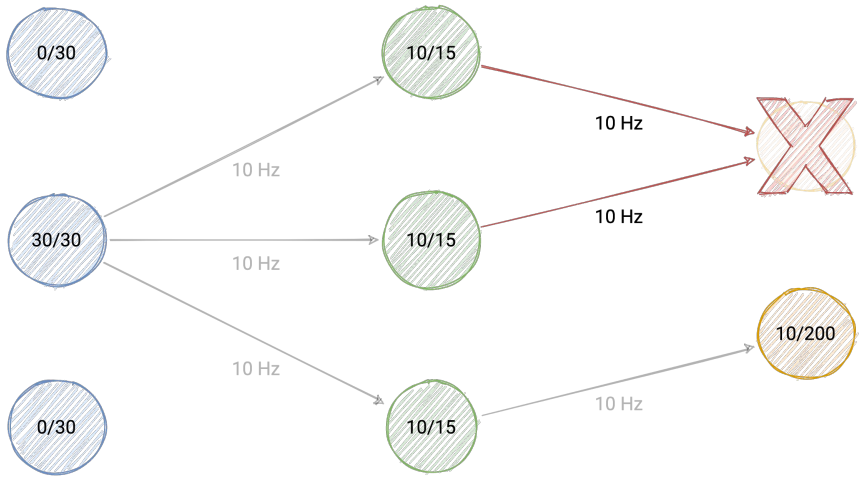

```
WorkerClass   NAME      CAPACITY
              blue      30
              green     15
              yellow    200

Pipeline      NAME      LOAD    STATUS
              first     10      Ready
              second    10      Ready
              third     10      Ready

PipelineBinding   NAME              STATUS
                  first-blue        Ready
                  first-green       Ready
                  first-yellow      Ready
                  second-blue       Ready
                  second-green      Ready
                  second-yellow     Ready
                  third-blue        Ready
                  third-green       Ready
                  third-yellow      Ready
```



WorkerClass	NAME	CAPACITY
	blue	30
	green	15
	yellow	200

Pipeline	NAME	LOAD	STATUS
	first	10	Ready
	second	10	Ready
	third	10	Ready

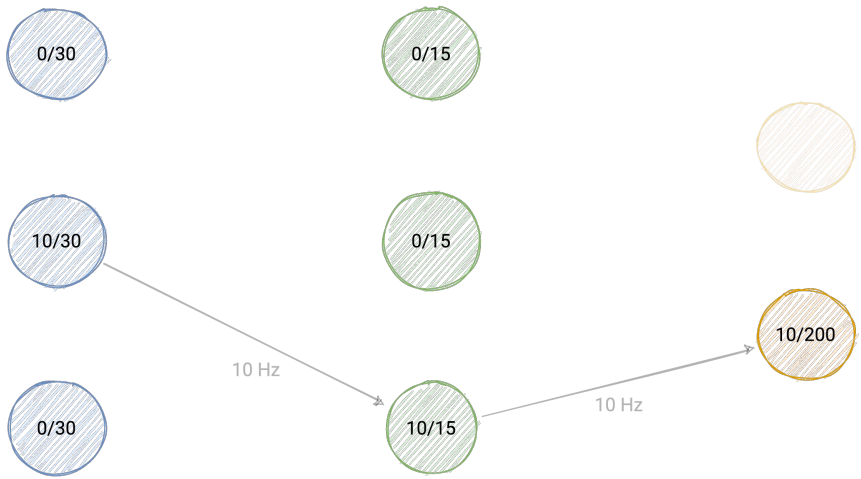
PipelineBinding	NAME	STATUS
	first-blue	Ready
	first-green	Ready
	first-yellow	**DeadEnd**
	second-blue	Ready
	second-green	Ready
	second-yellow	**DeadEnd**
	third-blue	Ready
	third-green	Ready
	third-yellow	Ready



WorkerClass	NAME	CAPACITY
	blue	30
	green	15
	yellow	200

Pipeline	NAME	LOAD	STATUS
	first	10	**Broken**
	second	10	**Broken**
	third	10	Ready

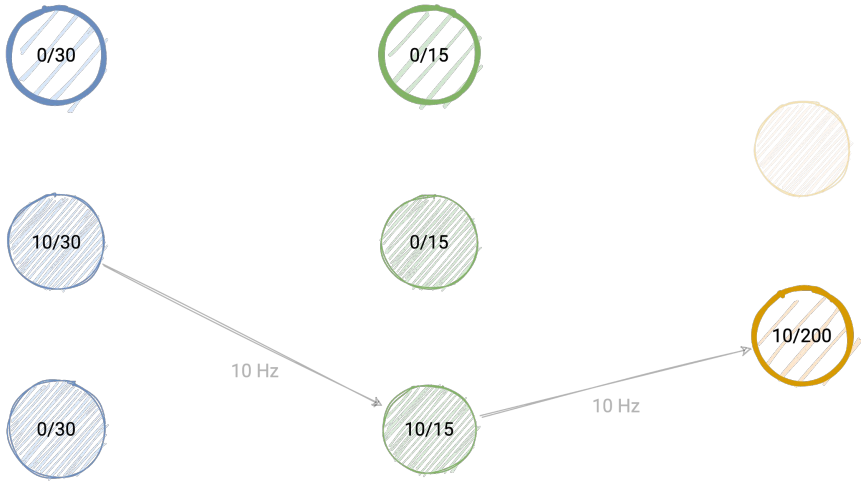
PipelineBinding	NAME	STATUS
	first-blue	Ready
	first-green	Ready
	first-yellow	DeadEnd
	second-blue	Ready
	second-green	Ready
	second-yellow	DeadEnd
	third-blue	Ready
	third-green	Ready
	third-yellow	Ready



WorkerClass	NAME	CAPACITY
	blue	30
	green	15
	yellow	200

Pipeline	NAME	LOAD	STATUS
	first	10	Broken
	second	10	Broken
	third	10	Ready

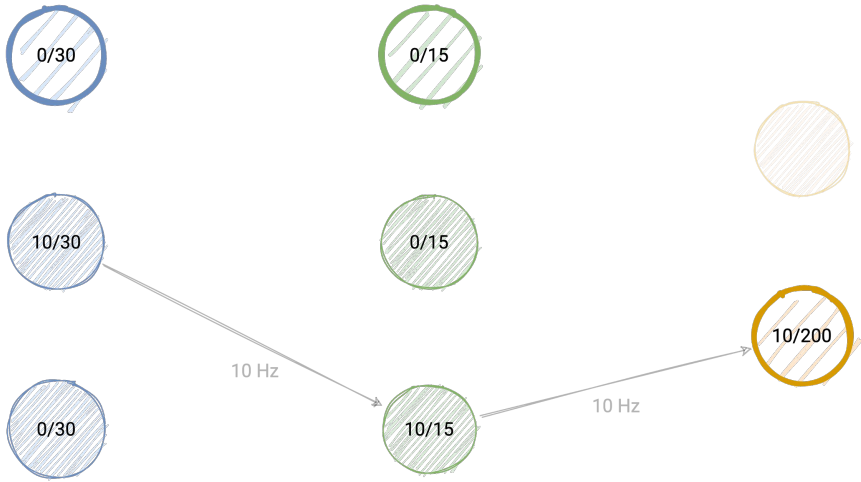
PipelineBinding	NAME	STATUS
	first-blue	**Unset**
	first-green	**Unset**
	first-yellow	**Unset**
	second-blue	**Unset**
	second-green	**Unset**
	second-yellow	**Unset**
	third-blue	Ready
	third-green	Ready
	third-yellow	Ready



WorkerClass	NAME	CAPACITY
	blue	30
	green	15
	yellow	200

Pipeline	NAME	LOAD	STATUS
	first	10	**Plotted**
	second	10	Broken
	third	10	Ready

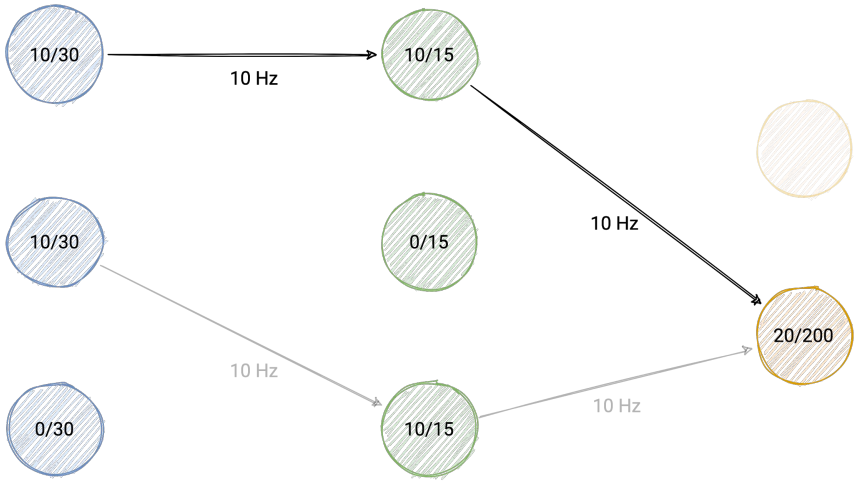
PipelineBinding	NAME	STATUS
	first-blue	**Pending**
	first-green	**Pending**
	first-yellow	**Pending**
	second-blue	Unset
	second-green	Unset
	second-yellow	Unset
	third-blue	Ready
	third-green	Ready
	third-yellow	Ready



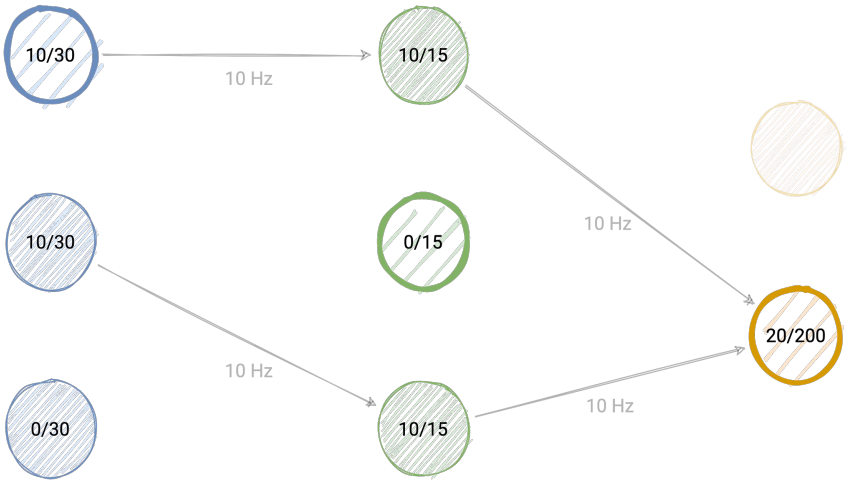
WorkerClass	NAME	CAPACITY
	blue	30
	green	15
	yellow	200

Pipeline	NAME	LOAD	STATUS
	first	10	Plotted
	second	10	Broken
	third	10	Ready

PipelineBinding	NAME	STATUS
	first-blue	**Ready**
	first-green	**Ready**
	first-yellow	**Ready**
	second-blue	Unset
	second-green	Unset
	second-yellow	Unset
	third-blue	Ready
	third-green	Ready
	third-yellow	Ready



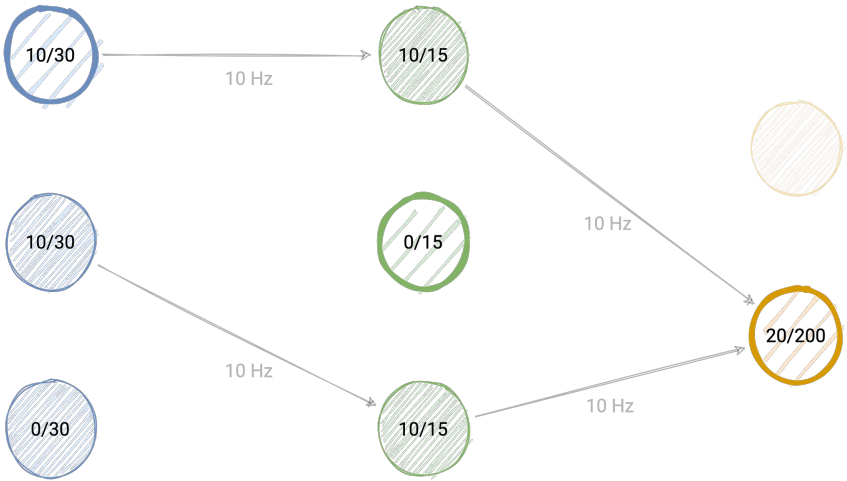

```
WorkerClass   NAME      CAPACITY
              blue      30
              green     15
              yellow    200

Pipeline      NAME      LOAD    STATUS
              first     10      Ready
              second    10      Broken
              third     10      Ready

PipelineBinding   NAME              STATUS
                  first-blue        Ready
                  first-green       Ready
                  first-yellow      Ready
                  second-blue       Unset
                  second-green      Unset
                  second-yellow     Unset
                  third-blue        Ready
                  third-green       Ready
                  third-yellow      Ready
```



WorkerClass	NAME	CAPACITY
	blue	30
	green	15
	yellow	200

Pipeline	NAME	LOAD	STATUS
	first	10	Ready
	second	10	**Plotted**
	third	10	Ready

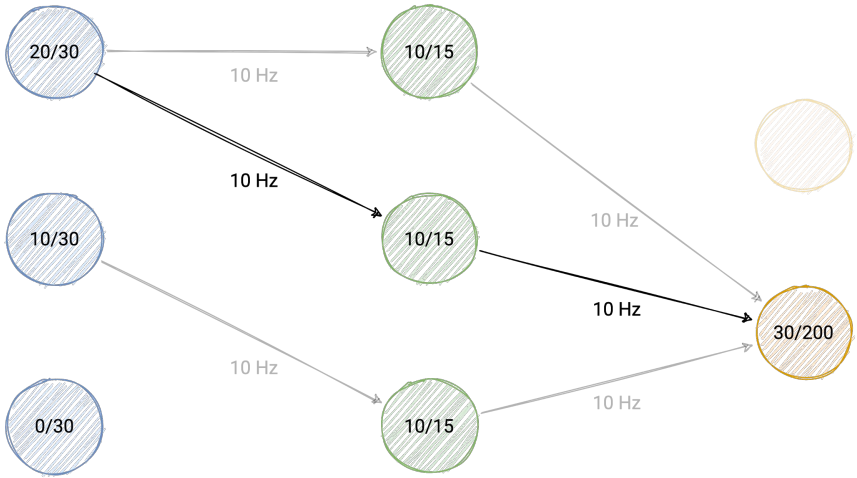
PipelineBinding	NAME	STATUS
	first-blue	Ready
	first-green	Ready
	first-yellow	Ready
	second-blue	**Pending**
	second-green	**Pending**
	second-yellow	**Pending**
	third-blue	Ready
	third-green	Ready
	third-yellow	Ready



WorkerClass	NAME	CAPACITY
	blue	30
	green	15
	yellow	200

Pipeline	NAME	LOAD	STATUS
	first	10	Ready
	second	10	Plotted
	third	10	Ready

PipelineBinding	NAME	STATUS
	first-blue	Ready
	first-green	Ready
	first-yellow	Ready
	second-blue	**Ready**
	second-green	**Ready**
	second-yellow	**Ready**
	third-blue	Ready
	third-green	Ready
	third-yellow	Ready



WorkerClass	NAME	CAPACITY
	blue	30
	green	15
	yellow	200

Pipeline	NAME	LOAD	STATUS
	first	10	Ready
	second	10	**Ready**
	third	10	Ready

PipelineBinding	NAME	STATUS
	first-blue	Ready
	first-green	Ready
	first-yellow	Ready
	second-blue	Ready
	second-green	Ready
	second-yellow	Ready
	third-blue	Ready
	third-green	Ready
	third-yellow	Ready

- Why we built it
- What we built it with
- How it works
- Lessons learned
- Overall experience

Operator does not scale horizontally

Only one instance can create/update resources at once

We have active-passive high availability

Services can’t select Pods by name

Have to use StatefulSet to have `pod-name` label

Could we implement `WorkerSet` custom resource?

Beware of race conditions

Resources of different kind will be handled concurrently

Resources of same kind will be handled sequentially

Beware of deadlocks

Pipeline can remain Pending forever if HPA does not scale

Solution: constraints on Pipeline spec
depending on which workers are used

- Why we built it
- What we built it with
- How it works
- Lessons learned
- Overall experience

Was it worth it? Absolutely.

The operator is fast and reliable

Adding new behavior is a breeze

Kubebuilder is great

Handles all the plumbing

95% of our time was spent implementing features

Significant design phase

Choice of custom resources

Finite state machine design, possible transitions, etc.

Should you build your own?

Yes, if you have specific scheduling needs

Perfect for stateful applications running in Kubernetes

Thank you

We are hiring 

PADOK XXII


Arthur Busser


arthurb@padok.fr


ArthurBusser



busser