



Kubernetes + Python = ❤

CLOUD NATIVE PRAGUE

2019-09-05

HENNING JACOBS

@try_except_



ZALANDO AT A GLANCE

~ **5.4** billion EUR

revenue 2018

> 15.000

employees in
Europe

> 79%

of visits via
mobile devices

> 250
million

visits
per
month

> 26

million
active customers

> 300.000

product choices

~ 2.000

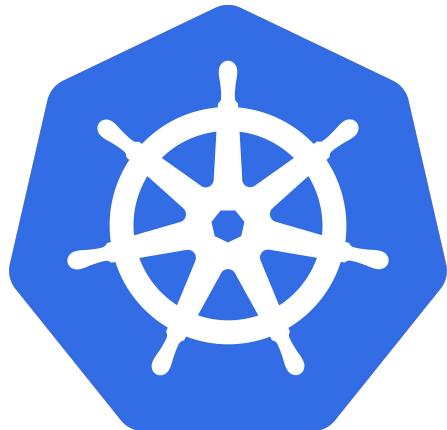
brands

17

countries

SCALE

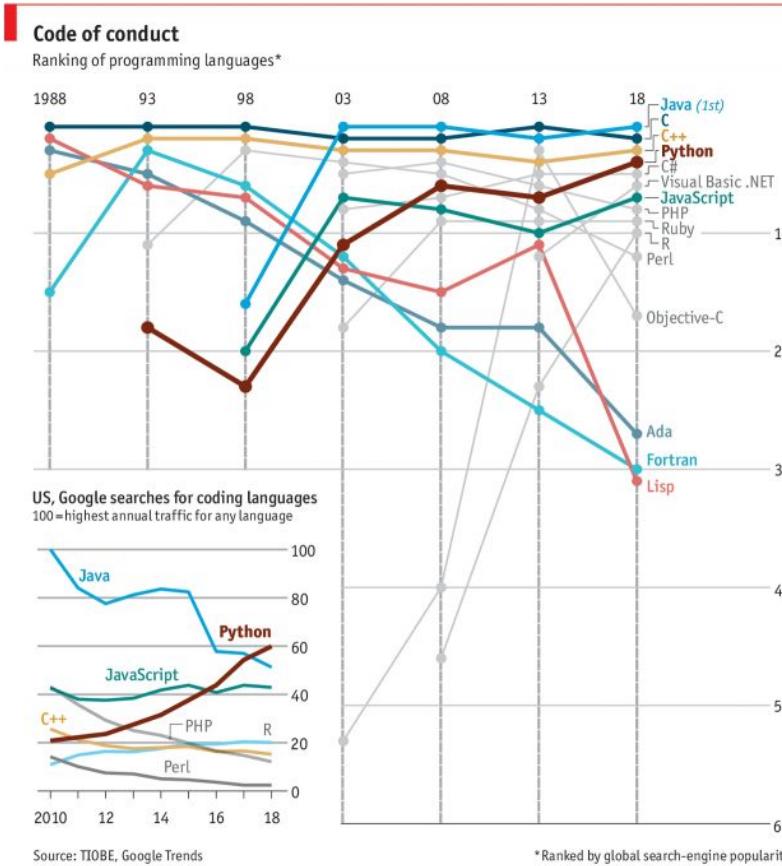
396 Accounts



140 Clusters

PYTHON

- Very popular
- Easy to learn
- Huge community



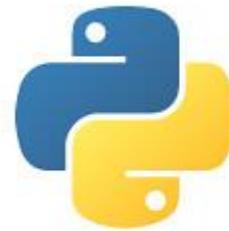
PYTHON



Michael Gasch
@embano1

Antwort an @try_except_ und @ZalandoTech

thx! asked in several sessions on preferred language
from vSphere admin perspective and it is: Python 😊
cc/ @tom_schwaller



[Tweet übersetzen](#)

2:59 nachm. · 31. Aug. 2019 · [Twitter for iPhone](#)

PYTHON IN ZALANDO



61 (Go) vs 257 (Python) profiles



14k (Go) vs 52k (Python) internal GitHub hits

Machine Learning / Data Science

patroni

A template for PostgreSQL High Availability with ZooKeeper, etcd, or Consul



kubernetes consul raft postgresql zookeeper etcd haproxy

Python MIT 298 2,703 77 (6 issues need help) 7 Updated 4 hours ago

connexion

Swagger/OpenAPI First framework for Python on top of Flask with automatic endpoint validation & OAuth2 support



python microservices web swagger openapi api-rest

flask-extensions

Python 438 2,530 212 (15 issues need help) 51 Updated 4 days ago



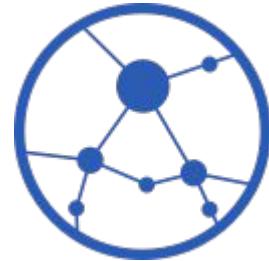
RUNNING PYTHON ON KUBERNETES

```
from aiohttp import web

async def hello(request):
    return web.Response(text="Hello, world")

async def health(request):
    return web.Response(text="OK")

app = web.Application()
app.add_routes([web.get("/", hello)])
app.add_routes([web.get("/health", health)])
# disable SIGTERM handling for disruption-free rolling updates
web.run_app(app, handle_signals=False)
```



RUNNING PYTHON ON KUBERNETES

```
FROM python:3.7-alpine  
  
RUN pip install aiohttp  
  
COPY web.py /  
  
ENTRYPOINT ["python", "web.py"]
```



⇒ ~114 MiB Docker image

DEPLOYMENT MANIFEST

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: aiohttp-helloworld
spec:
  replicas: 1
  selector:
    matchLabels:
      application: aiohttp-helloworld
  template:
    metadata:
      labels:
        application: aiohttp-helloworld
    spec:
      containers:
        - name: mycontainer
          image: hjacobs/aiohttp-helloworld:latest
          imagePullPolicy: IfNotPresent # For our E2E tests.
          readinessProbe:
            httpGet:
              path: /health
              port: 8080
          resources:
            limits:
              memory: 50Mi
            requests:
              cpu: 5m
              memory: 50Mi
          securityContext:
            readOnlyRootFilesystem: true
            runAsNonRoot: true
            runAsUser: 1000
```

Never without a
readinessProbe

Security "stuff"

HELLO, WORLD

```
kubectl apply -f deployment.yaml  
kubectl port-forward service/aiohttp-helloworld 8080:80  
http localhost:8080
```



HTTP/1.1 200 OK
Content-Length: 12
Content-Type: text/plain; charset=utf-8
Date: Thu, 05 Sep 2019 09:42:42 GMT
Server: Python/3.7 aiohttp/3.5.4

Hello, world

HOW TO USE KUBERNETES

How to draw an owl

1.



2.



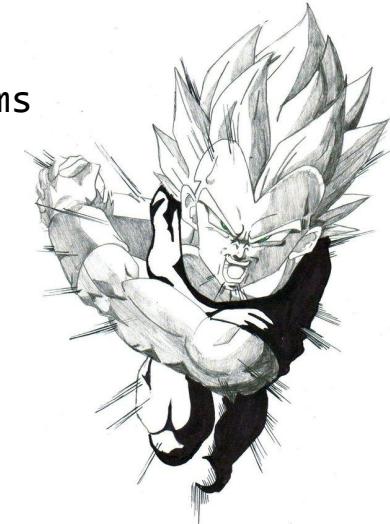
**1. run
"Hello World"**

**2. configure
the rest of your app**

IS IT FAST ENOUGH?

```
echo 'GET http://localhost:8080' | vegeta attack -rate 200 | vegeta report
```

Requests	[total, rate]	3300, 200.06
Duration	[total, attack, wait]	16.498s, 16.494, 3.261ms
Latencies	[mean, 50, 95, 99, max]	3.185ms, 3.229ms, 4.182ms, 4.791ms, 21.153ms
Bytes In	[total, mean]	39600, 12.00
Bytes Out	[total, mean]	0, 0.00
Success	[ratio]	100.00%
Status Codes	[code:count]	200:3300



local run on my laptop, single Pod in kind cluster

RUNNING PYTHON ON KUBERNETES

```
from aiohttp import web

async def hello(request):
    return web.Response(text="Hello, world")

async def health(request):
    return web.Response(text="OK")

app = web.Application()
app.add_routes([web.get("/", hello)])
app.add_routes([web.get("/health", health)])
# disable SIGTERM handling for disruption-free rolling updates
web.run_app(app, handle_signals=False)
```



"GRACEFUL" TERMINATION IN KUBERNETES

1. kubectl delete pod
2. **SIGTERM** + remove endpoints
3. grace period (default: 30s)
4. **SIGKILL**

kube-proxy/ iptables,
Load Balancer update

⇒ Going down on SIGTERM leads to **request errors**



TESTING THE ROLLING UPDATE

tests/e2e/test_update.py::test_rolling_update_no_signal_handling

Elapsed: 41.60 seconds

Successes: 2056

Errors: 0

Requests/s: 49.42 rps

tests/e2e/test_update.py::test_rolling_update_with_signal_handling

Elapsed: 15.16 seconds

Successes: 816

Errors: **2**

Requests/s: 53.95 rps



ALTERNATIVE: THE PRESTOP TRICK

```
containers:
  - name: mycontainer
    ...
    # alternative approach to ensure safe rolling updates
    # add additional sleep of 20 seconds
    # the hook will be executed before sending SIGTERM
  lifecycle:
    preStop:
      exec:
        command: ["sleep", "20"]
```

TESTING WITH PYTEST AND KIND

```
import requests

def test_web_hello_world(kind_cluster):
    kind_cluster.load_docker_image("hjacobs/aiohttp-helloworld:latest")
    kind_cluster.kubectl("apply", "-f", "deployment.yaml")

    # wait for rolling update
    kind_cluster.kubectl("rollout", "status", "deployment/aiohttp-helloworld")

    # now check whether our app returns the expected text
    with kind_cluster.port_forward("service/aiohttp-helloworld", 80) as port:
        response = requests.get(f"http://localhost:{port}/")
        response.raise_for_status()
    assert response.text == "Hello, world"
```



```
docker build -t hjacobs/aiohttp-helloworld:latest .
Sending build context to Docker daemon 94.88MB
Step 1/4 : FROM python:3.7-alpine
--> 39fb80313465
Step 2/4 : RUN pip install aiohttp
--> Using cache
--> 75e094c4447b
Step 3/4 : COPY web.py /
--> Using cache
--> dc25bceb0482
Step 4/4 : ENTRYPOINT ["python", "web.py"]
--> Using cache
--> f6160a68a628
Successfully built f6160a68a628
Successfully tagged hjacobs/aiohttp-helloworld:latest
poetry install
Installing dependencies from lock file
Nothing to install or update

poetry run pytest -r=a \
    --log-cli-level info \
    --log-cli-format '%(asctime)s %(levelname)s %(message)s' \
    --cluster-name=aiohttp-helloworld
=====
===== test session starts =====
platform linux -- Python 3.7.3, pytest-5.1.2, py-1.8.0, pluggy-0.12.0
rootdir: /home/hjacobs/workspace/kubernetes-and-python/01-running-python-on-kubernetes/aiohttp-helloworld
plugins: kind-19.9.0
collected 1 item

tests/test_web.py::test_web_hello_world
----- live log setup -----
21:10:19 INFO Creating cluster aiohttp-helloworld..
----- live log call -----
21:10:43 INFO Loading Docker image hjacobs/aiohttp-helloworld:latest in cluster (usually ~5s)..
PASSED
----- live log teardown -----
21:11:26 INFO Deleting cluster aiohttp-helloworld..

===== 1 passed in 67.97s (0:01:07) =====
```

TESTING WITH PYTEST AND KIND

pytest-kind



build passing pypi v19.9.1 python 3.7 license GPL-3.0 calver YY.MM.MICRO

Test your Python Kubernetes app/operator end-to-end with [kind](#) and [pytest](#).

`pytest-kind` is a plugin for pytest which provides the `kind_cluster` fixture. The fixture will install kind 0.5.1, create a Kubernetes 1.15 cluster, and provide convenience functionality such as port forwarding.

<https://pypi.org/project/pytest-kind/>

USING KIND CLUSTER DIRECTLY

pytest-kind

build passing pypi v19.9.1 python 3.7 license GPL-3.0 calver YY.MM.MICRO

```
from pytest_kind import KindCluster

cluster = KindCluster("myclustername")
cluster.create()
cluster.kubectl("apply", "-f", "..")
# ...
cluster.delete()
```



<https://pypi.org/project/pytest-kind/>

ACCESSING KUBERNETES FROM PYTHON



Ratnadeep Debnath

@rtnpro

Follow



"client-go isn't for mortals" - [@bryarl](#) @
keynote [#KubeConEU](#) 2019 [#KubeCon](#)
[#Kubernetes](#)



ACCESSING KUBERNETES FROM PYTHON

- Official Kubernetes client - github.com/kubernetes-client/python
- Pykube-ng - github.com/hjacobs/pykube

```
du -csh ../../lib/python3.7/site-packages/kubernetes/  
23M      total
```

```
du -csh ../../lib/python3.7/site-packages/pykube/  
160K      total
```

LISTING CONTAINER IMAGES

```
from kubernetes import client, config, watch

config.load_kube_config()
v1 = client.CoreV1Api()
ret = v1.list_namespaced_pod(args.namespace)
for pod in ret.items:
    images = [c.image for c in pod.spec.containers]
    print(pod.metadata.name, ", ".join(images))
```

```
from pykube import HTTPClient, KubeConfig, Pod

api = HTTPClient(KubeConfig.from_file())

for pod in Pod.objects(api).filter(namespace=args.namespace):
    images = [c["image"] for c in pod.obj["spec"]["containers"]]
    print(pod.name, ", ".join(images))
```

OFFICIAL CLIENT VS PYKUBE-NG

Kubernetes Python Client

[build](#) [passing](#) [pypi package](#) [10.0.1](#) [codecov](#) [unknown](#) [python 2.7 | 3.4 | 3.5 | 3.6 | 3.7](#) [Kubernetes client](#) [Silver](#) [kubernetes client](#) [beta](#)

- "Official"
- Generated (Swagger Codegen)
- Less "Pythonic"
- urllib3
- Supports many auth methods
- Heavy (23 MiB)
- Python 2.7 and 3

pykube-ng

[build](#) [passing](#) [coverage](#) [79%](#) [docs](#) [passing](#) [pypi](#) [v19.9.1](#) [python 3.6 | 3.7](#) [license](#) [apache](#) [calver](#) [YY.MM.MICRO](#)

- Kel project 2015
- Dynamic
- More "Pythonic"
- requests
- Limited auth methods
- Light (160 KiB)
- Only Python 3



INTERACTIVE CONSOLE

```
$ pip3 install pykube-ng

$ python3 -m pykube
Pykube v19.9.1, loaded "/home/hjacobs/.kube/config" with context "mycluster".
```

Example commands:

```
[d.name for d in Deployment.objects(api)]          # get names of deployments in default namespace
list(DaemonSet.objects(api, namespace='kube-system')) # list daemonsets in "kube-system"
Pod.objects(api).get_by_name('mypod').labels        # labels of pod "mypod"
```

Use Ctrl-D to exit

>>>

WRITING A SIMPLE CONTROLLER

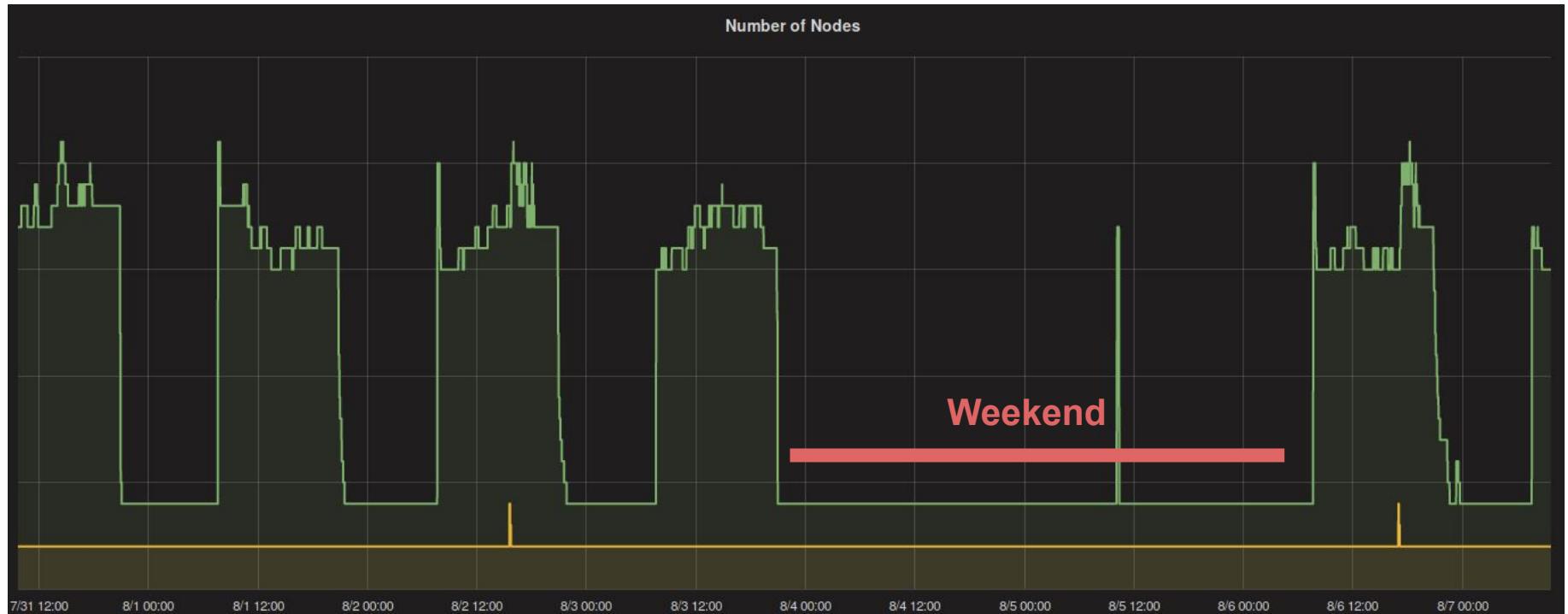
```
import datetime, os, pykube, time

while True:
    # loads in-cluster auth or local ~/.kube/config for testing
    config = pykube.KubeConfig.from_env()
    api = pykube.HTTPClient(config)

    # no timezone handling!
    weekday = datetime.datetime.today().isoweekday()
    is_weekend = weekday in (6, 7)

    for deploy in pykube.Deployment.objects(api, namespace=pykube.all):
        if "scale-down-on-weekend" in deploy.annotations and is_weekend:
            print(f"Updating deployment {deploy.namespace}/{deploy.name}..")
            deploy.replicas = 0
            deploy.update()
    time.sleep(15)
```

DOWNSCALING DURING OFF-HOURS

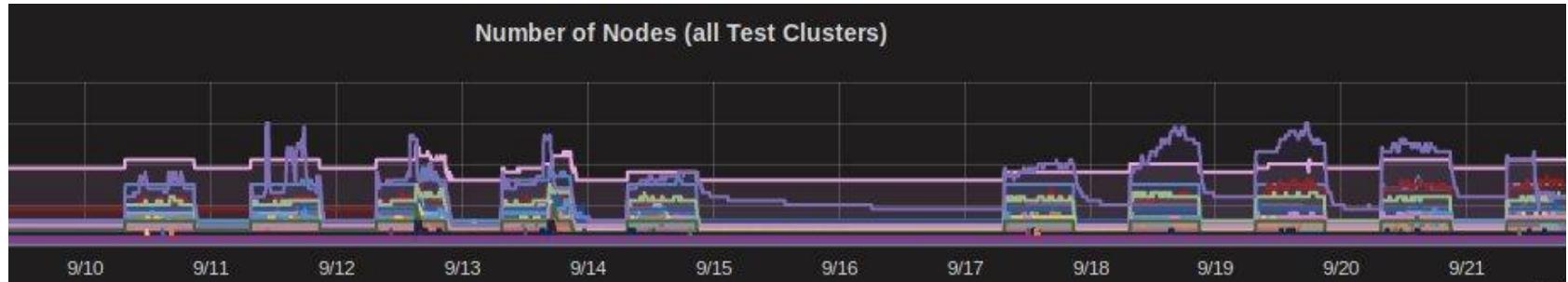


DOWNSCALING DURING OFF-HOURS

`DEFAULT_UPTIME="Mon-Fri 07:30-20:30 CET"`

annotations:

`downscaler/exclude: "true"`



HOUSEKEEPING

- Delete prototypes after X days
- Clean up temporary deployments
- Remove resources without owner



KUBERNETES JANITOR

 [hjacobs / kube-janitor](#)

 [Code](#)  [Issues 4](#)  [Pull requests 2](#)  [Insights](#)

 [Watch](#) 1  [Star](#) 55  [Fork](#) 2

Clean up (delete) Kubernetes resources after a configured TTL (time to live)

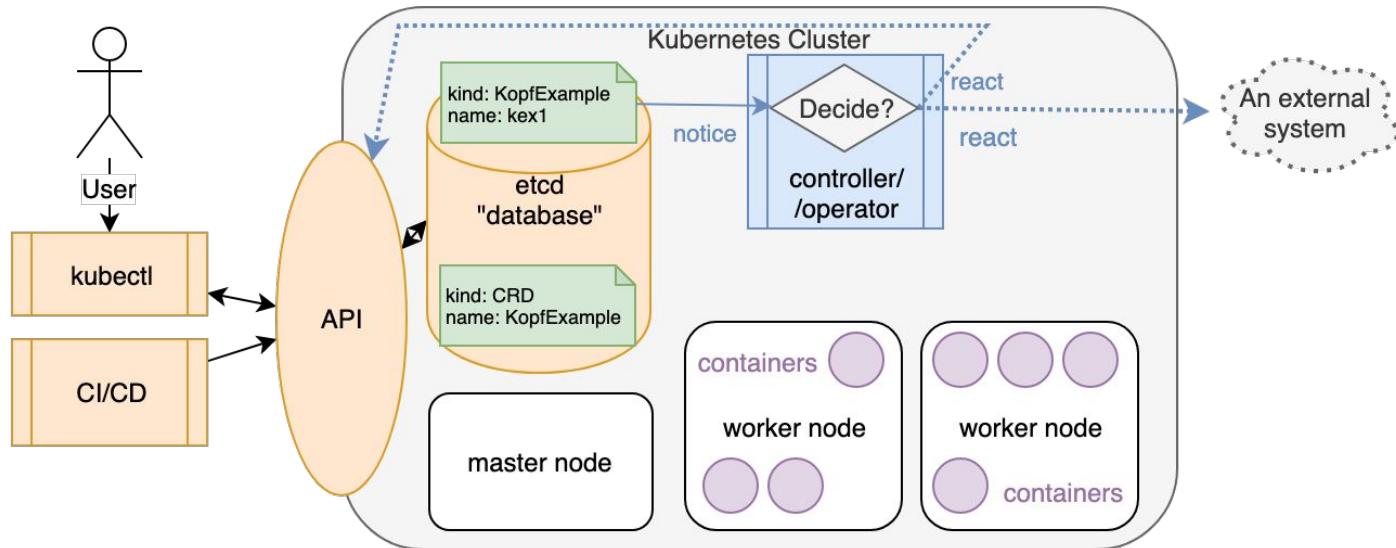
[kubernetes](#) [kubernetes-operator](#) [cleanup](#) [resource-management](#) [ttl](#) [garbage-collector](#)

- **TTL** and **expiry date** annotations, e.g.
 - set time-to-live for your test deployment
- **Custom rules**, e.g.
 - delete everything without "app" label after 7 days

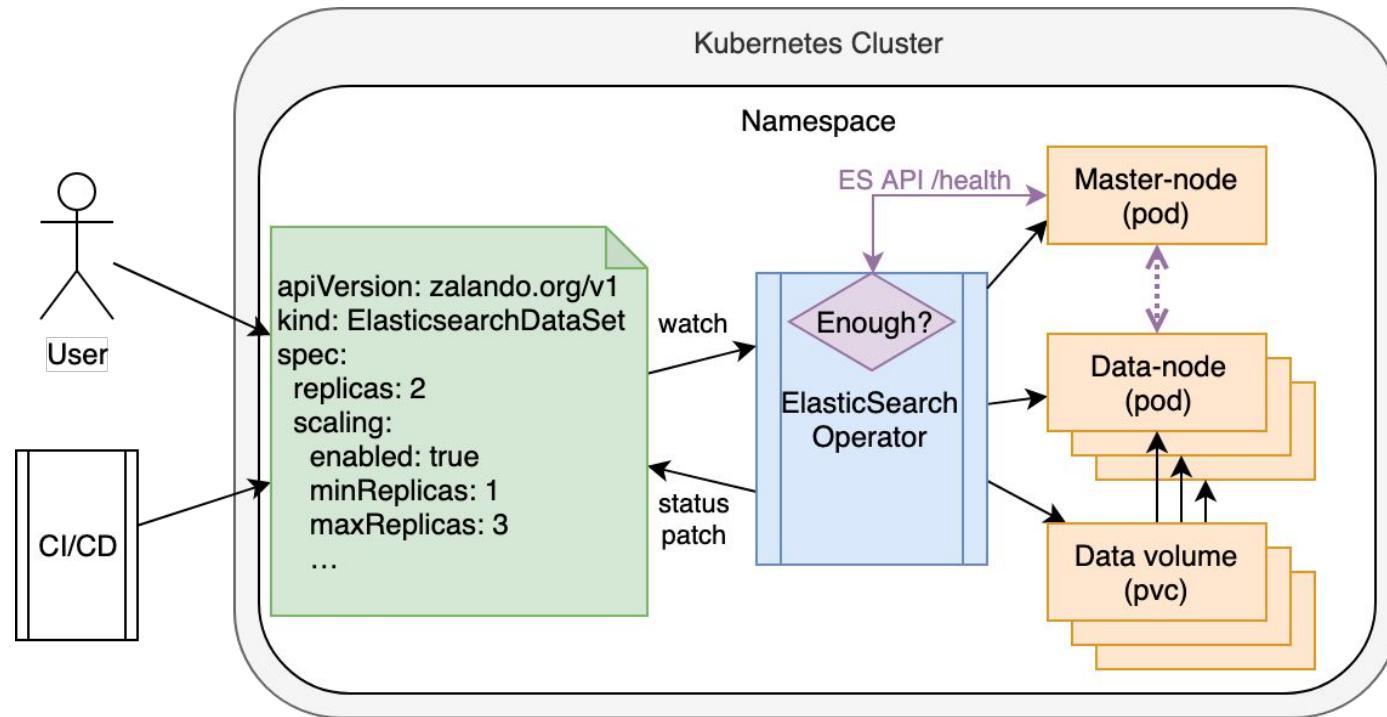
JANITOR TTL ANNOTATION

```
# let's try out nginx, but only for 1 hour
kubectl run nginx --image=nginx
kubectl annotate deploy nginx janitor/ttl=1h
```

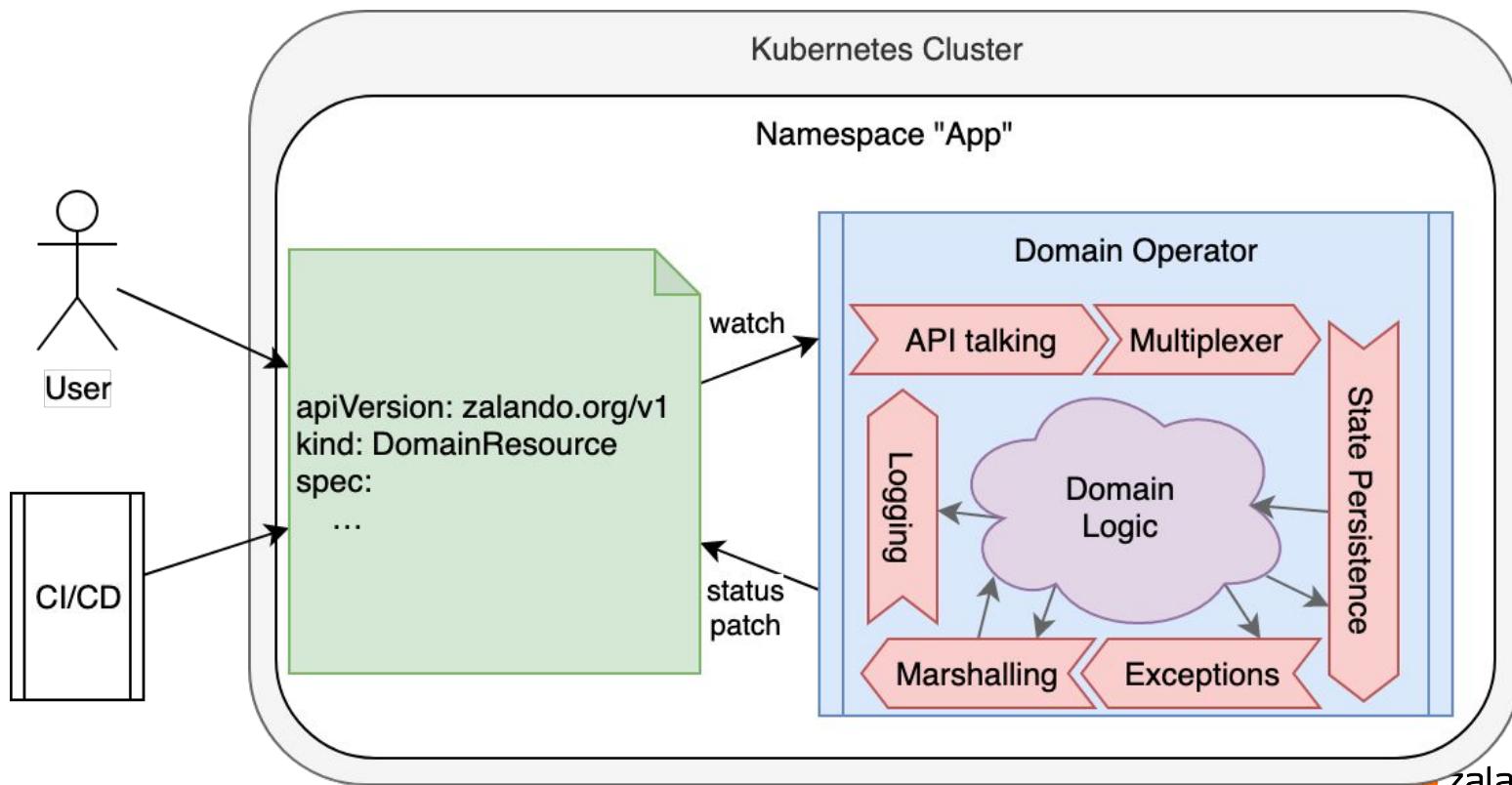
EXTENDING KUBERNETES WITH OPERATORS



ELASTICSEARCH OPERATOR



WRITING AN OPERATOR



MAKE A FRAMEWORK!

Kubernetes Operator Pythonic Framework (Kopf)



Kopf—Kubernetes Operator Pythonic Framework—is a framework and a library to make Kubernetes operators development easier, just in few lines of Python code.

github.com/zalando-incubator/kopf

KOPF: SIMPLE SPY-HANDLERS

React on events from K8s API.

Raw payload, no interpretation. Fire-and-forget, ignore errors.

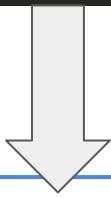
```
1 import kopf
2 ➤
3
4 @kopf.on.event('zalando.org', 'v1', 'kopfexamples')
5 def event_fn_with_error(**kwargs):
6     raise Exception("Oops!")
7
8
9 @kopf.on.event('zalando.org', 'v1', 'kopfexamples')
10 def normal_event_fn(event, **kwargs):
11     print(f"Event received: {event!r}")
12
```

KOPF: CONVENIENT CAUSE AND DIFF HANDLERS

```
1 import kopf
2
3 @kopf.on.create('zalando.org', 'v1', 'kopfexamples')
4 def create_fn_1(spec, **kwargs):
5     print(f'CREATED 1st: field={spec.field}')
6
7 @kopf.on.create('zalando.org', 'v1', 'kopfexamples')
8 def create_fn_2(meta, **kwargs):
9     print(f'CREATED 2nd: name={meta["name"]}')
10
11 @kopf.on.update('zalando.org', 'v1', 'kopfexamples')
12 def update_fn(old, new, diff, **kwargs):
13     print(f'UPDATED: diff={diff}')
14
15 @kopf.on.delete('zalando.org', 'v1', 'kopfexamples')
16 def delete_fn_1(**kwargs):
17     print('DELETED')
18
19 @kopf.on.field('zalando.org', 'v1', 'kopfexamples', field='spec.field')
20 def field_fn(old, new, **kwargs):
21     print(f'FIELD CHANGED: {old} -> {new}')
```

RUNNING FROM DEVELOPMENT ENVIRONMENT

```
1 import kopf
2
3
4 @kopf.on.create('zalando.org', 'v1', 'kopfexamples')
5 def create_fn(spec, **kwargs):
6     print(f"And here we are! Creating: {spec}")
7     return {'message': 'hello world'} # will be the new status
```



```
$ kopf run scripts.py [--verbose]
```

```
And here we are! Creating: {'duration': '1m', 'field': 'value',
'items': ['item1', 'item2']}
```

```
[2019-02-25 14:06:54,742] kopf.reactor.handlin [INFO      ]
[asf-preprocessing/kopf-example-1] Handler create_fn succeeded.
```

```
[2019-02-25 14:06:54,856] kopf.reactor.handlin [INFO      ]
[asf-preprocessing/kopf-example-1] All handlers succeeded for
creation.
```

```
1 # A demo custom resource for
2 apiVersion: zalando.org/v1
3 kind: KopfExample
4 metadata:
5   name: kopf-example-1
6   labels:
7     somelabel: somevalue
8 spec:
9   duration: 1m
10  field: value
11  items:
12    - item1
13    - item2
```

```
$ kubectl apply -f ../obj.yaml
$ kubectl describe -f ../obj.yaml
```

```
Name:          kopf-example-1
```

```
...
```

```
Status:
```

```
  create_fn:
```

```
    Message: hello world
```

```
Events:
```

Type	Reason	Age	From	Message
----	-----	---	----	-----
Normal	Success	81s	kopf	Handler create_fn succeeded.

"HELLO WORLD" EXAMPLE: CRD

```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: helloworlds.example.org
spec:
  scope: Namespaced
  group: example.org
  versions:
    - name: v1
      served: true
      storage: true
  names:
    kind: HelloWorld
    plural: helloworlds
    singular: helloworld
    shortNames:
      - hw
```

8 LINES OF PYTHON

```
import kopf
```

```
@kopf.on.create("example.org", "v1", "helloworlds")
def on_create(spec, **kwargs):
    print(f"Create handler is called with spec: {spec}")
    return {"message": f"Hello {spec['name']}!"}
```

```
@kopf.on.update("example.org", "v1", "helloworlds")
def on_update(body, **kwargs):
    print(f"Update handler is called with body: {body}")
```

4 LINES OF DOCKER

```
FROM python:3.7-alpine  
  
RUN pip install kopf  
  
COPY handlers.py /  
  
ENTRYPOINT ["kopf", "run", "handlers.py"]
```

CRD OBJECT

```
apiVersion: example.org/v1
kind: HelloWorld
metadata:
  name: test-1
spec:
  name: "Prague"
```

STATUS: "HELLO PRAGUE!"

```
apiVersion: example.org/v1
kind: HelloWorld
metadata:
  annotations:
    kopf.zalando.org/last-handled-configuration: '{"spec": {"name": "Prague"}}'
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"example.org/v1","kind":"HelloWorld","metadata":{"annotations":{},"name":"test-1","namespace":"default"}}
  creationTimestamp: 2019-09-04T19:32:59Z
  generation: 2
  name: test-1
  namespace: default
  resourceVersion: "763"
  selfLink: /apis/example.org/v1/namespaces/default/helloworlds/test-1
  uid: 2c313666-118c-4a25-8c58-fbcea6ffd686
spec:
  name: Prague
status:
  kopf: {}
  on_create:
    message: Hello Prague!
```

VMWORLD 2019 SESSION WITH KOPF

The image shows a developer's environment with two main windows. On the left is a code editor with several files open, including 'PREP.MD', 'DEMO.MD', 'crd.yaml', 'controller.py', 'example.yaml', and 'example'. The 'example.yaml' file contains YAML configuration for a Kopf operator. On the right is a 'vSphere Client' window showing a virtual machine named 'kopf-vm-template'. The vSphere interface displays details like Guest OS (VMware Photon OS (64-bit)), Compatibility (ESXi 6.7 and later (VM version 14)), and IP Address (10.160.187.155). Below the vSphere client is a table of recent tasks, all of which show a status of 'Completed'.

Task Name	Target	Status	Initiator	Queued For	Start Time	Completion Time	Server
Clone virtual machine	kopf-vm-template	✓ Completed	VSHERE.LOCAL\Administr...	19 ms	08/28/2019, 10:51:20 AM	08/28/2019, 10:51:20 AM	sc-rdops-vm06-dhcp-167...
Clone virtual machine	kopf-vm-template	✓ Completed	VSHERE.LOCAL\Administr...	11 ms	08/28/2019, 10:51:20 AM	08/28/2019, 10:51:18 AM	sc-rdops-vm06-dhcp-167...
Clone virtual machine	kopf-vm-template	✓ Completed	VSHERE.LOCAL\Administr...	undefined	08/28/2019, 10:51:12 AM	08/28/2019, 10:51:20 AM	sc-rdops-vm06-dhcp-167...
Clone virtual machine	kopf-vm-template	✓ Completed	VSHERE.LOCAL\Administr...	10 ms	08/28/2019, 10:48:36 AM	08/28/2019, 10:48:45 AM	sc-rdops-vm06-dhcp-167...

KOPF: FEATURES

- Custom & built-in resources supported (CRDs, Pods, Services, etc)
- Agnostic to API clients: kubernetes-client, pykube-ng, raw HTTP, etc
- Immediate reaction to changes and events
- Predefined behavioural patterns:
 - Simple spy-handlers for event watching
 - Advanced cause & diff detection for actual change tracking
 - Retry-until-success approach to handlers
- Operator resilience:
 - Restores its state on restarts
- Operator testing toolkit (minimally sufficient)

TESTING WITH KIND

```
from pykube.objects import NamespacedAPIObject

class HelloWorld(NamespacedAPIObject):
    version = "example.org/v1"
    kind = "HelloWorld"
    endpoint = "helloworlds"

    def test_web_hello_world(kind_cluster):
        kind_cluster.load_docker_image("kopf-example")
        kind_cluster.kubectl("apply", "-f", "deploy/")

        kind_cluster.kubectl("rollout", "status", "deployment/kopfexample-operator")

        kind_cluster.kubectl("apply", "-f", "example.yaml")

        for i in range(10):
            obj = HelloWorld.objects(kind_cluster.api).get(name="test-1")
            if "status" in obj.obj:
                break
            time.sleep(2)

        assert obj.obj["status"]["on_create"]["message"] == "Hello Prague!"
```





Kubernetes Web View

"kubectl for the web"

MOTIVATION: CHAT PING-PONG

D Oct 31, 1:48 PM
if you've done that, the next step would be to do a zkubectl describe on the pcs in the cluster, it may have some debugging info

[REDACTED] Oct 31, 1:49 PM
zkubectl describe gives me MountVolume.SetUp failed for volume '
"-201605-credentials" : secrets
"-201605-credentials" not found

D Oct 31, 1:50 PM
zkubectl describe pcs <your-platform-credential-set>

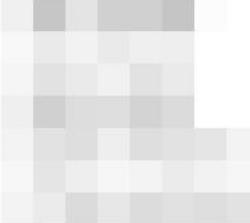
D Daniel Neuhaeuser Oct 31, 1:50 PM
not the pod

K Aug 20, 5:40 PM
I did that. Tried creating the job from cron

[REDACTED] Aug 20, 5:41 PM
it's running: zkubectl describe job -n
-manual-trigger

K Aug 20, 5:42 PM
ok. got it. thank you

MOTIVATION: PENDING PODS INCIDENT

Title of Incident:	Multiple Kubernetes test clusters have “Pending” pods, deployments are stuck
Ticket(s):	
Document Owner:	
Contributor:	
Team(s):	
Folder:	

Short summary

Due to a mistake in copying the custom Ubuntu AMI the test clusters were configured to run a PR branch of the AMI which was deregistered. As a result test clusters could not create new instances to replace existing spot instances whose lease had expired leading to almost complete unavailability of test clusters.

⇒ I want to see all "Pending" Pods across all clusters!



Henning Jacobs
@try_except_

How many incidents happen when you are on the way
to work? While the mobile UX of kube-web-view is
not perfect yet, troubleshooting problematic
#Kubernetes pods is already helpful 😊

[Tweet übersetzen](#)

The screenshot shows a mobile browser interface with a blue header bar. Below it, the URL https://... is visible, followed by a lock icon and a '5' indicating multiple tabs. The main content area displays a table of Kubernetes pods. The table has columns for Namespace, Name, Application, Component, Ready, Status, Restarts, Age, and IP. There are four entries:

Namespace	Name	Application	Component	Ready	Status	Restarts	Age	IP
default				0/1	CreateContainerConfigError	0	6d19h	10.2.12.26
default				0/1	CreateContainerConfigError	0	5h11m	10.2.16.18
default	benchmark			0/1	Pending	0	6d18h	<none>
kube-system	coredns-hm545	coredns	cluster-dns	1/3	Error	119	3h27m	172.31.5.1

SOME USE CASES

Support interactions with colleagues

- share links
- avoid human mistakes (wrong cluster login on CLI, "-n" param, ..)
- deep link to resource YAML

Incident response and investigation

- find a resource by name across clusters to help troubleshoot
- find pending pods across all clusters
- help identify if a problem is only in one cluster or multiple ones



KUBERNETES WEB UIS IN 2019

Posted: 2019-08-12 22:24 | More posts about [kubernetes](#)



This post takes a look at different open source Kubernetes web Uis, my requirements, and why I created [Kubernetes Web View](#) to help with support and troubleshooting across multiple clusters.

Use Cases

At Zalando, we have a large number of Kubernetes users (900+) and clusters (100+). There are two main use cases where I would wish for some web tool support: 1) support interactions with colleagues 2) incident response and investigation.

Support

I observed that support interactions in chat often look like:

- A: Help our service XYZ is not reachable
- B: What do you see when you do kubectl describe ingress ...?

Or similar for CRDs:

- A: I am facing an issue with identity service, ...
- B: What is the output of kubectl describe platformcredentialsset ...?

These interactions follow a pattern of using the `kubectl` command line to investigate and pinpoint a problem. This requires context switches on both sides: participants have to switch between terminal and web chat, plus they don't see the same situation.

I would wish for a Kubernetes web frontend to help with this:

- users can share links and see the same situation
- avoid human mistakes during support: login to wrong cluster on the command line, typos in CLI commands, etc
- allow crafting custom views to send to colleagues, i.e. add label columns, show multiple resource types on one page
- ideally the web tool should allow deep links into specific YAML sections (e.g. to point to a wrong spec causing the issues)

Incident Response & Investigation

Incident response for infrastructure requires gaining situational awareness, the ability to assess impact, and finding patterns across clusters. Some real-life example situations:

Search

Search Text



pipeline-id=l-2pqt9gs2ia81aef95hwcbtrdjv

Search!

Resource Types

- CronJob
 - DaemonSet
 - Deployment
 - Ingress
 - Namespace
 - Node
 - Pod
 - ReplicaSet
 - Service
 - StatefulSet
- x unselect all**

kube-web-view (Deployment)

paces/default/deployments/kube-web-view

Created: 2019-07-26 12:34:50

application: kube-web-view

deployment-id: d-293jpa847d74p32js5bxo1modc

pipeline-id: l-2pqt9gs2ia81aef95hwcbtrdjv

kube-web-view (Ingress)

paces/default/ingresses/kube-web-view

Created: 2019-07-26 12:34:51

deployment-id: d-293jpa847d74p32js5bxo1modc

pipeline-id: l-2pqt9gs2ia81aef95hwcbtrdjv

kube-web-view (Service)

paces/default/services/kube-web-view

Created: 2019-07-24 06:38:20

deployment-id: d-293jpa847d74p32js5bxo1modc

pipeline-id: l-2pqt9gs2ia81aef95hwcbtrdjv

3 results found. Searched 3 resource types in 146 clusters in 2.090 seconds.

SEARCHING FOR DOCKER IMAGE

all / search

Search

Search Text Search!

Resource Types CronJob DaemonSet Deployment Ingress Namespace Node PlatformCredentialsSet Pod ReplicaSet Service StackSet StatefulSet unselect all

etcd-operator (Deployment)
/cluster: namespaces/default/deployments/etcd-operator
Created: 2018-10-18 13:23:39 source.zalan.co **/etcd-operator:v0.9.2-master-2**
name: etcd-operator

etcd-operator (Deployment)
/cluster: namespaces/wpi/deployments/etcd-operator
Created: 2019-08-12 12:30:07 e.stups.zalan.co **/etcd-operator:v0.9.3**
application: deployment-id: d-e8yt17ub9hxty513sr27w66ea environment: staging pipeline-id: l-7bic5kvki6khadtqzq5hy3q version: master-7

etcd-operator (Deployment)
/cluster: namespaces/default/deployments/etcd-operator
Created: 2018-10-19 14:13:50 source.zalan.co **/etcd-operator:v0.9.2-master-3**
name: etcd-operator

etcd-operator (Deployment)
/clusters: namespaces/default/deployments/etcd-operator
Created: 2018-05-04 11:01:36 tups.zalan.co **/etcd-operator:v0.6.1-2**
app: etcd component: operator

etcd-operator (Deployment)
/clusters: namespaces/incentives/deployments/etcd-operator
Created: 2018-07-03 08:12:51 .zalan.co **/etcd-operator:v0.9.3**
application: deployment-id: d-so5ukevu2piyw5bdigzxc4gx3 environment: staging version: master-26

CLUSTER RESOURCES

all / all / pods

Namespaces

Nodes

PersistentVolumes

CONTROLLERS

Deployments

CronJobs

Jobs

DaemonSets

StatefulSets

POD MANAGEMENT

Ingresses

Services

Pods

ConfigMaps

META

Resource Types

Events

Pods



Label Columns

 Labels to show as columns (comma separated) or '*' to show all labels

Label Selector

 Label selector (label=value)

Filter

Status=Pending

Submit

Cluster	Namespace	Name	Ready	Status	Restarts	Age	IP	Node	Nominated Node	Readiness Gates	Created
	kube-system	3d6b56-h8bml	0/1	Pending	0	77s	<none>	<none>	<none>	<none>	2019-08-07 17:30:04
	kube-system	5798-lv6c5	0/1	Pending	0	17m	<none>	<none>	<none>	<none>	2019-08-07 17:13:39
	default	b7-8w66g	0/1	Pending	0	144m	<none>	<none>	<none>	<none>	2019-08-07 15:06:34
		676f-4x8g9	0/1	Pending	0	4h46m	<none>	<none>	<none>	<none>	2019-08-07 12:45:02
		676f-8jdvk	0/1	Pending	0	4h46m	<none>	<none>	<none>	<none>	2019-08-07 12:45:02
		676f-dmjg4	0/1	Pending	0	4h46m	<none>	<none>	<none>	<none>	2019-08-07 12:45:02
		676f-qj94v	0/1	Pending	0	4h46m	<none>	<none>	<none>	<none>	2019-08-07 12:45:02
		676f-rt4md	0/1	Pending	0	4h46m	<none>	<none>	<none>	<none>	2019-08-07 12:45:02

CRDS

/ default / postgresqls

postgresqls



additionalPrinterColumns

Name	Team	Version	Pods	Volume	CPU-Request	Memory-Request	Age	Status	Created
[REDACTED]	acid	11	1	50Gi	1000m	1Gi	81d	Running	2019-05-28 16:00:00
[REDACTED]	acid	10	3	10Gi	100m	1Gi	121d	Running	2019-04-18 15:56:36
[REDACTED]	[REDACTED]	10	3	5Gi	100m	1Gi	122d	Running	2019-04-17 14:59:28
[REDACTED]	[REDACTED]	11	3	10Gi	2000m	2Gi	37d	Running	2019-07-11 12:25:54
[REDACTED]	[REDACTED]	10	2	20Gi	300m	1Gi	103d	Running	2019-05-06 10:53:26
[REDACTED]	[REDACTED]	9.6	2	5Gi	100m	1Gi	59d	Running	2019-06-19 11:22:47
[REDACTED]	[REDACTED]	10	1	10Gi	100m	1Gi	122d	Running	2019-04-17 13:44:51
[REDACTED]	[REDACTED]	10	2	2Gi	100m	1Gi	121d	Running	2019-04-18 09:57:07
[REDACTED]	[REDACTED]	10	1	5Gi	100m	500Mi	74d	Running	2019-06-04 08:44:50
[REDACTED]	[REDACTED]	10	1	5Gi			112d	Running	2019-04-27 15:58:44
[REDACTED]	[REDACTED]	10	3	10Gi	100m	1Gi	59d	SyncFailed	2019-06-19 09:25:35
[REDACTED]	[REDACTED]	11	1	10Gi			8d	Running	2019-08-09 15:06:12

UPGRADE TO KUBERNETES 1.14

"Found 1223 rows for 1 resource type in 148 clusters in 3.301 seconds."

[all](#) / [nodes](#)

Nodes

Label Columns Labels to show as columns (comma separated) or "*" to show all labels

Label Selector Label selector (label=value)

Filter Roles=worker, Version=v1.14.6

Show CPU/Memory Usage

Cluster	Name	Status	Roles	Age	Version	Internal-IP	External-IP	OS-Image	Kernel-Version	Container-Runtime	Created
[REDACTED]	[REDACTED]	Ready	worker	4h33m	v1.14.6	[REDACTED]	[REDACTED]	Ubuntu 18.04.3 LTS	4.15.0-1045-aws	docker://18.9.7	2019-08-27 12:27:12
[REDACTED]	[REDACTED]	Ready	worker	17m	v1.14.6	[REDACTED]	[REDACTED]	Ubuntu 18.04.3 LTS	4.15.0-1045-aws	docker://18.9.7	2019-08-27 16:44:04
[REDACTED]	[REDACTED]	Ready	worker	152m	v1.14.6	[REDACTED]	[REDACTED]	Ubuntu 18.04.3 LTS	4.15.0-1045-aws	docker://18.9.7	2019-08-27 14:29:10

RESOURCE YAML VIEW

teapot / e2e-tests-zalando-kube-admission-controller-cvbcz / pods / deployment-info-test-fabc1348-b2ae-11e9-a3a6-784f4384f2b4-mj9z5

deployment-info-test-fabc1348-b2ae-11e9-a3a6-784f4384f2b4-mj9z5



Default YAML Logs

```
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    annotations:
5      kubernetes.io/psp: e2e-test-privileged-psp
6      creationTimestamp: '2019-08-15T08:40:17Z'
7      generateName: deployment-info-test-fabc1348-b2ae-11e9-a3a6-784f4384f2b4-dd77cb479-
8      labels:
9        application: e2e-test-application
10       pod-template-hash: dd77cb479
11       name: deployment-info-test-fabc1348-b2ae-11e9-a3a6-784f4384f2b4-mj9z5
12       namespace: e2e-tests-zalando-kube-admission-controller-cvbcz
13       ownerReferences:
14         - apiVersion: apps/v1
15           blockOwnerDeletion: true
```

MULTIPLE CONDITIONS

all / all / pods

Pods



Label Columns

Labels to show as columns (comma separated) or 'true' to enable column headers

Label Selector

application=coredns

Filter

Status!=Running,Status!=Completed

Submit

kubectl -l application=coredns

Cluster	Namespace	Name	Application	Component	Ready	Status	Restarts	Age	IP	Node	Nominated Node	Readiness Gates	CPU Usage	Memory Usage	Created
	kube-system	coredns-hm545	coredns	cluster-dns	2/3	CrashLoopBackOff	136	4h10m	[REDACTED]	[REDACTED]	<none>	<none>	6m	35 MiB	2019-08-20 03:51:58

Found 1 row for 1 resource type in 146 clusters in 2.358 seconds.

MULTIPLE RESOURCE TYPES

Clusters default ▾

Search Kubernetes objects. 

CLUSTER RESOURCES / default / pods,stacks,deployments,services

Namespaces

Nodes

PersistentVolumes

CONTROLLERS

StackSets

even-master-33-5db9d68c8d-5srrh even 1/1 Running 0 5d18h <none> <none> 3m 313 MiB 2019-08-29 16:35:03

Stacks

Deployments

CronJobs

Jobs

StatefulSets

POD MANAGEMENT

Ingresses

Services

Pods

ConfigMaps

CRDS

PlatformCredentialsSets

postgresqls

META

Resource Types

Events

Pods   

Name	Application	Component	Ready	Status	Restarts	Age	IP	Node	Nominated Node	Readiness Gates	CPU Usage	Memory Usage	Created
even-master-33-5db9d68c8d-5srrh	even		1/1	Running	0	5d18h			<none>	<none>	3m	313 MiB	2019-08-29 16:35:03
even-master-33-5db9d68c8d-72px8	even		1/1	Running	0	5d18h			<none>	<none>	3m	297 MiB	2019-08-29 17:05:16
even-master-33-5db9d68c8d-rhh9m	even		1/1	Running	0	5d18h			<none>	<none>	2m	312 MiB	2019-08-29 17:20:13

Stacks   

Name	Desired	Current	Up-to-date	Available	Traffic	No-Traffic-Since	Age	Created
even-master-27	3	0	0	0	0	131d	210d	2019-02-05 20:11:56
even-master-29	3	0	0	0	0	131d	140d	2019-04-17 08:30:22
even-master-30	3	0	0	0	0	131d	131d	2019-04-25 12:19:11
even-master-31	3	0	0	0	0	50d	131d	2019-04-25 12:30:11
even-master-32	3	0	0	0	0	64d	64d	2019-07-01 15:19:45
even-master-33	3	3	3	3	100		50d	2019-07-15 12:18:58

Deployments   

Name	Ready	Up-to-date	Available	Age	Containers	Images	Selector	Created
------	-------	------------	-----------	-----	------------	--------	----------	---------

kubectl get pods,stacks,deployments,...

LABEL & CUSTOM COLUMNS

labelcols=application

customcols=CPU+Requests=join(', ', spec.containers[*].resources.requests.cpu)

Pods

Namespace	Name	Application	Ready	Status	Restarts	Age	IP	Node	CPU Usage	Memory Usage	CPU Requests	Memory Requests	Created
default	kube-ops-view-7b9dd46fd8-kmv89	kube-ops-view	1/1	Running	0	13d	10.42.0.54	k3s-demo	11m	37 MiB	50m	50Mi	2019-08-22 07:53:43
default	kube-ops-view-redis-577f846477-lrgk2	kube-ops-view-redis	1/1	Running	0	39d	10.42.0.6	k3s-demo	86m	663 MiB	50m	50Mi	2019-07-27 14:43:17
default	kube-resource-report-5f77c8f5d9-lwtnl	kube-resource-report	2/2	Running	0	24d	10.42.0.44	k3s-demo	0m	36 MiB	5m, 5m	50Mi, 20Mi	2019-08-11 11:37:24
default	kube-web-view-599d9ff485-46w44	kube-web-view	1/1	Running	0	38h	10.42.0.59	k3s-demo	1m	39 MiB	5m	100Mi	2019-09-03 16:29:26
default	nginx-c9767ffdf-22tgk		1/1	Running	0	39d	10.42.0.20	k3s-demo	0m	2 MiB	2m	20M	2019-07-27 20:59:56

uses [JMESPath](#) expression

URL STRUCTURE

/clusters/{cluster}/namespaces/{namespace}/{resource-type}

cluster: cluster aliases separated by "," or "_all"

namespace: namespace name or "_all"

resource-type: plural names separated by "," or "all" (like kubectl get all)

QUERY PARAMETERS FOR LIST VIEW

sort: column name to sort by, optionally with ":desc"

labelcols: additional label columns (comma separated) or "*"

selector: label selector (like kubectl -l), will be passed to Kubernetes API

filter: post-processing filter, either a simple string, Column=Value or
Column!=Value

hidecols: hide certain columns

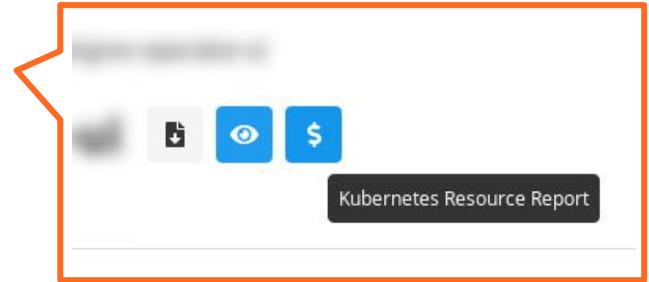
customcols: custom columns using JMESPath

limit: number of rows to show

CUSTOMIZATION

Nearly all aspects can be customized for your org via CLI options:

- External **Links** (monitoring dashboards, ..)
- Default Label and Custom **Columns**
- **Sidebar**, HTML templates
- **OAuth2** integration



LINKING FROM DEPLOYMENT STATUS

The screenshot shows a user interface for monitoring deployment status. At the top, there are three tabs: "DEPLOYMENT UNITS" (orange), "BUILD" (orange), and "QA-210" (orange). Below these are two main sections: "GENERAL" and "DEPLOYMENT STATUS". The "DEPLOYMENT STATUS" section has a yellow circular badge with the number "1", indicating one pending update. A green vertical bar on the left indicates the status of various components.

- Deployment**: Status: **qa**. A warning message: "Warning: container -qa: Memory limit and request do not match. You must use the same value to avoid any disruption to your application." with a "Docs" link.
- ReplicaSet**: Status: **-qa-8664b77c89**
- Pod**: Status: **-qa-8664b77c89-r8x4r**. Sub-links: "kube-web-view" and "scalyr logs".
- Ingress**: Status: **!-qa**
- Service**: Status: **-qa**

KUBE-WEB-VIEW.DEMO.J-SERV.DE

local / default / pods

Pods



Name	Application	Ready	Status	Restarts	Age
kube-ops-view-7b9dd46fd8-vfqgt	kube-ops-view	1/1	Running	0	46h
kube-ops-view-redis-577f846477-lrgk2	kube-ops-view-redis	1/1	Running	0	15d
kube-resource-report-5f77c8f5d9-lwtlnl	kube-resource-report	2/2	Running	0	8h
kube-web-view-75b996b8c8-z2sgt	kube-web-view	1/1	Running	0	57s
nginx-c9767ffdf-22tgk		1/1	Running	0	14d
nginx-c9767ffdf-8qzqr		1/1	Running	0	14d

TESTING WITH PYTEST-KIND AND PURE HTML

```
def test_filter_pods_with_custom_columns(session):
    """
    Test that filtering on custom columns works
    """

    response = session.get(
        "/clusters/local/namespaces/default/pods?customcols=Images=spec.containers[*].image&filter=hjacobs/"
    )
    response.raise_for_status()
    check_links(response, session)
    ths = response.html.find("main table thead th")
    # note: pods have an extra "Links" column (--object-links)
    assert ths[-3].text == "Images"

    rows = response.html.find("main table tbody tr")
    for row in rows:
        cells = row.find("td")
        assert cells[-3].text.startswith("[ 'hjacobs/")

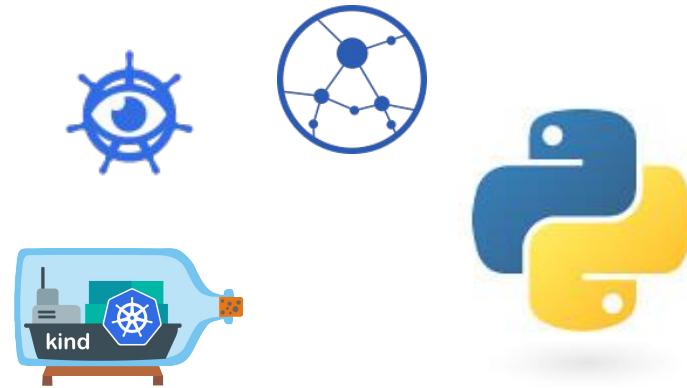


```



KUBERNETES + PYTHON?

- Pykube
- asyncio / aiohttp
- pytest-kind
- Jinja2 templates



codeberg.org/hjacobs/kube-web-view/

KUBERNETES RESOURCE REPORT

Overview Clusters Ingresses Teams Applications Pods

Cluster [REDACTED]
https://[REDACTED]

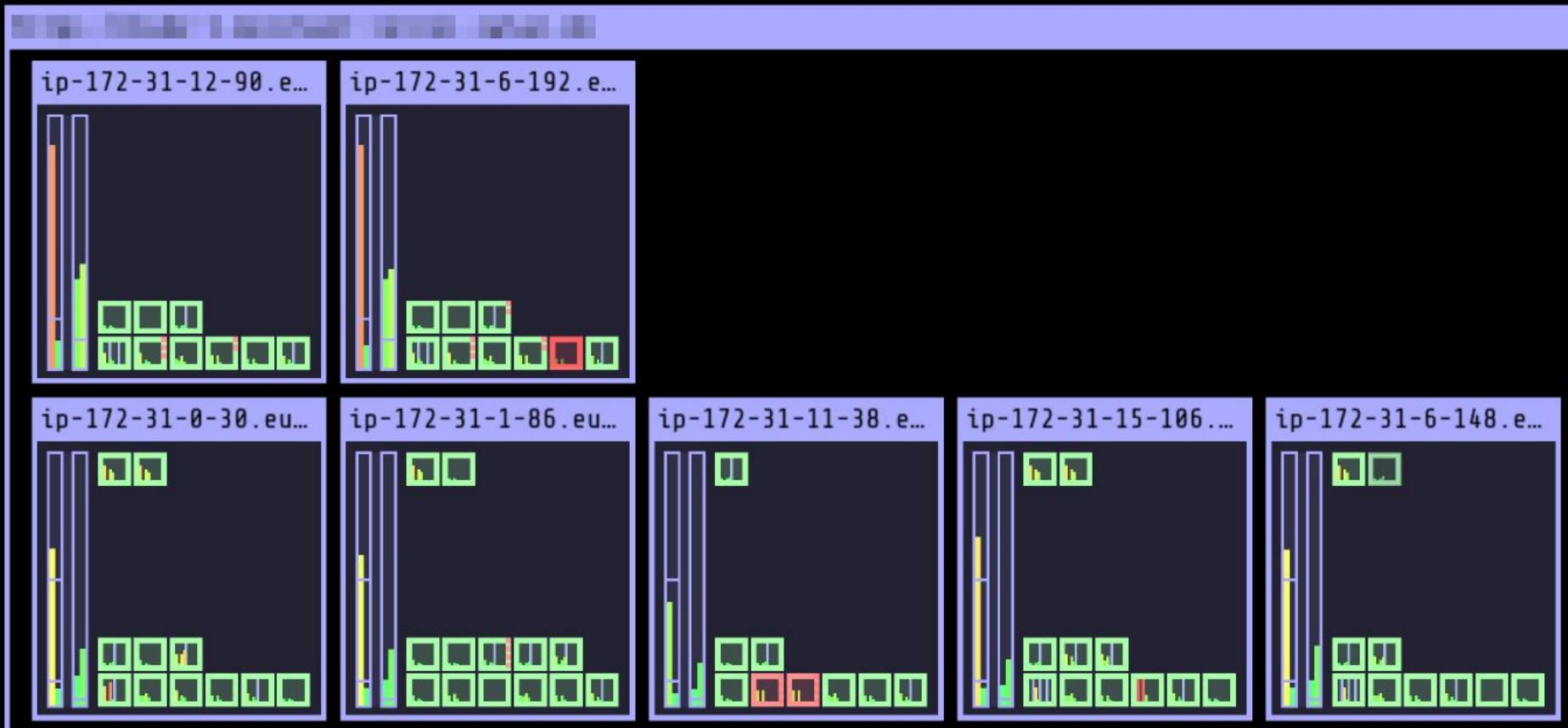
MASTER NODES	WORKER NODES	PODS	CPU REQUESTS / ALLOCATABLE	MEMORY REQUESTS / ALLOCATABLE	MONTHLY COST
2	15	325	55.7 / 60.6	183.1 GiB / 241.5 GiB	1,687.16 USD

You can potentially save [REDACTED] every month by optimizing resource requests and reducing slack.

Price per requested vCPU is [REDACTED] per hour and per requested GiB memory is [REDACTED] per hour.

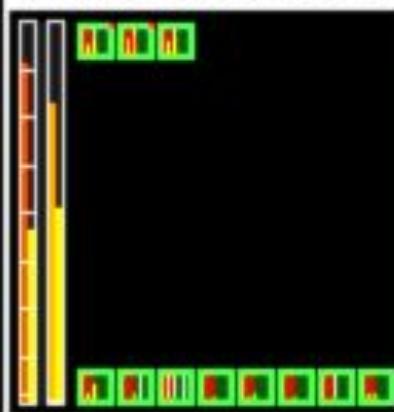
Nodes

Name	Role	Instance Type	S?	Version	CC	MC	CPU	Memory (GiB)	Cost
[REDACTED]	worker	m4.xlarge	✗	v1.10.5	4	15.7 GiB	0.7 3.7 3.8 3.7 10.8 15.1	70.13	
[REDACTED]	worker	m4.xlarge	✗	v1.10.5	4	15.7 GiB	1.0 3.4 3.8 7.2 14.1 15.1	70.13	
[REDACTED]	worker	m4.xlarge	✗	v1.10.5	4	15.7 GiB	0.2 3.8 3.8 3.4 8.0 15.1	70.13	
[REDACTED]	master	m4.large		v1.10.5	2	7.8 GiB	0.3 1.0 1.8 2.8 1.3 7.3	87.66	
[REDACTED]	worker	m4.xlarge	✗	v1.10.5	4	15.7 GiB	0.2 2.7 3.8 3.3 9.4 15.1	70.13	



<https://github.com/hjacobs/kube-ops-view>

ip-172-31-8-228.eu-cen..



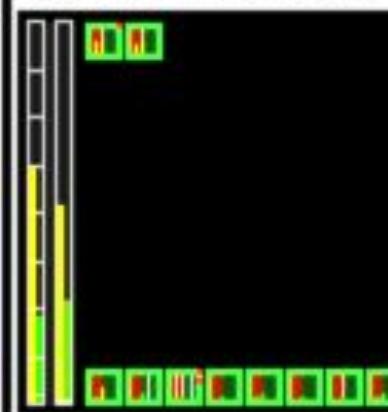
ip-172-31-8-225.eu-cen..



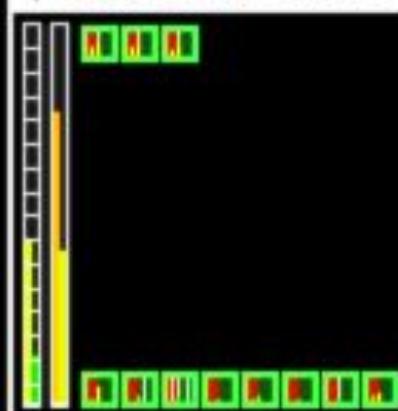
ip-172-31-8-23.eu-cent..



ip-172-31-1-85.eu-cent..



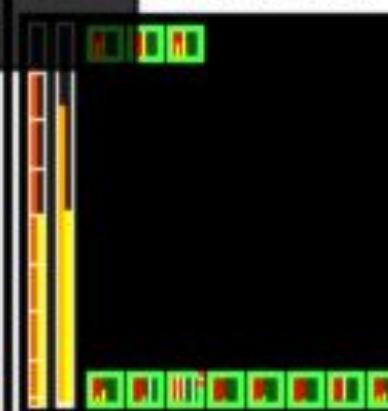
ip-172-31-15-3.eu-cent..



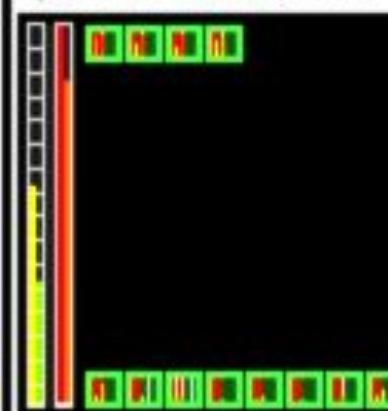
ip-1-16-24.eu-cen..



ip-172-31-16-24.eu-cen..



ip-172-31-16-254.eu-ce..



KUBERNETES + PYTHON = ❤

github.com/hjacobs

kube-resource-report

Report Kubernetes cluster and pod resource requests vs usage and generate static HTML

kubernetes kubernetes-resources kubernetes-cluster

HTML ★ 300 35 GNU General Public License v3.0 Updated 1 minute ago

kube-ops-view

Kubernetes Operational View - read-only system dashboard for multiple K8s clusters

kubernetes kubernetes-monitoring admin-dashboard

JavaScript ★ 1,113 150 GNU General Public License v3.0 3 issues need help Updated 2 minutes ago

pykube

Forked from k8sproject/pykube

Lightweight Python 3.6+ client library for Kubernetes (pykube-ng)

kubernetes python3

Python ★ 45 197 Apache License 2.0 Updated 3 hours ago



kube-janitor

Clean up (delete) Kubernetes resources after a configured TTL (time to live)

kubernetes kubernetes-operator cleanup resource-management garbage-collector

Python ★ 140 12 GNU General Public License v3.0 Updated 3 hours ago

kube-downscaler

Scale down Kubernetes deployments after work hours

kubernetes scaling

Python ★ 239 35 GNU General Public License v3.0 Updated 7 hours ago

pytest-kind

Mirror of pytest-kind: Test your Python Kubernetes app/operator end-to-end with kind and pytest

kubernetes pytest kind end-to-end-testing python3

Python ★ 3 35 GNU General Public License v3.0 Updated 23 hours ago

kube-web-view

Mirror of Kubernetes Web View, allows to list and view all Kubernetes resources (incl. CRDs) with permalink-friendly URLs in a plain-HTML frontend

kubernetes kubernetes-web cluster

Python ★ 37 5 GNU General Public License v3.0 Updated yesterday



KUBERNETES + PYTHON = ❤

code examples from this talk can be found in
codeberg.org/hjacobs/kubernetes-and-python

ZALANDO OPEN SOURCE

Kubernetes on AWS

github.com/zalando-incubator/kubernetes-on-aws

AWS ALB Ingress controller

github.com/zalando-incubator/kube-ingress-aws-controller

External DNS

github.com/kubernetes-incubator/external-dns

Postgres Operator

github.com/zalando/postgres-operator

Elasticsearch Operator

github.com/zalando-incubator/es-operator

Kubernetes Metrics Adapter

github.com/zalando-incubator/kube-metrics-adapter





QUESTIONS?

HENNING JACOBS

henning@zalando.de

[@try_except](https://twitter.com/try_except)

Illustrations by [@01k](https://twitter.com/01k)

