Emitting, Consuming, and Presenting

The Event Lifecycle

Jesse Dearing

- SRE @ VMware
- Love working with systems, infrastructure, and tooling to manage it all
- Kubernetes since 1.6

VMware Tanzu

Why events?

Let's talk events because...

- Found the API for writing events
- I like the idea of streaming and evented constructs
- Events, the DNA of Kubernetes by Michael Gasch

What are events?

Get Events with Kubectl

```
k describe pod contour-65576967f6-n49ck

# ..snip..
Events:
  Type     Reason     Age                 From
    Message
  ----     ------     ----                ----
    -------
  Warning  Unhealthy  60s (x9 over 84s)  kubelet, ip-10-37-3-137.us-
west-2.compute.internal  Readiness probe failed: HTTP probe failed with statuscode: 503
```

Again, this time with YAML

k get events -o yaml

```
  apiVersion: v1
  count: 9
  eventTime: null
  firstTimestamp: "2019-11-12T07:12:35Z"
  involvedObject:
    apiVersion: v1
    fieldPath: spec.containers{envoy}
    kind: Pod
    name: contour-65576967f6-n49ck
    namespace: contour-internal
    resourceVersion: "16061114"
    uid: d17b6473-68dc-42e5-8e40-0c412da00489
  kind: Event
  lastTimestamp: "2019-11-12T07:12:59Z"
# (cont'd)
```

Again, this time with YAML (continued)

```
k -n tmc-dearingj get events -o yaml # continued

  message: 'Readiness probe failed: HTTP probe failed with statuscode: 503'
  metadata:
    creationTimestamp: "2019-11-12T07:12:35Z"
    name: contour-65576967f6-n49ck.15d658cc824e55e0
    namespace: contour-internal
    resourceVersion: "16097623"
    selfLink: /api/v1/namespaces/contour-internal/events/contour-65576967f6-n49ck.
15d658cc824e55e0
    uid: 4adf1ced-3161-4e4d-82fa-2fc24bf96d92
  reason: Unhealthy
  reportingComponent: ""
  reportingInstance: ""
  source:
    component: kubelet
    host: ip-10-37-3-137.us-west-2.compute.internal
  type: Warning
```

Event Lifecycle Defaults

- Name: <object name>.<unique id or hash>
- TTL: 1 hour default (--event-ttl)
- Events live in namespaces

What can we do with these events?

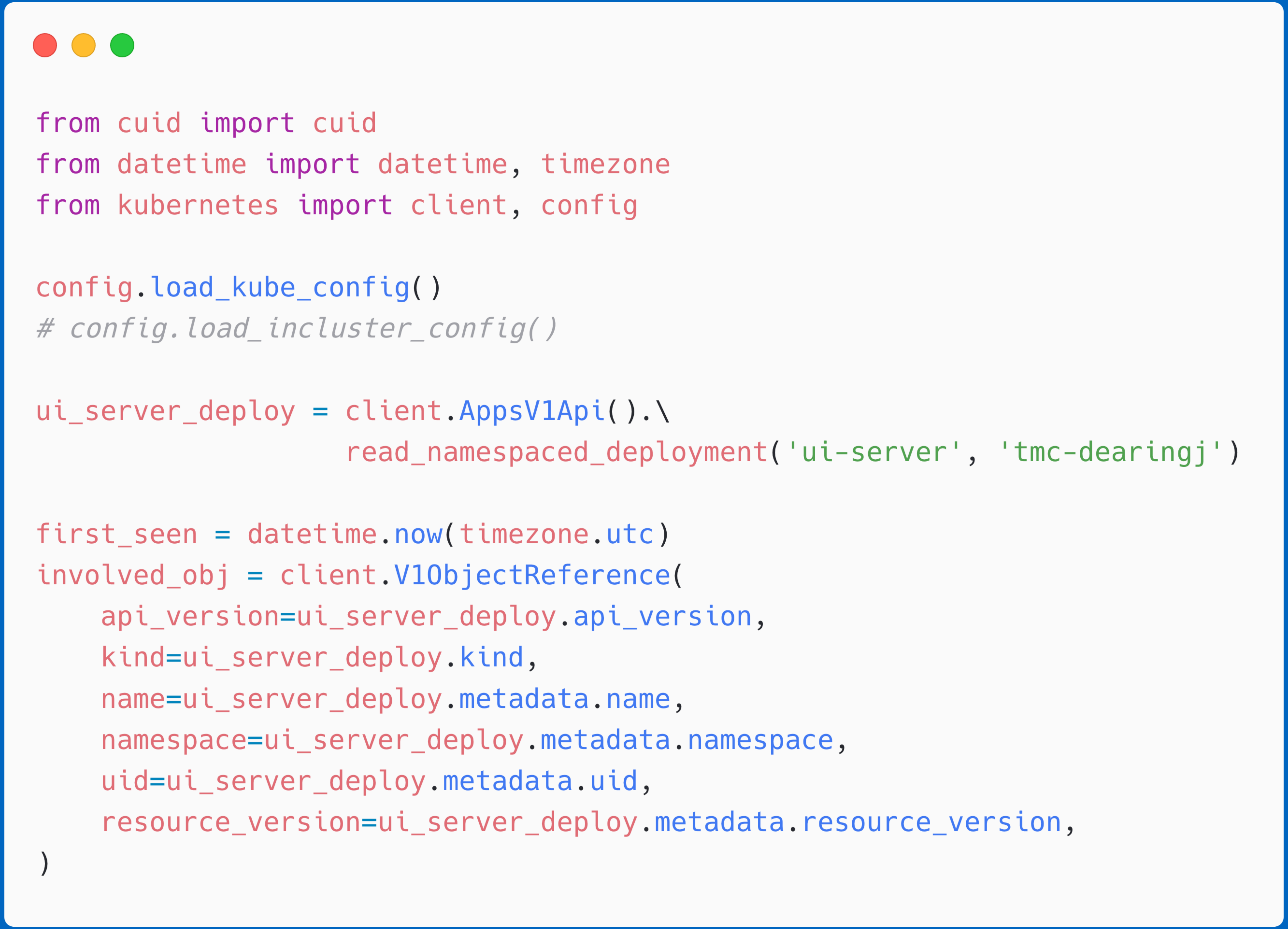
Emitting Events

Building the Event Object

- Requires a “involved object”
- Events are a namespaced object

Emitting Events in Python

- Python Kubernetes client
- Use CUID to create a simple uid
- Lookup a deployment in the namespace
- Emit event based on deployment

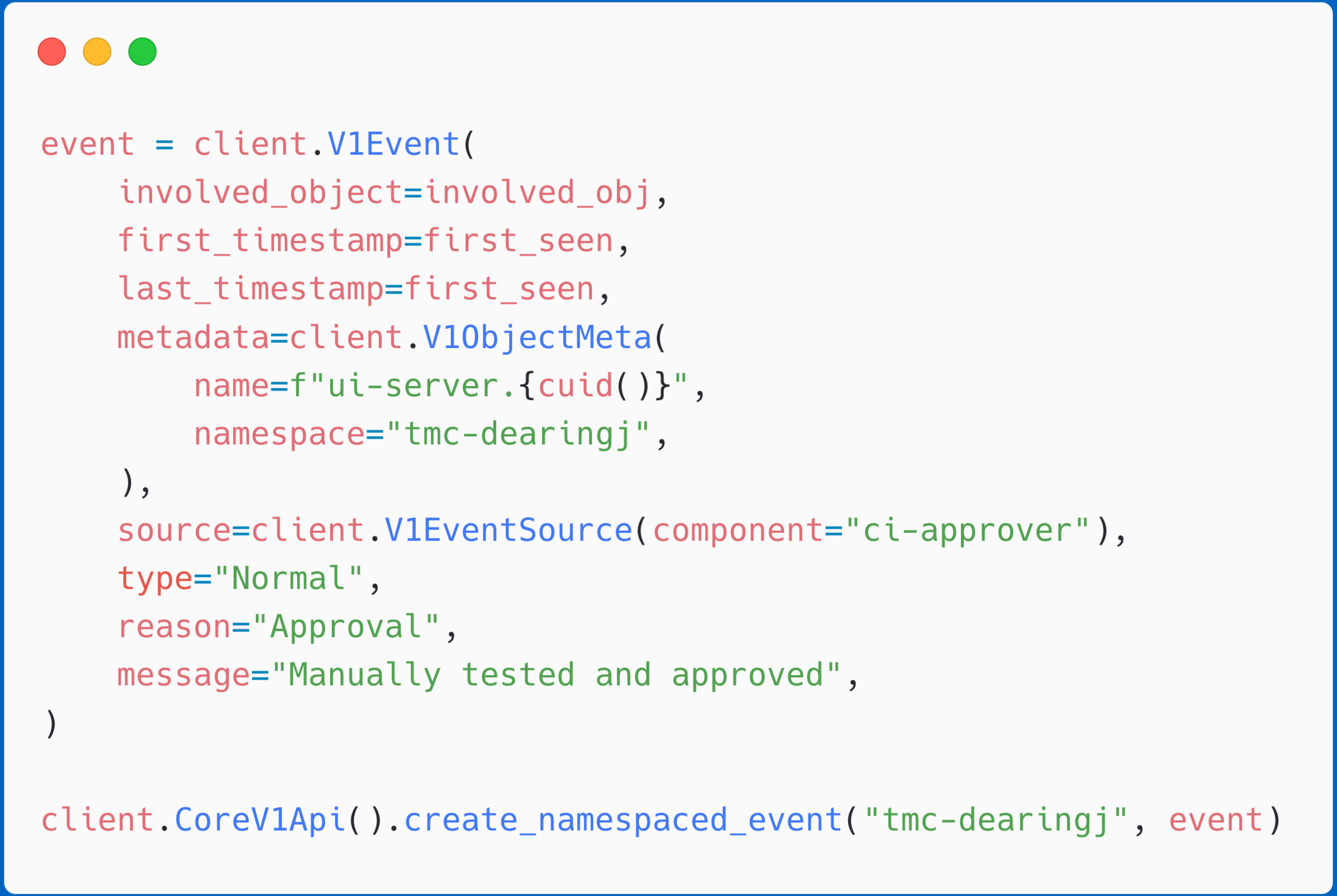


```python
from cuid import cuid
from datetime import datetime, timezone
from kubernetes import client, config

config.load_kube_config()
# config.load_incluster_config()

ui_server_deploy = client.AppsV1Api().\
                   read_namespaced_deployment('ui-server', 'tmc-dearingj')

first_seen = datetime.now(timezone.utc)
involved_obj = client.V1ObjectReference(
    api_version=ui_server_deploy.api_version,
    kind=ui_server_deploy.kind,
    name=ui_server_deploy.metadata.name,
    namespace=ui_server_deploy.metadata.namespace,
    uid=ui_server_deploy.metadata.uid,
    resource_version=ui_server_deploy.metadata.resource_version,
)
```



```python
event = client.V1Event(
    involved_object=involved_obj,
    first_timestamp=first_seen,
    last_timestamp=first_seen,
    metadata=client.V1ObjectMeta(
        name=f"ui-server.{cuid()}",
        namespace="tmc-dearingj",
    ),
    source=client.V1EventSource(component="ci-approver"),
    type="Normal",
    reason="Approval",
    message="Manually tested and approved",
)

client.CoreV1Api().create_namespaced_event("tmc-dearingj", event)
```

View Our Event in a Sea of Events

```
k -n tmc-dearingj get events
LAST SEEN   TYPE     REASON     OBJECT                     MESSAGE
5s          Normal   Approval   deployment/ui-server       Manually tested and approved
```

Our Event in Context of a Deployment

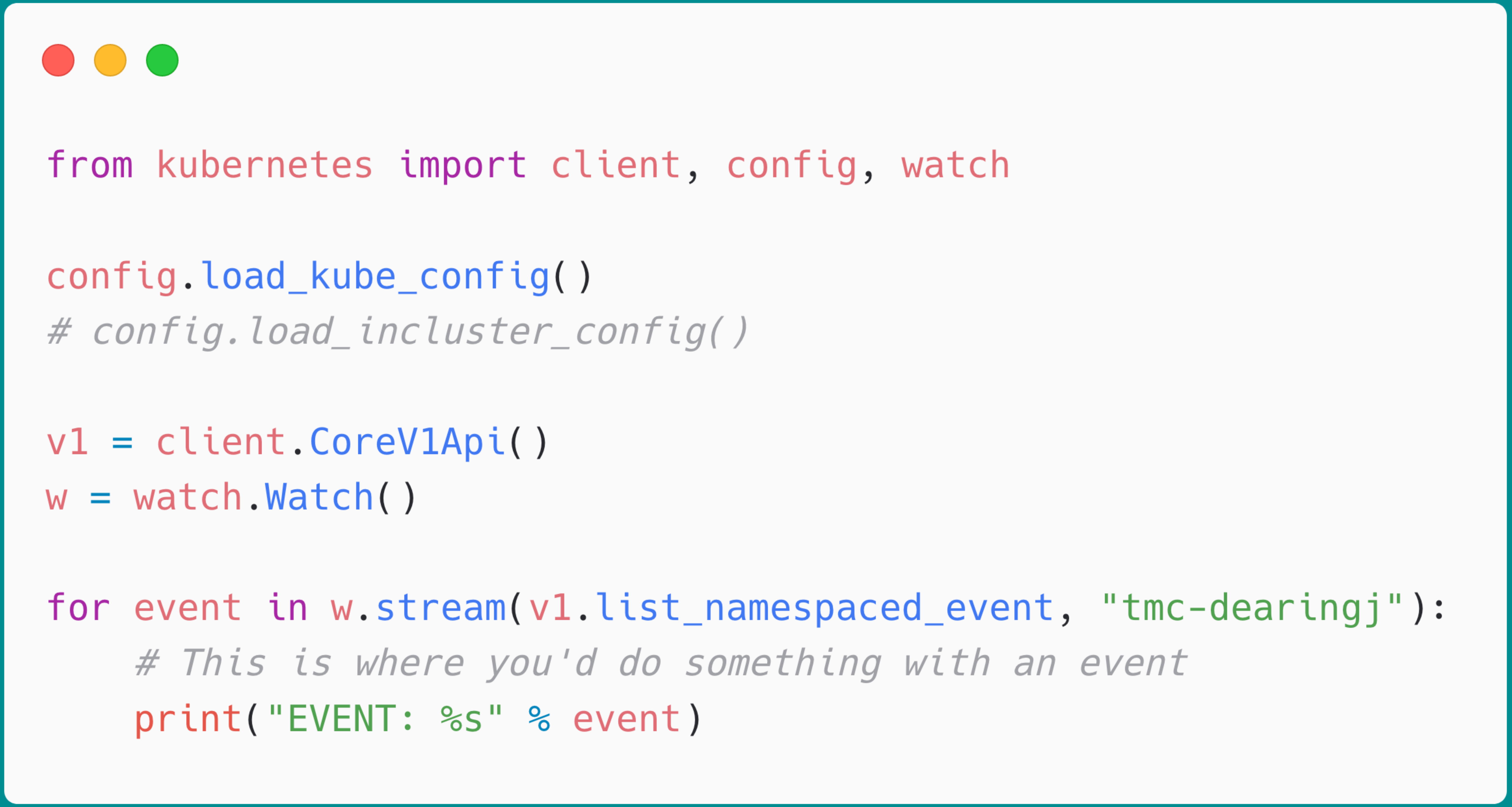
```
k describe deployment ui-server
# ..snip..
OldReplicaSets:  <none>
NewReplicaSet:   ui-server-658d6ccf66 (1/1 replicas created)
Events:
  Type    Reason    Age   From          Message
  ----    -------   ----  ----          -------
  Normal  Approval  59s   ci-approver  Manually tested and approved
```

Consuming Events

Make your own ~~controller~~

Operator

- Where do you get started?
- Let’s look at one in a few lines of Python



```python
from kubernetes import client, config, watch

config.load_kube_config()
# config.load_incluster_config()

v1 = client.CoreV1Api()
w = watch.Watch()

for event in w.stream(v1.list_namespaced_event, "tmc-dearingj"):
    # This is where you'd do something with an event
    print("EVENT: %s" % event)
```

Heptio Labs Event Router

- https://github.com/heptiolabs/eventrouter
- Many sinks
  - STDOUT for logging into Elasticsearch via Fluentd
- Prometheus Metrics

Presenting Events

Monitoring Events

- Can use Prometheus exporters
  - Built into heptiolabs/eventrouter
  - caicloud/event_exporter
- https://sched.co/UaYM - Exporting Event Objects for Better Observability
- Monitoring providers
  - Google Cloud - Stackdriver
  - Datadog - Agent
  - Logs via (eventrouter → fluentd)
  - Quite a few more…

What Events Tell Us Now

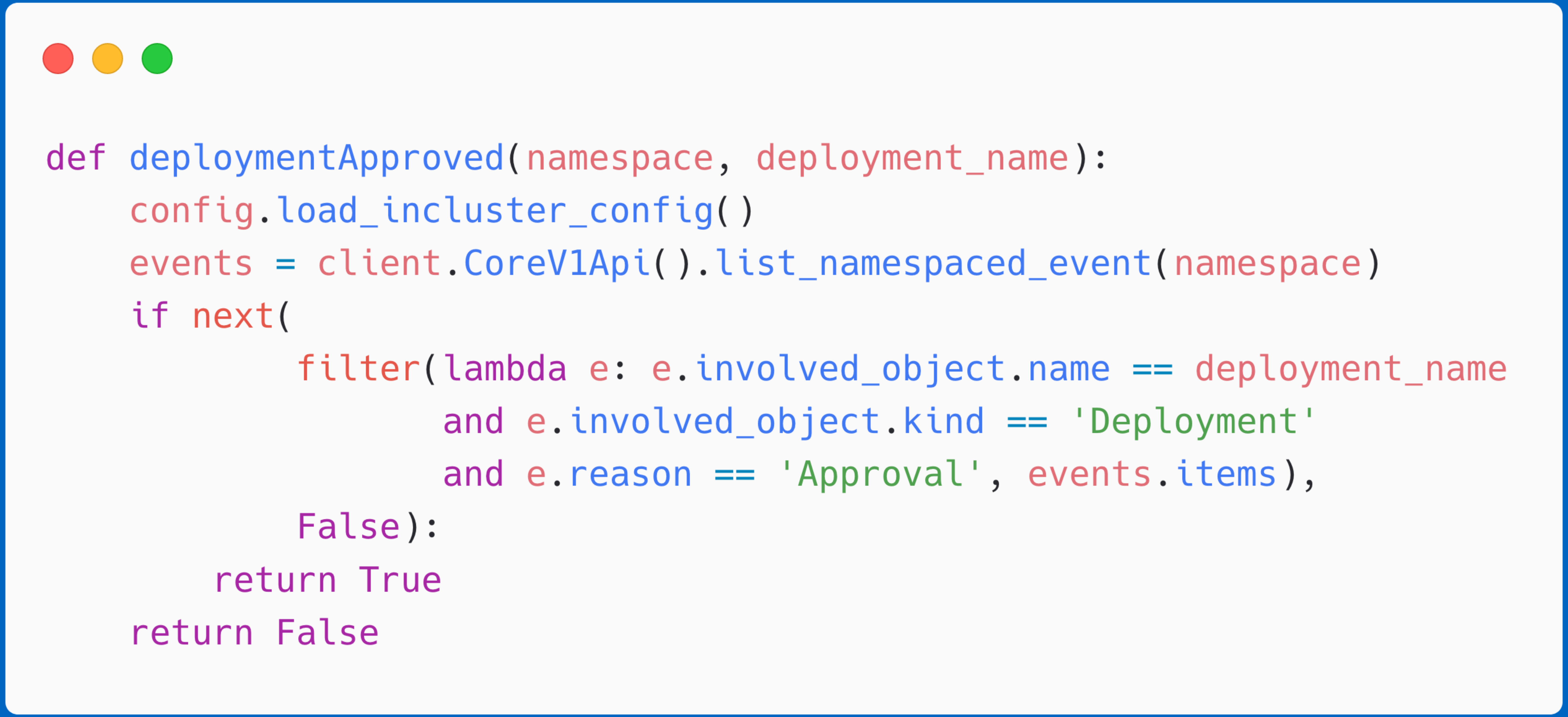
- Liveness/Readiness Probe Failures
- Back-off
- Job creation/completion
- Scheduling

Use Cases for Custom Events

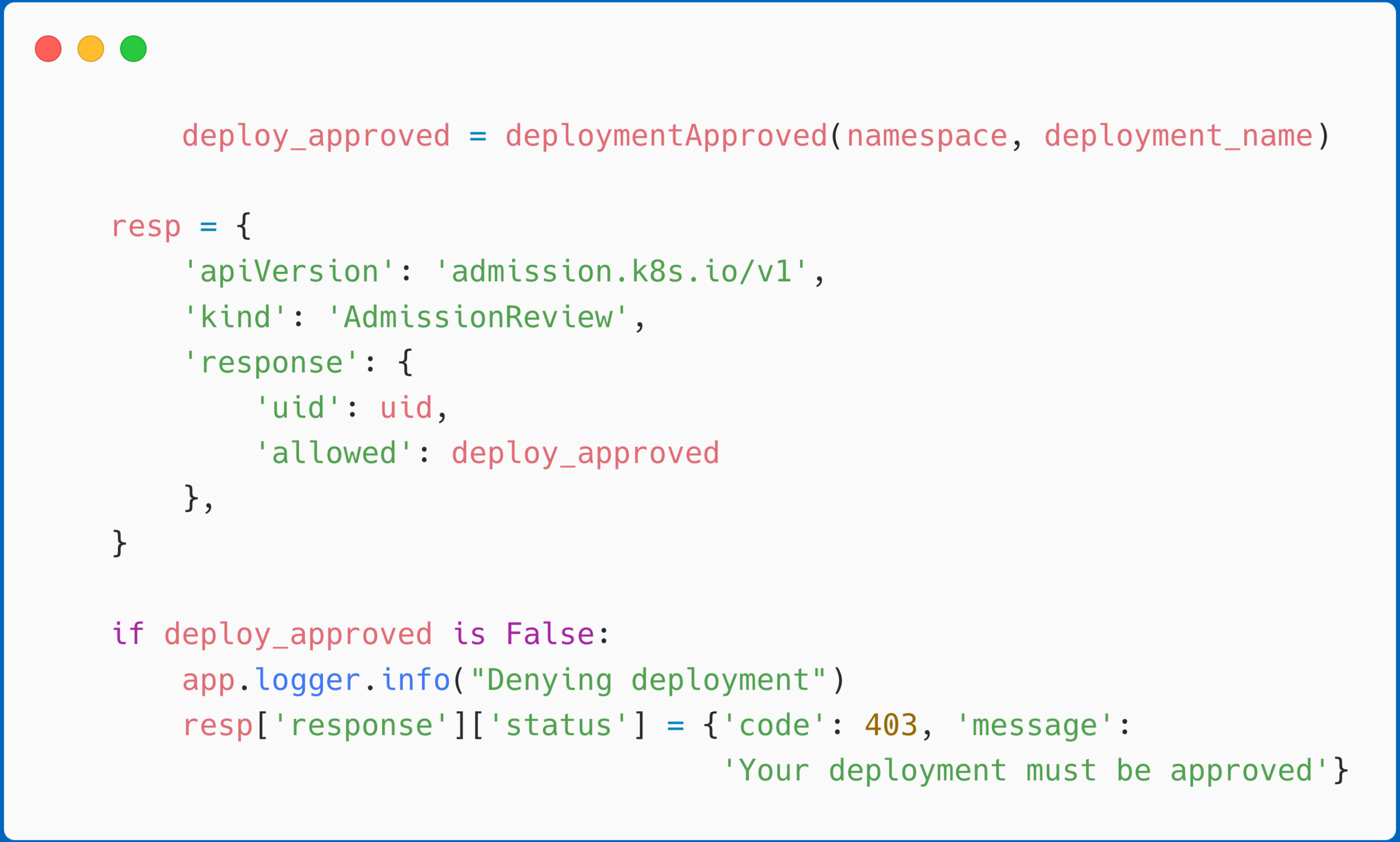
- Automated deployments for environments needing approvals
- Use an event to approve a deploy
- Use an admission webhook to enforce

Webhook

- Listening for operations on replica set resources that checks events on deployments



```python
def deploymentApproved(namespace, deployment_name):
    config.load_incluster_config()
    events = client.CoreV1Api().list_namespaced_event(namespace)
    if next(
            filter(lambda e: e.involved_object.name == deployment_name
                   and e.involved_object.kind == 'Deployment'
                   and e.reason == 'Approval', events.items),
            False):
        return True
    return False
```



```python
    deploy_approved = deploymentApproved(namespace, deployment_name)

resp = {
    'apiVersion': 'admission.k8s.io/v1',
    'kind': 'AdmissionReview',
    'response': {
        'uid': uid,
        'allowed': deploy_approved
    },
}

if deploy_approved is False:
    app.logger.info("Denying deployment")
    resp['response']['status'] = {'code': 403, 'message':
                                  'Your deployment must be approved'}
```

Webhook: without approval

```
k -n tmc-dearingj describe deployment.app ui-server
Events:
  Type     Reason                 Age                 From                   Message
  ----     -------                ----                ----                   -------
  Warning  ReplicaSetCreateError  1s (x12 over 12s)  deployment-controller  Failed
to create new replica set "ui-server-59f6b8d947": admission webhook "event-
webhook.jesse.dev" denied the request: Your deployment must be approved
```

Webhook: with approval

```
k -n tmc-dearingj describe deployment.app ui-server

Events:
  Type     Reason                 Age                       From                   Message
  ----     ------                 ----                      ----                   -------


  Normal   Approval               33s                       ci-approver
Manually tested and approved
  Normal   ScalingReplicaSet      2s                        deployment-controller  Scaled
up replica set ui-server-59f6b8d947 to 1
  Normal   ScalingReplicaSet      2s                        deployment-controller  Scaled
down replica set ui-server-59f6b8d947 to 0
  Normal   ScalingReplicaSet      2s                        deployment-controller  Scaled
up replica set ui-server-d57549564 to 1
```

Other use cases

- Relating external state to Kubernetes objects
  - Emit an event related to Ingress on site ping failures
- Logging operator changes to cluster objects

Events

- TTL’d
- Need to refer to another object
- Show up in describe
- Easy to consume via watches
- Monitor and log with already available tools

Thank you!

@JesseDearing
jesse.dev

Source

https://github.com/jessedearing/events-kubecon-na-2019