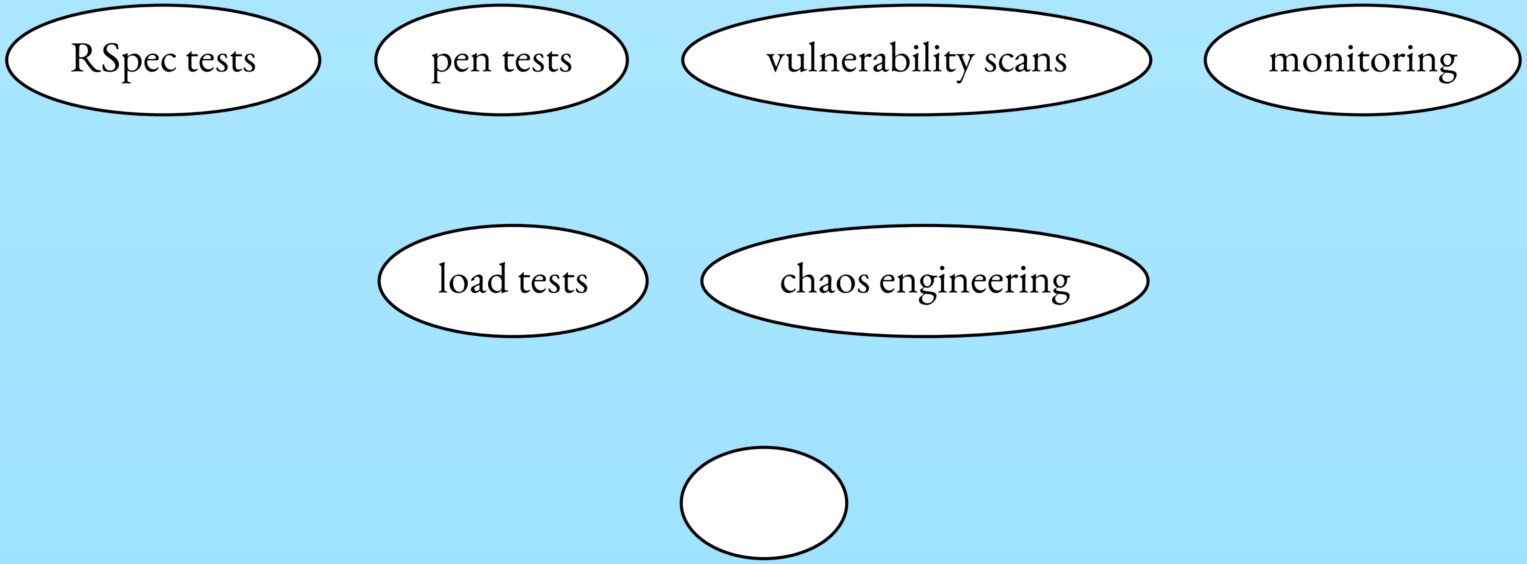
Infrastructure tests for lazy people



RSpec tests

pen tests

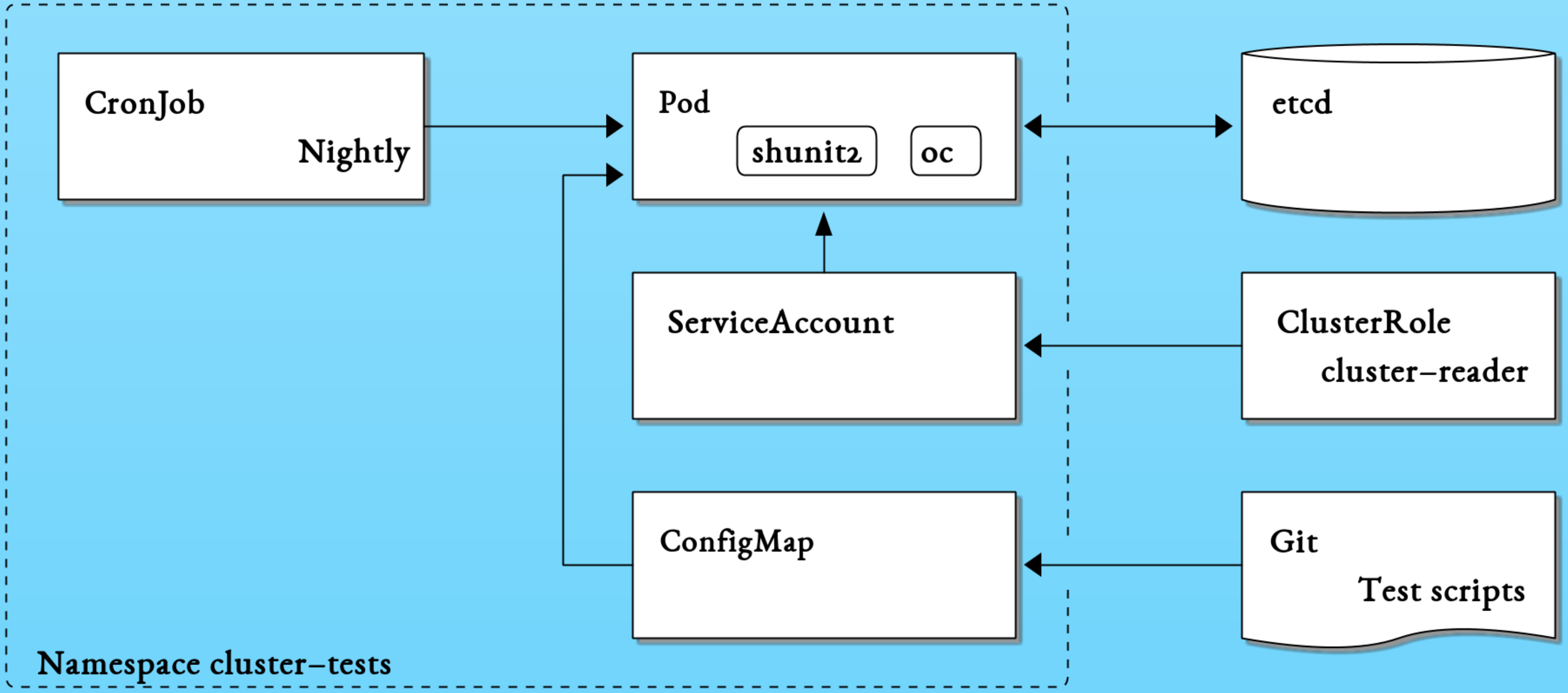
vulnerability scans

monitoring

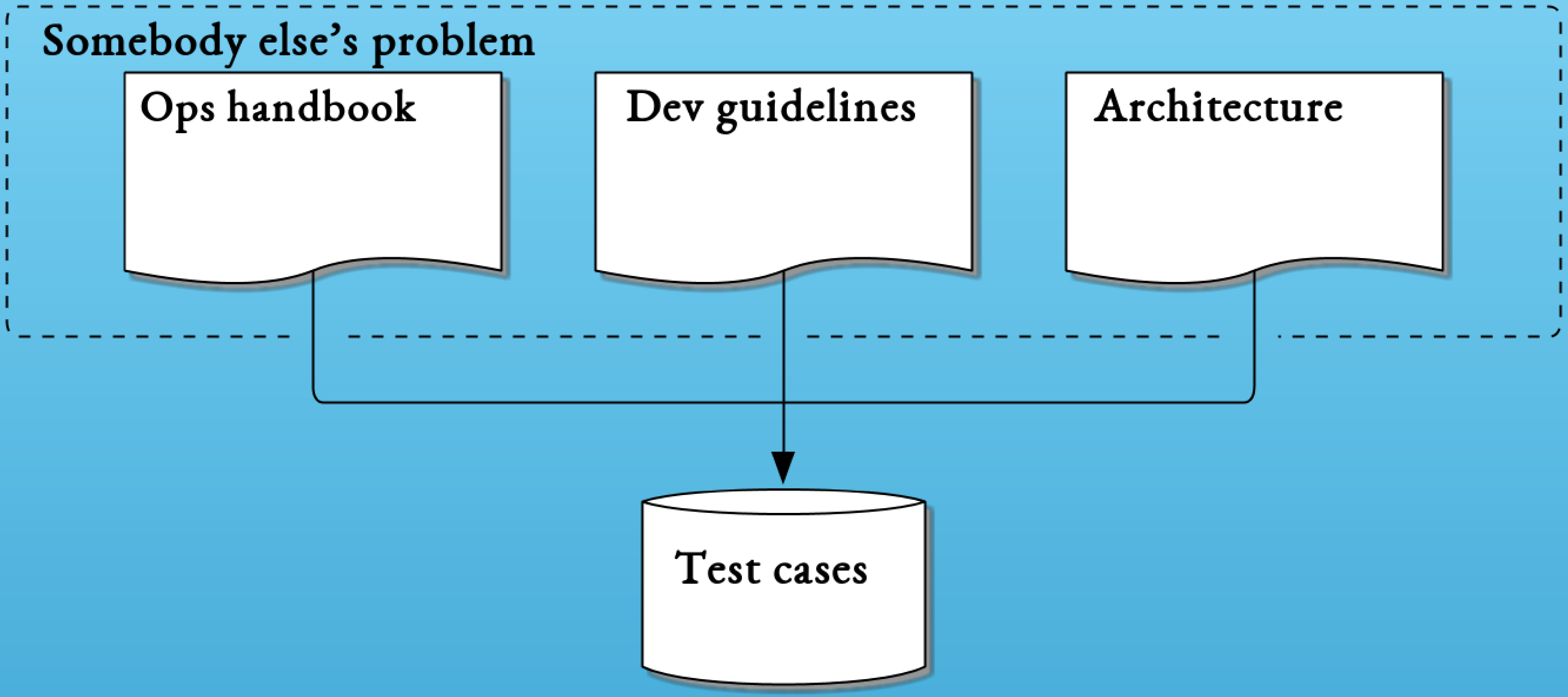
load tests

chaos engineering

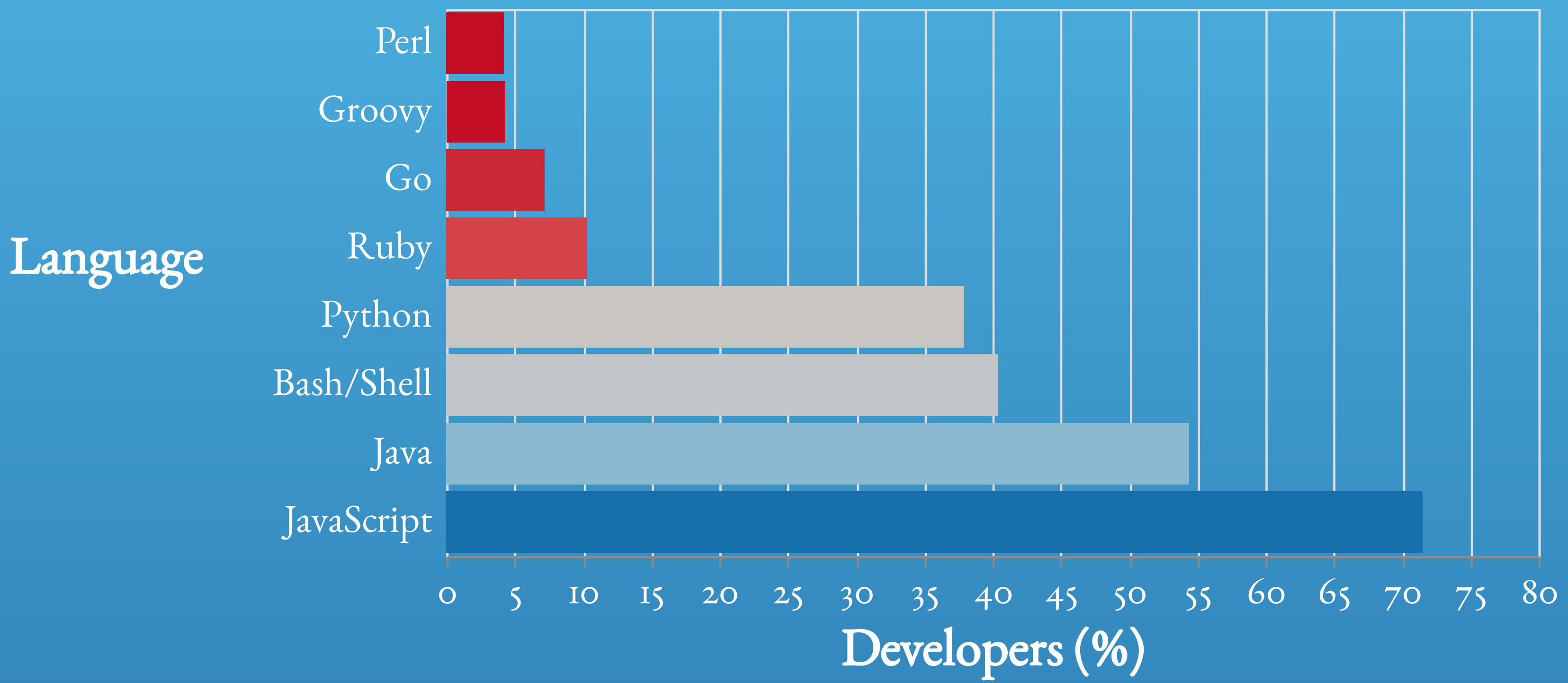
Let someone else do the testing



Let someone else write the brief



Get someone else to write the tests



Source: “Programming, scripting and markup languages”, Stack Overflow survey 2018

Operations

```
test_self_provisioner() {
  count_self_provisioner=`oc adm policy who-can create projectrequests \
    2>/dev/null | grep -c system:authenticated`
  assertEquals " non-admin users must not create project requests;" \
    0 ${count_self_provisioner}
}
suite_addTest test_self_provisioner
```

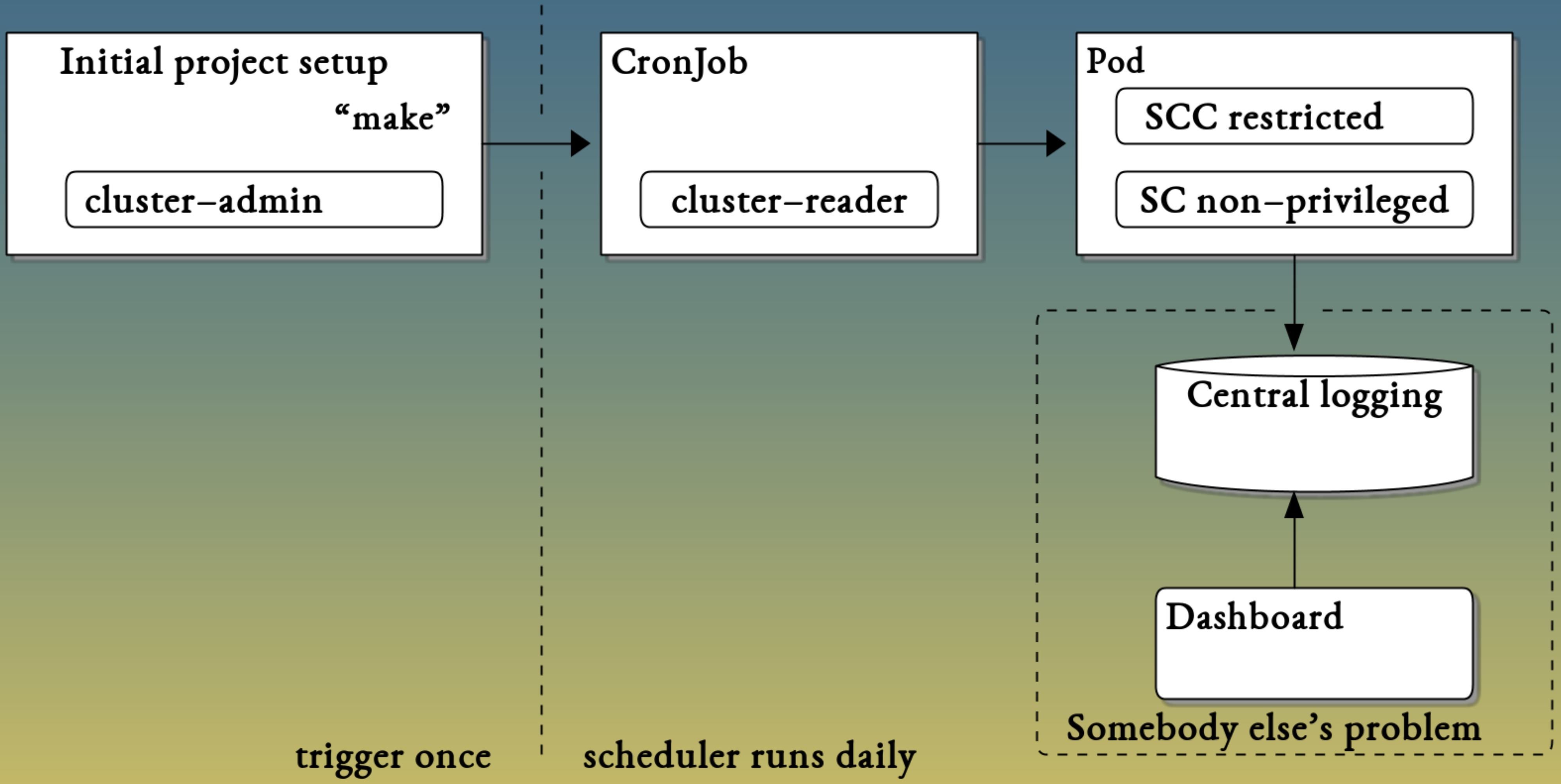
Development

```
test_anyuid() {
  scc_anyuid=`oc describe scc anyuid 2>/dev/null`
  for project in ${USER_PROJECTS}; do
    count_anyuid_default=`echo ${scc_anyuid} | \
      grep -c "Users:.*system:serviceaccount:${project}:default"`
    assertEquals \
      " service account default in project ${project} has SCC anyuid;" \
      0 ${count_anyuid_default}
  done
}
suite_addTest test_anyuid
```

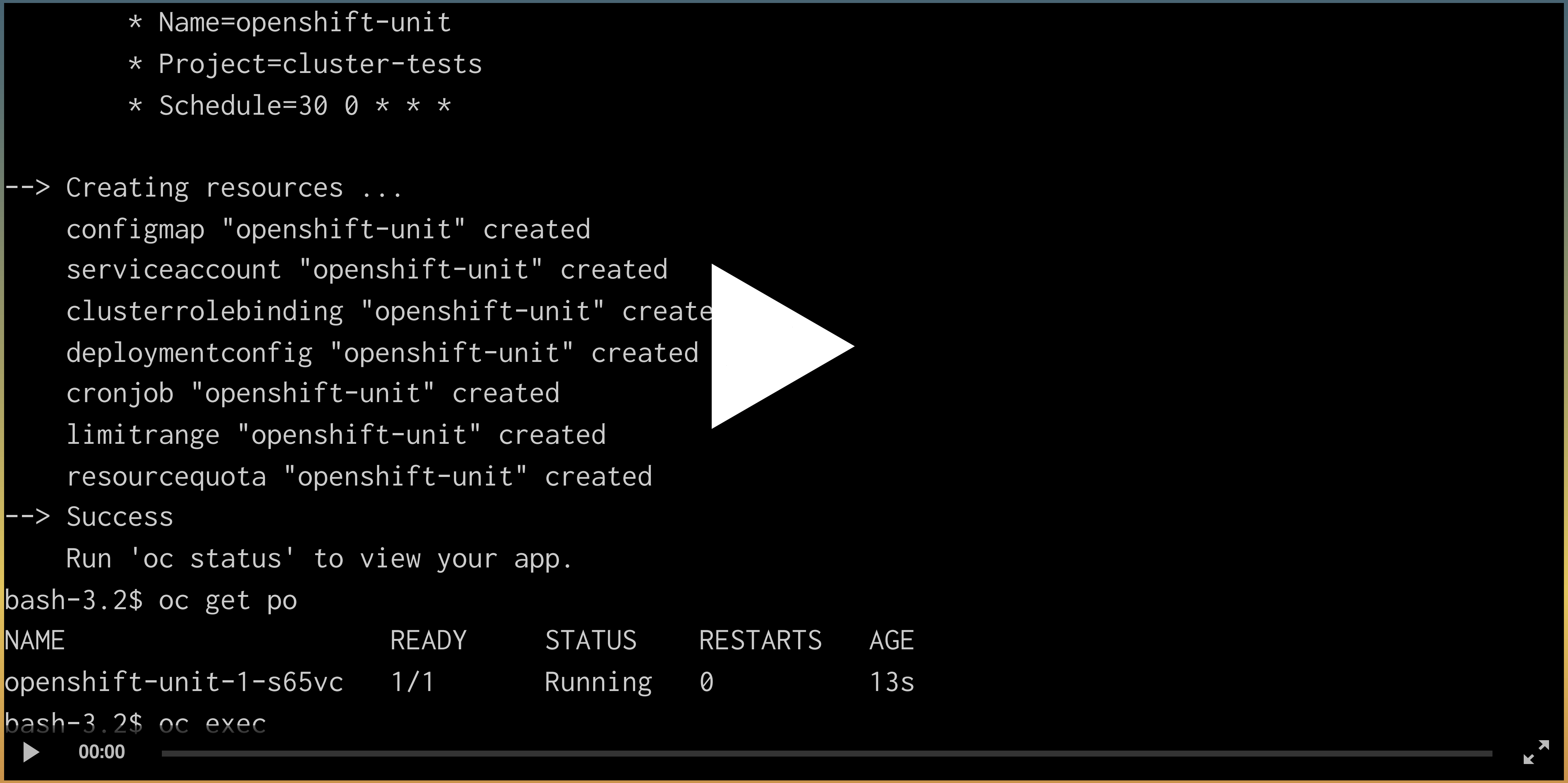
Architecture

```
test_high_availability() {
  for svc in ${HA_SERVICES}; do
    nodes=`oc get po --all-namespaces -o wide | grep ${svc} | \
      awk '{ print $8 }'`
    zones=""
    for node in ${nodes}; do
      zones="${zones} `oc get node/${node} -L zone | \
        awk '{print $6}' | tail -n +2`"
    done
    zone_count=`echo ${zones} | tr ' ' '\n' | sort -u | wc -l`
    ha=$((zone_count > 2))
    assertTrue " ${svc} must be distributed across 3 zones;" ${ha}
  done
}
suite_addTest test_high_availability
```

Let others build the infrastructure for you



What’s left?



```
        * Name=openshift-unit
        * Project=cluster-tests
        * Schedule=30 0 * * *

--> Creating resources ...
    configmap "openshift-unit" created
    serviceaccount "openshift-unit" created
    clusterrolebinding "openshift-unit" create
    deploymentconfig "openshift-unit" created
    cronjob "openshift-unit" created
    limitrange "openshift-unit" created
    resourcequota "openshift-unit" created
--> Success
    Run 'oc status' to view your app.
bash-3.2$ oc get po
NAME                        READY     STATUS    RESTARTS   AGE
openshift-unit-1-s65vc      1/1       Running   0          13s
bash-3.2$ oc exec
```

Why do today what can be done tomorrow?

Do use tests to define the desired cluster state

Don’t fret if it takes longer than expected to reach it

Do take pride in the work not done

Unwind on GitHub

OpenShift cluster tests gerald1248/openshift-unit

OpenShift project backup gerald1248/openshift-backup

Slides courtesy of markdeck by @arnehilmann arnehilmann/markdeck