Checking Best Practices With Clusterlint

Varsha Varadarajan & Adam Wolfe Gordon
DigitalOcean

KubeCon North America 2019

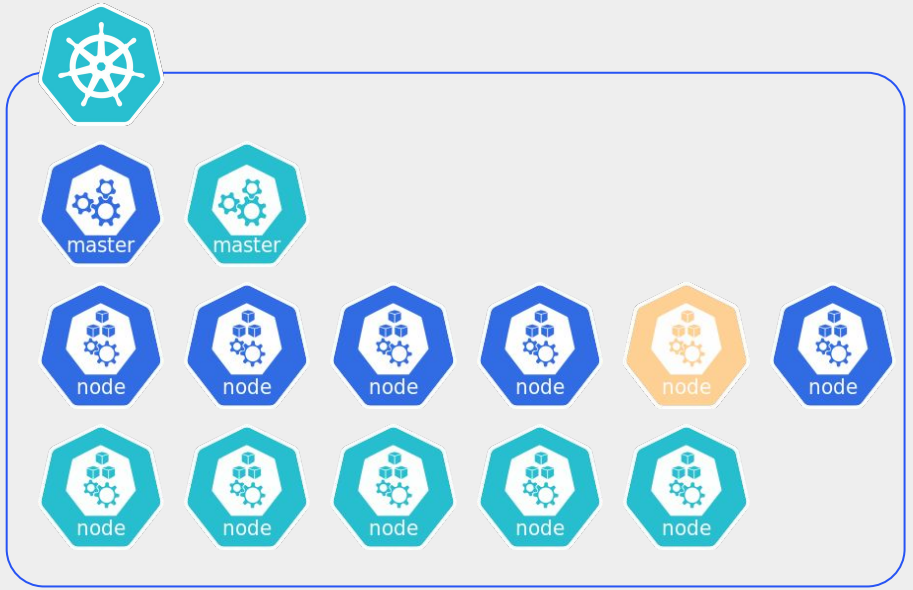
Kubernetes is Flexible

- Runs almost anywhere.
- Can be configured many different ways.
- Can be used many different ways.
- Many strategies for maintenance/upgrades.

- Lots of ways to get things wrong.

Example: DOKS

- In DOKS nodes are immutable.
- If a node breaks, we replace it.
- Upgrades via replacement.



Implications for Workloads in DOKS

- Node names aren’t stable.
  - Don’t use them for scheduling!
- Node labels aren’t persistent.
  - Don’t use them for scheduling!
- Node IP addresses aren’t stable.
  - Don’t point anything at them directly!
- Node filesystems aren’t persistent.
  - Don’t keep important data on them!

Introducing clusterlint

https://github.com/digitalocean/clusterlint

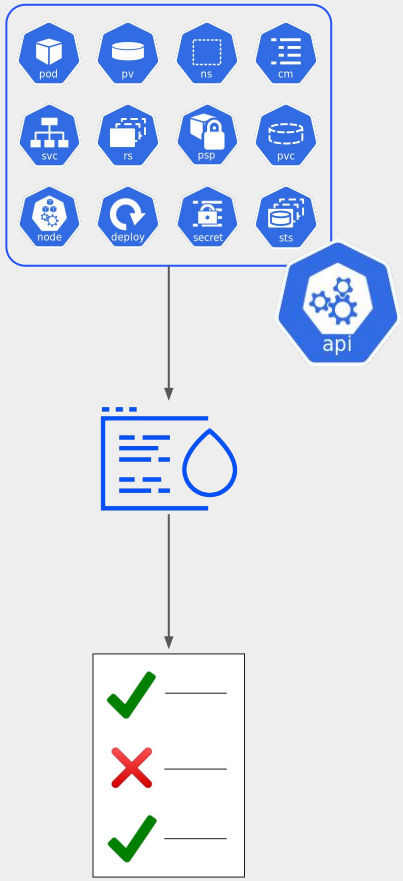
“Clusterlint queries live Kubernetes clusters for resources, executes common and platform specific checks against these resources and provides actionable feedback to cluster operators.”

Goals for clusterlint

- Run against a live cluster, not manifests.
- Allow for platform-specific checks.
- Integrate easily into other code.

How clusterlint Works

1. Fetches resources from k8s.
2. Runs checks on them.
3. Reports results.



Check Registry

- Check Groups
  - doks
  - security
  - basic
- 1:m between checks and groups
- Choose groups to run on command-line with
  - `$ clusterlint run -g doks`
  - `$ clusterlint run -G aws`

Check API

- A check has:
  - Metadata methods.
  - A Run method.
- Checks `Register` themselves.

```
// Check is a check that can run on Kubernetes objects.
type Check interface {
  // Name returns a unique name for this check.
  Name() string
  // Groups returns a list of group names ...
  Groups() []string
  // Description returns a detailed human-readable description ...
  Description() string
  // Run runs this check on a set of Kubernetes objects.
  Run(*kube.Objects) ([]Diagnostic, error)
}

// Register registers a check. This should be called from each check
// implementation's init().
func Register(check Check) error {
  // ...
}
```

Example Check: hostPath Volumes

```
func (h *hostPathCheck) Run(objects *kube.Objects) ([]checks.Diagnostic, error) {
    var diagnostics []checks.Diagnostic
    for _, pod := range objects.Pods.Items {
        for _, volume := range pod.Spec.Volumes {
            if volume.VolumeSource.HostPath != nil {
                d := checks.Diagnostic{
                    Severity: checks.Warning,
                    Message:  fmt.Sprintf("Avoid using hostpath for volume '%s'.", volume.Name),
                    Kind:     checks.Pod,
                    Object:   &pod.ObjectMeta,
                    Owners:   pod.ObjectMeta.GetOwnerReferences(),
                }
                diagnostics = append(diagnostics, d)
            }
        }
    }
    return diagnostics, nil
}
```

Example Check: node name pod selector

```go
func (p *podSelectorCheck) Run(objects *kube.Objects) ([]checks.Diagnostic, error) {
    var diagnostics []checks.Diagnostic
    for _, pod := range objects.Pods.Items {
        nodeSelectorMap := pod.Spec.NodeSelector
        if _, ok := nodeSelectorMap[corev1.LabelHostname]; ok {
            d := checks.Diagnostic{
                Severity: checks.Warning,
                Message:  "Avoid node name label for node selector.",
                Kind:     checks.Pod,
                Object:   &pod.ObjectMeta,
                Owners:   pod.ObjectMeta.GetOwnerReferences(),
            }
            diagnostics = append(diagnostics, d)
        }
    }
    return diagnostics, nil
}
```

How to fix object configuration

```yaml
# Not recommended: Defining resources with no
# namespace, which adds them to the default.
apiVersion: v1
kind: Pod
metadata:
  name: mypod
  labels:
    name: mypod
spec:
  containers:
  - name: mypod
    image: nginx:1.17.0
```

```yaml
# Recommended: Explicitly specify a namespace in
# the object config
apiVersion: v1
kind: Pod
metadata:
  name: mypod
  namespace: test
  labels:
    name: mypod
spec:
  containers:
  - name: mypod
      image: nginx:1.17.0
```

How to fix object configuration

```yaml
# Not recommended: Using a raw DigitalOcean
# resource name in the nodeSelector
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
  nodeSelector:
    kubernetes.io/hostname: pool-y25ag12r1-xxxx
```

```yaml
# Recommended: Use the DOKS-specific node pool
# label
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
  nodeSelector:
      doks.digitalocean.com/node-pool: pool-y25ag12r1
```

Suppressing Checks

- Via command-line or API:
  - Explicitly include or exclude checks.
  - Explicitly include or exclude groups of checks.
- Via annotations:
  - Annotate an object to have it exempt from certain checks:

```
metadata:
 annotations:
   clusterlint.digitalocean.com/disabled-checks: "privileged-containers"
```

DOKS Product Integration

- API and UI to run clusterlint on a DOKS cluster.
- Runs asynchronously.
  - API call to request a run returns a run ID.
  - API call to get results.

DOKS API Integration

Request Run

```
$ curl -X POST https://api.digitalocean.com/v2/kubernetes/clusters/$CLUSTER_ID/clusterlint

{"run_id":"cd259c55-0501-4ea8-a417-cb0fcbc04921"}
```

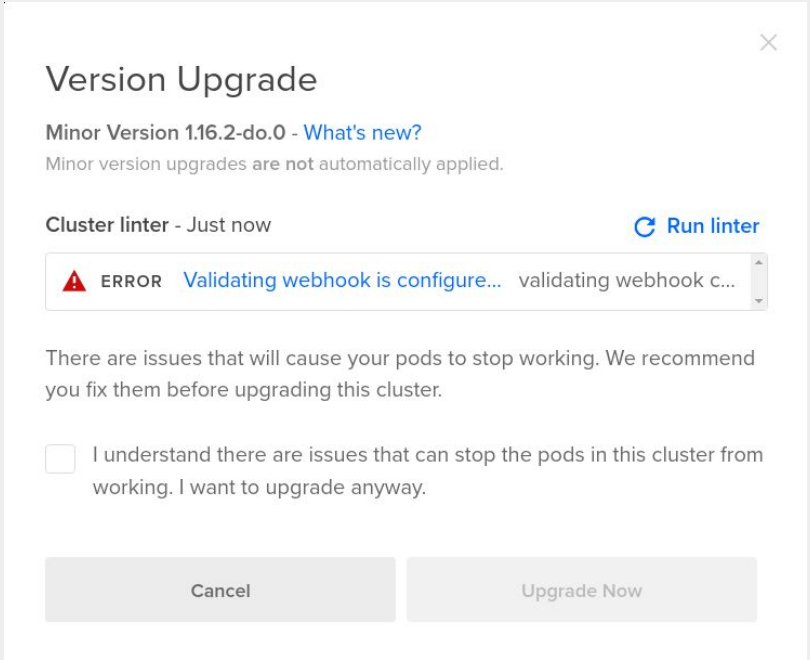
Get Results

```
$ curl https://api.digitalocean.com/v2/kubernetes/clusters/$CLUSTER_ID/clusterlint

{
  "run_id": "cd259c55-0501-4ea8-a417-cb0fcbc04921",
  "requested_at": "2019-10-28T18:50:29Z",
  "completed_at": "2019-10-28T18:50:31Z",
  "diagnostics": [
    {
      "check_name": "admission-controller-webhook",
      "severity": "error",
      "message": "Validating webhook is configured in such a way that it may be problematic during
upgrades.",
      "object": {
        "kind": "validating webhook configuration",
        "name": "webhook.example.com",
        "namespace": ""
      }
    }
  ]
}
```

DOKS UI Integration

- Runs automatically before upgrades.
  - Option to cancel if there are problems.
- Button to trigger a re-run.



github.com/digitalocean/clusterlint

digitalocean.com

Future Plans

- More checks.
- Better integration in DOKS.
  - Run as part of auto-upgrades, warn via email.
  - Run before other disruptive operations.
  - Run periodically and show results in control panel.
- Version-specific checks.
  - E.g., for API group deprecation.

Other Similar Tools

- Manifest-based linters:
  - kubeval - https://github.com/instrumenta/kubeval
  - copper - https://copper.sh
  - kube-lint - https://github.com/viglesiasce/kube-lint
- Security checker:
  - kube-bench - https://github.com/aquasecurity/kube-bench
- Dashboard:
  - Polaris - https://github.com/FairwindsOps/polaris
- Command-line “report card” tool:
  - Popeye - https://github.com/derailed/popeye

Help Wanted!

- Please try it on your cluster and fix things!
  - Or report bugs!
- Please add checks!
  - Especially environment-specific ones!

Thank You!

Adam Wolfe Gordon - awg@do.co
Varsha Varadarajan - vvaradarajan@do.co

https://github.com/digitalocean/clusterlint

DigitalOcean


DigitalOcean