Vallery Lancey, Lyft

SIG-Usability Deep Dive

@vllry

What We Do

Work to make Kubernetes more usable.

github.com/kubernetes/community/tree/master/sig-usability

We’re New Here

- SIG-Usability formed this summer.
- “Horizontal” SIG.
  - Owns cross-SIG concepts.
- Working on user surveys, user personas, etc.

Dive: Default Security Considerations

- NetworkPolicy must be configured manually
- PodSecurityPolicy must be configured manually

Many people run production workloads without these.

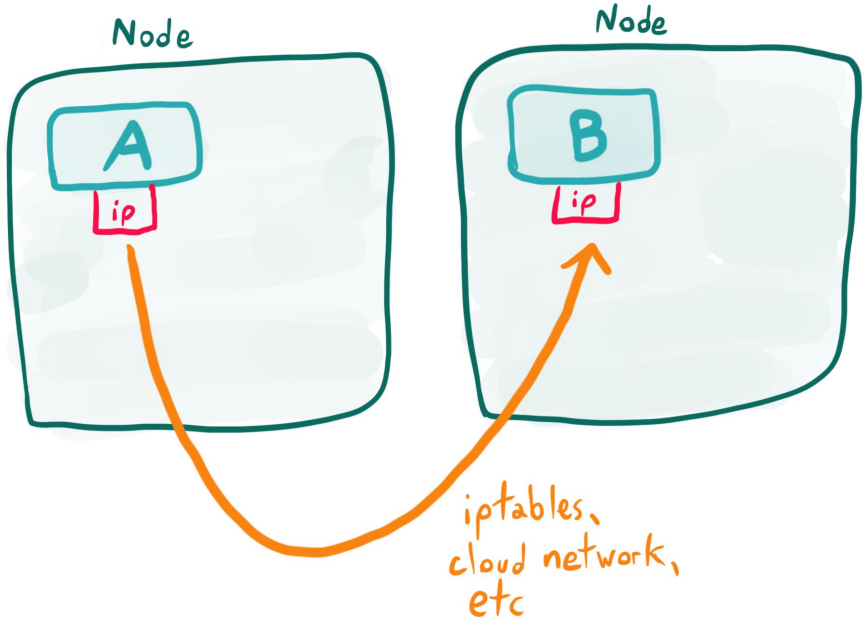

KubeCon
CloudNativeCon
North America 2019
NetworkPolicy
@vllry

Network Abstractions

- Services and Endpoints are defined within namespaces.
- Service DNS short-names are only visible within their namespace.

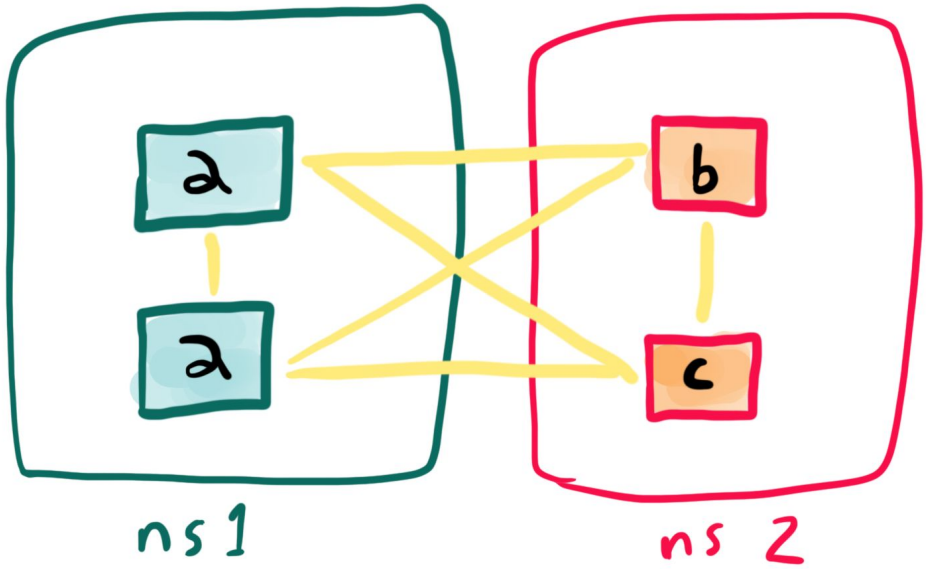
Kubernetes CNI

- Need a CNI (Container Network Interface) driver to enable pod-pod networking.
- CNI gives pods IP addresses, and handles host/cloud networking.



Pod-Pod Networking

- Any pod can talk to any other pod.
- This isn't obvious in documentation or in practice.

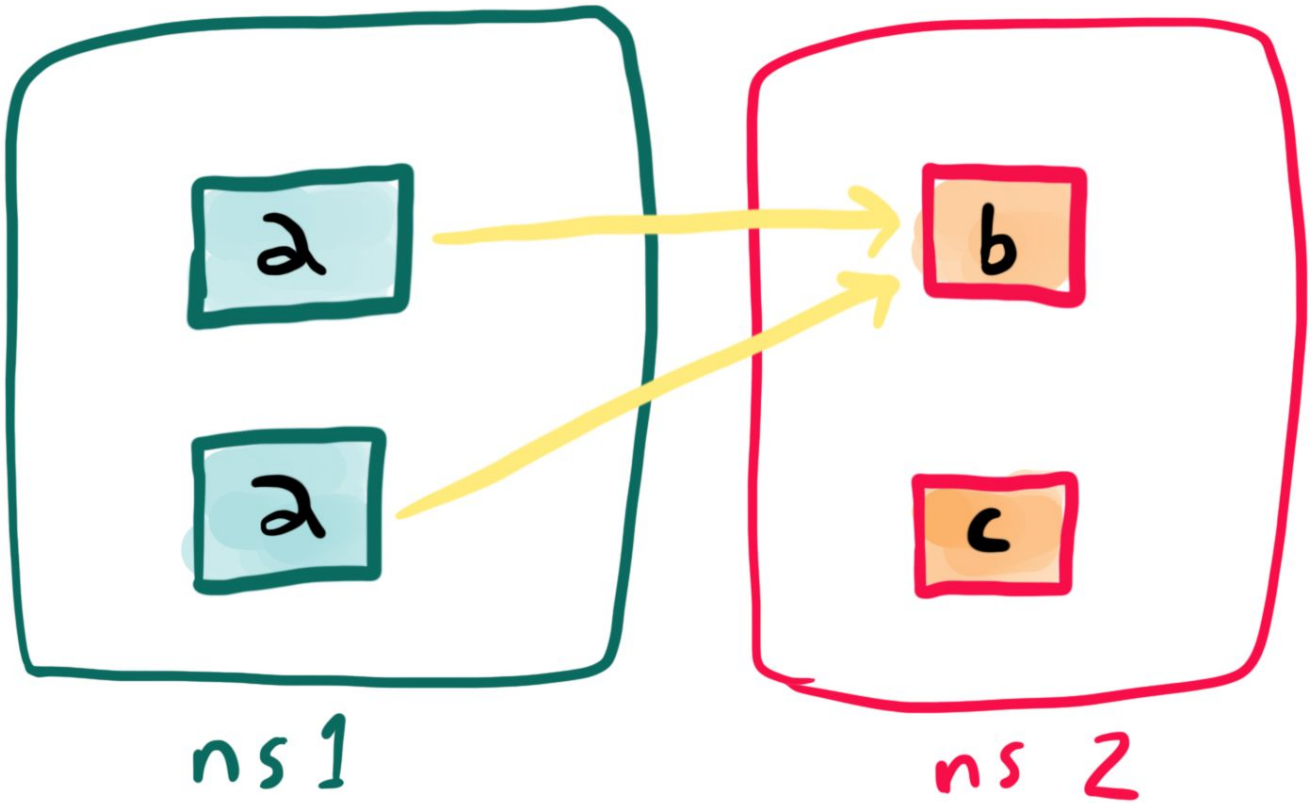


NetworkPolicy

- The NetworkPolicy object restricts pod-pod network access.
- NetworkPolicy must be supported by the CNI driver.
  - Restricts access in iptables rules, cloud network rules, etc.
- Pods are default unrestricted, *unless* a NetworkPolicy selects them.

NetworkPolicy - Example


```
...
 spec:
   podSelector:
     matchLabels:
       app: b
   ingress:
     - from:
       - namespaceSelector:
           matchLabels:
             team: a
         podSelector:
           matchLabels:
             app: a
```


a
a
b
c
ns 1
ns 2

Let’s Change This…?

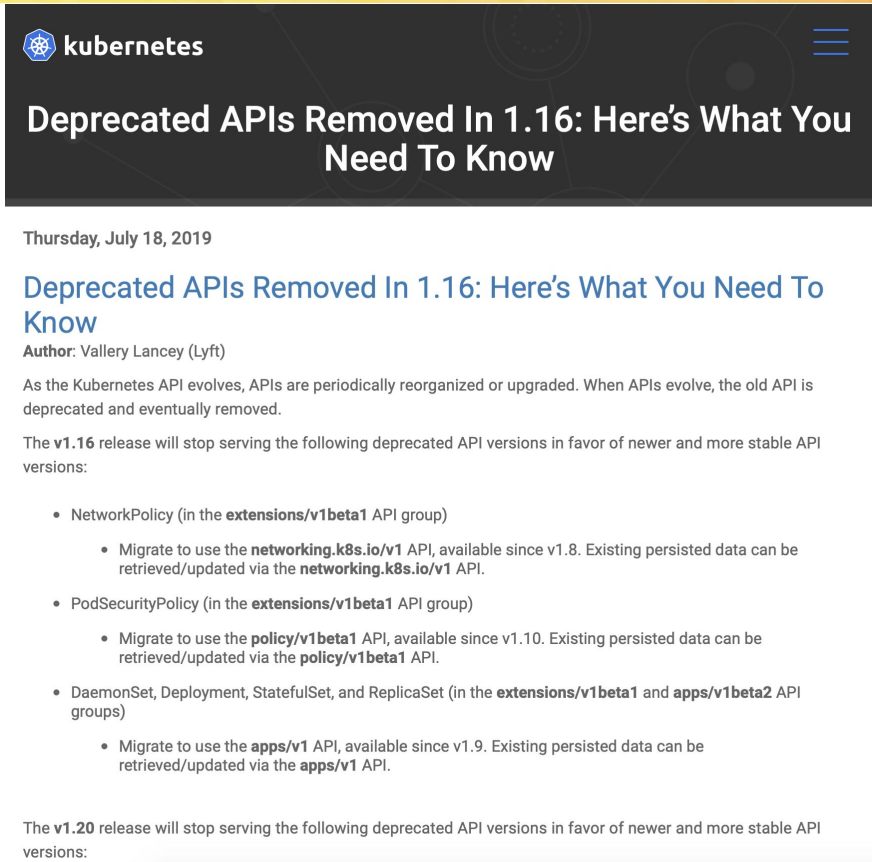
- If we make the network “default-closed”, users will need to create explicit rules.
  - Users can explicitly create a wide-open network if they want, and accept the risks.

Why Isn’t It Like This Already?

- **Compatibility** with existing systems.
- **Support** for NetworkPolicy is (deliberately) flexible.
- **Convenience** for users.

Lessons From The 1.16 API Removal

- Some old API versions were removed in 1.16.
- We talked about this a lot.
- Lots broke and lots of people were angry anyway.



kubernetes

Deprecated APIs Removed In 1.16: Here's What You Need To Know

Thursday, July 18, 2019

Deprecated APIs Removed In 1.16: Here's What You Need To Know

Author: Vallery Lancey (Lyft)

As the Kubernetes API evolves, APIs are periodically reorganized or upgraded. When APIs evolve, the old API is deprecated and eventually removed.

The **v1.16** release will stop serving the following deprecated API versions in favor of newer and more stable API versions:

- NetworkPolicy (in the **extensions/v1beta1** API group)
  - Migrate to use the **networking.k8s.io/v1** API, available since v1.8. Existing persisted data can be retrieved/updated via the **networking.k8s.io/v1** API.
- PodSecurityPolicy (in the **extensions/v1beta1** API group)
  - Migrate to use the **policy/v1beta1** API, available since v1.10. Existing persisted data can be retrieved/updated via the **policy/v1beta1** API.
- DaemonSet, Deployment, StatefulSet, and ReplicaSet (in the **extensions/v1beta1** and **apps/v1beta2** API groups)
  - Migrate to use the **apps/v1** API, available since v1.9. Existing persisted data can be retrieved/updated via the **apps/v1** API.

The **v1.20** release will stop serving the following deprecated API versions in favor of newer and more stable API versions:

How Do We Introduce A Default?

- New behavior can’t break existing setups or clusters.
- *Has* to (start) opt-in.
  - ...set a CLI flag.
  - ...set a cluster config object.
  - ...create a new resource version.
- *Should* be easy to migrate to.

How Do We Introduce A Default?

- Changing the behavior of “unselected” pods isn’t possible.
  - Breaks API contracts.
- What about a NetworkPolicy v2…?

How Do We Introduce A Default?

- What about a *namespace* v2?
- Namespace v2's could automatically create a deny-all NetworkPolicy.
  - Similar to the default service account creation.
- This respects the expectation: no NetworkPolicy selection → no restriction.
- (Theoretical) KEP in progress: github.com/kubernetes/enhancements/pull/1329

Kubernetes Conformance Logistics

- Kubernetes has a “conformance” test suite.
  - Essential E2E tests for a distro to be considered compliant with Kubernetes.
- If we change behavior, many network conformance tests may break.

User Experience

- Setting up NetworkPolicy config shouldn’t be hard.
  - Must be clear.
  - Must not be overly repetitive.
  - Must not have complex coupling.
- Making more users use it will expose sharp edges.
  - Unpopular design choice in Istio.


KubeCon
CloudNativeCon
North America 2019
PodSecurityPolicy
@vllry

Basics

- PodSecurityPolicies restrict what pods can do.
- Acts via admission hooks.
- Blocks the creation/update of pods with a spec that violates the policy.

If you’re not using PodSecurityPolicy, someone who can create pods can hijack your cluster.

@vllry

PSP

...blocks “privileged” (root access) containers

```
spec:
   privileged: false
   ...
```

@vllry

PSP

```
spec:
  volumes:
     - 'configMap'
     - 'emptyDir'
     - 'projected'
     - 'secret'
     - 'downwardAPI'
     - 'persistentVolumeClaim'
   ...
```

...blocks mounting arbitrary directories, like /proc

@vllry

Obstacles

- **Compatibility** with existing systems.
- **Convenience** for users.
- **Future** of PSP?

Is PSP The Long-Term Solution?

- In short: probably not.
- Admission control is complex, and needs vary drastically. PSP has limited scope.
  - See OPA, etc.
- See SIG-Auth session tomorrow.

Want to Get Involved?

Things We're Working On

- User studies.
- "Jobs to be done" profiling.
- Better out-of-the-box behavior.
- Documentation (with sig-docs).

Where To Find Us

github.com/kubernetes/community/tree/master/sig-usability

- **#sig-usability** on slack.k8s.io
- **kubernetes-sig-usability** on Google Groups