KubeCon | CloudNativeCon
North America 2020
Virtual

Kubelet Deep Dive: Writing a Kubelet in Rust

Kevin Flansburg

About Me

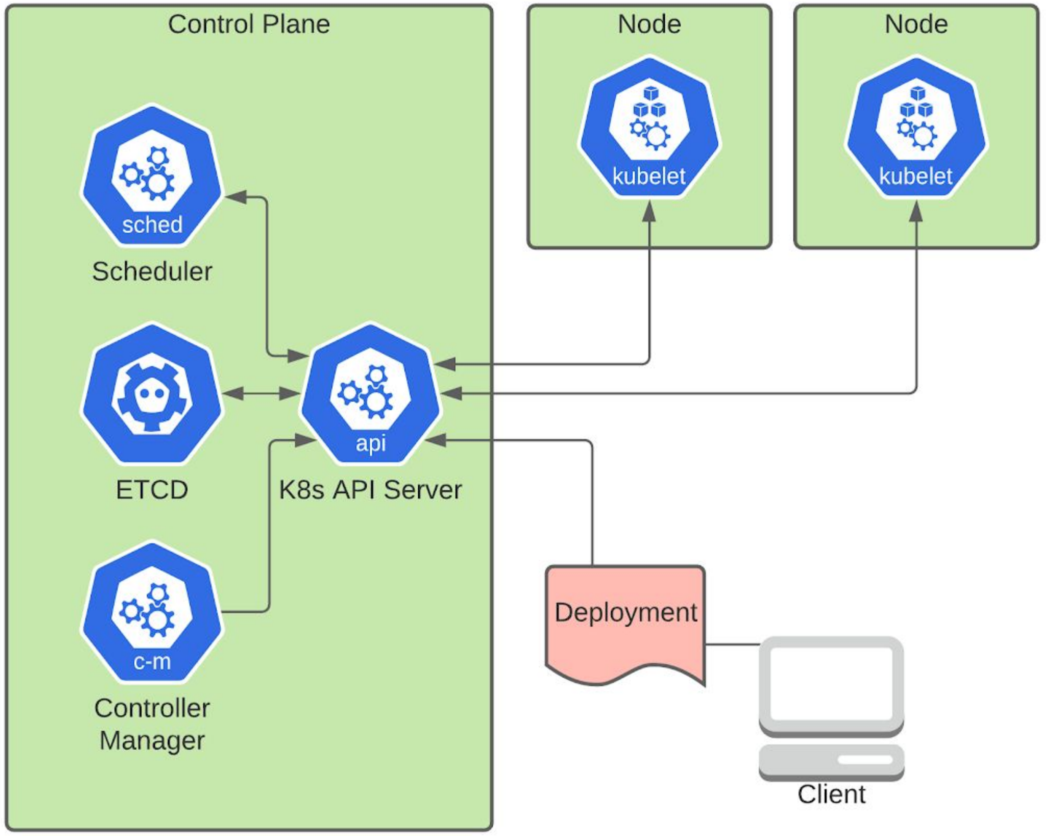
- Rust - 3 years
  - Microservices
  - DevOps
- Kubernetes - 1 year
- Krustlet maintainer

Krustlet Project

- **K**ubernetes **Rust** Kube**let**
- Deis Labs
- GitHub Repository

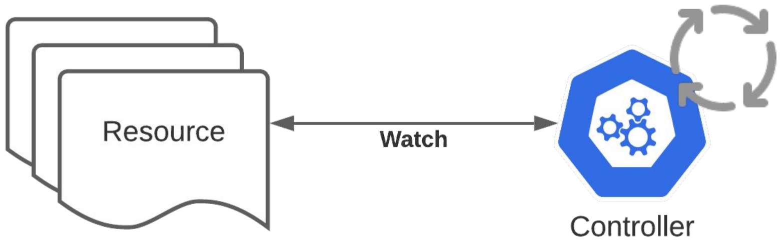
- `kubelet` - crate for building Kubelets
- Kubelet Implementations:
  - waSCC
  - WASI
  - Linux Containers via CRI

Kubernetes Architecture



Controller Pattern

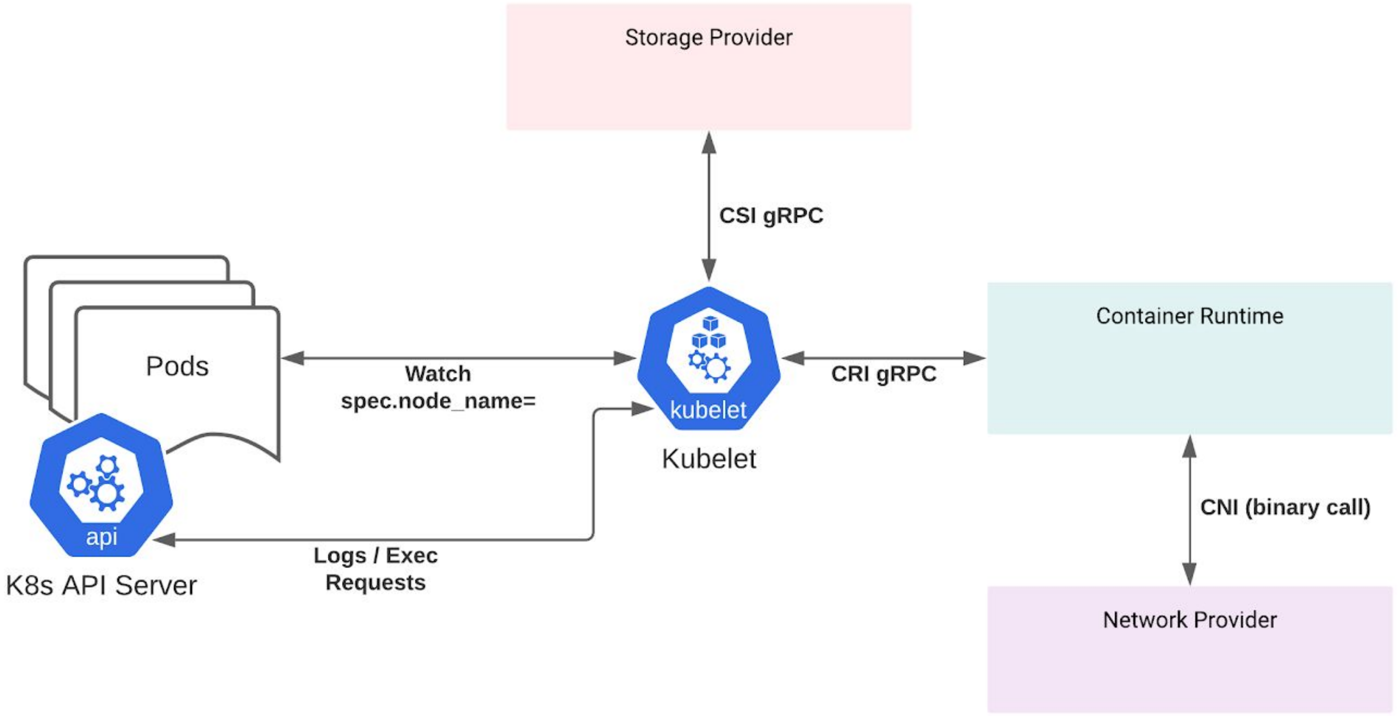
- (Mostly) immutable resource objects.
- Monitor for changes to objects of a particular resource.
  - Add / Modify / Delete
  - “Informer” pattern
- Drive cluster state to match this desired state.



Aside: Operators

- Controller with:
  - Custom Resource Definition
  - Application / domain specific

Kubelet: Franken-controller



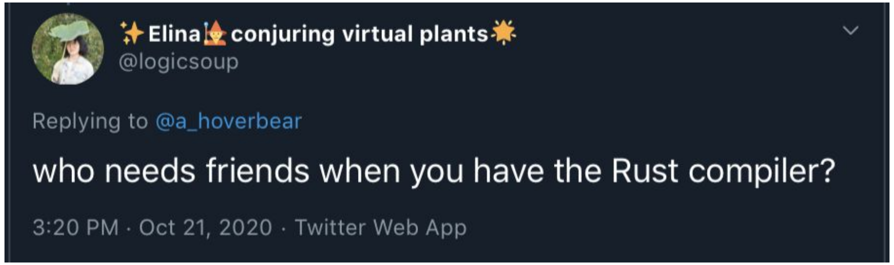
Rust for Distributed Apps

- Performance
- Strongly typed
- Ownership - “fearless concurrency”
- `async/await`
- Error handling

```
let result = failable_fn();
match result {
     Ok(value) => …,
     Err(e) => …,
}
```

Or simply:

```
let result = failable_fn()?;
```

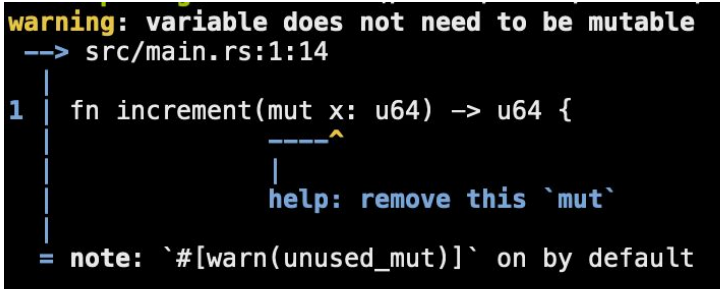


✨Elina🔥conjuring virtual plants🌟
@logicsoup

Replying to @a_hoverbear

who needs friends when you have the Rust compiler?

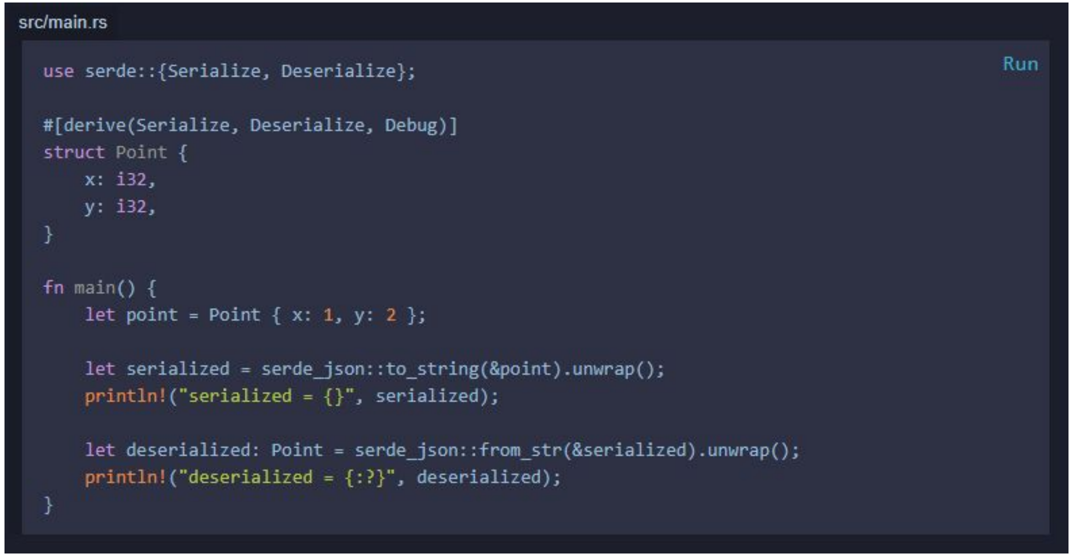
3:20 PM · Oct 21, 2020 · Twitter Web App



```
warning: variable does not need to be mutable
 --> src/main.rs:1:14
  |
1 | fn increment(mut x: u64) -> u64 {
  |              ----^
  |              |
  |              help: remove this `mut`
  |
  = note: `#[warn(unused_mut)]` on by default
```

Rust Ecosystem

- Many fantastic crates
  - serde
  - tracing
  - prost / tonic
- Documentation
- Dependency management
- Great community



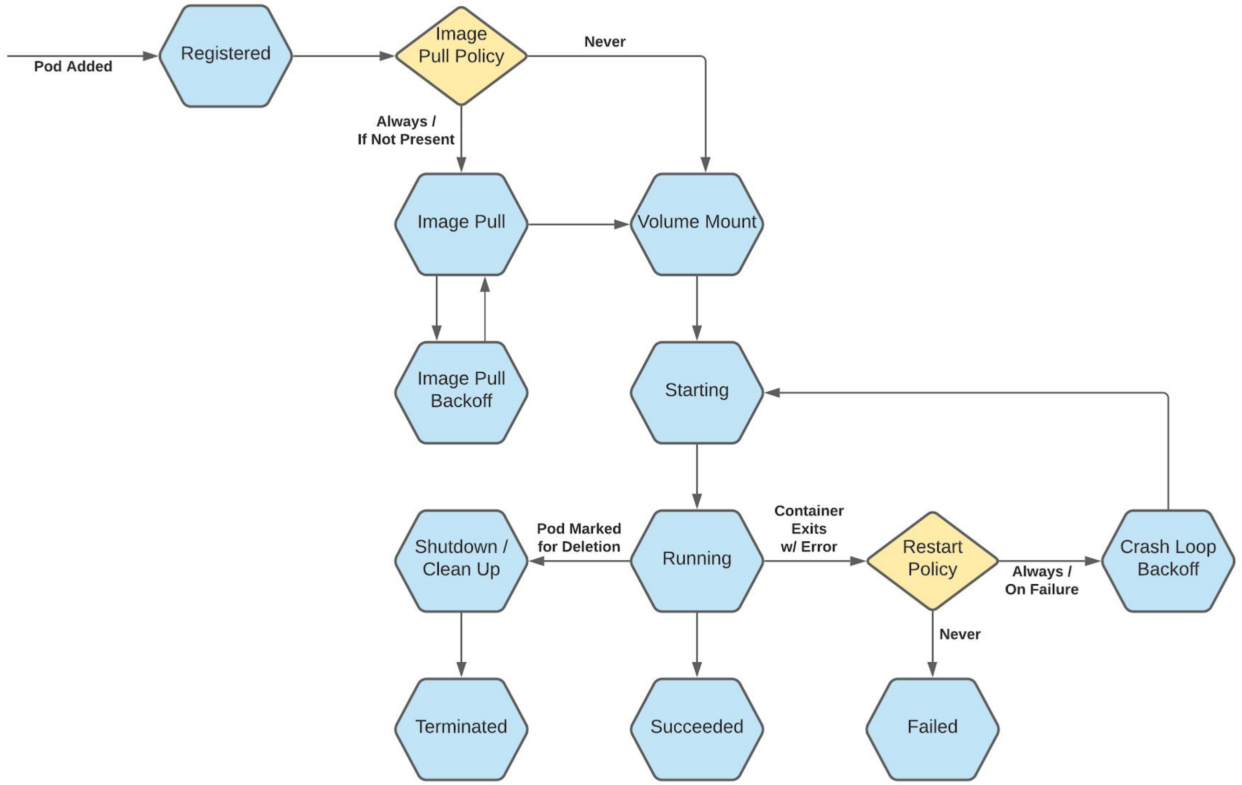
```rust
use serde::{Serialize, Deserialize};

#[derive(Serialize, Deserialize, Debug)]
struct Point {
    x: i32,
    y: i32,
}

fn main() {
    let point = Point { x: 1, y: 2 };

    let serialized = serde_json::to_string(&point).unwrap();
    println!("serialized = {}", serialized);

    let deserialized: Point = serde_json::from_str(&serialized).unwrap();
    println!("deserialized = {:?}", deserialized);
}
```

Serde Demo in Runnable Documentation Example

Useful Kubernetes Crates

- `k8s-openapi` - Rust types for Kubernetes API resources.
- `kube` - Kubernetes client.
- `k8s-cri` - gRPC client for Container Runtime Interface (CRI)
- `k8s-csi` - gRPC client for Container Storage Interface (CSI)

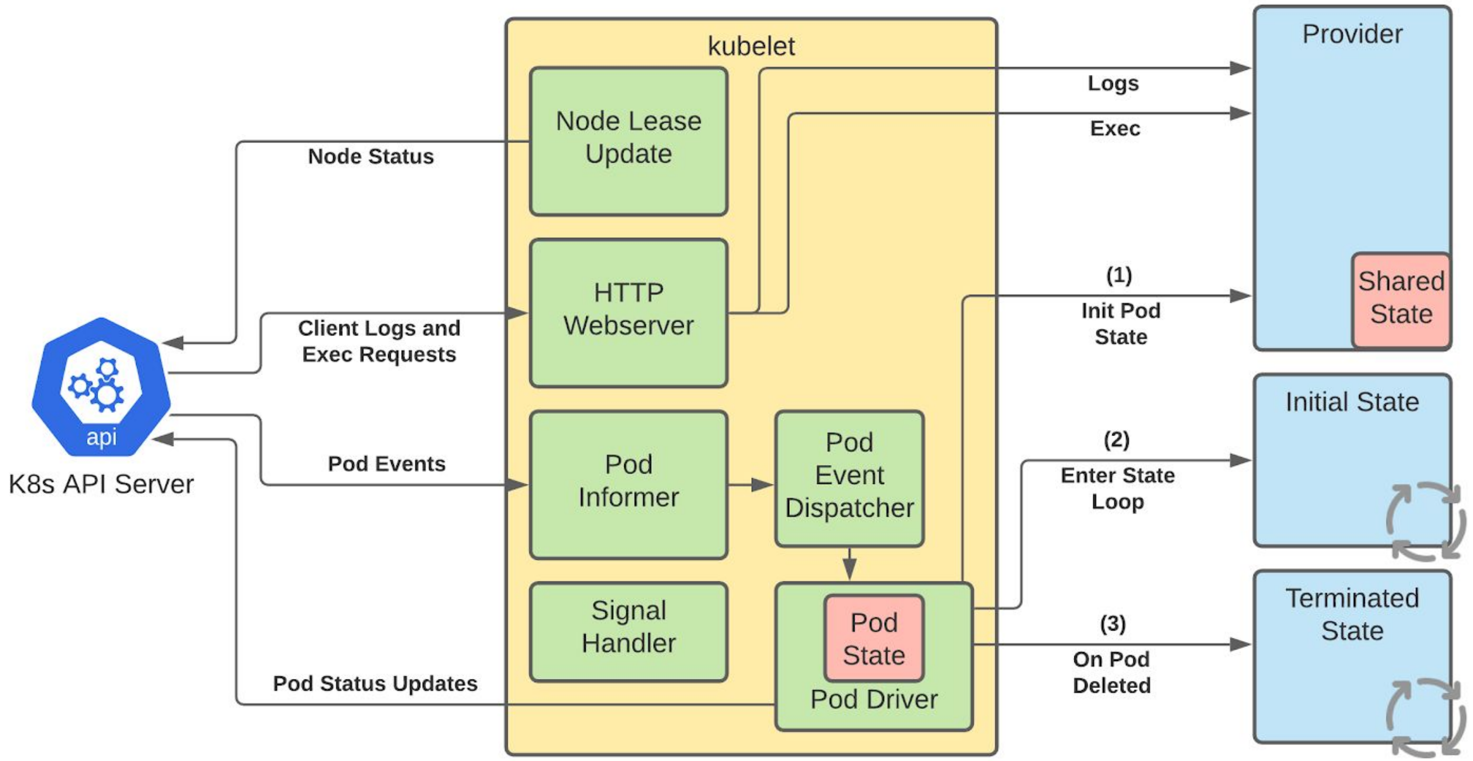
Kubelet Control Loop



Rust State Machine

- A Fistful of States: More State Machine Patterns in Rust
- Flexible framework for implementing Kubelet control loop.
- Enforced at compile time:
  - Valid states
  - Valid state transitions
- Automatic Pod status updates
- Error handling within context of control loop.

Krustlet Architecture



Conclusion

- Kubelet communication patterns
- Pod behavior
- Rust
- Shout outs
    - Taylor Thomas
    - Matt Fisher
    - Ivan Towlson
- Contributing to Krustlet

KEEP CLOUD NATIVE EVERYWHERE

KubeCon | CloudNativeCon

North America 2020

Virtual