



KubeCon



CloudNativeCon

North America 2024





KubeCon



CloudNativeCon

North America 2024

Testing Kubernetes... without Kubernetes

John Howard (Solo.io)



KubeCon



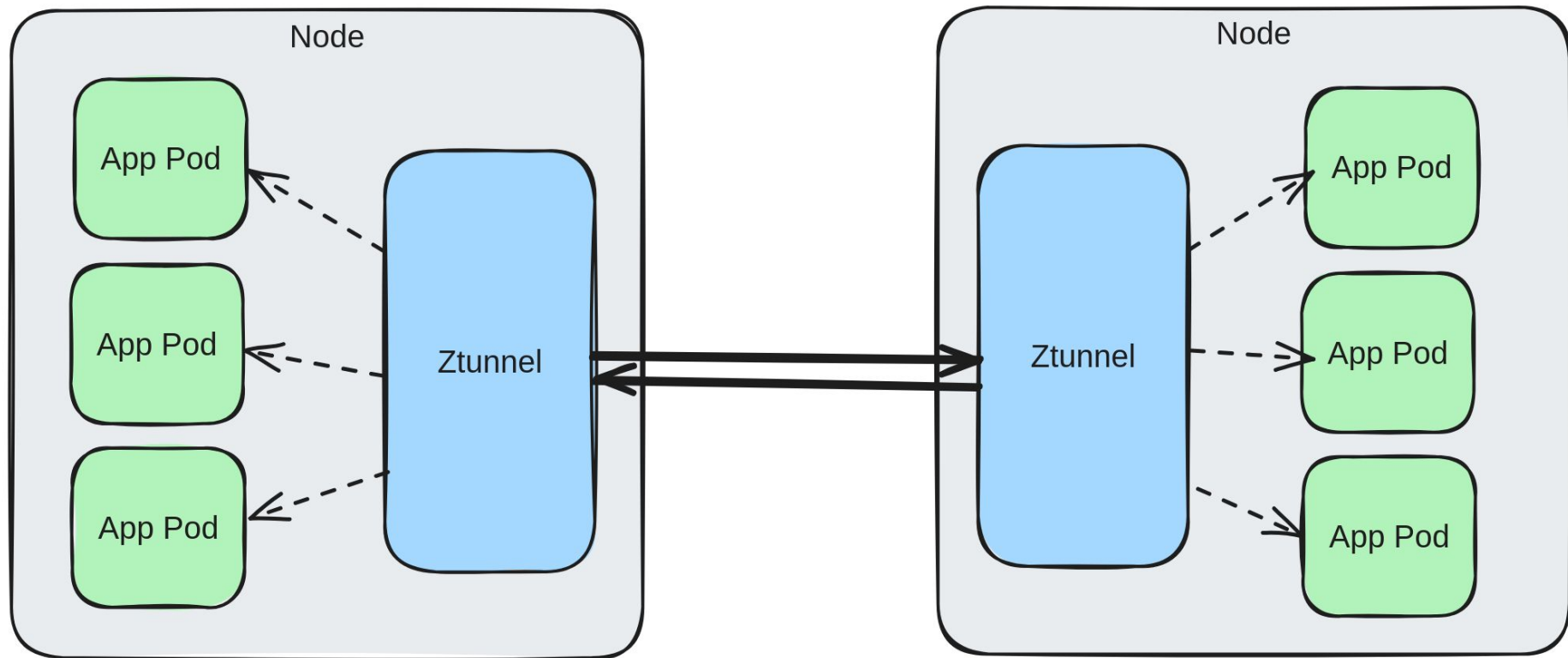
CloudNativeCon

North America 2024

...is ~~rough!~~ necessary?

- Slow
- Hard to debug
- Hard to onboard
- Expensive, Unreliable, ...

Building a Next Generation Service Mesh



Istio Ambient Mode Networking

- Testing goals
- Kubernetes internals
 - What is a namespace anyways?
- Simulating Kubernetes namespaces
- Building tests
- More namespaces!

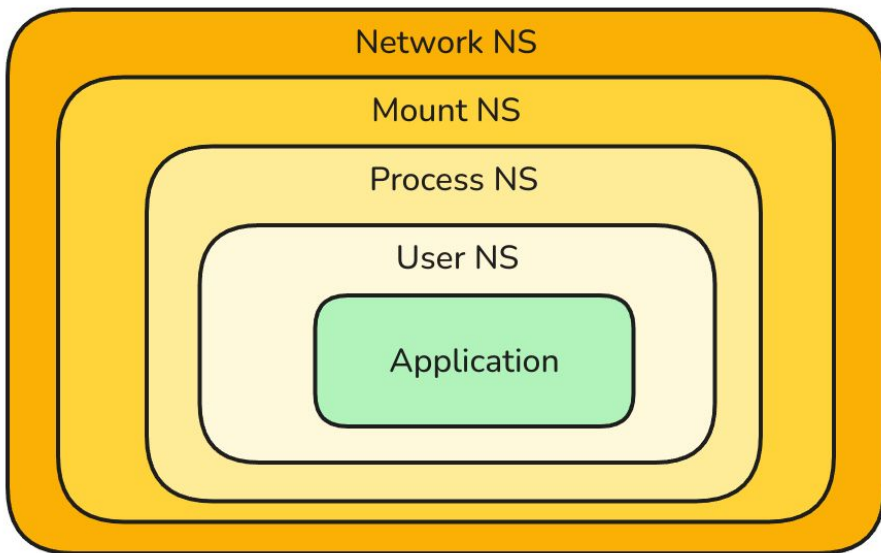
Tests can run...

- Quickly (<1s per test)
- Without any setup (`go test ./...`)
- Easy to debug (with standard tooling)
- Without needing Kubernetes
 - Or docker
 - Or root?

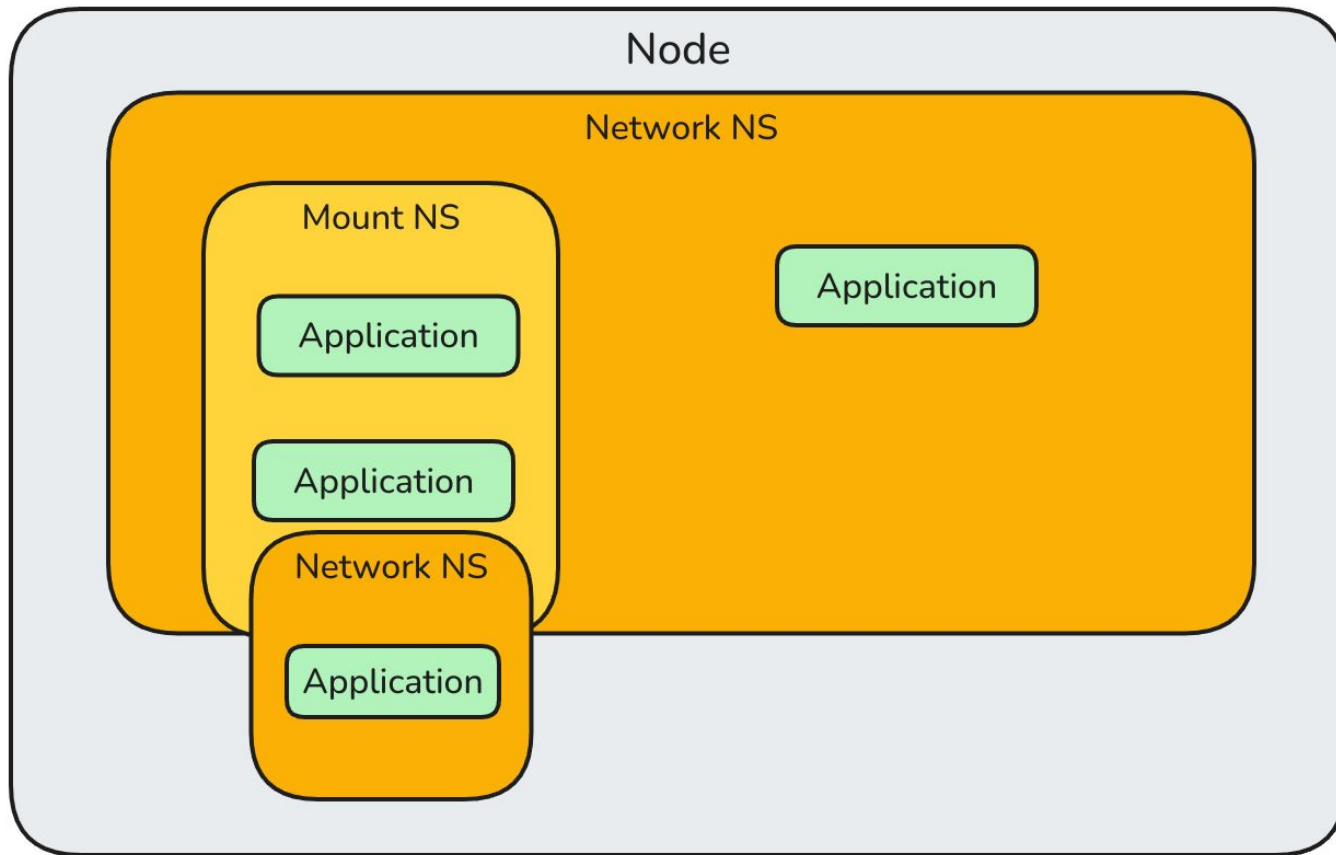
Perception



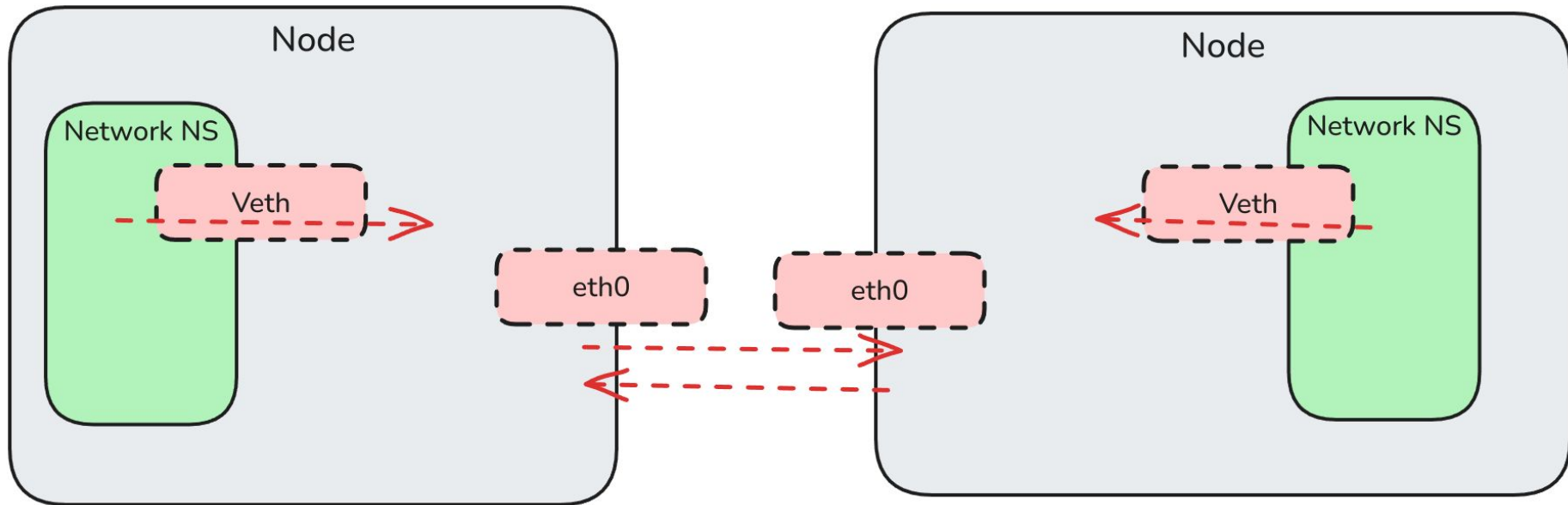
Reality?



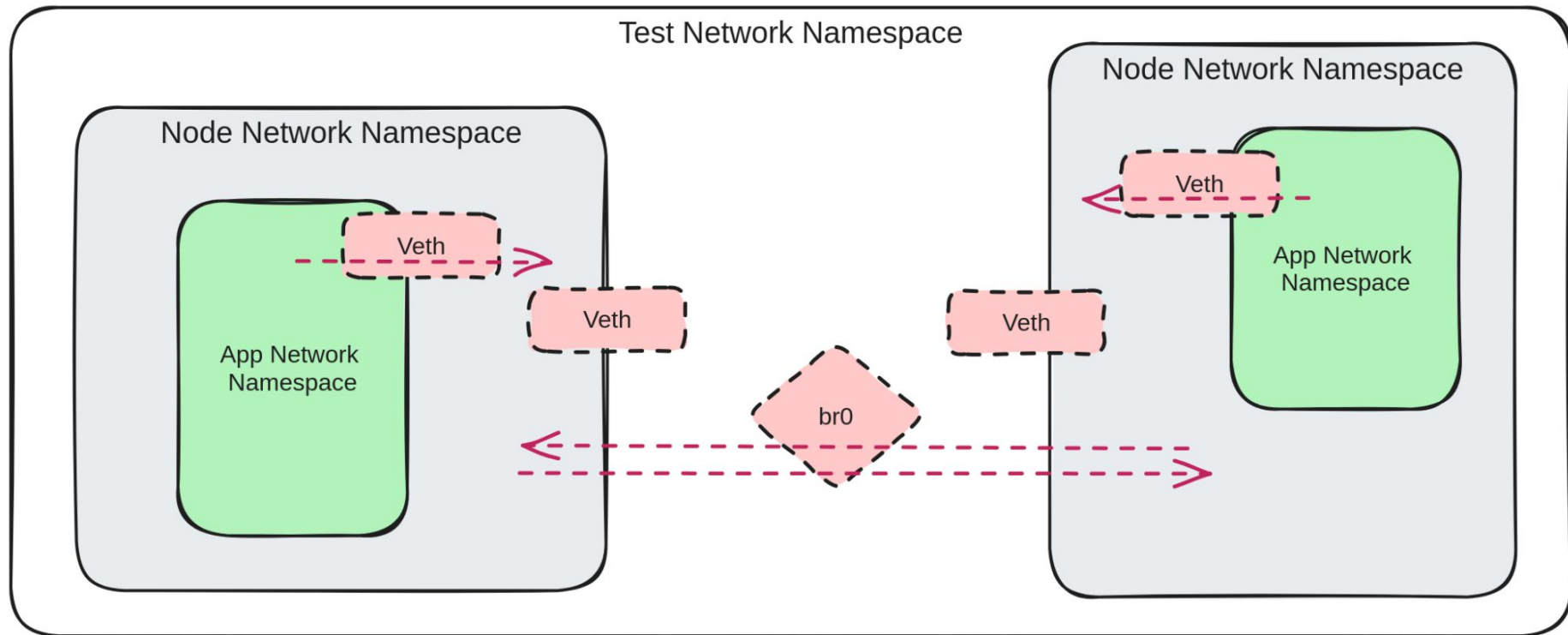
Namespaces 101



Kubernetes Networking



Test Networking



Test Networking

```
// Give the namespace a veth and configure routing
// Largely inspired by https://github.com/containernetworking/plugins/blob/main/plugins/main/ptp/ptp.go
helpers::run_command(&format!(
```

```
"
```

```
set -ex
```

```
ip -n {nod
```

```
ip -n {nod
```

```
ip -n {nod
```

```
ip -n {nod
```

```
ip -n {net
```

```
ip -n {net
```

```
ip -n {net
```

```
ip -n {net
```

```
ip -n {net
```

```
ip -n {net
```

```
ip -n {net
```

```
ip netns exec {node_net} sysctl -w net.ipv4.conf.all.rp_filter=0
```

```
ip netns exec {node_net} sysctl -w net.ipv4.conf.{veth}.rp_filter=0
```

```
"
```



CNI

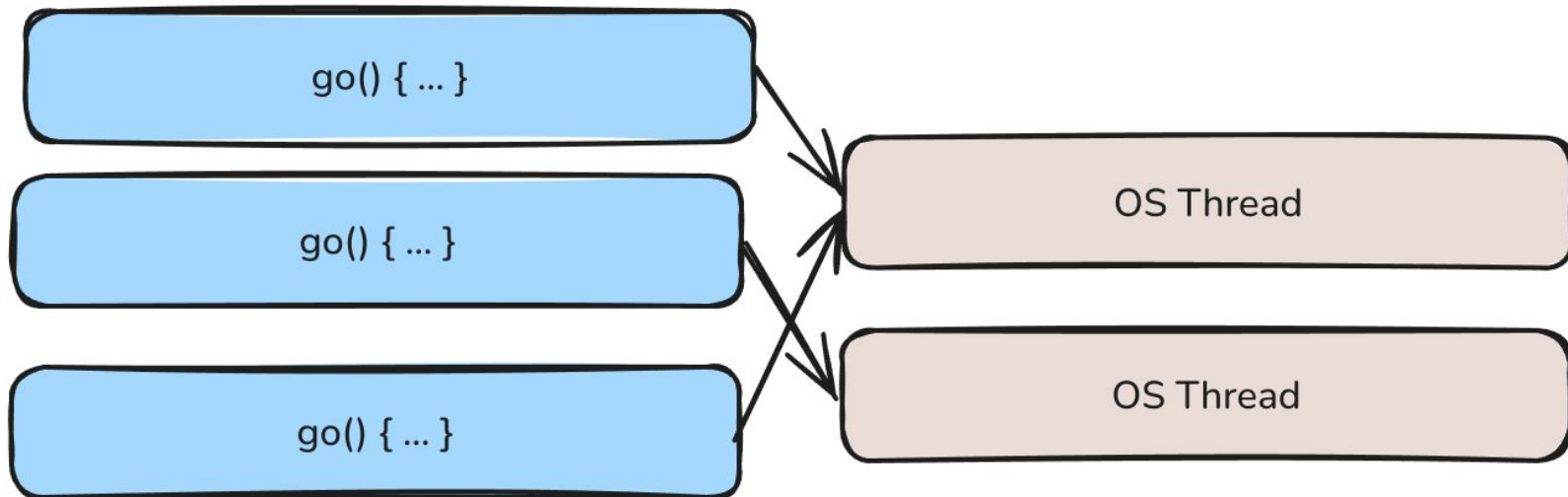
setns(2)

System Calls Manual

setns(2)

NAME `top`

`setns` - reassociate thread with a namespace



```
func NetnsDo(ns NetnsFd, f func()) {  
    runtime.LockOSThread()  
    originalNS := GetCurrentNS()  
    defer func() {  
        NetnsSet(originalNS)  
        runtime.UnlockOSThread()  
    }()  
    NetnsSet(fd)  
    # Do NOT spawn a goroutine under f()!  
    f()  
}
```

Warning: this code is *not* robust

Pinning threads: Rust



KubeCon



CloudNativeCon

North America 2024

```
async fn async_run_in_namespace(  
    namespace: Namespace,  
    f: async Fn(),  
) {  
    thread::spawn(move || {  
        run_in_namespace(namespace, || {  
            let rt = tokio::runtime::new();  
            rt.block_on(f())  
        })  
    });  
}
```

Writing tests



KubeCon




CloudNativeCon

North America 2024

```
#[tokio::test]
async fn simple_test() {
    let ztunnel = manager.deploy_ztunnel(DEFAULT_NODE).await?;
    let server = manager
        .workload_builder("server", DEFAULT_NODE)
        .register()
        .await?;
    run_tcp_server(server)?;

    let client = manager
        .workload_builder("client", DEFAULT_NODE)
        .register()
        .await?;
    run_tcp_client(client, manager.resolve("server"))?;

    // ... some assertions here
}
```



Running tests is easy!



KubeCon



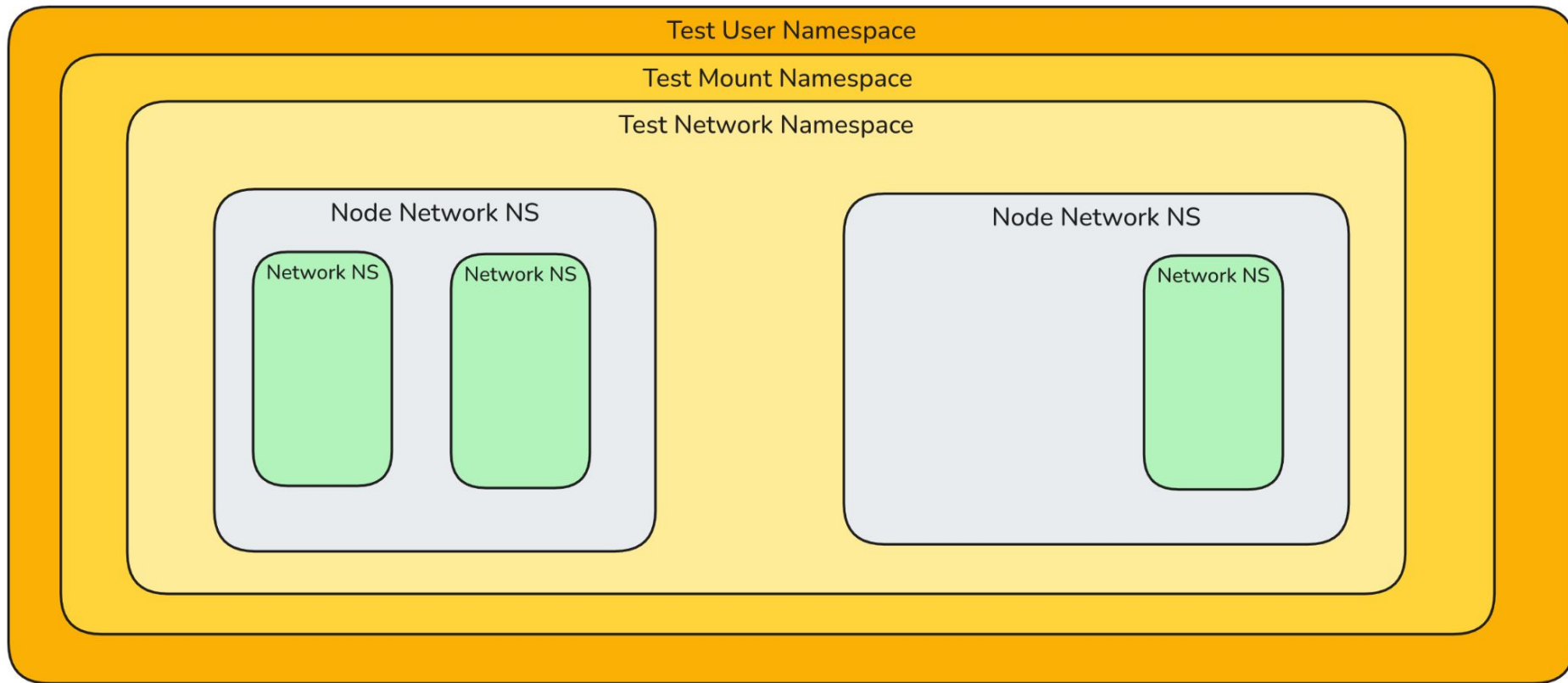
CloudNativeCon

North America 2024

```
$ cargo test
thread panicked at src/test/namespaced.rs:
write failed: { code: 1, kind: PermissionDenied,
message: "Operation not permitted" }
```

```
$ sudo cargo test
test e2e::simple ... ok
```

Dropping Root



Dropping Root

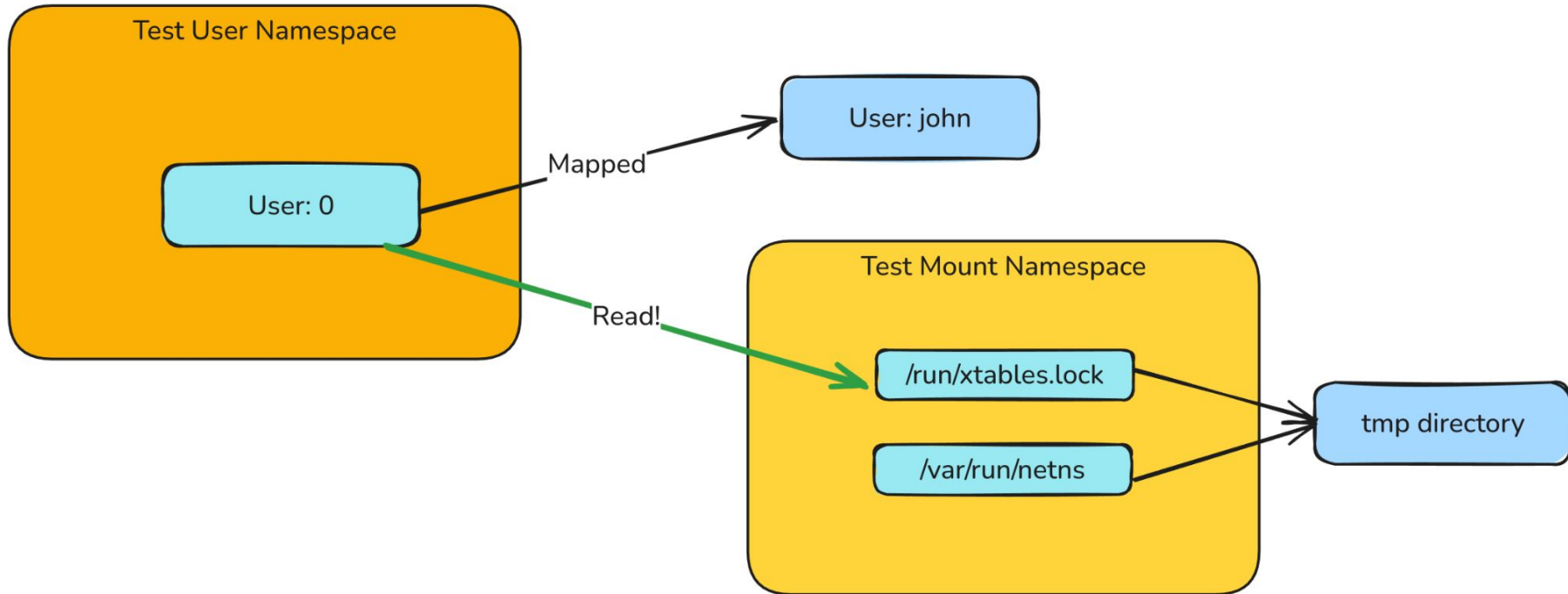


KubeCon



CloudNativeCon

North America 2024



setns(2)

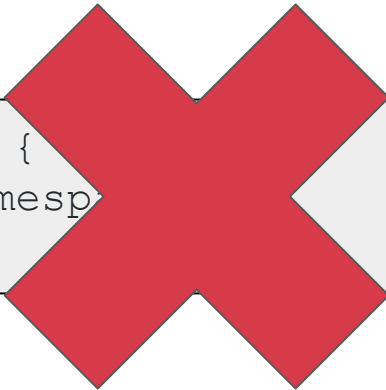
System Calls Manual

setns(2)

NAME [top](#)

setns - reassociate thread with a namespace

```
func init() {  
    setupNamesp  
}
```



Dropping Root... for good

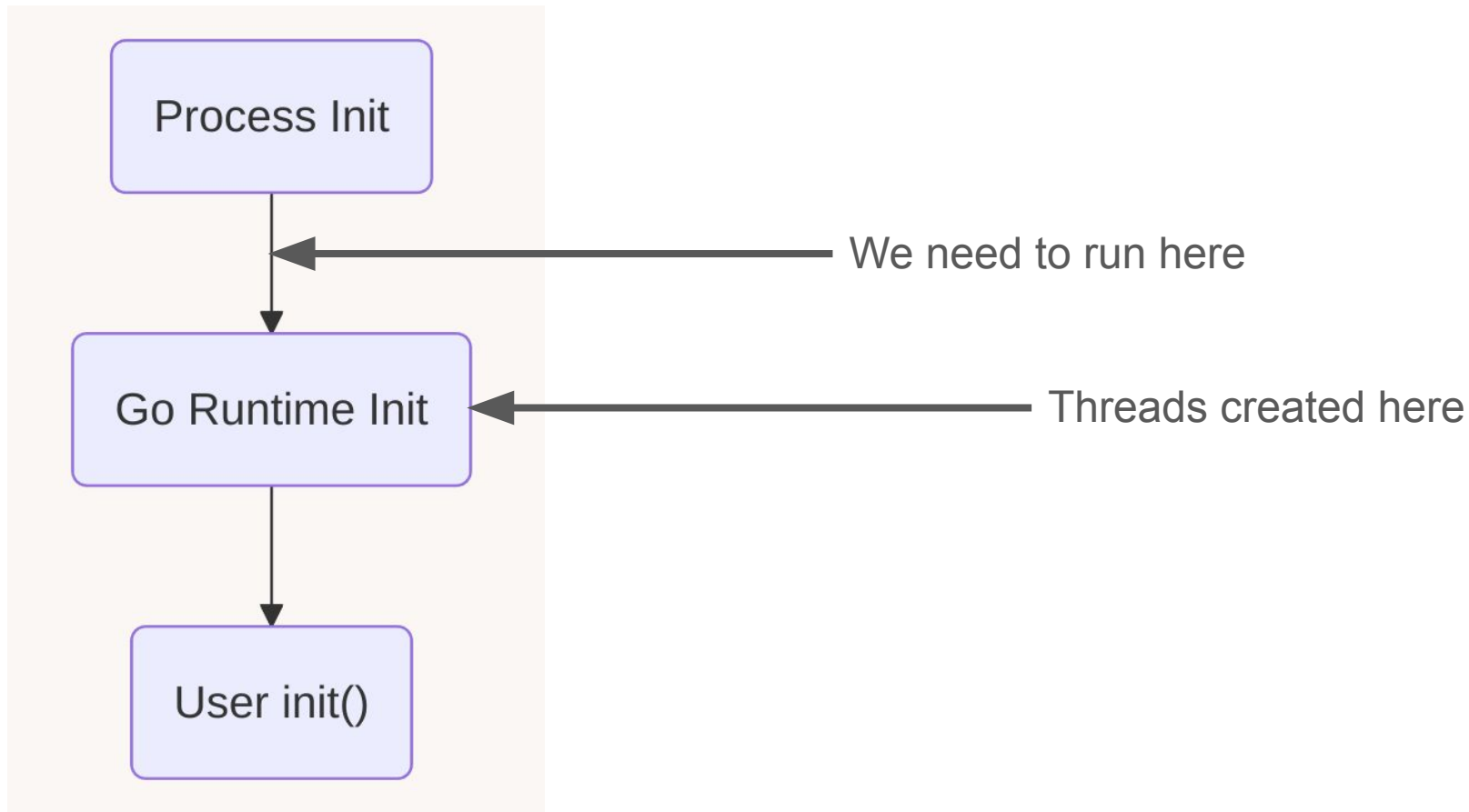


KubeCon



CloudNativeCon

North America 2024



Dropping Root... for good



KubeCon



CloudNativeCon

North America 2024

```
// Package netns makes the process enter a new network namespace.
package netns

/*
#cgo

__attribute__((constructor(100))) void enter_netns(void) {
    unshare(CLONE_NEWNET)
}
*/
import "C"
```

```
import (
    // Create a new user namespace.
    _ "github.com/howardjohn/unshare-go/usersns"
    // Create a new network namespace.
    _ "github.com/howardjohn/unshare-go/netns"
    // Create a new mount namespace.
    _ "github.com/howardjohn/unshare-go/mountns"
)
```

Dropping Root... for good (Rust)

```
#[ctor::ctor]
fn initialize_namespace_tests() {
    unshare(CloneFlags::CLONE_NEWNS);
    // ...
}
```

Results



KubeCon



CloudNativeCon

North America 2024

```
$ cargo nextest run --test namespaced
Finished `test` profile [unoptimized + debuginfo] target(s) in 0.15s
Starting 21 tests across 1 binary (run ID: ba66681c-9a81-40b3-877e-ca1e40d368ca, nextest
PASS [ 0.212s] ztunnel::namespaced namespaced::hbone_ip_mismatch
PASS [ 0.208s] ztunnel::namespaced namespaced::test_policy
PASS [ 0.230s] ztunnel::namespaced namespaced::server_uncaptured_dedicated
PASS [ 0.234s] ztunnel::namespaced namespaced::client_uncaptured_dedicated
PASS [ 0.234s] ztunnel::namespaced namespaced::malicious_calls_inpod
PASS [ 0.253s] ztunnel::namespaced namespaced::service_waypoint
PASS [ 0.272s] ztunnel::namespaced namespaced::local_captured_inpod
PASS [ 0.274s] ztunnel::namespaced namespaced::server_uncaptured_inpod ✓
PASS [ 0.285s] ztunnel::namespaced namespaced::client_uncaptured_inpod
PASS [ 0.286s] ztunnel::namespaced namespaced::service_waypoint_hbone_name ✓
PASS [ 0.289s] ztunnel::namespaced namespaced::service_waypoint_workload_hostname
PASS [ 0.300s] ztunnel::namespaced namespaced::sandwich_waypoint_plain
PASS [ 0.308s] ztunnel::namespaced namespaced::cross_node_captured_dedicated ✓
PASS [ 0.309s] ztunnel::namespaced namespaced::sandwich_waypoint_proxy_protocol
PASS [ 0.690s] ztunnel::namespaced namespaced::cross_node_captured_inpod ✓
PASS [ 0.493s] ztunnel::namespaced namespaced::test_server_shutdown
PASS [ 0.495s] ztunnel::namespaced namespaced::trust_domain_mismatch_rejected ✓
PASS [ 0.501s] ztunnel::namespaced namespaced::test_ztunnel_shutdown
PASS [ 0.529s] ztunnel::namespaced namespaced::test_prefetch_forget_certs
PASS [ 0.541s] ztunnel::namespaced namespaced::workload_waypoint
PASS [ 0.847s] ztunnel::namespaced namespaced::service_loadbalancing

-----
Summary [ 0.854s] 21 tests run: 21 passed, 0 skipped
```


A note on the runtime

```
$ hyperfine -N 'unshare --map-root-user --net'  
Benchmark 1: unshare --map-root-user --net  
Time (mean  $\pm$   $\sigma$ ):          5.7 ms  $\pm$  0.9 ms
```

```
$ stress -p 256 unshare --map-root-user --net  
2304 runs so far, 0 failures (100.00% pass rate). 1.202223414s avg, 2.361418569s max,
```



KubeCon



CloudNativeCon

North America 2024

Thank you!

John Howard (Solo.io)