

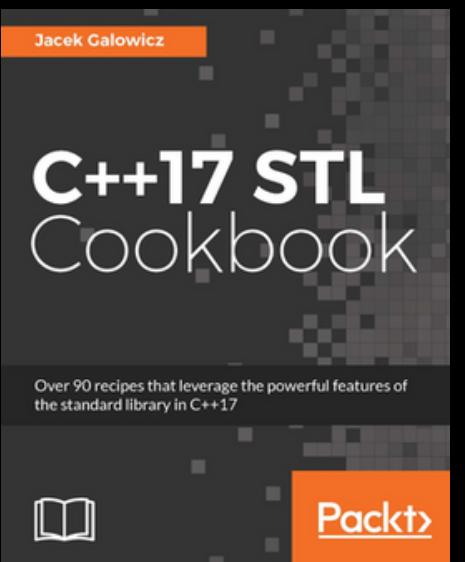
Jacek Galowicz

Mastering NixOS Integration Tests

Advanced Techniques
for Fast and Robust
Multi-VM Tests

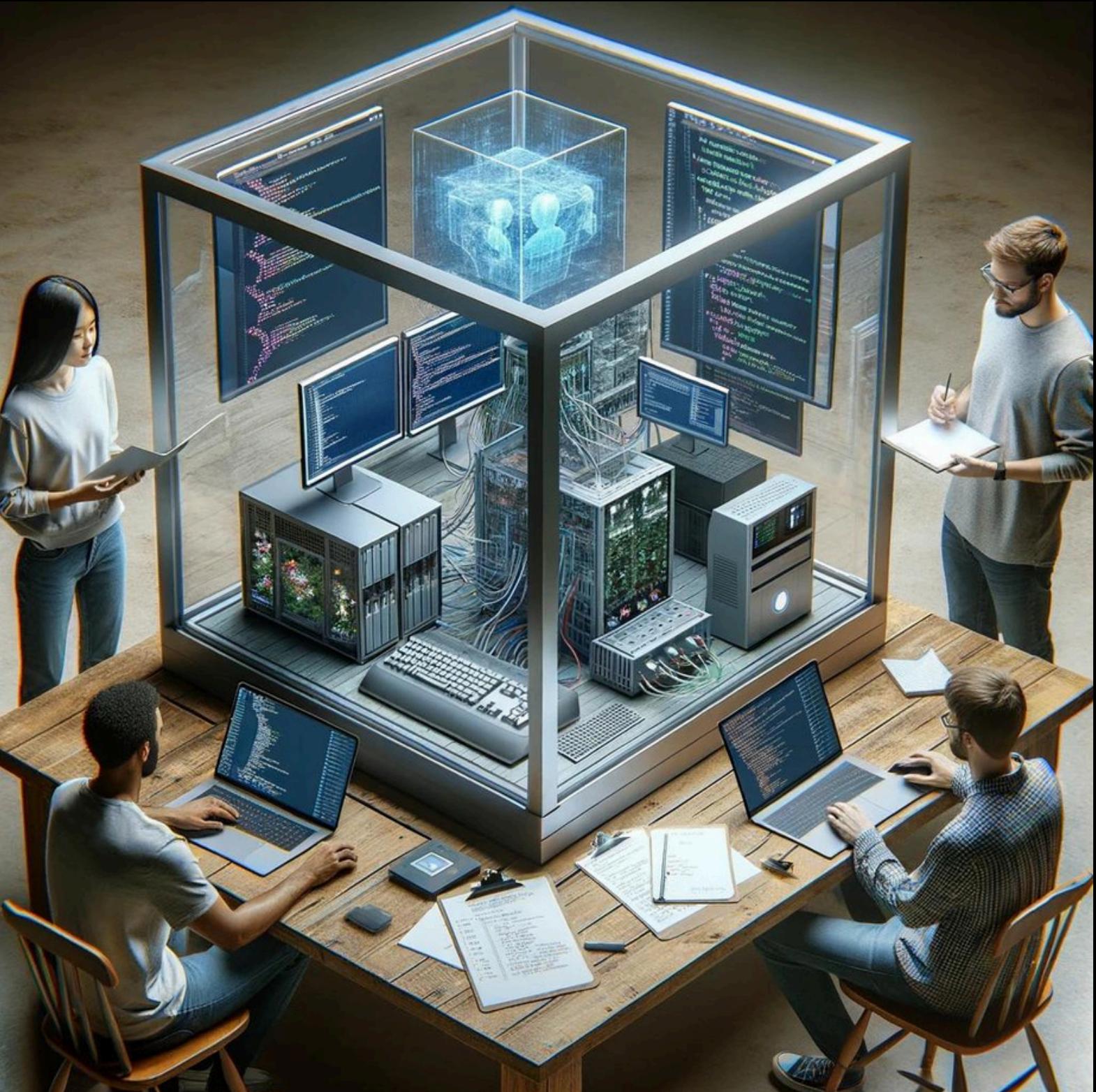
Jacek Galowicz

- NixOS Test Driver Maintainer
- Nix(OS) Consulting
- CEO of Applicative Systems, Nixcademy
 - → taught >300 professionals Nix(OS)
- Author of Nix Articles for c't



www.nixcademy.com





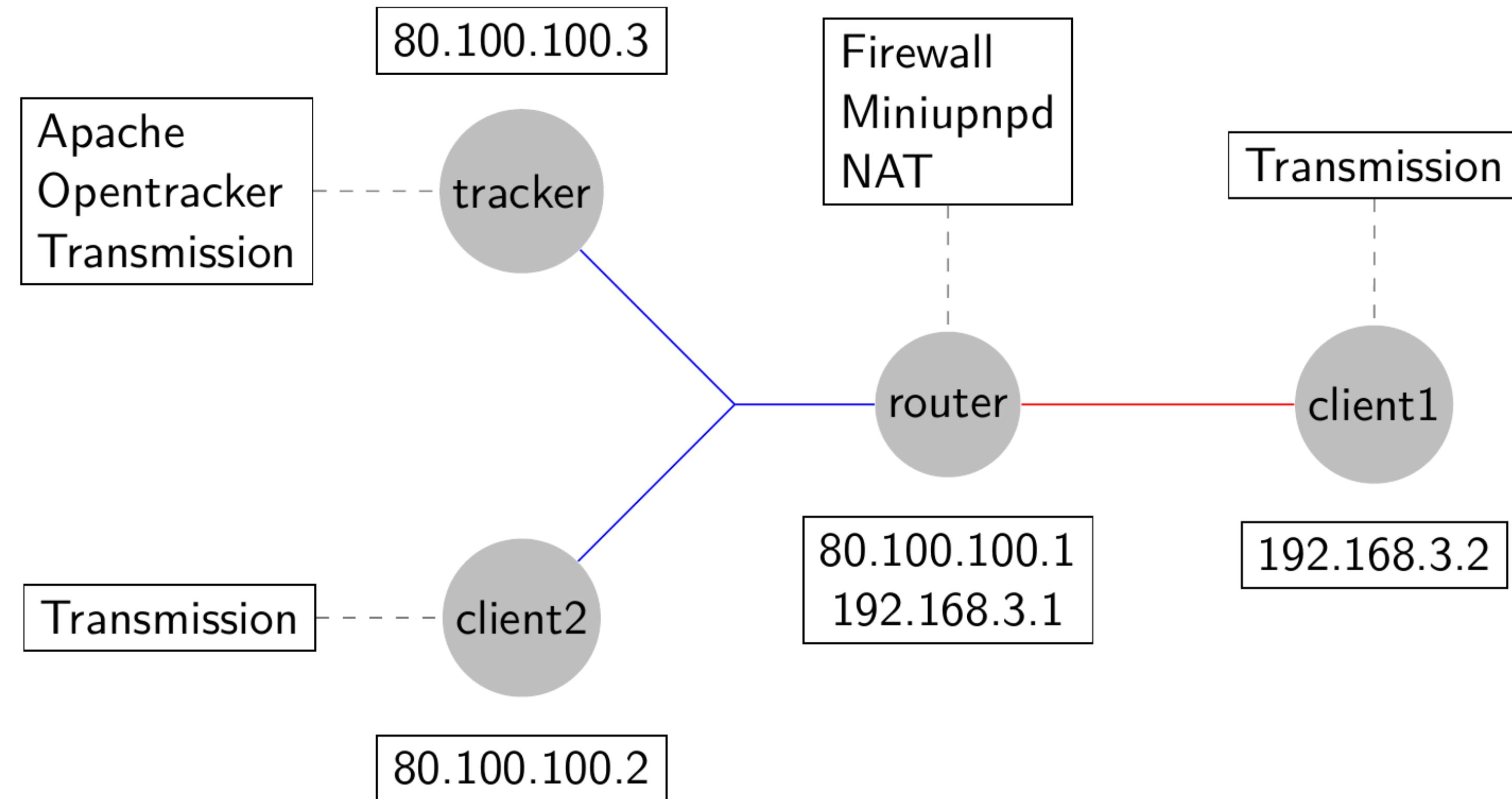
Nixpkgs contains
>1000

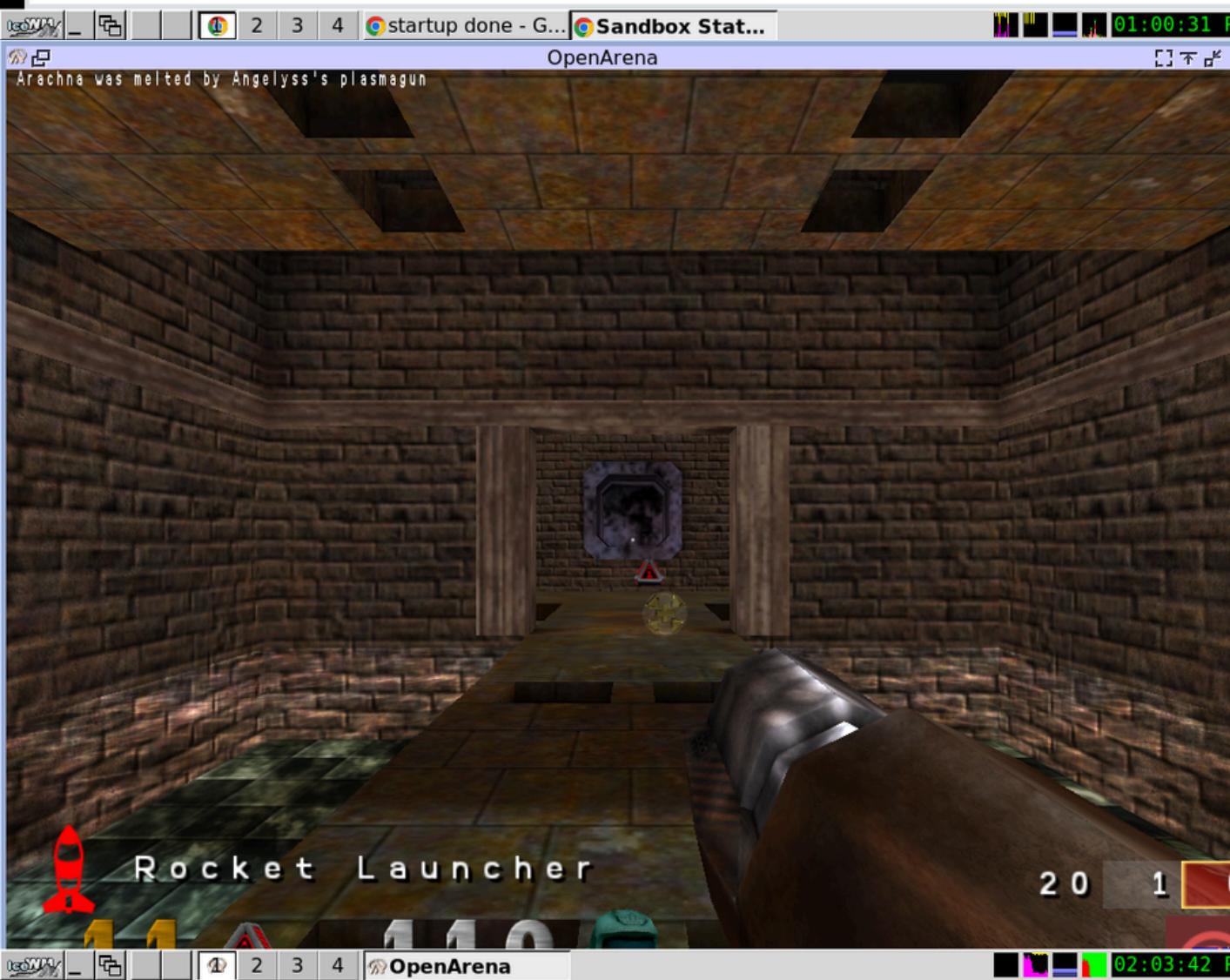
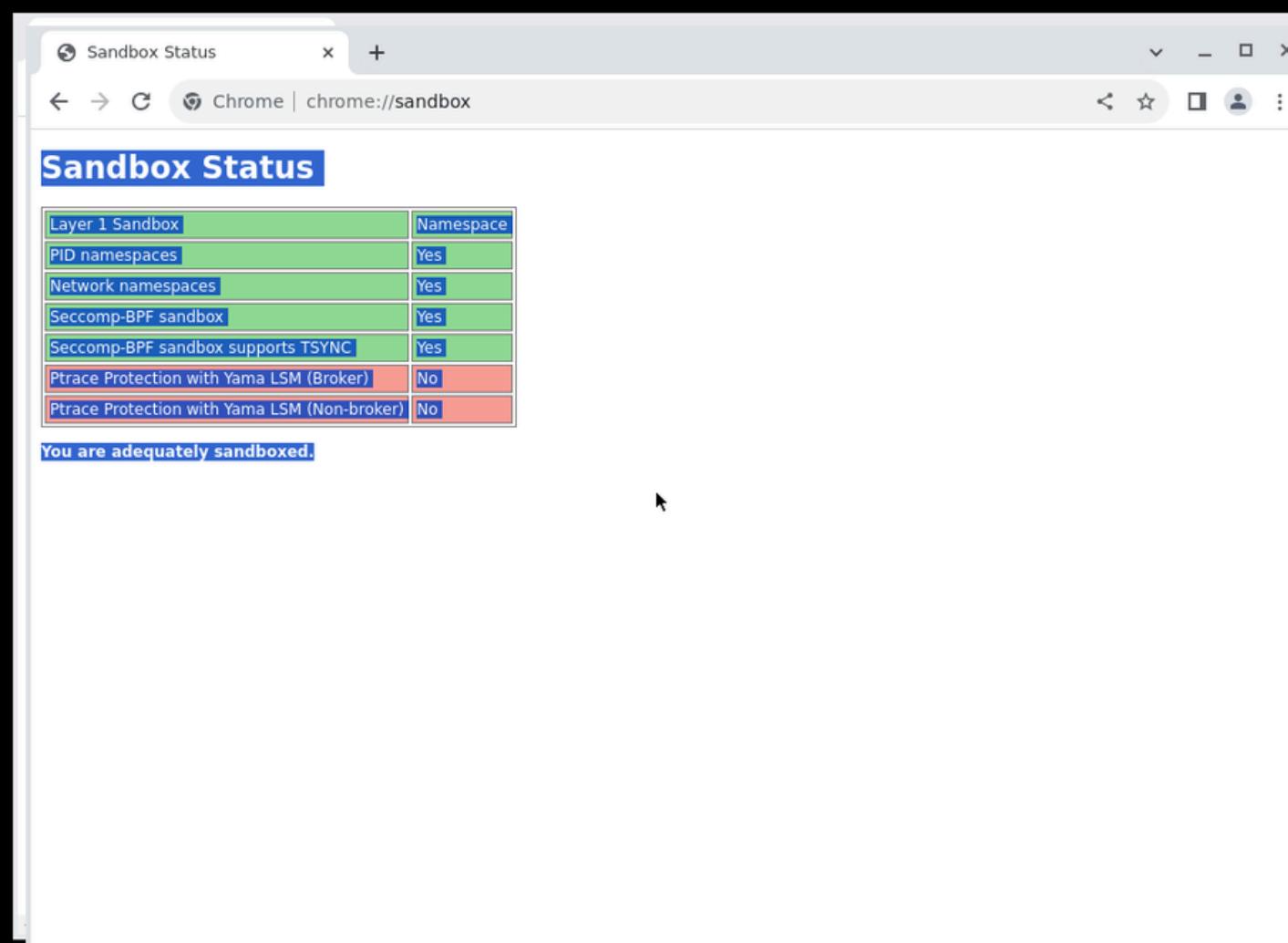
NixOS integration tests

location: nixpkgs/nixos/tests

NixOS Integration Test Example: BitTorrent Service

Complex Network&Service Definitions in ~50 LOC





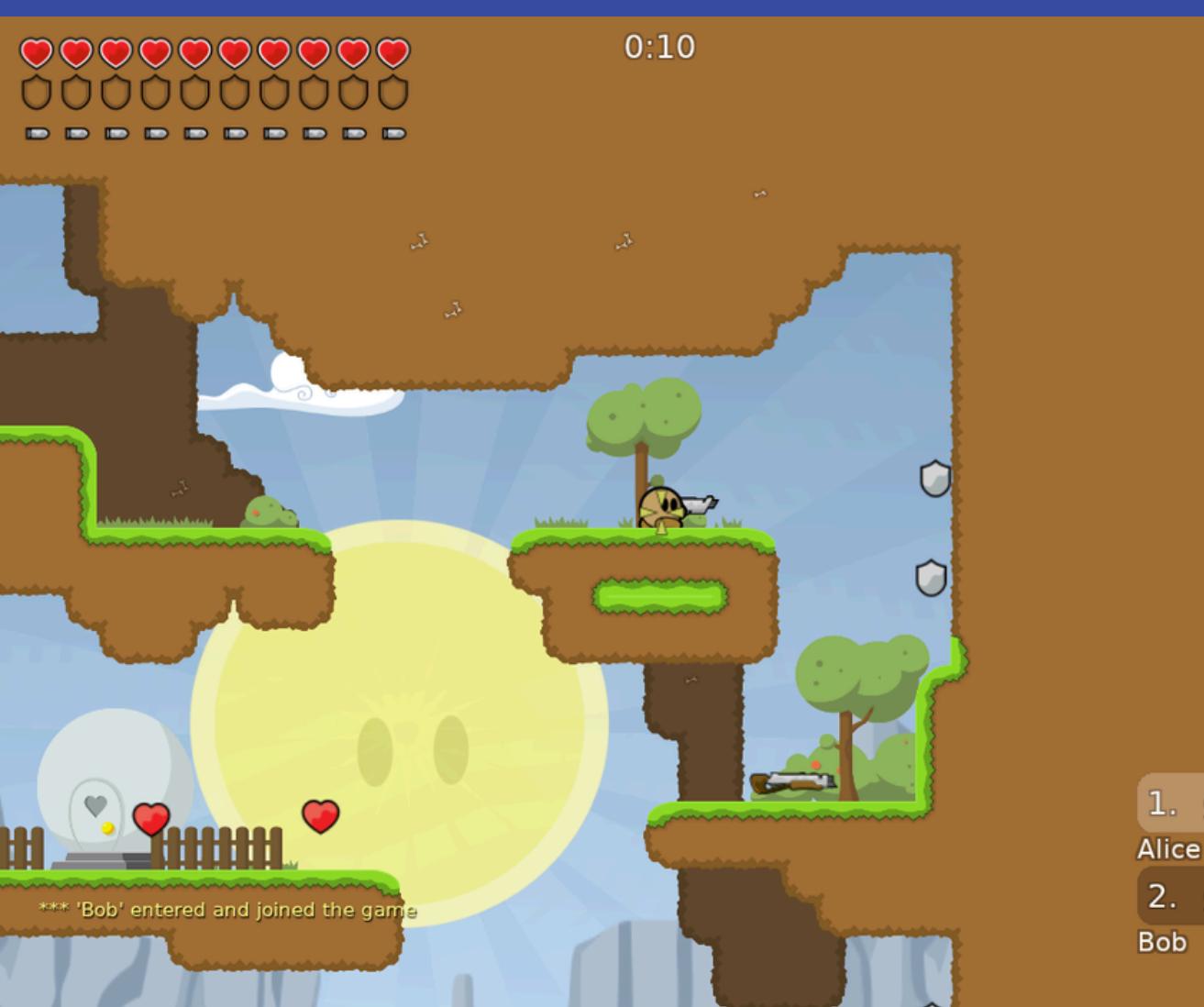
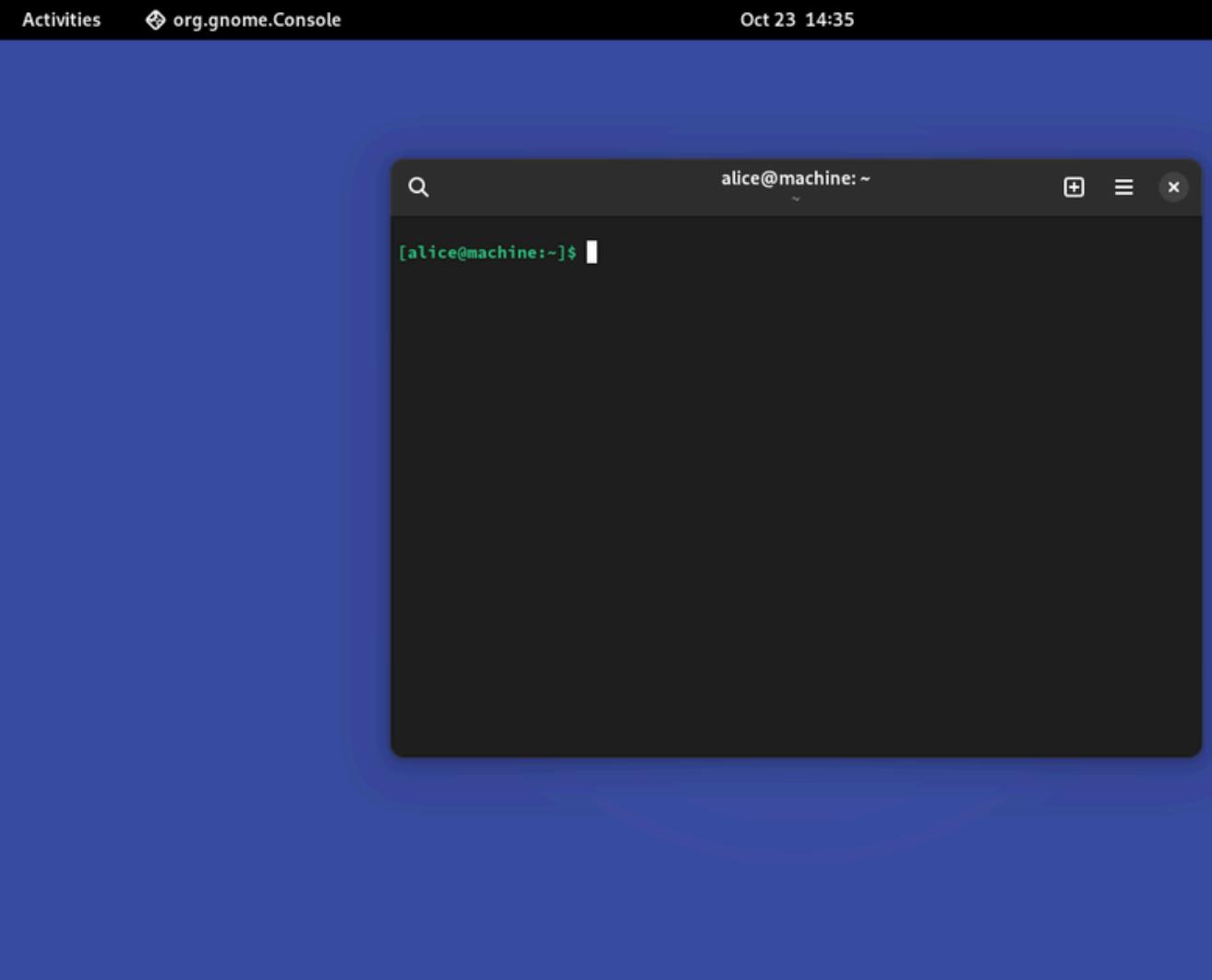
Chrome Sandbox Test

Gnome Desktop Test

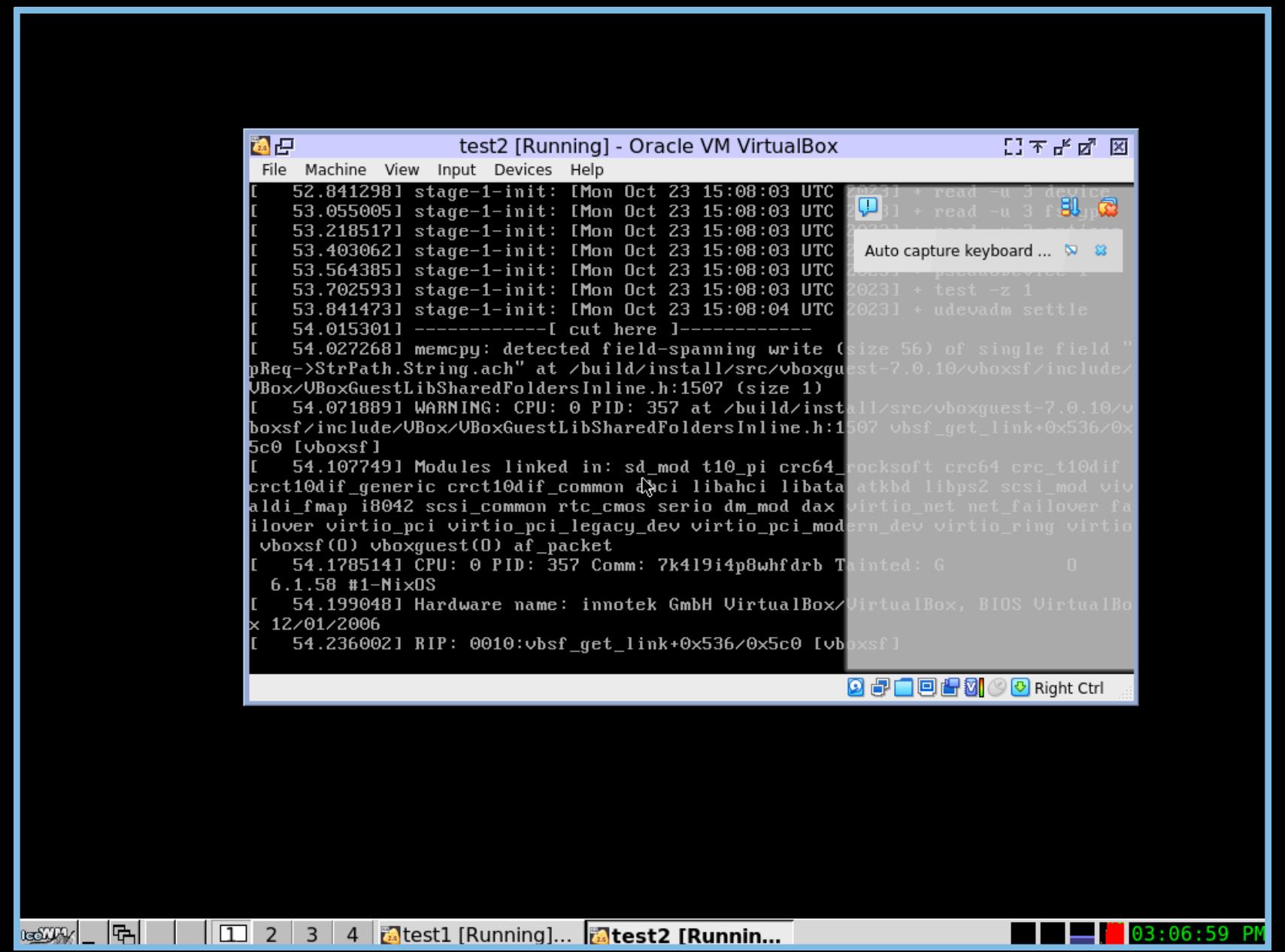
Example Tests with UI



Openarena &
Teeworlds
Tests



Tests with nested virtualization





Run NixOS Integration Tests on macOS

<https://nixcademy.com/posts/running-nixos-integration-tests-on-macos/>



Declarative VM configuration

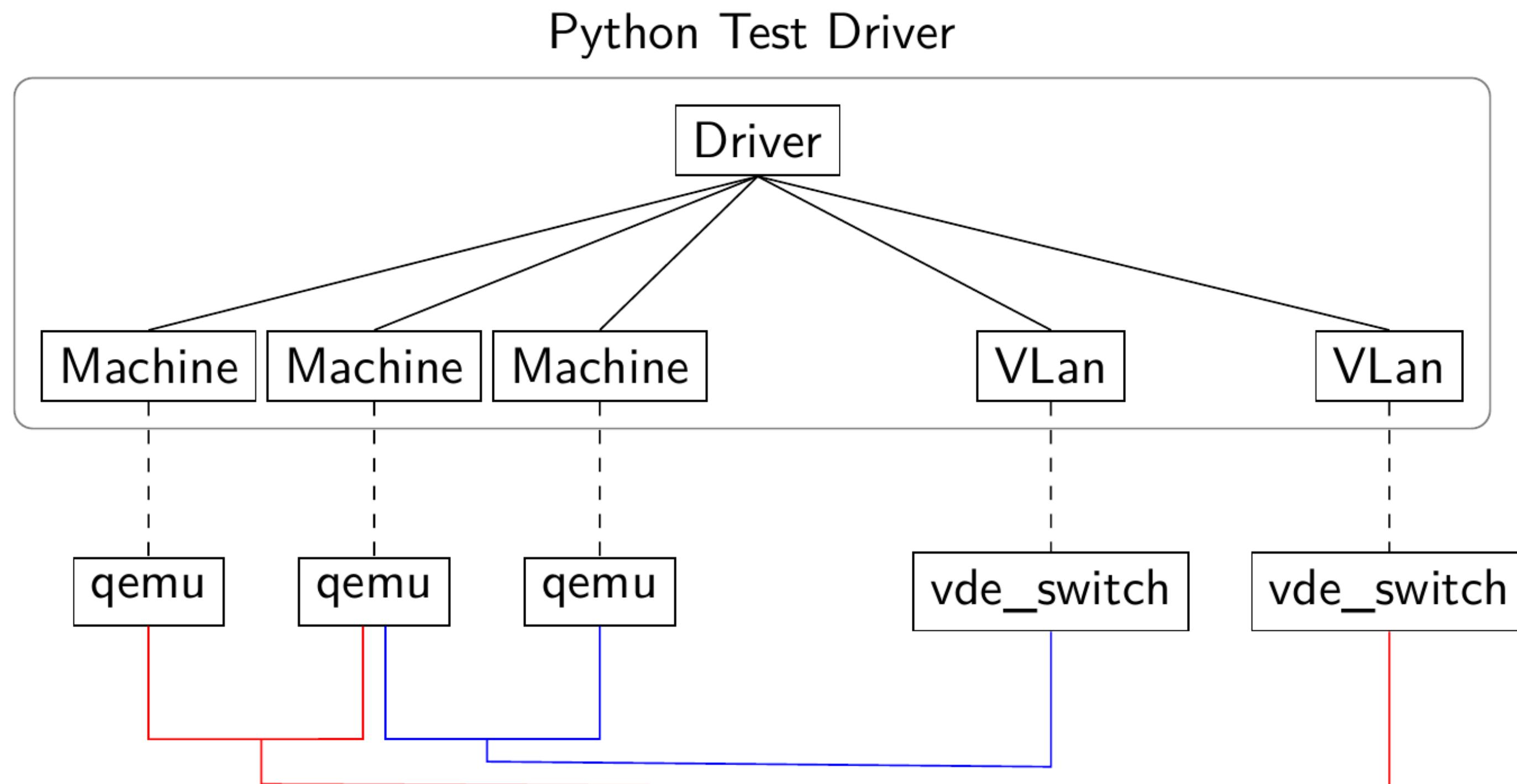
(Nix)

Imperative Test Sequence

(Python)

```
{  
    name = "nixcon24-demo-test";  
  
    defaults = {  
        networking.firewall.enable = false;  
    };  
  
    nodes = {  
        machine1 = { ... }; ← usual NixOS Configs  
        machine2 = { ... }; ←  
    };  
  
    testScript = ''  
    start_all()  
  
    for m in [ machine1, machine2 ]:  
        m.systemctl("start network-online.target")  
        m.wait_for_unit("network-online.target")  
  
        machine1.succeed("ping -c 1 machine2")  
        machine2.succeed("ping -c 1 machine1")  
    '';  
}
```

The NixOS Integration Test Driver



Python Machine Class

start, stop, run commands, disconnect from network, wait
for systemd units, wait for open network ports, save
screenshot, perform OCR, ...

```
pkgs testers runNixOSTest ./test.nix
```

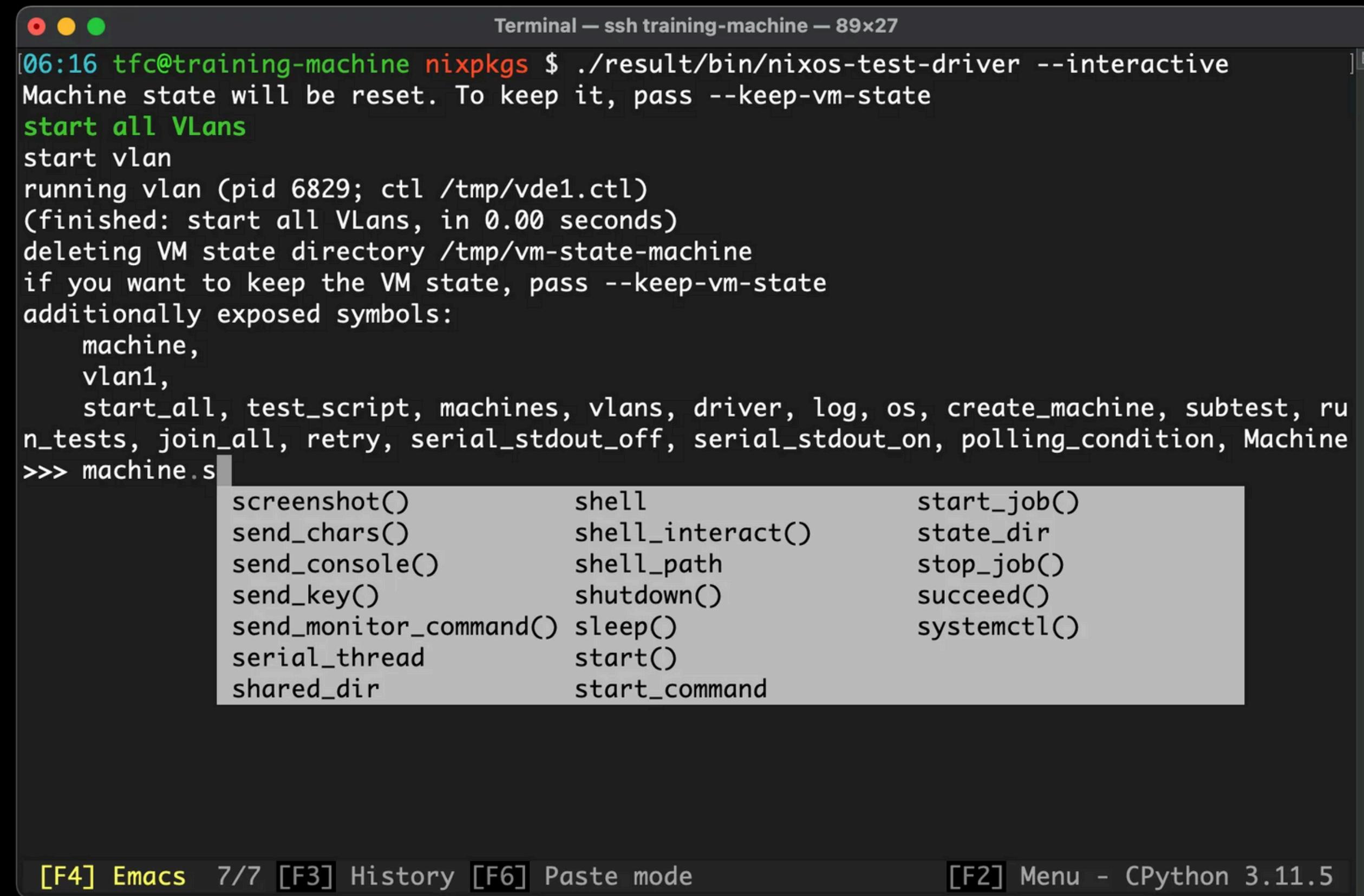
Demo

Code:

<https://github.com/applicative-systems/nixos-test-driver-nixcon24>



Interactive Mode



```
Terminal — ssh training-machine — 89x27
[06:16 tfc@training-machine nixpkgs $ ./result/bin/nixos-test-driver --interactive
Machine state will be reset. To keep it, pass --keep-vm-state
start all VLans
start vlan
running vlan (pid 6829; ctl /tmp/vde1.ctl)
(finished: start all VLans, in 0.00 seconds)
deleting VM state directory /tmp/vm-state-machine
if you want to keep the VM state, pass --keep-vm-state
additionally exposed symbols:
  machine,
  vlan1,
  start_all, test_script, machines, vlans, driver, log, os, create_machine, subtest, run_tests, join_all, retry, serial_stdout_off, serial_stdout_on, polling_condition, Machine
>>> machine.s
    screenshot()          shell                  start_job()
    send_chars()          shell_interact()       state_dir
    send_console()        shell_path            stop_job()
    send_key()            shutdown()            succeed()
    send_monitor_command() sleep()              systemctl()
    serial_thread         start()               start_command
    shared_dir
```

[F4] Emacs 7/7 [F3] History [F6] Paste mode [F2] Menu - CPython 3.11.5

Automatic Linting & Type Checking

- mypy
- pyflakes

can be disabled:

```
skipLint = true;  
skipTypeCheck = true;
```

VM Configuration

Per-VM settings:

```
virtualisation.memorySize = 10000;  
virtualisation.resolution = { x = 1024; y = 768; };  
virtualisation.cores = 8;  
virtualisation.forwardPorts = ...;
```

see all options:

[nixos/modules/virtualisation/qemu-vm.nix](#)

Documentation

NixOS docs → “NixOS Tests”

<https://nixos.org/manual/nixos/stable/#sec-nixos-tests>





Thank you for your attention!

We provide private and public trainings: shop.nixcademy.com



Subscribe our Newsletter:
biweekly NixOS
blogosphere summaries



x.com/nixcademy