Orchestrating jails with nomad and pot

A container-based cloud computing platform for FreeBSD

FOSDEM 2020 – BSD Devroom
20200202 – Bruxelles

pizzamig@FreeBSD.org
whoami(1)

- Luca Pizzamiglio aka pizzamig@
- FreeBSD enthusiast
- Port committer since August 2017
- Exploring more FreeBSD use cases
- Building packages at trivago
Orchestrating jails with nomad and pot

- Jails and pot
  - Presented here at FOSDEM in 2018
- A service mesh on FreeBSD
- pot images
- Future works
- QA
pot framework

- Journey started in 2018
- Ambitious goal: evaluate and create a new container model based on FreeBSD
- Based on jail, ZFS, pf, VNET, rctl and cpuset
- The framework is taking care of:
  - jail configuration
  - datasets management
  - network configuration and management
    - NAT
    - TCP port exposure and redirection
  - Resource limitation
Service mesh - a simple overview

[Diagram showing the components of a service mesh, including Developer, Nomad Server (orchestrator), Container images database, Nomad Clients (workers), Proxy/LB (traefik), and User.]

Orchestrating jails with nomad and pot
pot framework - features needed

• **Ability to deal with pot images**
  • create/export an image
  • load the image to a registry
  • download/import an image

• **Different paradigm**
  • Create a jail image and upload to a registry/catalog
  • Deploy the jail on available nodes
Pot Framework - KISS

- **Focus on single dataset pot**
  - easier to implement
  - that’s already a lot of complexity

- **Export a pot:**
  - Create a pot and customize it
  - Take a snapshot (`zfs snapshot`)
  - Create the image (`zfs send | xz`)

- **Import a pot:**
  - Import the image (`unxz | zfs receive`)
  - Clone the snapshot (`zfs clone`)
nomad and pot

• **nomad and consul are FreeBSD friendly**
  • Already available in the FreeBSD portstree
  • nomad has an internal architecture designed to support different type of containers
    • e.g. java, raw exec, isolated exec

• **Let’s write our own driver for nomad to interact with pot! What can possibly go wrong?**
  • Esteban Barrios developed the nomad-pot driver
  • Available on: [https://github.com/trivago/nomad-pot-driver](https://github.com/trivago/nomad-pot-driver)
  • Available also as port/package
    pkg install nomad-pot-driver
Job description

```yaml
job "example" {
  datacenters = ["dc1"]
  type = "service"
  group "example-group" {
    task "nginx-pot" {
      driver = "pot"
      service {
        tags = ["nginx-pot"]
        name = "webexample"
        port = "http"
        check {
          type = "tcp"
          name = "tcp"
          interval = "5s"
          timeout = "2s"
        }
      }
    }
    config {
      image = "https://pot-registry.zapto.org/registry/
      pot = "FBSD121-nginx"
      tag = "1.2"
      command = "nginx -g 'daemon off;''
      port_map = { http = "80" }
    }
    resources {
      cpu = 200
      memory = 128
      network {
        mbits = 10
        port "http" {}
      }
    }
  }
}
**pot framework - Unexpected obstacles**

- **Deal with an exec.start that doesn’t return**
  - `pot start` will steal your shell
  - poststart hooks not executed at all or executed when the jail is already gone

- **Containers should be non persistent**
  - `nopersist` parameter is applied as poststart

- **Jails do not cleanup themselves**
  - AKA, poststop hooks are not automatically executed
    - `jail -r` execute them!
Orchestrating jails with nomad and pot

- **Currently only two network setups supported**
  - Host (host network stack AKA inherit)
  - Public-bridge (internal virtual network based on VNET)

- **private-bridge : dedicated bridge**
  - Support for private bridges has been added to pot
  - Support for nomad is more complicated than expected
    - The driver works at task level, but the private bridge needs to be created at group level

- **alias : static IP**
  - Typical jail setup – support available in pot
  - Not available in nomad yet, it could make sense for services strictly limited to one instance (?)
minipot

- Service mesh has many components to be correctly configured
- **minipot**
  - it’s like minikube, but for FreeBSD and based on nomad and pot
  - It’s a service mesh installed on one node
  - Available at [https://github.com/pizzamig/minipot](https://github.com/pizzamig/minipot) or as package `pkg install minipot`
- **Not for production!**
- Anyone can try and play with it
- Not for production!
minipot

- Demo?
pot images

• **Creating images is a new challenge**
  • Automation
  • Reproducibility
  • Speed & size
  • Portability & usability
**pot images - flavours**

- **pot create can run provisioning scripts to improve automation**
  - Provisioning script are called flavours
  - Multiple flavours can be passed to create command
    - They will be executed in sequence

  ```
  pot create -p mypot -b 12.1 -t single -f fbsd-update \ 
  -f nginx -f slim -f nginx-cmd
  ```

- **Few flavours available out of the box**
  - `fbsd-update`: update your base system
  - `slim`: reduce the image size, deleting documentation and the toolchain

- **Problems**
  - It works only on FreeBSD
  - Restart from the beginning every time
pot images - pot machine

- **WIP:** [https://github.com/ebarriosjr/potMachine](https://github.com/ebarriosjr/potMachine)
  - Imitating docker-machine

- **potMachine allows to create and run pot images on different OS**
  - Currently MacOSX and Linux tested
  - Based on vagrant
  - Extends the commands available on pot

- **WIP²: Potfile**
  - potMachine extension
  - experimental way to specify bootstrapping similarly to Dockerfile
  - Potfiles are translated into a flavour and executed with create
pot images - the registry

• a registry is a http server with pot images

• there is no public image registry
  • I don’t want to maintain one
  • Security concerns
  • I’d keep a flavours catalog

• [https://pot-registry.zapto.org/registry](https://pot-registry.zapto.org/registry) is not docker hub
  • Usable only to run examples
  • Not for production use!

• Not for production use!
Future development

- **Image creation**
  - Size shrinking
    - Try different approaches
  - Image inheritance
  - A web site to share flavours

- **Nomad driver**
  - Heavily improve debug messages
  - Add minor features

- **pot: Many ideas, not enough time to implement them**
  - pot-oom killer
  - dual stack support (nat on IPv6??)
  - potd or jaild, as pot/jail supervisor
Thanks!

- Thanks for listening!
- Thanks to the contributors!
  - 0mp, nkfilis, grembo

https://github.com/pizzamig/pot
https://github.com/trivago/nomad-pot-driver
https://github.com/pizzamig/minipot
https://github.com/ebarriosjr/potMachine
Questions?

Any question I’m able to answer?

Please send any feedback

- Your opensource developer will be really grateful

email: pizzamig@FreeBSD.org
github: https://github.com/pizzamig
twitter: https://twitter.com/pizzamig