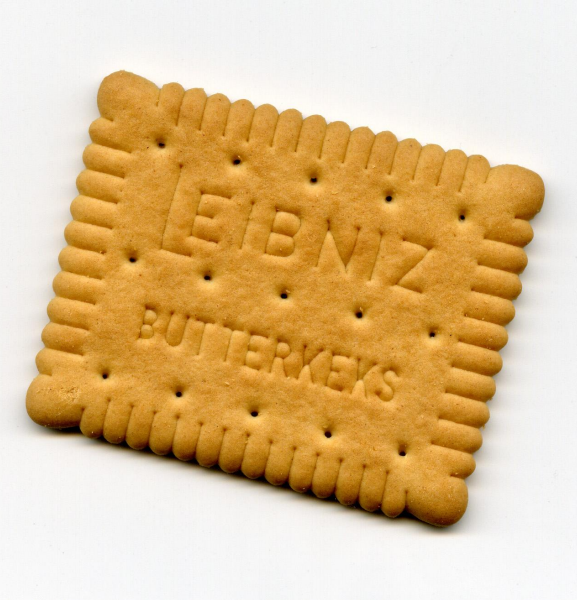
Development Environment

Based on Proxmox, Terraform, Bolt and OpenVox Container

© betadots GmbH 2024

about: me



Name: Lennart Betz

Your of Construction: 1972

Beginning in 'IT': 1985, Microsoft Basic v2

First encounter with Linux: 1993 at Leibniz University

Started to earn money with it: 1995 and 2000 in fulltime

First meeting with Puppet: 2011, professional started at 2012

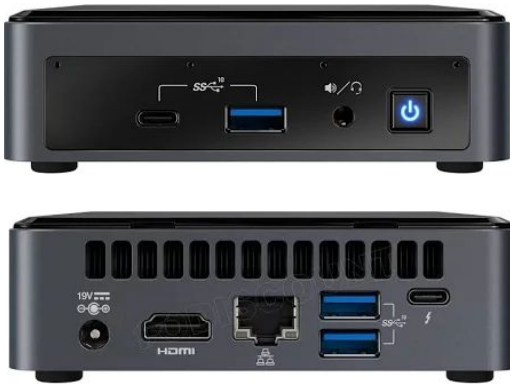
Working at betadots: since Feb 2025

© betadots GmbH 2024

Motivation

© betadots GmbH 2024

Starting Point



- most Customers use x64_86
- liked Vagrant
  - easy to deploy
  - mount local filesystems
- new MacBook with M4
  - missing boxes for ARM (build yourself as alternative)
  - no ARM support for older distributions (outdated)
- own Proxmox Cluster
  - 3x NUC10i7FNK (6 cores, 64GB)
  - Ceph via USBnet 3x 1TB SSD M.2
  - Vagrant Provider does work and is outdated

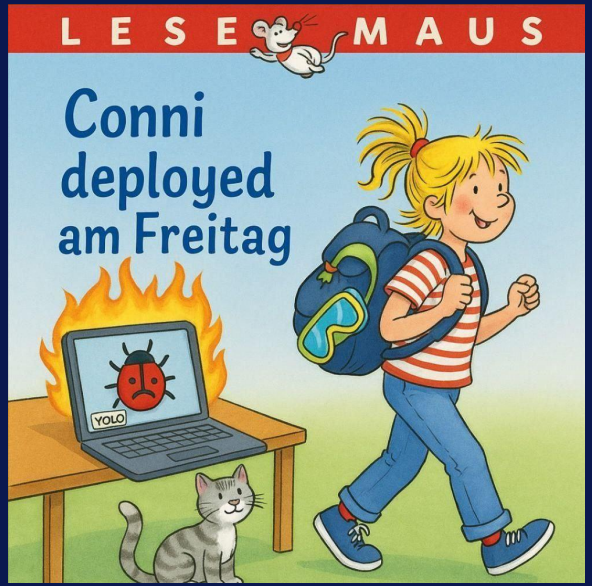
© betadots GmbH 2024

Opportunities

- fork https://github.com/telcat/vagrant-proxmox
  - less experience with such a Ruby code
  - Time to learn the Proxmox API

- linking different projects
  - Terraform knowledge (two different Proxmox provider)
  - provisioning: OpenVox Bolt
  - sshfs => no Windows, so what
  - expanded to include other clouds (like Hetzner)

© betadots GmbH 2024



Cloud Development Environment

© betadots GmbH 2024

Terraform: Virtual Machines

```
cat terraform/vms.auto.tfvars
vms = {
  "debian12" = {
    template = "debian12"
    node     = "pve02"
  }
  "ubuntu24" = {
    template = "ubuntu24"
    node     = "pve03"
  }
  "alma9" = {
    template = "alma9"
    type     = "medium"
    node     = "pve01"
  }
}
```

- provider https://registry.terraform.io/providers/bpg/proxmox/latest/docs
- self written Terraform module for VMs
  - easier to deploy from templates than LXC (Linux Container)

Template requirements:

- qemu guest agent, running
- cloud init
  - user: cloud
  - public ssh key
  - dhcp

© betadots GmbH 2024

Terraform: VM Sizing

```
cat terraform/types.auto.tfvars
types = {
  "large" = {
    "cores"  = 4
    "memory" = 4096
  },
  "medium" = {
    "cores"  = 2
    "memory" = 2048
  }
  "small" = {
    "cores"  = 1
    "memory" = 1024
  }
  "tiny" = {
    "cores"  = 1
    "memory" = 512
  }
}
```

- CPU cores
- RAM (memory)
  - ballooning depends on template
- disk size
  - also depends on template
  - default standard size (works for me)

© betadots GmbH 2024

Terraform: VM Defaults

- type: Sizing
- network: Proxmox network
- sshfs: list of Terraform data object
  - user: local logged in user
  - key_file: copied into VM
  - mounts: src/dst path
  - uses facter to determine local ip

OpenVox (bolt task via Terraform provison):

- openvox: setup OpenVox collection
- Openvox_prod_env:
  link production environment path to

```
cat terraform/defaults.auto.tfvars
vm_default = {
  type     = "small"
  network   = "openvox"
  sshfs = {
    user     = "lbetz"
    key_file = "~/.ssh/id_ed25519.terraform"
    mounts   = [{
      src = "puppetcode"
      dst = "/root/puppetcode"
    }]
  }
  openvox  = "8"
  openvox_prod_env = "/root/puppetcode"
}
```


© betadots GmbH 2024

Terraform: Networking

```
cat terraform/networks.auto.tfvars
networks = {
  openvox = {
    domain = "dev.prefork.local"
    bridge = "vxlan64"
  }
  betadots = {
    domain = "betadots.local"
    bridge = "vxlan65"
  }
}
```

- Map of networks
  - use key as reference
  - bridge: Proxmox network bridge
  - domain: defined network domain (DHCP)

- Requirement of a DHCP server
  - already a PiHole (customized dnsmasq) in use
  - needs pre defined networks

© betadots GmbH 2024

OpenVox Bolt

Uses puppetlabs/terraform module

- apply and destroy VMs
- inventory plugin

```
---
groups:
  - name: terraform
    targets:
      - _plugin: terraform
        dir: ./terraform
        resource_type: proxmox_virtual_environment_vm.this
        target_mapping:
           uri: ipv4_addresses.1.0
           name: name
```

```
config:
  transport: ssh
  ssh:
    native-ssh: true
    user: cloud
    host-key-check: false
    run-as: root
    run-as-command:
      - sudo
      - '-nksSEu'
```


© betadots GmbH 2024

Controlled by Bash Script

$ git clone https://github.com/betadots/cde.git

$ cd cde

$ ./bin/cde help

$ ./bin/cde <init|up|status|destroy|ssh> [host target] [-h] [-v]

requires:

- Terraform
- Bolt / Facter
- jq

© betadots GmbH 2024

Terraform Provisioner

© betadots GmbH 2024

Install OpenVox Agent

- Self written bolt task
- executed on local machine
- ip is used as target (inventory is not needed)
- links */etc/puppetlabs/code/environments/production* to *var.openvox_prod_env*

```
provisioner "local-exec" {
  command = "bolt task run cde::install_agent \
      collection=openvox${var.openvox} \
      version='latest' \
      stop_service=true \
      production_env=${var.openvox_prod_env} \
      --targets ${flatten(self.ipv4_addresses)[1]}"
}
```


© betadots GmbH 2024

sshfs Mounts

- Self written bolt plan
- also executed on local machine and used ip as target
- one or more src/dst tuple are passed as JSON
- local ip via facter (local.myip)

```
provisioner "file" {
  # require connection
  source      = var.sshfs.key_file
  destination = "/tmp/identity"
}

provisioner "local-exec" {
  command = var.sshfs.mounts == null ? "exit 0" : "bolt plan run cde::mount \
    targets=${flatten(proxmox_virtual_environment_vm.this.ipv4_addresses)[1]} \
    sshfs_user=${var.sshfs.user} \
    sshfs_host=${local.myip} \
    mounts='${jsonencode(var.sshfs.mounts)}'"
}
```


© betadots GmbH 2024

Extend with your own Plans or Tasks

- plan cde::dispatch to carry out further plans or task
- all map values in Terraform are limited to the same data type
  args restricted to strings!

=>

```
"puppet.openvox" = {
  hostname = "puppet"
  network  = "openvox"
  template = "debian12"
  type     = "large"
  node     = "pve03"
  sshfs    = {
    mounts   = [{
      src = "puppetcode"
      dst = "/var/opt/puppetlabs/code/environments/production"
    }]
  }
  provision = [
    {name="cde::crafty", type="plan", args={code_dir="/var/opt/puppetlabs/code"}}
  ]
  openvox_prod_env = "/var/opt/puppetlabs/code/environments/production"
}
```


© betadots GmbH 2024

Crafty

- fancy collection of container compose files
  https://github.com/voxpupuli/crafty
- for deployment of an OpenVox environment
  - OpenVox server
    https://github.com/OpenVoxProject/container-openvoxserver
  - OpenVoxDB
    https://github.com/OpenVoxProject/container-openvoxdb
  - PostgreSQL
  - Puppetboard
    https://github.com/voxpupuli/puppetboard
  - HDM (Hiera Data Manager)
    https://github.com/betadots/hdm
  - Webhook
    https://github.com/voxpupuli/container-r10k-webhook

© betadots GmbH 2024

Summary

© betadots GmbH 2024

Summary

- Maybe I should have done it with vagrant
  - more time than expected
  - HCL is a horror
  - problems with DHCP vs. cloud init
  - sshfs code dir thru VPN, OpenVox server (container) do not start
  - and … I need a VPN connection

© betadots GmbH 2024