Incus and OpenTofu Setup

Connor Horning

What Am I Using?

Server Specs:

- Dell Optiplex 3050
- 16 Gigs of RAM
- 512 GB Hard Drive
- Ubuntu Server 24.04

What is the Plan?

- I plan on installing Incus and using ansible playbooks to easily create and destroy Incus containers.
- This document will contain how I set up Incus and how to use OpenTofu to automate the deployment of Incus containers.

Part 1 (Installing Incus):

Installing Incus:

- Run the command “sudo apt update && sudo apt install wget curl git openssh-server openvswitch-switch –y"
- Run the command “sudo apt install incus”

Assigning a static IP to Server:

- Go to root user by running “sudo su -”
- CD into netplan by running “cd /etc/netplan”
- When you run “ls” you should get a file similar to the name of “50-cloud-init.yaml”

```
root@incus:/etc/netplan# ls
50-cloud-init.yaml
```

- Make a backup of this file by running “cp 50-cloud-init.yaml 50-cloud-init.yaml.bak”

```
root@incus:/etc/netplan# cp 50-cloud-init.yaml 50-cloud-init.yaml.bak
root@incus:/etc/netplan# lls
Command 'lls' not found, but there are 16 similar ones.
root@incus:/etc/netplan# ls
50-cloud-init.yaml  50-cloud-init.yaml.bak
root@incus:/etc/netplan#
```

- Make note of your network interfaces by running “ip –br –c a”

```
root@incus:/etc/netplan# ip -br -c a
lo               UNKNOWN        127.0.0.1/8 ::1/128
enp2s0           UP             10.0.0.191/24 metric 100 2601:983:827f:5f00::610f/128 2601:983:827f:5f00:529a:4cff:fe51:d664/64 fe80::529a:4cff:fe51:d664/64
tailscale0       UNKNOWN        100.86.90.125/32 fd7a:115c:a1e0::4901:5a7e/128 fe80::d5b1:68bc:2d5a:3324/64
docker0          DOWN           172.17.0.1/16
root@incus:/etc/netplan#
```

- Nano into the 50-cloud-init.yaml file and add the following configuration:

```yaml
network:
    ethernets:
        enp2s0:
            dhcp4: false
            dhcp6: false
    version: 2


    bridges:
      bridge0:
        interfaces: [enp2s0]
        addresses: [10.0.0.191/24]
        routes:
          - to: default
            via: 10.0.0.1
        nameservers:
          addresses:
            - 10.0.0.1
            - 75.75.75.75
            - 75.75.76.76
        parameters:
          stp: true
          forward-delay: 4
        dhcp4: no
```

Important:

Things you need to change include:

- Adapter name (Example: enp2s0)
- IP and Subnet (Example: 10.0.0.191/24)

- Default gateway (Example: 10.0.0.1)
- Name Servers (Examples: 10.0.0.1, 75.75.75.75, 75.75.76.76)

- Save the file by doing CTRL + O, enter, and CTRL + X.
- Test the configuration by running “netplan try” and you should get something like this:

```
root@incus:/etc/netplan# netplan try
bridge0: reverting custom parameters for bridges and bonds is not supported

Please carefully review the configuration and use 'netplan apply' directly.
root@incus:/etc/netplan#
```

- If you got an error, check the configuration that we just made.
- If you got no error, use “netplan apply” to apply the configuration.

Add your user to the Incus Admin group:

- Leave root user by running “exit”
- Run the command “sudo usermod –aG incus-admin connor” (connor being the username of your account)
- Log out and log back in for the changes to take effect.

Install ZFS and BTRFS for Storage Pools:

- Run the command “sudo apt install zfsutils-linux btrfs-progs –y"

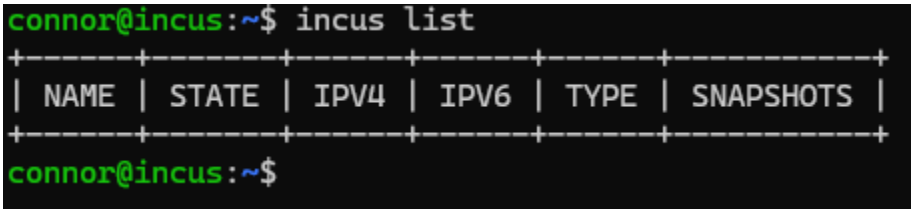
Initialize Incus Installation:

- Run “incus admin init”
- You will get asked a series of questions. You can answer these based on your requirements. Here is what I used:

```
connor@incus:~$ incus admin init
Would you like to use clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
Name of the storage backend to use (dir, lvm, lvmcluster, zfs, btrfs) [default=zfs]: btrfs
Create a new BTRFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty block device (e.g. a disk or partition)? (yes/no) [default=no]:
Size in GiB of the new loop device (1GiB minimum) [default=19GiB]: 200GiB
Would you like to create a new local network bridge? (yes/no) [default=yes]:
What should the new bridge be called? [default=incusbr0]:
What IPv4 address should be used? (CIDR subnet notation, “auto” or “none”) [default=auto]:
What IPv6 address should be used? (CIDR subnet notation, “auto” or “none”) [default=auto]:
Would you like the server to be available over the network? (yes/no) [default=no]: Yes
Address to bind to (not including port) [default=all]:
Port to bind to [default=8443]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]:
Would you like a YAML "init" preseed to be printed? (yes/no) [default=no]:
```

Listing Incus Containers:

- Run “incus list” to view any active containers running. It is blank since we haven’t made one yet.



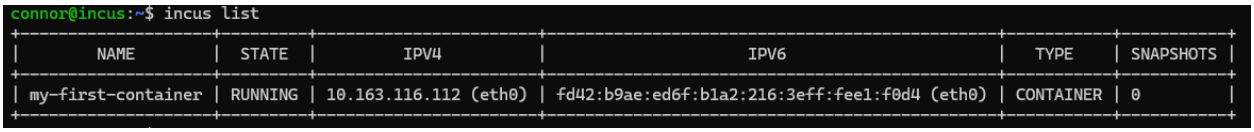
- Run “incus image list images:”
    - The output is a crazy long list of different images you can use.

Create a Bridge for our Containers:

- This bridge will allow our containers to get an IP address assigned to them.
- Run the command “incus profile create bridgeprofile”
- Next add a device connection to be used by your containers by running “incus profile device add bridgeprofile eth0 nic nictype=bridged parent=bridge0”

Launch your first Container:

- Run the command “incus launch images:ubuntu/24.04 my-first-container --profile default --profile bridgeprofile”
- Run the command “incus config device add my-first-container eth0 nic nictype=bridged parent=incusbr0 name=eth0”
- Check your new container by running “incus list”



Part 2: Using OpenTofu to Deploy and Destroy Containers

Installing OpenTofu:

- Run the command “curl -sSL https://get.opentofu.org/install.sh | sudo bash”

Install the Incus Terraform Provider

- Run the following two commands:
  - mkdir -p ~/.opentofu.d/plugins/github.com/lxc/incus/0.2.0/linux_amd64
  - curl -L https://github.com/lxc/terraform-provider-incus/releases/download/v0.2.0/terraform-provider-incus_0.2.0_linux_amd64.tar.gz | tar -xz -C ~/.opentofu.d/plugins/github.com/lxc/incus/0.2.0/linux_amd64

Preparing the Configuration Files:

- Make the directory using the command:
  - mkdir incus-tofu && cd incus-tofu
- Make a cloud-init.yaml file. Here is my configuration I used to create a user with passwordless sudo, importing my ssh keys, and connecting to my tailscale network.

```
connor@incus:~/incus-tofu$ cat cloud-init.yaml.tmpl
#cloud-config
users:
  - name: connor
    shell: /bin/bash
    groups: sudo
    sudo: ALL=(ALL) NOPASSWD:ALL

package_update: true
packages:
  - curl
  - openssh-server
  - ca-certificates

runcmd:
  # SSH Key Setup
  - mkdir -p /home/connor/.ssh
  - curl -fsSL https://github.com/ConnorH11.keys -o /home/connor/.ssh/authorized_keys_temp
  - if [ -s /home/connor/.ssh/authorized_keys_temp ]; then cat /home/connor/.ssh/authorized_keys_temp >> /home/connor/.ssh/authorized_keys; fi
  - rm -f /home/connor/.ssh/authorized_keys_temp
  - chown -R connor:connor /home/connor/.ssh
  - chmod 700 /home/connor/.ssh
  - chmod 600 /home/connor/.ssh/authorized_keys

  # SSH Service
  - systemctl enable ssh
  - systemctl restart ssh

  # Tailscale
  - curl -fsSL https://tailscale.com/install.sh | sh
  - tailscale up --authkey=${tailscale_auth_key} --hostname=${hostname}
```

- Make a main.tf file. This is the OpenTofu part of the deployment. It holds all the variables for when you run the initialization.

```
connor@incus:~/incus-tofu$ cat main.tf
terraform {
  required_providers {
    incus = {
      source  = "lxc/incus"
      version = "0.2.0"
    }
    template = {
      source  = "hashicorp/template"
      version = "2.2.0"
    }
  }
}

provider "incus" {
  remote {
    name    = "local"
    scheme  = "unix"
    address = "/var/lib/incus/unix.socket"
    default = true
  }
}

variable "tailscale_auth_key" {
  description = "Tailscale auth key"
  type        = string
  sensitive   = true
}

variable "hostname" {
  description = "Hostname for container and Tailscale"
  type        = string
  }

data "template_file" "cloudinit" {
  template = file("${path.module}/cloud-init.yaml.tmpl")
  vars = {
    tailscale_auth_key = var.tailscale_auth_key
    hostname           = var.hostname
  }
}

resource "incus_instance" "noble" {
  name    = var.hostname
  type    = "container"
  image   = "ubuntu-noble-cloud"
  profiles = ["default"]
  wait_for_network = true

  config = {
    "user.user-data" = data.template_file.cloudinit.rendered
  }
}
```

Incoporating my Tailscale Auth Key:

- To get your Tailscale authentication key, navigate to your tailscale administrative panel.
    - Click on settings
    - Under “Personal Settings” click “Keys”
    - Click “Generate auth key…”
    - This provides you with a key to copy. Important: This key will not be able to be copied after you close the pop up. Make sure to store it somewhere safe.

- Now that you have the auth key, go back to your Linux Machine in your incus-tofu directory and make a new file.
    - nano secrets.auto.tfvars
    - Type “tailscale_auth_key = “Paste your key here”

Initialize and Apply the Tofu Configuration:

- Type “tofu init”
    - If you get errors, there is something wrong with your configuration and you will want to troubleshoot that.

```
connor@incus:~/incus-tofu$ tofu init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/template from the dependency lock file
- Reusing previous version of lxc/incus from the dependency lock file
- Using previously-installed hashicorp/template v2.2.0
- Using previously-installed lxc/incus v0.2.0


  Warning: Additional provider information from registry

  The remote registry returned warnings for registry.opentofu.org/hashicorp/template:
  - This provider is deprecated. Please use the built-in template functions instead of the provider.


OpenTofu has been successfully initialized!

You may now begin working with OpenTofu. Try running "tofu plan" to see
any changes that are required for your infrastructure. All OpenTofu commands
should now work.

If you ever set or change modules or backend configuration for OpenTofu,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
connor@incus:~/incus-tofu$
```

- After it is initialized, run the command:
    - tofu apply
        - It will ask you if you want to perform the actions. Simply type “yes”

```
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  OpenTofu will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

incus_instance.noble: Creating...
incus_instance.noble: Creation complete after 7s [name=noble-cloud]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
connor@incus:~/incus-tofu$
```

Check your New Container:

- Run the command:
    - Incus list

```
connor@incus:~/incus-tofu$ incus list
+-------------+---------+-------------------------+-----------------------------------------------+-----------+-----------+
|    NAME     |  STATE  |          IPV4           |                     IPV6                      |   TYPE    | SNAPSHOTS |
+-------------+---------+-------------------------+-----------------------------------------------+-----------+-----------+
| noble-cloud | RUNNING | 100.68.53.20 (tailscale0) | fd7a:115c:a1e0::4701:3515 (tailscale0)        | CONTAINER | 0         |
|             |         | 10.163.116.52 (eth0)      | fd42:b9ae:ed6f:b1a2:216:3eff:feb7:f608 (eth0) |           |           |
+-------------+---------+-------------------------+-----------------------------------------------+-----------+-----------+
```

- As you can see, it has two IPv4 addresses, one being the LAN and the other being the tailscale network. This shows that tailscale has connected.
- Ssh into the container with “ssh connor@10.163.116.52”

```
connor@incus:~/incus-tofu$ ssh connor@10.163.116.52
The authenticity of host '10.163.116.52 (10.163.116.52)' can't be established.
ED25519 key fingerprint is SHA256:1a5KkPSIvOLcK5Np+K0ZzBbOVqs1Femt82OFE9NF+bM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.163.116.52' (ED25519) to the list of known hosts.
Enter passphrase for key '/home/connor/.ssh/id_ed25519':
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-59-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

connor@noble-cloud:~$
```

- We have successfully deployed an incus container with cloud-init and opentofu!

Access my configs on GitHub:

https://github.com/ConnorH11/incus-tofu-automation#