

# Ixd

Типы инстансов: <https://github.com/dustinkirkland/instance-type>

Мастхэв:

```
t1.micro:
  cpu: 1.0
  mem: 0.613
t2.xlarge:
  cpu: 8.0
  mem: 32.0
t2.large:
  cpu: 2.0
  mem: 8.0
t2.medium:
  cpu: 2.0
  mem: 4.0
t2.micro:
  cpu: 1.0
  mem: 1.0
t2.nano:
  cpu: 1.0
  mem: 0.5
t2.small:
  cpu: 1.0
  mem: 2.0
t2.xlarge:
  cpu: 4.0
  mem: 16.0
t3.2xlarge:
  cpu: 8.0
  mem: 32.0
t3.large:
  cpu: 2.0
```

```
mem: 8.0
t3.medium:
  cpu: 2.0
  mem: 4.0
t3.micro:
  cpu: 2.0
  mem: 1.0
t3.nano:
  cpu: 2.0
  mem: 0.5
t3.small:
  cpu: 2.0
  mem: 2.0
t3.xlarge:
  cpu: 4.0
  mem: 16.0
```

## Launch a VM that boots from an ISO

To launch a VM that boots from an ISO, you must first create a VM. Let's assume that we want to create a VM and install it from the ISO image. In this scenario, use the following command to create an empty VM:

```
lxc init iso-vm --empty --vm
```

The second step is to import an ISO image that can later be attached to the VM as a storage volume:

```
lxc storage volume import <pool> <path-to-image.iso> iso-volume --type=iso
```

Lastly, you need to attach the custom ISO volume to the VM using the following command:

```
lxc config device add iso-vm iso-volume disk pool=<pool> source=iso-volume boot.priority=10
```

The `boot.priority` configuration key ensures that the VM will boot from the ISO first. Start the VM and connect to the console as there might be a menu you need to interact with:

```
lxc start iso-vm --console
```

Once you're done in the serial console, you need to disconnect from the console using `ctrl+a-q`, and connect to the VGA console using the following command:

```
lxc console iso-vm --type=vga
```

You should now see the installer. After the installation is done, you need to detach the custom ISO volume:

```
lxc storage volume detach <pool> iso-volume iso-vm
```

Now the VM can be rebooted, and it will boot from disk.

## Подключить физический диск с хоста к виртуалке

```
lxc config device add vm1 disk1 disk source=/dev/sda
```

## Запуск FreeBSD/OPNSense/pfSense

Добавь в конфиг виртуалки:

```
config:
  raw.qemu: |
    -cpu host
  raw.qemu.conf: |
    [device "dev-qemu_rng"]
```

## Удаленное управление и вебморда

```
sudo snap set lxd ui.enable=true
sudo systemctl reload snap.lxd.daemon
lxc config set core.https_address :8443
lxc config trust add
```

## Проброс порта с виртуалки во внешку

[https://documentation.ubuntu.com/lxd/en/latest/reference/devices\\_proxy](https://documentation.ubuntu.com/lxd/en/latest/reference/devices_proxy)

```
lxc config device set mature-kangaroo eth0 ipv4.address=10.117.170.194
lxc config device add mature-kangaroo ssh-forward proxy listen=tcp:8.8.8.8:2222
connect=tcp:10.117.170.194:22 nat=true
```

Но можно и так

```
root@example:~# lxc network forward port add lxdbr0 IP-HOST tcp PORT-HOST IP-VM PORT-VM
```

## Random

```
# images list
incus image list images: architecture=x86_64 type=virtual-machine

# error
root@huanan:~# incus launch images:almalinux/9 alma1 --vm -c limits.cpu=2 -c limits.memory=2GiB -d
root,size=20GiB
Launching alma1
Error: Failed instance creation: Failed creating instance record: Add instance info to the database: This
"instances" entry already exists
root@huanan:~# incus config device add alma1 agent disk source=agent:config
Device agent added to alma1
```

## additional disks

```
lxc storage volume create hdd-storage iscsi-for-vm-export size=3000GiB --type=block

lxc config device add ftp iscsi-for-vm-export disk pool=hdd-storage source=iscsi-for-vm-export
```

# road to cluster storage

```
# sdb - iscsi disk
# os - ubuntu 24

iscsiadm -m discovery -t st -p X.X.X.X
iscsiadm -m node --targetname iqn.2023-09.com.example:stor.tgt2-incus --portal X.X.X.X --login
apt install lvm2-lockd dlm-controld

vim /etc/lvm/lvm.conf:
global {
    use_lvmlockd = 1
    locking_type = 1
}
systemctl status lvmlockd.service lvmlocks.service dlm.service
vgcreate --shared vg0 /dev/sdb
vgchange --lockstart
incus storage create pool1 lvmcluster source=vg0
```

## admin init

```
root@host1:~# incus admin init
Would you like to use clustering? (yes/no) [default=no]: yes
What IP address or DNS name should be used to reach this server? [default=10.146.134.217]:
Are you joining an existing cluster? (yes/no) [default=no]:
What member name should be used to identify this server in the cluster? [default=host1]:
Do you want to configure a new local storage pool? (yes/no) [default=yes]: no
Do you want to configure a new remote storage pool? (yes/no) [default=no]: yes
Create a new LVMCLUSTER pool? (yes/no) [default=yes]: yes
Name of the shared LVM volume group: vg0
Would you like to use an existing bridge or host interface? (yes/no) [default=no]: yes
Name of the existing bridge or host interface: enp5s0
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]: yes
config:
  core.https_address: 10.146.134.217:8443
networks: []
storage_pools:
- config:
  source: vg0
```

```
description: ""
name: remote
driver: lvmcluster
profiles:
- config: {}
  description: ""
  devices:
    eth0:
      name: eth0
      nictype: macvlan
      parent: enp5s0
      type: nic
    root:
      path: /
      pool: remote
      type: disk
  name: default
projects: []
cluster:
  server_name: host1
  enabled: true
  member_config: []
  cluster_address: ""
  cluster_certificate: ""
  server_address: ""
  cluster_token: ""
  cluster_certificate_path: ""
```

well, thats it. Теперь можно создавать тут виртуалки. Сторадж - remote.

Добавляем остальных челиков

```
# host1
incus cluster add 10.146.134.213
# host2

root@host2:~# cat boot | incus admin init --preseed
root@host2:~# cat boot
cluster:
  enabled: true
```

```

server_address: 10.146.134.213:8443 # адрес host2, который добавляем к кластеру
cluster_token: eyJzZXJ2ZXJf...yNzY1OVoifQ==
member_config:
- entity: storage-pool
  name: default
  key: source
  value: ""
- entity: storage-pool
  name: remote
  key: source
  value: "vg0"
- entity: storage-pool
  name: remote
  key: driver
  value: "lvmcluster"

```

кластеризация работает!

```

root@host2:~# incus ls
+-----+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS | LOCATION |
+-----+-----+-----+-----+-----+-----+-----+
| u2   | RUNNING | 10.146.134.93 (eth0) | fd42:6c9a:e05f:9e91:216:3eff:fe55:4480 (eth0) | CONTAINER | 0 |
host1 |
+-----+-----+-----+-----+-----+-----+-----+

root@host2:~# incus shell u2
root@u2:~# hostname
u2

```

Аналогичным образом добавляем третий хост. Как я понимаю, кворума нет в incus, поэтому достаточно иметь 2+ сервера в кластере.

```

root@host1:~# incus cluster list
+-----+-----+-----+-----+-----+-----+-----+
| NAME | URL | ROLES | ARCHITECTURE | FAILURE DOMAIN | DESCRIPTION | STATUS |
| MESSAGE |
+-----+-----+-----+-----+-----+-----+-----+
| 10.146.134.213 | https://10.146.134.213:8443 | database-standby | x86_64 | default | | ONLINE |
| Fully operational |
+-----+-----+-----+-----+-----+-----+-----+

```

```
| host1      | https://10.146.134.217:8443 | database-leader | x86_64    | default    |      | ONLINE |
```

```
Fully operational |
```

```
|           |           | database      |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

```
...
```

```
root@host1:~# incus cluster list
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
|  NAME  |    URL    |  ROLES  | ARCHITECTURE | FAILURE DOMAIN | DESCRIPTION | STATUS |
| MESSAGE |
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| 10.146.134.213 | https://10.146.134.213:8443 | database      | x86_64    | default    |      | ONLINE |
```

```
Fully operational |
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| host1    | https://10.146.134.217:8443 | database-leader | x86_64    | default    |      | ONLINE | Fully
operational |
```

```
|           |           | database      |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

```
| host3    | https://10.146.134.183:8443 | database      | x86_64    | default    |      | ONLINE | Fully
operational |
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Revision #20

Created 8 December 2023 20:34:36 by Ivan

Updated 15 September 2024 22:54:14 by Ivan