



MetaContainer.org

Share access to your files/remote desktop/virtual machines easily and securely.



Sharing ANY kind of resource should be easy and secure

```
user@host1$ metac fs export local:/home/user/shared  
ref:Aqe9ZukUXZe84Vgs2Rpj70EUXqBlJ6xK-yb1DKHSDxeYHzW5S3oHatOG<- this is the reference  
  
user@host2$ metac fs bind /home/user/shared-from-host1 ref:Aqe9ZukUXZe84Vg  
user@host2$
```



Sharing ANY kind of resource should be easy and secure

Remote desktop

```
user@host1$ metac desktop export localx11:
```

Ethernet network

```
user@host2$ metac net export local:eth0
```

Speaker/microphone

```
user@host2$ metac sound export  
local:alsa_output.pci-0000_00_1f.3.analog-stereo
```

USB device *(not implemented yet)*

```
user@host$ metac usb export local:058f:9540
```

Ideas: video (webcam, desktop stream, monitor), printer, serial port...

<http://metacontainer.org>



SSHFS, PulseAudio, QEMU, TigerVNC, USB-IP, ...

MetaContainer (references, persistence support)

encrypted overlay network



Virtual machines with remote resources

Virtual machines

```
user@host1$ metac vm start --drive uri=ref:OyPZV...V3166
                    --network uri=local:eth0
```

(Launch virtual machine with drive residing on another host)

Containers

```
user@host1$ metac net export local:br0
ref:0wVG47...3L55U
```

```
user@host2$ metac run --mount=/*/home/michal/mychroot
                    --network uri=ref:0wVG47...3L55U
                    --pty bash
```

(Launch container with access to network shared by another host)



Containers with remote resources are useful

Run on host with GPU:

```
user@host1$ metac run --host=s:mygpu  
--mount=/project=. --mount=/=/ --chdir=/project  
python train_neural_network.py
```

Technologies



Nix



**CAP'N
PROTO**
cerealization protocol

SSHFS, 9PFS, PulseAudio, TigerVNC, QEMU, USB-IP, ...



Thanks for attention and call to action

- <https://github.com/zielmicha/metac>
- <https://metacontainer.org>