

Virtual Training Platform

Technical Design



These materials are licensed under the Creative Commons Attribution-NonCommercial 4.0 International license
(<http://creativecommons.org/licenses/by-nc/4.0/>)



UNIVERSITY OF OREGON

Last updated 6th August 2026



Components of the VTP

- Understanding this may help you when *setting up or managing* the server
- The details aren't necessary for an instructor *using* the server



Base: Ubuntu Server OS



Ubuntu

Desktop GUI is not required



UNIVERSITY OF OREGON



Container management: incus

Incus containers are like "lightweight VMs"

You can manage them just like VMs
(e.g. enter shell, install packages etc)



Containers

gns3 lab1

gns3 lab2

vtp-proxy

incus

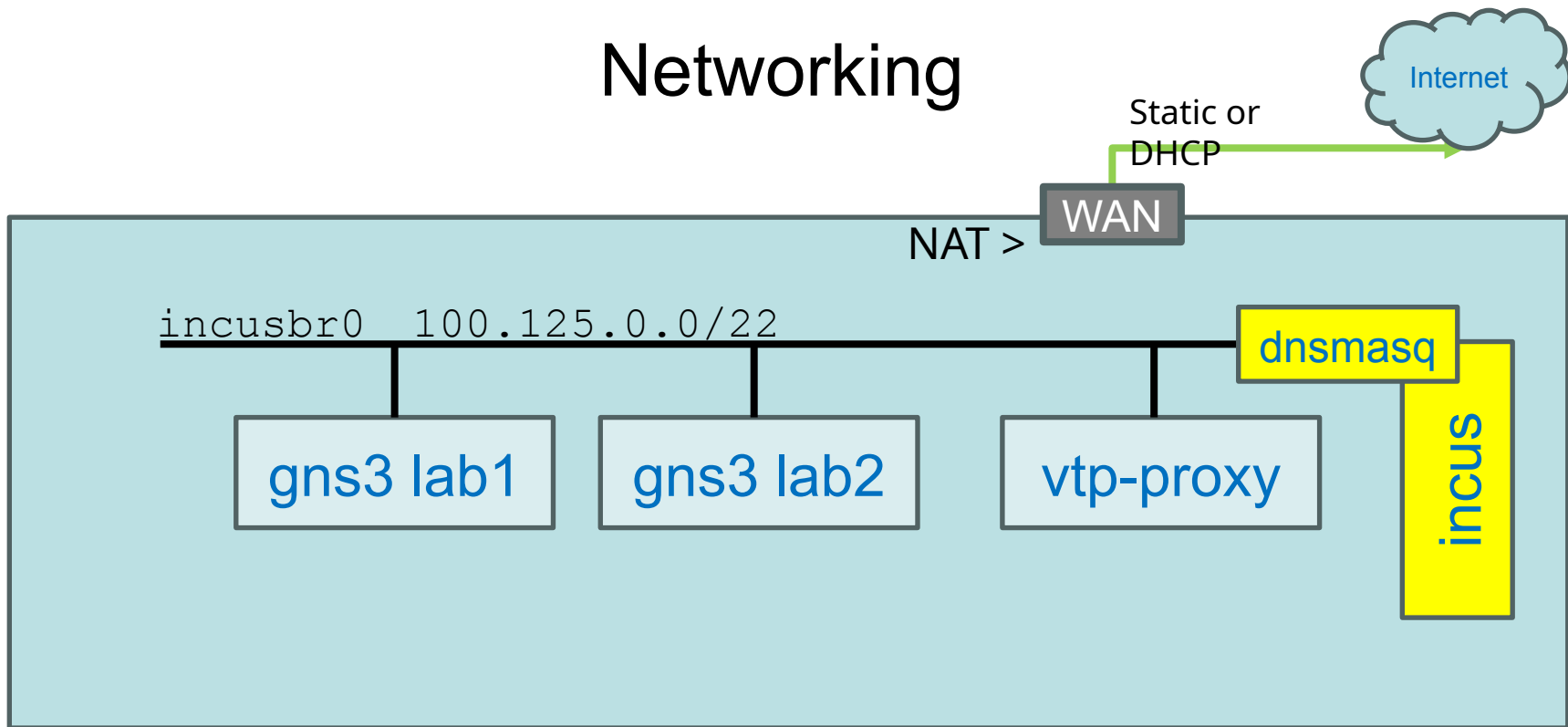
We publish ready-to-run container images



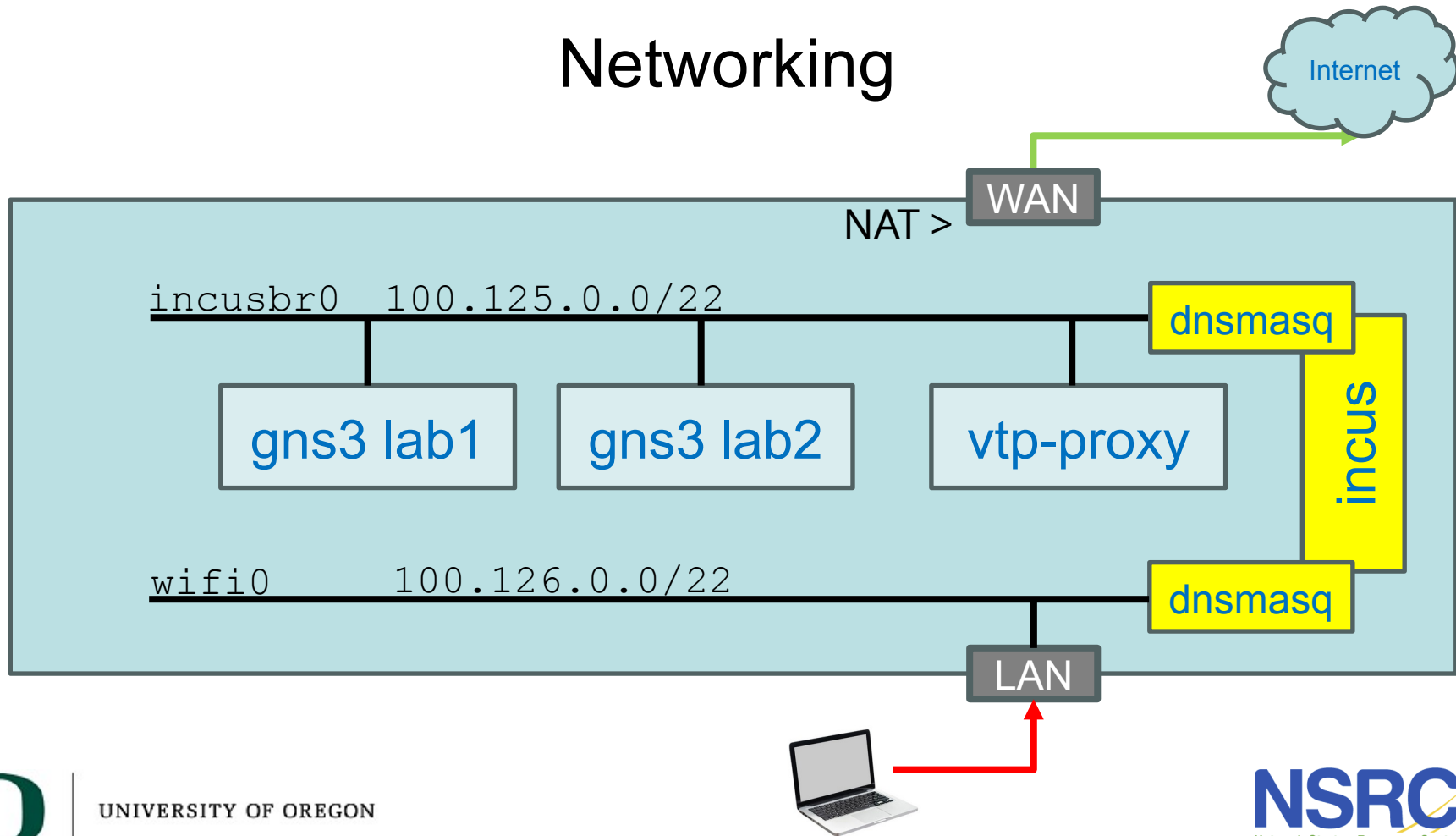
UNIVERSITY OF OREGON



Networking



Networking



dnsmasq

- Incus launches a dnsmasq instance on each managed network
- It's a combined DHCP + DNS server
- Containers get an address on incusbr0 (100.125.0.0/22)
 - DNS names "xxx.incus" are registered
 - These are only resolvable by other clients on the incusbr0 network
- Students get an address on wifi0 (100.126.0.0/22)

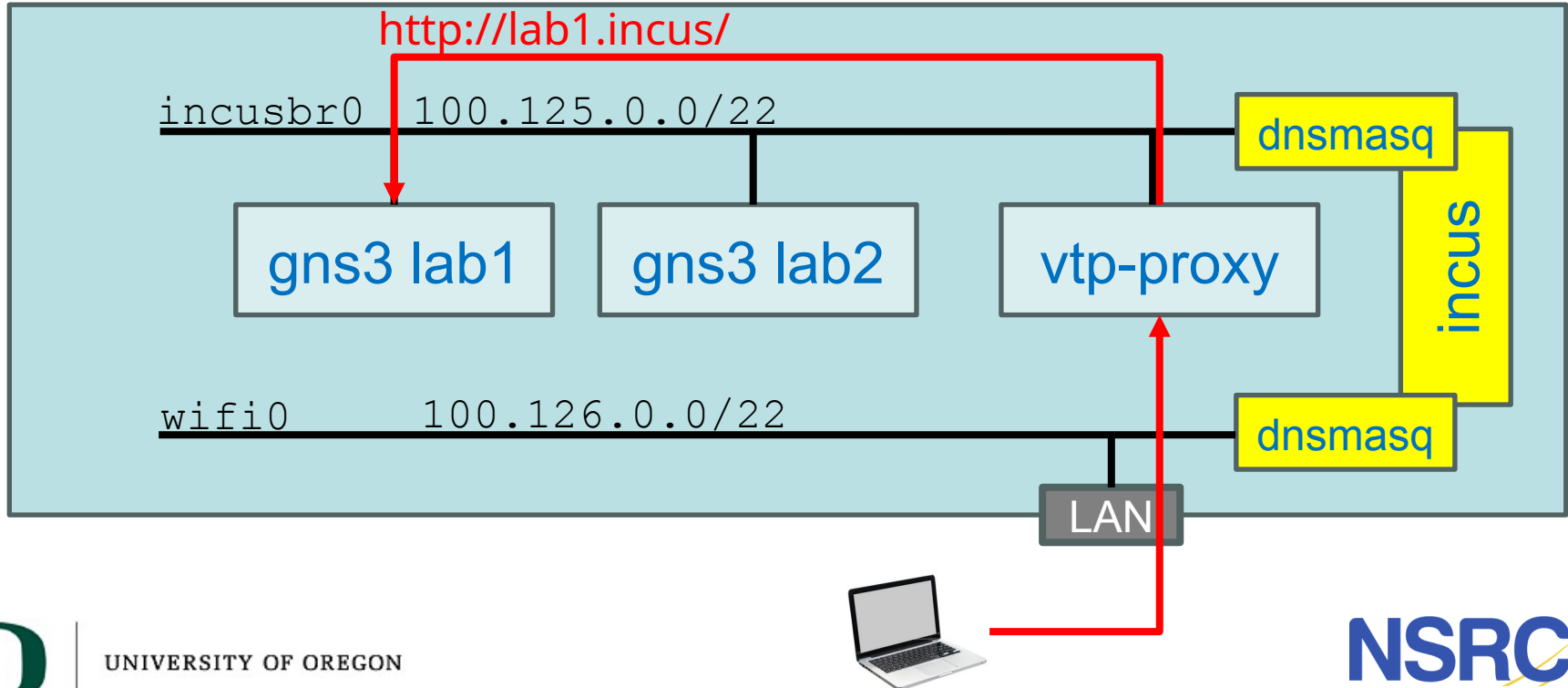


WAN and LAN for portable servers

- If server has only one on-board NIC, you can add a USB NIC
 - We recommend on-board = LAN, USB = WAN
 - The on-board NIC is usually more reliable, and this is where the student traffic hits the server hardest
- Or you can use VLANs into a switch (see documentation)



Web Access (local)



UNIVERSITY OF OREGON



<https://lab1.local.nsrc.org/>



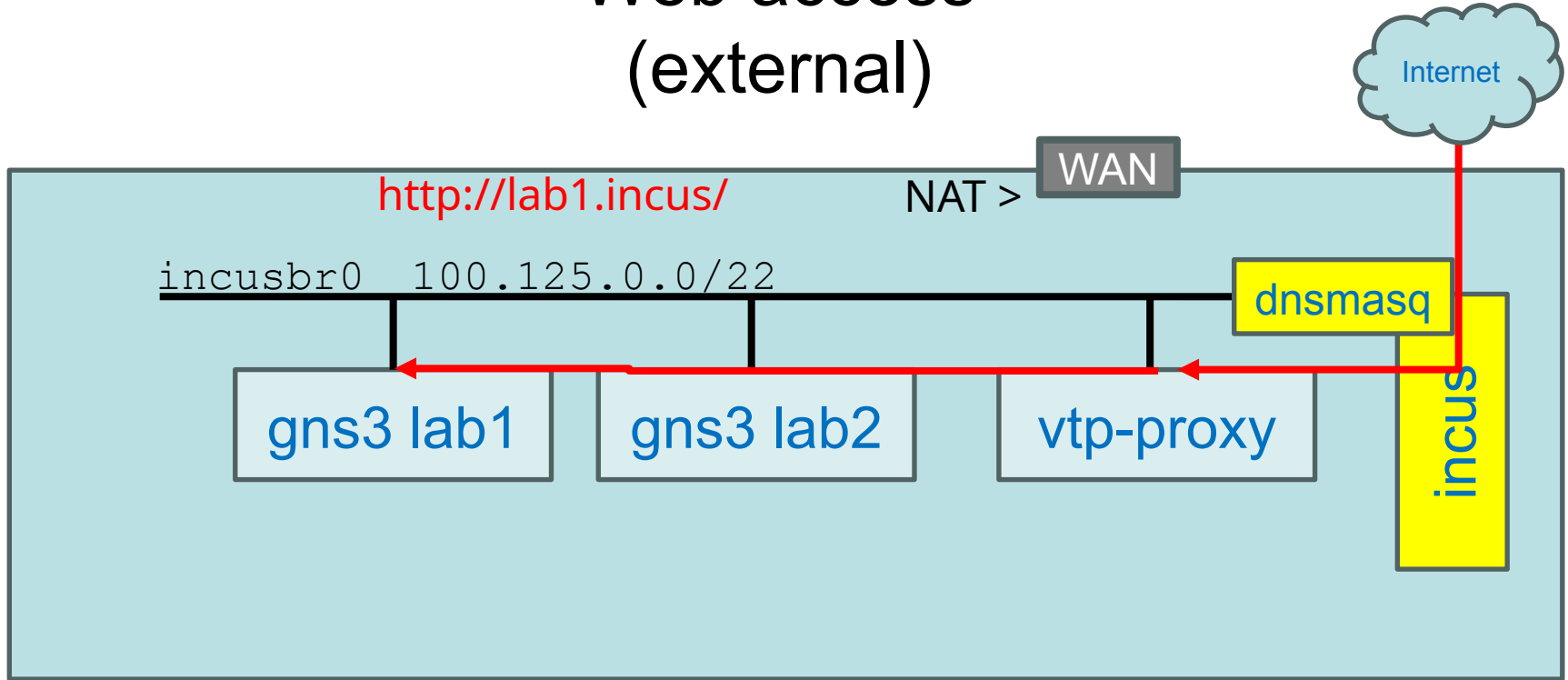
vtp-proxy

- **local.nsrc.org** and ***.local.nsrc.org** resolve to 100.126.0.1
 - requires students to use the DNS server we give out via DHCP
 - may be broken by static DNS configuration or VPNs
- All traffic arriving on the server on port 80 or 443 is forwarded by incus to the **vtp-proxy** container
- This performs HTTPS termination
 - we distribute keys and certs for ***.local.nsrc.org**
- It parses the URL: **foo.local.nsrc.org** and **xxx--foo.local.nsrc.org** are proxied to container "**foo**" (dnsmasq DNS name: **foo.incus**)



Web access (external)

<https://lab1.vtp.example.org/>



UNIVERSITY OF OREGON

External access

- Exactly the same, except coming in on the public interface
- vtp-proxy enforces authentication for requests not coming from trusted local addresses
 - Uses `apache mod_auth_form`
 - Shows a login page then sets a cookie
 - File with username(s) and password(s) for students
- You need a public IP address and DNS records

<code>vtp.example.org.</code>	<code>A</code>	<code>192.0.2.42</code>
<code>*.vtp.example.org.</code>	<code>A</code>	<code>192.0.2.42</code>
<code>_acme-challenge.vtp.example.org.</code>	<code>CNAME</code>	<code>xxxx.acme.nsrc.net.</code>

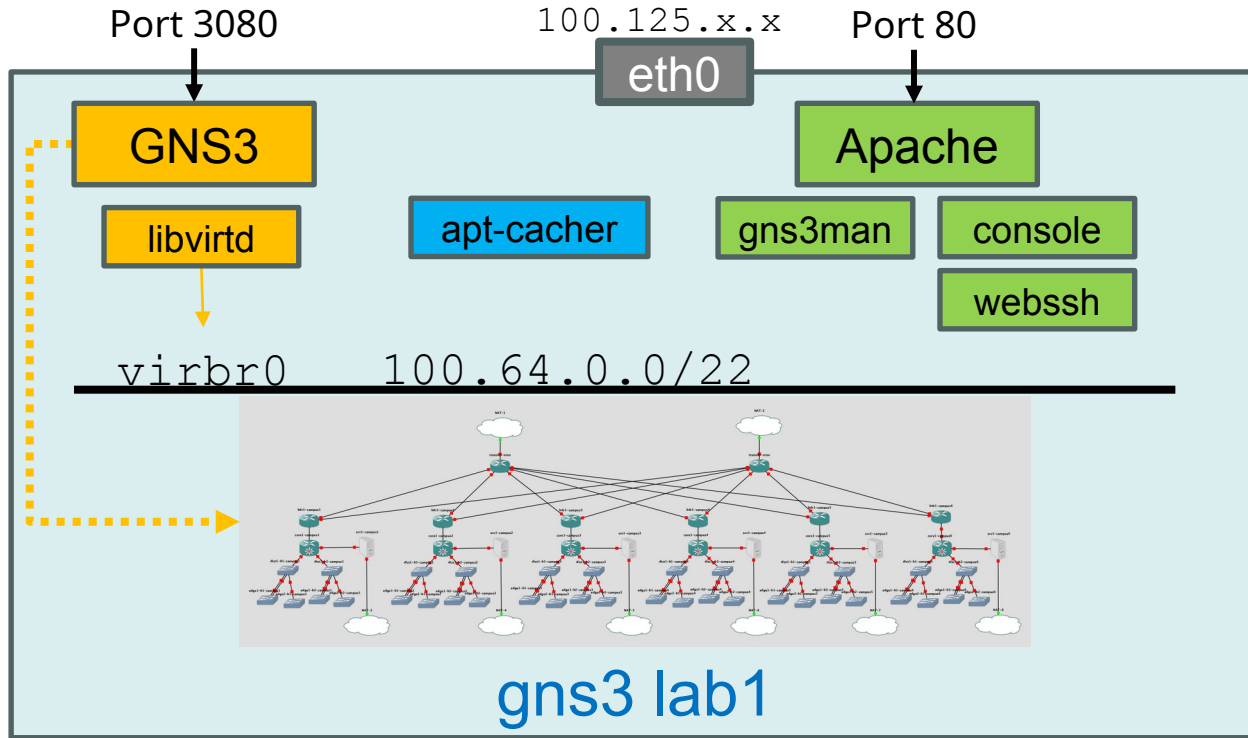


UNIVERSITY OF OREGON

xxxx = subdomain we generate to allow
your host to answer DNS-01 cert
challenges



Inside a GNS3 lab container



Inside a GNS3 container

- Addresses 100.64.x.x – 100.95.x.x reserved for containers
- GNS3 receives API / web UI traffic on port 3080
 - access via proxy as `p3080--lab1.local.nsrc.org`
- GNS3 starts the virtual routers (IOL) and servers (KVM VMs)
 - Connects their network interfaces together
 - The "NAT" connector (cloud icon) links to the virbr0 bridge
 - virbr0 is created by libvirt



Inside a GNS3 container

- Apache receives requests on port 80
 - Displays the index page, with all the links students need
 - Proxies requests for console, webssh, gns3man
 - Proxies requests to web servers inside the topology
 - Requests for `foo-p--lab1.anydomain` are proxied to `foo.ws.nsrc.org`
 - `ws.nsrc.org` names resolved in `/etc/hosts` inside the container
 - Example: `oob-srv1-campus1-p--lab1.anydomain` is proxied to `oob-srv1-campus1.ws.nsrc.org`
 - You don't need to know the details; the links point to the right places!
- Linux VMs fetch their packages via apt-cacher
 - Might move to a shared container (or vtp-proxy) in future

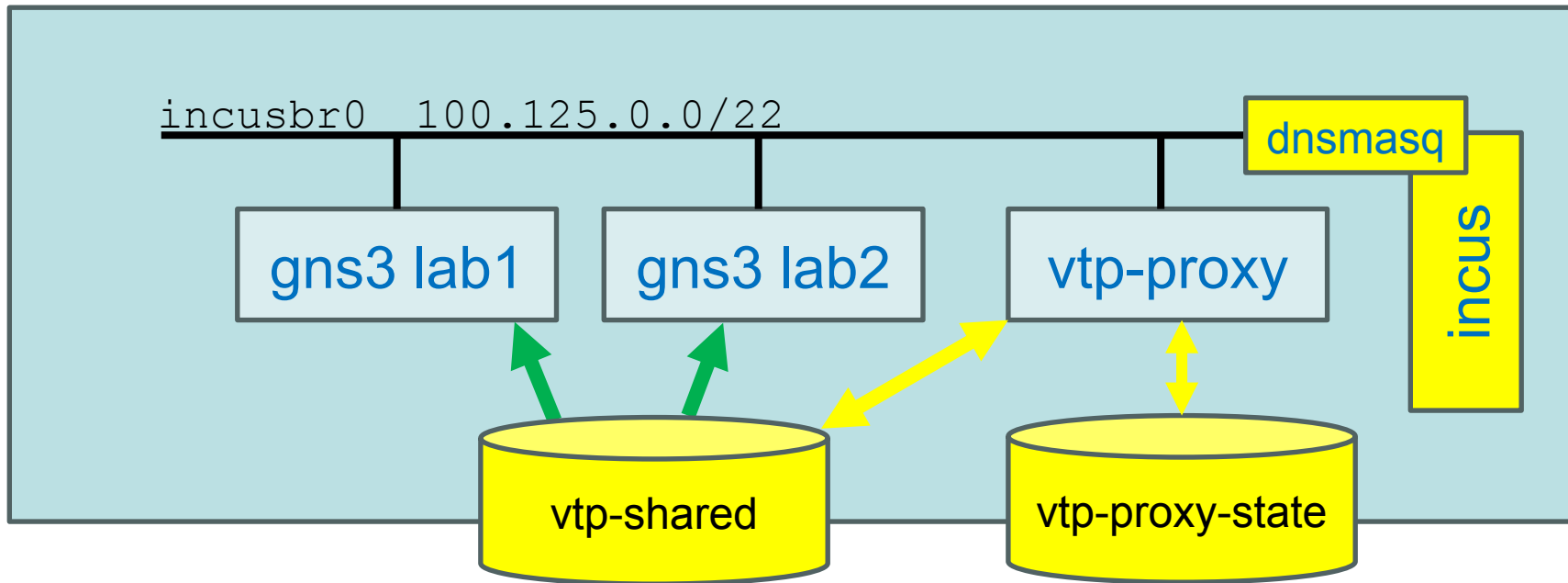


Pre-built container images

- The various GNS3 lab topologies differ by way of:
 - The GNS3 project file
 - The index page
 - The `/etc/hosts` file
- We build container variants which are pre-configured, e.g.
 - cndo = Campus Network Design and Operations
 - nmm = Network Monitoring and Management
 - bgp-deploy = BGP Deployment workshop
 - peering-ixp = Peering and IXP workshop
 - riso = Routing Infrastructure and Security Operations



Final piece: incus storage volumes



Incus storage volumes

- Some VM images are too large to include in the container
 - or we are not allowed to distribute them
- You fetch those yourself and put them in the **vtp-shared** volume
 - mounted read-only on all GNS3 containers
 - mounted read-write on the vtp-proxy container
 - also contains shared GNS3 settings (e.g. the API password)
- The **vtp-proxy-state** volume contains things like SSL settings and web passwords, and allows you to rebuild the vtp-proxy container without losing its data



Building the platform: outline

- Install Ubuntu LTS
- Run setup scripts (host-setup.sh, incus-setup.sh)
- Launch the vtp-proxy container

```
incus launch nsrc:vtp-proxy vtp-proxy -p vtp-proxy -p http-proxy
```

- Configure the vtp-proxy container

```
incus exec vtp-proxy -- sslsetup
```

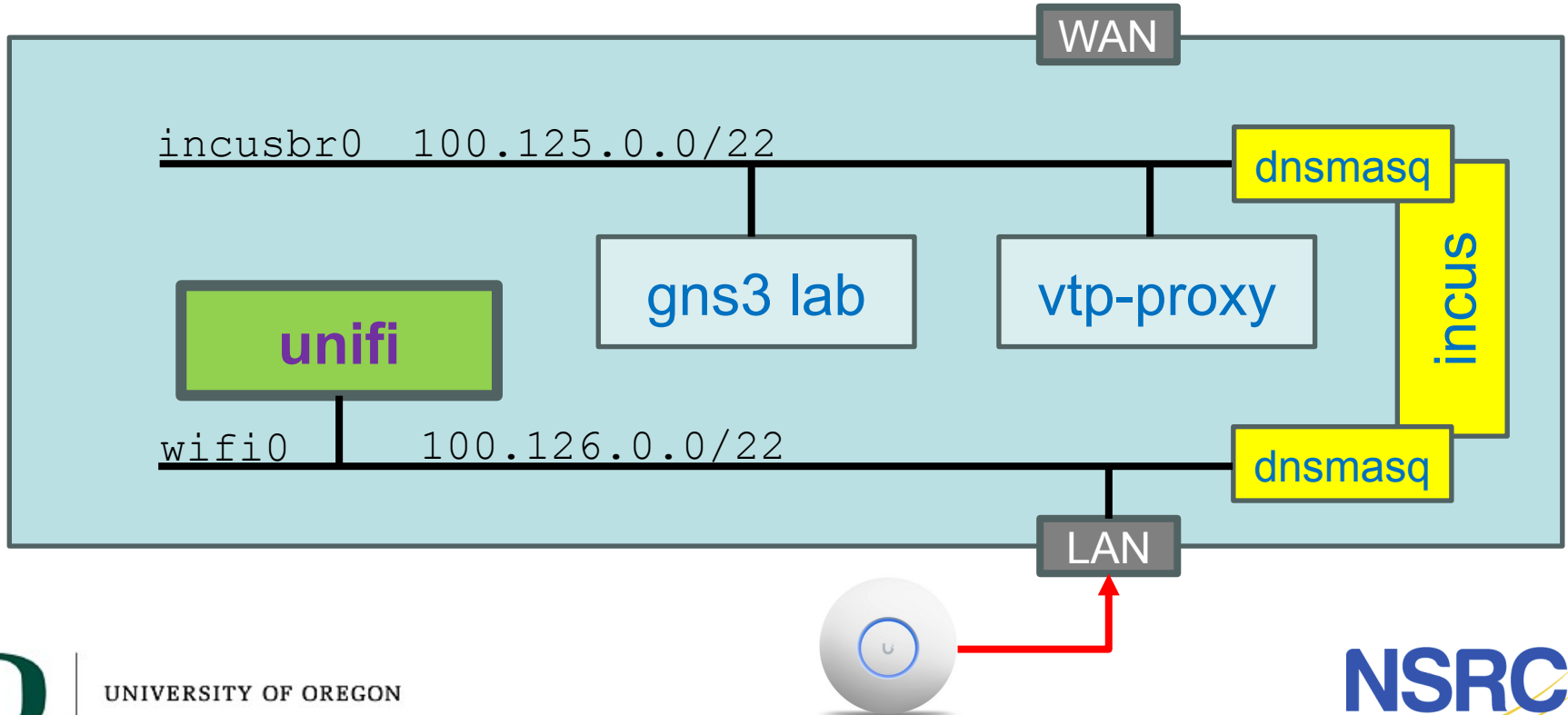
- Install the VM/IOL images
- Launch the desired topology container(s)

```
incus launch nsrc:cndo cndo -p vtp
```

- Done!



Running additional containers



Running additional containers

- Incus is very flexible
 - Launch your own OS instances and customize them however you like
- Documentation shows how to create a container for the Unifi controller and connect it to the wifi0 network
 - See under "Classroom Tools"



What you need next

- Documentation: images.nsrc.org/docs/
- images.nsrc.org also hosts the setup scripts, container images and NSRC VM images
 - You will need a username and password for this: just ask!
- Cisco IOL and IOL-L2 images
 - Direct from Cisco - register for a CML-FREE account

