Virtualization & Network Simulation

Proxmox VE and GNS3 Workshop

Estrada, Colline
Evangelista, John Marlo

How much work does it take to create and maintain a Virtual Machine?

Training in deploying and managing virtual machines and network topologies using open-source virtualization platforms

- Virtualization overview
- Proxmox VE concepts
- Demonstration of Proxmox installation in a bare metal
  - Network and storage configuration
  - Integrating an LDAP
- Creating a Virtual Machine
- GNS3 overview and installation
- Proxmox Cluster
- Proxmox Backup Server

Server virtualization in a nutshell

- Enables the creation of multiple virtual environments in a single physical hardware
  - Type I bare metal hypervisor. VM resources are directly scheduled to the hardware
    - KVM
    - VMWare ESXi
  - Type II hosted hypervisor. Runs on a host OS as an application
    - Oracle VM VirtualBox
    - VMware Workstation
- VMs viewed as files rather than physical devices [2]
  - CAN BE REPLICATED, COPIED, DEPLOYED

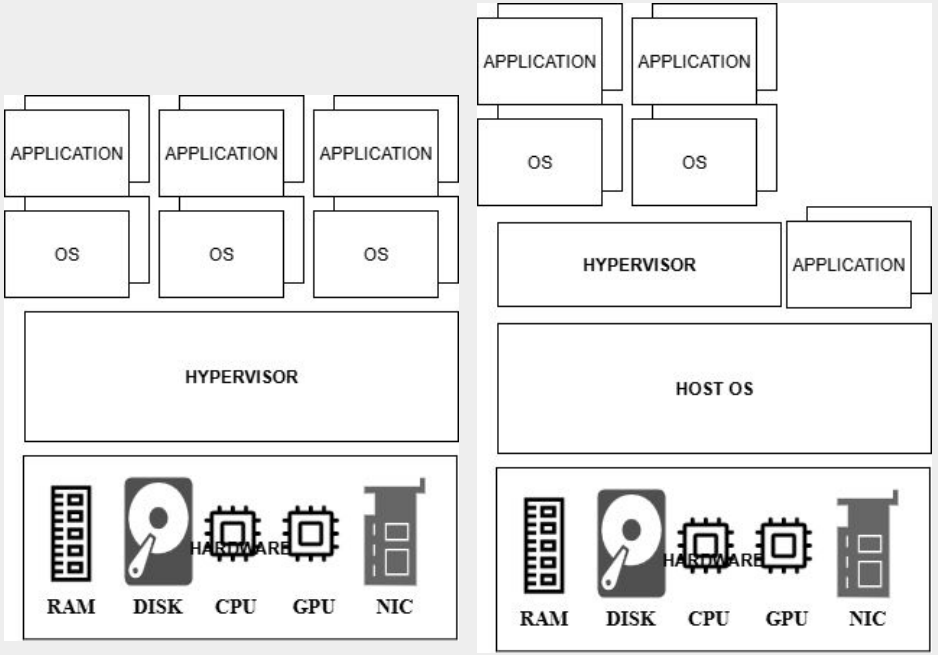


Figure 1. (a) Type I and (b) Type II hypervisor diagrams

[1] Accessed from: https://www.ibm.com/think/topics/virtualization
[2] M. Healey, C. Anderson, J. Humphreys, “IBM Virtualization Services,” October 2008. Available at https://public.dhe.ibm.com/software/uk/itsolutions/move/droodles/idc-white-paper-ibm-virtualization-services.pdf

Why did we choose Proxmox VE?

We wanted to find an alternative to KVM (Kernel-based Virtual Machine) + QEMU

- Free & open-source
- KVM (production servers, qcow2)
- Better web GUI (closest we got is virt-manager)
- Support for cluster management and HA
- Intuitive backup & snapshot tools

What is Proxmox Virtual Environment (Proxmox VE)?

Open-source server virtualization platform that allows you to

- Run and manage virtual machines (VMs) and containers
- Use a web-based GUI for easy administration
- Support KVM (Kernel-based Virtual Machine) and LXC containers
- Manage clusters, backups, and high availability (HA)
- Integrate software-defined storage (like Ceph or ZFS)
- Automate with tools like Ansible or API

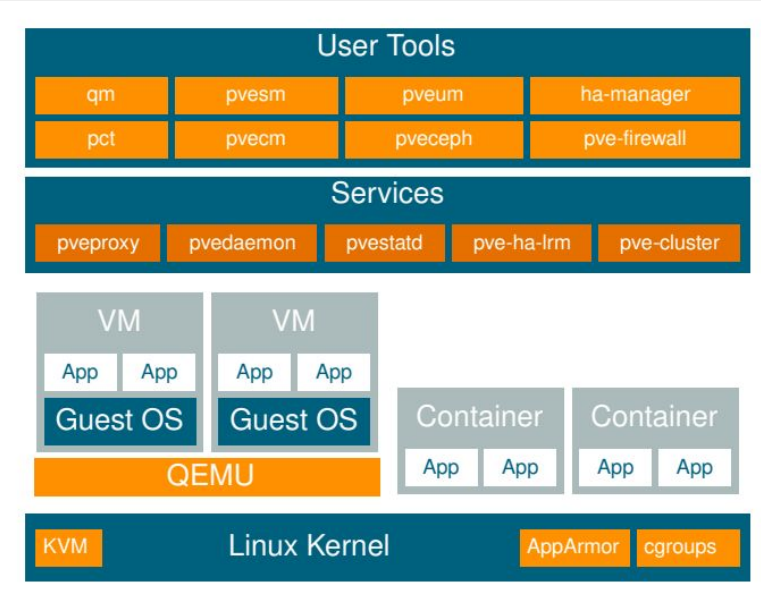


Figure 2. Proxmox VE architecture diagram

[3] Accessed from: https://pve.proxmox.com/pve-docs/pve-admin-guide.pdf

Hands on:

Proxmox VE Concepts & Live Demonstration

Preliminaries

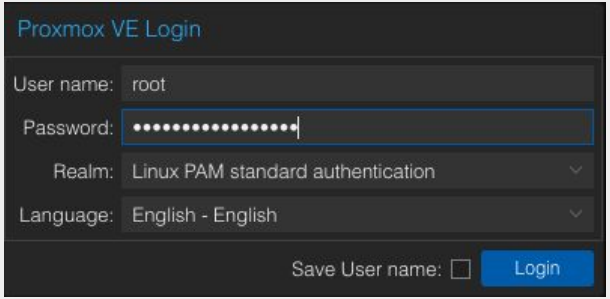
- **Download ISO https://www.proxmox.com/en/downloads/proxmox-virtual-environment**
- **Take note**
  - **Set the following:**
    - Target disk(s) for the installation
    - location, time zone, and keyboard layout
    - password of the superuser
    - email address
    - network configuration

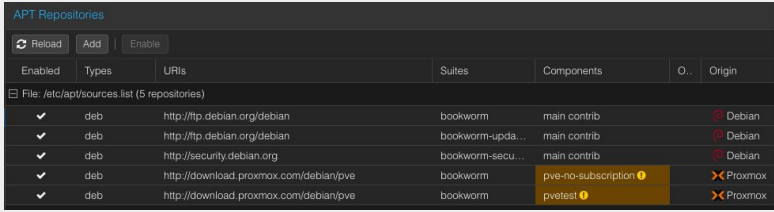
2.1.2 Recommended System Requirements

- Intel 64 or AMD64 with Intel VT/AMD-V CPU flag.
- Memory: Minimum 2 GB for the OS and Proxmox VE services, plus designated memory for guests. For Ceph and ZFS, additional memory is required; approximately 1GB of memory for every TB of used storage.
- Fast and redundant storage, best results are achieved with SSDs.
- OS storage: Use a hardware RAID with battery protected write cache (“BBU”) or non-RAID with ZFS (optional SSD for ZIL).
- VM storage:
  - For local storage, use either a hardware RAID with battery backed write cache (BBU) or non-RAID for ZFS and Ceph. Neither ZFS nor Ceph are compatible with a hardware RAID controller.
  - Shared and distributed storage is possible.
  - SSDs with Power-Loss-Protection (PLP) are recommended for good performance. Using consumer SSDs is discouraged.
- Redundant (Multi-)Gbit NICs, with additional NICs depending on the preferred storage technology and cluster setup.
- For PCI(e) passthrough the CPU needs to support the VT-d/AMD-d flag.

I. Installing and configuring Proxmox VE

1. Point your browser to the IP address given during the installation and port 8006, for example: https://youripaddress:8006
2. Log in using the root (realm PAM) username and the password chosen during installation
3. Configure repository
   a. Upload your subscription key to gain access to the Enterprise repository
   b. Otherwise, you will need to set up one of the public, less tested package repositories (No-Subscription) to get updates for security fixes, bug fixes, and new features
4. Network Configuration





II. Setting up networks and VLAN-aware bridges

- Uses Linux network stack
  - Linux Bond
    - If your switch supports the LACP (IEEE 802.3ad) protocol, then we recommend using the corresponding bonding mode (802.3ad)
    - Otherwise you should generally use the active-backup mode
  - Linux VLAN
    - VLAN awareness on the Linux bridge
    - "traditional" VLAN on the Linux bridge
  - Linux Bridge

III. Overview of storage options: local, ZFS, shared (NFS), and Ceph, etc

- **local**
  - Default local storage, usually mounted at /var/lib/vz
  - ISO images, container templates, backups
- **local-lvm**
  - LVM-thin volume for VM disks (faster, space-efficient, raw disk access)
  - Virtual machine disks (qcow2/raw)
- **ZFS**
  - built-in support, allows you to use it as local storage for VM disks, containers, backups, and ISO files
- **NFS**
  - access shared storage over a network — great for clusters, backups, and ISO storage
- **Ceph**
  - 3 or more Proxmox nodes (for quorum)
  - At least 1 unused disk per node (for OSDs)
  - Cluster already created in Datacenter > Cluster

[3] Accessed from: https://pve.proxmox.com/pve-docs/pve-admin-guide.pdf

IV. Creating and managing virtual machines

1. Upload OS ISO or download from URL
2. Create Virtual Machine
    a. **General Settings** - Node, VM ID (unique), name, resource pool
    b. **OS Settings** - proper Operating System(OS) allows Proxmox VE to optimize some low level parameters
    c. **System Settings** - display type, SCSI controller, QEMU Guest Agent - show some more information, and complete some actions more intelligently, different firmware and machine types, namely SeaBIOS and OVMF (PCIe passthrough), Machine Type - hardware layout of the VM's virtual motherboard
    d. **Hard disk** - VirtIO SCSI single and enabling the IO Thread setting for the attached disks is recommended if you aim for performance
        i. block devices (LVM, ZFS, Ceph) - raw disk image format
        ii. files based storages (Ext4, NFS, CIFS, GlusterFS) - raw disk image format or the QEMU image format

IV. Creating and managing virtual machines

e. Image format
f. CPU - CPU Type
    i. If you want an exact match, you can set the CPU type to **host**, but CPU flags might be missing during migration,
    ii. backend default is **kvm64** - works on all x86_64 host CPUs
    iii. UI default when creating a new VM is **x86-64-v2-AES**
g. Memory
h. Network device
    i. Intel E1000 is the default, and emulates an Intel Gigabit network card
    ii. VirtIO paravirtualized NIC - maximum performance, driver required

(Optional) Integrating identity management (e.g., LDAP) for user sync

- Base Domain Name: ou=People,dc=ldap-test,dc=com
- User attribute: uid
- Bind User: cn=Manager,dc=ldap-test,dc=com
- Bind Password
- Scope: Users

```
# user1 of People at ldap-test.com
dn: uid=user1,ou=People,dc=ldap-test,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: user1
cn: Test User 1
sn: Testers
description: This is the first test user.
```

Alternative

Proxmox VE authentication server realm for simple Unix-like password store

Creating pools, roles, and assigning granular permissions

Pools

- group a set of virtual machines and datastores
- simply set permissions on pools (/pool/{poolid}), inherited by all pool members
- Datacenter → Permissions → Pool

Roles

- simply a list of privileges
  - **Administrator**: has full privileges
  - **NoAccess**: has no privileges (used to forbid access)
  - **PVEAdmin**: can do most tasks, but has no rights to modify system settings (Sys.PowerMgmt, Sys.ModifyRealm.Allocate) or permissions (Permissions.Modify)

Privileges

- the right to perform a specific action
  - **VM.PowerMgmt:** power management (start, stop, reset, shutdown, . . .)
  - **VM.Config.Disk**: add/modify/remove disks

Demonstration of Proxmox Guest Agent usage for command execution and automation

- service which runs inside the VM
  - a communication channel between the host and the guest
    - Fetch IP addresses in the VM summary panel
    - Starting a backup, guest is told via the guest agent to sync outstanding writes via the fs-freeze and fs-thaw commands
    - To properly shutdown the guest, instead of relying on ACPI commands or windows policies
- Install
  - Install **qemu-guest-agent** on the guest VM
  - Enable guest agent communication - enabled in the VM's **Options** panel
    - Restart of the VM is necessary
- Run command from host to guest
  - qm guest exec $vmid -- <$cmd>
  - qm guest exec 100 -- ip addr

Q&A

Advanced Proxmox Features

Creating a Proxmox cluster

- **Corosync Cluster Engine** for reliable group communication
- Under Datacenter → Cluster, click on Create Cluster.
  - Enter the cluster name and select a network connection from the drop-down
- To join a cluster, copy cluster information in Datacenter → Cluster → Cluster Join Information

5.1 Requirements

- All nodes must be able to connect to each other via UDP ports 5405-5412 for corosync to work.
- Date and time must be synchronized.
- An SSH tunnel on TCP port 22 between nodes is required.
- If you are interested in High Availability, you need to have at least three nodes for reliable quorum. All nodes should have the same version.
- We recommend a dedicated NIC for the cluster traffic, especially if you use shared storage.

- The root password of a cluster node is required for adding nodes.
- Online migration of virtual machines is only supported when nodes have CPUs from the same vendor. It might work otherwise, but this is never guaranteed.

[3] Accessed from: https://pve.proxmox.com/pve-docs/pve-admin-guide.pdf

Performing live migration between nodes

1. Setup shared storage (example NFS)

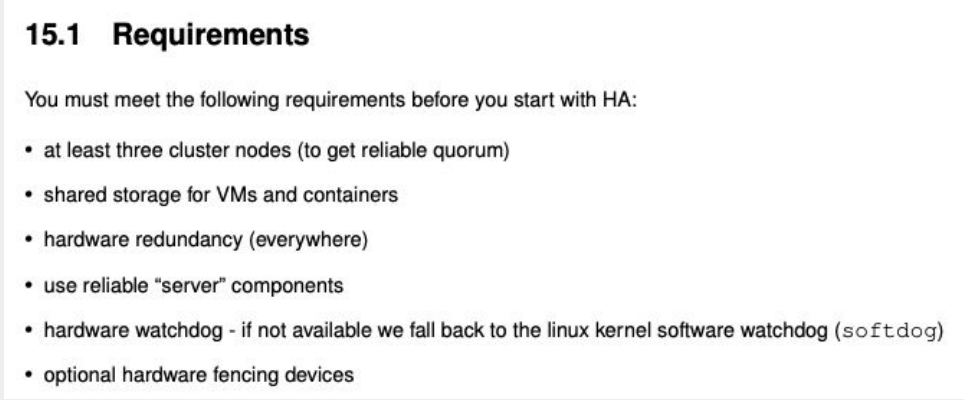
   Datacenter → Storage → Add → NFS

   Name: production-storage
   Server: 10.158.3.107
   Export: /mnt/nfs-share

2. Create VM with storage “production-storage”
3. Live migrate the VM to another node

Setting up High Availability (HA) for critical VMs

1. Datacenter → HA → Groups → Create
2. Datacenter → HA
   Max_relocate: Maximal number of service relocate tries when a service failed to start
   Max_restart: Maximal number of tries to restart the service on a node after its start failed
   Request State: tries to “restart” this service



15.1 Requirements

You must meet the following requirements before you start with HA:

- at least three cluster nodes (to get reliable quorum)
- shared storage for VMs and containers
- hardware redundancy (everywhere)
- use reliable “server” components
- hardware watchdog - if not available we fall back to the linux kernel software watchdog (`softdog`)
- optional hardware fencing devices

High Availability vs Failover

HA means

- if a node running a VM fails, the HA manager will detect the failure and restart the VM on another node
- **brief downtime** during detection, fencing (if needed), and restart

Failover means

- automatic switchover with **no downtime**
- Proxmox VE **does not currently support** zero-downtime failover

PVE backup and restore

- Full and live snapshot backups of VMs and CTs
  - GUI or via the vzdump command-line tool
  - Scheduled backups
  - Supports retention configuration

```
$ vzdump 777 --prune-backups keep-last=3,keep-daily=13,keep-yearly=9
```

- Restoring using the GUI or the CLI

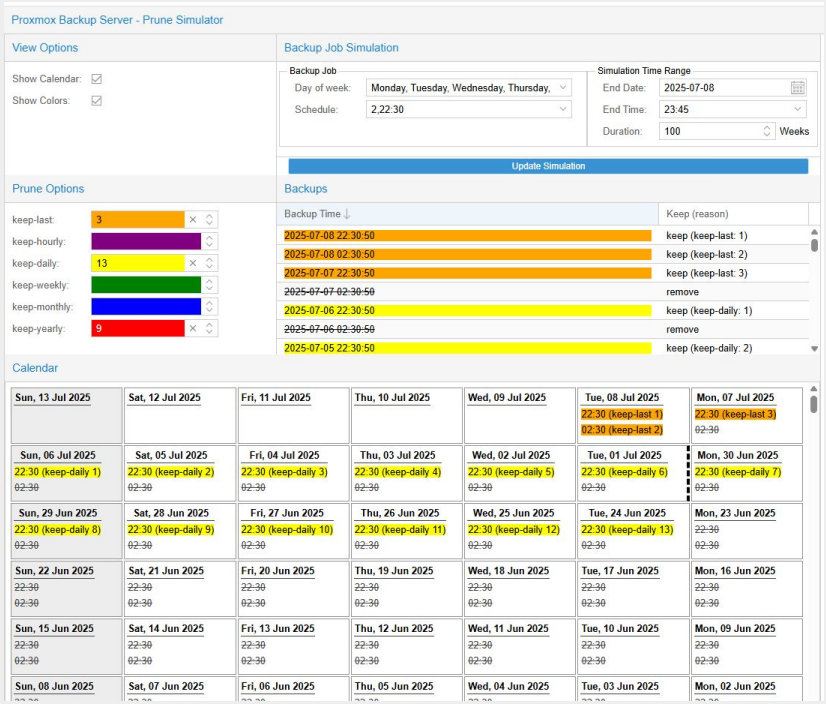


Figure 3. Prune Simulator based on the example vzdump sample [6]

[6] Accessed from: https://pbs.proxmox.com/docs/prune-simulator /

Incremental with deduplication backup

- Backup client reads and divides the data from the source into chunks
- Chunks are hashed
- Hashes are sent to the backup server
- Server checks against the data store
- Client uploads the chunk when it is new/modified
- Regardless of how many backups reference a chunk, deduplication ensures only on physical copy of each unique chunk is stored

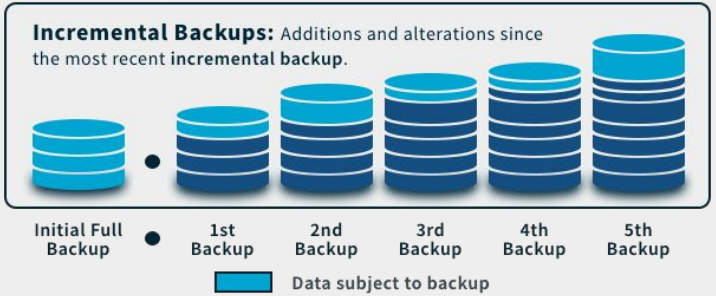


Figure 4. Incremental backup [7]

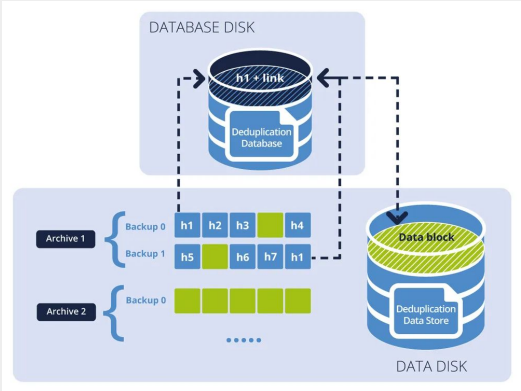


Figure 5. Deduplication [8]

[7] Accessed from: https://www.spanning.com/blog/types-of-backup-understanding-full-differential-incremental-backup/
[8] Accessed from: https://www.acronis.com/en-sg/blog/posts/deduplication/
[9] Accessed from: https://pbs.proxmox.com/docs/technical-overview.html

Setting up and using Proxmox Backup Server (PBS)

- Download PBS ISO: https://enterprise.proxmox.com/iso/proxmox-backup-server_3.4-1.iso

Recommended Server System Requirements

- CPU: Modern AMD or Intel 64-bit based CPU, with at least 4 cores
- Memory: minimum 4 GiB for the OS, filesystem cache and Proxmox Backup Server daemons. Add at least another GiB per TiB storage space.
- OS storage:
  - 32 GiB, or more, free storage space
  - Use a hardware RAID with battery protected write cache (*BBU*) or a redundant ZFS setup (ZFS is not compatible with a hardware RAID controller).
- Backup storage:
  - Prefer fast storage that delivers high IOPS for random IO workloads; use only enterprise SSDs for best results.
  - If HDDs are used: Using a metadata cache is highly recommended, for example, add a ZFS special device mirror.
- Redundant Multi-GBit/s network interface cards (NICs)

[4] Accessed from: https://pbs.proxmox.com/docs/installation.html
[5] Accessed from: https://www.proxmox.com/images/download/pbs/docs/proxmox-backup-3-4.pdf

Integration of the Proxmox Backup Server to the PVE

On PBS

- Create datastore (enter name and path)
- Backup datastore → Summary → Show connection information → Copy fingerprint

ON PVE

- Datacenter → Storage → Add → Proxmox Backup Server

Q&A

GNS3 Concepts & Live Demonstration

What is GNS3 (Graphical Network Simulator-3)?

open-source network emulation tool that allows you to

- Simulate real-world networks using virtual routers, switches, and firewalls
- Emulate Cisco, Juniper, Linux, and other devices
- Connect virtual devices to real networks or the internet
- Design network topologies visually with a drag-and-drop GUI
- Use Dynamips, QEMU, and Docker for virtualization

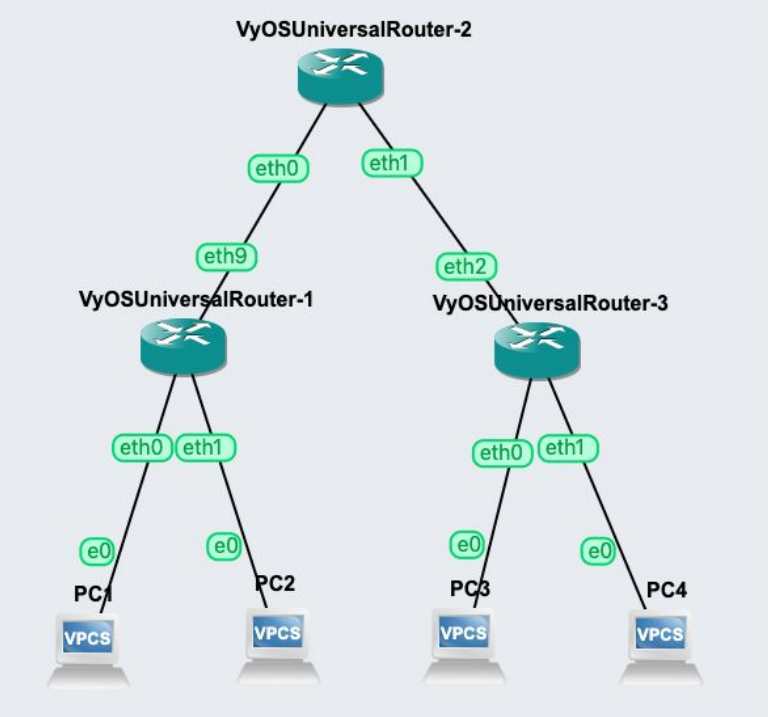
Installing and configuring GNS3

- Installation on Ubuntu 22.04 VM
  - https://download.eee.upd.edu.ph/GNS3-Installation.pdf
- Goto GNS3 UI http://<ip-address>:3080
- Enable authentication on /home/gns3/.config/GNS3/2.x/gns3_server.conf :

  set auth = True, username and password

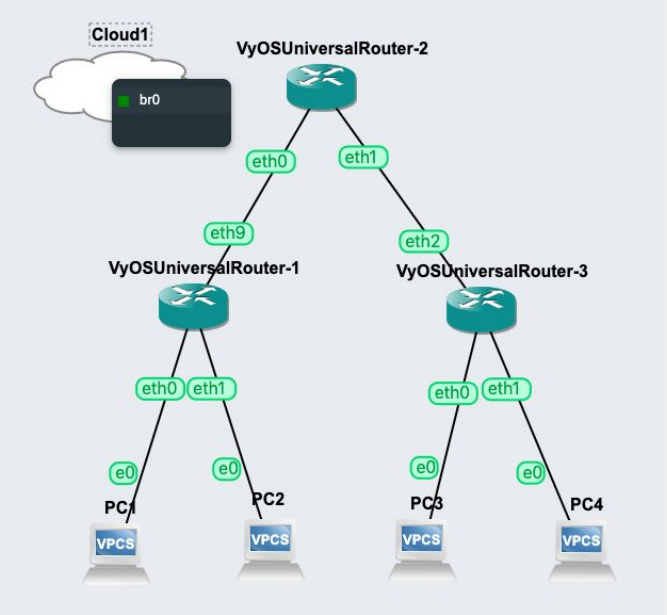
Building network topologies with virtual routers and switches

1. Goto GNS3 UI http://<ip-address>:3080
2. Create a “Test” project
3. Select the “+” sign
4. Select: New template
5. Select: Install new appliance from the GNS server
6. On the filter search: VyOS then select install
7. Select: Install the appliance locally
8. On the installed required files select the import button for vyos-x.x.x-kvm-amd64.qcow2
9. Import:
   Download here:
   https://cdn.as212934.net/routers/VyOS/
10. Click Create
11. Add Template name: VyOS
12. *Optional* Change symbol: Projects > Go to Preferences > Qemu > VyOS > General Settings > Symbol > search for a router symbol



Integrating GNS3 with Proxmox VMs

1. On the Proxmox VM, create a bridge interface on the guest VM
2. Create a bridge interface using netplan
   Sample:
   network:
     ethernets:
       ens18:
         dhcp4: false
     bridges:
       br0:
         interfaces: [ens18]
         dhcp4: true
     version: 2
3. Edit /home/gns3/.config/GNS3/2.x/gns3_server.conf
   a. Add bridge:
      allowed_interfaces = virbr0,br0
4. Restart gns3
   a. $ systemctl restart gns3
5. On GNS3 GUI, add node > Cloud > Add link. The external bridge should appear.



Other resource materials

Architectural Principles for Virtual Computer Systems by R.P. Goldberg. Accessed from: https://apps.dtic.mil/sti/tr/pdf/AD0772809.pdf

Proxmox VE Admin Guide. Accessed from: http://pve.proxmox.com/pve-docs/pve-admin-guide.pdf

Open Forum / Q&A