

# LS Lab 1 - Hypervisors & Virtualization

---

Individual for task 1 & 2, then in a team of two for task 3 and bonuses

*The whole point of true Type I Hypervisors is to put a bunch of them together in view to build-up a cluster farm. This, in turns, permits to enable virtualization-level High-Availability and automatic guest hot-migrations shuffling, to optimize the load across the farm. In fact, optimizing host's CPU load also increases guest's performance, as the hypervisor schedules guest's jobs.*

## Task 1 - Choose virtualization technology

---

- *Linux KVM* (default choice, but guest configs are XML and harder to synchronize among the farm...)
- *Community XEN* (harder to install and run, but there are some benefits that you can try to explore...)
- *XCP-NG XEN* on nested virtualization (no team, will be alone for the whole lab)
- *XenServer* on nested virtualization (no team, will be alone for the whole lab )
- *VMware vSphere* (be creative and explore advanced options due it's easier to configure)
- *Microsoft HyperV* (you need to be very creative there due it's pretty easier to configure)
- *Otherwise exotic choices & presentation are possible, discuss it first with TA*

## Task 2 - Local implementation

---

*Agree on which host system version you're going to work on with you teammate. Different OS and versions may work, and will make task 2 even more interesting. But if you don't want to take any risks, check `uname -r` and `lsb_release -a` first. Proceed with the following steps **INDIVIDUALLY**: everybody needs to know how to setup a single and true virtualization host.*

- **Host install** – Install the hypervisor and host tools on a physical machine (make sure VT/AMD-V is enabled).
- **Guest install** – Install a guest with a local virtual disk with whatever method fits best, just to validate that you get a VM up and running. *If using XEN, be clear and provide details on what type of guest you chose.*
- **Sparse-file virtual disk** – Install another guest hard and DIY way. Create a SPARSE file (either with dd seek or QCOW2, for example), mount it in a folder and use debootstrap to get a Ubuntu Server system over there, quick & dirty. For VMware/HyperV users, try to do something similar.
- **Network** – Setup the network manually (*always show how you do it in the report*). For *Community XEN* you need to setup a bridge. For *KVM/libvirt*, please get rid of the default setup and do it yourself. In other words, make your guests capable to obtain a DHCP lease from the SNE network (or any other local network in the case of remote work).
- **Text console** – Make sure you can reach the text console of the guest from the host. What configurations allows for both, the kernel and the userland system to show up there? Eventually disable the graphical console.
- **Snapshot** – Proceed with a hot-snapshot, meaning while the guest is running, take it. Attempt to make sure that the file system was properly dealt with...

## Task 3 - Cluster validation

---

*Now as a team, choose which one of the two machines is also going to provide the shared storage for virtual disks to live in. Set it up and share both ideally, guests virtual disks and configurations.*

- Take team member 1's favorite guest (virtual disk and configuration) and put it in the shared storage
- Take team member 2's favorite guest (virtual disk and configuration) and put it in the shared storage
- Eventually fix the pathes in the configuration and validate that both guests run as well as before
- Now shut them down and run them on the other team member's machine/host (cold-migration)
- Now don't even shut them down while migrating... (hot-migration/live-migration)

*You have to show a live DEMO of task 3 and bonuses. So, record a video of your process and upload among with report. By the way, you still need to include this task in your report as usual.*

## Bonuses

---

1. *Still as a team, choose a solution to orchestrate your guests among the farm, as described in the introduction: enable HA (when a host crashes, dead guests are restarted elsewhere) and DRS (live-migrations shuffles). Some examples:*

- Ganeti (XEN or KVM)
- XCP-NG Center (Windows client)
- (untested) XEN Orchestra (non free?)
- (untested) CloudStack
- (untested) OpenNebula
- (untested) DanubeCloud (missing features?)
- (untested) ManagelQ
- OpenStack

Or do parallel guests instead of HA with e.g. Remus, Kemari?, COLO?

2. If you are dealing with **VMWare, HyperV,...** present and explain key features and advanced guest optimization settings. Show - or try to justify - why this is so expansive compared to your colleague's solutions. Prepare and deliver a presentation for this task.