Configuration

Virtual cluster

On the host, you must set these sysctl settings:

```
net.bridge.bridge-nf-call-arptables = 0
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
```

You must define 3 network interfaces on each host of your cluster.

- One interface connects to a virtual network in NAT mode
- Two interfaces connect to two virtual networks with a MTU set to 9000 (it's to simulate an ethernet cable between two machines)

Inventories

The inventory must define these hosts to run:

- `cluster_machines`: Set of hosts in the cluster
- `standalone_machine`: To define only the cluster is composed with one host (replace `cluster_machines`)

The inventory must define these variables:

- `ansible_connection`: Protocol to use to connection to machine
- `ansible_python_interpreter`: Path to the python interpreter binary
- `ansible_ssh_common_args`: Arguments to add for the SSH connection
- `ansible_user`: Login to use for the connection to machine

Playbooks

Prerequisite

When the host is installed, the `ansible/playbooks/cluster_setup_prerequisdebian.yaml` need to launch to finish the installation.

The inventory must define these variables to run the playbook:

- `admin_user`: Default user with admin privileges
- `admin_passwd`: Password hash *(optional)*
- `admin_ssh_keys`: *(optional)*
- `apply_network_config`: Boolean to apply the network configuration
- `admin_ip_addr`: IP address for SNMP
- `cpumachinesnort`: Range of allowed CPUs for no RT machines
- `cpumachines`: Range of allowed CPUs for machines (RT and no RT)
- `cpumachinesrt`: Range of allowed CPUs for RT machines
- `cpuovs`: Range of allowed CPUs for OpenVSwitch
- `cpusystem`: Range of allowed CPUs for the system
- `cpuuser`: Range of allowed CPUs for the user
- `irqmask`: Set the `IRQBALANCE_BANNED_CPUS` environment variable, see `irqbalance` manual
- `livemigration_user`:
- `logstash_server_ip`: IP address for `logstash-seapath` alias in `/etc/hosts`
- `main_disk`: Main disk device to observe his temperature
- `workqueuemask`: The negation of the `irqmask` (`= ~irqmask`)

In this part, the playbook define the scheduling and the prioritization (see the section).

Hardening

The `ansible/playbooks/cluster_setup_hardening_debian.yaml` playbook enables system hardening and the `ansible/playbooks/cluster_setup_unhardening_debian.yaml` playbook disables it.

The hardened elements are:

- the kernel with the parameters of the command line (see below section), the *sysfs* and modules;

- the GRUB;
- the systemd services;
- adding of bash profiles;
- SSH server;
- adding of `sudo` rules;
- the shadow password suite configuration;
- the secure tty;
- the audit daemon.

Kernel

The project uses a real-time kernel, the Linux kernel with the PREEMPT_RT patch. So, he needs to have some parameters as:

- `cpufreq.default_governor=performance`: Use the `performance` governor by default (more details here).
- `hugepagesz=1G`: Uses `1` giga-bytes for HugeTLB pages (more details here).
- `intel_pstate=disable`: Disables the `intel_pstate` as the default scaling driver for supported processors (more details here).
- `isolcpus=nohz,domain,managed_irq`: `nohz` to disable the tick when a single task runs; `domain` to isolate from the general SMP balancing and scheduling algorithms; `managed_irq` to isolate from being targeted by managed. See the Scheduling and priorization section.
- `no_debug_object`: Disables object debugging.
- `nosoftlockup`: Disable the soft-lockup detector (more details here).
- `processors.max_cstate=1` and `intel_idle.max_cstate=1`: Discards of all the idle states deeper than idle state `1`, for the `acpi_idle` and `intel_idle` drivers, respectively (more details here).
- `rcu_nocbs`: See the Scheduling and priorization section.
- `rcu_nocb_poll`: Make the kthreads poll for callbacks.
- `rcutree.kthread_prio=10`: Set the SCHED_FIFO priority of the RCU per-CPU kthreads.
- `skew_tick=1`: Helps to smooth jitter on systems with latency-sensitive applications running.
- `tsc=reliable`: Disables clocksource verification at runtime, as well as the stability checks done at bootup.

In the hardening system, the kernel has these parameters:

- `init_on_alloc=1`: Fill newly allocated pages and heap objects with zeroes.
- `init_on_free=1`: Fill freed pages and heap objects with zeroes.
- `slab_nomerge`: Disable merging of slabs with similar size.
- `pti=on`: Enable the control Page Table Isolation of user and kernel address spaces.
- `slub_debug=ZF`: Enable red zoning (Z) and zanity checks (F) on for all slabs (more details here).
- `randomize_kstack_offset=on`: Enable kernel stack offset randomization.
- `slab_common.usercopy_fallback=N`:
- `iommu=pt`: Get best performance using the SR-IOV (TODO).
- `security=yama`: Use the `yama` security module to enable at boot.
- `mce=0`: Disables the time in us to wait for other CPUs on machine checks.
- `rng_core.default_quality=500`: Set the value of the entropy for the system.
- `lsm=apparmor,lockdown,capability,landlock,yama,bpf`: Set the order of LSM initialization.

More details on the kernel's parameters here.