User initiated shutdown detection

Michele Tartara - Ganeti Core Team - Google
GanetiCon 2014 - September 2, 2014

The current state

- Ganeti is responsible for starting-stopping instances
  - `gnt-instance start`
  - `gnt-instance stop`
- Keeps track of the desired state of all instances.
- The watcher takes care of fixing those with a wrong state
  - i.e.: restart them

The issue

- Users frequently shut down instances from inside
  - Correct, clean shutdown
  - The user expects the instance to stay down
- Ganeti is not informed
  - The watcher restarts the instance
  - Unexpected behaviour
  - An error is shown

Design and implementation of the solution

Detect and act

- Allow Ganeti to detect the situation
- Act accordingly to a configuration value
  - Switch the instance to ADMIN_down
  - Or reboot it
    - ...but this time it is the expected behavior

How to use user-initiated shutdown detection?

- Enable it at cluster level
  - `gnt-cluster init/modify --user-shutdown=yes`
- and also as a hypervisor parameter (KVM clusters only)
  - `gnt-cluster init/modify --hypervisor-parameters kvm:user_shutdown=true`

Xen implementation (I)

- Xen knows why a domain is being shutdown
  - Crash vs. Shutdown
  - The information is usually "lost"
    - The domain is destroyed in both cases
    - Only recorded in the logs
      - Too much noise
      - Not meant for parsing
    - Initial though: patch xen to expose a callback
      - Too long before solution is widely available
      - Big effort

Xen implementation (II)

- Further investigation (thanks, Xen Developer Summit!)
- Change default `on_poweroff`, `on_shutdown`, `on_crash` behavior to `preserve`
  - Domain not automatically destroyed
  - Shutdown cause visible via `xm list`
  - Domain explicitly destroyed afterwards

Xen implementation (III)

- Actual implementation
  - Set `preserve` as behavior
  - Ganeti commands know this, and they explicitly destroy the domain after shutdown completion
  - The watcher takes care of other instances stopped but still around
    - Read shutdown cause
    - Act accordingly (depending on instance-specific config)
      - If required, change expected status of the instance in the config
    - Send `InstanceShutdown` jobs to the node to destroy the domain

Related changes

- Implementing this functionality also required some other minor changes
  - gnt-instance list, gnt-instance info need to properly show the state of an instance internally shutdown but not yet cleaned up
  - `ADMIN_down` vs. `USER_down`

KVM implementation

- Here comes a new daemon: KvmD
  - ONLY responsibly for dealing with this
  - Shuts down immediately if user shutdown detection is disabled
    - Don't be surprised if you see this in the logs ;-)
  - Communicates with KVM over QMP sockets
  - Monitors the socket for `POWERDOWN`, `SHUTDOWN`, `STOP` events

Thank You!

Questions?