3 Maintenance

There are a lot of tools to perform maintenance operations. Main tools we use are: SSH connection to KVM host and virsh CLI client, VM console access provided by virsh console command from hypervisor and virt-manager GUI from remote desktop client.

3.1 VM serial console

Into VM define a serial console:

```
# nano -w /etc/init/ttyS0.conf

# ttyS0 - getty
#
# This service maintains a getty on ttyS0 from the point the system is
# started until it is shut down again.

start on stopped rc RUNLEVEL=[2345]
stop on runlevel [!2345]

respawn
exec /sbin/getty -L 115200 ttyS0 xterm
```

On KVM host, verify in VM XML file is present serial tag:

```
<domain>
  ...
  <devices>
    ...
    <serial type='pty'>
      <source path='/dev/pts/7'/>
      <target port='0'/>
      <alias name='serial0'/>
    </serial>
    ...
  </devices>
  ...
</domain>
```

SSH into KVM host then access VM `tubu1` console:

```
$ virsh console tubu1
```

3.2 Virtual Machine Manager

Virtual Machine Manager is a GUI to manage KVM hosts and VMs.

Install Virt-Manager from packages:

```
# apt-get install virt-manager
```

Launch Virt-Manager and connect to remote KVM host using Hypervisor=QEMU/KVM, Method=SSH and Hostname=KVMHOSTIP.

From within GUI you can manage hypervisor storage, not network. Also you can manage existing VM, create new VM or remove existing one and access graphic console of VM.

NOTE: due a bug (http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=663931) you need ***`/etc/init.d/udev restart`*** *on KVM host if virt-manager client timeout.*