# VirtualBox

## Host

- Add “virtualbox.list” to /etc/apt/sources.list.d

```
sudo deb http://download.virtualbox.org/virtualbox/debian stretch
contrib
```

- Add Oracle VirtualBox public key:

```
wget https://www.virtualbox.org/download/oracle_vbox_2016.asc
sudo apt-key add oracle_vbox_2016.asc
wget https://www.virtualbox.org/download/oracle_vbox.asc
sudo apt-key add oracle_vbox.asc
```

- Install Virtualbox:

```
sudo apt-get update
sudo apt-get install virtualbox-6.0
```

- Download the VirtualBox Extension Pack. Double click the file from within the GUI to install.
- VirtualBox: Encryption of Disk Images
- Configuring VirtualBox autostart on Linux

## Guest

## Guest Additions

- To upgrade the Guest Additions after upgrading the host version, download the guest additions pack, close all virtual machines, and start virtualbox as root

```
sudo virtualbox
```

- After installing the Guest Additions, close virtualbox and restart it as user

## Guest Additions for a Debian 9 guest on a Debian 7 host

- Add the following line to /etc/apt/sources.list.d:

```
deb http://ftp.debian.org/debian stretch-backports main contrib
```

- Install virtualbox-guest-dkms, kernel headers, and, optionally, virtualbox-guest-x11 (for the graphical guest utilities):

```
sudo apt update
```

```
sudo apt install virtualbox-guest-dkms virtualbox-guest-x11 linux-
headers-$(uname -r)
```

- Reboot the guest system.
- More info: How to install VirtualBox Guest Additions in a Debian virtual machine

# Autostart and Autostop

There are several options to autostart (and autostop) virtual machines. For a server environment with a Debian host the 1. option through rc.local is preferrable.

1. through rc.local
2. through X desktop
3. through VirtualBox command line

## through rc.local

### startup

1. install rc.local as a systemd service on your debian server. This emulates the *SysVinit* behavior of rc.local as Debian changed the default init system to *systemd* with Debian Jessie 8
2. mount encrypted file systems through rc.local as explained in Encrypted partitions/folders with auto-mount
3. start vms through rc.local headless. Since rc.local runs as root you need to use runuser to start the vm as the user who created the vm. You may also chose to put the startvm code into a script and add execution of the script to rc.local:

```
runuser -l <user> -c "vboxmanage startvm <vm-name> --type headless"
```

### shutdown

1. put a symlink to the following script into /lib/systemd/system-shutdown:

```
runuser -l <user> -c "ssh <vm-name> 'sudo systemctl poweroff </dev/null
&>/dev/null &'"
while [ "`runuser -l <user> -c 'vboxmanage list runningvms'`" != "" ]
do
    echo waiting for VMs to shutdown
    sleep 5
done
```

2. allow <user> to execute command 'systemctl poweroff' in the vm (do not ssh into the VM as root - security!). Note that 'systemctl poweroff' is the systemd command, do not use *poweroff* or *shutdown* commands:

```
sudo visudo
```

```
<user> ALL=(root) NOPASSWD: /sbin/systemctl poweroff
```

3. the scripts must be run as root so they can be executed through rc.local
4. also, the server must be shut down with *systemctl* rather than *reboot* or *shutdown* for this to work

- How to run commands at shutdown on Linux
- How to Prevent Accidental Shutdown or Reboot on Linux Using systemd Custom Service

## through XFCE Desktop

### startup

- On the host, open **Settings → Session and Startup**, then add a new task under **Application Autostart** with the following command:

```
VBoxManage startvm <VM name>
```

### shutdown

- On the client, open **Settings → Power Manager**, then set the desired action under “When power button is pressed”

## through VirtualBox command line

1. Configuration

```
sudo vim /etc/default/virtualbox
VBOXAUTOSTART_DB=/etc/vbox
VBOXAUTOSTART_CONFIG=/etc/vbox/autostart.cfg

sudo vim /etc/vbox/autostart.cfg
default_policy = deny
# Create an entry for each user allowed to use autostart
username = {
  allow = true
  startup_delay = 10
}

sudo chgrp vboxusers /etc/vbox
sudo chmod 1775 /etc/vbox
sudo usermod -aG vboxusers username
```

2. Choose VMs to automatically start and stop

```
VBoxManage setproperty autostartdbpath /etc/vbox # first time a user
configures autostart only
VBoxManage modifyvm <vmname> --autostart-enabled <on|off>
```

```
VBoxManage modifyvm <vmname> --autostop-type
<disabled|savestate|poweroff|acpishutdown>
```

3. Restart VirtualBox autostart service

```
sudo service vboxautostart-service restart
```

## Start VM

```
vboxmanage startvm MachineName
```

## Stop VM

```
vboxmanage controlvm MachineName acpipowerbutton
```

In xfce desktop, make sure the action on Power Button is set in Settings –> Power Manager

## Stop VM before powering down host through NUT

Set different timeouts to the shutdown command used by NUT. For example, set +0 for the clients, and +1 for the host.

## Check running VMs

One of the following commands:

```
VBoxManage list runningvms
vboxmanage list vms --long | grep -e "Name:" -e "State:"
```

## Links

- How can I tell if a virtual machine is up from the command line?
- autostop-type
- Configuring VirtualBox autostart on Linux
- How to start VirtualBox machines automatically as a service on Windows