# Autostart VM on Boot

In this guide, we are going to learn how to autostart VirtualBox VMs on system boot on Linux. On a Linux system with VirtualBox installed, you can start VMs automatically during system boot.

# AutoStart VirtualBox VMs on System Boot on Linux

There are multiple ways in which you can configure your VirtualBox vms to automatically start on system boot.

A few of these ways that we will discuss in this guide include;

1. Using VirtualBox Autostart Service
2. Using Systemd service unit
3. Using system cronjobs

# Using VirtualBox Autostart Service

VirtualBox comes with a service called vboxautostart-service that makes it easy to automatically start virtual machines during system reboot.

In order to configure a VirtualBox VM to start on system boot on Linux, you need to activate the autostart service. The autostart service can be activated by setting two variables in /etc/default/virtualbox;

- **VBOXAUTOSTART_DB** – which defines the absolute path to the autostart database directory, usually the /etc/vbox.
- **VBOXAUTOSTART_CONFIG** – defines the path to the virtual machine autostart configuration.

These variables can be defined as;

```
VBOXAUTOSTART_DB=/etc/vbox
VBOXAUTOSTART_CONFIG=/etc/vbox/autostartvm.cfg
```

To place these variables in the /etc/default/virtualbox, run the command below;

```
echo -e "VBOXAUTOSTART_DB=/etc/vbox\nVBOXAUTOSTART_CONFIG=/etc/vbox/autostartvm.cfg" | sudo tee /etc/default/virtualbox
```

Define the virtual machine autostart configuration settings. The autostart configuration file contains options that controls how the virtual machine is auto started.

`sudo vim /etc/vbox/autostartvm.cfg`

```
default_policy = deny


kifarunix = {
    allow = true
    startup_delay = 10
}
```

- **default_policy** – defines whether to allow or deny the virtual machine autostart by default. In our example above, we denied any one from auto-starting the VM and explicitly allow one user, amos.
- **username** (kifarunix) – with the default deny policy, you can define the specific users that are allowed to autostart the virtual machine (allow = trues). You can also define how long to delay the VM startup. 10 seconds is used in this demo.

# Set Ownership of Database directory

The database directory, /etc/vbox, should be writable by the user to be used to start VMs automatically. To make it easy, you can simply add your user to vboxusers group and set the group ownership of the database directory to vboxusers group. After that, set the write permissions for the group. In this case, amos user is to be used to automatically start the virtual machine.

`sudo usermod -aG vboxusers kifarunix`
`sudo chgrp vboxusers /etc/vbox`
Assign the group write permissions on the autostart database directory.

`sudo chmod g+w /etc/vbox`
To shield the directory from being modified or deleted by other users except the owner or the root user, set sticky bit.

`sudo chmod +t /etc/vbox`

# Enable Virtual Machine Autostart

As a user, you can enable autostart for individual machines. This requires that you define the path to the database directory first.

```
VBoxManage setproperty autostartdbpath /etc/vbox/
```

Once that is done, you can now setup the virtual machine to automatically start on system boot.

```
vboxmanage modifyvm fedora30 --autostart-enabled on
```

Where fedora30 is the name of your virtual machine. You can also use UUID instead.

You can get the names/UUIDs of the vms by running the command below;

vboxmanage list vms Restart vboxautostart-service The configuration is now done. To effect the settings, you need to restart the vboxautostart-service.

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

Testing the Virtual Machine autostart To test that your virtual machine can actually autostart on system boot, reboot you system and check. When system boots, your VM should now be running.

If you which to disable the virtual machine autostart;

```
vboxmanage modifyvm fedora30 --autostart-enabled off
```

# Using Systemd service unit

Similarly, you can auto-start a VirtualBox by a creating a systemd service that starts the respective VM when system reboots.

To use this approach, just create the systemd unit service as follows (replace the name of the vm accordingly);

```
sudo vim /etc/systemd/system/fedora30-vm.service
```

```
[Unit]
Description=Autostart VirtualBox VM
After=network.target vboxdrv.service

[Service]
User=kifarunix
```

```
ExecStart=/usr/bin/vboxheadless -s fedora30
ExecStop=/usr/bin/VBoxManage controlvm fedora30 acpipowerbutton

[Install]
WantedBy=multi-user.target
```

Replace the User=kifarunix, with your correct username.

Reload the systemd daemon to read the new unit file:

```
sudo systemctl daemon-reload
```

Enable the service to start automatically on boot:

```
sudo systemctl enable vmname.service
```

Reboot the system and confirm if all is good.

# Autostart VirtualBox VM using System Cronjob

It is also possible to use cronjob to automatically start a virtualbox machine on system reboot.

Simply, as user with which you want to start the VirtualBox vm as, create a cronjob as follows;

```
crontab -e
```

Enter this line to start your VM. Replace the names accordingly;

```
@reboot sleep 60 && vboxheadless --startvm "fedora30" &
```

The command simply starts the vm 1 minute after the system boots.

That is all! You have learnt how to autostart VirtualBox VMs on system boot on Linux using the following methods;

Credits: https://kifarunix.com/autostart-virtualbox-vms-on-system-boot-on-linux

---

Revision #3
Created 2023-10-26 17:50:54 UTC by tinfoil-hat
Updated 2023-10-26 18:09:37 UTC by tinfoil-hat