Install WebVirtCloud KVM Web Dashboard on Ubuntu 20.04 | 18.04

By **Wekesa Collins** - September 28, 2022

Virtualization is the process of using software (Hypervisors) to create a virtual instance that emulates properties of the underlying hardware. This gives organizations to run and manage multiple virtual devices such as networks, storage and guest OS machines. Two main types of hypervisors are Type 1 Hypervisors and Type two Hypervisors.

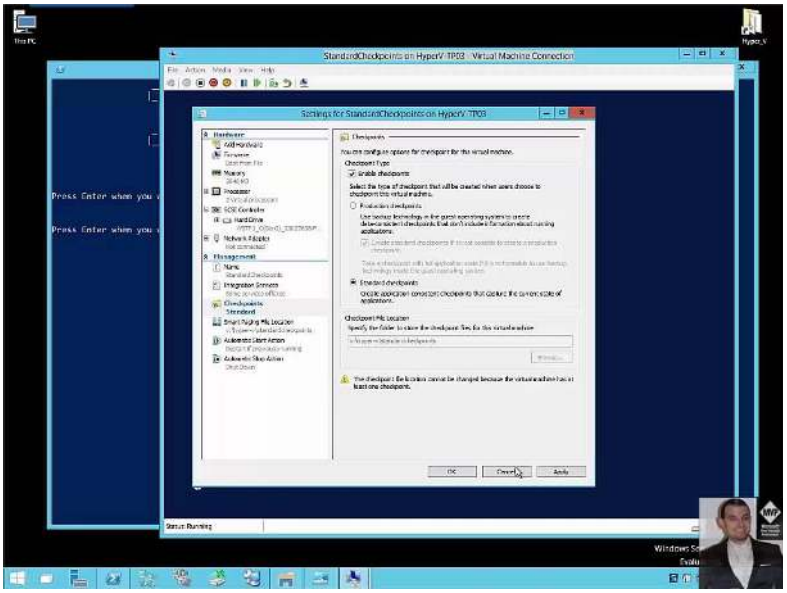
KVM (Kernel-Based Virtual Machine) is an example of a Type 1 Hypervisor. Guest OS interacts directly with the hardware no CPU is involved in between, unlike Type 2 Hypervisors. KVM is a Linux hypervisor that is pre-installed and only requires some modules to be installed for activation. VMs created with KVM can be managed either command line or GUI. Virt-Manager has been providing a good interface for management but the disadvantage is that you need access to a server either physically or SSH session. With gradual development in technology, now KVM VMs has web solution as a management platform. In this guide, am going to take you through WebVirtCloud.

WebVirtCloud is a Django (python) open-source software that provides a web interface with an admin view where one can create, deploy and manage virtual machines. Currently, KVM is the only Hypervisor that is supported on the platform. Has more interesting features including web console access to a virtual machine. In this work through, we are going to learn how to install, configure and manage virtual machines with WebVirtCloud.

Features of WebVirtCloud KVM Web Dashboard

- Manage multiple QEMU/KVM Hypervisors.
- QEMU/KVM Hypervisor management console.
- Provides browser console access to an instance.
- QEMU/KVM instance management by create, delete and even update
- Supports cloud-init datasource interface.
- Manage Hypervisor Datastore pools.
- Manage Hypervisor Networks.
- Provides web-based stats for both Hypervisor and instances.
- It is user based authorization and authentication.
- Users can add an SSH public key to the root in instance. (Verified on Ubuntu only)
- Users can change root passwords for instance. (Verified on Ubuntu only)

Installation of WebVirtCloud on Ubuntu 20.04 | 18.04



#HyperV vNext DEMO03-Production Checkpoints

Above was an introduction to WebVirtCloud just to equip you with a basic understanding. At this point, we want to perform the actual installation of WebVirtCloud. This guide will covers installation on both Ubuntu 20.04 and Ubuntu 18.04.

Step 1: Install KVM Hypervisor

Since Webvirtcloud is a web management console for QEMU/KVM, you will first need to install and configure KVM on your server. At this point, we are not going through KVM installation. In case you had not installed please check on this article:

- How To Install KVM Hypervisor on Ubuntu

Ensure that services are running before you to the next step.

```
$ sudo systemctl status libvirtd
● libvirtd.service - Virtualization daemon
        Loaded: loaded (/lib/systemd/system
/libvirtd.service; enabled; vendor preset: ena>
        Active: active (running) since Mon 2021-07-19
18:46:05 EAT; 14h ago
TriggeredBy: ● libvirtd-ro.socket
               ● libvirtd.socket
               ● libvirtd-admin.socket
        Docs: man:libvirtd(8)
               https://libvirt.org
   Main PID: 811 (libvirtd)
        Tasks: 20 (limit: 32768)
        Memory: 27.6M
        CGroup: /system.slice/libvirtd.service
               ├─ 811 /usr/sbin/libvirtd
               ├─1280 /usr/sbin/dnsmasq --conf-
file=/var/lib/libvirt/dnsmasq/default.con>
               └─1281 /usr/sbin/dnsmasq --conf-
file=/var/lib/libvirt/dnsmasq/default.con>
```

Step 2: Download and install necessary modules and dependecies.

Run the command below in the terminal to install python3, virtual environment for python, Nginx which will be used as a reverse proxy and other modules.

```
sudo apt update
sudo apt -y install git virtualenv python3-virtualenv
python3-dev python3-lxml libvirt-dev zlib1g-dev
libxslt1-dev nginx supervisor libsasl2-modules gcc
pkg-config python3-guestfs libsasl2-dev libldap2-dev
libssl-dev
```

Use git command to clone the WebVirtCloud repo from GitHub.

```
remote: Enumerating objects: 8724, done.
remote: Counting objects: 100% (696/696), done.
remote: Compressing objects: 100% (472/472), done.
remote: Total 8724 (delta 335), reused 441 (delta
215), pack-reused 8028
Receiving objects: 100% (8724/8724), 8.07 MiB |
129.00 KiB/s, done.
Resolving deltas: 100% (5826/5826), done
```

Navigate to the webvirtcloud folder. Copy all configuration files from **/webvirtcloud/settings.py.template** to **/webvirtcloud/settings.py**.

```
cd webvirtcloud
cp webvirtcloud/settings.py.template
webvirtcloud/settings.py
```

Generate your ***SECRET-KEY***. Edit settings.py file the paste your secrete key in the ***SECRET-KEY*** option. If you leave it blank you will encounter an error at the end.

```
$ vim ./webvirtcloud/settings.py

"""
Django settings for webvirtcloud project.
"""
```

```
BASE_DIR = Path(__file__).resolve().parent.parent
SECRET_KEY = "MyStrongSecretKEY"
DEBUG = False
ALLOWED_HOSTS = ["*"]
# Application definition
INSTALLED_APPS = [
        "django.contrib.auth",
        "django.contrib.contenttypes",
        "django.contrib.sessions",
        "django.contrib.messages",
        "django.contrib.staticfiles",
        "django_bootstrap5",
        "django_icons",
        "django_otp",
        "django_otp.plugins.otp_totp",
        "accounts",
        "admin",
        "appsettings",
        "computes",
```

Use the below commands to add WebVirtCloud configs to Nginx then exit the directory.

```
sudo cp conf/supervisor/webvirtcloud.conf
/etc/supervisor/conf.d
sudo cp conf/nginx/webvirtcloud.conf /etc/nginx
/conf.d
cd ..
```

Move **webvirtcloud** directory to **/srv/** then change ownership to **www-data**.

```
sudo mv webvirtcloud /srv
sudo chown -R www-data:www-data /srv/webvirtcloud
```

Switch to root user:

installed.

```
cd /srv/webvirtcloud
virtualenv -p python3 venv
```

Activate virtual environment.

```
source venv/bin/activate
```

Now install Django and the necessary tool with the below command.

```
# pip install -r conf/requirements.txt

Install Django.
Collecting Django==3.2.5
  Downloading Django-3.2.5-py3-none-any.whl (7.9 MB)
        |████████████████████████████████| 7.9 MB 57
kB/s
Collecting django_bootstrap5==2.0.1
  Downloading django_bootstrap5-2.0.1-py3-none-
any.whl (23 kB)
Collecting django-icons==4.0.0
  Downloading django_icons-4.0.0-py3-none-any.whl (45
kB)
        |████████████████████████████████| 45 kB 44
```

```
py3-none-any.whl (4.6 kB)
Collecting django-otp==1.0.6
  Downloading django_otp-1.0.6-py3-none-any.whl (58
kB)
        |████████████████                | 30 kB 68
kB/s eta 0:00:01
.....................................................................................................
.....................................................................
```

Multiple components of Django will be installed. Give it time to run to completion without interfering with the process. If you encounter a timeout error just rerun the command a process will pick from where it left.

Run migrate command to generate and create models (tables).

```
# python3 manage.py migrate

Operations to perform:
  Apply all migrations: accounts, admin, appsettings,
auth, computes, contenttypes, instances, logs,
otp_totp, sessions
Running migrations:
  Applying computes.0001_initial... OK
  Applying instances.0001_initial... OK
  Applying instances.0002_permissionset... OK
  Applying instances.0003_auto_20200615_0637... OK
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying accounts.0001_initial... OK
  Applying accounts.0002_permissionset... OK
  Applying accounts.0003_auto_20200604_0930... OK
```

```
contenttypes.0002_remove_content_type_name... OK
  Applying
auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length...
OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying
auth.0007_alter_validators_add_error_messages... OK
  Applying
auth.0008_alter_user_username_max_length... OK
  Applying
auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length...
OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_auto_20200609_0830... OK
  Applying appsettings.0001_initial... OK
  Applying appsettings.0002_auto_20200527_1603... OK
  Applying appsettings.0003_auto_20200615_0637... OK
  Applying appsettings.0004_auto_20200716_0637... OK
  Applying appsettings.0005_auto_20200911_1233... OK
  Applying
auth.0012_alter_user_first_name_max_length... OK
  Applying computes.0002_auto_20200529_1320... OK
  Applying computes.0003_auto_20200615_0637... OK
  Applying instances.0004_auto_20200618_0817... OK
  Applying instances.0005_flavor... OK
  Applying instances.0006_addFlavors... OK
  Applying instances.0007_auto_20200624_0821... OK
  Applying instances.0008_auto_20200708_0950... OK
  Applying instances.0009_auto_20200717_0524... OK
  Applying logs.0001_initial... OK
  Applying logs.0002_auto_20200615_0637... OK
```

```
  Applying sessions.0001_initial... OK
* Creating default admin user
! SHOW_PROFILE_EDIT_PASSWORD is found inside
settings.py
* Applying permission can_change_password for all
users
! Warning!!! Setting to True for all users
! Don`t forget to remove the option from settings.py
* Migrating can_clone_instaces user attribute to
permission
* Applying permission passwordless_console for all
users
```

Change ownership of **/srv/webvirtcloud** to **www-data:www:data**. Delete the Nginx default file from the enabled site. This is to ensure that the Nginx test file is loaded to cause a bug.

```
sudo chown -R www-data:www-data /srv/webvirtcloud
sudo rm /etc/nginx/sites-enabled/default
```

Restart services to apply changes.

```
sudo systemctl restart nginx
sudo systemctl restart supervisor
```

Run the below command to set up KVM and Libvirt.

```
sudo wget -O - https://bit.ly/36baWUu | sudo sh
```

Add **www-data** user to kvm.

```
sudo adduser www-data kvm
```

Step 3: Accessing WebVirtCloud in Browser.

Open your browser, enter your server_IP or domain name to access WebVirtCloud.

HTTP://SERVER_IP/

Use the credentials below to login as your default login. Consider changing

```
Password : admin
```

Step 4: Creating Computes, Instances, Storages and other Configurations.

When you login to WebVirtCloud, this is the page that you see.

Page is blank because there are no computes and instances created. In the next steps, we are going to create computes and instances.

Create a compute.

To create a compute, navigate to compute menu and click. on the new page click local to add a compute.

Add Storages to Compute

Click on compute, list of computes created will appear. Click on the name of the compute where you want to add storage. In the menu given, select storage, choose the type of storage you want to create.

Add Volumes to Storage

Create virtual disks which can be used as default installation volumes. You can as well adjust later during installation to meet your desired capacity.

In your terminal, run these commands.

```
$ sudo qemu-img create -f qcow2 /var/lib/libvirt
/storage/small.qcow 10G

Formatting '/var/lib/libvirt/storage/small.qcow',
fmt=qcow2 size=21474836480 cluster_size=65536
lazy_refcounts=off refcount_bits=16

$ sudo qemu-img create -f qcow2 /var/lib/libvirt
/storage/medium.qcow 20G

Formatting '/var/lib/libvirt/storage/medium.qcow',
fmt=qcow2 size=53687091200 cluster_size=65536
lazy_refcounts=off refcount_bits=16

$ sudo qemu-img create -f qcow2 /var/lib/libvirt
/storage/large.qcow 30G

Formatting '/var/lib/libvirt/storage/large.qcow',
fmt=qcow2 size=107374182400 cluster_size=65536
lazy_refcounts=off refcount_bits=16
```

Create Linux Instance

First of all, download the Linux installation file to **iso storage** created above

default.

You can use the already customized specifications under **Flavor**. For this guide, We will select customize to create new specifications from scratch.

Give your instance a name. Choose the number of CPUs and RAM to be allocated. Under HDD, select your storage (primary-storage) and virtual disk under add image…. Finally, add network then create.

Navigate to settings then Disk. Add iso image as shown above then click on the mount. To resize disk size use Resize menu tab. When done, power on the instance.

To view the instance console, click on the instance menu tab. In the list shown, select the eye icon on the VM. Below is a console like the one you should see.

Install Windows Virtual Machine.

To run windows install with WebVirtCloud, additional driver software is needed to give windows the capability to see the installation target (storage). Download VirtIO drivers iso file with the command below. Make sure to copy or move file to iso storage so that the file can be attached to CD-ROM.

```
wget https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/stable-virtio/virtio-win.iso
```

Now create a windows instance.

Attach and mount both windows iso file and virtio.win.iso.

Power on the VM to start the installation process. Go to the console to complete the installation. When you reach this step where it says couldn't find any drivers, click on load to load drivers in virtio.win.iso in CD-ROM-2.

Click **OK** on the prompt screen below.

On the below screen output, select a driver that suits your version. I will select **w10** since **Windows 10** is the version am installing.

Give it time to install drivers. Now the error is clear and the storage disk is visible.

Create new partition then click next to continue with installation

Below is a list of active and running instances that we have created.

Conclusion

WebVirtCloud as web management portal/console. We have seen that indeed web interface makes it simple and easier to run and manage instances with infrastructures without guessing or any doubt of output unlike doing it on command line. Cheers to Django for greater contribution and making cloud environment more friendly and less complex.

You can also read on ;

How To Install KVM Hypervisor on Ubuntu

How To Install VirtualBox on Rocky Linux 8

How To Install Clear Linux on VirtualBox / VMWare Workstation

Video Courses to Learn Linux System Administration:

- Linux Mastery: Master the Linux Command Line in 11.5 Hours
- Complete Linux Training Course to Get Your Dream IT Job
- Learn Linux in 5 Days and Level Up Your Career
- Linux Administration Bootcamp: Go from Beginner to Advanced
- Complete Linux Bash Shell Scripting with Real Life Examples
- Linux Shell Scripting: A Project-Based Approach to Learning

Wekesa Collins

in