|  |
| --- |
| **VyOS cloud-init**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#vyos-cloud-init) |

|  |
| --- |
| Cloud and virtualized instances of VyOS are initialized using the industry-standard cloud- |

|  |
| --- |
| init. Via cloud-init, the system performs tasks such as injecting SSH keys and configuring |

|  |
| --- |
| the network. In addition, the user can supply a custom configuration at the time of |

|  |
| --- |
| instance launch. |

**Config Sources**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#config-sources)

VyOS support three types of config sources.

|  |  |  |
| --- | --- | --- |
|    | Metadata - Metadata is sourced by the cloud platform or hypervisor. In some clouds, there is implemented as an HTTP endpoint at http://169.254.169.254 .   |  | | --- | |  |   Network configuration - This config source informs the system about the network |

settings like IP addresses, routes, DNS. Available only in several cloud and virtualization platforms.

 User-data - User-data is specified by the user. This config source offers the ability to insert any CLI configuration commands into the configuration before the first boot.

**User-data**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#user-data)

Major cloud providers offer a means of providing user-data at the time of instance launch. It can be provided as plain text or as base64-encoded text, depending on cloud provider. Also, it can be compressed using gzip, which makes sense with a long configuration commands list, because of the hard limit to ~16384 bytes for the whole user-data.

The easiest way to configure the system via user-data is the Cloud-config syntax described below.

**Cloud-config modules**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#cloud-config-modules)

In VyOS, by default, enables only two modules:

|  |  |  |
| --- | --- | --- |
|  | |  | | --- | |  |   write\_files - this module allows to insert any files into the filesystem before the first |

boot, for example, pre-generated encryption keys, certificates, or even a

|  |  |
| --- | --- |
| whole config.boot file. The format is described in the cloudinit documentation Cloud-   |  | | --- | |  | |

[init-write\_files](https://cloudinit.readthedocs.io/en/latest/topics/examples.html#writing-out-arbitrary-files).

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | | --- | |  |   vyos\_userdata - the module accepts a list of CLI configuration commands in  a vyos\_config\_commands section, which gives an easy way to configure the system   |  | | --- | |  | |

during deployment.

**cloud-config file format**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#cloud-config-file-format)

|  |  |  |  |
| --- | --- | --- | --- |
| A cloud-config document is written in YAML. The file must begin with #cloud-config line. The   |  | | --- | |  |   only supported top-level keys are vyos\_config\_commands and write\_files . The use of these keys   |  | | --- | |  |  |  | | --- | |  | |

is described in the following two sections.

**Initial Configuration**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#initial-configuration)

|  |  |
| --- | --- |
| The key used to designate a VyOS configuration is vyos\_config\_commands . What follows is   |  | | --- | |  | |

VyOS configuration using the “set-style” syntax. Both “set” and “delete” commands are

supported.

Commands requirements:

|  |  |
| --- | --- |
|  | One command per line. |
|  | If command ends in a value, it must be inside single quotes. |
|  | A single-quote symbol is not allowed inside command or value. |
| The commands list produced by the show configuration commands command on a VyOS router   |  | | --- | |  | | |

should comply with all the requirements, so it is easy to get a proper commands list by copying it

from another router.

The configuration specified in the cloud-config document overwrites default configuration

values and values configured via Metadata.

Here is an example cloud-config that appends configuration at the time of first boot.

|  |
| --- |
| *#cloud-config*  **vyos\_config\_commands**:  -set system host-name 'vyos-prod-ashburn'  -set service ntp server 1.pool.ntp.org  -set service ntp server 2.pool.ntp.org  -delete interfaces ethernet eth1 address 'dhcp'  -set interfaces ethernet eth1 address '192.0.2.247/24'  -set protocols static route 198.51.100.0/24 next-hop '192.0.2.1' |

**System Defaults/Fallbacks**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#system-defaults-fallbacks)

These are the VyOS defaults and fallbacks.

|  |  |  |  |
| --- | --- | --- | --- |
|     | SSH is configured on port 22.   |  | | --- | |  |   vyos / vyos credentials if no others specified by data source.   |  | | --- | |  |   DHCP on first Ethernet interface if no network configuration is provided. |

All of these can be overridden using the configuration in user-data.

**Command Execution at Initial Boot**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#command-execution-at-initial-boot)

VyOS supports the execution of operational commands and linux commands at initial boot. This

|  |  |
| --- | --- |
| is accomplished using write\_files to certain files in the /opt/vyatta/etc/config/scripts directory.   |  | | --- | |  | |

Commands specified in opt/vyatta/etc/config/scripts/vyos-preconfig-bootup.script are executed prior to configuration. The /opt/vyatta/etc/config/scripts/vyos-postconfig-bootup.script file contains commands to be executed after configuration. In both cases, commands are executed as the root user.

Note that the /opt/vyatta/etc/config is used instead of the /config/scripts directory referenced in the [Command Scripting](https://docs.vyos.io/en/latest/automation/command-scripting.html#command-scripting) section of the documentation because the /config/script directory isn’t

|  |  |
| --- | --- |
| mounted when the write\_files module executes.   |  | | --- | |  | |

The following example shows how to execute commands after the initial configuration.

|  |
| --- |
| *#cloud-config*  **write\_files**:  -**path**:/opt/vyatta/etc/config/scripts/vyos-postconfig-bootup.script **owner**:root:vyattacfg   **permissions**:'0775'   **content**:|   #!/bin/vbash   source /opt/vyatta/etc/functions/script-template   filename=/tmp/bgp\_status\_`date +"%Y\_%m\_%d\_%I\_%M\_%p"`.log run show ip bgp summary >> $filename |

If you need to gather information from linux commands to configure VyOS, you can execute commands and then configure VyOS in the same script.

The following example sets the hostname based on the instance identifier obtained from the EC2 metadata service.

|  |
| --- |
| *#cloud-config*  **write\_files**:  -**path**:/opt/vyatta/etc/config/scripts/vyos-postconfig-bootup.script   **owner**:root:vyattacfg   **permissions**:'0775'   **content**:|   #!/bin/vbash   source /opt/vyatta/etc/functions/script-template   hostname=`curl -s http://169.254.169.254/latest/meta-data/instance-id` configure   set system host-name $hostname   commit   exit |

**NoCloud**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#nocloud)

Injecting configuration data is not limited to cloud platforms. Users can employ the NoCloud data source to inject user-data and meta-data on virtualization platforms such as VMware, Hyper-V and KVM.

While other methods exist, the most straightforward method for using the NoCloud data source is creating a seed ISO and attaching it to the virtual machine as a CD drive. The volume must be formatted as a vfat or ISO 9660 file system with the label “cidata” or “CIDATA”.

Create text files named user-data and meta-data. On linux-based systems, the mkisofs utility can be used to create the seed ISO. The following syntax will add these files to the ISO 9660 file system.

|  |
| --- |
| mkisofs -joliet -rock -volid "cidata" -output seed.iso meta-data user-data |

The seed.iso file can be attached to the virtual machine. As an example, the method with KVM to attach the ISO as a CD drive follows.

|  |
| --- |
| $ virt-install -n vyos\_r1 \  --ram 4096 \  --vcpus 2 \  --cdrom seed.iso \  --os-type linux \  --os-variant debian10 \  --network network=default \  --graphics vnc \  --hvm \  --virt-type kvm \  --disk path=/var/lib/libvirt/images/vyos\_kvm.qcow2,bus=virtio \ --import \  --noautoconsole |

For more information on the NoCloud data source, visit its [page](https://cloudinit.readthedocs.io/en/latest/reference/datasources/nocloud.html) in the cloud-init documentation.

**Troubleshooting**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#troubleshooting)

If you encounter problems, verify that the cloud-config document contains valid YAML. Online resources such as <https://www.yamllint.com/> provide a simple tool for validating YAML.

cloud-init logs to /var/log/cloud-init.log. This file can be helpful in determining why the configuration varies from what you expect. You can fetch the most important data filtering

|  |  |
| --- | --- |
| output for vyos keyword:   |  | | --- | |  | |

|  |
| --- |
| sudo grep vyos /var/log/cloud-init.log |

**Cloud-init on Proxmox**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#cloud-init-on-proxmox)

Before starting, please refer to cloud-init [network-config-docs](https://cloudinit.readthedocs.io/en/latest/topics/network-config.html) in order to know how to import user and network configurations.

Most important keys that needs to be considered:

 VyOS configuration commands are defined in user-data file.

 Networking configurations shouldn’t be passed in user-data file.

 If no networking configuration is provided, then dhcp client is going to be enabled on first interface. Bare in mind that this configuration will be inyected at an OS level, so don’t expect to find dhcp client configuration on vyos cli. Because of this behavior, in next example lab we will disable dhcp-client configuration on eth0.

Also, this lab considers:

 Proxmox IP address: **192.168.0.253/24**

 Storaged used: volume local, which is mounted on directory **/var/lib/vz**, and contains all type of content, including snippets.

 Remove default dhcp client on first interface, and load other configuration during first boot, using cloud-init.

**Generate qcow image**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#generate-qcow-image)

A VyOS qcow image with cloud-init options is needed. This can be obtained using [vyos-vm-images](https://github.com/vyos/vyos-vm-images) repo. After clonning the repo, edit the file **qemu.yml** and comment the **download-iso** role.

In this lab, we are using 1.3.0 VyOS version and setting a disk of 10G. Download VyOS .iso file

|  |  |
| --- | --- |
| and save it as /tmp/vyos.iso . Command used for generating qcow image:   |  | | --- | |  | |

|  |
| --- |
| sudoansible-playbookqemu.yml-edisk\_size=10**\** -eiso\_local=/tmp/vyos.iso-egrub\_console=serial-evyos\_version=1.3.0**\**-ecloud\_init=true-ecloud\_init\_ds=NoCloud |
| File generated with previous command: /tmp/vyos-1.3.0-cloud-init-10G-qemu.qcow2   |  | | --- | |  | |

Now, that file needs to be copied to proxmox server:

|  |
| --- |
| sudoscp/tmp/vyos-1.3.0-cloud-init-10G-qemu.qcow2root@192.168.0.253:/tmp/ |

**Prepare cloud-init files**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#prepare-cloud-init-files)

In Proxmox server three files are going to be used for this setup:

|  |  |
| --- | --- |
|  | **network-config**: file that will indicate to avoid dhcp client on first interface. |
|  | **user-data**: includes vyos-commands. |
|  | **meta-data**: empty file (required). |
| In this lab, all files are located in /tmp/ . So, before going on, lets move to that directory:   |  | | --- | |  | | |

|  |
| --- |
| cd/tmp/ |
| **user-data** file must start with #cloud-config and contains vyos-commands. For example:   |  | | --- | |  | |

|  |
| --- |
| #cloud-config  vyos\_config\_commands:  - set system host-name 'vyos-BRAS'  - set service ntp server 1.pool.ntp.org  - set service ntp server 2.pool.ntp.org  - delete interfaces ethernet eth0 address 'dhcp'  - set interfaces ethernet eth0 address '198.51.100.2/30'  - set interfaces ethernet eth0 description 'WAN - ISP01'  - set interfaces ethernet eth1 address '192.168.25.1/24'  - set interfaces ethernet eth1 description 'Comming through VLAN 25' - set interfaces ethernet eth2 address '192.168.26.1/24'  - set interfaces ethernet eth2 description 'Comming through VLAN 26' - set protocols static route 0.0.0.0/0 next-hop '198.51.100.1' |

**network-config** file only has configuration that disables the automatic dhcp client on first interface.

Content of network-config file:

|  |
| --- |
| version: 2  ethernets:  eth0:  dhcp4: false |

|  |
| --- |
| dhcp6: false |

Finaly, file **meta-data** has no content, but it’s required.

**Create seed.iso**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#create-seed-iso)

|  |  |
| --- | --- |
| Once the three files were created, it’s time to generate the seed.iso image, which needs to be   |  | | --- | |  | |

mounted to the new VM as a cd.

|  |  |
| --- | --- |
| Command for generating seed.iso   |  | | --- | |  | |

|  |
| --- |
| mkisofs-joliet-rock-volid"cidata"-outputseed.isometa-data**\** user-datanetwork-config |

**NOTE**: be carefull while copying and pasting previous commands. Doble quotes may need to be corrected.

**Creating the VM**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#creating-the-vm)

Notes for this particular example, that may need to be modified in other setups:

|  |  |
| --- | --- |
|  | VM ID: in this example, VM ID used is 555. |
|  | VM Storage: local volume is used.   |  | | --- | |  | |
|  | ISO files storage: local volume is used for .iso file storage. In this   |  | | --- | |  |  |  | | --- | |  | |
| scenario local volume type is set to **directory**, abd attached to /var/lib/vz .   |  | | --- | |  |  |  | | --- | |  | | |
|  | VM Resources: these parameters can be modified as needed. |
| |  | | --- | |  |   seed.iso was previously created in directory /tmp/ . It’s necessary to move it   |  | | --- | |  |   to /var/lib/vz/template/iso   |  | | --- | |  | | |

|  |
| --- |
| mv/tmp/seed.iso/var/lib/vz/template/iso/ |

On proxmox server:

|  |
| --- |
| ## Create VM, import disk and define boot order  qm create 555 --name vyos-1.3.0-cloudinit --memory 1024 --net0 virtio,bridge=vmbr0 qm importdisk 555 vyos-1.3.0-cloud-init-10G-qemu.qcow2 local  qm set 555 --virtio0 local:555/vm-555-disk-0.raw  qm set 555 --boot order=virtio0 |

|  |
| --- |
| ## Import seed.iso for cloud init  qm set 555 --ide2 media=cdrom,file=local:iso/seed.iso |

|  |
| --- |
| ## Since this server has 1 nic, lets add network intefaces (vlan 25 and 26) qm set 555 --net1 virtio,bridge=vmbr0,firewall=1,tag=25  qm set 555 --net2 virtio,bridge=vmbr0,firewall=1,tag=26 |

**Power on VM and verifications**[](https://docs.vyos.io/en/latest/automation/cloud-init.html#power-on-vm-and-verifications)  
From cli or GUI, power on VM, and after it boots, verify configuration