Install on Google Cloud Platform

For supported software information, click here.

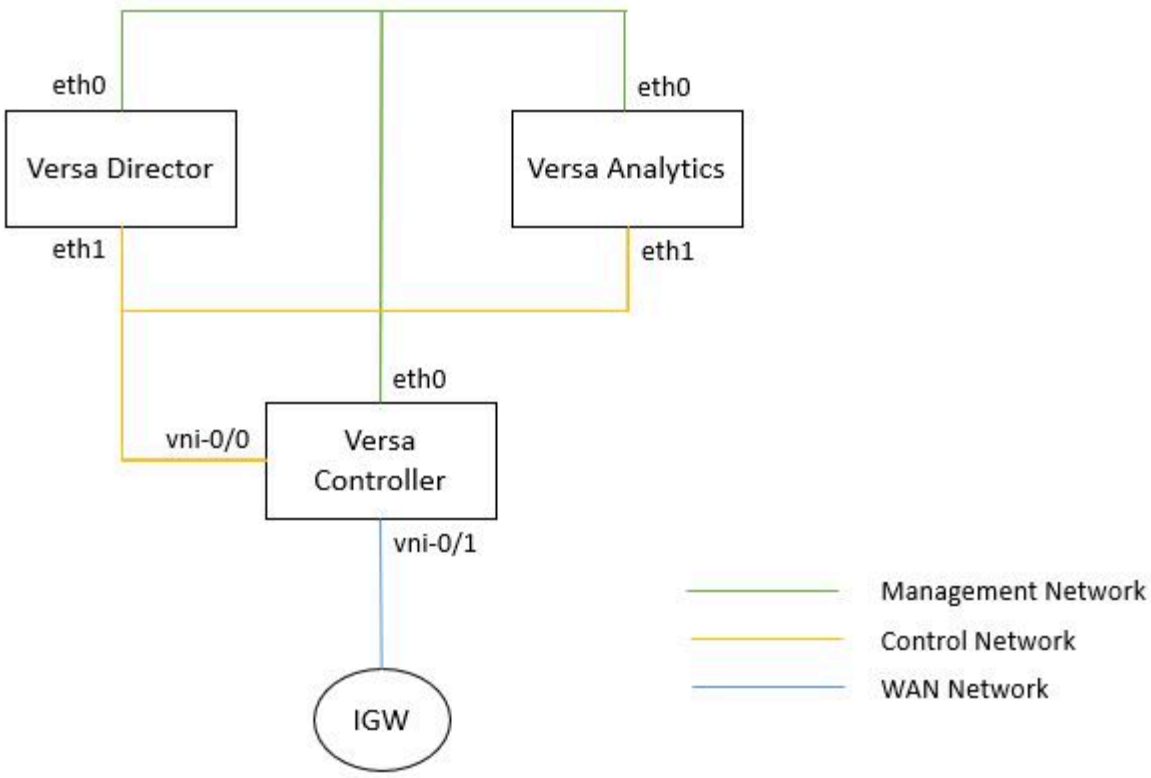
To install the Versa headend components on the Google Cloud Platform in standalone mode, you upload the Versa software images to the Google Cloud portal, and then you use the Terraform templates to create virtual machines (VMs) for the Versa headend components. You obtain the Terraform template files the public repository hosted by Versa Networks, at https://gitlab.com/versa-networks/terraform-templates-gcp-he (for headend deployments) and https://gitlab.com/versa-networks/devops/-/tree/master/terraform/automation/GCP (for branch deployments).

This article describes how to use Terraform templates to automatically create VMs for Versa headend components on Google Cloud Platform.

Headend Topology Created by Terraform Template

The following figure illustrates the standalone headend topology created by the Terraform template. The Terraform template provisions networks and Versa headend instances, and it assigns networks to the instances.

- For MGMT_NETWORK, the management IP address is assigned using this network. The public IP addresses assigned are associated with the three ports to the three Versa headend components.
- For Director-Controller-VAN_Network, the Director southbound, Controller northbound, and Analytics southbound IP addresses are assigned using this network.
- For Controller-Branch_Network, the Controller southbound IP address is assigned using this network. This IP address is used to connect to the branch. The public IP address is also assigned to the Controller WAN port.



Before You Begin

- Obtain the Terraform template files from Versa Networks Customer Support. When requesting the template, specify whether you are using a standalone headend topology.
- Install Terraform on your system. For information, see the Download Terraform article on the Terraform website.
- Set up Terraform access to Google cloud service account to enable Terraform to provision resources into Google Cloud Platform. For information, see the Google Cloud Platform articles on the Google cloud website.
- Create a service account in Google cloud account.
- Create service account keys in JSON file format for the service account created. For information, see the Create Service Account Keys article on the Google cloud website.
- Obtain the JSON file for the service account to authenticate Google cloud platform for logging in to Terraform.
- Obtain images for the Versa headend components—Versa Director, Versa Analytics, and Versa Controller—from Versa Networks Customer Support.

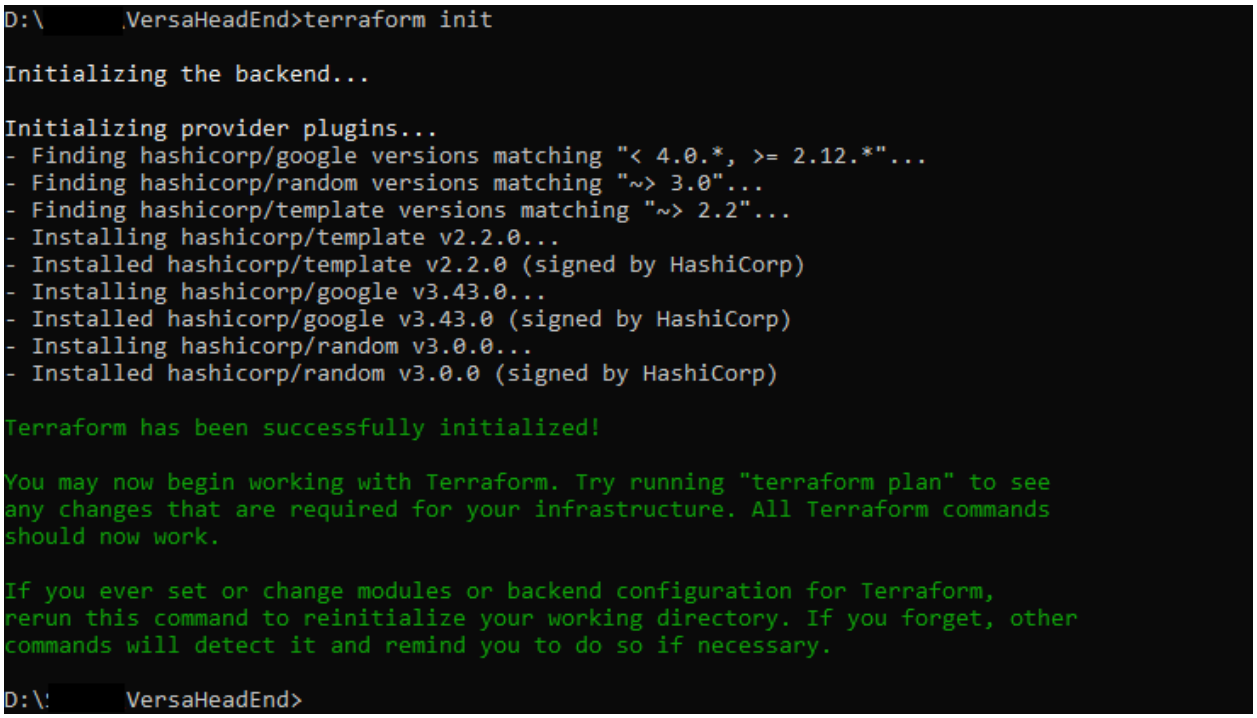
Create the VMs

To create VMs using Terraform templates:

1. Contact Versa Customer Support to obtain the Terraform template. When requesting the template, specify whether you are using a standalone or a redundant headend topology.

2. When you receive the folder that contains the template files, save them to the local system on which Terraform is installed. The template folder contains the files needed to deploy the Versa headend VM resources on Google Cloud Platform. The table at the end of this section describes each of the files.
3. If you want to make changes to any of the template files, for example, if you want to run another instance of Terraform, make a copy of the original folder and make your changes in the copy. It is recommended that you do not make any changes to the files in the original template folder.
4. Open a console window on the local system where Terraform was installed. Go to the folder where all the required files are placed from the console window.
5. Initialize Terraform. The initialization process downloads the Terraform plugins that are required to run the template.

   > ~$ **terraform init**



6. Display all the resources provisioned as part of the template.

   > ~$ **terraform plan**

```
D:\      VersaHeadEnd>terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

data.template_file.user_data_director: Refreshing state...

------------------------------------------------------------------------

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create
 <= read (data resources)

Terraform will perform the following actions:

  # data.template_file.user_data_van will be read during apply
  # (config refers to values not yet known)
 <= data "template_file" "user_data_van"  {
      + id       = (known after apply)
      + rendered = (known after apply)
      + template = <<~EOT
            #!/bin/bash
            log_path="/etc/bootLog.txt"
            if [ -f "$log_path" ]
            then
                echo "Cloud Init script already ran earlier during first time boot.." >> $log_path
            else
                touch $log_path
            SSHKey="${sshkey}"
            KeyDir="/home/versa/.ssh"
            KeyFile="/home/versa/.ssh/authorized_keys"
            DirIP="${dir_mgmt_ip}"
            Address="Match Address $DirIP"
            SSH_Conf="/etc/ssh/sshd_config"
            UBUNTU_RELEASE="$(lsb_release -cs)"
            echo "Starting cloud init script..." > $log_path
```

7. Run the template to deploy all the VM resources on Google Cloud Platform.

   ```
   ~$ terraform apply
   ```

```
D:\     _ .VersaHeadEnd>terraform apply
data.template_file.user_data_director: Refreshing state...

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create
 <= read (data resources)

Terraform will perform the following actions:

  # data.template_file.user_data_van will be read during apply
  # (config refers to values not yet known)
 <= data "template_file" "user_data_van"  {
      + id       = (known after apply)
      + rendered = (known after apply)
      + template = <<~EOT
            #!/bin/bash
            log_path="/etc/bootLog.txt"
            if [ -f "$log_path" ]
            then
                echo "Cloud Init script already ran earlier during first time boot.." >> $log_path
            else
                touch $log_path
            SSHKey="${sshkey}"
            KeyDir="/home/versa/.ssh"
            KeyFile="/home/versa/.ssh/authorized_keys"
            DirIP="${dir_mgmt_ip}"
            Address="Match Address $DirIP"
            SSH_Conf="/etc/ssh/sshd_config"
            UBUNTU_RELEASE="$(lsb_release -cs)"
            echo "Starting cloud init script..." > $log_path
```

The following table describes the content of each Terraform template file and the actions performed by each file.

Filename	Description or Action
main.tf	• Provision four virtual networks (VPCs). To change the IP prefix, edit the terraform.tfvars file. • Provision four subnetworks, one in each VPC with the IP CIDR range provided in the variable f terraform.tfvars file. ◦ 10.231.1.0/24 subnet is for management of all the headend instances. ◦ 10.231.2.0/24 subnet is the control network. This subnetwork is used for Director downstre (northbound) ports, and Analytics downstream (southbound ports. ◦ 10.231.3.0/24 subnet is the WAN network. This subnetwork is used for Controller downstr connectivity. ◦ 10.231.4.0/24 subnet is the LAN network. This subnetwork is used for branch and client c • Assign a public IP address on the management port of all management port instances. • Assign a static public IP address for the Controller WAN port. • Provision a route (a user-defined route) for the Controller address to use as a gateway for the used to send Netconf traffic originating from the Director. • Provision a network security group and add all the firewall rules required to set up the headend • Install a Director instance and run the cloud-init script to: ◦ Update the /etc/network/interface file.

Filename	Description or Action
	◦ Update the /etc/hosts and /etc/hostname file. ◦ Add the SSH key for the Administrator user. ◦ Generate new certificates. ◦ Run the vnms-startup script in non-interactive mode. • Install a Controller instance and run the cloud-init script to: ◦ Update the /etc/network/interface file. ◦ Add the SSH key for the Administrator user • Install an Analytics instance and run the cloud-init script to: ◦ Update the /etc/network/interface file. ◦ Update the /etc/hosts and /etc/hostname file. ◦ Add the SSH key for the Versa user.
var.tf	Provide definitions of all variables defined and used in the template. Do not make any changes to t
terraform.tfvars	User-defined input variables that are used to populate the Terraform templates. Edit this file to set • credentials_file—Credential file path obtained as part of the initial setup process for the service • project_id—Google cloud project identifier in which to deploy the headend. • region—Region where the user wants to deploy Versa headend setup. • zone—Zone from the selected region to create all the resources that belong to Versa headend • ssh_key—SSH public key. This key is required to log in to the headend instances. To generate key generator command. • ip_cidr_range—Address space information to create the subnetwork in Google in each networ created with address space 10.231.1.0/24, 10.231.2.0/24, 10.231.3.0/24, and 10.231.4.0/24. • destination_ip_range—Destination IP range to add a custom route to use controller address as address for the netconf traffic originates from Director instance. This is an SD-WAN overlay ro • labels—Labels and tags to add to the instances. • versa_director_image—Filrname of the Director image. • versa_flexvnf_image—Filename of the Controller image. • versa_van_image—Filename of the Analytics image. • hostname_director—Hostname of the Director instance. The default is versa-director-gce. • hostname_analytics—Hostname of the Analytics instance. The default is versa-Analytics-gce. • machine_type_dir—Instance type and size used to provision the Director instance. The default • machine_type_controller—Instance type and size used to provision the Controller instance. Th • machine_type_van—Instance type and size used to provision the Analytics instance. The defa

Filename	Description or Action
output.tf	Output parameters, including instance ID and public IP address, for all instances. Do not make any
director.sh	Bash script that runs as part of cloud-init script on the Versa Director instance. Do not make any ch
controller.sh	Bash script that runs as part of cloud-init script on Versa Controller instance. Do not make any char
van.sh	Bash script that runs as part of cloud-init script on Versa Analytics instance. Do not make any chan
credentials.json	Authentication information. Place this file in the same folder where you download all authentication

Supported Software Information

Releases 20.2 and later support all content described in this article.

Additional Information

Hardware and Software Requirements for Headend
Headend Initial Configuration
Headend Overview
Headend Verification