Install on Azure

For supported software information, click here.

To install the Versa headend components on Microsoft Azure, you upload the Versa software images to the Azure portal and then you create virtual machines (VMs) for the Versa headened components. Versa Networks provides Terraform template files that you can use to automate the deployment of these components. You can also use other methods to deploy the components, such as Azure Portal, Azure CLI, and Azure ARM templates.

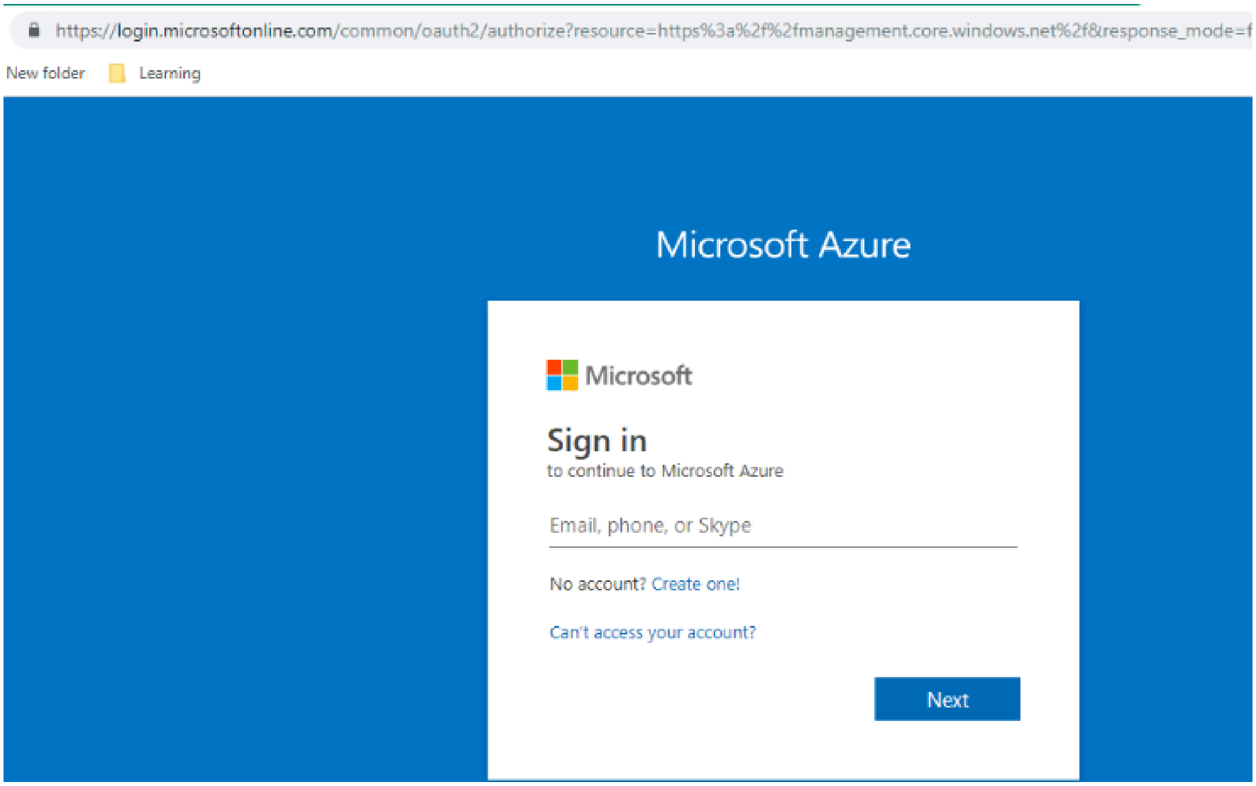
Upload Versa Images to Azure Portal

Before you upload the Versa software images to the Azure portal, ensure that you have the following:

- Valid Microsoft Azure subscription
- Administrator or co-administrator credentials to the Azure management portal

To upload the Versa software images to the Azure portal:

1. Log in to the Azure portal.



https://login.microsoftonline.com/common/oauth2/authorize?resource=https%3a%2f%2fmanagement.core.windows.net%2f&response_mode=f

Sign in

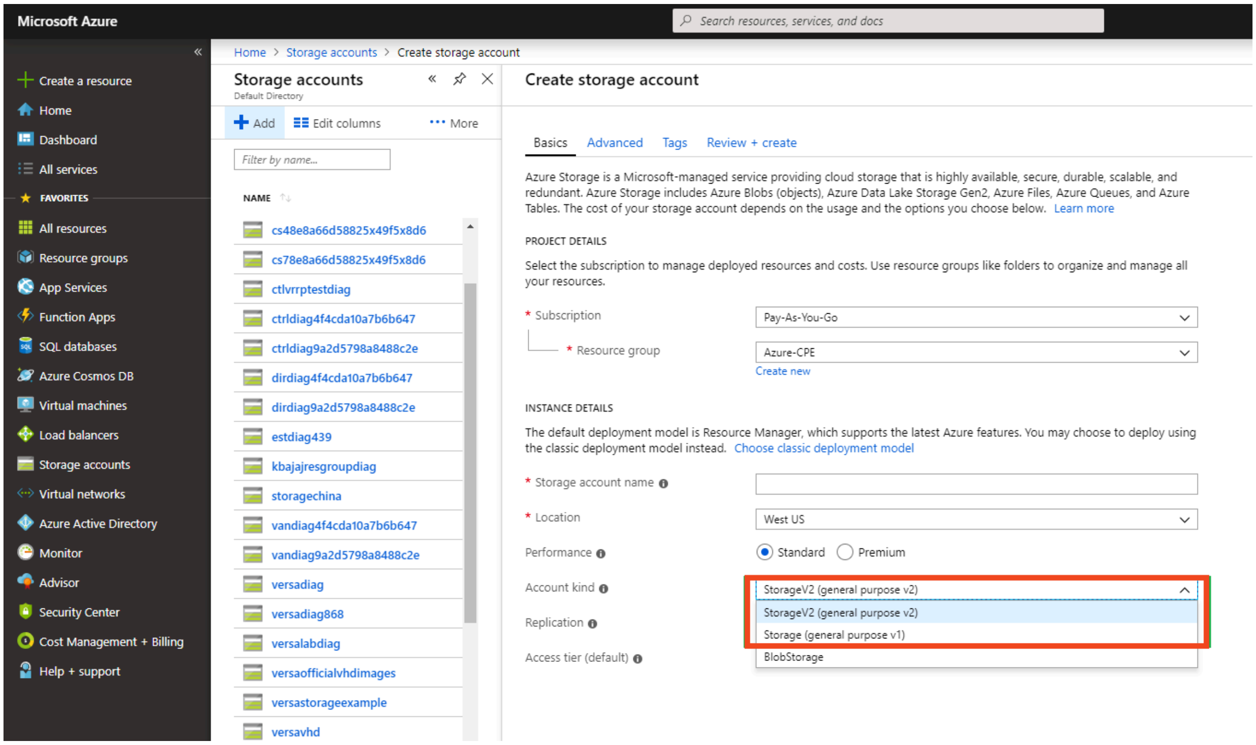
to continue to Microsoft Azure

Email, phone, or Skype

No account? Create one!

Can't access your account?

2. In the left menu bar, click Storage Accounts.
3. In the Storage Accounts menu bar, click the + Add icon. The Create Storage Account window displays.
   a. In the Subscription field, select the subscription in which you want to create the storage account.
   b. Select Resource Group or create new Resource Group where you want to create the storage account.
   c. In the Storage Account Name field, enter a name for the storage account name.
   d. In the Location field, select the location.
   e. In the Account Kind field, select either Storage V2 (General Purpose V2) or Storage (General Purpose V1). Do not select BlobStorage.
   f. Click Review + Create button to create a new storage account to store the Versa component images.



Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. Learn more

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription: Pay-As-You-Go

* Resource group: Azure-CPE

Create new

INSTANCE DETAILS

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. Choose classic deployment model

* Storage account name:

* Location: West US

Performance: ● Standard ○ Premium

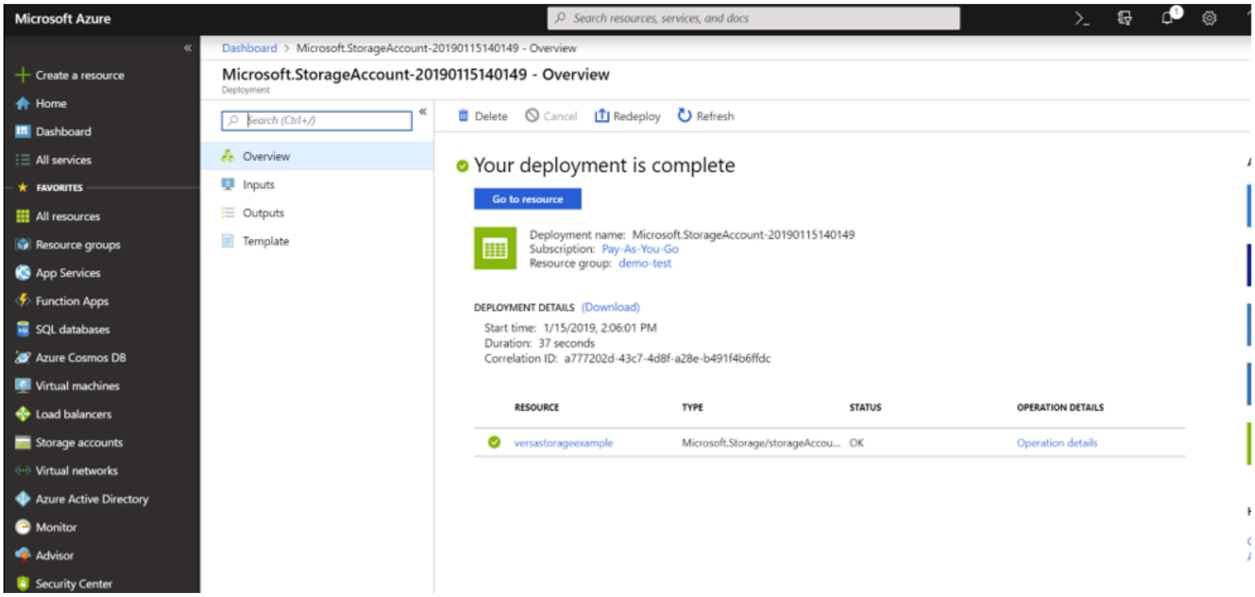
Account kind: StorageV2 (general purpose v2)

- StorageV2 (general purpose v2)
- Storage (general purpose v1)
- BlobStorage

Replication

Access tier (default)

A confirmation window displays.



Microsoft.StorageAccount-20190115140149 - Overview

Deployment

Your deployment is complete

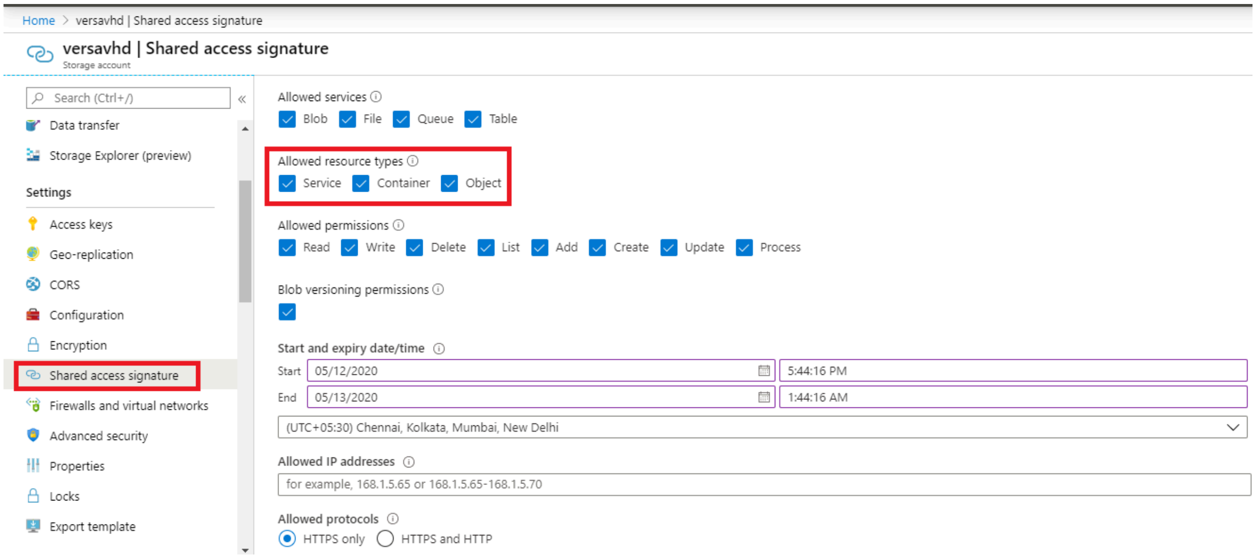
Deployment name: Microsoft.StorageAccount-20190115140149
Subscription: Pay-As-You-Go
Resource group: demo-test

DEPLOYMENT DETAILS (Download)

Start time: 1/15/2019, 2:06:01 PM
Duration: 37 seconds
Correlation ID: a777202d-43c7-4d8f-a28e-b491f4b6ffdc

RESOURCE	TYPE	STATUS	OPERATION DETAILS
✓ versastorageexample	Microsoft.Storage/storageAccou...	OK	Operation details

4. In the Storage Accounts menu bar, click the newly created storage account. The storage account's window displays.
5. Generate an SAS token:
    a. In the storage account's menu bar, click the Shared Access Signature option.



Home > versavhd | Shared access signature

versavhd | Shared access signature

Storage account

Allowed services
☑ Blob ☑ File ☑ Queue ☑ Table

Allowed resource types
☑ Service ☑ Container ☑ Object

Allowed permissions
☑ Read ☑ Write ☑ Delete ☑ List ☑ Add ☑ Create ☑ Update ☑ Process

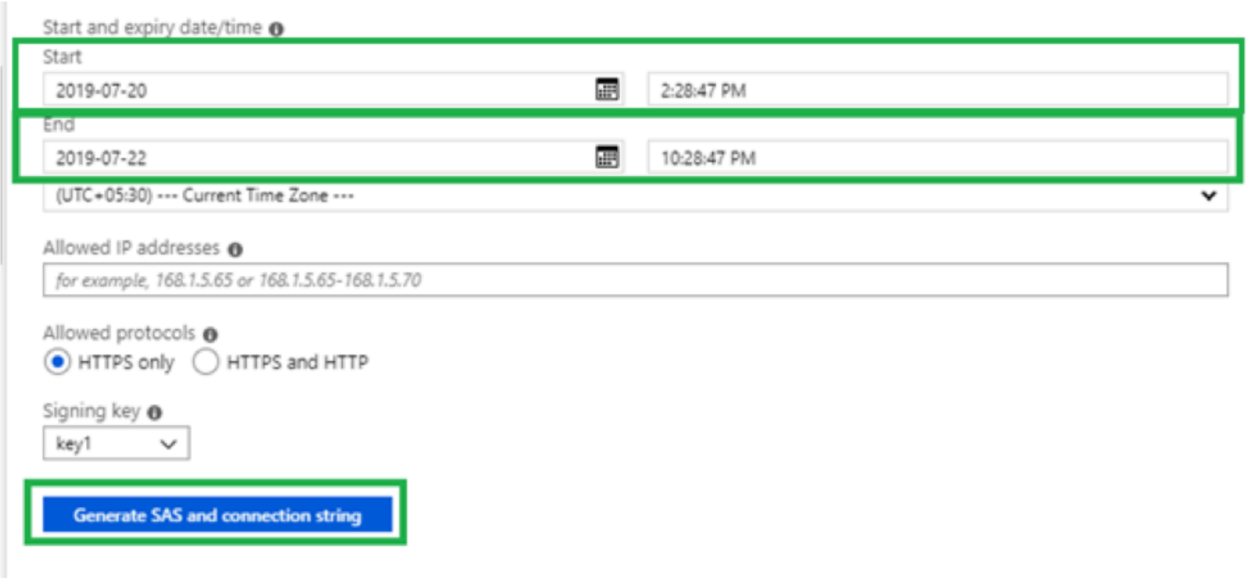
Blob versioning permissions
☑

Start and expiry date/time
Start 05/12/2020 5:44:16 PM
End 05/13/2020 1:44:16 AM
(UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi

Allowed IP addresses
for example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols
● HTTPS only ○ HTTPS and HTTP

b. In the Allowed Resource Types field, select all options.
c. In the Start Date field, select the date that is one day before the current date.
d. In the End Date field, select the date that is two days after the current date, to avoid having the upload operation expire if it is being performed across different time zones.



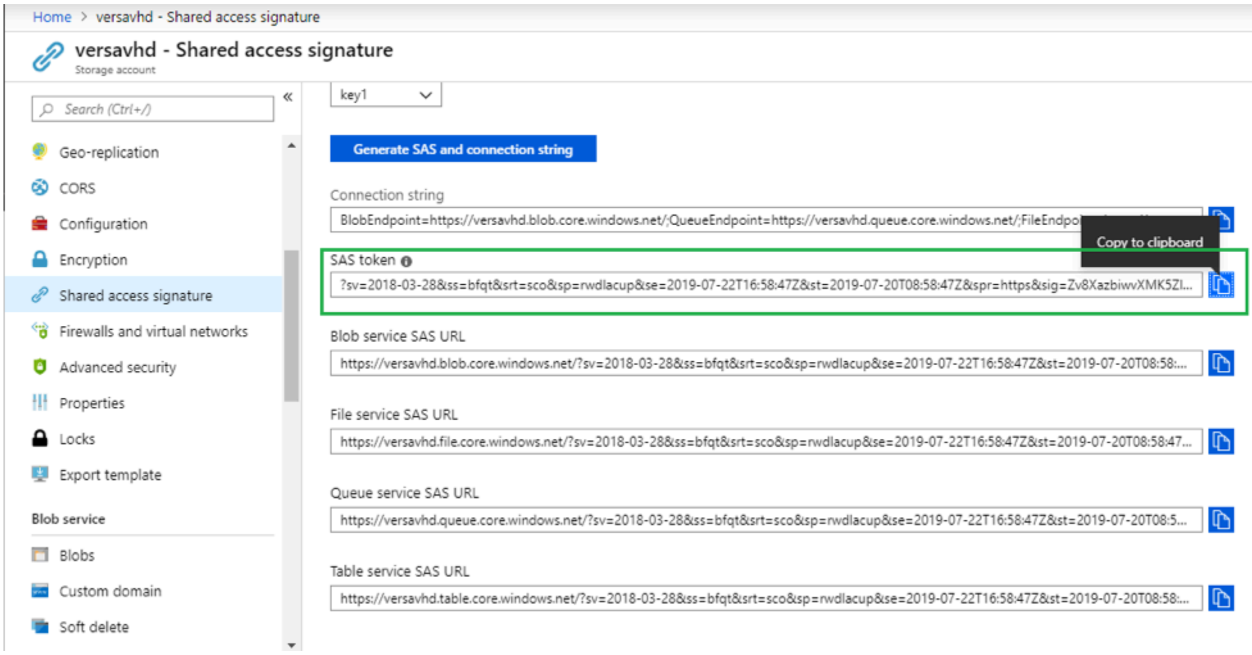
Start and expiry date/time
Start
2019-07-20 2:28:47 PM
End
2019-07-22 10:28:47 PM
(UTC+05:30) --- Current Time Zone ---

Allowed IP addresses
for example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols
● HTTPS only ○ HTTPS and HTTP

Signing key
key1

e. Keep all other options at the default values.
f. Click Generate SAS and Connection String to generate the SAS token.
g. Copy the generated SAS token (the second option on the screen, as highlighted in the screenshot below) to the clipboard to avoid any copy-and-paste errors.



key1

Connection string
BlobEndpoint=https://versavhd.blob.core.windows.net/;QueueEndpoint=https://versavhd.queue.core.windows.net/;FileEndpo

SAS token
?sv=2018-03-28&ss=bfqt&srt=sco&sp=rwdlacup&se=2019-07-22T16:58:47Z&st=2019-07-20T08:58:47Z&spr=https&sig=Zv8XazbiwvXMK5ZI...

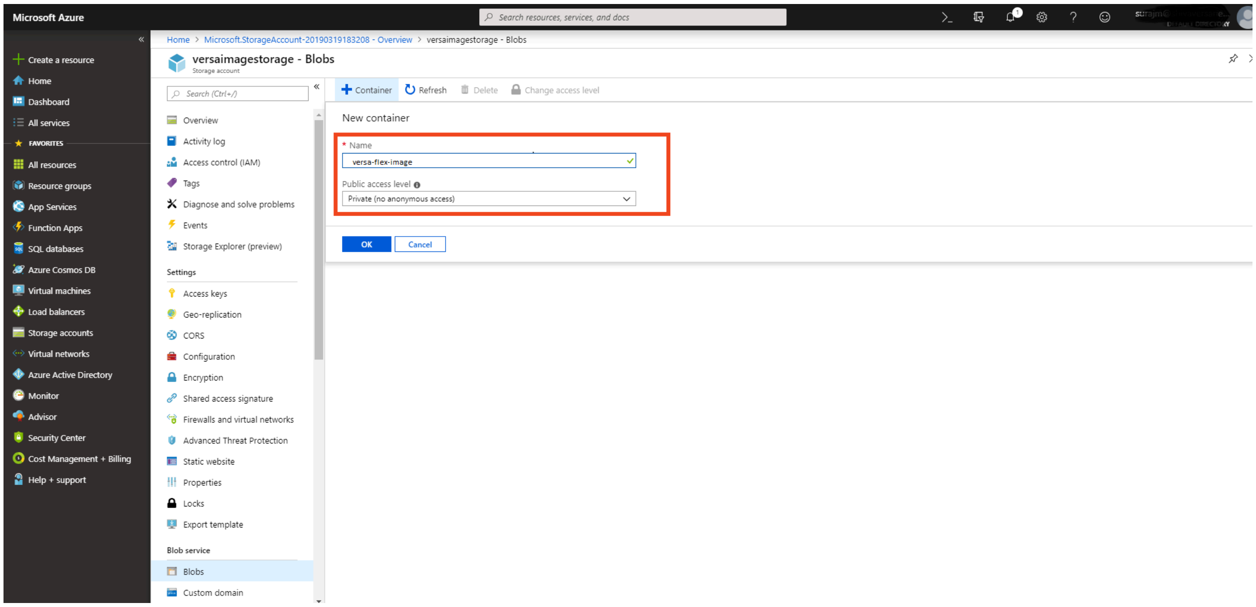
Blob service SAS URL
https://versavhd.blob.core.windows.net/?sv=2018-03-28&ss=bfqt&srt=sco&sp=rwdlacup&se=2019-07-22T16:58:47Z&st=2019-07-20T08:58:...

File service SAS URL
https://versavhd.file.core.windows.net/?sv=2018-03-28&ss=bfqt&srt=sco&sp=rwdlacup&se=2019-07-22T16:58:47Z&st=2019-07-20T08:58:47...

Queue service SAS URL
https://versavhd.queue.core.windows.net/?sv=2018-03-28&ss=bfqt&srt=sco&sp=rwdlacup&se=2019-07-22T16:58:47Z&st=2019-07-20T08:5...

Table service SAS URL
https://versavhd.table.core.windows.net/?sv=2018-03-28&ss=bfqt&srt=sco&sp=rwdlacup&se=2019-07-22T16:58:47Z&st=2019-07-20T08:58:...

6. Select the storage account that you created in Step 3. In the storage account's menu bar, select Containers. The Containers window displays.
   a. Click the + Container icon to add a container in which to store the Versa images.
   b. In the Name field, enter a name for the container.
   c. In the Public Access Level field, select Private access to the container.
   d. Click OK to create the container.



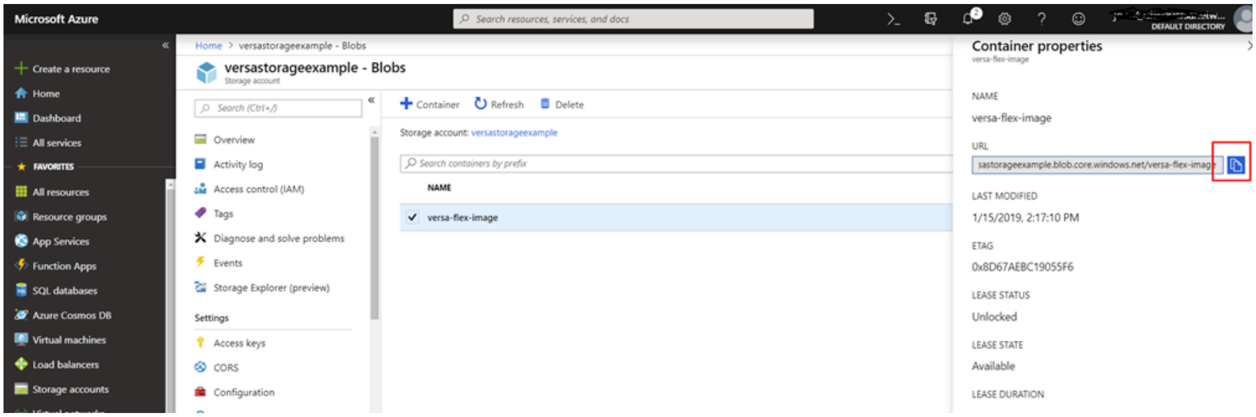
New container

* Name
versa-flex-image

Public access level
Private (no anonymous access)

7. The Container window displays. In the Container Properties menu, make a note of the container's URL, which is

shown in the URL field.



Storage account: versastorageexample

Container properties

versa-flex-image

NAME

versa-flex-image

URL

sastorageexample.blob.core.windows.net/versa-flex-image

LAST MODIFIED

1/15/2019, 2:17:10 PM

ETAG

0x8D67AEBC19055F6

LEASE STATUS

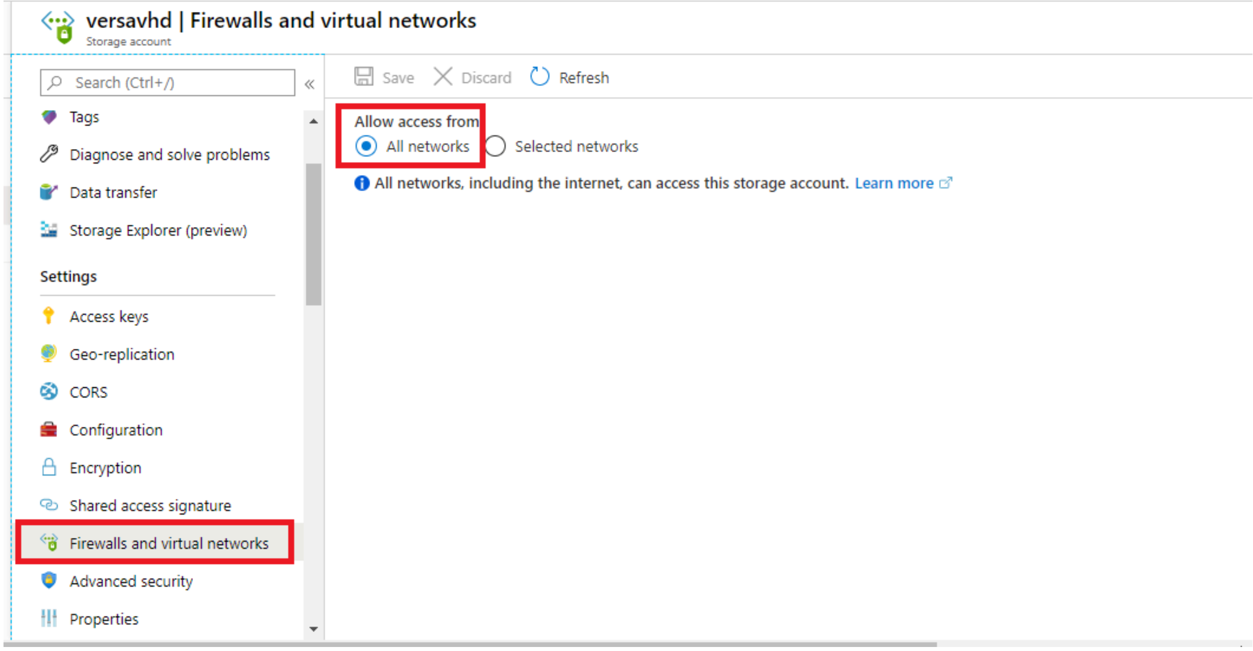
Unlocked

LEASE STATE

Available

LEASE DURATION

8. In the storage account's menu bar, select Firewalls and Virtual Networks. In the Allow Access From field, ensure that All Networks is selected.



Allow access from

● All networks ○ Selected networks

All networks, including the internet, can access this storage account. Learn more

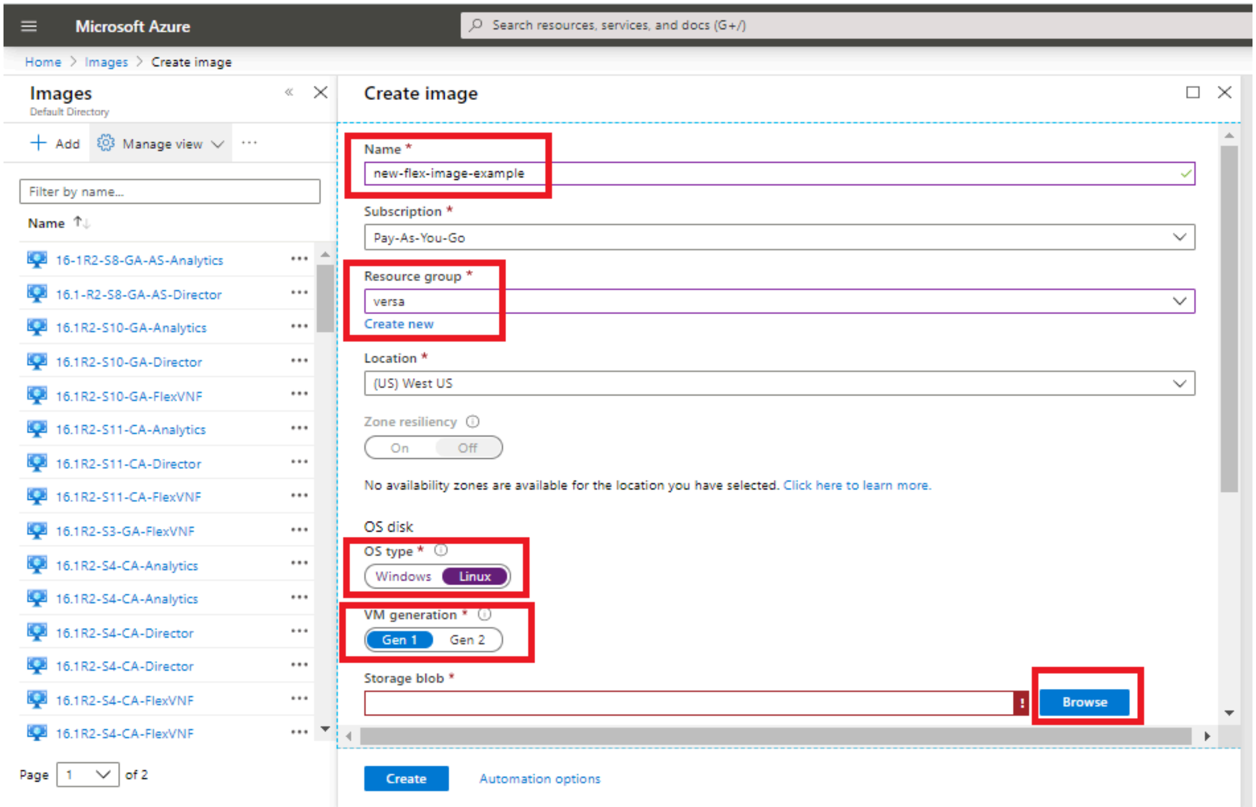
9. Provide Versa Customer Support with the SAS token generated in Step 5, the URL of the container obtained in Step 7, and the required software version to install Versa headend components. Versa using this information to transfer the Azure VHD image to the customer's storage account, which you use in the next step to create an image.
10. After Versa has transferred the VHD into the customer's account, create the image from that VHD.
    a. In the Azure portal's main menu, click the Images option.
    b. In the Images menu bar, click the + Add icon. The Create Image window displays.
    c. In the Name field, enter a name for the image.
    d. In the Subscription field, select the subscription to which Versa transferred the Azure VHD images.
    e. In the Resource Group field, select the resource group or create a new resource group.

f. In the Location field, select the region name where you want to create the image. Ensure that you use the same region where you created the storage account and where Versa transferred the images.

g. In the OS Disk field, click Linux.

h. In the VM Generation field, select Gen 1. Do not select Gen 2.

i. In the Storage Blob field, click Browse.



Images list:

- 16-1R2-S8-GA-AS-Analytics
- 16.1-R2-S8-GA-AS-Director
- 16.1R2-S10-GA-Analytics
- 16.1R2-S10-GA-Director
- 16.1R2-S10-GA-FlexVNF
- 16.1R2-S11-CA-Analytics
- 16.1R2-S11-CA-Director
- 16.1R2-S11-CA-FlexVNF
- 16.1R2-S3-GA-FlexVNF
- 16.1R2-S4-CA-Analytics
- 16.1R2-S4-CA-Analytics
- 16.1R2-S4-CA-Director
- 16.1R2-S4-CA-Director
- 16.1R2-S4-CA-FlexVNF
- 16.1R2-S4-CA-FlexVNF

Name *: new-flex-image-example

Subscription *: Pay-As-You-Go

Resource group *: versa
Create new

Location *: (US) West US

Zone resiliency: On [✓ Off]

No availability zones are available for the location you have selected. Click here to learn more.

OS disk

OS type *: Windows [✓ Linux]

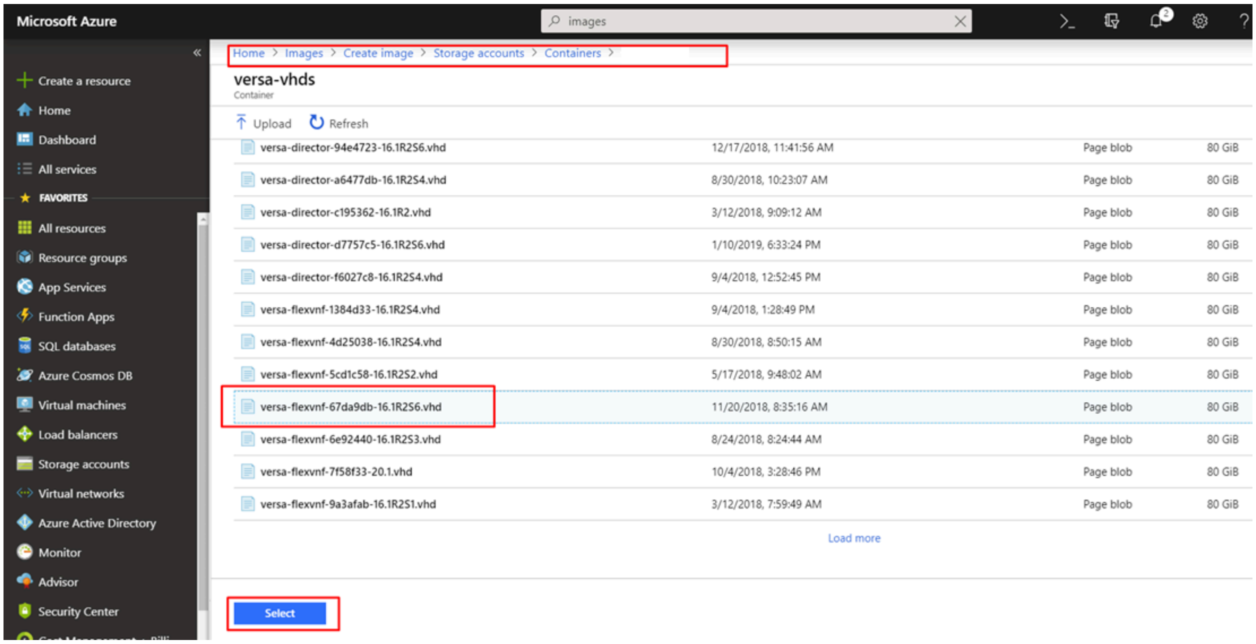
VM generation *: [✓ Gen 1] Gen 2

Storage blob *:

j. Navigate to Storage account > Containers.

   i. Select the container to which Versa transferred the VHD images.

   ii. Select the VHD image and click Select.



versa-vhds
Container

versa-director-94e4723-16.1R2S6.vhd	12/17/2018, 11:41:56 AM	Page blob	80 GiB
versa-director-a6477db-16.1R2S4.vhd	8/30/2018, 10:23:07 AM	Page blob	80 GiB
versa-director-c195362-16.1R2.vhd	3/12/2018, 9:09:12 AM	Page blob	80 GiB
versa-director-d7757c5-16.1R2S6.vhd	1/10/2019, 6:33:24 PM	Page blob	80 GiB
versa-director-f6027c8-16.1R2S4.vhd	9/4/2018, 12:52:45 PM	Page blob	80 GiB
versa-flexvnf-1384d33-16.1R2S4.vhd	9/4/2018, 1:28:49 PM	Page blob	80 GiB
versa-flexvnf-4d25038-16.1R2S4.vhd	8/30/2018, 8:50:15 AM	Page blob	80 GiB
versa-flexvnf-5cd1c58-16.1R2S2.vhd	5/17/2018, 9:48:02 AM	Page blob	80 GiB
versa-flexvnf-67da9db-16.1R2S6.vhd	11/20/2018, 8:35:16 AM	Page blob	80 GiB
versa-flexvnf-6e92440-16.1R2S3.vhd	8/24/2018, 8:24:44 AM	Page blob	80 GiB
versa-flexvnf-7f58f33-20.1.vhd	10/4/2018, 3:28:46 PM	Page blob	80 GiB
versa-flexvnf-9a3afab-16.1R2S1.vhd	3/12/2018, 7:59:49 AM	Page blob	80 GiB

k. Select the Account type as Standard HDD or SSD.

l. Click the Create button. The Images window lists the newly created image.

Create VMs for Versa Headend Components

Versa Networks provides Terraform template files that you can use to automate the deployment of the Versa headend components. Terraform is a single-pane cloud infrastructure deployment tool. It provides an automated deployment method that eases the manual efforts if, for example, you are using Azure Portal, which does not allow you to add more than one NIC when you initially install a VM.

You can also use other methods to deploy the headend components, such as Azure Portal, Azure CLI, and Azure ARM templates. For information about the benefits of using Terraform templates and comparing the deployment methods, see the Terraform documentation.

This section describes how to use Terraform templates to automatically create VMs for Versa headend components.

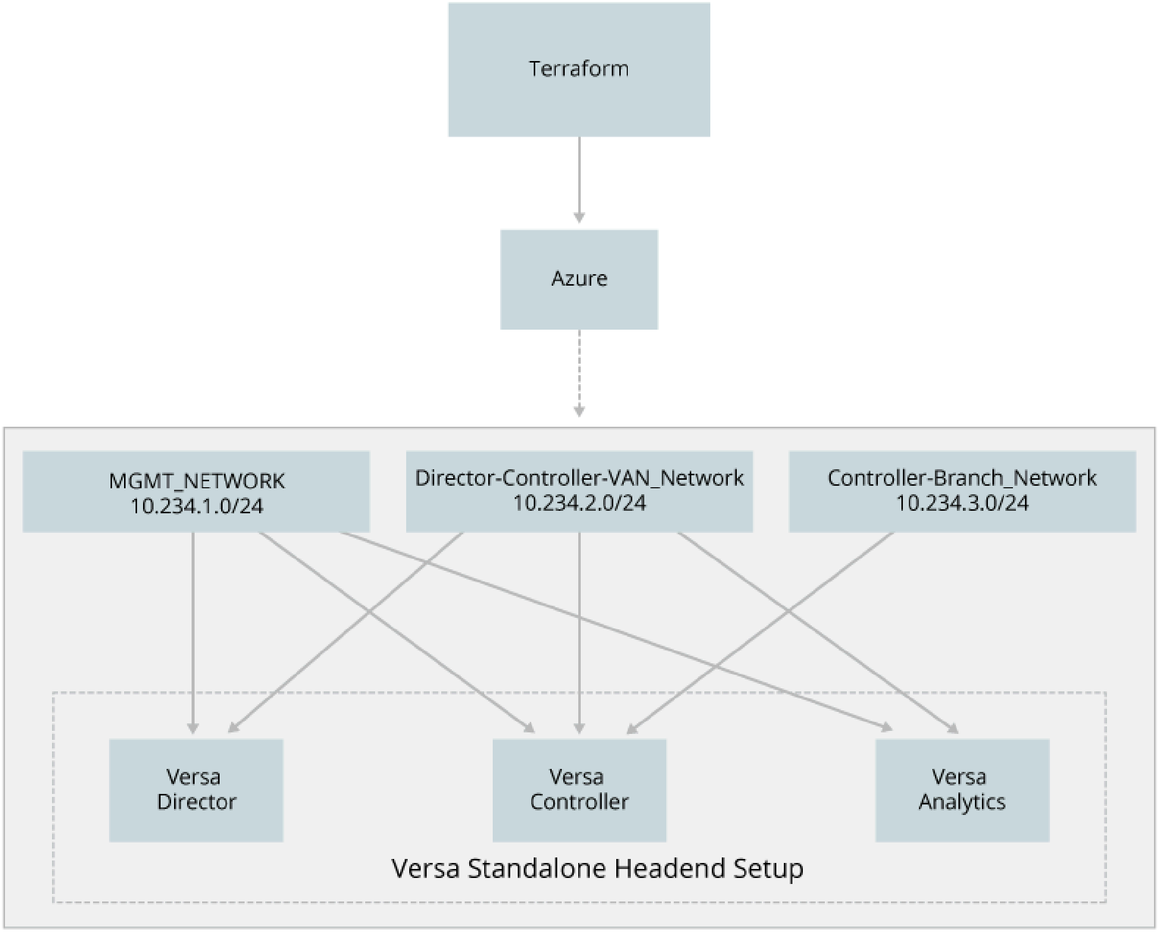
Before you begin:

- Obtain the Terraform template files from Versa Networks Customer Support. When requesting the template, specify whether you are using a standalone or a redundant headend topology. The figures below illustrate the topologies created by the Terraform template. You can also obtain the Terraform template files the public repository hosted by Versa Networks, at https://gitlab.com/versa-networks/terraform-azure-headend-deployment (for headend deployments) and https://gitlab.com/versa-networks/terraform-azure-flexvnf-deployment (for branch deployments).
- Install Terraform on your system. For information, see the Download Terraform article on the Terraform website.
- Set up Terraform access to Azure to enable Terraform to provision resources into Azure. For information, see the Microsoft Azure articles on the Microsoft website.
- Obtain subscription ID, client ID, client secret, and tenant ID for logging in to Terraform.

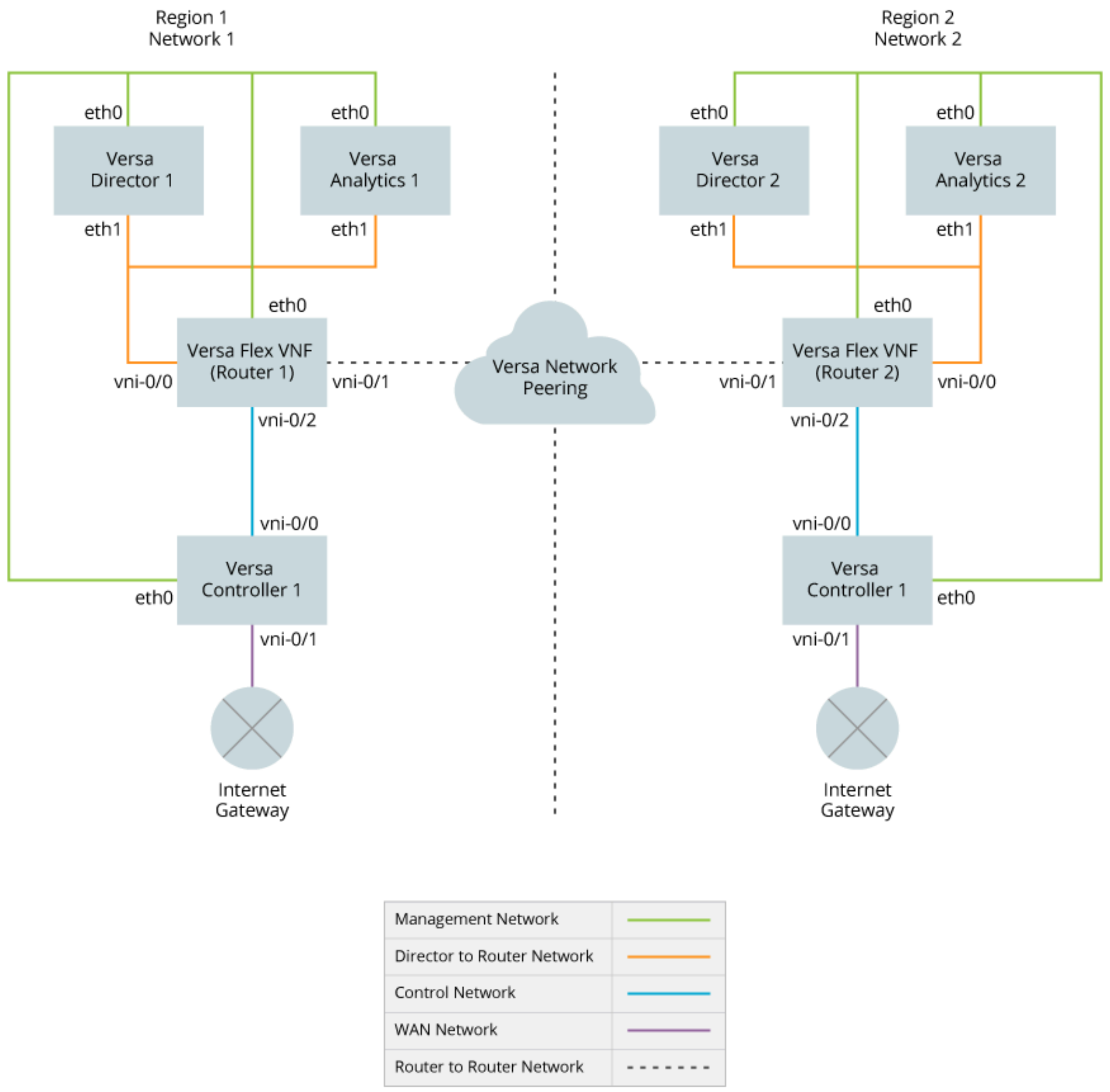
- Obtain Terraform Contributor-level user permission so that you can run the template.
- Obtain images for the Versa headend components—Versa Director, Versa Analytics, and Versa Controller—from Versa Networks Customer Support in .vhd format and create images using those .VHD files.

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

- For MGMT_NETWORK, the management IP is assigned using this network, and the public IP addresses assigned are associated with the three ports to the three Versa headend components.
- For Director-Controller-VAN_Network, the Director_SB, Controller_NB, and Analytics_NB IP addresses are assigned using this network. (SB is southbound, and NB is northbound.)
- For Controller-Branch_Network, the Controller_SB IP address is assigned using this work. This IP address is used to connect to the branch. The public IP address is also assigned on the Controller WAN port.



The following figure illustrates the redundant (high availability) headend topology created by the Terraform template.



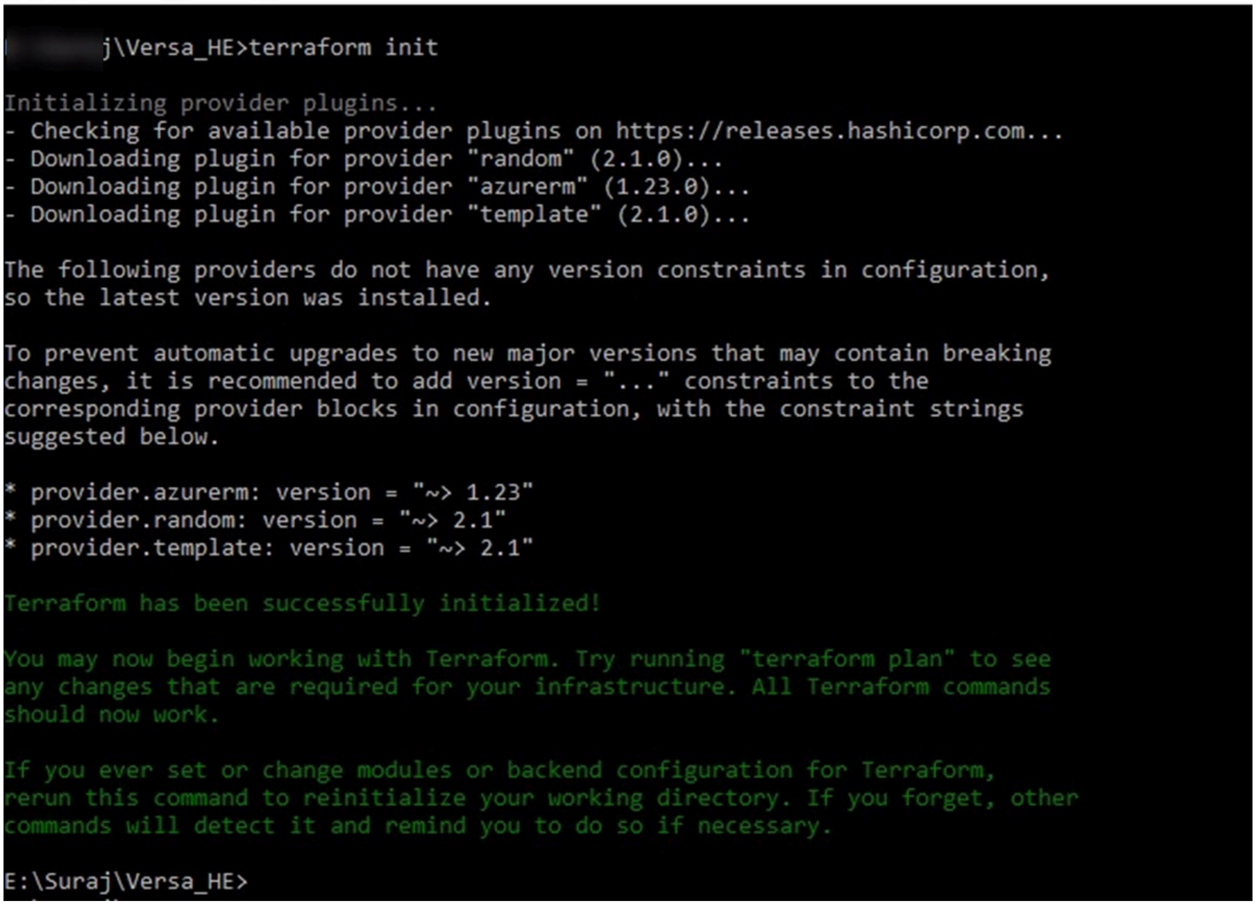
Versa High Availability Headend Setup

To create VMs using Terraform templates:

1. Save the folder that contains the Terraform template files to the system on which Terraform is installed. The template folder contains the files needed to deploy the Versa headend VM resources on Azure. The table at the end of this section describes each of the files.
2. 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.
3. 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.

4. Initialize Terraform. The initialization process downloads the Terraform plugins that are required to run the template.

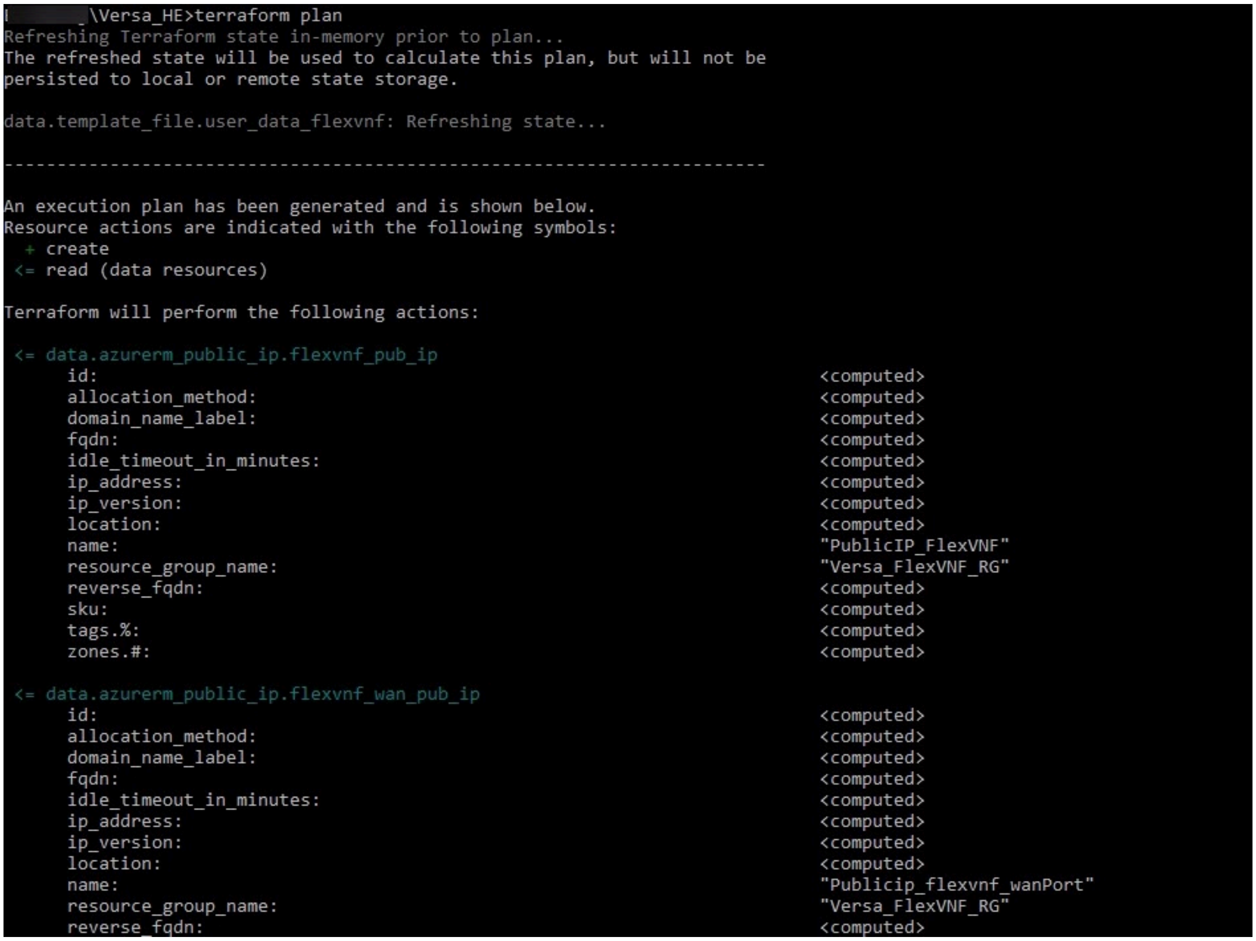
> ~$ **terraform init**



```
[redacted]j\Versa_HE>terraform init

Initializing provider plugins...
- Checking for available provider plugins on https://releases.hashicorp.com...
- Downloading plugin for provider "random" (2.1.0)...
- Downloading plugin for provider "azurerm" (1.23.0)...
- Downloading plugin for provider "template" (2.1.0)...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.azurerm: version = "~> 1.23"
* provider.random: version = "~> 2.1"
* provider.template: version = "~> 2.1"

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

E:\Suraj\Versa_HE>
```

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

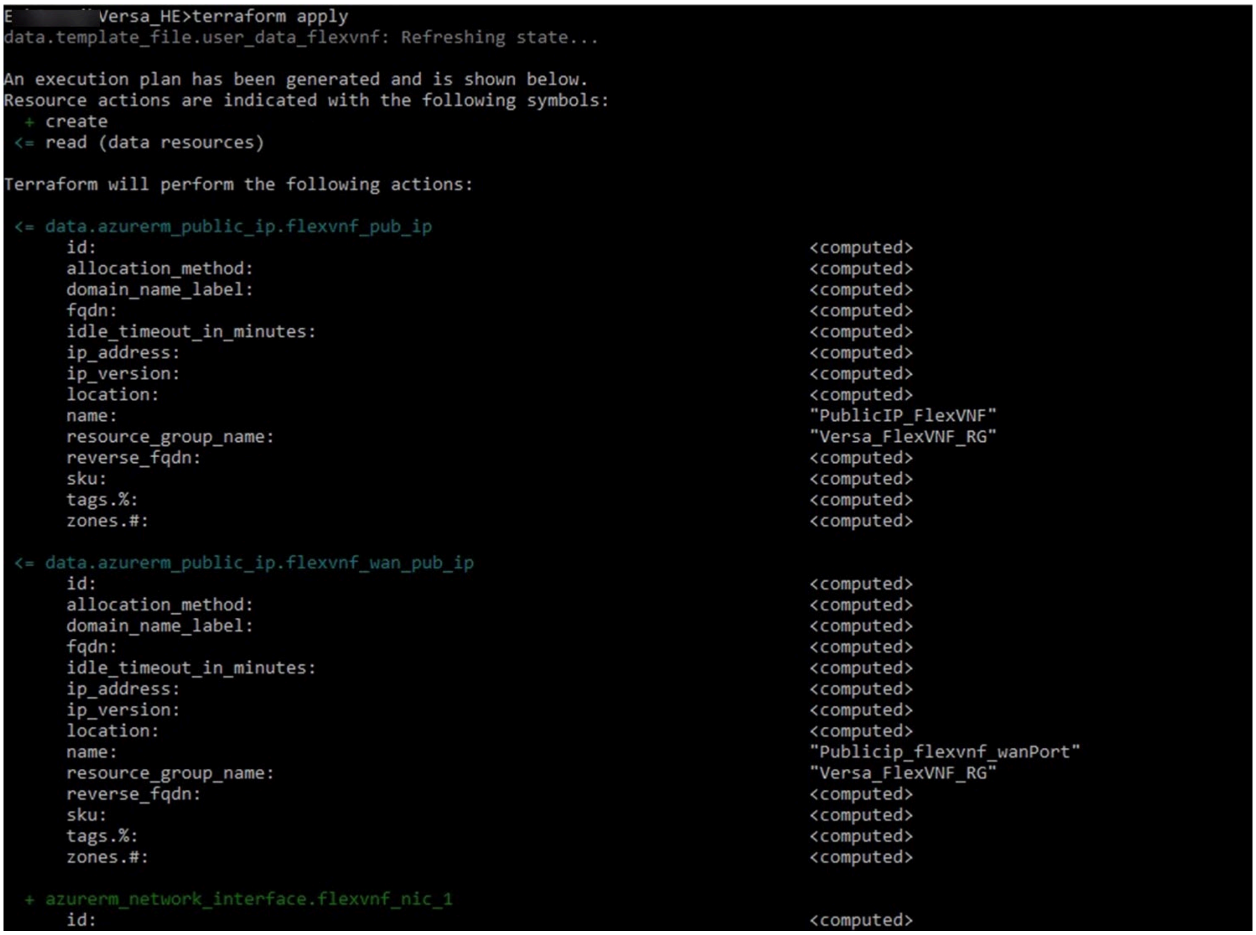
> ~$ **terraform plan**



```
[redacted]\Versa_HE>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_flexvnf: 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.azurerm_public_ip.flexvnf_pub_ip
      id:                                                                    <computed>
      allocation_method:                                                     <computed>
      domain_name_label:                                                     <computed>
      fqdn:                                                                  <computed>
      idle_timeout_in_minutes:                                               <computed>
      ip_address:                                                            <computed>
      ip_version:                                                            <computed>
      location:                                                              <computed>
      name:                                                                  "PublicIP_FlexVNF"
      resource_group_name:                                                   "Versa_FlexVNF_RG"
      reverse_fqdn:                                                          <computed>
      sku:                                                                   <computed>
      tags.%:                                                                <computed>
      zones.#:                                                               <computed>

 <= data.azurerm_public_ip.flexvnf_wan_pub_ip
      id:                                                                    <computed>
      allocation_method:                                                     <computed>
      domain_name_label:                                                     <computed>
      fqdn:                                                                  <computed>
      idle_timeout_in_minutes:                                               <computed>
      ip_address:                                                            <computed>
      ip_version:                                                            <computed>
      location:                                                              <computed>
      name:                                                                  "Publicip_flexvnf_wanPort"
      resource_group_name:                                                   "Versa_FlexVNF_RG"
      reverse_fqdn:                                                          <computed>
```

7. Run the template to deploy all the VM resources on Azure:

   > ~$ **terraform apply**



```
E[redacted]\Versa_HE>terraform apply
data.template_file.user_data_flexvnf: 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.azurerm_public_ip.flexvnf_pub_ip
      id:                                                                    <computed>
      allocation_method:                                                     <computed>
      domain_name_label:                                                     <computed>
      fqdn:                                                                  <computed>
      idle_timeout_in_minutes:                                               <computed>
      ip_address:                                                            <computed>
      ip_version:                                                            <computed>
      location:                                                              <computed>
      name:                                                                  "PublicIP_FlexVNF"
      resource_group_name:                                                   "Versa_FlexVNF_RG"
      reverse_fqdn:                                                          <computed>
      sku:                                                                   <computed>
      tags.%:                                                                <computed>
      zones.#:                                                               <computed>

 <= data.azurerm_public_ip.flexvnf_wan_pub_ip
      id:                                                                    <computed>
      allocation_method:                                                     <computed>
      domain_name_label:                                                     <computed>
      fqdn:                                                                  <computed>
      idle_timeout_in_minutes:                                               <computed>
      ip_address:                                                            <computed>
      ip_version:                                                            <computed>
      location:                                                              <computed>
      name:                                                                  "Publicip_flexvnf_wanPort"
      resource_group_name:                                                   "Versa_FlexVNF_RG"
      reverse_fqdn:                                                          <computed>
      sku:                                                                   <computed>
      tags.%:                                                                <computed>
      zones.#:                                                               <computed>

  + azurerm_network_interface.flexvnf_nic_1
      id:                                                                    <computed>
```

The following table describes the contents of each Terrform template file and the actions performed by each file.

Filename	**Description or Action**
main.tf	• Provision one resource group, called Versa_HE. To change the resource group name, edit the terraform.tfvars file. • Provision the virtual network 10.234.0.0/16. To change the IP prefix, edit the terraform.tfvars file. • Provision three /24 subnetworks. By default, 10.234.1.0/24, 10.234.2.0/24 & 10.234.3.0/24 subnets are used. To change the subnets, edit the terraform.tfvars file. ◦ 10.234.1.0/24 subnet is for management of all the headend instances. ◦ 10.234.2.0/24 subnet is the control network. This subnetwork is used for Director

Filename	Description or Action
	downstream (southbound) ports, Controller upstream (northbound) ports, and Analytics downstream (southbound ports. ◦ 10.234.3.0/24 subnet is the WAN network. This subnetwork is used for Controller downstream (southbound) ports and branch connectivity. • 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 IPSec overlay network address. This route is 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. ◦ 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 tun the runcloud-init script to: ◦ Update the /etc/network/interface file. ◦ Add the SSH key for the Admin user • Install a Analytics instance and run the cloud-init script to: ◦ Update the /etc/network/interface file. ◦ Update the /etc/hosts file. ◦ Add the ssh key for the Versa user ◦ Copy the certificate from the Director and install it on the Analytics node.
var.tf	Provide definitions of all variables defined and used in the template. Do not make any changes to this file.

Filename	Description or Action
terraform.tfvars	User-defined input variables that are used to populate the Terraform templates. Edit this file to set or change the following variables: • analytics_vm_size—Instance type and size used to provision the Analytics instance. The default is Standard_DS3. • client_id—Client identifier information obtained as part of Setting up Terraform access. • client_secret—Client secret information obtained as part of setting up Terraform access. • controller_vm_size—Instance type and size used to provision the Controller instance. The default is Standard_DS3. • director_vm_size—Instance type and size used to provision the Director instance. The default is Standard_DS3. • hostname_director—Hostname of the Director instance. The default is versa-director. • image_analytics—Filename of the Analytics image. • image_controller—Filename of the Controller image. • image_director—Filename of the Director image. • location—Enter the string that identifies the headend location. For example, west-us or west-europe. • resource_group—Name of the resource group in which to place the resources for the headend. The default is Versa_HE. • ssh_key–SSH public key—This key is required to log in to the headend instances. To generate the SSH key, use the sshkey-gen or putty key generator command. You cannot generate keys within Azure. • tenant_id—Tenant identifier information obtained as part of Setting up Terraform access.
output.tf	Output parameters, including management IP address, public IP address, CLI commands, and UI login information, for all instances. Do not make any changes to this file.
director.sh	Bash script that runs as part of cloud-init script on the Versa Director instance. Do not make any changes to this file.

Filename	Description or Action
controller.sh	Bash script that runs as part of cloud-init script on Versa Controller instance. Do not make any changes to this file.
van.sh	Bash script that runs as part of cloud-init script on Versa Analytics instance. Do not make any changes to this file.

Deploy a VOS Device as a Branch Using Terraform Templates

This section describes how to use Terraform templates to automatically deploy a Versa Operating System^{TM} (VOS^{TM}) device as a branch. You obtain the Terraform template files from Versa Networks Customer Support. To deploy a VOS device as a branch, ask for the Versa_FlexVNF_SingleBranch_Staging.zip file, which contains the Terraform templates for the branch deployment.

To deploy a VOS device as a branch using Terraform templates, follow Step 1 to Step 6 as described in Create VMs for Versa Headend Components.

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

Filename	Description or Action
vars.tf	Definitions of all variables defined and used in the template. Do not make any changes to this file.
terraform.tfvars	User-defined input variables that are used to populate the Terraform templates. Edit this file to set or change the following variables: • subscription_id—Subscription identifier information for the Azure account. • client_id—Client identifier information obtained as part of setting up Terraform access. • client_secret—Client secret information obtained as part of setting up Terraform access. • tenant_id—Tenant identifier information obtained as part of Setting up Terraform access. • location—Enter the string that identifies the VOS device location. For example, west-us or west-europe. • resource_group—Name of the resource group in which to place the resources for the VOS device. The default is Versa_FlexVNF_RG. • ssh_key—SSH public key. This key is required to log in to the VOS instances. To generate the SSH key, use the sshkey-gen or putty key generator command. You cannot generate keys within Azure. • mgmt_subnet—Management subnet ID used to create interfaces on the management subnet. • wan_subnet—Subnet of the WAN network used to to create the WAN network interface. • lan_subnet—Subnet of the LAN network used to create the LAN network interfaces. • vm_name—Name of the VM that is displayed in the VM list of the Azure Portal. The default is Versa_FlexVNF. • flexvnf_vm_size—Instance type and size used to provision the FlexVNF-1 VM. The default is Standard_F4s. • controller_wan_ip—WAN IP of the Controller device used for staging with the branch. This can be a private IP address or a public IP address where the branch can reach the Controller device. • local_auth—Local authentication at the branch used

	for staging. • remote_auth—Remote authentication to use for staging. This is Controller-side authentication key used during staging of the branch. • branch_serial_num—Serial number of branch to be set. This must be the same as the serial number that is provided during the corresponding Workflow device deployment in Versa Director. • branch_wan_nic—Indicates the port number of WAN port. To use vni-0/0 as the branch WAN interface or port, the value must be 0; to use vni-0/1, the value must be 1; and so on. The default is 0. • director_northboud_ip—Northbound IP address of the Director node. • director_southboud_ip—Southbound IP address of the Director node.
output.tf	Output parameters, including management IP address, public IP address, WAN IP address, LAN IP address, and CLI commands, for all instances. Do not make any changes to this file.
flexvnf.sh	Bash script that runs as part of cloud-init script on the VOS instance. Do not make any changes to this file.
main.tf	• Provision one resource group, called Versa_FlexVNF_RG. To change the resource group name, edit the terraform.tfvars file. • Assign a public IP address on the management port of the VOS device instance. • Assign a static public IP address for the VOS device WAN port. • Provision a network security group, and add all the firewall rules required to set up the VOS device. • Install a VOS instance and run the cloud-init script to: ◦ Update the /etc/network/interface file. ◦ Add the SSH key for the admin user. ◦ Add a cron job to trigger the staging.py script required for ZTP.

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](#)
[Install on Azure](#)
[Qualified AWS and Azure Instances](#)