TOOLS

Azure Outbound Terraform Template Deployment Guide

Table of Contents

Overview

This Terraform template deploys two Palo Alto Networks VM-Series Firewalls with a Standard Azure internal load-balancer, and an Ubuntu test VM, as shown in figure 1 below. It allows the user to demo Azure egress security using Palo Alto Networks VM-Series firewalls.

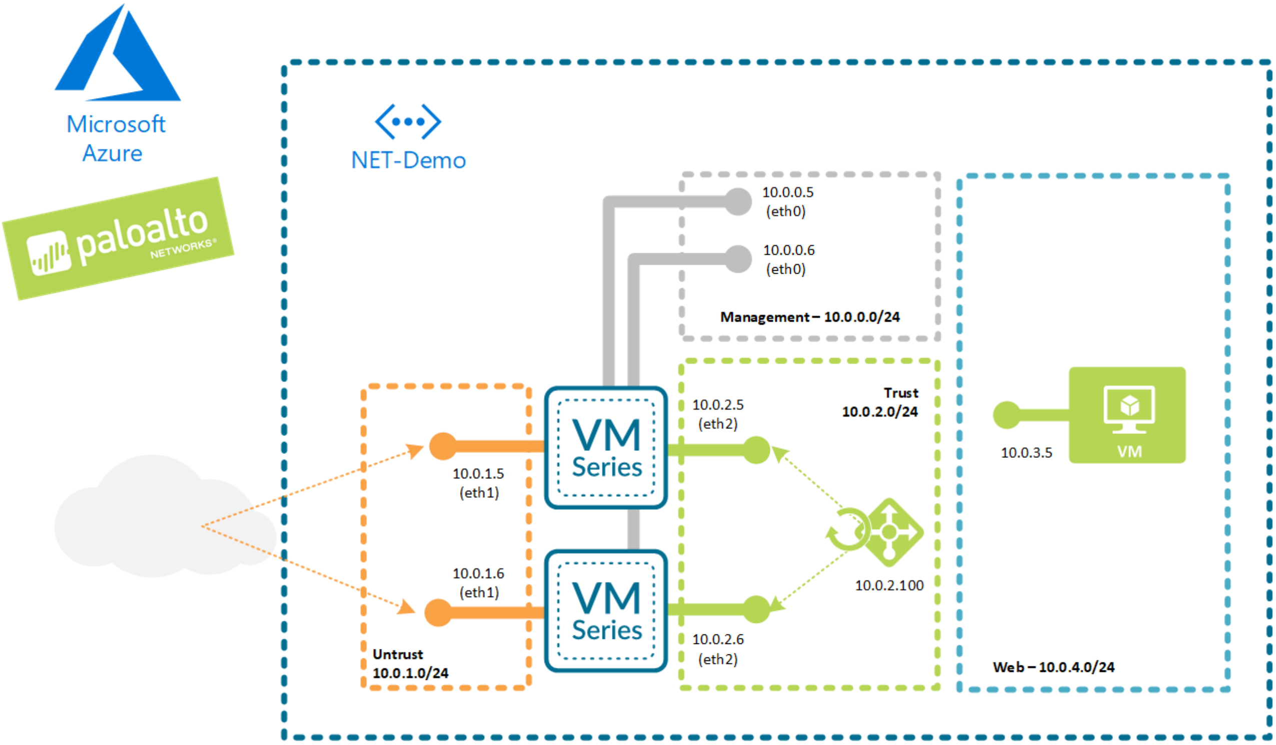


Figure 1

Implementation Details

The following components are deployed by this Terraform template:

- One Azure Internal Load Balancer (Standard SKU)
- Two Palo Alto Networks VM-Series Firewalls in an Availability Set
- One Ubuntu Server to test outbound traffic
- One VNET, multiple Subnets, UDR and NSGs to support traffic flow

The following virtual machines sizes are used by default. These can be changed if needed.

Instance	VM Size
VM-Series Firewalls	Standard_D3_v2
Ubuntu Server	Standard_DS1_v2

Access to the firewalls are through public IP addresses attached to the management interfaces of the firewalls. Access to the Ubuntu Test server is via Serial Console from the Azure Portal.

Access to the Internet from the Ubuntu Server via the VM-Series firewalls is through public IP addresses attached to the Untrust interface of the firewalls.

If you are using bootstrap, the login credentials are:

User: paloalto

Password: PaloAlt0!123!!

Prerequisites

Azure Account

You can use the corporate Azure subscription to deploy this template. Alternatively, you can use your personal Azure subscription as well. Do take note of the charges involved.

Terraform

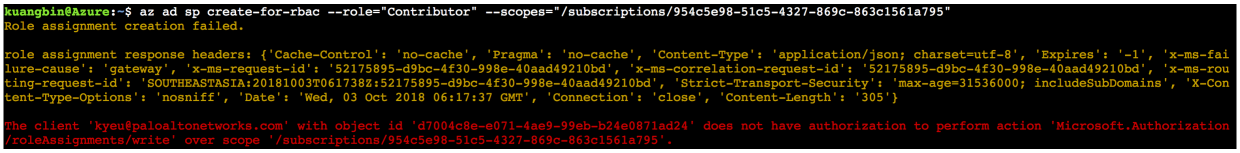
You can download and install Terraform for your platform from https://www.terraform.io/downloads.html.

Terraform supports authenticating to Azure through a Service Principle or the Azure CLI.

https://www.terraform.io/docs/providers/azurerm/authenticating_via_service_principal.html

https://www.terraform.io/docs/providers/azurerm/authenticating_via_azure_cli.html

Do note that if you are using a Corporate subscription, you might not be able to create a Service Principle with a Contributor role that has read/write access to the subscription, as shown in the diagram below. In that case, use the Azure CLI for authentication.



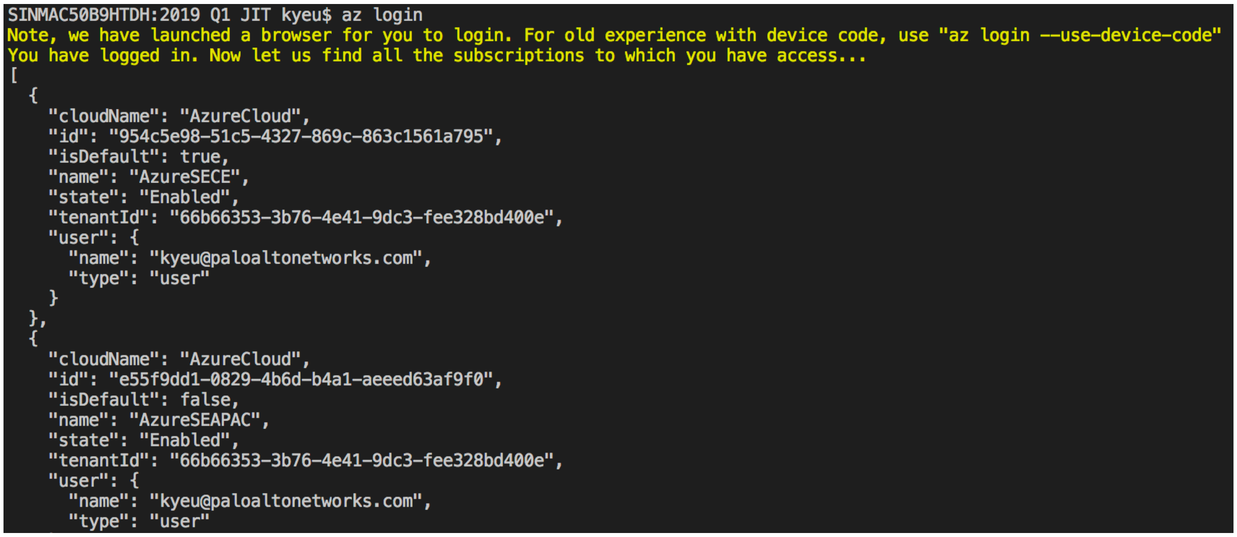
```
kuangbin@Azure:~$ az ad sp create-for-rbac --role="Contributor" --scopes="/subscriptions/954c5e98-51c5-4327-869c-863c1561a795"
Role assignment creation failed.

role assignment response headers: {'Cache-Control': 'no-cache', 'Pragma': 'no-cache', 'Content-Type': 'application/json; charset=utf-8', 'Expires': '-1', 'x-ms-fai
lure-cause': 'gateway', 'x-ms-request-id': '52175895-d9bc-4f30-998e-40aad49210bd', 'x-ms-correlation-request-id': '52175895-d9bc-4f30-998e-40aad49210bd', 'x-ms-rou
ting-request-id': 'SOUTHEASTASIA:20181003T061738Z:52175895-d9bc-4f30-998e-40aad49210bd', 'Strict-Transport-Security': 'max-age=31536000; includeSubDomains', 'X-Con
tent-Type-Options': 'nosniff', 'Date': 'Wed, 03 Oct 2018 06:17:37 GMT', 'Connection': 'close', 'Content-Length': '305'}

The client 'kyeu@paloaltonetworks.com' with object id 'd7004c8e-e071-4ae9-99eb-b24e0871ad24' does not have authorization to perform action 'Microsoft.Authorization
/roleAssignments/write' over scope '/subscriptions/954c5e98-51c5-4327-869c-863c1561a795'.
```

Azure CLI

This is optional. This is required if you are using Azure CLI for authentication.

You can download and install Azure CLI for your platform from https://docs.microsoft.com/en-us/cli/azure/install-azure-cli.

After successful installation, you can login to Azure by issuing the command “az login”. This will redirect you to a web browser where you can login to Azure. After logging in, Azure CLI will show your account information, as show in the diagram below.



```
SINMAC50B9HTDH:2019 Q1 JIT kyeu$ az login
Note, we have launched a browser for you to login. For old experience with device code, use "az login --use-device-code"
You have logged in. Now let us find all the subscriptions to which you have access...
[
  {
    "cloudName": "AzureCloud",
    "id": "954c5e98-51c5-4327-869c-863c1561a795",
    "isDefault": true,
    "name": "AzureSECE",
    "state": "Enabled",
    "tenantId": "66b66353-3b76-4e41-9dc3-fee328bd400e",
    "user": {
      "name": "kyeu@paloaltonetworks.com",
      "type": "user"
    }
  },
  {
    "cloudName": "AzureCloud",
    "id": "e55f9dd1-0829-4b6d-b4a1-aeeed63af9f0",
    "isDefault": false,
    "name": "AzureSEAPAC",
    "state": "Enabled",
    "tenantId": "66b66353-3b76-4e41-9dc3-fee328bd400e",
    "user": {
      "name": "kyeu@paloaltonetworks.com",
      "type": "user"
```

Download Template and Bootstrap Files

Download the template files and bootstrap files to a local directory.

Azure File Share for Bootstrap

You can create a file share in Azure according to the instructions listed at https://www.paloaltonetworks.com/documentation/81/virtualization/virtualization/bootstrap-the-vm-series-firewall/bootstrap-the-vm-series-firewall-in-azure

From the Azure portal, upload the “init-cfg.txt” and “bootstrap.xml” files to the “config” directory.

Do note the following information which is required at a later stage (see Step 2 in the url mentioned earlier).

- Storage Account Name
- Storage Access Key
- File-share Name

Getting Started

Providing Information

You need to modify the file “terraform.tfvars” to provide the required information to run the template successfully.

The options in the “terraform.tfvars” file are:

- resource_group_name

  Name of the Resource Group. It can be a new or an existing Resource Group. However, do note that if you deploy the template using an existing Resource Group, existing resources might be deleted if they are not defined in the template.

- location

  This is to specify the Azure region to launch the temple. The options are:

  "eastasia", "southeastasia", "centralus", "eastus", "eastus2", "westus", "northcentralus", "southcentralus", "northeurope", "westeurope", "japanwest", "japaneast", "brazilsouth", "australiaeast", "australiasoutheast", "southindia", "centralindia", "westindia", "canadacentral", "canadaeast", "uksouth", "ukwest", "westcentralus", "westus2", "koreacentral", "koreasouth", "francecentral", "francesouth", "australiacentral", "australiacentral2"

- virtualNetworkName

  Name of the Virtual Network

- bootstrap

  This is to specify whether you want to bootstrap the firewalls. The options are:“yes” or “no”

- vmSize

  The VM Size for the VM-Series Firewalls. Typically used options are: “Standard_D3_v2”, Standard_D4_v2”, “Standard_D5_v2”, “Standard_DS3_v2”, Standard_DS4_v2”, “Standard_DS5_v2”

- imageSku

  SKU for the VM-Series Firewalls. The options are: “byol”, “bundle1”, “bundle2”

- imageVersion

  The version of VM-Series Firewalls you want to use. The options are: “latest”, “8.1.0”, “8.0.0”

- customdata

  This is the information required for bootstrapping. It is in the format:

  "storage-account=<storage account name>,access-key=<storage access key>,file-share=<file-share name>,share-directory=None"

  The <storage account name>, <storage access key>, and <file-share name> are obtained earlier in the prerequisites.

- adminUsername

  This is the admin username for the Ubuntu test server. If bootstrap is not used, this will also be the admin username for the VM-Series firewalls.

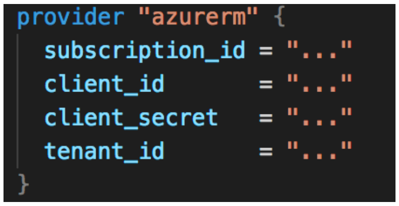
- adminPassword

  This is the admin password for the Ubuntu test server. If bootstrap is not used, this will also be the admin password for the VM-Series firewalls.

- AllowedSourceIPRange

  This is the source IP or IP subnet allowed to access the web management interface of the VM-Series firewalls.

If you are using Service Principle for authentication to Azure, you need to modify the “main.tf” file to provide the following information.

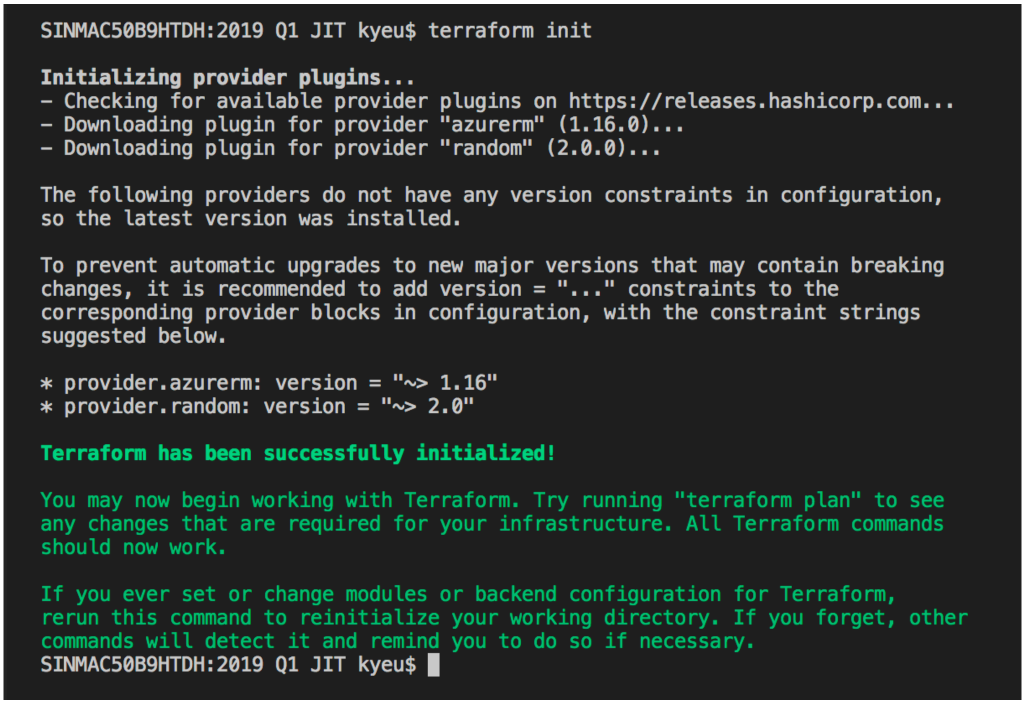


```
provider "azurerm" {
  subscription_id = "..."
  client_id       = "..."
  client_secret   = "..."
  tenant_id       = "..."
}
```

We can start to deploy the template after providing the required information.

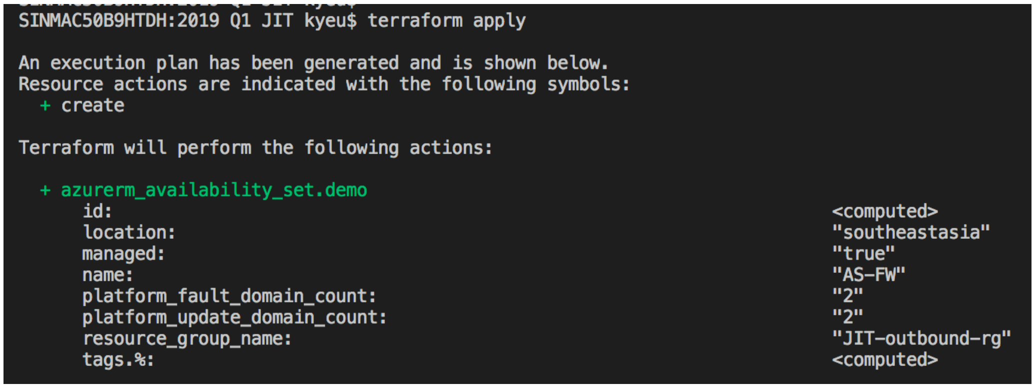
Launch the Template

From the terminal window, go to the directory where you saved the template file, then issue the command “terraform init”. This is to initialize terraform and download any providers that you may have used in your template.



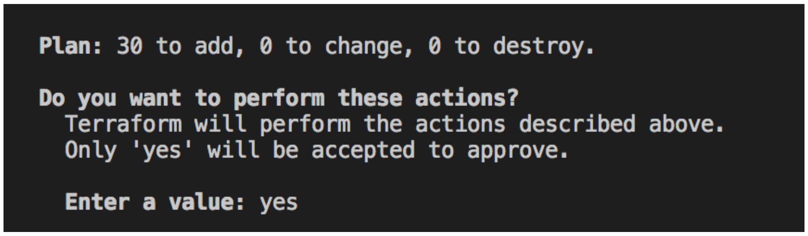
```
SINMAC50B9HTDH:2019 Q1 JIT kyeu$ terraform init

Initializing provider plugins...
- Checking for available provider plugins on https://releases.hashicorp.com...
- Downloading plugin for provider "azurerm" (1.16.0)...
- Downloading plugin for provider "random" (2.0.0)...

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 = "~> 1.16"
* provider.random: version = "~> 2.0"

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.
SINMAC50B9HTDH:2019 Q1 JIT kyeu$ █
```

Once the command has been successfully completed, enter the command “terraform apply”. Terraform will then generate an execution plan and show you what are the actions that will be taken.



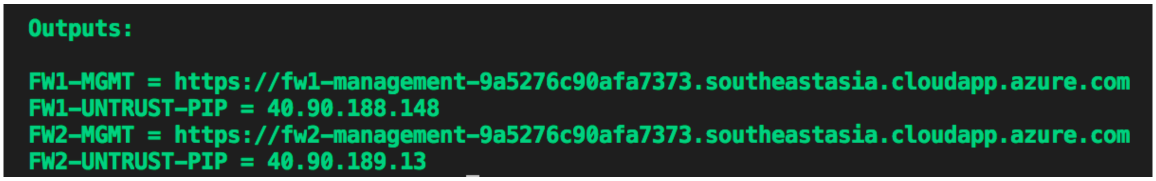
```
SINMAC50B9HTDH:2019 Q1 JIT kyeu$ terraform apply

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  + azurerm_availability_set.demo
      id:                                  <computed>
      location:                            "southeastasia"
      managed:                             "true"
      name:                                "AS-FW"
      platform_fault_domain_count:         "2"
      platform_update_domain_count:        "2"
      resource_group_name:                 "JIT-outbound-rg"
      tags.%:                              <computed>
```

You will be prompted to confirm that you want to perform the actions. Once you type “yes” Terraform will start to deploy the template to Azure.



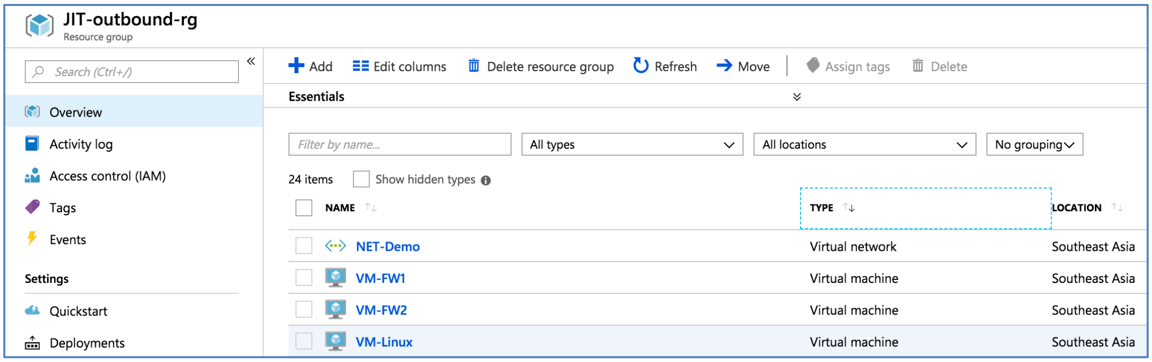
```
Plan: 30 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes
```

Once the script has been completed (takes around 6-8 minutes), you will see the following



```
Outputs:

FW1-MGMT = https://fw1-management-9a5276c90afa7373.southeastasia.cloudapp.azure.com
FW1-UNTRUST-PIP = 40.90.188.148
FW2-MGMT = https://fw2-management-9a5276c90afa7373.southeastasia.cloudapp.azure.com
FW2-UNTRUST-PIP = 40.90.189.13
```

From Azure portal, you should see the resources created in the resource group you specified.



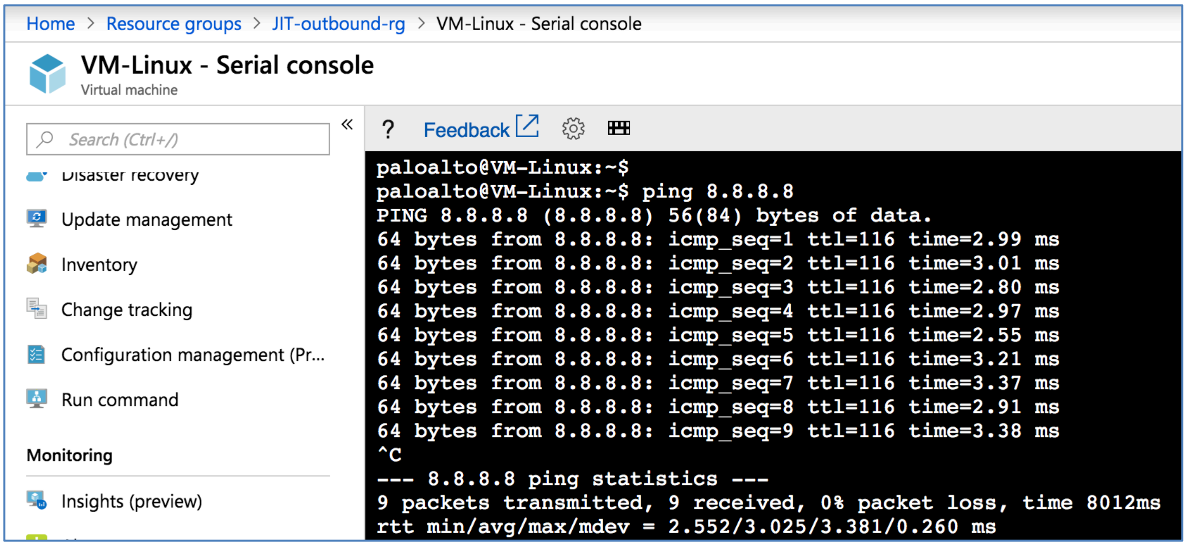
24 items

NAME	TYPE	LOCATION
NET-Demo	Virtual network	Southeast Asia
VM-FW1	Virtual machine	Southeast Asia
VM-FW2	Virtual machine	Southeast Asia
VM-Linux	Virtual machine	Southeast Asia

Wait a few more minutes for the VM-Series firewalls and Ubuntu server to fully boot up.

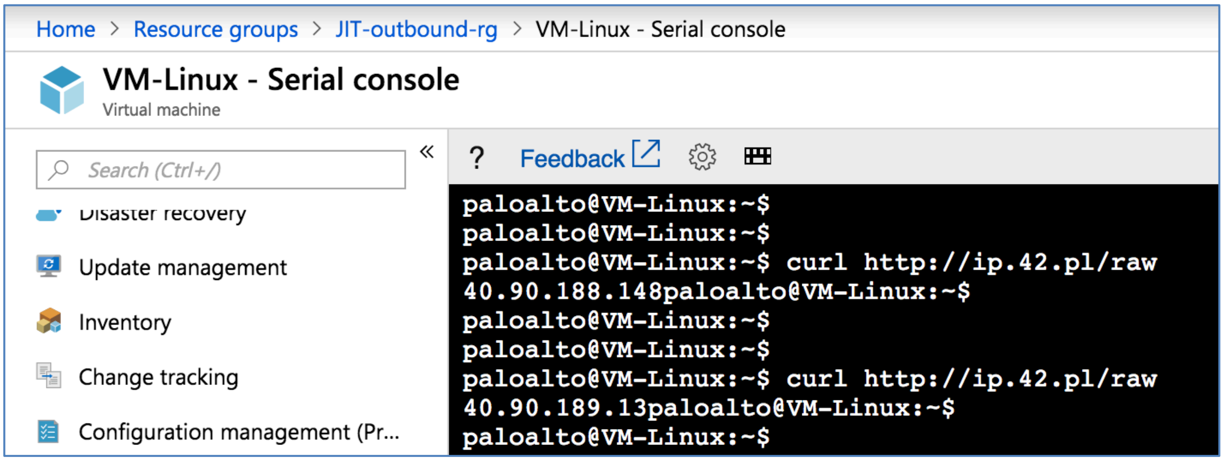
Test Outbound Connectivity

Access the Serial Console of the Ubuntu Server by navigating to “Home > Resource groups > resource-group-name > ubuntu-server-name” and then clicking on “Serial console” on the left-panel. Login using the credentials you provided in the “terraform.tfvars” file. Then issue the command “ping 8.8.8.8” and confirm that the ping is successful.



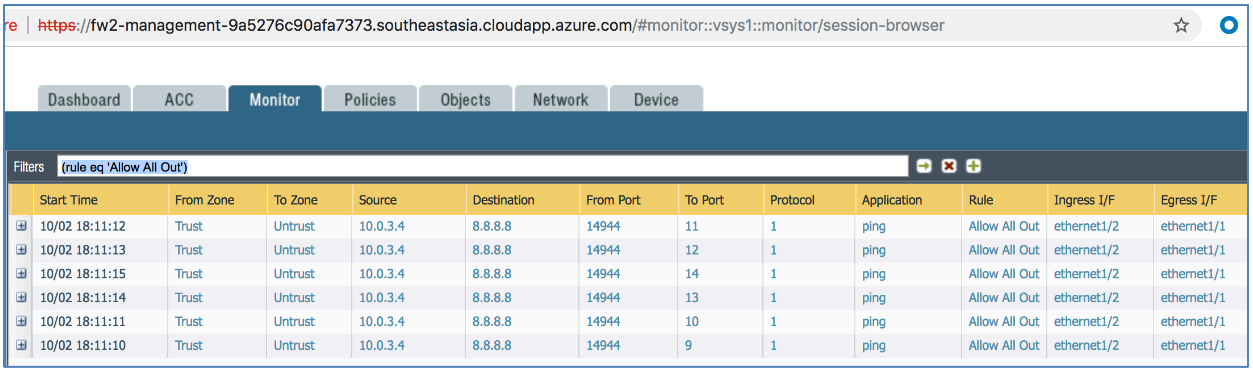
```
paloalto@VM-Linux:~$
paloalto@VM-Linux:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=2.99 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=3.01 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=2.80 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=116 time=2.97 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=116 time=2.55 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=116 time=3.21 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=116 time=3.37 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=116 time=2.91 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=116 time=3.38 ms
^C
--- 8.8.8.8 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8012ms
rtt min/avg/max/mdev = 2.552/3.025/3.381/0.260 ms
```

Then issue the command “curl http://ip.42.pl/raw” and check what is the source IP address used to access the website. Issue the command again a while later. You will see a different source IP address. Note: You might have to issue the command multiple times to see different source IP addresses being used. The two source IP addresses that you see should correspond to the public IP addresses attached to the untrust interface of the VM-Series firewalls.


Home > Resource groups > JIT-outbound-rg > VM-Linux - Serial console
VM-Linux - Serial console
Virtual machine
Search (Ctrl+/)
Disaster recovery
Update management
Inventory
Change tracking
Configuration management (Pr...
?
Feedback

```
paloalto@VM-Linux:~$
paloalto@VM-Linux:~$
paloalto@VM-Linux:~$ curl http://ip.42.pl/raw
40.90.188.148paloalto@VM-Linux:~$
paloalto@VM-Linux:~$
paloalto@VM-Linux:~$
paloalto@VM-Linux:~$ curl http://ip.42.pl/raw
40.90.189.13paloalto@VM-Linux:~$
paloalto@VM-Linux:~$
```

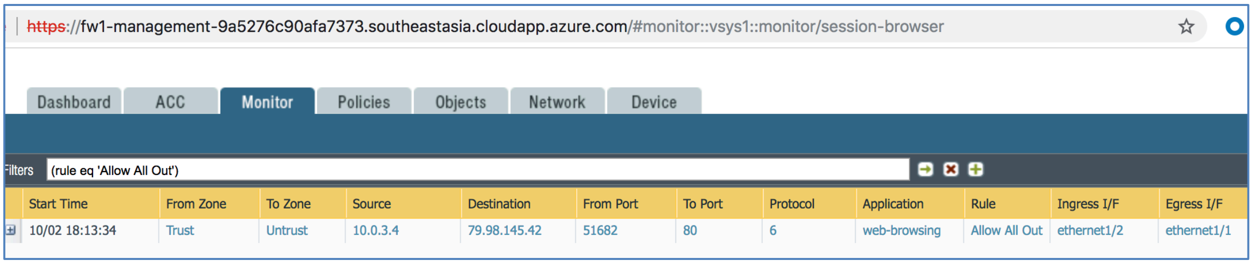
Login to the management interface of both VM-Series firewalls. You should be able to see the ping and http sessions. Please check the session logs from the firewalls immediately after you run the ping and curl commands.



~~https~~://fw2-management-9a5276c90afa7373.southeastasia.cloudapp.azure.com/#monitor::vsys1::monitor/session-browser

Filters (rule eq 'Allow All Out')

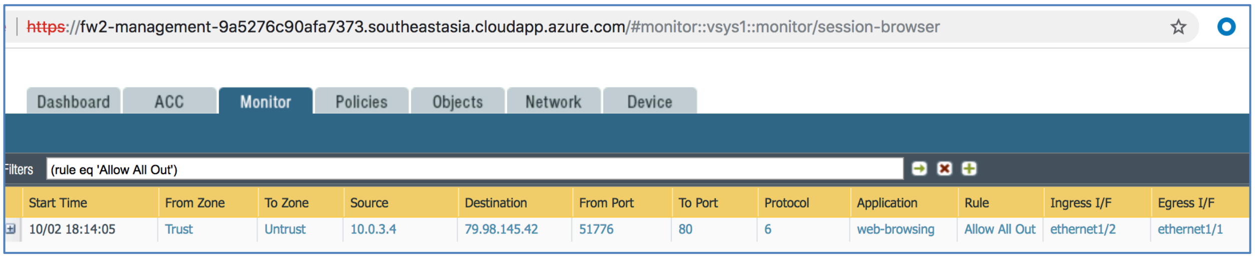
Start Time	From Zone	To Zone	Source	Destination	From Port	To Port	Protocol	Application	Rule	Ingress I/F	Egress I/F
10/02 18:11:12	Trust	Untrust	10.0.3.4	8.8.8.8	14944	11	1	ping	Allow All Out	ethernet1/2	ethernet1/1
10/02 18:11:13	Trust	Untrust	10.0.3.4	8.8.8.8	14944	12	1	ping	Allow All Out	ethernet1/2	ethernet1/1
10/02 18:11:15	Trust	Untrust	10.0.3.4	8.8.8.8	14944	14	1	ping	Allow All Out	ethernet1/2	ethernet1/1
10/02 18:11:14	Trust	Untrust	10.0.3.4	8.8.8.8	14944	13	1	ping	Allow All Out	ethernet1/2	ethernet1/1
10/02 18:11:11	Trust	Untrust	10.0.3.4	8.8.8.8	14944	10	1	ping	Allow All Out	ethernet1/2	ethernet1/1
10/02 18:11:10	Trust	Untrust	10.0.3.4	8.8.8.8	14944	9	1	ping	Allow All Out	ethernet1/2	ethernet1/1



~~https~~://fw1-management-9a5276c90afa7373.southeastasia.cloudapp.azure.com/#monitor::vsys1::monitor/session-browser

Filters (rule eq 'Allow All Out')

Start Time	From Zone	To Zone	Source	Destination	From Port	To Port	Protocol	Application	Rule	Ingress I/F	Egress I/F
10/02 18:13:34	Trust	Untrust	10.0.3.4	79.98.145.42	51682	80	6	web-browsing	Allow All Out	ethernet1/2	ethernet1/1



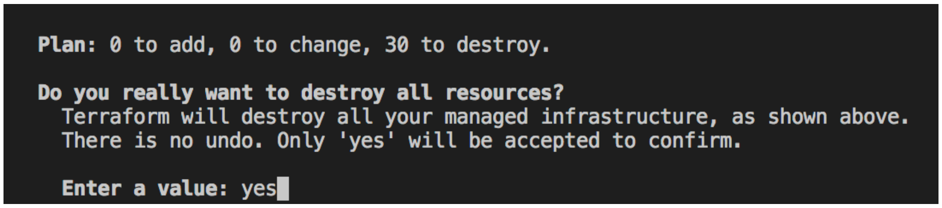
~~https~~://fw2-management-9a5276c90afa7373.southeastasia.cloudapp.azure.com/#monitor::vsys1::monitor/session-browser

Filters (rule eq 'Allow All Out')

Start Time	From Zone	To Zone	Source	Destination	From Port	To Port	Protocol	Application	Rule	Ingress I/F	Egress I/F
10/02 18:14:05	Trust	Untrust	10.0.3.4	79.98.145.42	51776	80	6	web-browsing	Allow All Out	ethernet1/2	ethernet1/1

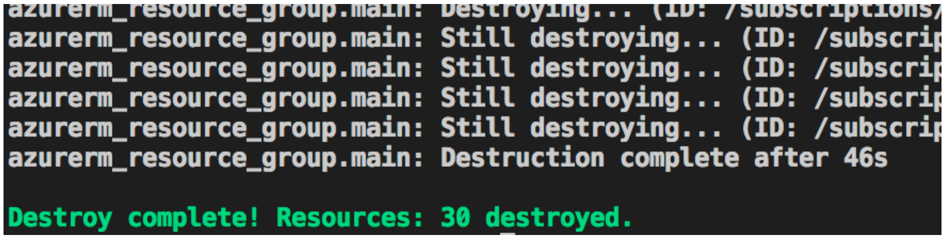
Clean Up

After the demo, you should delete the environment. Go back to your terminal window and issue the command “terraform destroy”. Again, you will be asked to confirm the action. Type “yes” to delete the environment.



```
Plan: 0 to add, 0 to change, 30 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes
```

Once deletion is completed, you will see that the destroy has been completed.



```
azurerm_resource_group.main: Destroying... (ID: /subscriptions/
azurerm_resource_group.main: Still destroying... (ID: /subscrip
azurerm_resource_group.main: Still destroying... (ID: /subscrip
azurerm_resource_group.main: Still destroying... (ID: /subscrip
azurerm_resource_group.main: Still destroying... (ID: /subscrip
azurerm_resource_group.main: Destruction complete after 46s

Destroy complete! Resources: 30 destroyed.
```

Check the Azure Portal and ensure that the resource group has been deleted.