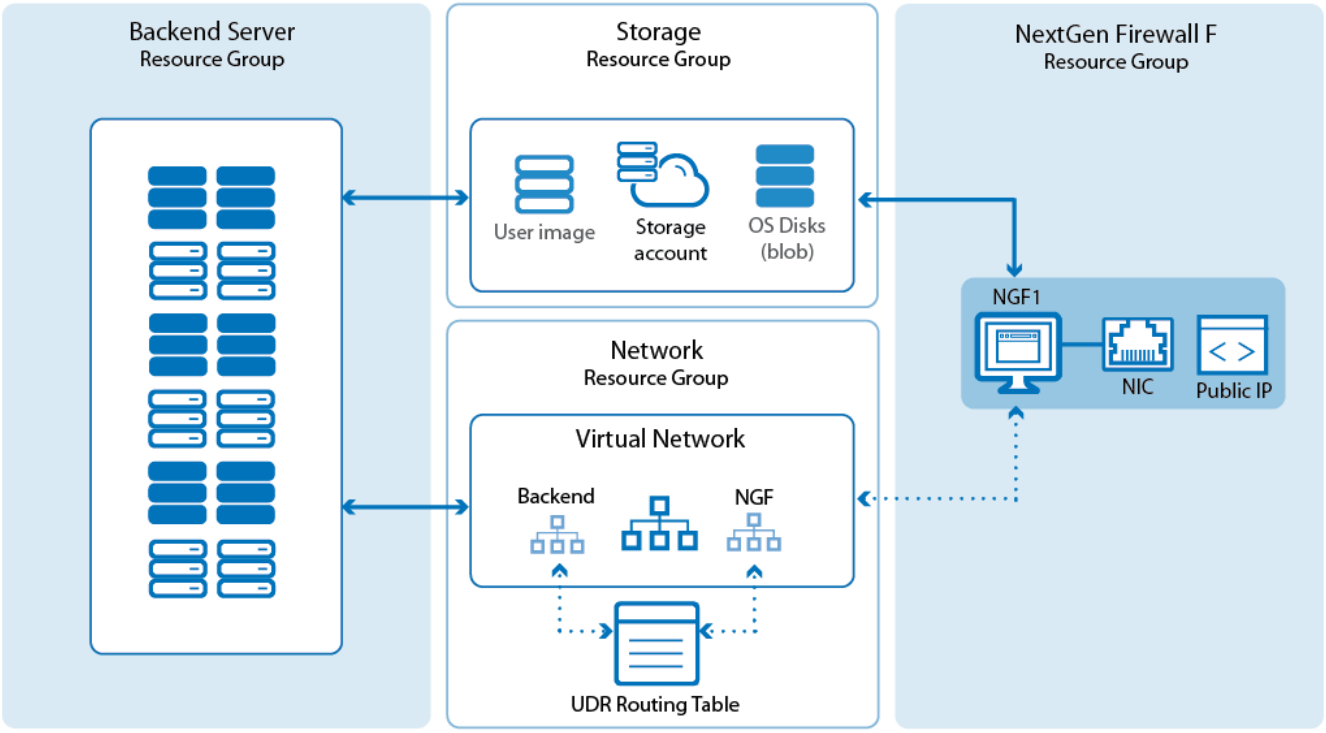
How to Deploy an F-Series Firewall in Microsoft Azure using PowerShell and ARM

https://campus.barracuda.com/doc/48203338/

For most advanced networking features in the Microsoft Azure Cloud, such as multiple network interfaces or user images, you must deploy the Barracuda NextGen Firewall F via PowerShell. You can either enter the commands directly into the Azure PowerShell or combine the commandlets to a custom deployment script. Using a custom PowerShell script allows for rapid deployment and fast recovery in case of failure. The NextGen Control Center for Microsoft Azure is deployed just like the NextGen Firewall F except that it is limited to one network interface. The maximum number of network interfaces depends on the Instance size. To organize the resources in the cloud, it is recommend to use multiple resource groups. This way it is possible to separate storage from networking and the VMs. You can also assign different permissions in Azure to control access to the resources. We are using three resource groups in total:

- **Storage resource group** – Contains the storage accounts holding user-defined images and OS disk images for the VMs.
- **Networking resource group** – Contains the Azure Virtual network. For HA clusters, the loadbalancer would also be placed in this resource group. You can also add VNET to VNET Azure VPN Gateways to this group. For stand-alone NGF VMs, you can also add the UDR route table to this resource group.
- **NextGen Firewall F resource group** – Contains the firewall VM as well as NICs, public IP addresses, and, if needed, the UDR routing table for HA clusters.

Microsoft Azure charges apply. For more information, see the Microsoft Azure Pricing Calculator.



Example deployment script

You can combine the PowerShell commandlets to customize the deployment of your Barracuda NextGen Firewall F-Series in the Microsoft Azure cloud. See below for an example deployment script. This script assumes that you already configured a virtual network and storage account and their respective resource groups and that you are logged in to your Azure Account from the PowerShell.

Fill in the variable at the top of the script, then execute it to deploy the NextGen Firewall F.

```
################################################
# Modify the variables below
################################################
# Enable verbose output and stop on error
$VerbosePreference = 'Continue'
$ErrorActionPreference = 'Stop'

# Location
$location = 'your_location' # E.g., West Europe

# Storage Account Name
$storageAccountName = 'your_storage_account_name'
$storageAccountContainerName = 'your_blob_container_name'
```

```
$storageAccountResourceGroupName = 'your_storage_resource_group_name'

# Enter to use a User Defined VM image E.g.,
https://docstorage0.blob.core.windows.net/vhds/GWAY-6.2.0-216-Azure.vhd
# Leave empty to use the latest image from the Azure Marketplace
$customSourceImageUri = ''

# Select the License type
$vmLicenseType = 'hourly' # set this to 'hourly' to use the PAYG image, or
'byol' for the BYOL image

# Set the product type
$vmProductType ='barracuda-ng-firewall' # Use 'barracuda-ng-firewall' for F-
Series Firewall or 'barracuda-ng-cc' for the NextGen Control Center

# VNET
$vnetName = 'your_virtual_network_name'
$vnetResourceGroupName = 'your_virtual_network_resource_group_name'

# Availability Set
# always set a availability set in case you want to deploy a second firewall
for HA later.
$vmAvSetName ='NGF-AV-SET'

# Static IP address for the NIC
$nic1InternalIP = '' # always make sure this IP address is available or leave
this variable empty to use the next available IP address

# Barracuda NextGen Firewall F VM settings
$NGFResourceGroupName = 'NGF_RG'
$rootPassword = 'NGf1r3wall$$'
$vmSuffix = 'NGF' #
$vmName = '{0}' -f $vmSuffix
$vmSize = 'Standard_A3'
$nicName = '{0}-NIC1' -f $vmSuffix
$nicName2 = '{0}-NIC2' -f $vmSuffix
$ipName = '{0}-IP' -f $vmSuffix
$domName = $vmSuffix.ToLower()
$diskName = 'osdisk'
$datadiskName1 = 'datadisk1'
$datadiskName2 = 'datadisk2'
$datadiskName3 = 'datadisk3'
# size of a single data disk size in GB. Multiply the size by the number of
disks to received the total disk size of the RAID device
$datadisksize = 40
```

```
##############################################
#
# No configuration variables past this point
#
##############################################

Write-Host 'Starting Deployment - this may take a while'

# Authenticate
Login-AzureRmAccount

# Create the ResourceGroup for the Barracuda NextGen Firewall F
Write-Verbose ('Creating NGF Resource Group {0}' -f $NGFresourceGroupName)
New-AzureRmResourceGroup -Name $NGFresourceGroupName -Location $location -
ErrorAction Stop


# Use existing storage account
$storageAccount = Get-AzureRmStorageAccount -Name $storageAccountName -
ResourceGroupName $storageAccountResourceGroupName

# Use an existing Virtual Network
Write-Verbose ('Using VNET {0} in Resource Group {1}' -f
$vnetNamem,$vnetResourceGroupName )
$vnet = Get-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
$vnetResourceGroupName

# Create Availability Set if it does not exist yet
$vmAvSet = New-AzureRmAvailabilitySet -Name $vmAvSetName -ResourceGroupName
$NGFResourceGroupName -Location $location -WarningAction SilentlyContinue

# Create the NIC and new Public IP
Write-Verbose 'Creating Public IP'
$pip = New-AzureRmPublicIpAddress -ResourceGroupName $NGFresourceGroupName -
Location $location -Name $ipName -DomainNameLabel $domName -AllocationMethod
Static


Write-Verbose 'Creating NIC'
if ($nic1InternalIP -eq '')
{
    $nic = New-AzureRmNetworkInterface -ResourceGroupName
$NGFresourceGroupName -Location $location -Name $nicName -PublicIpAddressId
$pip.Id -SubnetId $vnet.Subnets[0].Id -EnableIPForwarding
}
else
```

```
{
    $nic = New-AzureRmNetworkInterface -ResourceGroupName
$NGFresourceGroupName -Location $location -Name $nicName -PrivateIpAddress
$nic1InternalIP -PublicIpAddressId $pip.Id -SubnetId $vnet.Subnets[0].Id -
EnableIPForwarding
}

# NIC #2 - OPTIONAL
#$nic2 = New-AzureRmNetworkInterface -ResourceGroupName $NGFresourceGroupName
-Location $location -Name $nicName2 -SubnetId $vnet.Subnets[1].Id -
EnableIPForwarding -PrivateIpAddress $nic2IP


# Create the VM Configuration

Write-Verbose 'Creating NGF VM Configuration'

$vm = New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId
$vmAvSet.Id

# Set root password
$cred = New-Object PSCredential 'placeholderusername', ($rootPassword |
ConvertTo-SecureString -AsPlainText -Force)
$vm = Set-AzureRmVMOperatingSystem -VM $vm -Linux -ComputerName $vmName -
Credential $cred -ErrorAction Stop

# Add primary network interface
$vm = Add-AzureRmVMNetworkInterface -VM $vm -Id $nic.Id -ErrorAction Stop -
Primary

# Add NIC #2
#$vm = Add-AzureRmVMNetworkInterface -VM $vm -Id $nic2.Id -ErrorAction Stop

# generate the name for the OS disk
$osDiskUri = '{0}vhds/{1}{2}.vhd' -f
$storageAccount.PrimaryEndpoints.Blob.ToString(), $vmName.ToLower(),
$diskName

# generate URI for the datadisks
$dataDiskUri1 = '{0}vhds/{1}{2}.vhd' -f
$storageAccount.PrimaryEndpoints.Blob.ToString(), $vmName.ToLower(),
$datadiskName1
$dataDiskUri2 = '{0}vhds/{1}{2}.vhd' -f
$storageAccount.PrimaryEndpoints.Blob.ToString(), $vmName.ToLower(),
$datadiskName2
$dataDiskUri3 = '{0}vhds/{1}{2}.vhd' -f
```

```
$storageAccount.PrimaryEndpoints.Blob.ToString(), $vmName.ToLower(),
$datadiskName3

# Set the name and storage for the OS Disk image.
$vm = Set-AzureRmVMOSDisk -VM $vm -Name $diskName -VhdUri $osDiskUri -
CreateOption fromImage


# Specify the OS disk with user image
if ($customSourceImageUri -eq '')
{
    Write-Verbose 'Using lasted image from the Azure Marketplace'
    $vm.Plan = @{'name'= $vmLicenseType; 'publisher'= 'barracudanetworks';
'product' = $vmProductType}
    $vm = Set-AzureRmVMSourceImage -VM $vm -PublisherName 'barracudanetworks'
-Skus $vmLicenseType -Offer $vmProductType -Version 'latest' -ErrorAction
Stop
    $vm = Set-AzureRmVMOSDisk -VM $vm -Name $diskName -VhdUri $osDiskUri -
CreateOption fromImage
}
else
{
    Write-Verbose ('Using user defined image {0}' -f $customSourceImageUri)
    $vm = Set-AzureRmVMOSDisk -VM $vm -Name $diskName -VhdUri $osDiskUri -
CreateOption fromImage -SourceImageUri $customSourceImageUri -Linux
}

# add the datadisks
Write-Verbose 'Adding data disks'
$vm = Add-AzureRmVMDataDisk -VM $vm -Name $datadiskName1 -DiskSizeInGB
$datadisksize -CreateOption Empty -Lun 1 -VhdUri $dataDiskUri1
$vm = Add-AzureRmVMDataDisk -VM $vm -Name $datadiskName2 -DiskSizeInGB
$datadisksize -CreateOption Empty -Lun 2 -VhdUri $dataDiskUri2
$vm = Add-AzureRmVMDataDisk -VM $vm -Name $datadiskName3 -DiskSizeInGB
$datadisksize -CreateOption Empty -Lun 3 -VhdUri $dataDiskUri3

Write-Verbose 'Creating Barracuda NextGen Firewall F VM. This can take a
while ....'
$result = New-AzureRmVM -ResourceGroupName $NGFresourceGroupName -Location
$location -VM $vm


if($result.IsSuccessStatusCode -eq 'True') {
   $result
   Write-Host ('Barracuda NextGen Firewall F VM ''{0}'' was successfully
deployed.  Connect to the firewall at {2} with the username: root and
```

```
password: {1}' -f $vmName, $rootPassword, (Get-AzureRmPublicIpAddress -
ResourceGroupName $NGFResourceGroupName -Name $ipName).IpAddress)
} else {
    Write-Host ('Deployment Failed. {0}' -f $result.ReasonPhrase)
}
```

Before you begin

- Install Azure PowerShell version 3.1.0 or higher.
- Log into your Azure account with `Login-AzureRmAccount`.
- Purchase a Barracuda NextGen Firewall F or Control Center for Azure license, or request an evaluation license from the Barracuda Networks Evaluation page.

Step 1. Store location in a variable

It is required that all resource groups and their resources be in the same location. Store the location to a variable.

1. Open the Azure PowerShell.
2. Store the location to a variable

   For a list of available locations, enter:
   PS C:\> ((Get-AzureRmResourceProvider -ProviderNamespace Microsoft.Compute).ResourceTypes | Where-Object ResourceTypeName -eq virtualMachines).Locations

   ```
   $location = 'YOUR_LOCATION'
   ```

Step 2. Create an Azure VNET

Create an Azure Virtual Network (VNET). The F-Series Firewall VM must be deployed into its own subnet for user defined Azure route tables to be applied. Create additional subnets for the backend VMs. These VMs connect to the Internet or your on-premises resources through the firewall VM. To be able to easily replace the VMs, it is recommended to use a separate resource group for the virtual network.

1. Open the Azure PowerShell.
2. (recommended) Create an Azure resource group for the networking resources:

```
New-AzureRmResourceGroup -Name NETWORK_RESOURCE_GROUP_NAME -Location
$location
```

```
PS C:\> New-AzureRmResourceGroup -Name DOC-Networking -Location 'West Europe'

ResourceGroupName : DOC-Networking
Location          : westeurope
ProvisioningState : Succeeded
Tags              :
ResourceId        : /subscriptions/[redacted]/resourceGroups/DOC-Networking
```

3. Define the subnets for the firewall and the backend, and then create the virtual network. Select an address prefix that does not overlap with your on-premise network.

```
$NGFSubnet = New-AzureRmVirtualNetworkSubnetConfig -Name NGF_SUBNET_NAME
-AddressPrefix 10.8.1.0/24
$backendSubnet = New-AzureRmVirtualNetworkSubnetConfig -Name
BACKEND_SUBNET_NAME -AddressPrefix 10.8.2.0/24
New-AzureRmVirtualNetwork -Name VNET_NAME -ResourceGroupName
NETWORKING_RG_NAME -Location $location -AddressPrefix 10.8.0.0/16 -
Subnet $NGFSubnet, $backendSubnet
```

```
PS C:\> $NGFSubnet = New-AzureRmVirtualNetworkSubnetConfig -Name NGF -AddressPrefix 10.8.1.0/24
PS C:\> $backendSubnet = New-AzureRmVirtualNetworkSubnetConfig -Name Backend -AddressPrefix 10.8.2.0/24
PS C:\> New-AzureRmVirtualNetwork -Name DOC-VNET1 -ResourceGroupName DOC-Networking -Location 'West Europe' -AddressPref
ix 10.8.0.0/16 -Subnet $NGFSubnet, $backendSubnet


Name              : DOC-VNET1
ResourceGroupName : DOC-Networking
Location          : westeurope
Id                : /subscriptions/bde58b49-9951-466e-90e2-592c0920ce77/resourceGroups/DOC-Networking/providers/Microso
                    ft.Network/virtualNetworks/DOC-VNET1
Etag              : W/"3725676f-489d-4b53-846b-c6b5a578cf96"
ResourceGuid      : 33ca442b-3adc-4150-9a07-2b0cdddf27a2
ProvisioningState : Succeeded
Tags              :
AddressSpace      : {
                      "AddressPrefixes": [
                        "10.8.0.0/16"
                      ]
                    }
DhcpOptions       : {}
Subnets           : [
                      {
                        "Name": "NGF",
                        "Etag": "W/\"3725676f-489d-4b53-846b-c6b5a578cf96\"",
                        "Id": "/subscriptions/bde58b49-9951-466e-90e2-592c0920ce77/resourceGroups/DOC-Networking/provid
                    ers/Microsoft.Network/virtualNetworks/DOC-VNET1/subnets/NGF",
                        "AddressPrefix": "10.8.1.0/24",
                        "IpConfigurations": [],
                        "ProvisioningState": "Succeeded"
                      },
                      {
                        "Name": "Backend",
                        "Etag": "W/\"3725676f-489d-4b53-846b-c6b5a578cf96\"",
                        "Id": "/subscriptions/bde58b49-9951-466e-90e2-592c0920ce77/resourceGroups/DOC-Networking/provid
                    ers/Microsoft.Network/virtualNetworks/DOC-VNET1/subnets/Backend",
                        "AddressPrefix": "10.8.2.0/24",
                        "IpConfigurations": [],
                        "ProvisioningState": "Succeeded"
                      }
                    ]
```

You can now deploy the firewall VM to the NGF subnet.

Step 3. Create an Azure storage account

To be able to use user-defined images, you must create an Azure storage account that is not in the resource group the firewall VM is deployed to. This allows you to delete the resource group the firewall is in without having to re-upload the VHD disk images. Skip this step to use an existing Azure storage account.

1. Open an Azure PowerShell.
2. Create a resource group for the your storage account(s). The name of the storage account must be lowercase letters and numbers only.

   ```
   New-AzureRmResourceGroup -Name RESOURCE_GROUP_NAME -Location $location
   ```

   ```
   PS C:\> New-AzureRmResourceGroup -Name DOC-Storage -Location 'West Europe'

   ResourceGroupName : DOC-Storage
   Location          : westeurope
   ProvisioningState : Succeeded
   Tags              :
   ResourceId        : /subscriptions/bde58b49-9951-466e-90e2-592c0920ce77/resourceGroups/DOC-Storage
   ```

3. Create a storage account.

   ```
   New-AzureRmStorageAccount -ResourceGroupName RG_NAME -Name
   STORAGE_ACCOUNT_NAME -Type Standard_LRS -Location $location
   ```

   ```
   PS C:\> New-AzureRmStorageAccount -ResourceGroupName DOC-Storage -Name docstorage1 -Type Standard_LRS -Location "West Eu
   rope"

   ResourceGroupName    : doc-storage
   StorageAccountName   : docstorage1
   Id                   : /subscriptions/[redacted]/resourceGroups/doc-storage/providers/Microsof
                          t.Storage/storageAccounts/docstorage1
   Location             : westeurope
   AccountType          : StandardLRS
   CreationTime         : 11.01.2016 13:55:29
   CustomDomain         :
   LastGeoFailoverTime  :
   PrimaryEndpoints     : Microsoft.Azure.Management.Storage.Models.Endpoints
   PrimaryLocation      : westeurope
   ProvisioningState    : Succeeded
   SecondaryEndpoints   :
   SecondaryLocation    :
   StatusOfPrimary      : Available
   StatusOfSecondary    :
   Tags                 : {}
   Context              : Microsoft.WindowsAzure.Commands.Common.Storage.AzureStorageContext
   ```

Step 4. Create a resource group for the Firewall VM

Create the resource group for the F-Series Firewall VM.

1. Open an Azure PowerShell.
2. Create the resource group:

   ```
   New-AzureRmResourceGroup -Name NGF_RESOURCE_GROUP_NAME -Location
   $location
   ```

Step 5. Create an availability set

To be able to add the firewall to a high availability cluster later, you need to add it to an availability

set.

1. Open an Azure PowerShell.
2. Create the availability set:

```
# Create Availability Set
$vmAvSet = New-AzureRmAvailabilitySet -Name AV_SET_NAME -
ResourceGroupName NGF_RESOURCE_GROUP_NAME -Location $location
```

```
PS C:\>
PS C:\> $vmAvSet = New-AzureRmAvailabilitySet -Name 'NGF-AVSET' -ResourceGroupName 'DOC-NGF' -Location 'West Europe'
PS C:\>
```

Step 6. Create network interfaces and public IP

Create the network interface(s) and the public IP address to use for the VM. Multiple network interfaces must be supported by the Azure instance the firewall is deployed on. Using multiple network interfaces is not possible if you want to use the VM in a high availability cluster.

1. Open an Azure PowerShell.
2. Store the virtual network in a variable:

```
$vnet = Get-AzureRmVirtualNetwork -Name VNET_NAME -ResourceGroupName
NETWORKING_RESOURCE_GROUP_NAME
```

```
PS C:\> $vnet = Get-AzureRmVirtualNetwork -Name DOC-VNET -ResourceGroupName DOC-Networking
PS C:\>
```

3. Create a static Azure public IP:

```
$pip = New-AzureRmPublicIpAddress -ResourceGroupName
NGF_RESOURCE_GROUP_NAME -Location $location -Name PIP_NAME -
DomainNameLabel DOMAIN_NAME -AllocationMethod Static
```

```
PS C:\> $pip = New-AzureRmPublicIpAddress -ResourceGroupName DOC-NGF -Location "West Europe" -Name "NGF-PIP" -DomainName
Label "doc-ngf" -AllocationMethod Static
PS C:\>
```

4. Create the first network interface:

```
$nic = New-AzureRmNetworkInterface -ResourceGroupName
NGF_RESOURCE_GROUP_NAME  -Location $location -Name NIC1_NAME -
PublicIpAddressId $pip.Id -SubnetId  $vnet.Subnets[0].Id -
EnableIPForwarding
```

```
PS C:\> $nic = New-AzureRmNetworkInterface -ResourceGroupName DOC-NGF -Location "West Europe" -Name DOC-NGF-NIC -Private
IpAddress 10.8.1.10 -SubnetId $vnet.Subnets[0].Id -EnableIPForwarding
PS C:\>
```

5. (optional) To use multiple NICs on instances that support it, create a second network interface. Multiple network interfaces are not possible for HA deployments.

```
$nic2 = New-AzureRmNetworkInterface -ResourceGroupName
```

```
NGF_RESOURCE_GROUP_NAME -Location $location -Name NIC2_NAME -SubnetId
$vnet.Subnets[1].Id
```

Step 7. Create the firewall VM configuration and deploy the VM

Create the configuration for the F-Series Firewall VM and deploy the VM.

1. Open an Azure PowerShell.
2. Store the storage account in a variable:

```
$storageAccount = Get-AzureRmStorageAccount -Name STORAGE_ACCOUNT_NAME -
ResourceGroupName STORAGE_RESOURCE_GROUP_NAME
```

```
PS C:\> $storageAccount = Get-AzureRmStorageAccount -Name docstorage0 -ResourceGroupName DOC-Storage
PS C:\>
```

3. Create the VM configuration:

```
$vm = New-AzureRmVMConfig -VMName VM_NAME -VMSize VM_SIZE -
AvailabilitySetId $vmAvSet.Id
```

```
PS C:\>
PS C:\> $vm = New-AzureRmVMConfig -VMName DOC-NGF1 -VMSize 'Standard_A3' -AvailabilitySetId $vmAvSet.Id
PS C:\>
```

4. Create the credentials objects for the VM. The username must be entered, but is ignored by the firewall VM. Make sure the password matches the Microsoft Azure password requirements. E.g., NGF1r3wall$$

```
$cred = New-Object pscredential 'placeholderusername',
('YOUR_ROOT_PASSWORD' | ConvertTo-SecureString -AsPlainText -Force)
```

```
PS C:\> $cred = New-Object pscredential 'placeholderusername', ('NGf1r3wall$$' | ConvertTo-SecureString -AsPlainText -Force
PS C:\>
```

5. Set the operating system type to Linux and credentials for the VM:

```
$vm = Set-AzureRmVMOperatingSystem -VM $vm -Linux -ComputerName
NAME_OF_VM -Credential $cred
```

```
PS C:\> $vm=Set-AzureRmVMOperatingSystem -VM $vm -Linux -ComputerName "DOC-NGF1" -Credential $cred
PS C:\>
```

6. Add the network interface created in step 3 to the VM:

```
$vm = Add-AzureRmVMNetworkInterface -VM $vm -Id $nic.Id -Primary
```

```
PS C:\> $vm = Add-AzureRmVMNetworkInterface -VM $vm -Id $nic.Id -Primary
PS C:\>
```

7. (optional) Add the second network interface to the VM:

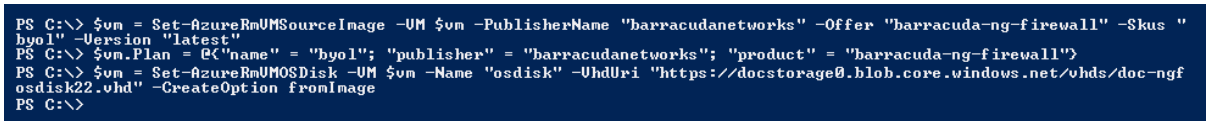
```
$vm = Add-AzureRmVMNetworkInterface -VM $vm -Id $nic2.Id
```

8. Set the OS disk:

- Set the plan information and source image to use the latest Marketplace image. To use the PAYG image, set **Skus** to `hourly`. Otherwise, set **Skus** to `byol` for the BYOL image:

  The **VhdUri** is determined as follows: **BLOB endpoint of your storage account + container name + disk name with the extension.vhd.** E.g., https://docstorage0.blob.core.windows.net/vhds/NGF1.vhd
  The BLOB endpoint of your storage endpoint can be obtained by entering: `$storageAccount.PrimaryEndpoints.Blob` in Azure PowerShell. The disk name must be unique.

```
$vm.Plan = @{'name'= 'byol'; 'publisher'= 'barracudanetworks';
'product' = 'barracuda-ng-firewall'}
$vm = Set-AzureRmVMSourceImage -VM $vm -PublisherName
'barracudanetworks' -Skus 'byol' -Offer 'barracuda-ng-firewall' -
Version  'latest'
$vm = Set-AzureRmVMOSDisk -VM $vm -Name $diskName -VhdUri
URI_TO_OS_DISK -CreateOption fromImage
```



- User Image (uploaded VHD). For more information, see How to Upload Azure VHD Images for User Defined Images using ARM

```
$vm = Set-AzureRmVMOSDisk -VM $vm -Name NAME_OF_DISK -VhdUri
DISK_URI -CreateOption fromImage -Linux
```

9. Add the datadisks to the VM. The data disk URIs are generated like the **VhdUri**. Each URI must be unique.

```
$vm = Add-AzureRmVMDataDisk -VM $vm -Name NAME_OF_DATA_DISK1 -
DiskSizeInGB DATA_DISK_SIZE_IN_GB -CreateOption Empty -Lun 1 -VhdUri
DATA_DISK1_URI
$vm = Add-AzureRmVMDataDisk -VM $vm -Name NAME_OF_DATA_DISK2 -
DiskSizeInGB DATA_DISK_SIZE_IN_GB -CreateOption Empty -Lun 2 -VhdUri
DATA_DISK2_URI
$vm = Add-AzureRmVMDataDisk -VM $vm -Name NAME_OF_DATA_DISK3 -
DiskSizeInGB DATA_DISK_SIZE_IN_GB -CreateOption Empty -Lun 3 -VhdUri
DATA_DISK3_URI
```

10. Create the firewall VM:

```
New-AzureRmVM -ResourceGroupName NGF_RESOURCE_GROUP_NAME -Location
$location -VM $vm
```

```
PS C:\> New-AzureRmVM -ResourceGroupName DOC-NGF -Location "West Europe" -VM $vm

Status                : Succeeded
StatusCode            : OK
RequestId             : f4dec201-d1fa-4ed2-92f2-2de9077f69a6
Output                :
Error                 :
StartTime             : 14.01.2016 12:41:26 +01:00
EndTime               : 14.01.2016 12:50:29 +01:00
TrackingOperationId   : bc647f47-4c88-4087-87e5-af51e7f0e943
```

Step 8. (optional) Network security groups (NSG)

You can put network security groups in place as an additional safeguard to isolate your backend subnets in case the Azure routing table fails or is misconfigured. Network security groups can be associated with a network interface attached to a VM or a subnet of a virtual network. Each NSG can include up to 200 rules for incoming and outgoing traffic. NSG rules can only be created for TCP and UDP traffic. ICMP is always allowed inside the virtual network. You do not need an NSG for the firewall VM.

Step 9. Get the IP address for the F-Series Firewall VM

To connect to the Barracuda NextGen Firewall F VM you just deployed in Azure, you must find out the public IP address that is assigned to the VM.

1. Open an Azure PowerShell.
2. Get the public IP address for the firewall VM:

```
(Get-AzureRmPublicIpAddress -ResourceGroupName NGF_RESOURCE_GROUP_NAME -
Name PUBLIC_IP_NAME).IpAddress
```

```
PS C:\> (Get-AzureRmPublicIpAddress -ResourceGroupName DOC-NGF -Name DOC-NG80129-IP).IpAddress
40.113.122.18
PS C:\>
```

Step 10. Configure Barracuda NextGen Admin

Verify that Barracuda NextGen Admin is configured to use SPoE as the connection method.

1. Launch Barracuda NextGen Admin.
2. Verify that SPoE is enabled in the NextGen Admin settings. For more information, see NextGen Admin Settings.

3. Select **Box**.
4. Enter the login information:
   - **Management IP** – Enter the public IP address of your firewall VM from step 5.
   - **Username** – Enter `root`.
   - **Password** – Enter the password you set during deployment.
5. Click **Log In**.

You are now successfully logged into your Barracuda NextGen Firewall F VM.

Next steps

- Activate the license. For more information, see How to Activate and License a Stand-alone Virtual or Public Cloud F-Series Firewall or Control Center.
- (**Important!**) Limit access to the management ports of the Barracuda CloudGen Firewall (TCP/807 and TCP/22) to only specific source IP addresses. For more information, see How to Change the Root Password and Management ACL.
- Configure UDR Azure route table. For more information, see How to Configure Azure Route Tables (UDR) using PowerShell and ARM.
- Configure Azure cloud integration on the firewall. For more information, see How to Configure Azure Cloud Integration using ARM.
- To use two firewalls in a high availability (HA) cluster, see How to Configure a High Availability Cluster in Azure using PowerShell and ARM.

Figures

1. azure_arm_single_backend_diagram-01.png
2. ARM_PS_00.png
3. ARM_PS_01.png
4. ARM_PS_02.png
5. ARM_PS_03.png
6. azure_vhd_upload_02.png
7. ARM_PS_02a.png
8. ARM_PS_04.png
9. ARM_PS_05.png
10. ARM_PS_06.png
11. ARM_PS_07.png
12. ARM_PS_08.png
13. ARM_PS_09.png
14. ARM_PS_10.png
15. ARM_PS_11.png
16. ARM_PS_12.png
17. ARM_PS_13.png
18. ARM_PS_Get_PIP_01.png