

Azure Firewall

Most Azure networking components used for security are there to stop unwanted incoming traffic. Whether we use network security groups, application security groups, or a **Web Application Firewall (WAF)**, they all have one single purpose—to stop unwanted traffic from reaching our services. Azure Firewall has similar functionality, including one extension that we can use to stop outbound traffic from leaving the virtual network.

We will cover the following recipes in this chapter:

- Creating a new firewall
- Creating a new firewall with PowerShell
- Configuring a new allow rule
- Configuring a new deny rule
- Configuring a route table
- Enabling diagnostic logs for Azure Firewall
- Configuring Azure Firewall in forced tunneling mode
- Creating an IP group
- Configuring Azure Firewall DNS settings

Technical requirements

For this chapter, the following is required:

- An Azure subscription
- Azure PowerShell

The code samples can be found at <https://github.com/PacktPublishing/Azure-Networking-Cookbook-Second-Edition/tree/master/Chapter07>.

Creating a new firewall

Azure Firewall gives us total control over our traffic. Besides controlling inbound traffic, with Azure Firewall, we can control outbound traffic as well.

Getting ready

Before we can create an Azure Firewall instance, we must first prepare a subnet.

In order to create a new subnet for Azure Firewall, we must do the following:

1. Locate the virtual network that will be associated with our Azure Firewall.
2. Select the **Subnets** option under **Settings** and click **Subnet** to add a new subnet, as shown in *Figure 7.1*:

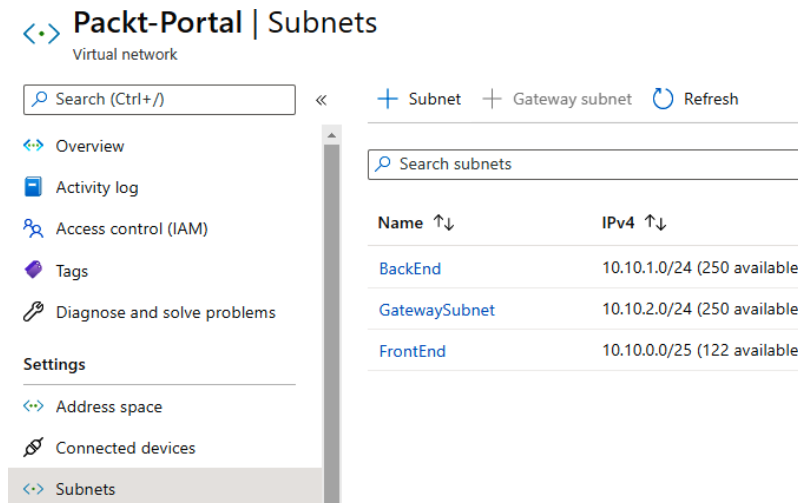


Figure 7.1: Adding a new subnet

3. In the new pane, we must provide values for the **Name** and **Address range** fields. It's very important that the subnet is named **AzureFirewallSubnet**:

Add subnet

✕

Packet-Portal

Name *

AzureFirewallSubnet ✓

Address range (CIDR block) * ⓘ

10.10.3.0/24 ✓

10.10.3.0 - 10.10.3.255 (251 + 5 Azure reserved addresses)

NAT gateway ⓘ

None ▼

☐ Add IPv6 address space

Network security group

None ▼

Route table

None ▼

Service endpoints

Services ⓘ

0 selected ▼

Subnet delegation

Delegate subnet to a service ⓘ

None ▼

Figure 7.2: Providing the name and address range of the subnet

How to do it...

In order to create a new Azure Firewall instance using the Azure portal, take the following steps:

1. In the Azure portal, select **Create a resource** and choose **Azure Firewall** under **Networking** services (or search for **Azure Firewall** in the search bar).
2. In the new pane, first, we must provide values for the **Subscription** and **Resource group** drop-down menus. We need to fill in the **Name** and **Region** fields for Azure Firewall, and optionally select an **Availability zone** option. Next, we proceed to virtual network selection. Only virtual networks in the region where the Azure Firewall instance will be created are available. Also, the selected virtual network must contain the **AzureFirewallSubnet** subnet we created earlier. Finally, we define a public IP address (we can choose an existing one or create a new one). Optionally, we can enable **Forced tunneling**:

Project details

Subscription *

Resource group * [Create new](#)

Instance details

Name * ✓

Region *

Availability zone ⓘ

Choose a virtual network ☐ Create new ☒ Use existing

Virtual network

Public IP address * [Add new](#)

Forced tunneling ⓘ ☐ Disabled

Figure 7.3: Adding Azure Firewall details

How it works...

Azure Firewall uses a set of rules to control outbound traffic. We can either block everything by default and allow only whitelisted traffic, or we can allow everything and block only blacklisted traffic. It's essentially the central point where we can set network policies, enforce these policies, and monitor network traffic across virtual networks or even subscriptions. As a firewall as a service, Azure Firewall is a managed service with built-in high availability and scalability.

Creating a new firewall with PowerShell

Alternatively, we can deploy Azure Firewall using PowerShell. This method is especially useful when services are part of a large deployment or any deployment that needs to be automated.

How to do it...

There are several steps that need to be executed in order to create a new firewall with Azure PowerShell:

1. First, we define the parameters:

```
$RG="Packt-Networking-Script"
$Location="West Europe"
$VNetName = "Packt-Script"
$AzFwIpName = "AzFW-Public-IP"
$AzFwname = "AzFw-Script"
```

2. Then, we need to create a separate subnet for Azure Firewall:

```
$vnet = Get-AzVirtualNetwork -ResourceGroupName $RG '
-Name $VnetName
Add-AzVirtualNetworkSubnetConfig -Name AzureFirewallSubnet '
-VirtualNetwork $vnet '
-AddressPrefix 10.11.3.0/24
Set-AzVirtualNetwork -VirtualNetwork $vnet
```

3. Next, we need to create a public IP address for Azure Firewall:

```
$AzFwIp = New-AzPublicIpAddress -Name $AzFwIpName '
-ResourceGroupName $RG '
-Location $Location '
-AllocationMethod Static '
-Sku Standard
```

4. Finally, we have all the components in place and can proceed to create the firewall:

```
$Azfw = New-AzFirewall -Name $AzFwname '
-ResourceGroupName $RG '
-Location $Location '
-VirtualNetworkName $vnet.Name '
-PublicIpName $AzFwIp.Name
```

How it works...

The firewall requires a separate subnet that is named **AzureFirewallSubnet**. So, we need to create such a subnet on the virtual network we intend to use. Another requirement is a public IP address. Finally, we are ready for deployment and can create a new Azure Firewall instance.

But deploying Azure Firewall is just the start. We need to configure our firewall by creating rules and routes. Let's proceed to the next recipe and see how rules are created.

Configuring a new allow rule

If we want to allow specific traffic, we must create an allow rule. Rules are applied based on priority level, so a rule will be applied only when there is no other rule with higher priority.

Getting ready

Open the PowerShell console and make sure you are connected to your Azure subscription.

How to do it...

In order to create a new allow rule in Azure Firewall, execute the following command:

```
$RG="Packt-Networking-Script"
$Location="West Europe"
$Azfw = Get-AzFirewall -ResourceGroupName $RG
$Rule = New-AzFirewallApplicationRule -Name Rule1 -Protocol
"http:80","https:443" -TargetFqdn "*packt.com"
$RuleCollection = New-AzFirewallApplicationRuleCollection -Name
RuleCollection1 -Priority 100 -Rule $Rule -ActionType "Allow"
$Azfw.ApplicationRuleCollections = $RuleCollection
Set-AzFirewall -AzureFirewall $Azfw
```

How it works...

An allow rule in Azure Firewall will whitelist specific traffic. If there is a rule that would also block this traffic, the higher-priority rule will be applied.

We can create deny rules as well. Let's see how we can do that in the next recipe.

Configuring a new deny rule

If we want to deny specific traffic, we must create a deny rule. Rules are applied by priority, so this rule will be applied only if there is not a higher-priority rule in effect.

Getting ready

Open the PowerShell console and make sure you are connected to your Azure subscription.

How to do it...

In order to create a new deny rule in Azure Firewall, execute the following command:

```
$RG="Packt-Networking-Script"
$Location="West Europe"
$Azfw = Get-AzFirewall -ResourceGroupName $RG
$Rule = New-AzFirewallApplicationRule -Name Rule1 -Protocol
"http:80","https:443" -TargetFqdn "*google.com"
$RuleCollection = New-AzFirewallApplicationRuleCollection -Name
RuleCollection1 -Priority 100 -Rule $Rule -ActionType "Deny"
$Azfw.ApplicationRuleCollections = $RuleCollection
Set-AzFirewall -AzureFirewall $Azfw
```

How it works...

The deny rule is the most commonly used option with Azure Firewall. An approach where you block everything and allow only whitelisted traffic isn't very practical, as we may end up adding a great many allow rules. Therefore, the most common approach is to use deny rules to block certain traffic that we want to prevent.

Configuring a route table

Route tables are commonly used with Azure Firewall when there is cross-connectivity. Cross-connectivity can either be with other Azure virtual networks or with on-premises networks. In such cases, Azure Firewall uses route tables to forward traffic based on the rules specified in the route tables.

Getting ready

Open the PowerShell console and make sure you are connected to your Azure subscription.

How to do it...

In order to create a new route table in Azure Firewall, execute the following command:

```
$RG="Packt-Networking-Script"
$Location="West Europe"
$Azfw = Get-AzFirewall -ResourceGroupName $RG
$config = $Azfw.IpConfigurations[0].PrivateIpAddress
$Route = New-AzRouteConfig -Name 'Route1' -AddressPrefix 0.0.0.0/0 -NextHopType
VirtualAppliance -NextHopIpAddress $config
$RouteTable = New-AzRouteTable -Name 'RouteTable1' -ResourceGroupName $RG
-location $Location -Route $Route
```

How it works...

Using route tables associated with Azure Firewall, we can define how traffic between networks is handled and how we route traffic from one network to another. In a multi-network environment, especially in a hybrid network where we connect an Azure virtual network with a local on-premises network, this option is very important. This allows us to determine what kind of traffic can flow where and how.

Enabling diagnostic logs for Azure Firewall

Diagnostics are a very important part of any IT system, and networking is no exception. The diagnostics settings in Azure Firewall allow us to collect various information that can be used for troubleshooting or auditing.

Getting ready

Before you start, open your browser and go to the Azure portal at <https://portal.azure.com>.

How to do it...

To enable diagnostics in Azure Firewall, we must follow these steps:

1. In the Azure Firewall pane, locate **Diagnostics settings** under **Monitoring**.
2. Select the **Add diagnostic setting** option, as shown in *Figure 7.4*:

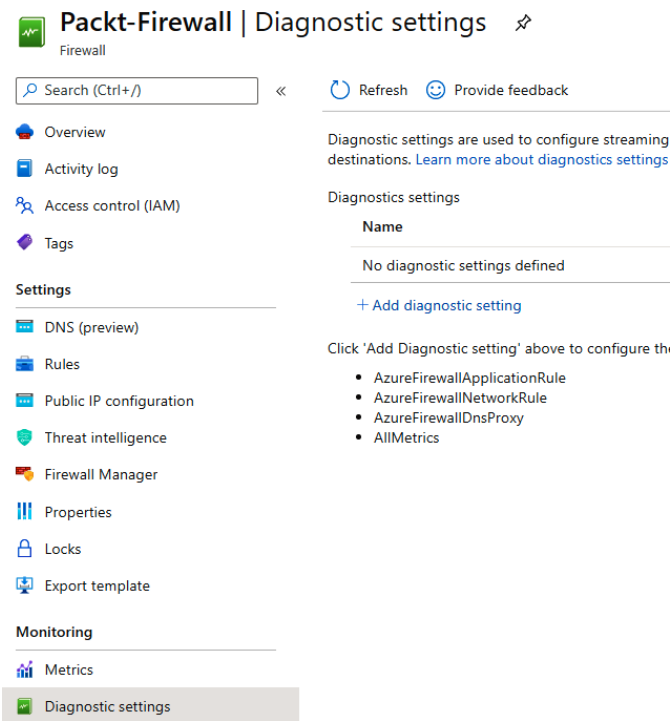


Figure 7.4: Adding a diagnostic setting

3. In the new pane, fill in the name field and specify where the logs will be stored. Choose the storage account where the logs will be stored, and specify the retention period and which logs will be stored, as shown in *Figure 7.5*:

Diagnostic setting name * Packt-Firewall ✓

Category details

log	Retention (days)
<input checked="" type="checkbox"/> AzureFirewallApplicationRule	90 ✓
<input checked="" type="checkbox"/> AzureFirewallNetworkRule	90 ✓
<input checked="" type="checkbox"/> AzureFirewallDnsProxy	90 ✓

metric

	Retention (days)
<input checked="" type="checkbox"/> AllMetrics	90 ✓

Destination details

☐ Send to Log Analytics

☒ Archive to a storage account

Location
West Europe

Subscription
Microsoft Azure Sponsorship

Storage account *
packtnetworkingportal251

☐ Stream to an event hub

Retention policy
Retention only applies to storage account. Retention policy ranges from 1 to 365 days. If you do not want to apply any retention policy and retain data forever, set retention (days) to 0.

Storage account
Showing all storage accounts including classic storage accounts

Charged
You'll be charged normal data rates for storage and transactions when you send diagnostics to a storage account.

Figure 7.5: Adding the log details

How it works...

Diagnostics has two purposes—auditing and troubleshooting. Based on traffic and settings, these logs can grow over time, so it's important to consider the main purpose of enabling diagnostics in the first place. If diagnostics are enabled for auditing, you will probably want to choose a maximum of 365 days of retention. If the main purpose is troubleshooting, the retention period can be kept at 7 days or an even shorter period of time. Setting the retention policy to **0** will store logs without removing them after a period of time. This can generate additional costs and you may need to set up a different procedure for removing logs.

If we don't want to store diagnostic logs in a storage account, we can choose Log Analytics or Event Hubs. The process, in this case, does not include setting retention periods as these settings are kept on the destination side.

Configuring Azure Firewall in forced tunneling mode

Forced tunneling allows us to force all internet-bound traffic to an on-premises firewall for inspection or audit. Because of different Azure dependencies, this is not enabled by default and requires User Defined Routes (USRs) to allow forced tunneling. This is also not possible by using **AzureFirewallSubnet**, and we need to add an additional subnet named **AzureFirewallManagementSubnet**. Note that this needs to be done prior to Azure Firewall deployment and will not work if the subnet is added afterward.

Getting ready

Before you start, open your browser and go to the Azure portal at <https://portal.azure.com>.

How to do it...

In order to add **AzureFirewallManagementSubnet** for forced tunneling, we need to do the following:

1. In the Azure portal, select **Create a resource** and choose **Route Table** under **Networking** services (or search for **Route Table** in the search bar).

- In the new pane, provide information for the **Subscription**, **Resource group**, **Region**, and **Name** fields for the route table. Make sure to select **No** for **Propagate gateway routes**:

Create Route table

Basics Tags Review + create

Project details

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

Subscription * ⓘ

Resource group * ⓘ [Create new](#)

Instance details

Region * ⓘ

Name * ⓘ

Propagate gateway routes * ⓘ ☐ Yes ☒ No

Figure 7.6: Creating a route table using the Azure portal

- Once the route table is created, we need to set a default internet route. Go to the route table we just created, and under **Routes** in the **Settings** section, select **Add**:

RouteTable1 | Routes ⓘ

Route table

<< [+ Add](#)

[Overview](#)

[Activity log](#)

[Access control \(IAM\)](#)

[Tags](#)

[Diagnose and solve problems](#)

Settings

[Configuration](#)

[Routes](#)

Name

No results.

Figure 7.7: Adding a default internet route for the route table

4. In the new pane, we need to provide a name for the route. We should also put `0.0.0.0/0` under **Address prefix** and **Internet** under **Next hop type**:

Add route

RouteTable1

Route name *

Internet

Address prefix * ⓘ

0.0.0.0/0

Next hop type ⓘ

Internet

Next hop address ⓘ

Figure 7.8: Configuring the default internet route for the route table

5. Now go to the virtual network where you plan to deploy Azure Firewall. Under **Subnets**, add a new subnet. Note that **AzureFirewallSubnet** still needs to be added as well:

<> Packt-Portal | Subnets

Virtual network

Search (Ctrl+/)

«

+ Subnet

+ Gateway subnet

Refresh

Manage users

Delete

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Address space

Connected devices

Subnets

Search subnets

Name ↑↓

IPv4 ↑↓

FrontEnd

10.10.0.0/25 (123 available)

BackEnd

10.10.1.0/24 (251 available)

GatewaySubnet

10.10.2.0/24 (251 available)

AzureFirewallSubnet

10.10.3.0/24 (251 available)

Figure 7.9: Adding a new subnet in the virtual network pane

6. In the new pane, set the name to **AzureFirewallManagementSubnet**, provide a value for the **Subnet address range** field (a minimum subnet size of /26 is required), and select the route table we created in the **Route table** field:

Add subnet ✕

Name *

AzureFirewallManagementSubnet ✓

Subnet address range * ⓘ

10.10.4.0/24

10.10.4.0 - 10.10.4.255 (251 + 5 Azure reserved addresses)

☐ Add IPv6 address space ⓘ

NAT gateway ⓘ

None ▼

Network security group

None ▼

Route table

RouteTable1 ▼

SERVICE ENDPOINTS

Create service endpoint policies to allow traffic to specific azure resources from your virtual network over service endpoints. [Learn more](#)

Services ⓘ

0 selected ▼

SUBNET DELEGATION

Delegate subnet to a service ⓘ

None ▼

Figure 7.10: Configuring the subnet settings in the new pane

7. Now we can proceed with Azure Firewall deployment. See the *Creating a new firewall* recipe.

How it works...

In order to support forced tunneling, traffic associated with service management is separated from the rest of the traffic. An additional subnet is required with a minimum size of /26, along with an associated public IP address. A route table is required with a single route defining the route to the internet, and **BGP route propagation (propagate gateway routes)** must be disabled. We can now include routes and define where exactly traffic needs to go (a virtual network appliance or on-premises firewall) in order to be inspected or audited before reaching the internet.

Creating an IP group

IP groups are Azure resources that help to group IP addresses for easier management. This way, we can apply Azure Firewall rules easier and with better visibility.

Getting ready

Before you start, open your browser and go to the Azure portal at <https://portal.azure.com>.

How to do it...

In order to create a new IP group, we need to do the following:

1. In the Azure portal, select **Create a resource** and choose **IP Group** under **Networking** services (or search for **IP group** in the search bar).
2. In the new pane, provide information for **Subscription**, **Resource group**, **Name**, and **Region**:

Create an IP Group

Basics IP addresses Tags Review + create

An IP group is a user-defined collection of static IP addresses, ranges, and subnets. It can be used with Azure Firewall for network, application, and network address translation (NAT) rules.

Project details

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

Subscription *

Resource group *

[Create new](#)

IP Group details

Name *

Region *


 IP Groups are global and can be used across regions regardless where they are stored.

Figure 7.11: Creating a new IP group using the Azure portal

- Under **IP addresses**, we need to provide something for the **IP address, range or subnet** field. In this example, we are adding a subnet:

Create an IP Group

Basics **IP addresses** Tags Review + create

↑ Import from File | Delete

IP address, range or subnet ⓘ	Validation Status ⚙
10.1.0.0/24 ✓	Valid

Enter a single IP address, multiple IP addresses, or ranges...

< Previous Page 1 of 1 Next >

Figure 7.12: Adding a subnet in the IP address, range or subnet field

- We can now proceed and deploy the IP group.

How it works...

IP groups allow us to associate multiple IP addresses with a single resource for easier management. We can associate any number of individual IP addresses (in **10.10.10.10** format), IP ranges (in **10.10.10.10-10.10.10.20** format), or subnets (in **10.10.10.0/24** format). Then, firewall rules can be associated with IP groups and all IP addresses under a defined IP group. Instead of creating a separate rule for each IP address, range, or subnet, we can now have a single rule for a single IP range. This means easier management and maintenance of Azure Firewall, along with better visibility of effective rules.

Configuring Azure Firewall DNS settings

We can use a custom DNS server with our Azure Firewall instance. This allows us to resolve custom names and apply filtering based on **Fully Qualified Domain Name (FQDN)**.

Getting ready

Before you start, open your browser and go to the Azure portal at <https://portal.azure.com>.

How to do it...

In order to configure custom DNS settings in Azure Firewall, we need to do the following:

1. In the Azure Firewall pane, locate **DNS** under **Settings**. We need to set it to **Enabled**. Select the type of DNS (default or custom) and whether we want to use a DNS proxy:

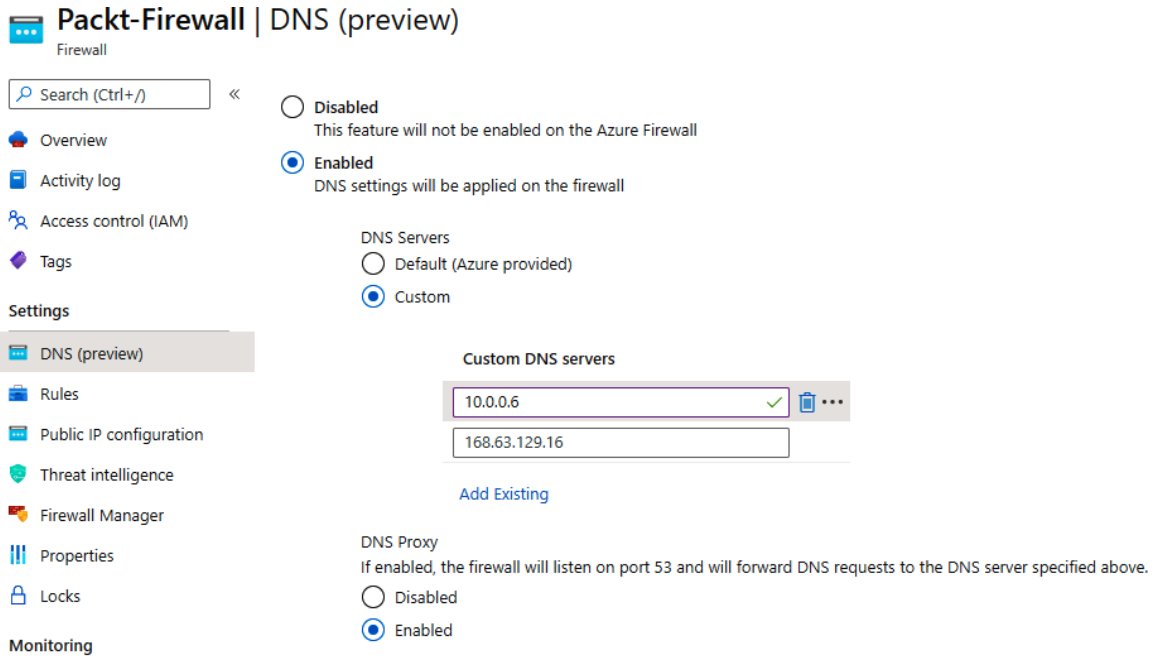


Figure 7.13: Configuring Azure Firewall DNS settings using the Azure portal

2. Once all the necessary settings are provided, select **Save** to apply them. It takes up to 30 minutes to correctly propagate routes and for them to take full effect.

How it works...

In order to use FQDN filtering, Azure Firewall needs to be able to resolve the FQDN in question. This can be achieved by enabling DNS settings on Azure Firewall. When enabled, we can choose between Azure-provided DNS or custom DNS. Custom DNS can be either an Azure DNS zone or a DNS server running on a virtual network.