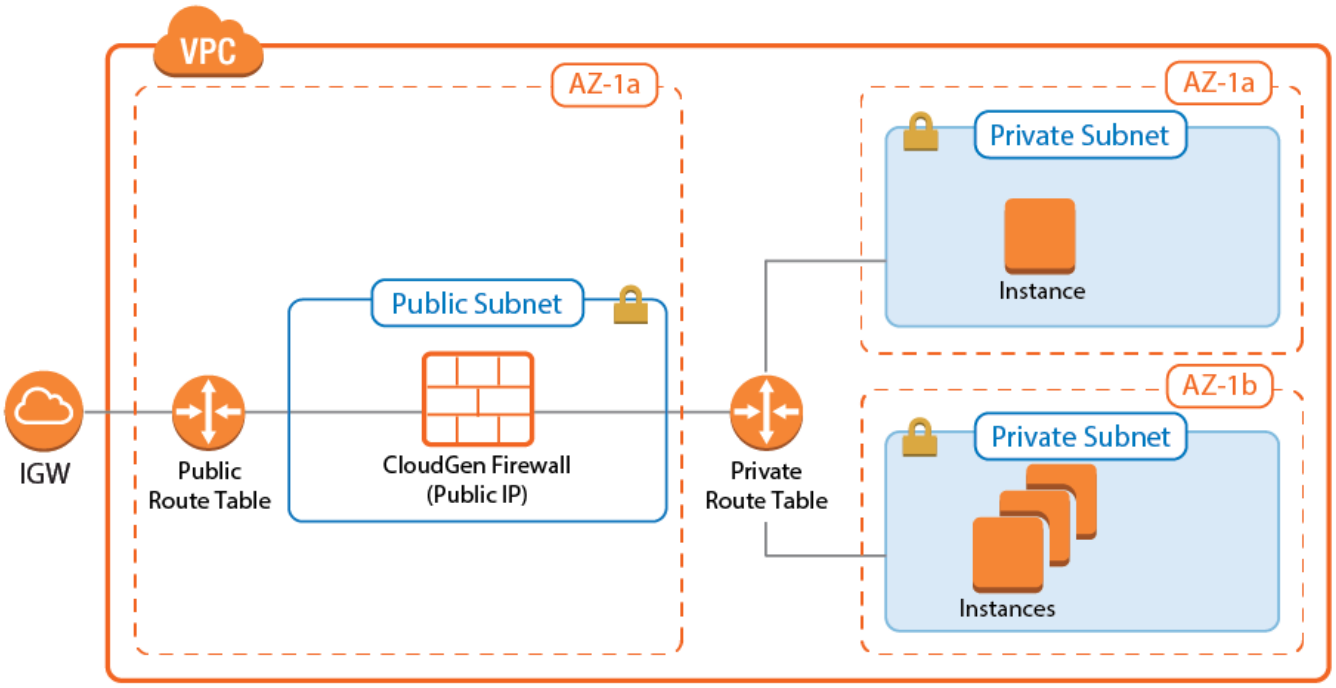
How to Deploy a CloudGen Firewall in AWS via AWS Console

https://campus.barracuda.com/doc/73719651/

The Barracuda CloudGen Firewall F secures and connects the services running in your AWS virtual private cloud (VPC). The firewall monitors and secures all traffic between subnets to and from the Internet. It also connects your cloud resources either to your on-premise networks with site-to-site VPN, or to your remote users with client-to-site VPN and SSL VPN. After the deployment the Instance ID is the root password set to log in via Barracuda Firewall Admin. Logging in via SSH is only possible through certificate file set during the last deployment step.



Step 1. Create an IAM Role for the Firewall

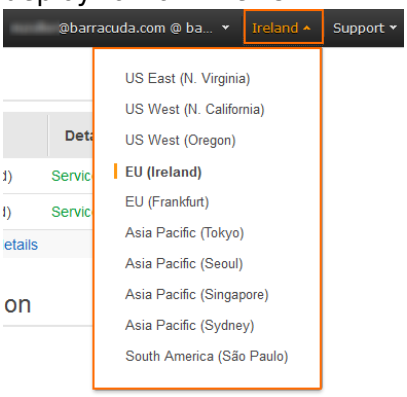
Create an IAM role for your firewall instance. Verify that the all the required IAM policies are attached to the role.

For step-by-step instructions, see How to Create an IAM Role for a CloudGen Firewall in AWS.

Step 2. Select the AWS Datacenter

1. Log into the AWS console.

2. In the upper right, click on the datacenter location, and select the datacenter you want to deploy to from the list.

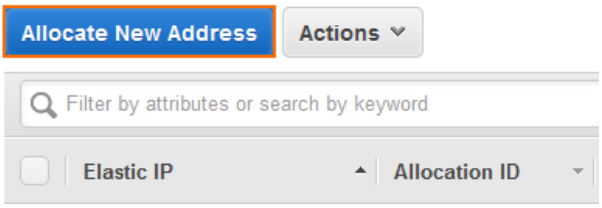


The selected datacenter location is now displayed in the AWS console.

Step 3. Create an Elastic IP

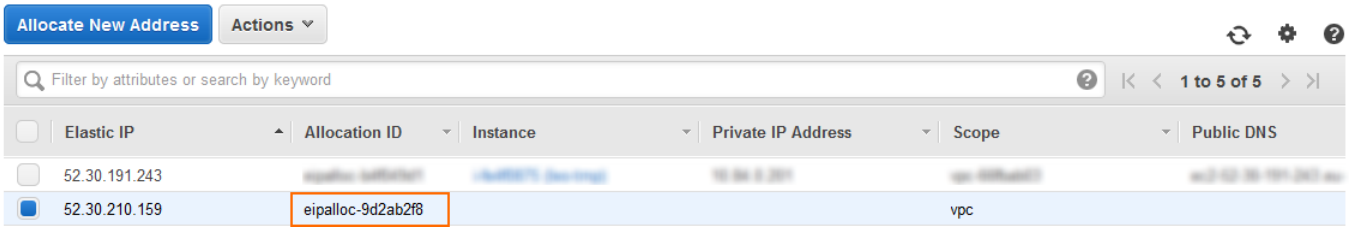
Create an elastic IP address. This is the public IP address that will be used for your firewall instance.

1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the **Network & Security** section of the left menu, click on **Elastic IPs**.
4. Click **Allocate New Address**.



5. Click **Yes, Allocate**.

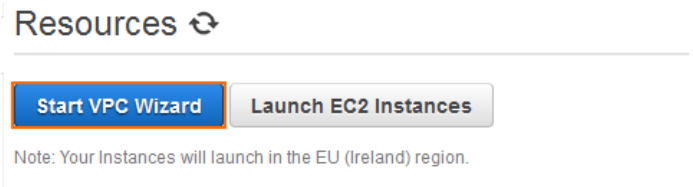
An unassigned elastic IP is now added to the list. Copy the **Allocation ID** for future use.



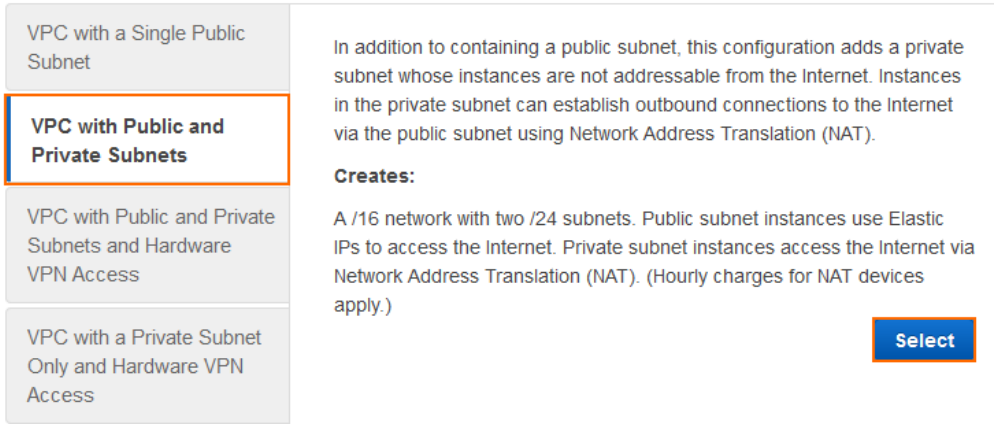
Step 4. Create VPC with VPC Wizard

Use the VPC wizard to create a VPC with one public and one private subnet. The firewall will be deployed in the public subnet. If needed, you can add additional subnets after the deployment.

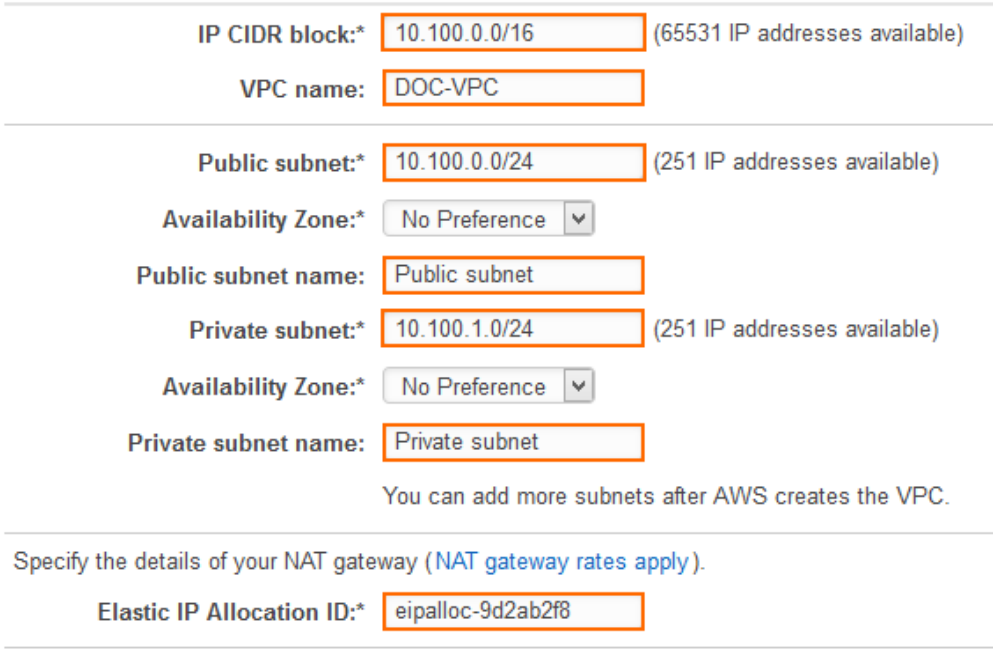
1. Log into the AWS console.
2. Click **Services** and select **VPC**.
3. Click **Start VPC Wizard**. The VPC wizard opens.



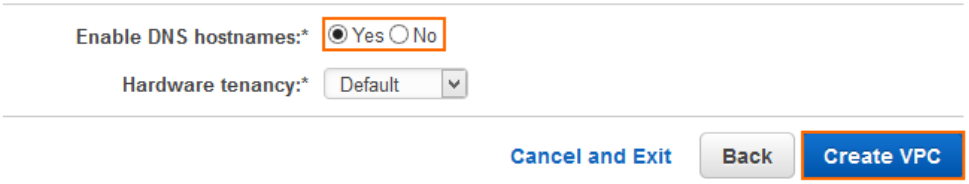
4. Select **VPC with Public and Private Subnets** and click **Select**.



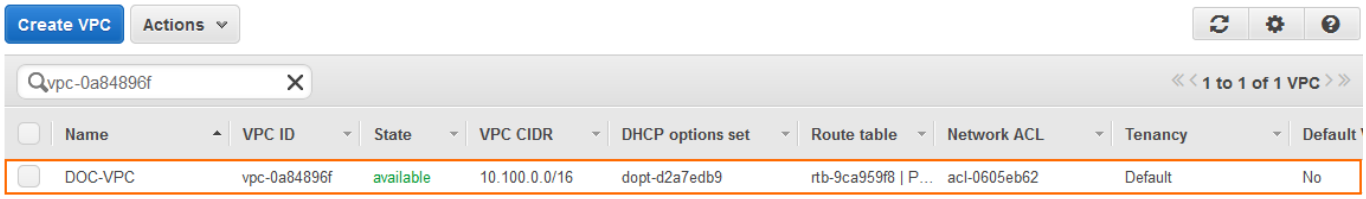
5. On the **VPC with Public and Private Subnets** change the following settings:
   - **IP CIDR block** – Enter a /16 CIDR block that does not overlap with any of your other networks.
   - **VPC Name** – Enter the name.
   - **Public subnet** – Enter the /24 subnet used for the firewall instance.
   - **Public subnet name** – Enter a name for the public subnet.
   - (optional) **Availability Zone** – Select which availability zone the VPC is created in. Select **No Preference** for AWS to assign it automatically.
   - **Private subnet** – Enter the /24 subnet used for the instances protected by the firewall.
   - **Private subnet name** – Enter a name for the private subnet.
   - **Elastic IP Allocation ID** – Enter the **Allocation ID** for the elastic IP address created in step 1.



6. (optional) Set **Enable DNS hostnames** to **NO** to only use IP addresses to access your VPC.
7. Click **Create VPC**.



The VPC is now listed in the **Your VPCs** list.



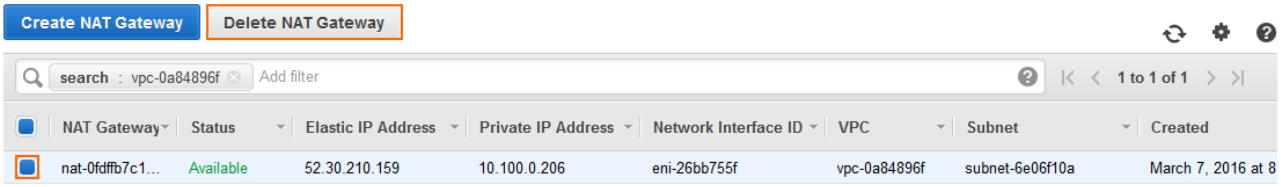
Step 5. Delete the NAT Gateway

Delete the NAT gateway.

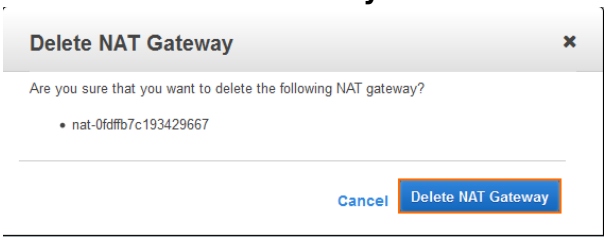
The VPC wizard automatically creates a NAT gateway instance. But since the firewall already includes this functionality, the NAT gateway instance must be deleted.

1. Log into the AWS console.
2. Click **Services** and select **VPC**.

3. In the **Virtual Private Cloud** section of the left menu, click on **NAT Gateways**.
4. (optional) Enter the VPC ID in the **search bar**.
5. Select the NAT gateway created for your VPC and click **Delete NAT Gateway**. The **Delete NAT Gateway** pop-over window opens.



6. Click **Delete NAT Gateway**.

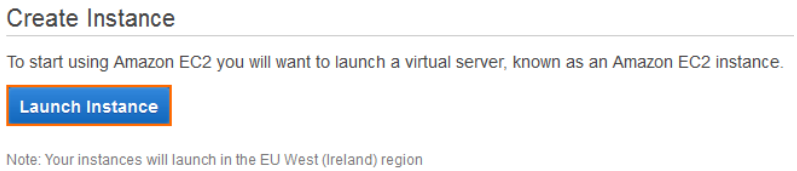


The elastic IP address associated with the NAT gateway is released automatically and is now free to use for the firewall instance.

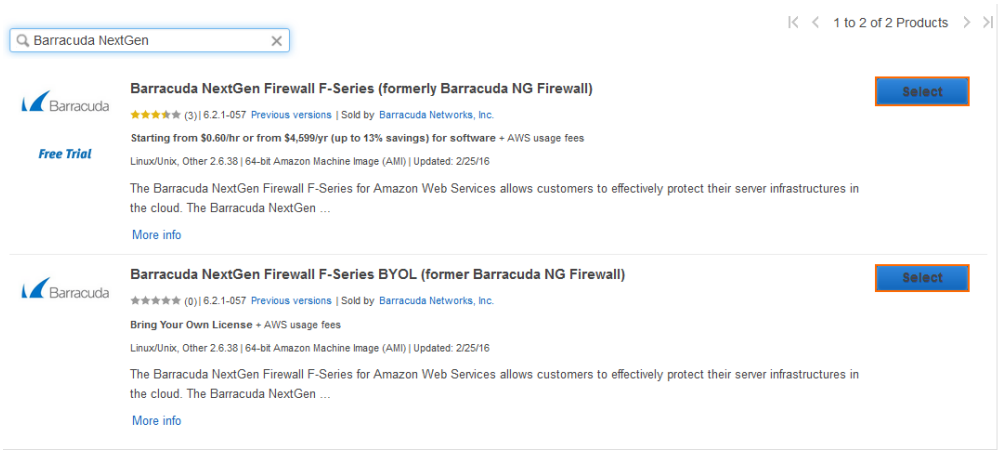
Step 6. Deploy the CloudGen Firewall F Instance

You can deploy the CloudGen Firewall F instance in two different ways from the AWS Marketplace: BYOL and hourly. The firewall instance is deployed into the public subnet and can be configured to use either a single network interface or one network interface per subnet. The number of network interfaces is limited by the instance size.

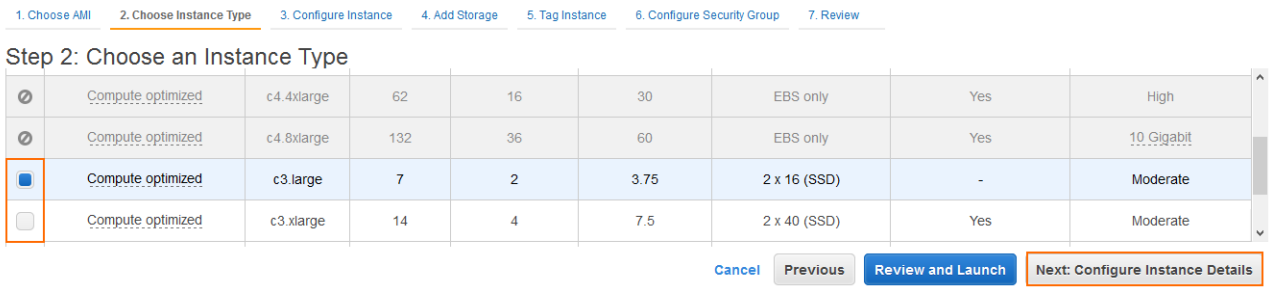
1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the **Create Instance** section, click **Launch Instance**. The **VPC wizard** starts.



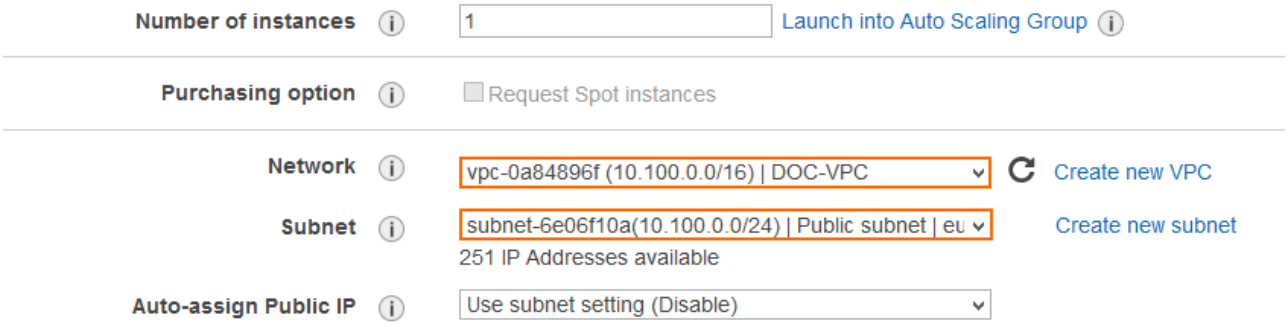
4. In the left menu, click **AWS Marketplace**.
5. Enter `Barracuda CloudGen` in the **Search for AWS Marketplace Product** search box**.**
6. Click **Select** next to the image type you want to deploy: BYOL or hourly.



7. Select the **Instance Type**. If you are deploying a BYOL image, verify that the number of CPU cores of the instance matches your license.
8. Click **Next: Configure Instance Details**.



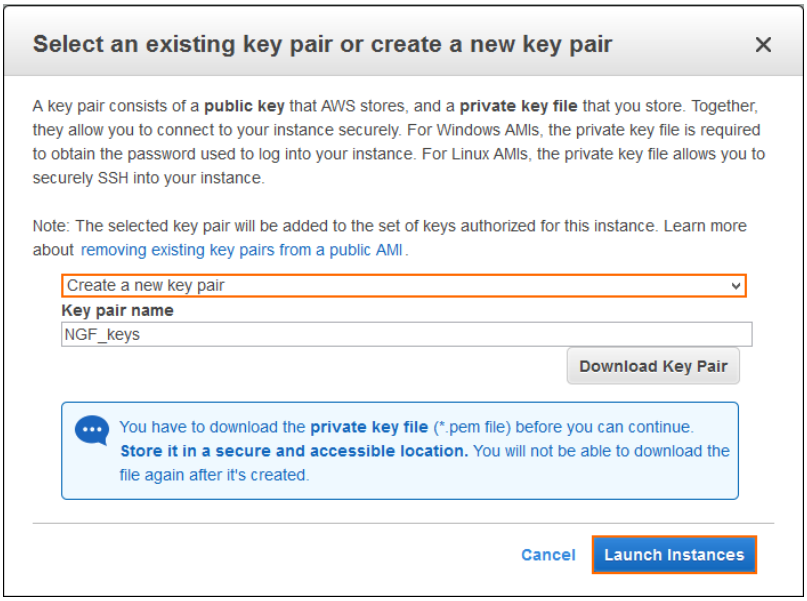
9. Configure the **Instance Details**:
   - **(HA only) Number of instances** – To deploy two instances to create an HA cluster, enter 2. For stand-alone deployments, deploy one instance.
   - **Network** – Select the VPC created in step 2.
   - **Subnet** – Select the public subnet.



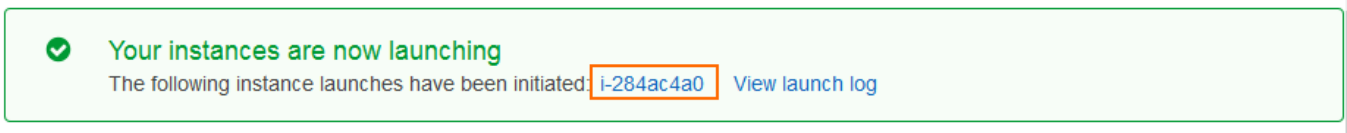
10. (optional) Add additional **Network Interfaces**:
    1. Click **Add Device**. The device is added to the list.
    2. Select the **Subnet** the network interface is connected to.
    3. (optional) Enter the **Primary IP** address for this interface. The IP address must be in the subnet selected above.
11. Click **Next:Add Storage**.
12. (optional) Change the **Volume Type** as needed.
13. Click **Next: Tag Instance**.
14. Click **Next: Configure Security Group**.
15. (optional) Click **Add Rule** and add rules for ICMP

- **Type** – Select **All ICMP**.
- **Source** – Select **Anywhere**.

16. (optional) Click **Add Rule** and add rules for HTTP
    - **Type** – Select **HTTP**.
    - **Source** – Select **Anywhere**.
17. Click **Review and Launch**.
18. Click **Launch**. The **Select and existing key pair or create a new key pair** pop-over window opens.
19. From the drop-down list, select **Choose an existing key pair** or **Create a new key pair**. The certificate is valid only for SSH logins with the root user. For Barracuda Firewall Admin the Instance ID is the default password.
20. Click the checkbox to verify that you have access to the selected key or click **Download Key Pair** to download a new key pair.
21. Click **Launch Instances**.



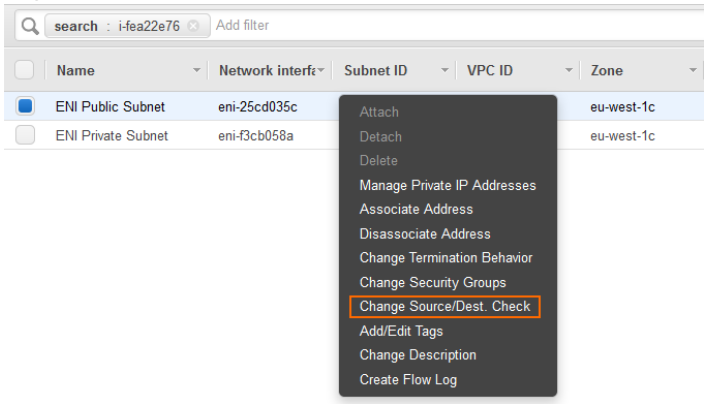
On the **Launch Status** page, locate and copy the **Instance IDs**. This is the default password used to log in via Barracuda Firewall Admin.



Step 7. Disable Source/Destination Check for the Network Interface

For the interface to be allowed to forward traffic with a destination IP address that is different from the IP addresses assigned to the network interfaces, you must disable the source/destination check.

1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the **Network & Security** section of the left menu, click on **Network Interfaces**.
4. (optional) Filter the list using the Instance ID.
5. Right-click on the network interface, and select **Change Source/Dest. Check**.



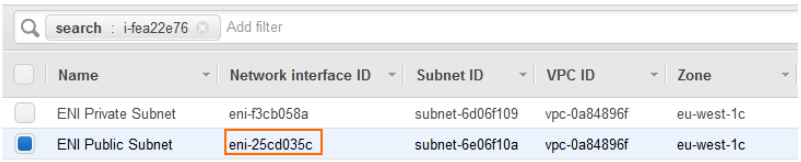
1. Set the **Source/dest. check** to **Disabled**.
2. Click **Save**.

The source/destination check is now disabled for the network interface connected to the firewall instance.

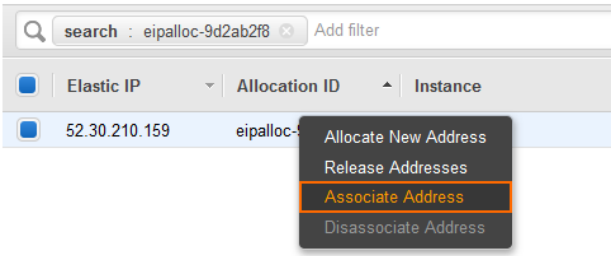
Step 8. Associate the Elastic IP with the Firewall

Use the Elastic IP (EIP) as the public IP address for the firewall network interface connected to the public subnet.

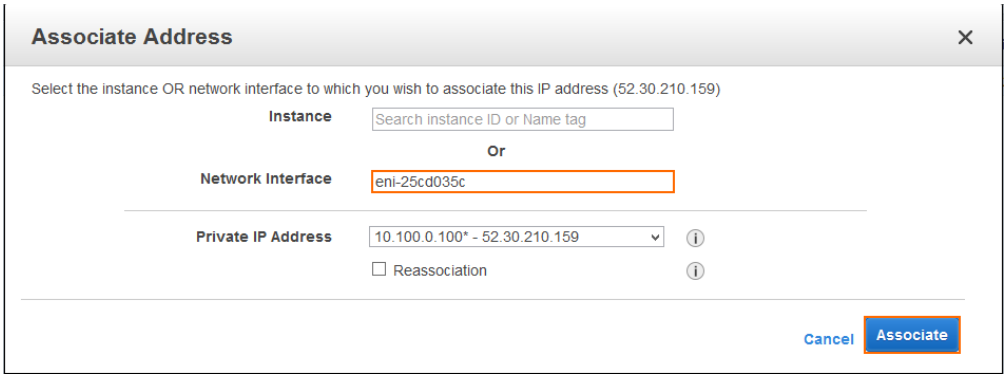
1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the **Network & Security** section of the left menu, click on **Network Interfaces**.
4. (optional) Filter the list using the Instance ID.
5. Locate the network interface connected to the public subnet, and copy the **Network interface ID**.



6. In the **Network & Security** section of the left menu, click on **Elastic IPs**.
7. Right-click the EIP created in step 2, and click **Associate Address**.



8. Enter the **Network Interface** ID, and click **Associate**.

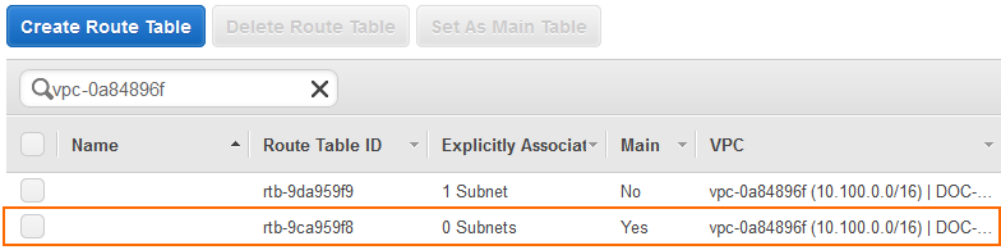


Traffic to the EIP is now automatically forwarded to the network interface attached to the public subnet of the VPC.

Step 9. Adjust the Routing Tables

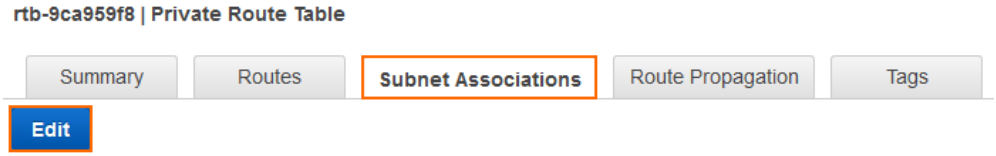
Adjust the routing table for the private subnets to use the firewall instance as the default gateway. Instances will always use the first IP address of the subnet as the default gateway. The AWS cloud fabric then internally reroutes the traffic to the configured network interface or instance.The route table attached to the public subnet does not need to be changed.

1. Log into the AWS console.
2. Click **Services** and select **VPC**
3. In the **Virtual Private Cloud** section of the left menu, click on **Route Tables**.
4. (optional) Filter the list using the VPC ID.
5. Select the route table that is not associated with the public subnet.



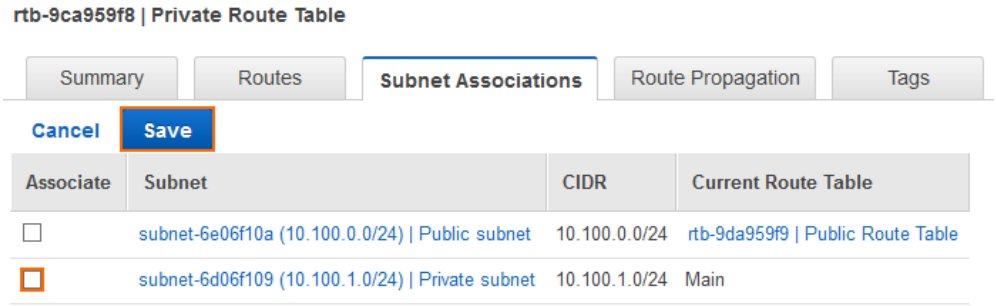
6. In the lower half of the page, click on the **Subnet Associations** tab.

7. Click **Edit**.

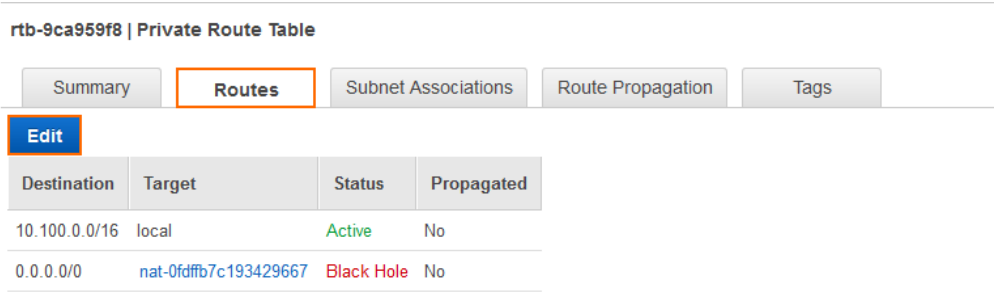


8. Select the private subnet and click **Save**.

   If you are deploying with multiple network interfaces, you must create a route table for each private network. If you are using one network interface, associate all private subnets with this route table.

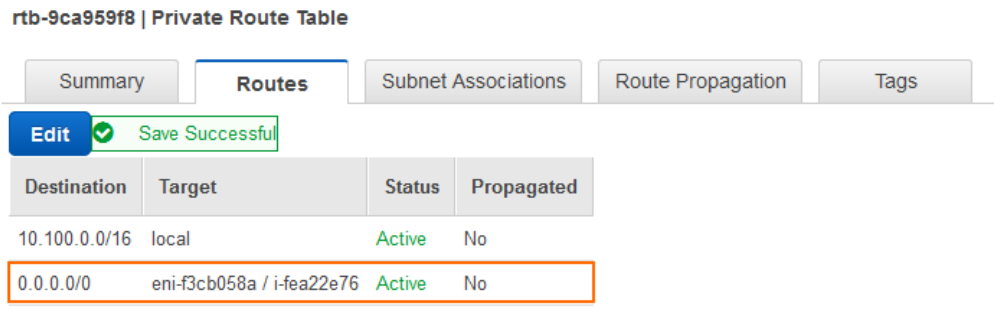


9. Click on the **Routes** tab.
10. Click **Edit**.



11. Depending on whether you are using single or multiple network interfaces:
    1. **Single NIC** – Enter the Instance ID of the firewall in the **Target** column of the route with the **Destination** 0.0.0.0/0.
    2. **Multiple NICs** – Enter the network interface ID of the network interface associated with this subnet in the **Target** column of the route with the **Destination** 0.0.0.0/0.
12. Click **Save**:

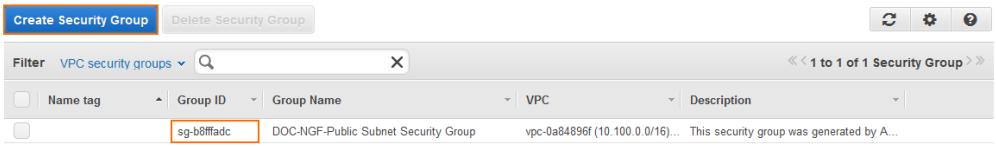
You now have a default route with the **Status** active and the target set to the correct firewall network interface.



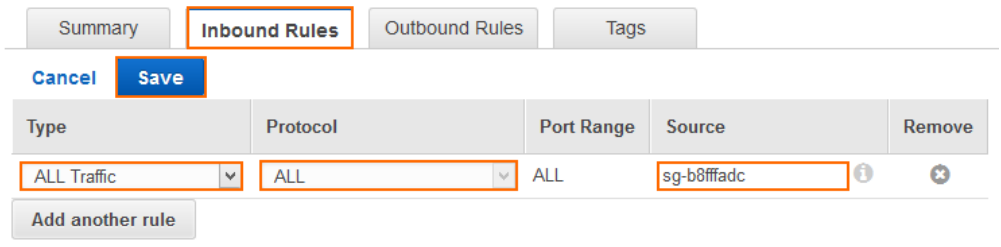
Step 10. Create a Security Group

Create a security group for the private networks that allow all traffic from the security group assigned to the firewall.

1. Log into the AWS console.
2. Click **Services** and select **VPC**
3. In the **Security** section of the left menu, click on **Security Groups**.
4. Locate the security group created during the firewall deployment, and copy the **Group ID**.



5. Click **Create Security Group**.
   - **Group name** – Enter a name for the security group.
   - **Description** – Enter a description for the security group.
   - **VPC** – Select the VPC you created in step 3 from the list.
6. Click **Yes, Create**.
7. In the lower half of the page, click on the **Inbound Rules** tab.
8. Click **Edit**.
9. Create a rule to allow traffic from the firewall security group:
   - **Type** – Select **All Traffic**.
   - **Protocol** – Select **ALL**.
   - **Source** – Enter the group ID of the security group assigned to your firewall.
10. Click **Save**.



When deploying Instances to one of the private subnets, use this security group. This will allow traffic

to and from the firewall.

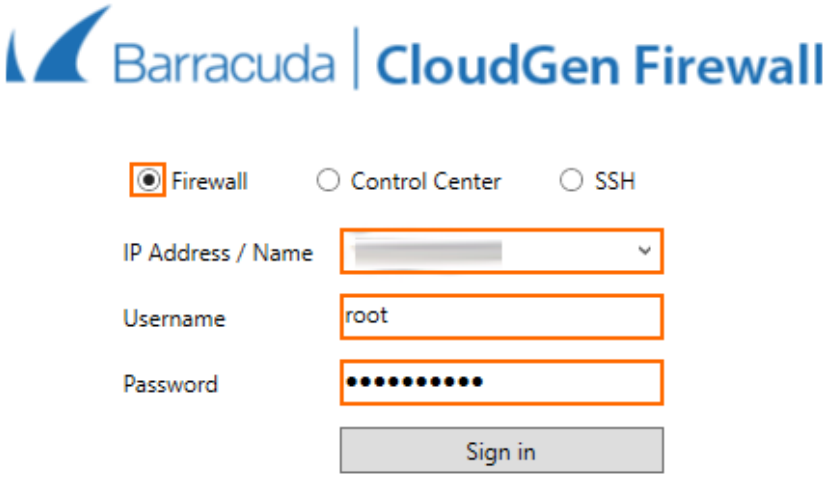
Step 11. (optional) Edit the Network ACLs

The Network ACLs created by the VPC wizard are configured by default to allow traffic through. If required, go **Network ACLs** to edit the network ACL assigned to your VPC.

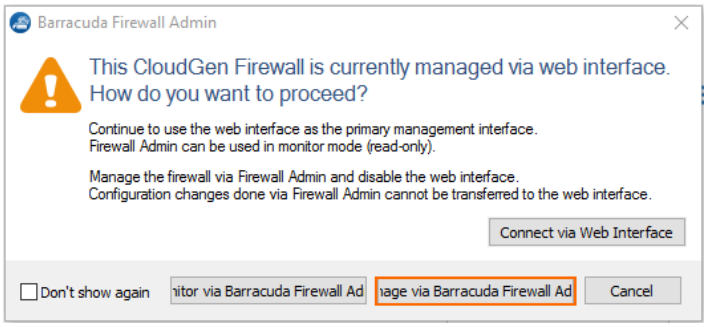
Step 12. Log in via Barracuda Firewall Admin

Use Barracuda Firewall Admin to log into your firewall.

1. Launch Barracuda Firewall Admin.
2. Log into the firewall:
   - Select **Firewall**.
   - **IP Address / Name** – Enter the elastic IP.
   - **Username** – Enter `root`.
   - **Password** – Enter the Instance ID of the firewall instance created in step 5.
3. Click **Sign in**.

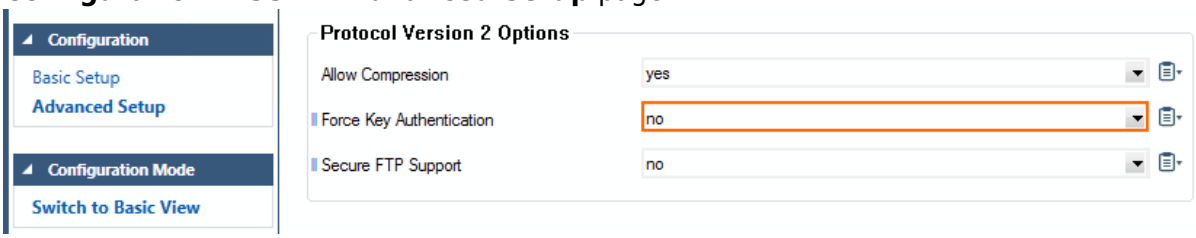


4. Renew your password.
5. The window for selecting how to manage the firewall is displayed.
6. Click **Manage via Barracuda Firewall Admin**.



Next Steps

- (BYOL only) License and activate the firewall. For more information, see How to Activate and License a Stand-alone Virtual or Public Cloud Firewall or Control Center.
- (optional) Re-enable SSH logins via password by setting **Force Key Authentication** to **No** in the **Advanced View** of the **CONFIGURATION > Configuration Tree > Box > Advanced Configuration > SSH > Advanced Setup** page.



Figures

1. aws_vpc_single.png
2. aws_deploy_00.png
3. aws_deploy_01.png
4. aws_deploy_02.png
5. aws_deploy_03.png
6. aws_deploy_04.png
7. aws_deploy_05.png
8. aws_deploy_06.png
9. aws_deploy_07.png
10. aws_deploy_08.png
11. aws_deploy_09.png
12. aws_deploy_10.png
13. aws_deploy_11.png
14. aws_deploy_12.png
15. aws_deploy_13.png
16. aws_deploy_15.png
17. aws_deploy_16.png
18. aws_deploy_17.png
19. aws_deploy_19.png
20. aws_deploy_20.png
21. aws_deploy_21.png
22. aws_deploy_22.png
23. aws_deploy_23.png
24. aws_deploy_24.png
25. aws_deploy_25.png
26. aws_deploy_26.png
27. aws_deploy_27.png
28. aws_deploy_28.png
29. aws_deploy_29.png
30. aws_manage_via_ngadmin.png
31. aws_deploy_30.png