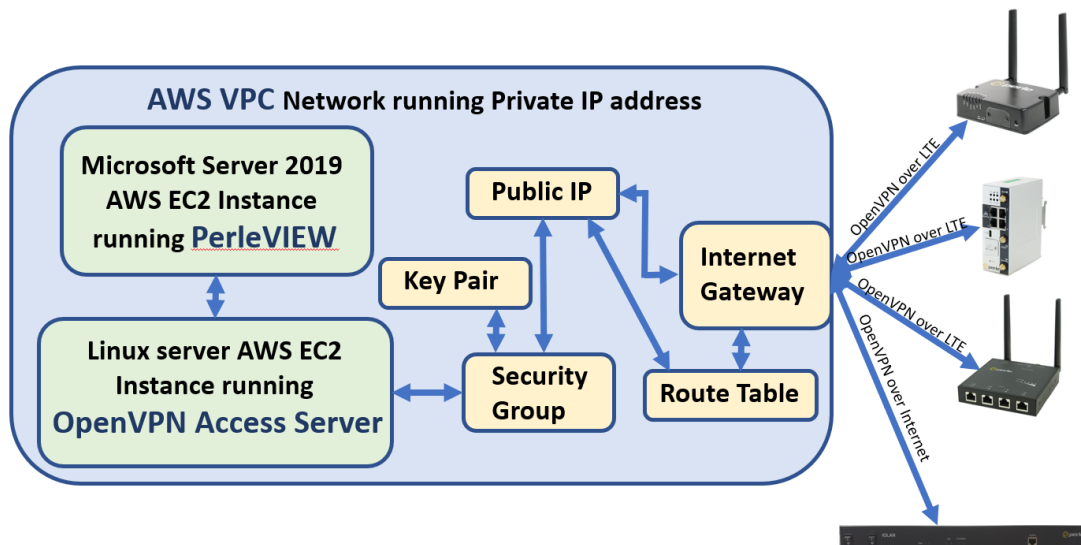


Deploy and Manage your Network from the Cloud



Open an AWS account.....	2
Create an IAM User (optional)	2
Create a Key Pair	3
Create an AWS Private Cloud.....	3
• Virtual Private Cloud (VPC)	3
• Subnet	4
• Route Table.....	5
• Internet Gateway	6
• Create a Security Group	7
Install PerleVIEW on AWS EC2 Instance.....	8
• Launch a Microsoft Server 2019 AWS EC2 Instance	8
• Install PerleVIEW on AWS EC2 Instance	10
Deploy OpenVPN Server on AWS VPC.....	10
• Add a Public Static IP address	11
IRG5000 Router Configuration for AWS OpenVPN Access Server.....	13
IOLAN SCR Configuration for AWS OpenVPN Access Server	14

Amazon offers a walkthrough of the necessary steps here:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html>

Open an AWS account

When you sign up for Amazon Web Services (AWS), you get access to all services in AWS but are only charged for the services that you use. You can monitor all the cost associated with your AWS project through “My Billing Dashboard” from AWS. To open an AWS account, go to:

<https://portal.aws.amazon.com/billing/signup>.

Create an IAM User (optional)

For added security, Amazon recommends the use of an AWS Identity and Access Management (IAM) user account. Create an IAM user, and then add the user to an IAM group with administrative permissions or grant this user administrative permissions. For more details, see [Working with the AWS Management Console](#).

To create an administrator user for yourself and add the user to an administrator group (console):

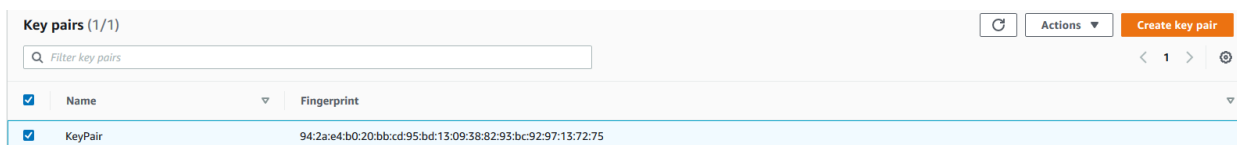
1. Use your AWS account email address and password to sign in as the [AWS account root user](#) to the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, click **Users** and then click **Add user**.
3. For **Username**, enter **Administrator**.
4. Click the check box next to AWS Management Console access. Select **Custom password**, and then enter a new password.
5. Choose **Next: Permissions**.
6. Under **Set permissions**, click **Add user to group**.
7. Click **Create group**.
8. In the **Create group** dialog box, for **Group name** enter **Administrators**.
9. Click **Filter policies**, and then select **AWS managed -job function** to filter the table contents.
10. In the policy list, select the check box for **Administrator Access**. Then click **Create group**.
11. In the list of groups, select the check box for your new group. If necessary, click **Refresh** to see the group in the list.
12. Click **Next: Tags**. If desired, add metadata to the user by attaching tags as key-value pairs.
13. Click **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, click **Create user**.
14. To sign in as this new IAM user, sign out of the AWS console, then use the following URL, where *your_aws_account_id* is your AWS account number without the hyphens (for example, if your AWS account number is **4823-9462-5624**, your AWS account ID is **482394625624**): https://your_aws_account_id.signin.aws.amazon.com/console/
15. Enter the IAM username and password that you just created. When you're signed in, the navigation bar displays "*your_user_name @ your_aws_account_id*".

Create a Key Pair

AWS requires the use of a key pair to securely log in to your EC2 instances. You specify the name of the key pair when you launch your instance, then provide the private key when you log in using SSH. Generate a Key Pair using the Amazon EC2 console:

1. From the AWS dashboard, click **EC2** to open the Amazon EC2 console.
2. From the navigation bar, select a region for the key pair - key pairs are specific to a region.
3. In the navigation pane, under **NETWORK & SECURITY**, click **Key Pair**.
4. Click **Create Key Pair**.
5. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then click **Create**.
6. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is **.pem**. Save the private key file in a safe place as AWS for security purposes will not generate the file again.

Sample **Key Pair** configuration screen:



Key pairs (1/1)			Actions	Create key pair
Filter key pairs			< 1 > ⌕	
<input checked="" type="checkbox"/>	Name	Fingerprint		
<input checked="" type="checkbox"/>	KeyPair	94:2a:e4:b0:20:bb:cd:95:bd:13:09:38:82:93:bc:92:97:13:72:75		

Create an AWS Private Cloud

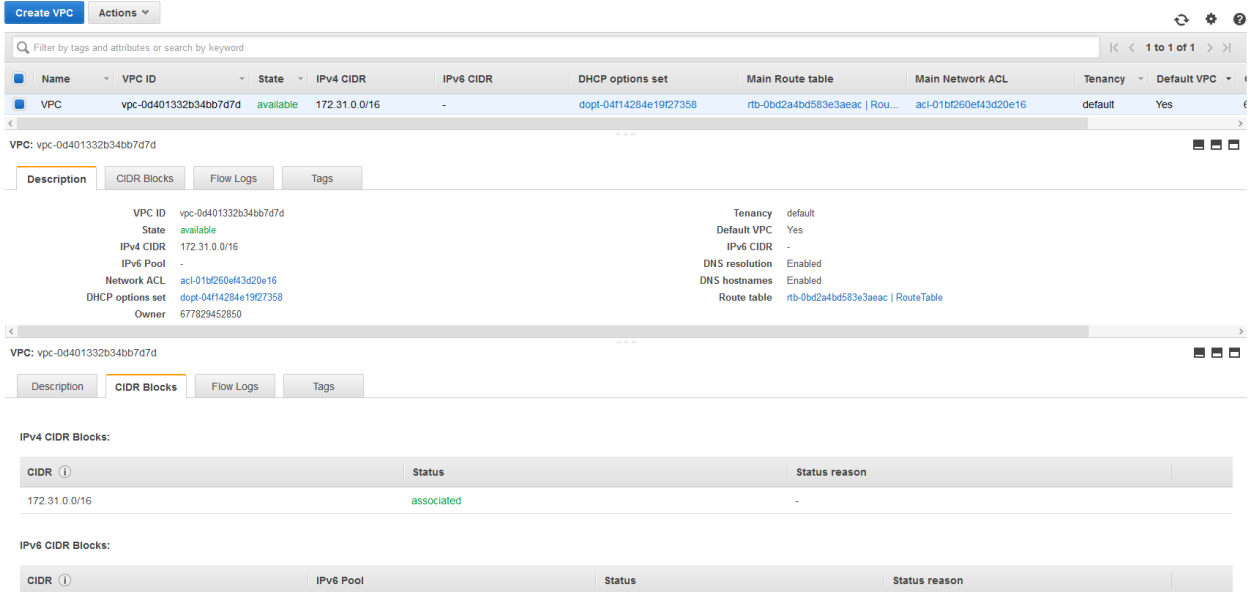
Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS. Amazon VPC is the networking layer for Amazon EC2 instances on which your applications will be running.

The following are the VPC key configuration areas that need to be enabled for your EC2 instances:

- **Virtual Private Cloud (VPC)** is the virtual network dedicated to your AWS account. EC2 instance types require that you launch your instances in a VPC. To create a non-default VPC:
 1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
 2. From the navigation bar, select a region for the VPC. Select the same region in which you created your key pair.
 3. On the VPC dashboard, choose **Launch VPC Wizard**.
 4. On **Step 1: Select a VPC Configuration** page, make sure **VPC with a Single Public Subnet** is selected, and click **Select**.

- On **Step 2: VPC with a Single Public Subnet** page, enter a name for your VPC in the **VPC name**. Leave the other default configuration settings and click **Create VPC**. On the confirmation page, click **OK**.

Sample **VPC** configuration screen:



- Subnet** is the range of the IP addresses in your VPC – configure as appropriate:
 - In the navigation pane, choose **Subnets, Create subnet**.
 - Specify the subnet details as necessary and choose **Create**.
 - Name tag**: Optionally provide a name for your subnet. Doing so creates a tag with a key of Name and the value that you specify.
 - VPC**: Choose the VPC for which you're creating the subnet.
 - Availability Zone**: Optionally choose a Zone in which your subnet will reside, or leave the default **No Preference** to let AWS choose an Availability Zone for you.
 - For information about the Regions and Zones, see [Regions and zones](#) in the *Amazon EC2 User Guide for Linux Instances*.
 - IPv4 CIDR block**: Specify an IPv4 CIDR block for your subnet, for example, 10.0.1.0/24. For more information, see [VPC and subnet sizing for IPv4](#).
 - IPv6 CIDR block** (Optional): If you've associated an IPv6 CIDR block with your VPC, choose **Specify a custom IPv6 CIDR**. Specify the hexadecimal pair value for the subnet or leave the default value.

Sample Subnet configuration screen:

Sample Subnet configuration screen showing the AWS Management Console interface for a Subnet.

Subnet Details:

- Subnet ID: subnet-0149fe080965d6417
- VPC: vpc-0d401332b34bb7d7d | VPC
- Available IPv4 Addresses: 4090
- Availability Zone: us-east-2a (use2-az1)
- Network ACL: acl-01b260ef43d20e16
- Auto-assign public IPv4 address: Yes
- Outpost ID: -
- State: available
- IPv4 CIDR: 172.31.0.0/20
- IPv6 CIDR: -
- Route Table: rtb-0bd2a4bd583e3aeac | RouteTable
- Default subnet: Yes
- Auto-assign IPv6 address: No
- Owner: 677829452850

Route Table Association:

Route Table: rtb-0bd2a4bd583e3aeac | RouteTable

Destination	Target
162.242.174.125/32	vgw-0ed85a11fa2880624
23.253.56.54/32	vgw-0ed85a11fa2880624
172.31.0.0/16	local
0.0.0.0/0	lgw-0e55ef6253a7f1591

Network ACL Association:

Network ACL: acl-01b260ef43d20e16

Inbound rules:

Rule #	Type	Protocol	Port Range / ICMP Type	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Outbound rules:

Rule #	Type	Protocol	Port Range / ICMP Type	Destination	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

- Route Table** contains a set of rules, called routes, that are used to determine where your network traffic is directed. For the purpose of a PerleVIEW deployment the main scope for the AWS Route Table is to generate a single routing domain that has access to the Internet. For more details on how the AWS Route Table works see: https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Route_Tables.html
 - In the navigation pane, choose **Route Tables**
 - Specify the route table details as necessary Make sure that the Route Table is associated with your desired VPC and Subnet as well as it is using the specified subnet IP addresses.

Sample Route Table configuration screen:

Sample Route Table configuration screen showing the configuration of a Route Table (rtb-0bd2a4bd583e3aeac) associated with Subnet (subnet-0149fe080965d6417).

The interface includes tabs for Summary, Routes, Subnet Associations, Edge Associations, Route Propagation, and Tags.

Routes Tab: Shows a list of routes with columns: Destination, Target, Status, and Propagated.

Destination	Target	Status	Propagated
172.31.0.0/16	local	active	No
0.0.0.0/0	igw-0e55ef6253a7f1591	active	No
23.253.56.54/32	vgw-0ed85a11fa2880824	active	No
162.242.174.125/32	vgw-0ed85a11fa2880824	active	No

Subnet Associations Tab: Shows a list of subnets with columns: Subnet ID, IPv4 CIDR, and IPv6 CIDR.

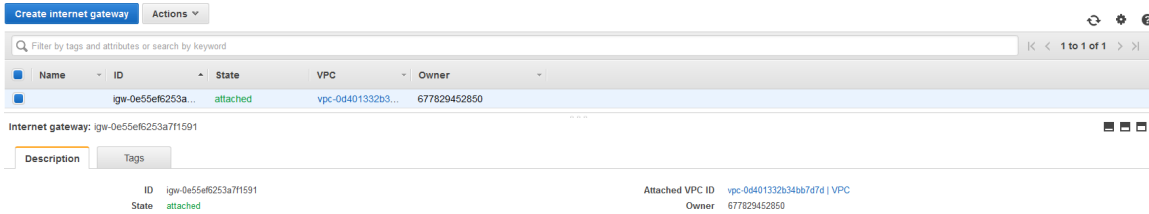
Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-0149fe080965d6417	172.31.0.0/20	-

Route Propagation Tab: Shows a list of Virtual Private Gateways with columns: Virtual Private Gateway and Propagate.

Virtual Private Gateway	Propagate
vgw-0ed85a11fa2880824 VPG	No

- Internet Gateway** is a gateway that you attach to your VPC to enable communication between resources in your VPC and the internet. It is a horizontally scaled, redundant, and highly available VPC gateway component that allows communication between resources (i.e. EC2 instances) in your VPC and the internet. The Internet Gateway will also be performing the network address translation (NAT) for instances that have been assigned public IPv4 addresses. More details can be found at : https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html
 - In the navigation pane, choose **Internet Gateway**
 - Attach an internet gateway to your **VPC**.
 - Add a route to your subnet's route table that directs internet-bound traffic to the internet gateway. If a subnet is associated with a route table that has a route to an internet gateway, it's known as a *public subnet*. If a subnet is associated with a route table that does not have a route to an internet gateway, it's known as a *private subnet*.
 - Ensure that instances in your subnet have a globally unique IP address (public IPv4 address, Elastic IP address, or IPv6 address).
 - Ensure that your network access control lists and security group rules allow the relevant traffic to flow to and from your instance.

Sample Internet Gateway configuration screen:



Internet gateway: igw-0e55ef6253a711591

Description

ID igw-0e55ef6253a711591
State attached

Attached VPC ID vpc-0d401332b34b67d7d | VPC
Owner 677829452850

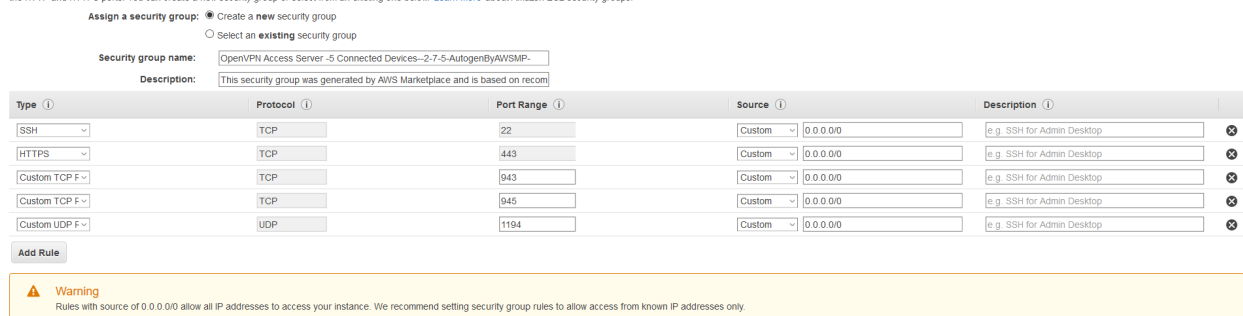
- **Security Group** acts as a virtual firewall. A security group must be created with rules that enable you to connect to your instance from your IP address using SSH. You'll need the public IPv4 address of your local computer. If you are connecting through an Internet service provider (ISP) or from behind a firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a region for the security group. Select the same region in which you created your key pair.
3. Click **Security Groups** in the navigation pane.
4. Click **Create Security Group**.
5. Enter a name for the new security group and a description.
6. In the **VPC** list, select your VPC. If you have a default VPC it is marked with an asterisk (*).
7. On the **Inbound** tab, create the following rules (choose **Add Rule** for each new rule), and then click **Create**:
 1. Choose **HTTP** from the **Type** list, and make sure that **Source** is set to **Anywhere (0.0.0.0/0)**.
 2. Choose **HTTPS** from the **Type** list, and make sure that **Source** is set to **Anywhere (0.0.0.0/0)**.
 3. Choose **SSH** from the **Type** In the **Source** box, choose **My IP** to automatically populate the field with the public IPv4 address of your local computer. Or choose **Custom** and specify the public IPv4 address of your computer or network in CIDR notation.

Sample Security Group configuration screen:

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.



Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name: OpenVPN Access Server -5 Connected Devices-2-7-5-AutoGenByAWSMP-

Description: This security group was generated by AWS Marketplace and is based on recom

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	943	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	945	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom UDP F	UDP	1194	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

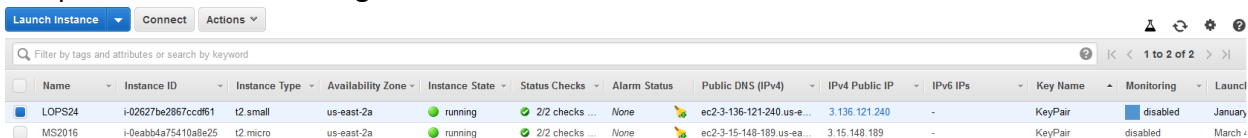
Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Install PerleVIEW on AWS EC2 Instance

- **Launch a Microsoft Server 2019 AWS EC2 Instance** using the AWS Management Console:

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console dashboard, click **Launch Instance**.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations that serve as templates for your instance. This is where you choose the hardware type & size as well as the Operating System including some additions like an SQL server. The minimum AMI configuration you will need is: **Microsoft Windows Server 2019 Base**. Please note that PerleVIEW is coming default with SQL Light, but you can choose to use your own database.
4. On the **Choose an Instance Type** page, click the hardware configuration of your instance and choose at least a small or **medium** (recommended) **sized image** or larger.
5. Click **Review and Launch** to let the wizard complete the other configuration settings for you.
6. On the **Review Instance Launch** page, under **Security Groups**, the wizard created and selected a security group for you. You can use this security group, or you can select the security group that you created when getting set up using the following steps:
 1. Choose **Edit security groups**.
 2. On the **Configure Security Group** page, ensure that **Select an existing security group** is selected.
 3. Select your security group from the list of existing security groups, and then click **Review and Launch**.
7. On the **Review Instance Launch** page, choose **Launch**.
8. When prompted for a key pair, select **Choose an existing key pair**, then select the key pair you created.
9. Click the acknowledgement check box, and then choose **Launch Instances**.
10. Click **View Instances** to close the confirmation page and return to the console.
11. It can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks in the **Status Checks**

Sample EC2 Instances configuration screen:



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launch
LOPS24	i-02627be2967ccdf61	t2.small	us-east-2a	running	2/2 checks ...	None	ec2-3-136-121-240 us-e...	3.136.121.240	-	KeyPair	disabled	January
MS2016	i-0eabb475410a8e25	t2.micro	us-east-2a	running	2/2 checks ...	None	ec2-3-15-148-189 us-ea...	3.15.148.189	-	KeyPair	disabled	March

12. Use RDS (Remote Desktop Connection) to remotely access this instance using your Windows Administrator password. A default password was created by AWS when the instance was launched and is available encrypted in the system log. To get the password

select the EC2 instance and then on the “Action” click on “Retrieve Default Windows Administrator Password”. Follow the instruction on that page:

Connect to your instance > Get Password X

Connection method ☒ A standalone RDP client (i) ☐ Session Manager (i)

The following Key Pair was associated with this instance when it was created.

Key Name KeyPair.pem

In order to retrieve your password you will need to specify the path of this Key Pair on your local machine:

Key Pair Path No file selected.

Or you can copy and paste the contents of the Key Pair below:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAje09w7T/NkfzG5gXYxgkmwLHmBodC4U1nBcJo2DR7gbveSkOJTPM7Zy9yJO
PZbFGHmB0lbimGpNnts7XSmy238QTF0uwnQg9SsuvsSWg0rF3iOg0/bSV8Wqc++2wgw6FkkzuPI4
oSzqFhCpgV+3viXwW1BcB+HxVV+5ywh+I2U5xN3U7hBKnD7FXnl3B3EZ47nSa+TlctXNZ+wruij
+XJvhbP8OrLGy2e2Cy3cSXRlvuY10QBzwcTeG1aRFGWJyxpVILoayDKPlsH+5yPhmMCilCtqrwe
KGUu4dwW06vHmoal+MCSAJYvkhH69CT/Ux1OWZx8pH1dsOXrdvOQIDAQABAEAuilDnyAX/
```

13. Download from AWS the RDS client

Connect to your instance X

Connection method ☒ A standalone RDP client (i) ☐ Session Manager (i)

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

When prompted, connect to your instance using the following details:

Public DNS ec2-3-15-148-189.us-east-2.compute.amazonaws.com

User name Administrator

Password ūwN6o-6AA5xgNeWc7t.) q=%!GvzhW%\$%Tp

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

14. Connect to your Microsoft Server 2019 AWS EC2 Instance:

Remote Desktop Connection

Remote Desktop Connection

General | Display | Local Resources | Experience | Advanced

Logon settings

Enter the name of the remote computer.

Computer: 3.15.148.189

User name: MRKT-AP19\Administrator

Saved credentials will be used to connect to this computer. You can [edit](#) or [delete](#) these credentials.

☐ Always ask for credentials

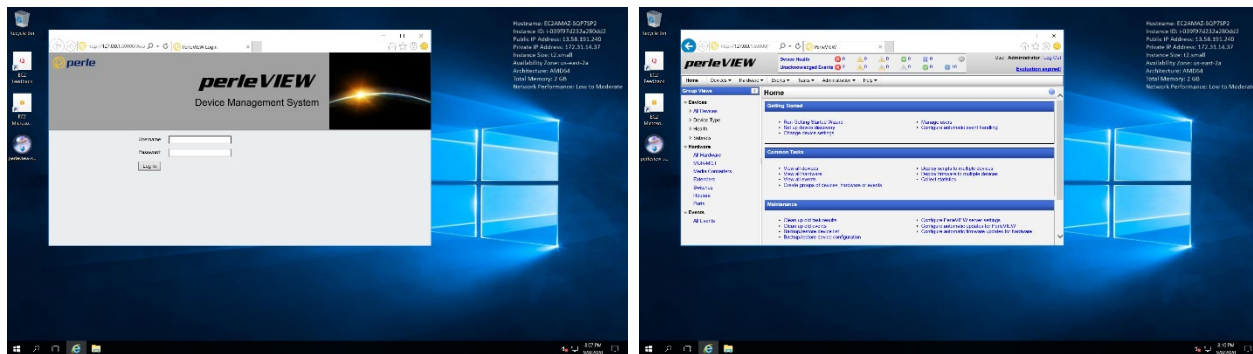
Connection settings

Save the current connection settings to an RDP file or open a saved connection.

15. Now you are ready to install PerleVIEW on your Microsoft Server 2019 AWS EC2 Instance

- **Install PerleVIEW on AWS EC2 Instance:**

1. Download PerleVIEW evaluation software from Perle web site at:
<https://www.perle.com/products/perleview-evaluation.aspx>
 - You will receive a 30-day evaluation license key free of charge
2. Download PerleVIEW manual from Perle web site at:
<https://www.perle.com/downloads/perleview.shtml> and follow the installation instructions. Note: Enable SNMP on Perle devices.
3. Congratulations, you have now installed PerleVIEW on AWS:

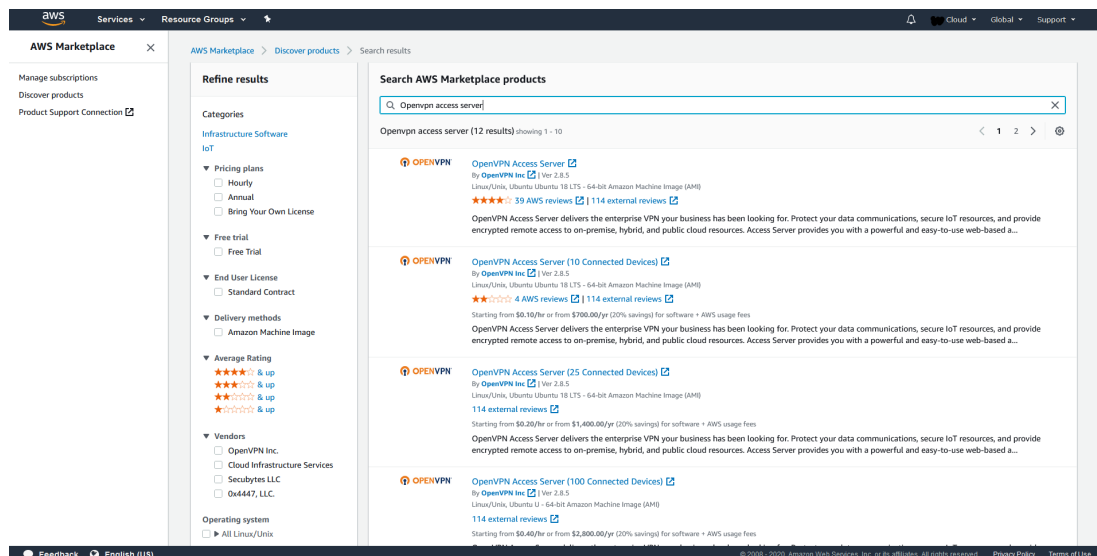


Deploy OpenVPN Server on AWS VPC

The OpenVPN server will allow Perle devices to securely connect to your AWS cloud. It will also allow you to define the VPN routing tables, such that you can easily enable / disable the data traffic between the different remote sites.

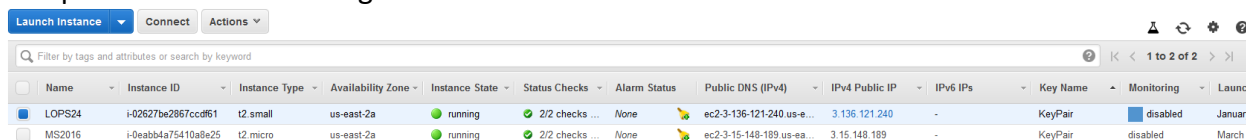
On AWS, when you choose the OpenVPN server to install, it will automatically install a Linux server as well. Go to AWS Marketplace search screen and enter **OpenVPN Access Server** sized for your needs:

<https://console.aws.amazon.com/marketplace/home#/search!mpSearch/search>



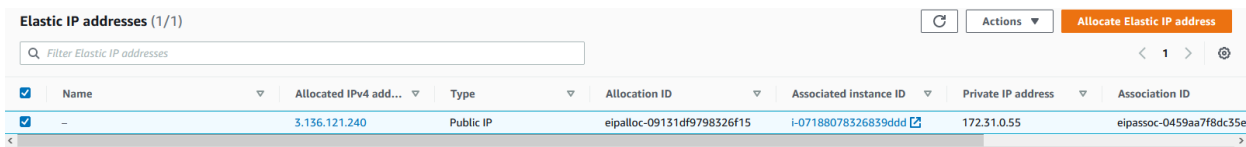
1. Select the OpenVPN Access Server size you want and follow the installation instructions
2. During the installation you will be asked if you want to **Create a new security group** or to **Select an existing security group**: choose **Select an existing security group**
3. During the installation you will be asked if you want to use a default IP address or your own. Choose the default IP address.
4. Go to your EC2 Instances configuration screen to check that your Linux server for OpenVPN Access Server is up and running.

Sample EC2 Instances configuration screen:



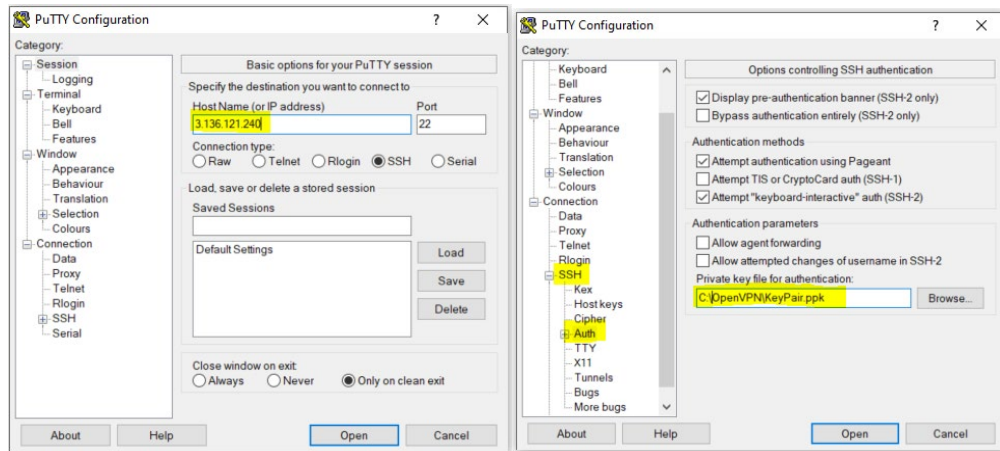
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launch
LOPS24	i-02627be2867cc0d61	t2.small	us-east-2a	running	2/2 checks ...	None	ec2-3-136-121-240 us-e...	3.136.121.240	-	KeyPair	disabled	January
MS2016	i-0eabb4a75410a8e25	t2.micro	us-east-2a	running	2/2 checks ...	None	ec2-3-15-148-189 us-ea...	3.15.148.189	-	KeyPair	disabled	March

- **Add a Public Static IP address:** AWS Elastic IP address allows you to reserve a public IP address that you can assign to any EC2 instance in a particular region, until you choose to release it. Allocate an Elastic IP address to your account:
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>
 - b. In the navigation pane, choose **Elastic IPs**.
 - c. Choose **Allocate Elastic IP address**.
 - d. Select **Actions** and then click on **Associate Elastic IP Address**
 - e. On the follow-up screen choose **Instance** as the **Resource Type**
 - f. Select your **EC2 Instance**, which should be your **Linux OpenVPN Access Server**

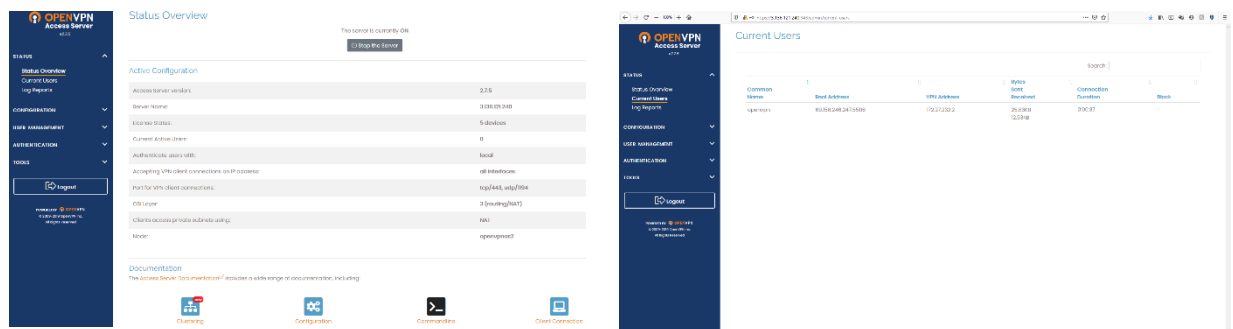


Name	Allocated IPv4 address	Type	Allocation ID	Associated instance ID	Private IP address	Association ID
-	3.136.121.240	Public IP	eipalloc-09131df9798326f15	i-07188078326839ddd	172.31.0.55	eipassoc-0459aa7f8dc35e

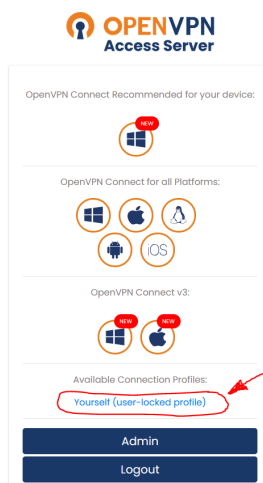
5. Once you provide a few initial configuration settings, **OpenVPN Access Server** can be configured by accessing its **Admin Web UI** using your **Web browser**.
6. To connect to your Linux EC2 Instance:
 - a. using an **SSH client** use the following command to set the permissions of your private key file so that only you can read it: `chmod 400 your_user_name-key-pair-region_name.pem`
 - b. using your **Key Pair**:
 - i. To connect to your Linux instance with a Mac or Linux computer, specify the .pem file to your SSH client with the -i option and the path to your private key.
 - ii. To connect to your Linux instance from a computer running Windows, you can use PuTTY (**download PuTTY at: <https://www.putty.org/>**), the Windows Subsystem for Linux, or AWS Systems Manager Session Manager. If you plan to use PuTTY, you'll need to convert the .pem file to a .ppk file. PuTTY config screens are below:



7. You can now continue configuring OpenVPN Access Server by directing your browser to: **<https://your AWS Elastic Public IP Address:943/admin>**. Login as **"openvpn"** with the same password used to authenticate to your Linux OpenVPN Access host.



8. To download the OpenVPN Client configuration go to: **<https://your AWS Elastic Public IP Address:943/>** and download the user-locked profile.



9. For more details information about how to use OpenVPN Access Server go to: **<https://openvpn.net/vpn-server-resources/>**

IRG5000 Router Configuration for AWS OpenVPN Access Server

Any Perle IRG5000 Router can be configured to connect over LTE to the AWS OpenVPN Access Server:

1. Download the OpenVPN Client configuration from:
https://your AWS Elastic Public IP Address:943/
 - a. Where “your AWS Elastic Public IP Address” is your Public Static IP address of your AWS EC2 Linux server running the OpenVPN Access Server
 - b. Save the OpenVPN Client configuration file “client.ovpn” on your computer
2. Open the **client.ovpn** file with a text editor and do the following:
 - a. Look for and save in a separate text file named **ca.crt** everything between the lines starting with **<ca>** and ending with **</ca>**. Your newly created **ca.crt** file should look like this:

```
-----BEGIN CERTIFICATE-----
MIICuCCAaCgAwIBAgIEXjCDnJANBgkqhkiG9w0BAQsFADAVMRMwEQYDVQDDApP
cGVuVlBOIENBMB4XDTEwMDEyMTE4NTUyNlloXDTMwMDEyNTE4NTUyNlloFTETMBEG
vGmY+isyQ3vUUMHG1gNIhvp9N4JD0MGPjHhPj1RMgqiYPwVpMMpu5JrJz9qxNAZP
tkPpQHw/A1ttoKING1C5PHq+tNEpQwdiZ/4pGhMy5vWk8RVZZzMobQ2DzWQLK1Y3
46GmOmWf6bEAiGTFt9IfE5f+5AiWiZQnwkxSPA==
-----END CERTIFICATE-----
```

- b. Look for and save in a separate text file named **client.crt** everything between the lines starting with **<cert>** and ending with **</cert>**. The **client.crt** file should look like this:

```
-----BEGIN CERTIFICATE-----
MIICwjCCAaqAwIBAgIBAgIBANBgkqhkiG9w0BAQsFADAVMRMwEQYDVQDDApPcGVu
VlBOIENBMB4XDTEwMDEyMTE4NTUyNlloXDTMwMDEyNTE4NTUyNlloFTETMBEG
vGmY+isyQ3vUUMHG1gNIhvp9N4JD0MGPjHhPj1RMgqiYPwVpMMpu5JrJz9qxNAZP
tkPpQHw/A1ttoKING1C5PHq+tNEpQwdiZ/4pGhMy5vWk8RVZZzMobQ2DzWQLK1Y3
46GmOmWf6bEAiGTFt9IfE5f+5AiWiZQnwkxSPA==
-----END CERTIFICATE-----
```

- c. Look for and save in a separate text file named **client.key** everything between the lines starting with **<key>** and ending with **</key>**. The **client.key** file should look like this:

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBAwggSjAgEAAoIBAQDIdTyY59245Q99
DHWG4Bgznnk1xsm1TyHN5amlAVEKHDOIGQK+qbXh06ClV3FrZLWlfeAexby+I2q
QRYANJuxSDtAsORWXSU/cyVxift/8Z2nVVq3uC20QRrceLkeJDXOtXD0tXaVfjak
uyv1iUt+cvAstywjtavqVZABIRpeIysGM0EDhx6OcNJJyiTPdxbsruWHZp6MX2sA
zbCBtJ5Br5xg4HQsdldgD0k=
-----END PRIVATE KEY-----
```

- d. Look for and save in a separate text file named **tls.key** everything between the lines starting with **<tls-auth>** and ending with **</tls-auth>**. The **tls.key** file should look like this:

```
-----BEGIN OpenVPN Static key V1-----
656df0a39de21daf5f120d37a0c8d002
47d3b2e89a6d591ef3d29765a613e106
d1353f0bf7b91ef5dc36b425a135f50b
e9051040bca989727a8e4aba763e96c8
-----END OpenVPN Static key V1-----
```

Please note that TLS Authentication is an alternate authentication method to the default Client Certificate method

- e. You can now save in a separate text file named **clientrouter.ovpn** everything but the commented lines starting with **#** and the certificates, keys and signatures encrypted lines. The resulted **clientrouter.ovpn** file should look like this:

```
cipher AES-256-CBC
setenv FORWARD_COMPATIBLE 1
client
server-poll-timeout 4
nobind
remote 3.136.121.240 1194 udp
remote 3.136.121.240 443 tcp
dev tun
dev-type tun
ns-cert-type server
setenv opt tls-version-min 1.0 or-highest
reneg-sec 604800
sndbuf 0
rcvbuf 0
comp-lzo no
verb 3
setenv PUSH_PEER_INFO
ca ca.crt
cert client.crt
key client.key
```

3. Follow the IRG5000 manual instruction on configuring the OpenVPN at:
https://www.perle.com/support_services/documentation_pdfs/lte-routers/lte-routers-user-guide.pdf

IOLAN SCR Configuration for AWS OpenVPN Access Server

Any IOLAN SCR Console Server can be configured to connect over Ethernet to the AWS OpenVPN Access Server:

1. Download the OpenVPN Client configuration from:
<https://your AWS Elastic Public IP Address:943/>
 - a. Where “your AWS Elastic Public IP Address” is your Public Static IP address of your AWS EC2 Linux server running the OpenVPN Access Server
 - b. Save the OpenVPN Client configuration file “client.ovpn” on your computer
2. Follow the IOLAN manual instruction on loading the OpenVPN configuration:

Note: PerleView supports the IOLAN SCR for central management, but direct IOLAN management or data traffic through the AWS cloud is supported with any IOLAN