

AWS Deployment Guide for OpenVPN Access Server

April 7, 2020

Table of Contents

1	Introduction	4
1.1	What is a Virtual Private Network (VPN)?	4
1.2	What is OpenVPN Access Server	5
1.2.1	OpenVPN Access Server Functional Architecture	6
1.2.2	OpenVPN Access Server Features.....	8
1.3	OpenVPN Access Server Use Cases	9
1.3.1	Secure Remote Access	9
1.3.2	Site-to-site connections to bring networks together.....	10
1.3.3	Multiple networks, subnets, gateways, and servers	10
1.3.4	Secure Internet traffic or contact limited-access systems.....	11
1.3.5	Secure Access to Cloud-Based Systems	12
2	Deploying Access Server	12
2.1	Prerequisites and Requirements	12
2.2	Typical AWS Deployment.....	13
2.2.1	Deployment Assets	13
2.2.2	Recommended Deployment.....	13
2.3	Installation from AWS Marketplace	14
2.3.1	Installation of Access Server - BYOL.....	14
2.4	Using an external database.....	27
2.4.1	Change database backend to MySQL or Amazon RDS.....	27
2.5	High Availability Cluster Configuration	30
3	Security	37
3.1	Secure the root user account	37
3.2	Secure the openvpn administrative user account.....	38
3.3	Installing an SSL certificate on the web interface	40
3.4	Hardening the web server cipher suite string.....	40
3.5	Going beyond recommended security procedures	41
4	Planning.....	41
4.1	Sizing.....	41
4.1.1	EC2 Instance sizing.....	41
4.1.2	RDS Sizing.....	42
4.2	Costs	42
4.2.1	Access Server Software Costs	42
4.2.2	Cost of AWS Resources	42
5	Operations	42
5.1	Logging and Debugging	42
5.1.1	Locating the client log files	43
5.1.2	Locating the server log files	43

5.1.3	Setting up log rotation for /var/log/openvpn.log.*	44
5.1.4	Logging to syslog instead of the standard log file.....	45
5.1.5	Redirecting to an external syslog server.....	46
5.1.6	A list of debugging flags	46
5.2	Troubleshooting	48
5.2.1	Authentication	48
5.2.2	VPN Connectivity	54
5.2.3	DNS Resolution	55
5.2.4	Clients cannot access the Internet through Access Server	59
5.2.5	Recovering damaged database configuration files.....	59
5.2.6	Troubleshooting License Activation.....	61
5.3	Backups & Recovery	70
5.3.1	AWS Backup.....	70
5.3.2	Access Server Configuration Backup.....	71
5.3.3	Recovering Access Server configuration from backup.....	73
5.3.4	Backing up and recovering SSL certificates.....	74
6	Support	79
6.1	Getting support for the OpenVPN Access Server	79
6.2	OpenVPN Connect Client for Windows and macOS	80
6.3	OpenVPN Connect for iOS and Android	81
7	Access Server Resources	81

Access Server AWS Deployment Guide

1 Introduction

This section provides an introduction to terms such as *private network*, *public network*, *virtual private network (VPN)*, *VPN Server*, and *VPN Clients*. It then delves deeper into the OpenVPN Access Server, its functional architecture, use cases, and typical AWS deployment.

1.1 What is a Virtual Private Network (VPN)?

To understand the term *Virtual Private Network*, we first need to understand what a *Private Network* means. We are all well aware of the various services that we obtain from the Internet—world wide web, internet radio, social networking, instant messaging, and other services—these services are meant for public consumption. The servers on the Internet offering these services are meant to be accessed by anyone and are on the public-facing side of the service.

While these servers are meant to serve legitimate users, their exposure to the Internet means that these servers on the *public network* are open to probing and attacks from malicious users. These malicious users probe Internet-accessible servers for security weaknesses and exploit them to access sensitive information.

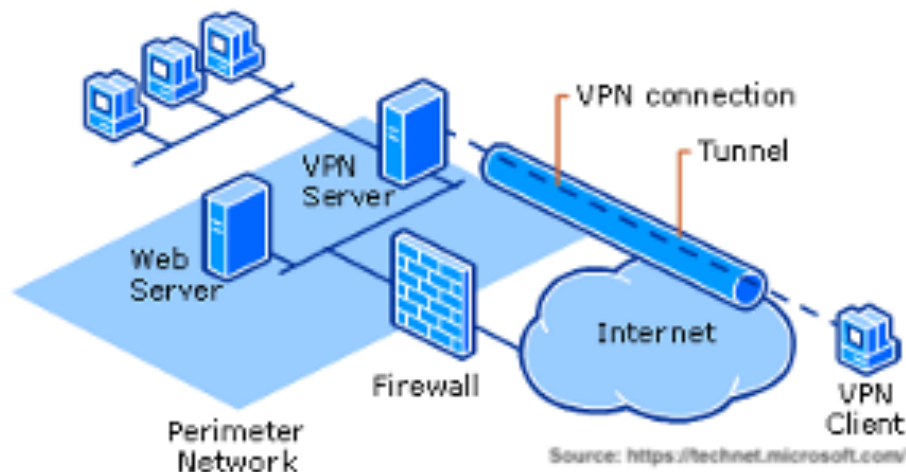
The best way to protect sensitive data and applications is to restrict access to them over *public networks* such as the Internet. The networks that connect the infrastructure that house sensitive data are isolated from the Internet, to keep them secure, by using a range of IP addresses that are unreachable over the Internet. Security is strengthened by placing access restrictions on these networks so only specific traffic only from authorized external devices can get access. These isolated and access restricted networks are referred to as *private networks*.

One can think of the security model of a private network as being similar to a castle protected by a deep and wide moat and drawbridges. The moat that isolates the castle from attack can be equated to the use of non-routable IP address ranges or use of firewalls, while the use of drawbridges to allow entry/exit can be thought of as strict access control applied to traffic and external devices.

An enterprise can have a *private network* that connects all its IT infrastructure and employee's computers to form a corporate intranet. This network allows for access to all internal IT services such as payroll, email, etc., at the enterprise's main headquarters. As the enterprise grows, the *private network* may also need to be extended to additional branch offices.

To establish connectivity between offices for their *private network* while keeping the network separate from the Internet, dedicated data transport with leased telecommunication circuits are often used. The telecommunication services used to create this connectivity between locations are quite expensive and a more economical alternative was desired.

With advances in cryptography, computing technology, and pervasiveness of the Internet, it became possible to encrypt data traffic and tunnel it over the Internet to a *server* located in the private network. The secure tunnel creates a virtual link that extends the *private network* over a *public network*. This kind of network that makes use of *public networks* to provide *private network* connectivity is called *Virtual Private Network (VPN)*.



A VPN can make use of one of many technologies such as Internet Protocol Security (IPsec), Transport Layer Security (SSL/TLS), Datagram Transport Layer Security (DTLS), to securely connect devices or networks, over *public networks*, in order to extend or form a *private network*.

The same technology that is used to create virtual connectivity between networks can also be used to connect a user's devices to a *private network*. A common use of VPNs is to provide remote employees secure access over the Internet to their company's IT services. Employees use VPN clients installed on corporate laptops or mobile devices to connect to a *VPN server* that is present in the company's private network.

The remote access use case is not limited to access for employees. Any Internet-connected device can use a VPN to be a part of a *private network*. Devices can range from normal computing devices like laptops to specialized industrial sensors or consumer electronics like smart TVs.

1.2 What is OpenVPN Access Server

OpenVPN Access Server is a software application that performs the function of a *VPN Server*. A *VPN Server* is typically deployed in the DMZ or perimeter network of a *private network* and accepts incoming VPN connections from *VPN Clients* over the Internet and provides them access to resources on the *private network*. A *VPN Client* is software that is installed on devices such as

computers, smartphones, routers and other internet-connected devices that want to be a part of the *private network* that the VPN Server is installed on.

1.2.1 OpenVPN Access Server Functional Architecture

The key functions that OpenVPN Access Server performs are:

OpenVPN protocol processing

This is the core VPN processing operations associated with establishing and maintaining VPN tunnel using OpenVPN protocol, encrypting and decrypting traffic.

Client Profile generation and storage

In order to be able to make a connection from a device to an OpenVPN server, it is necessary to have an OpenVPN client program installed. If your OpenVPN client program is installed but has no instructions yet on what server to make a connection to, and how to do so, with which certificates and encryption ciphers and so on, it is at that point basically useless. It needs configuration in order to be able to do its job. With the OpenVPN programs, this configuration can be provided to the OpenVPN client program by giving it a text file that contains the necessary information which typically includes the client private and public certificates, the server's public certificate, along with OpenVPN protocol directives.

Access Server sets up a PKI service and generates the certificates needed for the profiles. These client certificates are stored locally in an SQLite database and can alternatively use an external SQL-like database.

Authentication

After the mutual certificate authentication, further authentication of the user's identity can be carried out using PAM, LDAP, and RADIUS with an external identity server or a local database can be used for username/password authentication.

Access Server also supports second-factor authentication using Time-based One-Time Password (TOTP). Support for Google Authenticator is built-in.

Routing and Access Control

Access Server can use Network Address Translation (NAT) for remote-access or client VPN role or act as a complete layer-3 router for site-to-site networking. By default, the OpenVPN Access Server gives VPN clients access to your private network by using the NAT method (Network Address Translation). Using this method, traffic originating from the VPN clients will appear to be coming from the local IP address of the Access Server. For that reason, routing is not necessary and is much easier to implement. However, one drawback of using such method is that traffic from the private network itself cannot directly access a VPN client as the NAT engine prevents such direct contact. In order to allow a VPN client to be directly addressable via the

private network, you will need to configure the Access Server to use the routing method instead of NAT. Once that is done, the source IP address of packets coming from the VPN clients is kept intact, and direct access from the private network to the VPN client subnet is then possible.

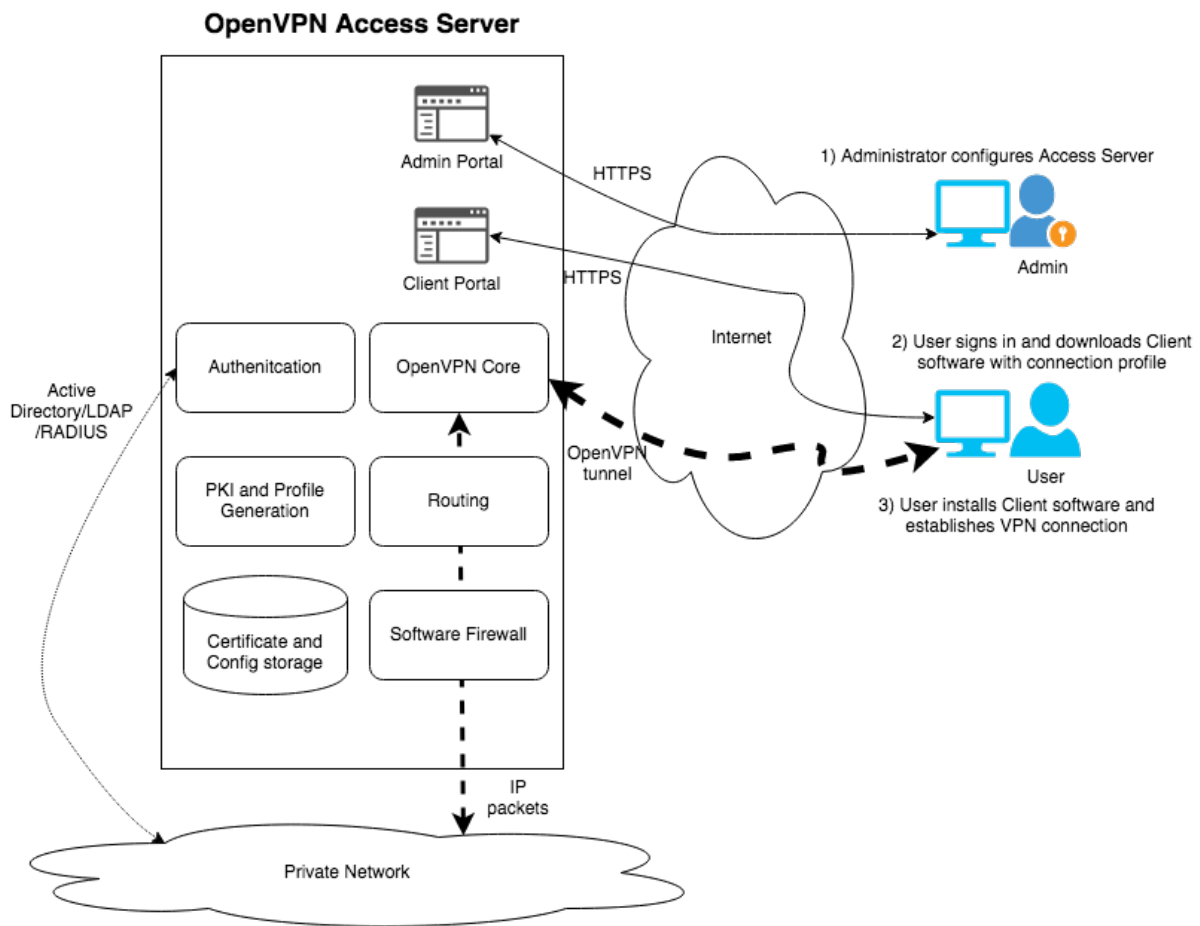
On admission, a particular client can be restricted to access only specific destinations based on the Access Control Lists (ACL) configured at the global, group, and user levels. Linux iptables is used as a software firewall to provide limited access to users based on ACLs.

Client software and client profile distribution

In order for users to retrieve their connection profiles and VPN Client software, Access Server provides a User portal which is sometimes referred to as Client Web Server (CWS). After uername/password and optional MFA authentication, the user is presented with clients for macOS and Windows that is bundled with the profile and separate profile downloads.

Configuration web portal

A web portal for the Administrator to sign in and use to configure the various settings of Access Server.



1.2.2 OpenVPN Access Server Features

- VPN protocol: Layer 3 VPN using OpenVPN protocol. OpenVPN protocol is our award-winning open source de-facto standard VPN protocol. OpenVPN runs a custom security protocol based on TLS. TLS version 1.2 is used by default, version 1.3 is supported. OpenVPN can use either UDP or TCP to tunnel traffic.
- VPN Clients: OpenVPN Clients free your users to choose their favorite device with support for Android, iOS, Linux, macOS, and Windows.
- PKI: X.509 PKI is built-in, but can use external PKI as well.
- Cryptographic Services: OpenSSL provides the core for secure communications and cryptography. The crypto suite can be customized to suit your needs, the default are AES-256-CBC cipher for encryption, HMAC-SHA256 for authentication, Diffie-Hellman Group 14, and 2048-bit RSA key length.
- Operating System: Ubuntu 18
- Database: Supports MySQL (defaults to SQLite database)
- Cross-platform Client Availability: Windows, Mac OS X, Linux, Android, Apple iOS
- Client Configuration Capability: IP address, DNS servers, WINS server, split-tunneling, specific routes, client-side scripts
- Ease of Client Deployment: Users can download customized and pre-configured clients for their device directly from Access Server's User Portal.
- Split-Tunneling: Full-tunnel and split-tunnel redirection are possible (all VPN client Internet traffic goes through the VPN tunnel or only specified traffic).
- Management Tools: CLI, XML-RPC, and Administration web portal
- Reporting & Logging: Detailed client access logs are searchable and viewable
- Authentication Methods: Supports local user database, Pluggable Authentication Modules(PAM), LDAP, Secure LDAP, Active Directory, and RADIUS access to external directories.
- Two-factor Authentication: Integrated with Google Authenticator. Extensible by plugins to support Duo MFA and other tokens
- Access Control: ACL allows for rules to contain IP address, IP address ranges, protocol, and port
- Access Control Levels: Global, Group, and User

- **Fault Tolerance:** Multiple Access Servers can be configured to act as a single cluster. Thus, deployments can scale horizontally, as needed, depending on the volume of incoming connections.

Clustering provides for active/active redundancy for fault-tolerant deployments.

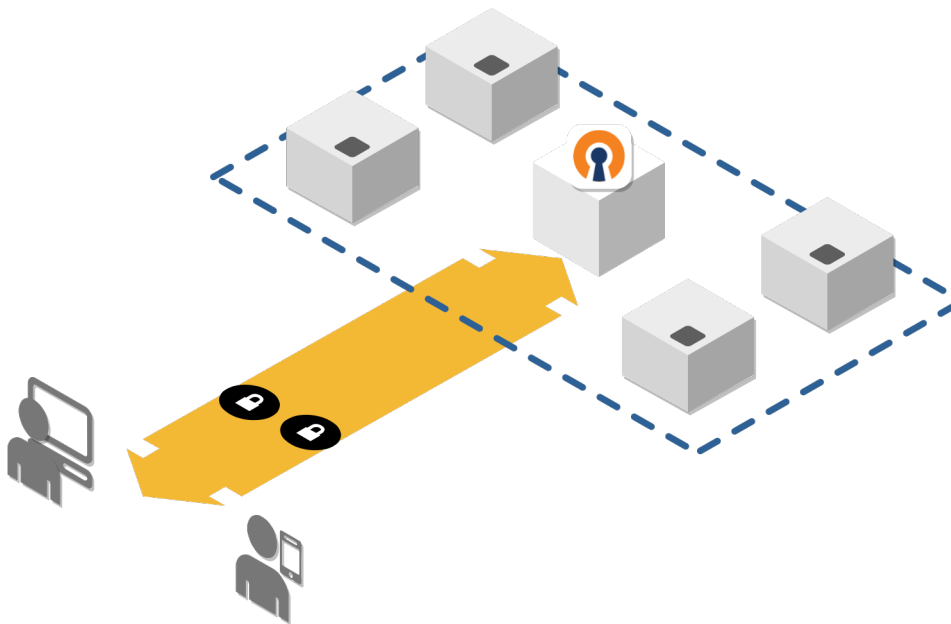
- **Branding:** Access Server portals can be branded with your logos
- **Licensing:** Based on number of concurrent connected devices. BYOL and Tiered billing.

1.3 OpenVPN Access Server Use Cases

It's important to note that due to the flexibility of computer networks and the OpenVPN Access Server product, there are many use-cases possible. The following example use-cases are not exhaustive, but they do showcase some of the possibilities.

Most of the following use-cases for Access Server assume that you are going to install the product on a server you provide, either physical or virtual, on-premise or in the cloud. An example use-case is provided for adding a VPN to a virtual private cloud like AWS or Google.

1.3.1 Secure Remote Access



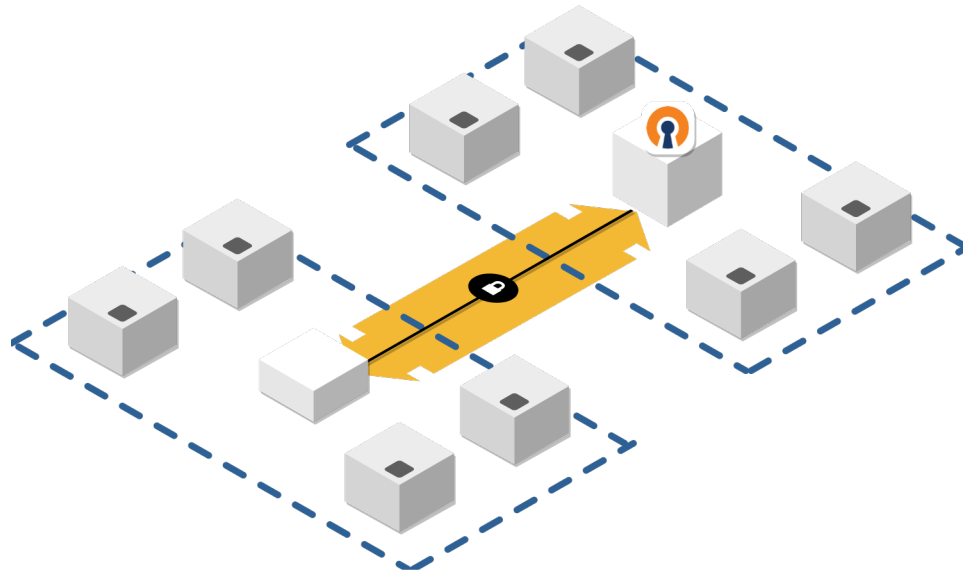
Securely access resources remotely

Whether you have servers in your office, an off-site data center, or a cloud-based system containing all of your data, OpenVPN Access Server can provide secure access. In the diagram on the right, users on their desktop computers and mobile devices are using the OpenVPN client program to make a secure connection over the Internet to the OpenVPN Access Server.

Depending on how you configure the access control rules in the Access Server, users can then transparently access either all of the resources there or only specific systems or services.

[Detailed Use Case: Remote Access VPN](#)

1.3.2 Site-to-site connections to bring networks together

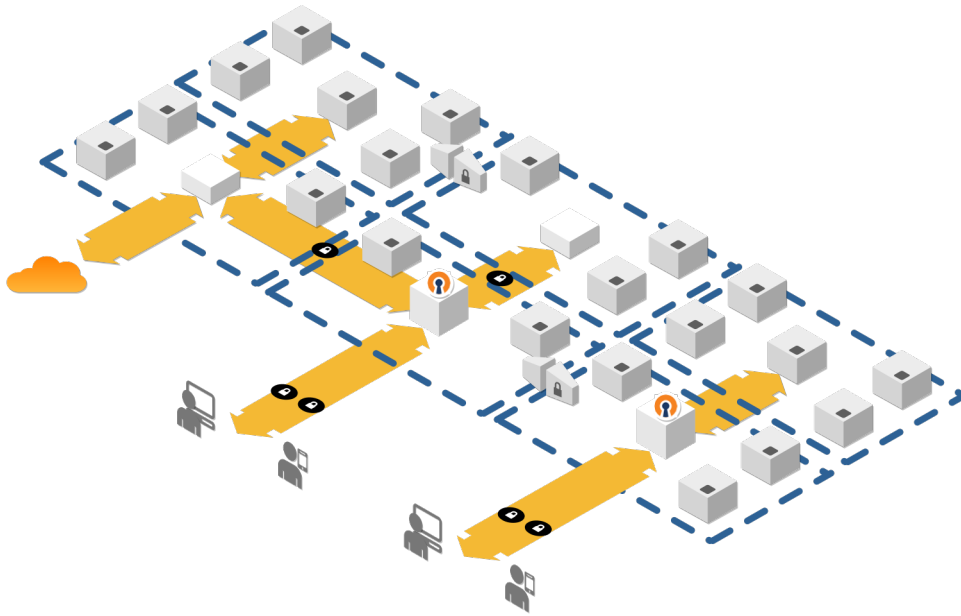


Create site-to-site connections

Using the client-server model in the OpenVPN Access Server it is possible to connect a Linux client system in one network to an OpenVPN Access Server in another network and use this connected client as a VPN concentrator or VPN client gateway system. Both terms mean to say that traffic from a whole network can go through the already established VPN tunnel between the client and the server and reach the other network. Traffic can pass in both directions which makes it possible to connect two networks together and makes accessing resources from one network on the other network transparent and easy.

[Detailed Use Case: Secure Site-to-Site Networking](#)

1.3.3 Multiple networks, subnets, gateways, and servers

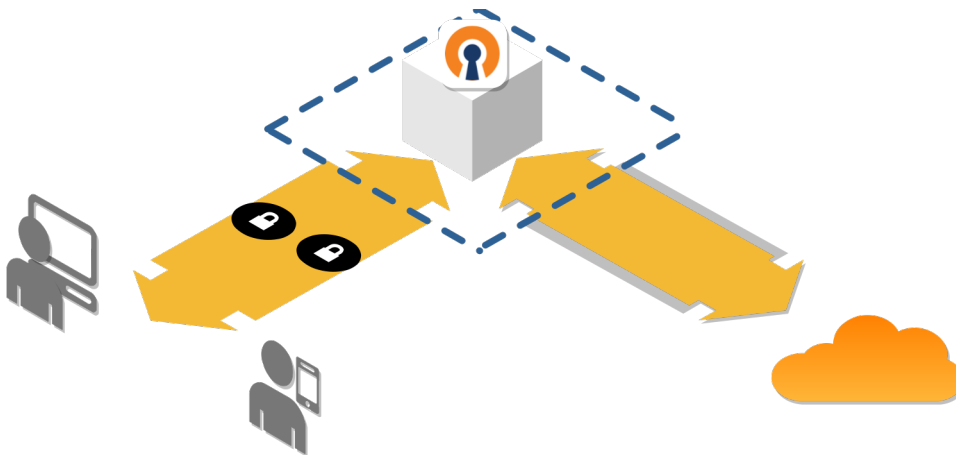


Complex inter-connectivity is possible

No matter how complex your existing setup is, the OpenVPN Access Server should integrate well. It is capable of sending specific IP addresses and ranges of traffic from a VPN client through the server. It can also send client Internet traffic through the VPN tunnel depending on what you configure. It can forward traffic coming in through the VPN tunnel intended for another subnet through the specified gateway server (handled in the OS routing table). It can be used to connect multiple different networks together in a site-to-site setup. Access Servers can be connected with each other to give access to resources or VPN clients.

Basically, if it can be routed, the OpenVPN Access Server should be able to handle it.

1.3.4 Secure Internet traffic or contact limited-access systems

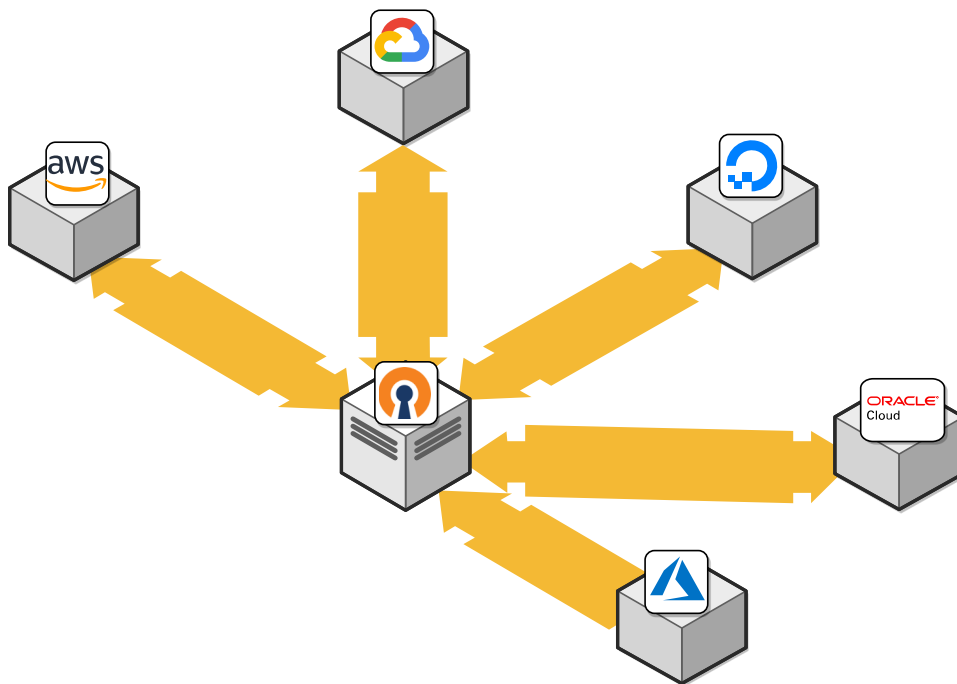


Optionally protect your Internet access

If OpenVPN Access Server is installed in a data center or cloud system, it can be used to secure your client devices' Internet connection. If, for example, you are on a public network you might want to ensure that all your Internet traffic goes into a secure encrypted VPN tunnel and to your own Access Server. From there the traffic can continue to its destination, and responses are sent back via the same path. This way programs and people snooping on the network you're on can only see encrypted packets of data that are useless to them.

Another use-case for the type of setup shown in the diagram is the ability to have traffic from connected VPN clients appear to come from the public address of the OpenVPN Access Server itself. This is useful if you have a server on the Internet or in a datacenter that blocks all access except from a whitelist of specific IP addresses that do have access. You can have VPN clients connect to the Access Server and have it handle the traffic for only that limited access system. This traffic will then appear to be coming from the Access Server, which you can add to your whitelist. Any connected VPN client will then have access to this server in a secure manner.

1.3.5 Secure Access to Cloud-Based Systems



You can extend the benefits of an IaaS cloud provider to your VPN server by using one of our preconfigured solutions. You have the option to install OpenVPN Access Server via the following cloud providers: Amazon Web Services, Google Cloud Platform, Oracle, DigitalOcean, and Microsoft Azure.

2 Deploying Access Server

2.1 Prerequisites and Requirements

Basic knowledge of IP network address and routing is required. IAM permissions to add routes to the VPC route table; add and modify security groups; assign IP addresses, create and manage EC2 instances are needed. For more information, see

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies.html

2.2 Typical AWS Deployment

Access Server can be deployed using the AWS Marketplace listing.

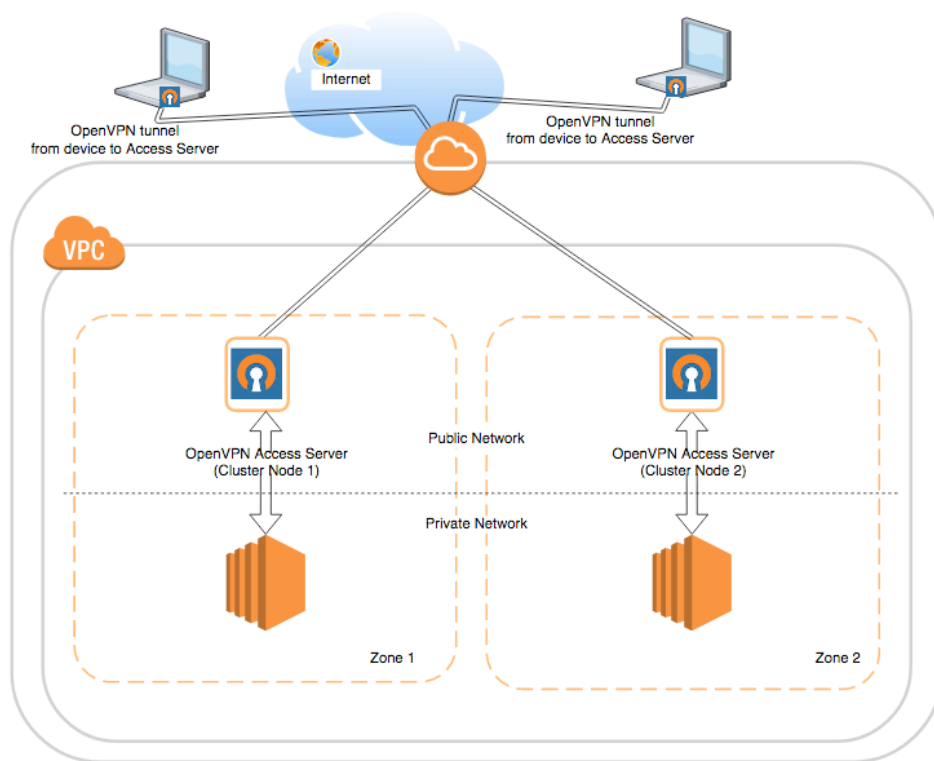
2.2.1 Deployment Assets

Deployment Assets after deployment using the AWS Marketplace listing is a new EC2 instance created using the AMI of the listing and an associated Security Group. The Security Group associated with the EC2 instance already has the appropriate incoming traffic rules for Access Server to function. More details about the installation and security group are covered in the Installation section.

2.2.2 Recommended Deployment

Access Server can provide VPN access to your VPC with just one instance. But, for a fault tolerant operation, we recommend at least one instance per Availability Zone. Depending on your fault tolerance requirements, you can have multiple instances of Access Server deployed across regions. For users to use the same credentials regardless of which Access Server they connect with, an external highly-available database should be used along with the Access Server Clustering feature.

If a single instance is deployed and the database is not externalized, a local SQLite database is used by Access Server to save configuration and digital certificates used for VPN authentication.



MULTI-AZ DEPLOYMENT OF ACCESS SERVER

2.3 Installation from AWS Marketplace

AWS Marketplace has two kinds of listings for Access Server: Bring Your Own License (BYOL) listing and tiered instances. Tiered instances come bundled with a license for a fixed amount of VPN Connections. The only difference between the two is that for BYOL one needs to purchase an Activation Key from openvpn.net to activate the Access Server whereas for tiered instances activation is not needed. For tiered, charges will be automatically billed to your Amazon AWS account. Charges consist of software license costs for the tiered instances and the cost of running the instance on AWS EC2 itself.

2.3.1 Installation of Access Server - BYOL

The Amazon Web Services (AWS) EC2 appliance (AMI) is a 64-bit based appliance that is based on Ubuntu LTS (Long Term Support) you can quickly launch on your AWS EC2/VPC in order to quickly setup your VPN server on the web. To make it more convenient for you to deploy your server in the region closest to you, we currently offer the AMI in all of Amazon's publicly available regions.

If you are looking for the specific AMI ID for one of our images on Amazon AWS for automation purposes, you can find it by going to the AWS Marketplace and going through the launch options until you reach the point where you have to select a region to launch the instance.

The appropriate AMI ID will then be displayed to you and you can cancel the launch process then.

2.3.1.1 Important notes about the BYOL licensing model

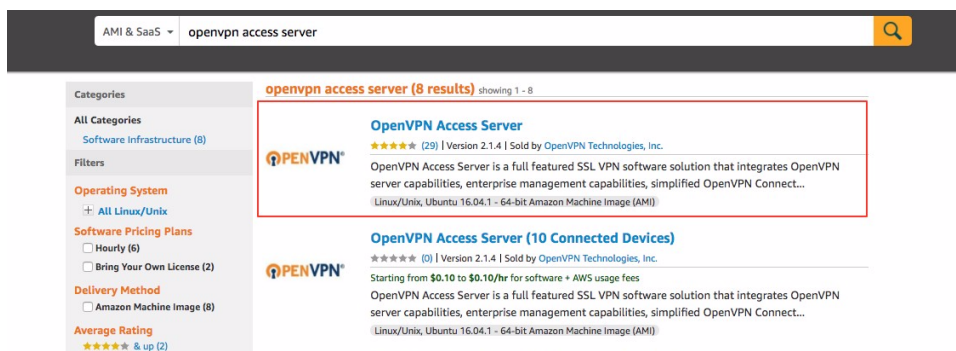
The BYOL (Bring Your Own License) licensing model is one that relies on your purchasing a software license key separately from our openvpn.net website and activating it on your Access Server installation. This locks the key to the current hardware/software configuration on the instance in question. Making changes to the instance like imaging and relaunching it, or changing the instance type, or enabling auto-scaling, will result in the license key becoming invalid, requiring you to contact us for support on this. See our [troubleshooting page regarding BYOL type license keys](#) for more information on how to request a license key reissue or check the licensing state of your BYOL type key. If you expect to have to change instance virtual hardware type or use auto-scaling on Amazon then we urge you to use our [Amazon AWS tiered instances](#) instead.

It's also important to note here that when you launch the BYOL type AMI with the instructions given below, then you do not actually need to provide a license key. If you do not provide a license key, the Access Server goes into a type of demonstration mode where all functions are available without time limit, but only 2 simultaneous VPN connections can be made at a time. To unlock more connections, you need to [purchase and activate a license key](#) on your Access Server installation.


Another licensing model we have available is the AWS tiered instance licensing model which is also available on Amazon AWS. We have a [separate guide on how to launch an Amazon AWS EC2 tiered instance](#). The AWS tiered instance licensing type works without license keys and is licensed through Amazon's systems and billed through there as well. It can survive changes to instance type and can autoscale. This is a suitable instance type if your IT security policy includes tearing down and rebuilding nodes regularly.

2.3.1.2 Launching the AMI

To get started, visit the Amazon Marketplace site by clicking [here](#). In the search bar that appears, enter **OpenVPN Access Server**, and press **Enter**. At the top of the search results you will see one with no connected devices – this is the BYOL image. Choose this one by clicking on the name.



Review the instance information on the page, select the region you would like the instance in and click **Continue** to launch the instance. This page confirms that you are using the **Bring Your Own License** image below the region.



OpenVPN Access Server
Sold by: [OpenVPN Technologies, Inc.](#)

OpenVPN Access Server is a full featured SSL VPN software solution that integrates OpenVPN server capabilities, enterprise management capabilities, simplified OpenVPN Connect UI, and OpenVPN Client software packages that accommodate Windows, MAC, and Linux, mobile OS (Android and iOS) environments. OpenVPN Access Server supports a wide range of configurations, including secure and granular remote access to internal network and/ or private cloud network resources and applications with fine-grained access control.

Customer Rating ★★★★★ (29 Customer Reviews)

Latest Version 2.1.4 (Other available versions)

Operating System Linux/Unix, Ubuntu 16.04.1

Delivery Method 64-bit Amazon Machine Image (AMI) (Read more)

Support [See details below](#)

AWS Services Required Amazon EC2, Amazon EBS

Highlights

- Powerful web interface - OpenVPN Access Server features a powerful, easy to use web administration portal that allows you to get your VPN server up and running quickly without having to work with cumbersome configuration files.
- Automated certificate management - An automated PKI built-in to OpenVPN Access Server issues user certificates and keys automatically without requiring an existing PKI

Continue

You will have an opportunity to review your order before launching or being charged.

Pricing Information
Use the Region dropdown selector to see software and infrastructure pricing information for the chosen AWS region.

For Region
US West (Oregon)

Free Tier Eligible EC2 charges for Micro instances are free for up to **750 hours** a month if you qualify for the **AWS Free Tier**.

Bring Your Own License (BYOL) Available for customers with current licenses purchased via other channels.

For a simple launch process, follow the instructions below.

Launch on EC2:
OpenVPN Access Server

1-Click Launch
Review, modify and launch

Manual Launch
With EC2 Console, API or CLI

Click "Accept Software Terms & Launch with 1-Click" to launch this software with the settings below

Once you accept the terms, you will have access to launch any version of this software in any supported region. For future launches, you can return to this page or launch directly from the EC2 console, APIs or CLI.

Version
2.1.4, released 10/26/2016

Region
US West (Oregon)

EC2 Instance Type

t2.micro	Memory	3.75 GiB
t2.small	CPU	3 EC2 Compute Units (1 virtual core)
t2.medium	Storage	1 x 4 GB SSD
m3.medium	Platform	64-bit
m3.large		

Price for your Selections:

Bring Your Own License (BYOL)
Available for customers with current licenses purchased via other channels.

\$0.07 / hour
\$0.07 m3.medium EC2 Instance usage fees + \$0.00 hourly software fee

\$0.05 per GB-month of provisioned storage
EBS Magnetic volumes

\$0.05 per 1 million I/O requests
EBS Magnetic volumes

Free Tier Eligible
EC2 charges for Micro instances are free for up to **750 hours** a month if you qualify for the **AWS Free Tier**. See details.

Launch with 1-click

You will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's [End User License Agreement \(EULA\)](#) and your use of AWS services is subject to the [AWS Customer Agreement](#).

Review the instance information on the page, select the region you would like the instance in and click **Continue** to launch the instance.

2.3.1.3 Instance Launch Options

- **Version:** The default should launch the latest version available on Amazon Marketplace. It's strongly recommended that you always run the latest version of the software to ensure that security and stability fixes are in place.
- **Region:** Select the region you would like to launch your instance in. The default is US East (N. Virginia).
- **AWS EC2 Instance Type:** Select the instance type you would like to use for your newly launched instance. The micro or small instances should be appropriate for most small workloads, however, you may want to change the instance type to a higher tier if a higher demand is to be expected. Note that some instance types are not available for use when launching into a AWS EC2 network. Please select a VPC network for all instance type availability.
- **VPC Settings:** Select the VPC network or AWS EC2 network you would like to launch the instance in.
- **Security Group:** Select the security group you would like to use for this instance. The seller settings contain all of the default ports you would need in order to configure and access your instance. If you are using a custom security group, please ensure that all of the ports are listed properly so access can be granted appropriately.
- **TCP 22 – SSH,** used to remotely administrate your appliance. It is recommended that you restrict this port to trusted IP addresses. If you do not want to do this, leave the source as **0.0.0.0/0**. To restrict ports to a specific subnet, enter the port number, then the subnet in CIDR notation (e.g. **12.34.56.0/24**). For single IP addresses, **/32** will need to be appended at the end (e.g. **22.33.44.55/32** for IP address **22.33.44.55**).
- **TCP 943 –** The port number used by the Admin Web UI. By default, the Admin Web UI is also served on port **443**. For security reasons, you can turn this setting off and restrict the Admin Web UI port to trusted IP addresses only.
- **TCP 945 –** The port number used by the clustering feature. If you don't use this feature, you don't need this port open. If you do, then contact should be made possible between the cluster nodes on their public addresses.
- **TCP 443 – HTTPS,** used by OpenVPN Access Server for the Client Web Server. This is the interface used by your users to log on to the VPN server and retrieve their keying and installation information. It is recommended that you leave this open to the world (i.e. leaving the source as **0.0.0.0/0**). The OpenVPN Admin Web UI by default is also enabled on this port, although this can be turned off in the settings. In multi-daemon mode, the OpenVPN TCP daemon shares this port alongside with the Client Web Server, and your clients will initiate TCP based VPN sessions under this port number.
- **UDP 1194 –** OpenVPN UDP port, used by your clients to initiate UDP based VPN sessions to the VPN server. This is the preferred way for your clients to communicate and this port should be open to all of your clients. You may change this port number in the settings to a non-standard port in the Admin Web UI if desired.

After verifying the instance pricing details, click the **Launch with 1-Click** button to initiate the launching process. The following dialog should appear. You should then be able to access the instance on the AWS EC2 console.

✓ Thank you for launching OpenVPN Access Server

An instance of this software is now deploying on EC2.
You can check the status of this instance on [EC2 Console](#). You can also view all instances on [Your Software](#) page.
Software and AWS hourly usage fees apply when the instance is running and will appear on your monthly bill.

Next Steps:

- The software will be ready in a few minutes.

Software Installation Details

Product	OpenVPN Access Server
Version	2.1.4
Region	us-west-2
EC2 Instance Type	m3.medium
VPC	EC2 Classic (no VPC)
Security Group	Create new security group based on seller settings
Key Pair	ackeypub

[Return to Launch Page](#)

Related Links

- [AWS Management Console](#)
- [Your Software](#)
- [Continue shopping on AWS Marketplace](#)

Service Catalog

Click [here](#) for instructions to deploy Marketplace products in [AWS Service Catalog](#).

Important: The *Access Software* link in the software subscription portal will ***not*** work until the setup wizard is complete. Please see instructions towards the end of this guide for setup wizard instructions. If the instance was manually launched from the AWS EC2 console with user data information, the setup wizard will be automatically complete upon instance instantiation.

To confirm that the instance has successfully launched, watch the **Instances** section for status. You should see the newly created instance with the same security group and AMI ID you have selected previously.

Although not strictly necessary, you should allocate a static IP address for your appliance so the IP address can be reclaimed in case of machine failure/shutdown/reboot. To do so, visit the **Elastic IPs** section in the left navigation panel.

[-] NETWORK & SECURITY

Security Groups

Elastic IPs

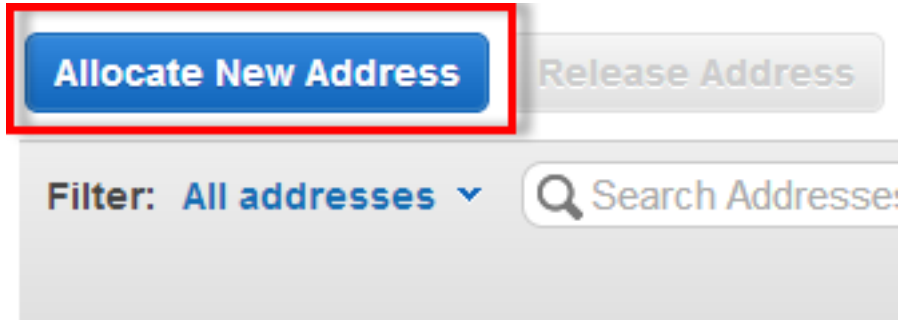
Placement Groups

Load Balancers

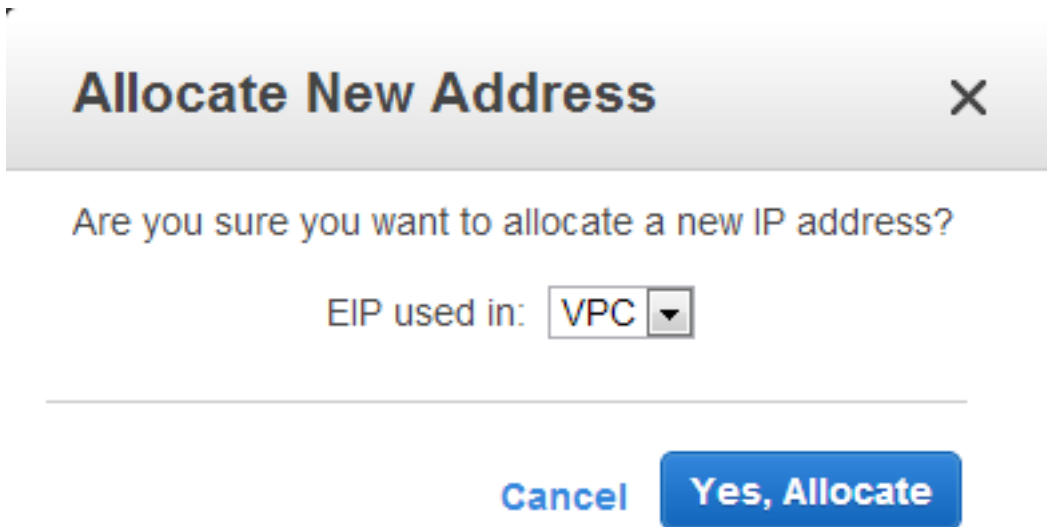
Key Pairs

Network Interfaces

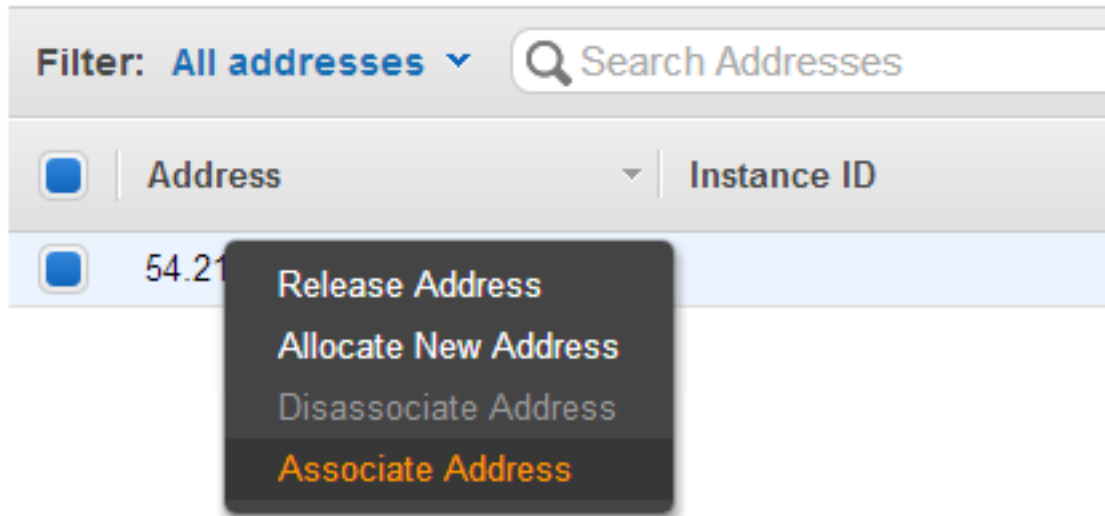
Click the **Allocate New Address** button.



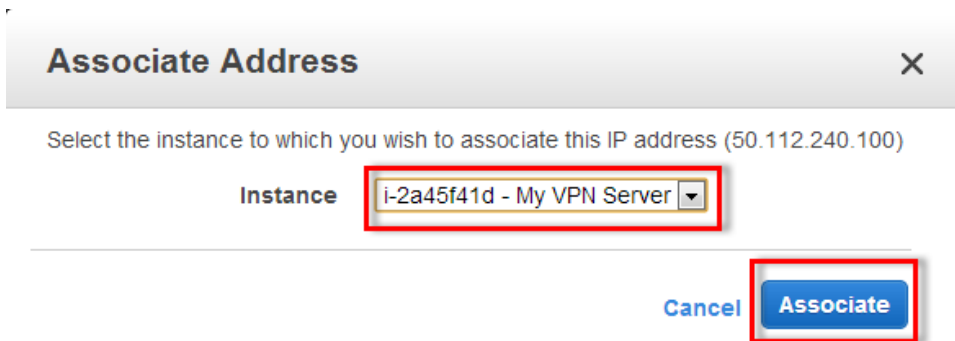
Select the IP address type you would like to allocate. This should match the type of instance you have launched previously. Afterward, click the **Yes, Allocate** button.



Right click the IP address that was created, and then click **Associate Address**.



Select the instance ID this IP address should be associated to. The instance ID can be found in the **Instances** section, and right next to the instance name. In our case, our instance name is **i-99a04ffe**.



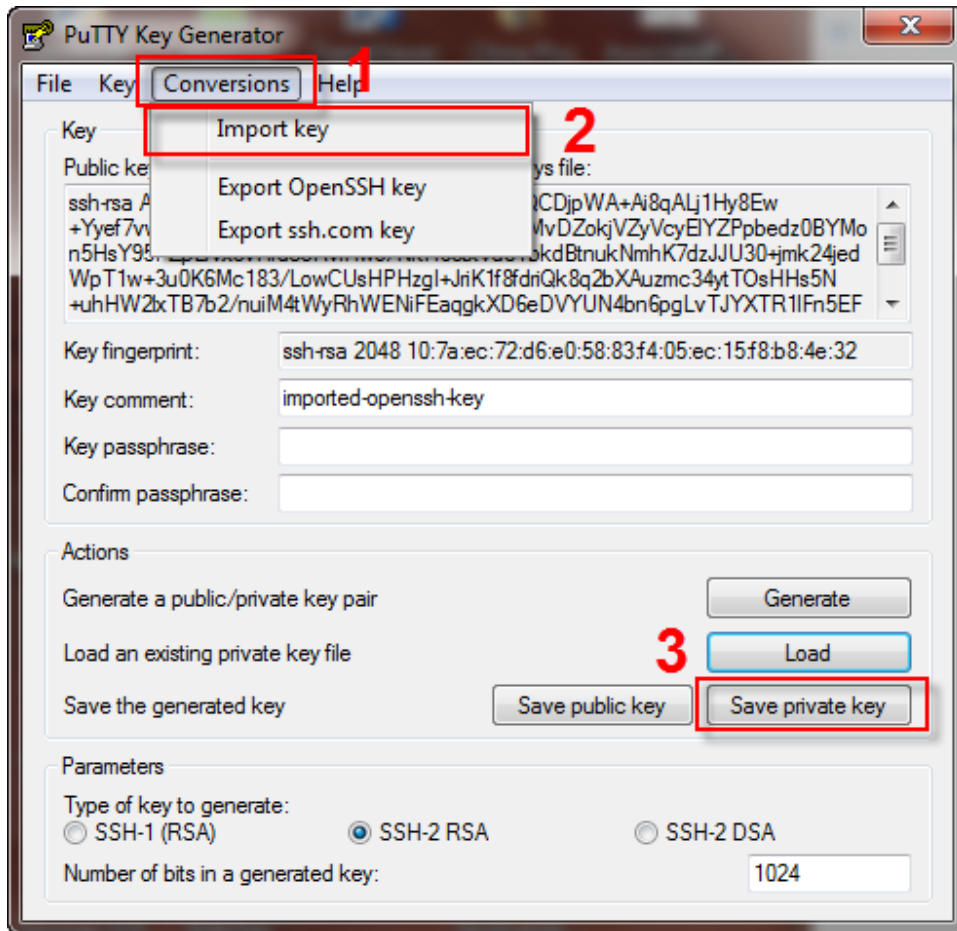
2.3.1.4 Connecting to the new Instance

Once your new AMI is successfully launched, you will need to SSH into the console using a SSH client software and the private key pair you have used/created previously. In this section, we will cover the most common case for users using the **Windows** operating system, and the **PuTTY** SSH client. If you have a different configuration, please follow Amazon's specific instructions on how to connect to your instance.

If you have not done so already, download the **PuTTY** and the **PuTTYgen** tools from this page: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

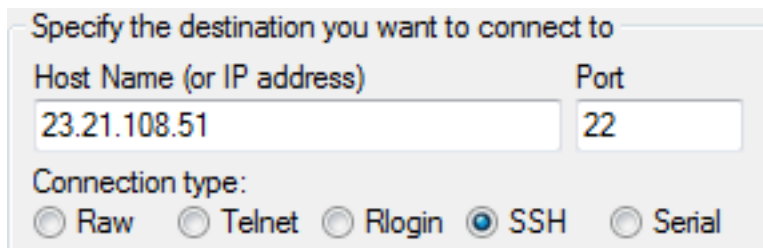
Launch the **PuTTYgen** tool. click **Conversions -> Import Key**. Select the key file you have previously used or generated, and click **Open**.

After **PuTTYgen** has successfully loaded your key file, click the **Save Private Key** button, and save the private key to a safe place. (You may want to protect your private key with a passphrase, although this is not strictly necessary.)

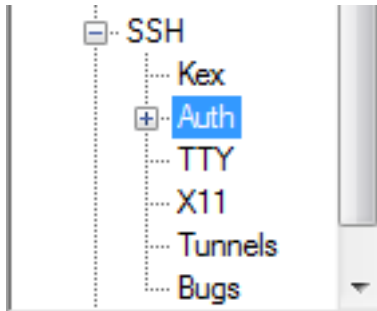


The **PuTTYgen** tool will no longer be needed at this point. To continue, open the **PuTTY** client you have downloaded earlier.

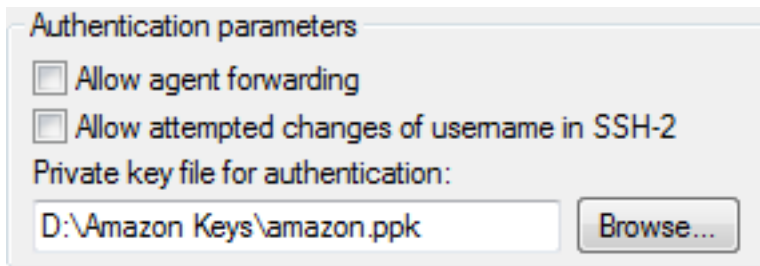
In the **Host name (or IP address)** section, enter the static IP address you have allocated previously. In our case, this is **23.21.108.51**.



Then, on the left navigation panel, navigate to **SSH->Auth**.

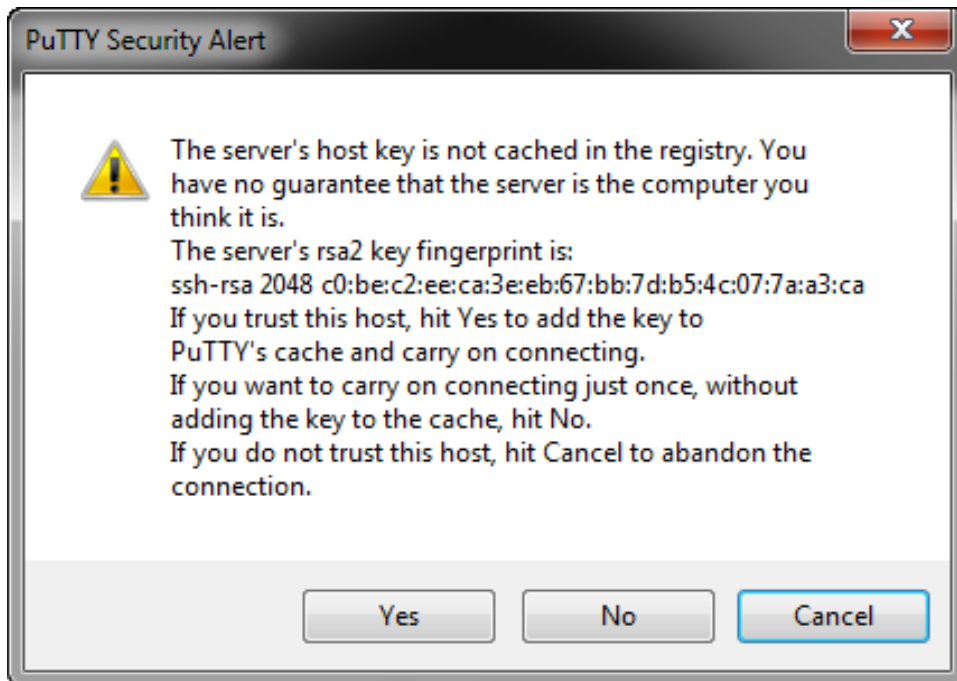


Under the **Private key file for authentication:** section, click **Browse...** and select the private key file that **PuTTYgen** has generated in the previous step.



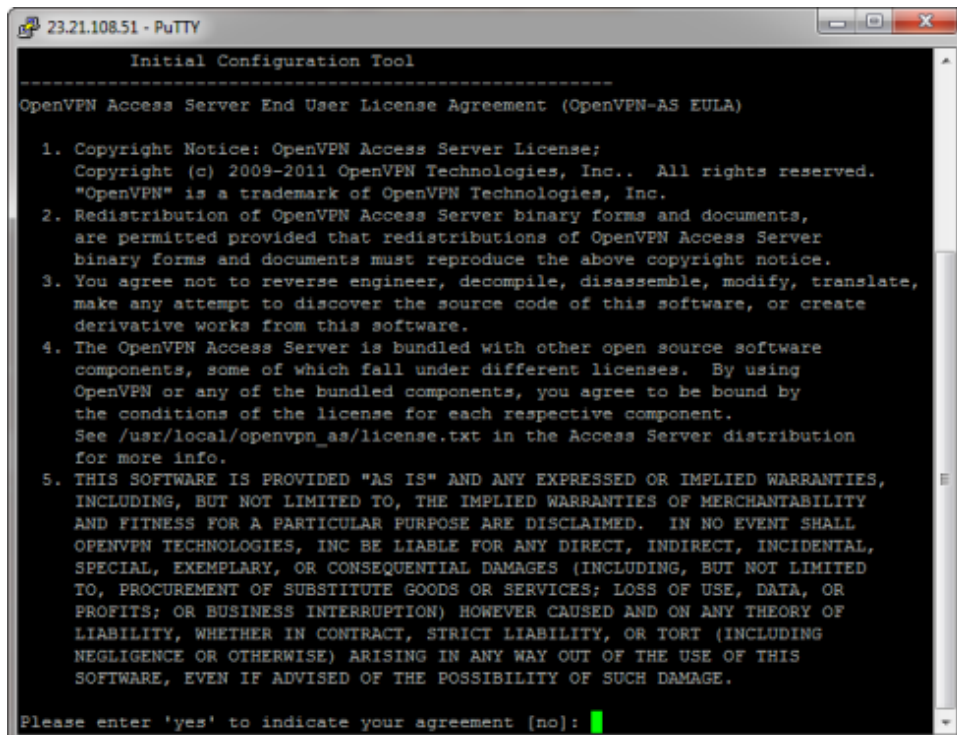
To connect to the server, simply click the **Open** button. However, to simplify the process in the future, you may want to save these settings as a profile. To do so, return to the **Session** category on the top, select a name for your session under the **Saved Sessions** box, and then click the **Save** button. The settings then can be loaded back by double clicking the profile, or by selecting the profile, and then clicking the **Load** button.

Upon connecting, you will receive a warning that **PuTTY** has not seen this server before. It is safe to simply click **Yes** on this dialog.



When prompted, login as **openvpnas**, and then press **Enter**. (**NOTE:** If you are using previous versions of our appliance, the username used is **root** instead of **openvpnas**)

If the private key you have specified was correct, you should now be logged in and the **OpenVPN Access Server Setup Wizard** should now be started. Follow the instructions below to begin configuring your server.



2.3.1.5 Running the Access Server Setup Wizard

(required, if no Amazon user-data was specified)

The OpenVPN Access Server Setup Wizard runs automatically upon your initial login to the appliance. If you would like to run this wizard again in the future, issue the **sudo ovpn-init -ec2** command in the terminal.

Read through the EULA, and enter **yes** to indicate your agreement.

> Will this be the primary Access Server node?

Explanation: If this is your initial Access Server node, press **Enter** to accept the default setting. Otherwise, if you are setting up your failover node, change this to say **no**.

> Please specify the network interface and IP address to be used by the Admin Web UI:

Explanation: This will be the interface where OpenVPN Access Server will listen to Admin Web UI requests. Make sure you have access to the interface listed otherwise you will be unable to login to your server. If you are uncertain on what interface to use, select option **1** for all interfaces. Do note that if your network did not assign your appliance a DHCP lease or if you are planning to use a static IP for your server, you will need to specify all interfaces here and follow the instructions for assigning a Static IP in the later section of this article. This option may be changed any time after the completion of the wizard in the Web Admin UI.

> Please specify the port number for the Admin Web UI.

Explanation: This is the port you will use to access the web-based administration area. It is usually safe to leave this at the default port unless customization is desired.

> Please specify the TCP port number for the OpenVPN Daemon

Explanation: This is the port clients will use to connect to your VPN server. This port will have to be forwarded to the Internet if your server is behind a NAT-based router. By default, the web-based administration area also runs on this port for your convenience, although this setting can be disabled in the Admin Web UI interface.

> Should client traffic be routed by default through the VPN?

Explanation: If you only have a small network you would like your remote users to connect to over the VPN, select **no**. Otherwise, if you would like everything to go through the VPN while the user is connected (especially useful if you want to secure data communications over an insecure link), select **yes** for this option.

> Should client DNS traffic be routed by default through the VPN?

Explanation: If you would like your VPN clients to be able to resolve local domain names using an on-site DNS server, select **yes** for this option. Otherwise, select **no**. Do note that if you selected **yes** for the previous option, all traffic will be routed over the VPN regardless what you set for this option here.

> Use local authentication via internal DB?

Explanation: If you would like OpenVPN Access Server to keep an internal authentication database for authenticating your users, select **yes** for this option. When this option is turned on, you will be able to define and/or change username and passwords within the Admin Web UI. If you select **no** for this option, Linux PAM authentication will be used and you will need to add/change/delete users within the Linux operating system itself. If you would like to use LDAP or RADIUS as your authentication method, you will need to change this after you login to the Web Admin UI.

> Should private subnets be accessible to clients by default?

Explanation: This option defines the default security setting of your OpenVPN Access Server. When **Should client traffic be routed by default through the VPN?** is set to **no**, it defines the list of subnets that your VPN clients are able to access. You are able to add more entries to this list once you login to the Admin Web UI area. This option will have no effect if **Should client traffic be routed by default through the VPN?** is set to **yes**.

> Do you wish to login to the Admin UI as “openvpn”?

Explanation: This defines the initial username in which you would use to login to the Access Server Admin UI area. This username will also serve as your “**lockout**” administrator username shall you ever lock yourself out of your own server. If you would like to specify your own username, select **no**. Otherwise, accept **yes** for the default.

> > Specify the username for an existing user or for the new user account:

Explanation: Enter the initial username you would like to use instead of the default ‘**openvpn**’.

> Type the password for the ‘user’ account:

> Confirm the password for the ‘user’ account:

Explanation: Specify the password you would like to use for the account.

> > Please specify your OpenVPN-AS license key (or leave blank to specify later):

Explanation: If you have purchased a license key for your OpenVPN Access Server software, enter it here. Otherwise, leave it blank. OpenVPN Access Server includes two free licenses for testing purposes.

After you complete the setup wizard, you can access the Admin Web UI area to configure other aspects of your VPN. Please note that as Amazon does not reveal the elastic/external IP inside the machine, the links displayed within the setup wizard will not work in accessing the web interfaces. For this reason, you will need to replace the internal IP address with the external IP that Amazon has given you. As mentioned previously, you will be able to access the Admin Web UI on both the VPN port and the Admin port unless you disable this behavior in the Admin Web UI.

Note: If you selected **yes** to the **Do you wish to login to the Admin UI as “openvpn”?** option in the setup wizard, you will need to define the password for this account by running:

sudo passwd openvpn
and press **Enter**.

2.3.1.6 Changing the default hostname

If you did not assign an elastic IP prior to launching the instance, or you have a custom hostname you would like to use, you will need to login to the Web Admin UI and configure the **Hostname** parameter manually (inside the Server Settings section). You may either use an IP address or a hostname here, although it is strongly recommended that you use a hostname since your clients will depend on this setting to be able to know where to connect to, and updating a DNS record is much easier than reinstalling all clients to update the IP address they need to connect to. Also, SSL certificates require a proper FQDN hostname in order to function properly.

Note: If you leave this setting as the default, NONE of your clients will be able to connect to your VPN server since by default it is set to a non-routable (private) IP address!

2.3.1.7 Changing default timezone

The default timezone is set to **US (Pacific – Los Angeles)**. If you reside at another timezone and you would like to change this setting, run the following command (you will be asked what timezone you would like to set):

```
sudo dpkg-reconfigure tzdata
```

The system will show the new local time after this setting is configured.

2.3.1.8 Installing NTP Client

This is recommended for all situations but especially for people that want to use Google Authenticator.

```
apt-get install ntp
```

2.3.1.9 Disabling Source/Dest Checking (Recommended)

If your VPN setup consists of a site-to-site setup between your cloud instances and your machines on-premises, you will need to disable source destination check protection on Amazon, otherwise routing will not function properly. To do this, right click on the VPN instance, select **Change Source/Dest. Check** and make sure the status is **Disabled**. This setting must also be used if for example want traffic from your VPC to go directly to the IP addresses of your VPN clients in the VPN client subnet or this security feature will block the traffic.

2.3.1.10 Setup static routes

By default, the OpenVPN Access Server gives VPN clients access to your VPC by using the NAT method (Network Address Translation). Using this method, traffic originating from the VPN clients will appear to be coming from the local IP address of the Access Server. For that reason, routing is not necessary and is much easier to implement. However, one drawback of using such method is that traffic from the VPC itself cannot directly access a VPN client as the NAT engine prevents such direct contact. In order to allow a VPN client to be directly addressable via the VPC, you will need to configure the Access Server to use the routing method instead of NAT. Once that is done, the source IP address of packets coming from the VPN

clients is kept intact, and direct access from the VPC network to the VPN client subnet is then possible. However, because the VPC does not automatically recognize the VPN subnet within the VPN instance, it does not know how to send the return traffic back to the instance. To correct this problem, you will need to add a static route in the Amazon routing table for your VPC so that the return traffic flows properly. To learn how to do this see this document on Amazon AWS VPC routing:

https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Route_Tables.html

2.4 Using an external database

2.4.1 Change database backend to MySQL or Amazon RDS

OpenVPN Access Server does not have to use the SQLite3 database files it uses by default. It can also use a database backend such as an MySQL or MariaDB server, or Amazon RDS. You can for example keep the logging database in a MySQL database while storing configuration, certificates and user properties on local SQLite3 databases. Any combination of storing locally or remotely in a database backend is possible. It should be worth noting however that there is no caching happening on the Access Server side of things. This means that while the connection between the Access Server and the remote database backend is interrupted you cannot connect to the OpenVPN Access Server. If you are going to implement this we recommend that the database server is either running locally on the system that the Access Server itself is on (and maybe use database replication), or that you run the connection between the Access Server and the database server on a reliable internal network and not over a far reaching Internet connection.

Using a shared database for the certificates it is for example possible for multiple OpenVPN Access Server nodes to accept the same certificates for incoming client connections. This does however require that each server has a copy of the same server keys stored in the config.db file. But you should not have servers share the configuration database itself if you want them to have different settings like different global VPN subnets. You could for example place an initial copy of the config.db SQLite3 database file on a new Access Server node, and then set it up to use the local SQLite3 config database, but use the shared user properties database on the MySQL database backend, so that it can share those client certificates with other Access Servers. For authentication you could use local authentication and share the user properties database, or you could use for example LDAP or RADIUS as an external shared credentials and authentication system and keep user properties separate per server by sticking with local userprop.db SQLite3 database files for each server. Running multiple Access Server nodes has some interesting possibilities but also some pitfalls. In this page however we are concentrating on the procedure of converting the database files to an RDS cluster on Amazon. The procedure is pretty much the same for MySQL and MariaDB as well; it all uses MySQL compatible protocol.

You will need at minimum **mysql-client** package installed on your system, and in some cases **libmysqlclient-dev** as well.

In this example we've setup an RDS cluster on Amazon AWS and our connection address is auroratest-cluster.cluster-ctqs9e0kxora.us-east-1.rds.amazonaws.com:3306. As usual for all our command line tools documentation we are assuming you are logged on as **root** user and

are in the `/usr/local/openvpn_as/scripts/` folder. We also always assume the default port 3306. If another port is used you can however configure it in the `.my.cnf` file below.

Open `/etc/.my.cnf` in the nano text editor:

```
nano /etc/.my.cnf
```

Add this text, and adjust the user name and password to the ones you've configured:

```
[client]
user=<MYSQL_USER_NAME>
password=<MYSQL_PASSWORD>
port=3306
```

If your username or password contains some strange characters, make sure to add quotes around it.

Press `ctrl+x`, then press `y`, and then press `enter`, to save and exit the file.

Set file permissions so only root can access it:

```
chmod go-rwx /etc/.my.cnf
```

Add a symbolic link so the root user can use the `mysql` command line tool without entering credentials:

```
ln -s /etc/.my.cnf /root/.my.cnf
```

Next, connect to the RDS instance with MySQL command line tool:

```
mysql -h auroratest-cluster.cluster-ctqs9e0kxora.us-east-1.rds.amazonaws.com
```

And use the MySQL command line prompt to create the databases:

```
mysql> create database as_certs;
Query OK, 1 row affected (0.01 sec)
mysql> create database as_config;
Query OK, 1 row affected (0.01 sec)
mysql> create database as_log;
Query OK, 1 row affected (0.01 sec)
mysql> create database as_userprop;
Query OK, 1 row affected (0.01 sec)
```

Make sure the web certificates are stored in the certificates database:

```
./sacli --import GetActiveWebCerts
```

Stop the Access Server service before converting the databases to the database backend:

```
service openvpnas stop
```

Convert the databases you want to convert using the commands below (you don't have to convert them all):

```
export SERVER=auroratest-cluster.cluster-ctqs9e0kxora.us-east-1.rds.amazonaws.com
./dbcvt -t config -s sqlite:///usr/local/openvpn_as/etc/db/config.db -d
mysql://$SERVER/as_config
./dbcvt -t certs -s sqlite:///usr/local/openvpn_as/etc/db/certs.db -d
mysql://$SERVER/as_certs
./dbcvt -t user_prop -s sqlite:///usr/local/openvpn_as/etc/db/userprop.db -d
mysql://$SERVER/as_userprop
./dbcvt -t log -s sqlite:///usr/local/openvpn_as/etc/db/log.db -d
mysql://$SERVER/as_log
unset SERVER
```

if during this step you see an error message like:

```
error opening DB mysql://server/database: libmysql lclient.so.20: cannot open
shared object file: No such file or directory
```

Then you may need to install the **libmysqlclient-dev** package on your system, and try again. Modify the **as.conf** file to tell it where to look for each database (you don't have to move them all to the database backend):

```
nano /usr/local/openvpn_as/etc/as.conf
```

Look for the lines starting with **config_db**, **user_prop_db**, **certs_db**, and **log_db**. Adjust them accordingly:

```
config_db=mysql://auroratest-cluster.cluster-ctqs9e0kxora.us-east-1.rds.amazonaws.com/as_config
user_prop_db=mysql://auroratest-cluster.cluster-ctqs9e0kxora.us-east-1.rds.amazonaws.com/as_userprop
log_db=mysql://auroratest-cluster.cluster-ctqs9e0kxora.us-east-1.rds.amazonaws.com/as_log
certs_db=mysql://auroratest-cluster.cluster-ctqs9e0kxora.us-east-1.rds.amazonaws.com/as_certs
```

Press ctrl+x, then press y, and then press enter, to save and exit the file. Finally, restart the OpenVPN Access Server:

```
service openvpnas start
```

The OpenVPN Access Server should now come back online and function with the configured database backend options instead. You can confirm by for example moving the SQLite3 database files you are no longer using out of the /etc/db folder to another location and restarting the Access Server service. If it comes back up fine then obviously it is not using those files anymore. If you are having problems getting the Access Server to start you can change your settings back

or take a close look at the `/var/log/openvpnas.log` file to determine what is going on exactly. Usually, any error messages are clearly visible there.

2.5 High Availability Cluster Configuration

A feature newly introduced in Access Server since the official release 2.7.4 is the ability to create a cluster of Access Servers for the purpose of high availability and increased load capacity. Such a cluster is an answer to the requirement that our customers have expressed for a high-availability solution and it also provides the ability to spread the load across multiple servers.

A single OpenVPN Access Server contains everything it needs to offer its services to connecting VPN clients. All the user credentials, certificates, services, and access rules, can all be present on one Access Server. This is also a single point of failure. To resolve this use the cluster feature.

In the cluster feature, we separate the storage of certificates and credentials from the OpenVPN service daemons. Our cluster solution allows you to use a MySQL type database system (MySQL, MariaDB, Amazon RDS, for example) to store all the configuration of the Access Server. These database systems can be single servers or clusters with high-availability. Amazon RDS for example can offer a fault-tolerant solution that automatically switches over to a backup server if the primary server fails, and is the recommended solution at this time. Multiple Access Servers in clustering mode can attach to such a central database and offer VPN services. A DNS-based round-robin system can ensure that VPN clients will have a single address to connect to, like vpn.example.com, and this resolves at random to any of the nodes in your cluster setup. A VPN will try these different servers in semi-random order. If an OpenVPN Access Server server fails, it will lead to a temporary connection failure for the connected client, until the client automatically tries to reconnect, and then ends up connecting to one of the other servers in the cluster. The failed server could then either be repaired, or it can be removed from the DNS records and a new Access Server could be setup, attached to the cluster, and then added to the round-robin DNS record to replace the failed node.

2.5.1.1 Before you begin

If you intend to upgrade an existing Access Server installation to the new cluster-ready version, then backup your settings first. You can use the [backup commands on the command line found here](#) to make a complete backup of the settings on your Access Server installation safely without having to stop operations on your server. If you've never made a backup before, now would be a good time to remind you that it would be a good idea to set up an automated backup plan. The information stored in the Access Server is unique and cannot be replaced, unless you wish to reinstall all your existing clients. Backups can prevent that scenario.

2.5.1.2 Setting up Amazon RDS

The cluster function requires that you migrate your settings to a MySQL type database such as MySQL, MariaDB, or Amazon RDS. In theory, as long as it is a MySQL 5.6 or higher compatible system, it should work. Amazon Aurora available as an engine for an Amazon RDS database is such a compatible system. With previous Access Server releases, the conversion from

SQLite3 local storage to a MySQL type system was already possible, and back then it had to be done on the command line manually. We now have a conversion tool built right into the Admin UI to handle that part. You do still need to have some database client support software installed for the connection to be made, and you do need to still set up the new database system yourself first. In our guide we are going to be assuming a situation where Amazon RDS is used, so this means we're doing this cluster setup on Amazon AWS, but a MySQL or MariaDB setup can also be used for the database storage, and that means it's not tied to Amazon AWS. To eliminate a single point of failure we do advise using a fault-tolerant setup for the database system, whatever solution you choose to use.

The conversion process from local SQLite3 to MySQL type database system currently has few safety checks in place. What this means is that if you convert an Access Server configuration to MySQL type database system, and then repeat that same act again from another Access Server, you will likely wipe the original settings, and the latest converted settings will be the only valid settings. So please do not repeat a database conversion to the same target MySQL database system. We will improve this behavior in future releases.

It is important to note here that you are not required to use Amazon RDS, but it is recommended, since it is fault-tolerant. Another MySQL or MariaDB system also works, and this also means you are not tied to Amazon AWS. We just assume this is the easiest setup for our customers and serves as a general guide on how to set up an OpenVPN Access Server cluster. The term cluster is also used in database systems that are fault-tolerant. Please do not confuse a database cluster setup with an Access Server cluster setup, they are two different things. You need both a cluster database setup and an Access Server cluster setup with multiple nodes to be fully fault-tolerant.

- Log on to the Amazon AWS console.
- Under **Services** look under the **Database** header for the **RDS** option (Managed Relational Database Service).
- Click **Create database** and select the engine of choice: **Amazon Aurora**, **MySQL**, or **MariaDB**.
- Default options are usually fine here, click the **Next** button to continue.
- Specify instance size and fault-tolerance settings – this depends entirely on how heavily you intend to use the system.
- Specify the **DB instance identifier**, the **Master username**, and the **Master password** (twice).
- Take note of the username and password as you will need them later, and click the **Next** button to continue.
- Select which VPC and subnet this should be launched on. You can choose if it should be publicly reachable or not.
- It is not necessary to create a default database here. Review the other settings to set them as you prefer, defaults are usually fine.
- Complete the launch and then find your new Amazon RDS database in the **Instances** or **Clusters** overview.
- Wait until the status shows that it is **Available** before attempting to use it.
- Find the **Cluster endpoint** if you are using a cluster, or just the instance's **Endpoint** when it's just an instance.

- Take note of the **endpoint** name, you will need it together with the **Master username** and the **Master password** later.

This should result in an Amazon RDS instance or cluster that is still empty right now, but is capable of storing database information. A logical next step is to install one OpenVPN Access Server from scratch, or take an existing Access Server setup, and updating it to the latest official OpenVPN Access Server release available on our software packages download page. Then the required MySQL client software can be installed, and a connection to this RDS system can be setup and tested. You are then ready to convert the database from local SQLite3 storage to the Amazon RDS database, and the cluster function can then be enabled, and additional Access Server nodes added.

Please keep in mind that Amazon RDS databases are protected by security groups. These are an Amazon specific security system that functions like a firewall. It is therefore necessary to adjust the security group settings so that your Access Server nodes that you intend to connect to this Amazon RDS database can actually reach this database.

Our example RDS database has these 3 important items that we will need later:

- **Endpoint:** astest-cluster.cluster-cw1zos4nytdr.us-west-1.rds.amazonaws.com
- **Master username:** adminuser
- **Master password:** Fw4MjHs5KPjScCkz7Yxyp5YKNh
(*this is just some made-up example*)

2.5.1.3 Initial setup of first Access Server cluster node

At this moment, version 2.7.4 which contains the clustering functionality, is available on our website as software package but we also offer images for, ESXi, HyperV, Microsoft Azure, Google Cloud Platform, and DigitalOcean. Amazon AWS version 2.7.4 is on its way and is expected to go live in August 2019. If you want to use it on AWS now, you can deploy version 2.6.1 and update it to 2.7.4 with the software package on our website.

Our images are based on Ubuntu18 x64. You may also choose to set up your own Linux system from scratch. The available installation options are listed here:

- [OpenVPN Access Server installations options](#)

When you choose to use an existing Access Server that you are already using now, and that already has users and settings configured, you can upgrade it to the latest version with the instructions below, and the licenses and users and settings will all be retained. As mentioned earlier we do advise making a backup first. The new version of Access Server with version number 2.7.4 comes with a new optional cluster feature added that does not necessarily have to be used. But in this guide of course we are going to explain how to enable and use that feature.

If the image you have launched is not version 2.7.4, or you install our software on your own self-installed OS, then in order to upgrade the Access Server to this new release, you will need to download the Access Server package from the page above and place it somewhere on the

intended server host. You can do this via a roundabout way by using your desktop computer to download the installation package from our website, and then uploading it using a tool such as SCP or WinSCP. But an easier method is to use wget, which is a tool designed to retrieve files directly from the Internet and save it directly on the file system of the Linux operating system where you are upgrading the OpenVPN Access Server program.

You can right-click the download link and select “Copy Link Address” or “Copy target” or such. The exact wording depends on the browser used. The goal is having the link to the installation package in your copy/paste buffer. Next go to the Linux server where you want to install the OpenVPN Access Server program and use wget to download the installation package file directly to the server.

Type wget followed by the pasted URL:

```
wget <paste copied url>
```

For example for Ubuntu 18 x64 installation package:

```
wget https://openvpn.net/downloads/openvpn-as-latest-ubuntu18.amd_64.deb
```

Note: if you get a certificate mismatch warning we suggest you use the `--no-check-certificate` flag to force the download.

Optional step for advanced users: You can compare the downloaded file with the SHA256sum hash mentioned on the software package overview page. Use command line “sha256sum openvpn-as-x.x.x-Ubuntu18.amd64.deb” to generate the hash, and compare it to what is listed on the site. If they match you can be certain that you have the right file and it has downloaded correctly.

Now that the installation package file is downloaded to your system you can perform the upgrade with the following command:

Upgrade installation package on Debian/Ubuntu system:

```
dpkg -i openvpn-as-latest-Ubuntu18.amd64.deb
```

Upgrade installation package on RedHat/CentOS/Fedora system:

```
rpm -Uvh openvpn-as-latest-CentOS7.x86_64.rpm
```

The upgrade process should then commence and finish. Afterwards you should reboot:

```
reboot
```

You should now have an Access Server that runs our latest release version, and you should be able to log in to the new Admin UI and see the new functions in the menu for clustering.

2.5.1.4 Firewall configuration – ports to open

These are the ports that need to be open. Most of the cloud images and appliances we offer have these ports open already by default. Amazon AWS is an exception, port 945 isn't open by default there, so this may need to be opened in the security group settings.

These ports mentioned below assume standard configuration. If you know you have changed your ports then please adjust as necessary.

- **TCP 22** – for SSH access
- **TCP 443** – for web interface access, and OpenVPN TCP connections
- **TCP 943** – for web interface access
- **TCP 945** – for cluster control channel <- this port is new and might not be open on existing installations yet, so be sure to open it.
- **UDP 1194** – for OpenVPN UDP connections

2.5.1.5 Configure and test database server connection

From the steps in the section about [setting up Amazon RDS](#) we will take the **endpoint** name, the **Master username**, and the **Master password**, and use them to test if this connection works. When access Server can make a connection to the database, you can continue with the steps to setup a new cluster. To reiterate, these are the example settings we are using in this guide:

- **Endpoint:** astest-cluster.cluster-cw1zos4nytdr.us-west-1.rds.amazonaws.com
- **Master username:** adminuser
- **Master password:** Fw4MjHs5KPjScCkz7Yxyp5YKNh
(this is just a made-up)

As usual for all our command line tools documentation we are assuming you are logged on as **root** user and are in the `/usr/local/openssl_as/scripts/` folder. These instructions are for Ubuntu/Debian but by replacing **apt-get** with **yum** you should be able to achieve the correct results on CentOS and Red Hat.

Install the required software:

```
apt-get update
apt-get install mysql-client libmysqlclient-dev
```

Next, connect to the RDS instance with MySQL command line tool:

```
mysql -h astest-cluster.cluster-cw1zos4nytdr.us-west-1.rds.amazonaws.com -u
adminuser -p
```

When asked for a password, provide the **Master password**.

You should be seeing a message like this, and you can exit with the **exit** command:

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Your MySQL connection id is 13
Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.
mysql> exit
Bye
```

This system is now able to make a successful connection to the Amazon RDS database. You can now proceed to the next step.

2.5.1.6 Setup a new cluster

Log on to the Admin UI of the Access Server, and go to **Configuration** and then go to **Cluster**. You will see a page where you can select to **Setup a New Cluster**. For our initial Access Server node, this is what we'll choose. For any future nodes you want to add to an existing cluster setup, you can choose **Join existing cluster** instead.

Note: there is a check in Access Server that tries to verify that the required MySQL library is installed in the operating system. On Ubuntu systems this is called libmysqlclient-dev and can be installed with the following command:

```
apt-get install mysql-client libmysqlclient-dev
```

If you see that this check fails with the message "AS has detected that MySQL Client is not installed on this machine" and doesn't let you continue, but you're sure the required libraries are installed, you can override this check by adding this line into the file **as.conf** and restarting the Access Server and trying again. To do this follow the steps below:

Open as.conf in nano text editor:

```
nano /usr/local/openvpn_as/etc/as.conf
```

At the bottom add this:

```
OVERRIDE_LIBMYSQLCLIENT_ASSERT=1
```

Press ctrl+x and press enter to save settings and exit the file.

Then restart the Access Server service:

```
service openvpnas restart
```

And try again.

If your database backend is on MySQL already, then you can just click **Save** to confirm that you want to convert your current Access Server node to a cluster node. It will then restart and you're ready to start using your cluster setup. However, if your database backend is the default SQLite3, and not yet on the MySQL type database system, then you will see additional fields where you

can enter the following information to do the conversion to MySQL type database. In the following steps we will assume you have yet to convert to a MySQL type database.

During this step of setting up a new cluster, any existing user certificates and settings will be converted and placed into the cluster configuration. This conversion can only be done once, so it's not possible to repeat this to combine multiple different Access Servers into one cluster. The first Access Server you use to setup a cluster will be the master data set and any Access Server nodes that you add to the cluster later on will use that master data set, and changes made afterward to user configurations in a cluster will apply for all the nodes in the cluster.

- **MySQL Username:** enter the **Master username** here.
- **MySQL Password:** enter the **Master password** here.
- **Hostname or IP:** enter the **endpoint** name here.
- **MySQL port:** the default is port 3306.

Press the **Save** button to convert the local SQLite3 databases to the new MySQL type databases. The Access Server will now take a short while to convert things. If your user base is very large, it may take some time for this to complete fully. Once it is done, a restart of the Access Server will be done automatically. Once ready you will be back at the Admin UI login prompt, and you're ready to start using your cluster setup.

2.5.1.7 Round robin DNS

We advise that this cluster setup is used in combination with a round robin DNS record. This is basically a DNS name like vpn.yourcompany.com that contains multiple A records. Each A record points to one of the servers in your cluster. When a client connects, it will resolve to one of those nodes, and connect to it. If that node becomes unavailable for whatever reason, it will try the next node automatically. We have made adjustments in the server and the client software to accommodate automatic switching to the next available server node. If a node becomes defunct, you can remove it from the DNS record. We advise that you keep the TTL on such records low, so that it picks up on changes to the DNS records more quickly.

The web interface of the Access Server has an option in the cluster section to have new nodes that join the cluster automatically configure themselves to provide client connection profiles with that cluster-wide round robin DNS name, and also an option to set a single round-robin DNS name on all nodes currently in the cluster setup in one go. It is necessary for each individual node to know about this round robin DNS name, because each node individually is capable of generating client connection profiles, and these contain the connection endpoint address that a VPN client will need to know to start a connection. If it points to only a particular node (with a non round robin hostname or IP address) in the cluster, then when that node goes down, clients will not be able to establish a connection with the other nodes, as they will try to connect to this particular node (with non Round Robin hostname or IP address) only. This is why round robin DNS is recommended for this type of setup.

2.5.1.8 Fault detection and avoidance

AWS Route 53 can be used to implement DNS round-robin to the Access Server nodes in a cluster. Route 53 health checks as documented in <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-simple-configs.html> can be used to check the health of Access Server nodes. An HTTPS check to the admin portal of Access Server nodes in the cluster will detect AZ, instance and application fault.

2.5.1.9 Adding more nodes to the cluster

You can set up a new Access Server and install the latest Access Server build on it, and log on the Admin UI. Then you go to **Cluster** and you select **Join existing cluster**. You can then enter the database connection details and join the cluster. It is important to note that if this is an Access Server with existing users and configuration, then this node that you are adding will be wiped clean (but backups are made automatically just in case) and will get its user certificates and other information from the cluster instead.

3 Security

Always follow AWS best security practices

https://d0.awsstatic.com/whitepapers/Security/AWS_Security_Best_Practices.pdf

digital certificates and critical configuration are protected and only available for access with root system permission. It is important to secure root account. Any user passwords are encrypted and stored. Remember to audit security configurations,

<http://docs.aws.amazon.com/general/latest/gr/aws-security-audit-guide.html>

3.1 Secure the root user account

When you deploy one of our appliances for ESXi or HyperV it comes with a rather simplistic password for the root account. We do take the precaution with our appliances that accessing the root account over the network is by default not possible. But if someone has access to the console then the default password is not very good. To replace the account password for the root user simply first log on to the operating system and obtain root privileges. Via the console you can do this directly as root user. On our AWS appliances you are relatively safe though, and you may skip this step, because on Amazon the appliances you launch must use a secure private/public key pair on an unprivileged account (openvpnas) to get in and afterwards you can sudo up to gain root privileges. And on AWS there is no console so it can't be accessed in this way, and the root account is blocked from direct SSH access. But on the ESXi and HyperV appliances you can log on to the console directly using the root account, and as such you should protect it better than with the default password. Use the command below to set a new password once you are root:

Set a new password on the account you're logged on as:

```
passwd
```

On our ESXi or HyperV appliances, or in your own custom Linux installation, it may also be useful to create your own account in Linux that you can use for SSH access, since the root account will usually not be able to do so, unless you adjust the SSH server settings to allow it. The better solution is to create your own user account and give it sudo rights. You need the sudo program installed so the commands below will take you through the steps to install sudo, create a new user account with your chosen name, and give it the right to run commands as root user. The commands below are assumed to be run as root user on a Debian/Ubuntu system.

```
apt-get -y install sudo
adduser <USERNAME>
usermod -aG sudo <USERNAME>
```

Where <USERNAME> is a name of your own choice, without spaces or special characters. You can use this new user account to log in through SSH and use programs such as SCP or WinSCP to transfer files, although that is limited to files you actually have access to. If you want to get root privileges from this new account run this command and provide your own password (not the root password):

```
sudo su
```

You should aim to have a situation where the root user can only be used directly on the console, and not over the network, and obviously with a very secure password. Additionally you should have your own user account with a very secure password that you can use to log on over the network with, and has the ability to use sudo to run commands as root user. To make things even better you should set up an SSH key-pair for user login under your own user account, instead of simple username+password authentication. But that strays pretty far into Linux system management and we feel it is better if you refer to your operating system's documentation on how to do that. On Amazon AWS at least, they insist on having things set up this way, and so by default it is indeed set up as described, with an SSH key.

3.2 Secure the openvpn administrative user account

By default the OpenVPN Access Server comes configured with a user account called **openvpn** without a password set on it. That by itself is not immediately a security issue because an account without a password set on it normally cannot be used to log on at all, especially on the images we provide. You are expected to make your own password and set it on the openvpn user account to start logging in to the Admin UI and setting things up on the Access Server. So that is not the problem, but having an account with a predictable user name is of course not a good thing to have, especially when it's facing the Internet. And the openvpn user account is also a bootstrap account meaning it has special access privileges. For example it can bypass Google Authenticator and the authentication failure lockout policy. Therefore we recommend that one of the first things you do after setting up the OpenVPN Access Server is to create a new user for yourself and give it admin privileges. That will then be your administrative user account from that moment on. You can do this from the Admin UI under User Permissions by adding a user there. If you use local authentication you can set a password for the new account there as well. If you are using an external authentication system like PAM, RADIUS, or LDAP, remember to

also add the account there as well so you can actually use it to log on to the Admin UI. Obviously test this first before proceeding with the next steps.

Next we recommend disabling the openvpn account by locking the account:

```
passwd -l openvpn
```

Note: we advise that you test logging in with the account on the admin UI of the Access Server, to confirm that logging in is now not possible.

If you want to start using this account again in the future, unlock it and set a new password:

```
passwd -u openvpn  
passwd openvpn
```

If you want to take it a few steps further it is possible to completely erase all traces of this initial administrative account. To do that follow the steps outlined below. But we recommend that you only disable the account by removing its password, instead of removing it entirely from your Access Server. If you are using for example an LDAP server to authenticate users, and you change something on your LDAP server, like giving it a new IP address or changing the bind user's password, then nobody can log on at the Access Server's admin UI anymore. Including your administrative user that you created yourself. But the openvpn user can because it's a special bootstrap user that instead authenticates to the operating system. In such a case you can give the openvpn user a password again with the command **passwd openvpn** and you can log on to the Admin UI and make corrections to the LDAP authentication settings and get things running again. But if you want to continue with the steps to completely remove the openvpn account then do the following:

Delete the user from the operating system:

```
deluser openvpn
```

Open as.conf in a text editor:

```
nano /usr/local/openvpn_as/etc/as.conf
```

Locate this line:

```
boot_pam_users.0=openvpn
```

And comment it out like so:

```
#boot_pam_users.0=openvpn
```

Press ctrl+x, then y, and then enter, to save and exit the file. Then restart the Access Server service:

```
service openvpnas restart
```

Finally remove the user from the Access Server database:

```
./sacli --user "openvpn" UserPropDelAll
```

If you ever lose access to your server, either because of the steps above or because you have lost the password and have problems recovering access, then give our [troubleshooting authentication problems](#) page a try.

3.3 Installing an SSL certificate on the web interface

By default the OpenVPN Access Server comes with a self-signed certificate to at least get things working. Such a self-signed certificate cannot be automatically verified by your web browser or an OpenVPN client program to check if the server it is contacting is really your server, and not some other server pretending to be. SSL certificates allow for the web browser to automatically verify if you are connecting to the real server, and to automatically trust the server so that the web interface will not show a warning message about not being able to validate the authenticity of the server, but instead show a nice green padlock icon in the address bar in the browser.

This requires that your OpenVPN Access Server is set up with an FQDN DNS name that points to the public IP address that the Access Server can be reached at from the Internet, and that this FQDN DNS name is configured correctly in the Admin UI under Server Network Settings in the Host name or IP address field. We recommend that you set up this FQDN DNS name in all cases, not only because it is required for an SSL certificate to function properly, but also because if ever in the future you change the IP address of your Access Server, for example if you move it to another Internet connection, then you need only update the DNS record and all clients will be able to find the server again. If however you configure it to IP basis only, then you will have to reinstall all your clients if you move your server to another public IP address.

See the page on how to [install an SSL certificate on the Access Server web server](#) for more information on how to do this.

3.4 Hardening the web server cipher suite string

The web server built into the Access Server by default uses HTTPS SSL encryption. This secures the connection between the web browser and the web server, so that any credentials you enter on the web interface cannot be intercepted by a “man-in-the-middle” attack or be seen in plain text on the network connection. Instead that information is all nicely encrypted. The cipher used to encrypt this information is one that is agreed upon by the web server and the web browser. The server offers a number of ciphers that it allows to be used, and the web browser then picks (usually) the best one of those that it can support and uses that to encrypt information. The list of ciphers that the web server allows is called the cipher suite string. By default the cipher suite string that the Access Server comes shipped with is reasonably secure, but not overly so. There are some older ciphers allowed to offer compatibility for older web browsers and operating systems, like Windows XP for example. In most cases though you will probably want to run the web server through its paces using an online SSL security checker like [Qualys SSL Labs SSL Server Test](#) to see what grade your current settings get and then adjust the cipher suite string to eliminate weak ciphers and thereby improve the grade and thus the security of your web server.

This can have as consequence that older browsers and operating systems can't connect to the web interface anymore, though.

The cipher suite string must be set through the command line, and is described in the [custom cipher suite string for the web server](#) section of the [command line tools documentation](#).

3.5 Going beyond recommended security procedures

Some of our customers do not want the web services visible on the Internet, but only want the OpenVPN daemons reachable for VPN tunnel termination. We advise against doing this because of the fact that managing the Access Server without a web service makes things a lot more difficult. You would then have to rely on using the [command line interface tools](#) to manage the Access Server settings, users, and certificates, and also the distribution of the required connection profiles to the users. Having the web services available makes this a lot easier. Furthermore, the OpenVPN Connect Client is tied into the web services of the Access Server using a secure XML-RPC connection over SSL. In short, this allows any user with valid credentials to log in with the OpenVPN Connect Client, instead of having to install separate user-locked connection profiles for each and every user that needs to log in from a client computer. Making the web services unreachable from the Internet breaks this functionality and forces you to use user-locked or auto-login profiles only. In short, you would end up breaking some of the designed functionality, and force you to do some extra work.

However, if you really want to, you can choose to for example only allow ports TCP 443 and UDP 1194 default ports for the OpenVPN daemons from the Internet through your firewalls to your Access Server installation, and then disable the **service forwarding** options for the client web UI and the admin web UI in the **Server Network Settings** page. Those two actions together will make the web interface unreachable from the Internet but still allow incoming user-locked and auto-login connection profile based OpenVPN tunnels to make a connection. But server-locked profiles will not be able to connect anymore. To learn more about what the service forwarding is, and what effect it has, check the description in the command line page describing [how to configure the web service forwarding settings](#). To learn about the various types of connection profiles see the [connection profiles](#) page.

4 Planning

4.1 Sizing

4.1.1 EC2 Instance sizing

In general, to provide sufficient VPN data rate to VPN clients the number of CPU cores should increase as the number of concurrent VPN connections increases.

Our AWS Marketplace listings provide the recommended instance type for a given number of concurrent VPN connections. Those instance types along with the corresponding memory and EBS volumes are adequate for good performance.

If support for high volume of VPN Connections is desired, we recommend deploying multiple lower-tier instances of Access Servers instead of one high-powered instance.

For service limits, please refer to the appropriate AWS documentation

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-resource-limits.html>

For right-sizing, see:

<http://docs.aws.amazon.com/whitepapers/latest/cost-optimization-right-sizing/identifying-opportunities-to-right-size.html>

4.1.2 RDS Sizing

For cluster setup we recommend using a Multi-AZ deployment. The size of the database will increase as the number of users for which connection profiles are created increase. We suggest starting small with db.t3.micro instances and then scaling up as the number of users increase or choosing the right instance type based on the number of users you forecast.

4.2 Costs

4.2.1 Access Server Software Costs

Software costs is based on the number of VPN Connections that need to be handled at the same time. There are two options for paying for the software:

1. Bring Your Own License: A software activation key can be purchased online from openvpn.net. Pricing is subject to change and the latest prices are available on the website.
2. Tiered Billing: The software cost is billed per hour of use or on a yearly basis and is bundled into you AWS bill. The cost varies based on the number of VPN connections. The costs are provided in the AWS Marketplace listings.

4.2.2 Cost of AWS Resources

Please use the following AWS resources to estimate costs for your deployment

<https://aws.amazon.com/rds/pricing/>

<https://calculator.aws/#/>

<https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/billing-what-is.html>

Monitor your usage and costs:

<https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/monitoring-costs.html>

5 Operations

5.1 Logging and Debugging

There are log files on the client, which are most useful for figuring out why a client is having problems making a connection to a server, and figuring out which routes and instructions it is

receiving. And there is the Log Reports section in the Admin UI which is generally used to figure out when a user connected, for how long, when people logged onto the web interface, how much data they've used, and if there were any simple type of errors when authenticating and connecting. On the server there are log files that contain technical information, and this technical information can also instead be sent to syslog locally. If you want it sent to a remote server, configure a rule in the local syslog daemon to redirect it to a networked syslog server. The technical information contained in the server logs can be expanded to include various extra information. To do this, for specific functions in the Access Server, there are debug flags, which can be activated in **as.conf**. This is explained on this page as well, further down.

5.1.1 Locating the client log files

There are log files on the client, which are most useful for figuring out why a client is having problems making a connection to a server, and figuring out which routes and instructions it is receiving.

Log file location for the OpenVPN Connect Client for Windows:

C:\Program Files (x86)\OpenVPN Technologies\OpenVPN Client\etc\log\openvpn_(unique_name).log

The OpenVPN Connect Client for Mac:

/Library/Application Support/OpenVPN/log/openvpn_(unique_name).log

Macintosh may not show you this folder in finder as it only shows you certain things and hides others. So to get to the /Library folder, open Finder and in the menu at the top choose **Go** followed by **Go to folder** and then enter the path **/Library** to get into that directory. You can then go to the correct folder and look up the log file. Please also note that the OpenVPN Connect Client for Macintosh will have permissions set on the log file so that you cannot normally open it. To bypass this, right click the log file and choose the **Get info** option in the menu. Then at the bottom, under **Sharing & Permissions**, you will be able to use the yellow padlock icon to unlock the settings and to give **everyone** read access. Then you will be able to open the log file with a right click and selecting **Open with** and then choosing something like **Text editor** to view the contents of the log file.

5.1.2 Locating the server log files

On the server there are log files that contain technical information. The technical information contained in the server logs can be expanded to include various extra information. To do this, for specific functions in the Access Server, there are debug flags, which can be activated in **as.conf**. This is explained on this page as well, further down.

On the OpenVPN Access Server there is the server side log:

/var/log/openvpnas.log

/var/log/openvpnas.node.log (in case of a failover setup)

In the event that you are having problems with starting the Access Server or certain portions of it, for example the web services, then it may be useful to stop the Access Server service, move the log file aside, then start the Access Server service, and stop it again immediately. This creates a new clean log file that contains the startup and shutdown sequence of the Access Server and no other extraneous information. This makes analysis of the log file much easier. To do so use these commands in order:

```
service openvpnas stop
mv /var/log/openvpnas.log /var/log/openvpnas.log.old
service openvpnas start
service openvpnas stop
```

You can then grab the **/var/log/openvpnas.log** file for analysis and start the Access Server again:

```
service openvpnas start
```

5.1.3 Setting up log rotation for **/var/log/openvpnas.log.***

OpenVPN Access Server normally keeps on logging until the disk is full. It does do rotation of log files, but the amount of log files just grows endlessly. The threshold for the log file to be rotated out, meaning a threshold in bytes after which `openvpnas.log` gets renamed to `openvpnas.log.1`, and a new `openvpnas.log` file is started for logging purposes, is set by default to about 1 megabyte. Basically it creates a new log file and the old one gets renamed to `.1`, `.2`, `.3`, etc, as time goes on. It's always sequential meaning that `.1` is more recent than `.2`. The log file with the highest number will be the oldest file – it just keeps bumping up the file names.

You can set up a cron job that runs every so often and clears out any really old files. The amount of files you choose to retain times the file size of the log rotation setting determines how much log data you're going to be retaining in total, ensuring you never go over a certain amount of bytes used for the OpenVPN Access Server's log files.

Note: you can also simply log to syslog which is explained below, and syslog should always already have rotation rules set on it in the operating system, that cleans it up regularly.

We assume that you are logged on to the server with **root** privileges when doing any command line tasks on the Access Server. If you want to adjust the size of the log file before it gets rotated to a new file, then this can be set in **as.conf** with the **LOG_ROTATE_LENGTH** parameter using the instructions below.

Open **as.conf** for editing in nano text editor:

```
nano /usr/local/openvpn_as/etc/as.conf
```

At the bottom add this line. The number represents bytes. Default is around 1000000 bytes (about 1 megabyte):

```
LOG_ROTATE_LENGTH=1000000
```

Press ctrl+x, then press y, and then press enter, to save and exit the file. Then restart the Access Server service:

```
service openvpnas restart
```

Now your Access Server will make log files of the specified file size. But that doesn't clean up old log files yet. To do that, set up a cron job that, for example, once a day clears out any log files that are numbered .15 or higher at 4 am every night. Adjust the command as you like to set your own limits and time of execution.

Open the crontab file for the account you are logged on as:

```
crontab -e
```

When doing this for the first time you may be asked which text editor to use. We tend to advise nano as it's easy to use. At the bottom of the crontab file add this line:

```
0 4 * * * rm /var/log/openvpnas.log.{15..1000} >/dev/null 2>&1
```

Press ctrl+x, then press y, and then press enter, to save and exit file (if you use nano).

Now every night at 4 am, this script will delete files called **/var/log/openvpnas.log.15**, and the way through to **/var/log/openvpnas.log.1000**. That should pretty much guarantee that you only end up with the main log file and 14 extra older log files.

5.1.4 Logging to syslog instead of the standard log file

This is a configuration setting that enables logging to the local syslog daemon. Instead of logging to a file it logs to syslog instead. If you want to redirect to another syslog server on the network you can configure the operating system's syslog daemon to redirect any OpenVPN Access Server service syslog line to an external network syslog server. All syslog lines regarding Access Server will contain the keyword **openvpnas** in it so it is possible to filter for this with a rule in the syslog daemon, and forwarding only that information.

Open the **as.conf** file for editing:

```
nano /usr/local/openvpn_as/etc/as.conf
```

At the bottom add this line, making sure it's CAPITALIZED:

```
SYSLOG=1
```

Press ctrl+x, then press y, and then press enter, to save and exit the file. Then restart the Access Server service:

```
service openvpnas restart
```

It will now log to the syslog daemon, which by default is logging to the file **/var/log/syslog**.

5.1.5 Redirecting to an external syslog server

We're going to be assuming you're using the Ubuntu operating system, so these instructions are for that OS only. It's probably similar or the same on other operating systems too but you may have to look up documentation and make adjustments as needed. The appliances we provide are currently all based on Ubuntu so these instructions should work for you if you use one of our prepared systems on Amazon AWS, HyperV, or ESXi.

Create a file for the rsyslog daemon rule:

```
nano /etc/rsyslog.d/openvpnas.conf
```

The file should be a new empty file. Add this line to log to an external UDP syslog system:

```
if $programname == 'openvpnas' then @remote.syslog.server
```

Or instead this line if it is an external TCP syslog system:

```
if $programname == 'openvpnas' then @@remote.syslog.server
```

Press ctrl+x, then press y, and then press enter, to save and exit the file. Now restart the syslog daemon:

```
service rsyslog restart
```

As an aside, you can also instead of supplying a remote server address like **@remote.syslog.server** simply specify another file like **/var/log/myownfilename.log** and it will log it there instead.

5.1.6 A list of debugging flags

If you don't know what you're doing then the safest is to say: don't use these. Normally only OpenVPN Inc. support personnel are the ones that use these debugging flags, or to advise customers to use a particular debug flag, when there is a specific need to debug a particular problem. But we've decided to make some of the more useful debug flags available to the general public, because some can be useful in getting more data from the Access Server for purposes other than debugging, although it has its uses to solve problems as well of course. Some of these debug flags can greatly increase the amount of logging data produced by the OpenVPN Access Server, so beware filling your hard drive with log data and running out of disk space. Not all flags produce a lot of information, but some do. And some of them even log password data or session data to the log, so beware of this. Therefore it's usually best to use some of these flags to pinpoint a problem, get log data, and then disable the debug flag. Most debug flags are set in the **/usr/local/openvpn_as/etc/as.conf** file by adding it at the bottom of the file, and cold restarting the Access Server service afterwards with this command:

```
service openvpnas restart
```

5.1.6.1 DEBUG_AWSINFO=1

Logs extra information in **liman info** output and in `/var/log/openvpnas.log` regarding the licensing process when using an Amazon AWS prelicensed tiered instance. Especially in the case of problems reaching a license activation server, the output found here will be useful to determine what the issue is. See also the [troubleshooting section for the AWS tiered instance licensing system](#).

5.1.6.2 LOG_N_CLIENTS_CHANGE=1

Whenever the internal currently connected users count is altered, the log system now mentions this alteration. This can be useful if you suspect the connected user count is off for whatever reason. An example line from the log file:

```
0000-00-00 00:00:00+0000 [-] ***** N_CLIENTS CHANGE 0 -> 1
```

5.1.6.3 FAVOR_LZO=1

This is a debug flag to override the order in which compression algorithms are chosen for connecting clients. Forces the use of LZO. In extremely rare cases can help to resolve connectivity problems from iOS devices with very specific compression problems.

5.1.6.4 API_TRACE_SA=1

If you ever have the suspicion somebody or something is making alterations to your configuration settings without your knowledge, like for example a colleague with access to the system, or a browser plugin that is supposed to only help you with some tasks like filling in forms or remembering passwords for you, but is instead messing with settings you didn't touch while you're working on other settings, then this debug flag is useful. Or if you just want to log all the changes to the configuration settings. This debug flag logs all the activity between Access Server and the configuration databases. An example line from the log file showing that the user `openvpn` is signing on to the admin UI successfully:

```
2017-09-19 17:11:54+0200 [-] *** API CALL f=authenticate args=[{'username': 'openvpn', 'password': '[redacted]', 'client_ip_addr': '12.34.56.78'}, {'log_service_name': 'WEB_ADMIN', 'request_superuser_privileges': True}] time=0.012
```

5.1.6.5 DEBUG_LOGDB=1

Normally the Log Reports page and its contents are stored in the `log.db` database file on the Access Server, and this is kept out of the log system. The reasoning here is that the log system is used for tracking and resolving problems with the OpenVPN Access Server program itself, not to keep track of who logs in and how much bandwidth they're using. That is what the log database, the Log Reports page, and the `logdba` command line tool are for. However if for some reason you want to log everything that goes into the log database to the log system as well then this is

the flag to use. An example line from the log showing that the user `openvpn` has successfully logged on to the admin UI web service:

```
0000-00-00 00:00:00+0000 [-] LOG ERR: 'LOG_DB RECORD {"username": "openvpn",
"node": "OPENVPNAS", "service": "WEB_ADMIN", "real_ip": "12.34.56.78",
"timestamp": 1505833476, "start_time": 1505833476, "session_id":
"u1OfDeOuagO1sGQg", "auth": 1}'
```

5.1.6.6 LOG_DB_XML_API_VERBOSE=1

The Access Server has an XML-RPC interface that is normally limited only to authentication and retrieval of user specific data like a user-locked profile. The OpenVPN Connect Client for Windows and Macintosh uses the XML-RPC's limited set of commands for authentication and retrieving a user-locked profile, and other functions are disabled by default. See the [XML-RPC interface paragraph in the command line tools section](#) for more details. The calls made to the XML API can be logged in the log database kept by Access Server if this particular debug flag is used in Access Server. Once activated you can use the `logdba` tool to query for XML-RPC API calls like so:

```
./logdba --csv --service_filt=XML_API --columns="+api_method"
```

And with [API_TRACE_SA=1](#) this all also gets dumped in `openvpnas.log` or `syslog` if `syslog` function is enabled.

5.2 Troubleshooting

5.2.1 Authentication

There is an authentication testing tool available in the command line called `authcli`. Using this you can quickly run tests and get some useful debugging information in the process. For example any authentication results on the command line are reported to your screen and if the authentication is successful you can see what user-specific properties are applied on this user, if any. This way you can verify for example if an expected property is actually being picked up. If it isn't then the most common problem here is that the user name that you are entering does not match what is known in the Access Server. This problem can occur if the user name known in an external authentication system doesn't match with what's configured in the Access Server "User Permissions" table for a given user name. All command line tools that come with Access Server are assumed to be run as root user in the `/usr/local/openvpn_as/scripts/` folder. [For more information on the command line tools see the page here.](#)

To use `authcli` on the command line:

```
./authcli --user <USER_NAME> --pass <PASSWORD>
```

Sample output of a successful local authentication attempt:

```
API METHOD: authenticate
AUTH_RETURN
```



```
status : SUCCEEDED
session_id : AaJkamAuZgjXwsjk+N96eA==
reason : local auth succeeded
expire : 1505404548
user : test
proplist : {'pvt_password_digest':
'9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08', 'type':
'user_connect', 'prop_autogenerate': 'true'}
```

Most authentication systems are case-sensitive and should not have a problem with matching the user name that the user enters against the user name entry in the User Permissions table in the Access Server for applying user-specific properties like auto-login privileges, static IP address, etcetera. With PAM the user name is almost always case-sensitive, meaning that when you enter as user name “Gary” at the log in prompt, but the user name is actually “gary” it just won’t accept the user name. So you are forced to consistently use the correct case everywhere, which avoids ambiguity. but LDAP on Active Directory for example does not have to be case sensitive with user names. What can happen then is that the user enters “andrew” at the log in prompt, but it is known in the LDAP directory as “Andrew”, and this correct name is then sent back to the Access Server and used there for looking up user-specific properties for “Andrew”, not “andrew”. So you have to be sure to use the correct case in the user name.

When debugging problems with authenticating against an LDAP server generally the LDAP debug options are not necessary, but you just need to use some trial and error and the **authcli** tool. The most commonly encountered problems are related to the base DN search query. Especially in cases where your search query is very specific, you may have problems getting authentication to work initially. You will usually receive an error like “user not found that meets specified criteria”. What that means is that the user was not found in that location in the LDAP directory. The user name could very well exist, but not in that place. It helps to broaden the query by looking only in DC=example,DC=com (adjust to your DC values) which searches the whole directory. If that works, then you can work on refining your search query to look in a specific location, or add an LDAP expression that must evaluate to try using the additional LDAP requirements in the Access Server configuration.

5.2.1.1 Reset default openvpn account administrative access

By default, the OpenVPN Access Server comes with a default **openvpn** user account that has full admin access to the Admin UI and has special user privileges that let it bypass the requirement for Google Authenticator, and does not adhere to the password lockout policy, and is bootstrapped or tied to the PAM authentication system so that it can always log on. We designed it this way so that if for example you are setting up LDAP authentication, and you make a mistake, you can still log on to the Admin UI and make corrections. As per our [security recommendations](#) we recommend that administrators disable this account after initial setup, and to make your own administrative users instead, that do adhere to Google Authenticator and password policy lockouts.

Of course, there may come a time when you absolutely need to get back in, and you may have forgotten your administrative username or password. The steps below restore the **openvpn** administrative user account, set a new password on it, unblock the account in case it was

blocked, disable Google Authenticator requirement for this user, and make it an admin user for access to the Admin UI again. In other words, with the steps below, you should definitely be able to login to the Admin UI again. Afterwards when you're done with your administrative tasks, we recommend you [secure the openvpn administrative user account](#) again. Start by opening a console session or an SSH session to your OpenVPN Access Server, and obtain root privileges. If you have lost root privileges, see online for instructions on resetting your operating system's root password, and then come back here to continue the reset procedure for the openvpn administrative user account.

Ensure that the user account openvpn exists:

```
adduser openvpn
```

If the user doesn't exist you will have to provide the password, twice, the rest you can leave empty:

```
Adding user `openvpn' ...
Adding new group `openvpn' (1002) ...
Adding new user `openvpn' (1002) with group `openvpn' ...
Creating home directory `/home/openvpn' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for openvpn
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
```

But if you see this result, then this user exists already, and you can move on to the next steps:

```
adduser: The user `openvpn' already exists.
```

If the user already exists, and you need to set a new password for it, do this:

```
passwd openvpn
```

If in the past you locked the account you need to unlock it:

```
passwd -u openvpn
```

Now that we know for certain that the user account **openvpn** exists and has a password set on it that we know, you can now try to login at the Admin UI with the username **openvpn** and the password you have just reset. If that didn't work, the next step is going into the configuration file **as.conf** and making sure the bootstrap user is set to **openvpn** as well.

Open the text file **as.conf** in the nano text editor:

```
nano /usr/local/openvpn_as/etc/as.conf
```

Look up the **boot_pam_users.0** line and make sure the username is **openvpn** here. If it's not, change it so it looks like this:

```
boot_pam_users.0=openvpn
```

Press ctrl+x, press y, and then press enter, to save and exit the file.

Restart the OpenVPN Access Server service for the changes to take effect:

```
service openvpnas restart
```

If the bootstrap user line wasn't set right and you have corrected it now, then it is worth a try to log in with the username **openvpn** and the password you have set on that account. If that still didn't work, it is possible that the user account has not been granted admin access in the user properties database, or that the user is blocked from logging in, or that it requires a Google Authenticator code. In the case of a Google Authenticator requirement, that shouldn't be a problem, it should accept any code for the **openvpn** user because it is a special bootstrap user. But Google Authenticator requirement can also be disabled. With the steps below we can take care of all these 3 items in one go:

Unblock **openvpn** user, make it an admin, and disable Google Authenticator requirement for it:

```
cd /usr/local/openvpn_as/scripts/  
./sacli --user "openvpn" --key "prop_deny" --value "false" UserPropPut  
./sacli --user "openvpn" --key "prop_superuser" --value "true" UserPropPut  
./sacli --user "openvpn" --key "prop_google_auth" --value "false" UserPropPut  
./sacli start
```

Try logging in at the Admin UI now. On all the systems we've ever encountered so far where the administrator was locked out, some or all of these instructions together, were successful in gaining administrative access again. If it still fails, [contact us on our support ticket system](#) and explain your situation and what you have tried so far, and we'll try to work with you to figure out how to restore access.

5.2.1.2 Common authentication errors and suggested solutions

Here are some common error messages and causes related to authentication:

password verification failed or authentication failed

Speaks for itself of course; the password and/or user name that were provided are not correct.

no stored password digest found in authcred attributes

You're using the local authentication method, and the user account you are trying to log on with does exist, but there is no password set for this user yet. Use the **sacli SetLocalPassword**

function to do this. More information on how to use this function can be found on the [user and group management](#) page.

One other possibility exists here as well, if you're using the local authentication mode. If you are specifically using a user-locked profile for connecting to the Access Server, but you are using another user name than the one this user-locked profile is locked to and meant for, then you can also see this problem. For example if you download the user-locked profile for the user called "johan" but instead enter as user name "andrew", then the Access Server assumes that because you have a valid client certificate that the user name you are providing must also exist and tries to authenticate with it. This fails of course and in local authentication mode this error can then be produced. The solution here is to use the correct user name and password with the correct user-locked profile. You cannot mix and match profiles and credentials.

DENY: user in deny list, or, user account suspended

There are a few possible reasons for this. One of them is that in the User Permissions table, the checkbox **deny access** has been checked on this user. In that case the solution is simply to uncheck that box and save settings, to restore this user's access to the server. Another possible option is that you are using an external authentication system like PAM, LDAP, or RADIUS, and that in the User Permissions page all the way at the bottom, you have checked the restriction **require user permissions record for VPN access**, but this user is not correctly spelled or not at all present in the User Permissions table. This restriction is designed so that only those user names that you have created and have present in the User Permissions table in the Access Server can log on, even if the user account exists and is valid in your external authentication system. To resolve this simply make sure the exactly correct spelling of the user name is present in the User Permissions table and doesn't have the **deny access** checkbox set on it, or, to simply disable the restriction by unchecking the **require user permissions record for VPN access** option in the User Permissions table.

username-only match fail, client username='andrew', DB username='johan'

This is what happens when you use credentials for an existing user called "andrew" on your Access Server with a user-locked profile locked to and meant for the user account "johan". These don't match. The solution here is to use the correct user name and password with the correct user-locked profile. You cannot mix and match profiles and credentials. If you are going to use the user-locked profile for the user account "johan" you must use the user name "johan" and his password to log on to the VPN server successfully.

If you are looking for a more universal type of connection profile that lets any valid user on the Access Server log on then what you are looking for is the server-locked connection profile which works only in combination with the OpenVPN Connect Client for Windows and Macintosh. All other clients must have a connection profile that is specific to the user account. If you download OpenVPN Connect Client from your Access Server's web interface with a user account that does not have the auto-login privilege, then this is the type of OpenVPN Connect Client + server-locked connection profile installation that you will get.

user not found (that meets specified requirements)

You're using the LDAP authentication method, and the user name you entered could not be found with the LDAP query you specified. Try simplifying the query to just the base DN with for example DC=example,DC=com (adjust to your situation) and nothing else, thereby broadening the search. Often the issue is caused by the user not being known in the place you're searching or the attributes are different than you expected, and the LDAP server then reports this message. If even a directory wide search yields no results then the LDAP attribute you are searching may be different in your directory server. Try "sAMAccountName" or "uid" or otherwise look up documentation of your LDAP server to find out which attribute to use.

The **auth.ldap.0.uname_attr** controls which attribute to search for. Another possibility is that your LDAP server is case sensitive with containers and objects and that you need to use lowercase name instead (cn=blabla instead of CN=blabla). Again we refer to documentation for your LDAP server to find out which settings work on your server.

One notable issue people have been found running into is seeing this error message when trying to provide the additional query parameters, to allow only users from a specific group in the LDAP directory to log on. If for example the additional query **memberOf=CN=VPN Users** is specified, it may fail. But if you make it the full query, it should work in most cases: **memberOf=CN=VPN Users,OU=Security Groups,DC=company,DC=com.**

AcceptSecurityContext error: Invalid credentials, facility=admin_bind

The above error may appear in the **openvpnas.log** log file and indicates that the credentials entered for the bind to the LDAP server were incorrect or won't allow access to the LDAP directory. If you're trying to connect to an Active Directory server it may help to create a separate user in the domain and using this to bind. Use a format like username@domain.tld as username in the Access Server's bind username field. We have also seen the same problem reported when an SSL certificate was used for communication between Access Server and the LDAP server, and the SSL certificate had expired.

In order to perform this operation a successful bind must be completed on the connection

You're using LDAP authentication while trying to bind (connect) anonymously to the LDAP service, while the LDAP service does not allow anonymous binding. The solution is to create a bind user on the LDAP server and giving it read access to the LDAP objects you want to search for user authentication. Alternatively, you can enable anonymous LDAP binding on the LDAP server but this is not as secure a solution as using a special limited bind user account for the purpose of binding to the LDAP server and then looking up user credentials for authentication purposes.

user temporarily locked out due to multiple authentication failures

Access Server implements an automatic lockout policy, which is described in the lockout policy section on the [additional security options](#) page. This automatically temporarily blocks a user account from log on attempts when the password was incorrectly provided a number of times within a specified time. The lockout policy can be adjusted to meet requirements.

Google Authenticator must be set up for VPN access

When you have enabled the requirement for users to use Google Authenticator multi-factor authentication, but this user has not yet completed the Google Authenticator enrollment process

on the client web service of the Access Server, then the Access Server will not allow the user to establish a VPN tunnel connection and warns the user about this. The solution in this case is to have the user go to the client web service and switch the "CONNECT" option to "LOGIN" and log on. A list of available files will be shown that this user can download and use. Below that a Google Authenticator code and QR code will be shown. The user can either manually type this code into the Google Authenticator application, or use a camera to scan the QR code. Once this is done, click the button "I scanned the QR code" to confirm that the code has been stored in the Google Authenticator application. Now the user can start a VPN tunnel connection and the OpenVPN client will then ask for user name, password, and the Google Authenticator code.

Google Authenticator code must be a number

After the Google Authenticator shared secret code has been typed or scanned into the Google Authenticator application, it will generate a new 6 digit code every 30 seconds. If instead you enter something unexpected like your password instead of that 6 digit code, then you will see this error message. The solution is to use the Google Authenticator application and enter the 6 digit code into the Google Authenticator field when asked.

Google Authenticator code is incorrect

This means that the 6 digit code that was entered is not correct. To understand why is it going wrong there are a few things to note first. Google Authenticator uses the current date and time, and adjusts automatically for time zones. This assumes of course that the server and the device with the Google Authenticator app both have the correct timezone set, and the correct time and date set. That information plus a shared secret key that is known by the Access Server and the Google Authenticator application which was agreed upon when you initially typed/scanned the code during Google Authenticator enrollment is all that is used to create the unique 6 digit codes that are valid for 30 seconds.

So to oversimplify this: secret shared key + current time in correct timezone = 6 digit code. Such a code is valid for 30 seconds. We allow the immediate previous and following code as well to give a bit of a leeway in the timing. What all of this means is that there are really only two possibilities when you get the error message that the Google Authenticator code is incorrect. Either the date/time or timezone setting is wrong on the server or the device running the Google Authenticator application, or the shared secret key is wrong.

You can try resetting the Google Authenticator key for this user and completing enrollment again. In most cases though the issue is a drifting system clock on the server, especially on virtual cloud-based hardware, which can be solved by installing an NTP (Network Time Protocol) client program which can automatically pull the correct time for time servers on the Internet and ensure the time doesn't drift. A time difference of more than 30 seconds can already be a problem. Mobile devices usually already do time synchronization by themselves. In the past though we have seen a bug with a specific version of iOS which skewed the time by one minute, which upset Google Authenticator. That issue was resolved by updating iOS to a newer version.

5.2.2 VPN Connectivity

If you use NAT in the Access Server, then traffic from VPN clients will appear to the Amazon network as if it is coming from the Access Server instance itself. This means it looks just like local traffic and no special actions need to be taken. But, if you use routing mode, where the source IP of the packets coming from VPN clients remains intact, then the Amazon network may have security features that block this traffic. So with routing, special steps need to be taken. Also, you will need to implement a static route that guides replies to VPN client traffic back through the Access Server instance.

In Amazon AWS, when you use routing, your VPC should have a routing table set up that needs to contain a static route that points the VPN client subnet to the Access Server instance, so traffic can find its way there. Find that routing table in the Amazon AWS console by going to the **VPC Dashboard** and going to **Route Tables**. This is where you can set up routing for the VPN client subnet, or site-to-site traffic to additional subnets behind VPN clients. When you add a subnet to the routing table you must specify a target. The target can be the AMI ID of the Access Server instance. It should then recognize that this particular EC2 instance with Access Server running on it is the gateway to the VPN client subnet, or additional site-to-site subnets.

Another item specific to Amazon is **source/destination checking**. Crudely put this is a security setting on the EC2 instance itself that basically just looks at traffic coming from and going to the EC2 instance, and if it isn't traffic that has either a source or destination IP that matches that EC2 instance's network interface address, then it just gets filtered away. Since the VPN clients in routing mode, as well as site-to-site traffic, will send packets through the Access Server while retaining the original source IP of these packets, then this security setting will filter this traffic away. Likewise traffic going to the VPN client IP addresses or site-to-site subnets and trying to pass through the Access Server will be filtered away in the same way. To resolve this go to your **EC2 Dashboard** and go to **Instances** and look up your specific instance that runs Access Server. Then right click it and in the **Networking** menu choose **Change Source/Dest. Check**. Click the **Yes, disable** button to disable this setting and let the traffic pass through. If you run a site-to-site VPN client gateway system on Amazon you will have to do the same to that instance too. For more, see:

<https://openvpn.net/vpn-server-resources/troubleshooting-reaching-systems-over-the-vpn-tunnel/>

<https://openvpn.net/vpn-server-resources/troubleshooting-client-vpn-tunnel-connectivity/>

5.2.3 DNS Resolution

Companies often run their own DNS server that they use to resolve DNS names to private IP addresses, to make accessing systems easier for users. It is for example easier to tell a user to start their Remote Desktop client program and to connect to server1 instead of having to tell them to connect to 192.168.70.243. [To learn what DNS is, see this article](#). OpenVPN Access Server supports pushing an instruction to a connecting OpenVPN client to use a specific DNS server. Actually it supports pushing 2 DNS servers, in case the first one fails to respond. This can be configured in the Admin UI under VPN Settings. The Access Server also supports sending additional instructions for DNS Resolution Zones, which functions like a type of split-DNS where only queries for a specific DNS zone are sent to the VPN server, and DNS Default Suffix,

which provides a hint to Windows to ‘autocomplete’ a partial hostname to a Fully Qualified Domain Name, or FQDN.

Unfortunately, not every operating system behaves the same in regards to DNS. Some systems will try all DNS servers at once, and accept the response from the first to respond. Others will be able to do split-DNS, and others will not. This can lead to certain problems. The guide below provides a way of checking to see if the DNS query you are doing from your OpenVPN client device, is actually making it through the VPN tunnel to the OpenVPN Access Server. And from there, of course, to the target DNS server. This information is valuable in determining whether or not the problem is at the client end, or at the server end.

5.2.3.1 Testing DNS resolution from a client system

We are going to assume that you have a DNS server configured in the Admin UI of the Access Server, under VPN Settings. We are assuming you are not using the DNS Resolution Zones or the DNS Default Suffix fields. With this setting, all DNS request should be going from the OpenVPN client, through the OpenVPN Access Server, and then to the specified DNS server. In our example we are pushing the Google Public DNS server 8.8.8.8, and our test results will reflect this in the sample outputs as well.

Install your OpenVPN client program on your chosen client system. In our example we will be using a Windows 10 Professional client system with the OpenVPN Connect Client installed, and connected to the OpenVPN Access Server. Next open a console session or an SSH session to the OpenVPN Access Server, and obtain root privileges. We will be using the tool **tcpdump** to monitor activity on port 53 TCP and UDP, the default port where DNS queries are handled. We will be flushing the local DNS resolver cache on the client side, and then resolve a number of domains simply by pinging them by name. In our test situation, there are only a handful of clients connected, and the activity of DNS queries is very low, so we can monitor it easily. If you are testing on a production system and the **tcpdump** command gives too much output, you can append a grep filter by IP address, to filter queries coming only from your specific VPN client’s IP address, to make reading and locating the DNS query results easier.

On the Access Server run these commands:

```
apt-get update
apt-get install tcpdump
```

With TCPdump installed, now run it with these parameters:

```
tcpdump -eni any port 53
```

Or, if you want to filter it by the IP address of your VPN client (adjust as needed):

```
tcpdump -eni any port 53 | grep "172.27.10.22"
```

With this running in the background, go to your VPN client’s operating system, and open a command prompt. On Windows for example you can run the **cmd** program to open an old style

DOS prompt. With that open, use the following commands to wipe the local DNS resolver cache, so it won't pull results from its own local memory, and then do an actual query.

Wipe local DNS resolver cache on Windows:

```
ipconfig /flushdns
```

Resolve some domain names:

```
ping www.google.com
ping www.openvpn.net
ping www.facebook.com
```

Each of these should yield results that look somewhat like this:

```
Pinging www.google.com [216.58.212.228] with 32 bytes of data:
Reply from 216.58.212.228: bytes=32 time=4ms TTL=56
Reply from 216.58.212.228: bytes=32 time=3ms TTL=56
Reply from 216.58.212.228: bytes=32 time=3ms TTL=56
Reply from 216.58.212.228: bytes=32 time=3ms TTL=56
Ping statistics for 216.58.212.228:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 4ms, Average = 3ms
```

On the OpenVPN Access Server you should be seeing results that look somewhat like this:

```
18:03:07.976553 In ethertype IPv4 (0x0800), length 76: 172.27.232.2.49531 >
8.8.8.8.53: 53268+ A? www.google.com. (32)
18:03:07.976579 Out 00:0c:29:c7:60:e9 ethertype IPv4 (0x0800), length 76:
192.168.47.133.49531 > 8.8.8.8.53: 53268+ A? www.google.com. (32)
18:03:07.981162 In 34:31:c4:8e:b5:67 ethertype IPv4 (0x0800), length 92:
8.8.8.8.53 > 192.168.47.133.49531: 53268 1/0/0 A 216.58.211.100 (48)
18:03:07.981181 Out ethertype IPv4 (0x0800), length 92: 8.8.8.8.53 >
172.27.232.2.49531: 53268 1/0/0 A 216.58.211.100 (48)
```

The above result from tcpdump shows that a DNS request was received from the VPN client at 172.27.232.2, and that it was directed at the DNS server at 8.8.8.8, and the request was to find the A record (IP address) for the DNS name www.google.com. The first line shows that this request is coming in at the OpenVPN Access Server, from the VPN client. The second line shows the request leaving the Access Server through the network interface with MAC address 00:0c:29:c7:60:e9. In our test setup, this is the network interface of the Access Server that goes to the Internet, which makes sense, because the 8.8.8.8 DNS server is on the Internet. The third line shows that a DNS result has been received, and the fourth line shows that this result has been relayed back to the VPN client. In this case, DNS resolution is working.

5.2.3.2 Common errors and causes

Below are a number of common problems you can see that we try to explain here and where to look for a solution.

Ping request could not find domain (...). Please check the name and try again

This can happen when the DNS servers your client system is using is badly configured, cannot be reached, or if the DNS server it is using does not know the domain you are trying to resolve. For example with local DNS servers in your own network it is entirely possible that they only know local computer systems, and have no knowledge of online names like openvpn.net or such. Usually in such a case you can configure the DNS server to forward DNS queries to a public DNS server that does know the answer to those queries, so that it is able to respond to both queries for local names and also public names. A useful step in this situation may be to again run `tcpdump` as described in the [testing DNS resolution from a client system section](#) above, and checking to see what the output of `tcpdump` is.

If you see a result like this:

```
18:07:10.082330 In ethertype IPv4 (0x0800), length 94: 172.27.232.2.54519 >
8.8.8.8.53: 50281+ A? thisdomainreallydoesnotexist.com. (50)
18:07:10.082356 Out 00:0c:29:c7:60:e9 ethertype IPv4 (0x0800), length 94:
192.168.47.133.54519 > 8.8.8.8.53: 50281+ A?
thisdomainreallydoesnotexist.com. (50)
18:07:10.082507 In ethertype IPv4 (0x0800), length 94: 172.27.232.2.57858 >
8.8.8.8.53: 65054+ AAAA? thisdomainreallydoesnotexist.com. (50)
18:07:10.082521 Out 00:0c:29:c7:60:e9 ethertype IPv4 (0x0800), length 94:
192.168.47.133.57858 > 8.8.8.8.53: 65054+ AAAA?
thisdomainreallydoesnotexist.com. (50)
18:07:10.103610 In 34:31:c4:8e:b5:67 ethertype IPv4 (0x0800), length 167:
8.8.8.8.53 > 192.168.47.133.54519: 50281 NXDomain 0/1/0 (123)
18:07:10.103641 Out ethertype IPv4 (0x0800), length 167: 8.8.8.8.53 >
172.27.232.2.54519: 50281 NXDomain 0/1/0 (123)
```

Specifically the item **NXDomain** here is important. It means that this DNS server does not know the name we are trying to resolve. Another DNS might still know the name. but this one doesn't. In the example above however we have purposefully selected a name that does not exist (or at least it didn't when we ran the test – it is possible of course someone may register the name in the future) to be sure we see the error. If you are encountering this problem you may want to try to use the **nslookup** program on a computer with direct access to the DNS server, and use it to query the specific DNS server directly, to confirm that it does know the domain.

If you see a result like this, repeated a few times:

```
18:19:29.935439 Out 00:0c:29:c7:60:e9 ethertype IPv4 (0x0800), length 76:
192.168.47.133.60180 > 1.2.3.4.53: 16427+ AAAA? www.google.com. (32)
18:19:29.935479 In ethertype IPv4 (0x0800), length 76: 172.27.232.3.51334 >
1.2.3.4.53: 37513+ A? www.google.com. (32)
```

Then what you may notice here is that you do see a query arriving from the VPN client, pass through the Access Server, and go out to the Internet, but there is no reply. Usually this means that this DNS server is unreachable, or is not a DNS server at all. In the example I have chosen IP address 1.2.3.4 which I know for a fact is not a DNS server. Obviously the query will be repeated a few times but will ultimately fail. The obvious solution here is to choose a DNS server that works, or, to make sure that there is no firewall standing in the way, blocking the queries from the VPN clients to the DNS server. In some cases, when routing is used to give VPN clients

access to servers on the private network behind the Access Server, it is a matter of a missing route. In such a case that packets from VPN clients make it to the target DNS server just fine, but it is not able to respond because it is receiving packets from a subnet it does not know how to respond to. That can be solved by [implementing static routes for direct VPN client communication](#), or switching to giving access using NAT instead. In other cases we've seen, especially on Windows Server platforms, the built-in Windows Firewall could be blocking queries coming from a subnet outside of the local network. In such a case an adjustment to the firewall is necessary to allow the DNS server to receive the query and respond to it.

5.2.4 Clients cannot access the Internet through Access Server

This may be caused by the DNS settings. When a problem occurs with redirecting VPN client Internet traffic, the most common issue is that domain names are not being resolved to IP addresses by a DNS server. To resolve this, you need to push a valid DNS server. If you don't know one, you can use Google's public DNS server. You can update the VPN Settings in the Admin UI to use Google's servers: 8.8.8.8 and 8.8.4.4. Then save settings and update the server.

5.2.5 Recovering damaged database configuration files

SQLite3 is fairly robust, but sometimes things happen like unexpected shutdowns or hardware problems with storage, or perhaps an incomplete write due to lack of disk space. Whatever the cause, once you encounter a problem with the database files being unreadable, the following procedure may help to restore your data, minus perhaps one or two records that are simply unrecoverable.

When you see for example this error:

```
exception in AuthDelegateProplist: (DatabaseError) database disk image is malformed
```

This could indicate that the configuration database files have an issue.

By default OpenVPN Access Server uses SQLite3 database files.

5.2.5.1 Check the environment

You need to make sure that your server is stable, up-to-date, and that the hardware is not exhibiting problems with reading and writing data on the hard disk. To describe the full procedure to diagnose a server system goes beyond what this documentation website is intended for but you should be able to find suitable tools from your server or disk manufacturer, like memory testing programs, disk testing programs, hardware diagnostics, and so on, or basic tools like **badblocks** and **fsck** to figure out if there are any problems with the data storage. Also keep in mind that you need adequate free disk space. If your disk is full, problems can occur when writing data to the configuration files.

5.2.5.2 Stop Access Server and make backups

A logical step is to stop the Access Server and make backups.

Use the following commands on the command line interface, while logged on with root privileges:

```
service openvpnas stop
cd /usr/local/openvpn_as/etc/db/
mkdir backup
cp *.db ./backup
```

Now if anything happens during the database recovery steps, you can simply copy back the files in the **backup** subfolder and get things working as they were before you started the recovery process. This is a simplistic backup process that needs the Access Server to be stopped. If you want to [run live backups](#), see the documentation on our website elsewhere.

5.2.5.3 Run recovery process

This process uses **SQLite3** to read the contents of the database files, convert it to SQL commands, and dumps it straight into a new SQLite3 process to build up a new database. If the original database is undamaged this will result in a perfect replica of the original database, to a new database file. In the case of a damaged original database file, the SQLite3 program will try to read all the usable information it can and put that into a new database file. Anything it just cannot read at all will be skipped. With luck your damaged file only has one damaged record and you can then work on restoring whatever is broken after you have completed the procedure. For example if a single user account has been damaged then after this recovery procedure you can try to delete that one user, and add the user again, to restore it to normal functionality. This will have to be determined on a case-by-case basis.

The following commands repair all 4 database files. If you have only a problem with a specific file, you may skip the others. It is divided with divider lines to indicate the portions for each database file that you can repair.

Run the following commands while logged on as user with root privileges:

```
service openvpnas stop
cd /usr/local/openvpn_as/
./bin/sqlite3 ./etc/db/config.db .dump | ./bin/sqlite3 ./etc/db/config.db.new
mv ./etc/db/config.db ./etc/db/config.db.old
mv ./etc/db/config.db.new ./etc/db/config.db
./bin/sqlite3 ./etc/db/certs.db .dump | ./bin/sqlite3 ./etc/db/certs.db.new
mv ./etc/db/certs.db ./etc/db/certs.db.old
mv ./etc/db/certs.db.new ./etc/db/certs.db
./bin/sqlite3 ./etc/db/userprop.db .dump | ./bin/sqlite3
./etc/db/userprop.db.new
mv ./etc/db/userprop.db ./etc/db/userprop.db.old
mv ./etc/db/userprop.db.new ./etc/db/userprop.db
./bin/sqlite3 ./etc/db/log.db .dump | ./bin/sqlite3 ./etc/db/log.db.new
mv ./etc/db/log.db ./etc/db/log.db.old
mv ./etc/db/log.db.new ./etc/db/log.db
```

Now restart the Access Server service:

```
service openvpnas restart
```

Now test your Access Server and see if everything is working alright. If you encounter problems, you can restore the *.db.old files to their original names to get it back to how it was before you ran the above recovery procedure.

5.2.6 Troubleshooting License Activation

5.2.6.1 Troubleshoot Access Server license keys (BYOL)

Before you proceed, if you are encountering problems activating a license key after January 20th of 2019, please make sure your Access Server is either updated to at least version 2.6.1, or that the licensing server patch has been applied. There are changes that will have been implemented after that date requiring you to be either up to date or that you have the patch installed on your existing Access Server installation. Please see the following resources:

- [Important change to licensing system after January 20th of 2019](#)
- [Instructions on how to keep your Access Server up-to-date](#)
- [Instructions to patch licensing system on Access Servers older than 2.6.1](#)

All license keys sold for OpenVPN Access Server using the [BYOL licensing system](#) are single-activation and lock to the hardware and software properties that you installed the license key on. One of the more common reasons why a license key suddenly disappears is when the license key has simply expired. License keys you purchase from us have an expiration date. After that date, the license key will disappear and no longer unlock any additional allowed connections on your Access Server.

If you are using an Amazon AMI with a prelicensed amount of connections skip this section and check the [Amazon AMI licensing troubleshooting](#) instead.

Another common reason is that recent maintenance on your server has caused the hardware/software combination to change significantly enough for the licensing system to believe it is now running on a different server than the one the license key was originally activated on. This is our copy-protection system that prevents a license key from being used more than once. Even if you are using the OpenVPN Access Server on a virtual platform, moving the virtual machine from one hypervisor platform to another can cause the licensing system to see this hardware change and invalidate the license key. If you for example replace the network interface card on your server or perform a clean reinstall of your server operating system this can cause the license key to become invalid.

One other known cause of problems is if the operating system has run out of all available memory. This can happen on systems with very little memory or with a fairly large user base. One of the first things to go then is the licensing system. It's easy to rule this out as a cause; reboot the server. If the issue persists then try the next step.

You can use the command line licensing manager program to view the current state of the licensing system. On the command line as the **root** user you can use the commands below to see

which license keys are on your system and which of them are having problems, and why, and how many connections your server is currently licensed for.

View which license key files are present on your server's file system:

```
ls -la /usr/local/openvpn_as/etc/licenses/
```

Check the license manager tool to see any problems and how many connections your server is licensed for now:

```
/usr/local/openvpn_as/scripts/liman info
```

A sample output could look like this:

```
Manager: exception with license file /usr/local/openvpn_as/etc/licenses/ABCD-1234-EFGH-5678.lic:
  machine properties validation failed: verify fail: ABCD-1234-EFGH-5678
  [3:0:8]/mac=110/hd=000/cpu=110/pci=110/ino=110/iid=000 (LIC_VPROP)
Manager: exception with license file /usr/local/openvpn_as/etc/licenses/IJKL-0912-MNOP-3456.lic:
  license key ID is expired (LIC_KEY_EXP)
Manager: exception with license file /usr/local/openvpn_as/etc/licenses/QRST-7890-UVWX-1234.lic:
  signature verification failed (LIC_VERIFY)
INFO {'apc': False, 'concurrent_connections': 20}
```

As can be seen in the output above the license key **ABCD-1234-EFGH-5678** fails the machine properties validation. This means that the hardware specifics of the system that this license key was activated on are no longer the same as the system it is currently trying to run on. The system then considers this license key invalid and skips it. Also visible in the output above is the license key **IJKL-0912-MNOP-3456** which shows that it is expired. This means that this license key's expiration date has been reached and it is no longer valid for use. If you haven't renewed this key, you can do so on our website, or just buy a new license key. You'll then get a new license key that you can activate on your OpenVPN Access Server and it will then be licensed again. The last line in the output above shows that right now this server is registered for 20 simultaneous connections.

If your license key is not expired and shows the **machine properties validation failed** message while you believe this key should be valid for this system, and this key isn't in use on another system, [contact us on our support ticket system](#) and explain the problem you are seeing and mention the license key that you are having a problem with. We can then revoke the current key and issue a new replacement key.

If your license key shows the **signature verification failed** message, it means that the file itself is corrupt somehow. We rarely encounter this since the activation system is normally automatic and manages itself, and as such corruption to these files is not going to happen by Access Server itself. But if the file system is damaged or if these files are manually transferred from somewhere to this server, and something has gone wrong in this process, then it could explain why the file is corrupted. If you for example did an offline activation procedure and copied this file onto this

server, try obtaining a new copy of the file and trying again. Avoid copying and pasting the contents of the file and instead use a tool like SCP or WinSCP. If you received the file as an email attachment try obtaining it via another method, for example if you requested us to do an offline activation, try logging in to the support ticket system website directly and downloading the file there, and even try another browser in case that still fails.

5.2.6.2 License key activation troubleshooting

It is reasonably rare but license key activation errors can occur, and this is usually a problem with the environment the Access Server is running on. When you activate a license key via the Admin UI, or activate a license key via the command line, it is sent to our license activation system on the Internet. If that connection is not possible for whatever reason, then activation will fail. Also, if the license key has already been used before for activation, then it will also fail. If you want to move a license key from one server to another, contact us and request a license key reissue.

Activation occurs online. License keys are activated by communicating with our licensing server at licensing.openvpn.net or licserv.openvpn.net on port TCP 443. It is not possible to do activation through a proxy server. It must be done directly. If you have a firewall that blocks traffic, kindly make an exception for this server and port. The IP is reasonably static, we haven't changed it in years. But it could theoretically change in the future, although we do not expect this to happen anytime soon. The IP address is 54.183.149.72.

If you activate a license key and you see the message **Fault 9000: "twisted.internet.error.DNSLookupError: DNS lookup failed: address 'licensing.openvpn.net' not found: [Errno -2] Name or service not known."** then you're definitely dealing with a DNS issue. This can for example be caused by not having DNS servers configured, or the ones that are configured, are internal DNS servers that only handle an internal DNS zone and don't have a clue about outside zones on the Internet. It could even be a temporary problem with the DNS server.

If you activate a license key and you see the message **SESSION ERROR: SESSION: Your session has expired, please reauthenticate (9007)** or then one of the most likely explanations is that either the DNS settings are still somehow wrong in the operating system that the Access Server is installed on, or that Internet access to the licensing server was not possible for some reason or another. Now if you know for a fact that your Access Server does not have Internet access, and will not get any Internet access either, because that's how you set things up and that's your intention, then you can skip ahead and look at the [offline activation steps](#) since the troubleshooting steps below won't be able to fix your issue.

If you activate a license key and you see the message **<Fault 9000: "OpenSSL.SSL.Error: [('SSL routines', 'ssl3_get_server_certificate', 'certificate verify failed')]">** then for some reason the secure connection between your Access Server and our licensing server is failing. A possible explanation could be a firewall or proxy system that intercepts the traffic and presents its own SSL certificate. This won't match with the certificate that the Access Server is expecting to see and so the verification of the certificate fails. Another possibility is that your server's time and date are off quite badly. The certificate we use on our licensing server is valid within specific

dates, and if your server has a date set that is wildly off, then the verification fails. To correct such a problem set the date correctly and consider installing an NTP client. The commands below assume you are root on a Debian/Ubuntu type server:

Check the current date setting:

```
timedatectl
```

If the timezone is wrong, correct it:

```
dpkg-reconfigure tzdata
```

If the date or time are wrong, correct it:

```
date --set="25 DEC 2018 20:10:00"
```

If you don't have a network time protocol client, install one:

```
apt-get update  
apt-get install ntp
```

Now if your Access Server does have Internet access, and the date and time are correct, and you're sure your key has not already been activated before (you can check this on our website in your licensing overview) but you still can't activate your license key then continue with the troubleshooting steps. One of the first things to check is if you can ping a public address like google.com from the Access Server's operating system. So log on to the console or an SSH session to the Access Server and obtain root privileges.

Ping google.com:

```
ping -w 1 google.com
```

You should be seeing a result like this:

```
PING www.google.com (216.58.211.100) 56(84) bytes of data.  
64 bytes from ams15s32-in-f4.1e100.net (216.58.211.100): icmp_seq=1 ttl=56  
time=7.37 ms  
--- www.google.com ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 7.376/7.376/7.376/0.000 ms
```

If you're not seeing it resolve the address to an IP address, then your DNS server settings need work. If you are seeing it resolve but you get 100% packet loss and no ping reply then the issue could be related to Internet connectivity. Like for example a bad gateway setting or a firewall blocking Internet access. To solve a problem with the DNS servers you could edit the `/etc/resolv.conf` file and put a DNS server address in there manually. But doing that when your server is set for a dynamically obtained IP address (DHCP) will mean that it gets reset every time your DHCP client program in your operating system retrieves new DNS server

information. So if you use DHCP you may want to switch to [setting a static IP address on your Access Server](#) and then editing the file `/etc/resolv.conf` to set a proper DNS address.

If you don't know a DNS server to use, you can use Google's public DNS servers like 8.8.8.8 and 8.8.4.4. To configure this in the `resolv.conf` file open the file in a text editor and make changes to it.

Edit `/etc/resolv.conf` in nano text editor:

```
nano /etc/resolv.conf
```

Find any lines that start with "nameserver" and change them to look like this (if none exist, add them):

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Press `ctrl+x`, then press `y`, and then press `enter`, to save and exit the file.

The changes should take effect immediately. Test again if you can now ping www.google.com. If you now can and previously you couldn't then it seems your DNS problem is now resolved. You can now attempt activation of the license key. If it still not resolved there is one final thing you can try. Some providers, especially in a country like China, do DNS poisoning, which results in false IP address results for a DNS query, thus blocking the activation process. By editing the local hosts file, which is a file with host names and IP addresses that will be referenced before asking a DNS server, it is possible to manually specify which IP address to use when contacting our licensing server. To do so follow the commands below.

Edit `/etc/hosts` in nano text editor:

```
nano /etc/hosts
```

Go to the bottom of the file and add this line:

```
54.183.149.72          licserv.openvpn.net licensing.openvpn.net
```

Press `ctrl+x`, then press `y`, and then press `enter`, to save and exit the file.

You can now attempt activation of the license key again.

As one of the final steps to try, you should try to activate the license key via the command line:

```
/usr/local/openvpn_as/scripts/liman activate "LICE-NSEK-EYIN-HERE"
```

If you activate a license key via the command line method, and you see the message **unable to get local issuer certificate** followed by a subject name of the certificate that is different from the expected name **OpenVPN Licensing CA** then you have some sort of a firewall or proxy server between your Access Server and our licensing system that is intercepting or blocking the traffic,

and may be trying to show an error message relating to this blockade. Try the following command to determine if your Access Server can reach the licensing system:

```
wget -O- -q --no-check-certificate https://licensing.openvpn.net/ | grep "XML_PARSE"
```

You should be seeing this expected output:

```
<value><string>XML_PARSE: error parsing XML</string></value>
```

If you didn't see this, try this instead:

```
wget -O- --no-check-certificate https://licensing.openvpn.net/
```

If you now see HTML code that starts with `<HTML>` `<HEAD>` and so on, you are looking at a webpage that is not generated by our licensing system, but by some system standing in the way between your Access Server and our licensing system, most likely a local firewall system that blocks proxies/anonymizers. Since OpenVPN Access Server could be considered to fall in such a category, you may need to go into the settings of your local firewall system and lift that restriction so Access Server can function normally. If you receive some error message, for some other unknown reason, the licensing system is not reachable from your Access Server.

Cisco's Umbrella solution has the categories **Proxy/Anonymizer** and **Software/Technology** that the entire openvpn.net domains falls under. If you have that solution set to block those categories, then please unblock them or manually add the hosts entries to your hosts file to bypass the restriction, since this is most commonly only a DNS-based poisoning/redirection and not a DPI-based blockade.

If you have not been able to activate your license key with the steps above, consider doing an offline activation. The offline activation procedure is explained below. You can also contact us and explain your situation by [contacting us on our support ticket system](#).

5.2.6.3 Offline BYOL license activation procedure

In the event that an online activation is impossible, either due to very strict firewalls that you have no control over, or because of the fact that the Access Server installation is located in a purely local network without any Internet access, then an offline activation may be done instead. There are two possibilities. You can either do the offline activation yourself using a second (temporary) Access Server installation that does have a connection to the Internet, or you can relay the required hardware information file to us through our [support ticket system](#), together with the license key you want to activate, and let us do the offline activation for you. After the offline activation procedure you will end up with an activated `xxxx-xxxx-xxxx-xxxx.lic` file which must be placed back on your server for the license key activation procedure to be completed.

Option 1: using a (temporary) Access Server installation with normal Internet access

The activation process reads a number of unique machine facts from the system that your Access

Server is installed on, and uses it together with your license key to activate and lock the license key to your system. It unlocks the amount of connections that your license key is good for, and locks it to the system you activated the license key on. With an offline activation procedure, we take the machine facts from one offline (no Internet access) Access Server, export it to a text file, and copy that text file to another online (with Internet access) Access Server, and then use that Access Server to do to the activation process. The resulting license file can then be copied back to the original machine and will work there. The procedure below describes how to do this:

- Log on to the Access Server that you wish to activate, let's call this the **production** server. We'll need the hardware specifics from this server.
- Run this command on the production server as root user:

```
/usr/local/openvpn_as/scripts/liman id-marker >licinfo.txt
```

- **licinfo.txt** is a text file that now contains the hardware specifics for that production server that needs to be activated.
- Copy this file to an Access Server that has access to our licensing server, let's call this the **activation** server.
Please make sure you do not copy/paste the contents of the file, but use a tool like SCP or WinSCP to actually transfer the file itself.
- Run this command on the activation server as root user:

```
/usr/local/openvpn_as/scripts/liman -i licinfo.txt Activate "LICE-NSEK-EYIN-HERE"
```

- Go to the **/usr/local/openvpn_as/etc/licenses/** folder and copy the file LICE-NSEK-EYIN-HERE.lic file from the activation server to the production server.
Again, please make sure you do not copy/paste the contents of the file, but use a tool like SCP or WinSCP to actually transfer the file itself.
- The production server should now see the activated key and update the concurrent connection count.

In rare cases you may need to use **service openvpnas restart** to restart the Access Server service to read the new license key.

Option 2: request us to do the offline activation for you with your licinfo.txt file

If you do not have the opportunity to set up a second Access Server purely for the activation steps, then you may also [contact us on our support ticket system](#) and explain that you need an offline activation. Please be sure that you provide us with the file licinfo.txt as an attachment to the support ticket, along with the license key that you wish to activate on your server. We can then take care of your request. You will need to follow the steps below to obtain the **licinfo.txt** file that we will need:

- Log on to the Access Server that you wish to activate, let's call this the **production** server. We'll need the hardware specifics from this server.
- Run this command on the production server as root user:

```
/usr/local/openvpn_as/scripts/liman id-marker >licinfo.txt
```

- **licinfo.txt** is a text file that now contains the hardware specifics for that production server that needs to be activated.
- Copy this file to your computer that you are using to submit a support ticket request, and attach the file to the ticket.

Please make sure you do not copy/paste the contents of the file, but use a tool like SCP or WinSCP to actually transfer the file itself.

5.2.6.4 Licensing problems with Amazon AWS tiered instances

If you encounter the problem where an OpenVPN Access Server with x amount of connected devices using the [Amazon AWS tiered instance licensing model](#) is showing you that your server is only licensed for 2 connections, while you launched an instance for “xx connected devices”, then the most likely explanation here is that you are using a security group on this instance that is blocking access to the licensing servers. If that happens the OpenVPN Access Server cannot check to see if you are licensed and will fall back to its automatic built-in demonstration mode which allows all functionality without time limit, but allows only 2 simultaneous VPN connections.

Please note that this is not the same licensing system as the BYOL licensing model that uses separate license keys for activation. For that we have a separate [troubleshooting section for BYOL licensing](#) above on this page. If you have launched an Amazon AMI that has in its title “xx connected devices” then you are indeed using the Amazon AWS tiered instance licensing model and you should investigate why your system is not getting access to the licensing systems. These are the addresses that the licensing system will need contact to for the tiered instances to verify the licensed state and unlock the amount of connections stated on the OpenVPN Access Server AWS tiered instance type:

IP address 169.254.169.254, port 80:
<http://169.254.169.254/latest/meta-data/>

These DNS names with wide dynamic IP ranges, on port TCP 443:
awspc1.openvpn.net
awspc2.openvpn.net

And these DNS names with static IP addresses, on port TCP 443:
awspc3.openvpn.net, IP address: 107.191.99.82
awspc4.openvpn.net, IP address: 107.161.19.201

Important note: awspc3.openvpn.net and awspc4.openvpn.net are only supported as of Access Server 2.5. Previous versions only use awspc1 and awspc2.

If you are strict on your security permissions, then you need to release access to the meta data system mentioned above, and at least one of the two static IP addresses of awspc3 or awspc4 mentioned above. The licensing system in the Access Server is designed to try a specific licensing server first, and if that fails, move on to the next, and so on, until all 4 addresses have

been tried. As a result, if you only unblock for example awspc4 then it may be a minute or two before it picks up the license after the server has just started up, so please be patient.

For those curious, awspc3 will be tried first, then 2, then 4, then 1.

If you have unblocked these addresses, and are still experiencing problems, we recommend first temporarily unblocking everything on this particular system. To put it simply; to disable anything that can possibly block any type of connections. Be sure to check both iptables firewalls and security groups in Amazon, both of these can block traffic. The first thing to ensure is that neither of these are possibly blocking the traffic. And of course do a reboot of the system to be sure any transient issues are taken care of. Once this has been done, and there are still issues, then contact us please with any details you can provide so we can investigate the problem.

DNS can be a problem if you block it. But you can either resolve that by manually entering the names awspc3.openvpn.net and awspc4.openvpn.net with the IP address information shown above into the local hosts file, so resolution of those names can occur locally, or to allow DNS requests to go out normally to your DNS server.

If it is absolutely required by company policy that no external contact of any kind to the addresses mentioned above must be possible for your AWS instance, then the tiered instances are not suitable as they do need access to at least the meta data server and a licensing server. The [BYOL licensing type](#) may be suitable instead in this case if an offline activation is performed and no auto-scaling or instance type alterations are used that alter the virtual hardware and possibly break the locked license of the BYOL license model.

To further investigate the problems with the AWS tiered instances licensing system it can help to activate a special debug flag in **as.conf** and restarting the Access Server service.

The **/var/log/openvpnas.log** file will then log information specific to the AWS licensing system, and any errors mentioned in there may aid in understanding and fixing what is wrong. Providing such information when you are contacting us for support would be of tremendous help to us in resolving the problem quickly. To enable debugging follow the steps below.

Open as.conf in nano text editor:

```
nano /usr/local/openvpn_as/etc/as.conf
```

Go to the bottom of the file and add this line:

```
DEBUG_AWSINFO=1
```

Now restart the Access Server service so that the changes can take effect:

```
service openvpnas restart
```

After reboot run this command to filter for the words "AWS INFO" in the log file:

```
cat /var/log/openvpnas.log | grep -i "AWS INFO"
```

If you see lines like these in `/var/log/openvpn.log`, the meta data server was unreachable:

```
2017-10-04 19:32:30+0200 [Uninitialized] AWS INFO: error getting instance
info 'doc': : An error occurred while connecting: 113: No route to host.
(twisted.internet.error.ConnectError)
2017-10-04 19:32:30+0200 [Uninitialized] AWS INFO: error getting instance
info 'sig': : An error occurred while connecting: 113: No route to host.
(twisted.internet.error.ConnectError)
2017-10-04 19:32:30+0200 [Uninitialized] AWS INFO: error getting instance
info 'pc': : An error occurred while connecting: 113: No route to host.
(twisted.internet.error.ConnectError)
2017-10-04 19:32:30+0200 [Uninitialized] AWS not detected
2017-10-04 19:32:33+0200 [-] AWS INFO: error getting instance ID: 'NoneType'
object has no attribute '__getitem__': aws/info:271 (exceptions.TypeError)
2017-10-04 19:32:33+0200 [-] AWS INFO: error getting instance ID: 'NoneType'
object has no attribute '__getitem__': aws/info:271 (exceptions.TypeError)
```

You should be seeing a fair amount of debug information. You can attempt to make sense of this yourself or send it to us on the support ticket system so we can analyze it for you.

5.2.6.4.1 Known errors and possible solutions with Amazon AWS tiered instance licensing

This problem occurs when DNS resolution fails, so check your DNS setup or use a hosts file:

```
2018-08-28 16:33:39+0000 [twisted.names.dns.DNSDatagramProtocol (UDP)]
AWS INFO: error in product code validation, will retry in 30 seconds:
<twisted.names.dns.Message instance at 0x7fed9370e950>:
aws/info:202 (twisted.names.error.DNSServer Error)
```

5.3 Backups & Recovery

5.3.1 AWS Backup

AWS Backup automates backup and recovery jobs for Amazon EC2 instances without the need for custom scripts or third-party solutions, saving time and simplifying the backup process. Customers that use EC2 instances will now be able to perform their data protection requirements at the EC2 level, backing up both the Amazon Machine Instance (AMI) and the attached Amazon Elastic Block Store (EBS) volumes. You can now select an EC2 instance from the AWS Backup console, take an on-demand backup, or assign EC2 instances to a backup plan.

Additionally, AWS Backup will enable customers to restore an EC2 instance with the same configuration as the original, greatly simplifying the recovery process. You can restore from the AWS Backup console, SDK, or CLI, all at the EC2 instance level.

For more guidance on how to back up and recover EC2 instances, please see the AWS Backup documentation, [here](#).

[AWS Backup](#) offers a centralized, fully-managed, policy-based service to back up data across AWS services. With AWS Backup, you can centrally configure backup policies and monitor backup activity for AWS resources, including Amazon EBS volumes, Amazon Relational

Database Service (RDS) databases, Amazon DynamoDB tables, Amazon Elastic File System(EFS), Amazon EC2 instances and AWS Storage Gateway volumes.

For more information on where AWS Backup is available, see the [AWS region table](#). To learn more about AWS Backup, please see our [product page](#) and [documentation](#).

5.3.2 Access Server Configuration Backup

The Access Server uses 4 databases in versions older than 2.6.1, before clustering feature was introduced and 8 databases in releases after. There is also a text file to store connectivity details to databases. It is possible to modify these configurations via the **admin web UI** interface, **sacli** command line tool, **confdba** command line tool, and **sqlite/mysql** command line utilities. Changing settings should normally always be done either in the admin web UI or via the sacli command line tool. The **confdba** and **sqlite/mysql** programs should only be used in the event the Access Server has a program starting up and the admin web UI and sacli command line tools are unavailable.

All of the configuration of the OpenVPN Access Server is by default stored in these files, in a standard single server implementation. Note that it is possible to change the authentication backend to a database server like MariaDB or MySQL, and that if you have done this, then things are different for you, of course.

- /usr/local/openvpn_as/etc/db/config.db
- /usr/local/openvpn_as/etc/db/certs.db
- /usr/local/openvpn_as/etc/db/userprop.db
- /usr/local/openvpn_as/etc/db/log.db
- /usr/local/openvpn_as/etc/as.conf

These were added since Access Server 2.6.1:

- /usr/local/openvpn_as/etc/db/config_local.db
- /usr/local/openvpn_as/etc/db/cluster.db
- /usr/local/openvpn_as/etc/db/clusterdb.db
- /usr/local/oevpn_as/etc/db/notification.db

The as.conf file is a simple text file that is the same on all OpenVPN Access Server installations unless you have altered something here, like for example if you disabled client certificate authentication. The .db files are of the SQLite3 database type. It is possible to switch to another database backend if this is required for whatever reason. A database backend like for example MySQL or MariaDB is suitable for this purpose. By default though the SQLite3 database type is what Access Server uses to store all of its configuration settings, certificates, user specific properties, and a log database that contains entries about who logged in when, how long they were connected, and how much bandwidth they used. The log database can be queried with a separate **logdba** tool designed for pulling this information out of the database file.

If you are using local authentication mode, the user passwords are also stored in these configuration files. If you use PAM authentication mode, the user specific properties like auto-login privilege and static IP address and such are stored in the configuration files, but the password for the user is stored in the operating system. That is something that is not part of the OpenVPN Access Server configuration, but lies outside of it. Worst case what happens if you backup and restore this configuration to a new installation of Access Server on another server is that you need to set passwords for these users again if you use PAM authentication. If you use LDAP or RADIUS, then the passwords are stored in there and will remain the same.

We recommend that you set up an automatic backup system that saves the entire configuration of your Access Server regularly, preferably via an automated task. If you follow the steps laid out here you'll end up with SQLite3 DB dump files that you can use to restore your server. If you are using another database backend then please see that database backend documentation for instructions on how to make backups on that database backend. The default is that all configuration is stored in SQLite3 database files.

If at any point the configuration becomes lost, then all the currently installed OpenVPN clients are unable to connect to the server. Even reinstalling the server with the same user names and passwords will then simply not have any effect. Every installation of OpenVPN Access Server comes with a unique private key and public key, which are used internally in the certificate management system built into the Access Server to generate unique client certificates. The certificates for one installation of OpenVPN Access Server are not compatible with that of another installation. So in the event of a server crash, if you have a backup of all of the configuration files, you can restore this and get your clients connected again without requiring them all to reinstall their clients or connection profiles.

It is possible to stop the OpenVPN Access Server service and then copy the 5 files to a safe location, and to then start the OpenVPN Access Server service again. But that adds an interruption you may not want. Therefore instead we recommend you use the steps described below, which let the service stay online while you make a backup.

With the commands below you can make a backup of all these files while the OpenVPN Access Server is live. That means you do not need to stop the Access Server service to make a backup file; it can continue running while a backup is made. This way you can easily create an automated task, like a cron job, to handle the backup task unattended. It goes without saying of course that if anyone gets a hold of these backup files, the security of your VPN system is compromised; it contains all the settings and certificates. So please do pay attention to where you store these backups and how you store them.

For backing up Access Server:

```
apt -y install sqlite3
yum -y install sqlite3
cd /usr/local/openvpn_as/etc/db
[ -e config.db ]&&sqlite3 config.db .dump>../../../../config.db.bak
[ -e certs.db ]&&sqlite3 certs.db .dump>../../../../certs.db.bak
[ -e userprop.db ]&&sqlite3 userprop.db .dump>../../../../userprop.db.bak
[ -e log.db ]&&sqlite3 log.db .dump>../../../../log.db.bak
```



```

[ -e config_local.db ]&&sqlite3 config_local.db
.dump>../../../../config_local.db.bak
[ -e cluster.db ]&&sqlite3 cluster.db .dump>../../../../cluster.db.bak
[ -e clusterdb.db ]&&sqlite3 clusterdb.db .dump>../../../../clusterdb.db.bak
[ -e notification.db ]&&sqlite3 notification.db
.dump>../../../../notification.db.bak
cp ../as.conf ../../as.conf.bak

```

The resulting backup files ending in .bak can be found in the `/usr/local/openvpn_as/` directory now and contain everything unique about this OpenVPN Access Server installation. It's worth noting that with PAM authentication system the passwords are stored in the operating system and these are not backed up with these commands, while with local authentication mode they are stored in these backup files. And with LDAP and RADIUS the password are stored in those systems and thus are not involved.

5.3.3 Recovering Access Server configuration from backup

While making a backup can be done live, restoring a backup must never be done live. This can lead to a damaged configuration, after which you'll have to restore the backup again to fix that. So let's assume for the moment that you have just suffered an unfortunate problem with your server and you had to reinstall on the same or different hardware, or on a new setup entirely, and you now wish to restore the configuration backups you have so wisely created and have available. If you have backups created using the **SQLite3 .dump** command as is demonstrated in the [backing up the OpenVPN Access Server configuration](#) section, then you can use the following commands to restore a configuration to a freshly installed OpenVPN Access Server installation. Please note that if you follow these steps, the current configuration of the OpenVPN Access Server will be wiped out completely and will be replaced with the contents of the backup files instead. There is no way to "combine" a backup from one server with another production server. We are assuming in the commands below that the backup files are in the `/usr/local/openvpn_as/` directory, but you can adjust the commands as necessary of course.

Use the commands below to wipe current configuration and restore the backup files you provide:

```

service openvpnas stop
apt -y install sqlite3
yum -y install sqlite3
cd /usr/local/openvpn_as/
rm ./etc/db/config.db ./etc/db/certs.db ./etc/db/userprop.db ./etc/db/log.db
rm ./etc/db/config_local.db ./etc/db/cluster.db ./etc/db/clusterdb.db
rm ./etc/db/notification.db ./etc/as.conf
[ -e config.db.bak ]&&sqlite3 <./config.db.bak ./etc/db/config.db
[ -e certs.db.bak ]&&sqlite3 <./certs.db.bak ./etc/db/certs.db
[ -e userprop.db.bak ]&&sqlite3 <./userprop.db.bak ./etc/db/userprop.db
[ -e log.db.bak ]&&sqlite3 <./log.db.bak ./etc/db/log.db
[ -e config_local.db.bak ]&&sqlite3 <./config_local.db.bak
./etc/db/config_local.db
[ -e cluster.db.bak ]&&sqlite3 <./cluster.db.bak ./etc/db/cluster.db
[ -e clusterdb.db.bak ]&&sqlite3 <./clusterdb.db.bak ./etc/db/clusterdb.db
[ -e notification.db.bak ]&&sqlite3 <./notification.db.bak
./etc/db/notification.db
[ -e as.conf.bak ]&&cp ./as.conf.bak ./etc/as.conf

```

```
service openvpnas start
```

This restores the configuration backup. There are a few caveats to note here:

If you restore a configuration backup to another server, it's possible that you had your system configured a specific way that doesn't work on the new server installation anymore. Perhaps the IP address was different on your old server, and perhaps you had chosen to set the Access Server to only listen to a very specific IP address. If that address is then not present on your new installation, the web interface won't respond because it's set to listen to an address or interface that doesn't exist. To resolve this check the section on how to [reset the interface and port for the web services to listen on to default settings](#). Once you have access to the Admin UI again you can reconfigure it to whatever settings you wish. Alternatively, you can use the [command line tools to configure the web server settings](#) manually.

And finally, there is one more thing to check. The configuration database also contains the setting on how many TCP daemons and UDP daemons to launch. If this is set higher than the number of available CPU cores, the Access Server program may become unstable. So if you have restored this configuration on a different server, and the amount of CPU cores is different from the server the configuration backup came from, you should adjust this as described on the Server Network Settings page in the Admin UI, or use our [reset commands for the OpenVPN daemons](#) here.

5.3.4 Backing up and recovering SSL certificates

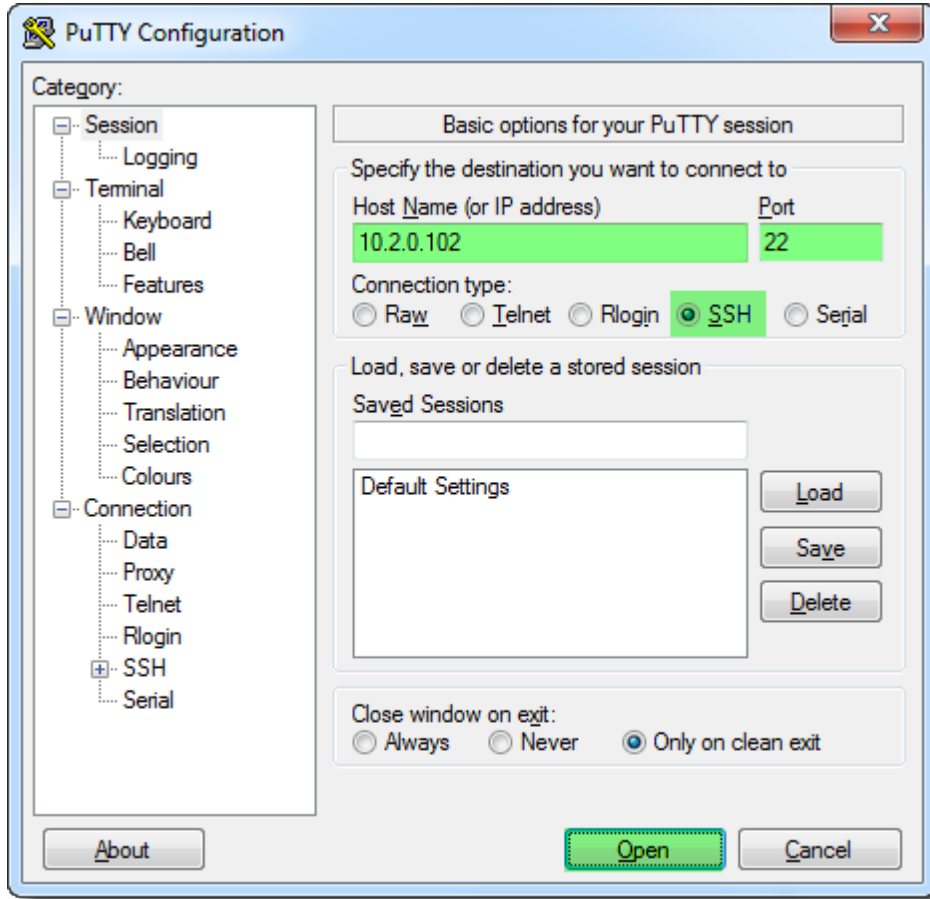
It has happened on occasion that people have installed an SSL web certificate on their Access Server, and that they needed to pull these files back out of the Access Server. For example, in the case of a wildcard certificate and you want to use that same SSL certificate for another server. Or just to make a backup in case the original files are lost, or when you want to transfer the SSL certificate to a new installation. Because Access Server stores the files in the configuration database, which is in SQLite3 format, it may be a little difficult to retrieve the original files. With the guide here, however, it is possible to pull the data out of the configuration database and store it in separate files again. Please note that this guide is only of any use to people that have already installed a commercial SSL certificate with private key and intermediary (CA) bundle files, and wish to recover these from the Access Server.

We're assuming you're on a Windows system. If you're on another system the connection program and steps may be different but the commands to extract the data are the same on the server.

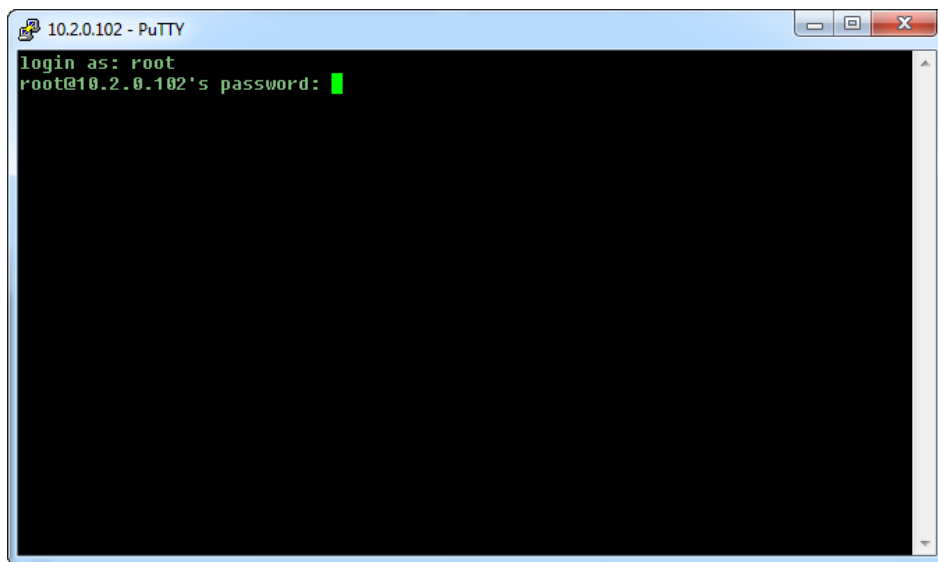
5.3.4.1 Backing up an already installed SSL certificate via CLI

In order to do so, you will need SSH access to your Access Server. To begin, you will need to launch a SSH client such as PuTTY to connect to your server using SSH. You will need root privileges to be able to do these tasks. If you are not able to log in as root directly but have to log in as another unprivileged user account, and then **sudo su** to get **root** privileges, then that's fine too.

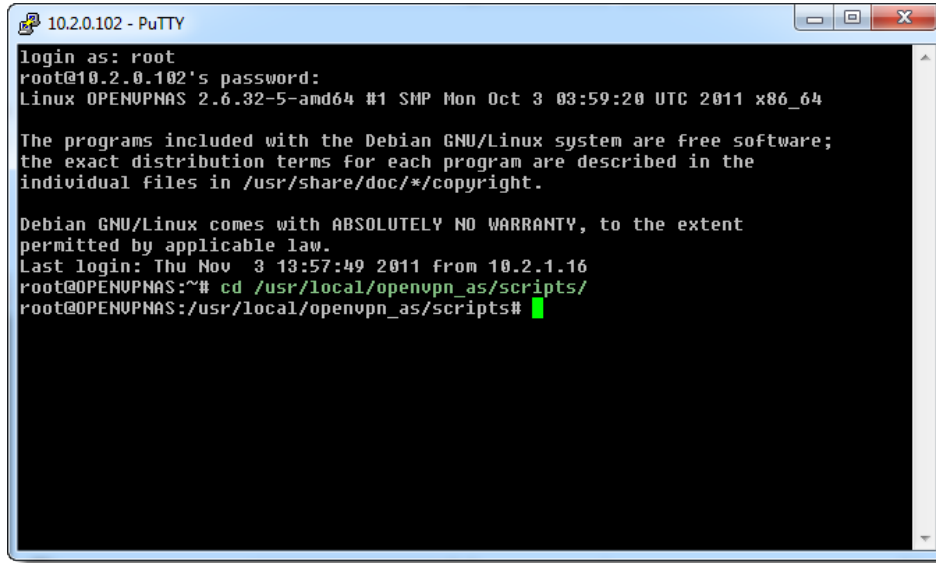
Start PuTTY and connect to the IP address of your server on port 22, SSH, and click 'Open'.



Enter the server's username and password. It must have root access. This is not the VPN client username!



Execute command: `cd /usr/local/openssl_as/scripts/`



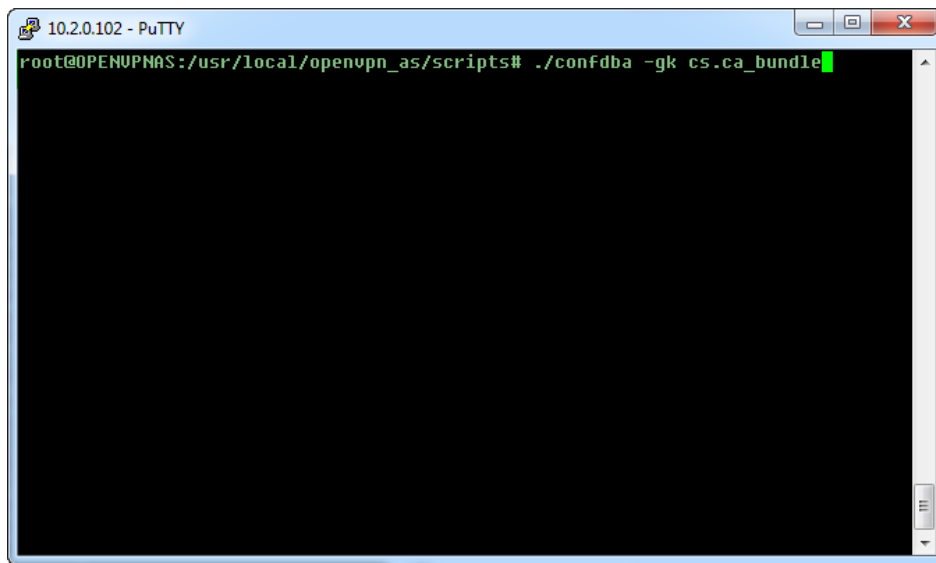
```
10.2.0.102 - PuTTY
login as: root
root@10.2.0.102's password:
Linux OPENVPNAS 2.6.32-5-amd64 #1 SMP Mon Oct 3 03:59:20 UTC 2011 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Nov  3 13:57:49 2011 from 10.2.1.16
root@OPENVPNAS:~# cd /usr/local/openssl_as/scripts/
root@OPENVPNAS:/usr/local/openssl_as/scripts#
```

5.3.4.2 Intermediary (CA) bundle file:

Execute command: `./confdba -gk cs.ca_bundle`



```
10.2.0.102 - PuTTY
root@OPENVPNAS:/usr/local/openssl_as/scripts# ./confdba -gk cs.ca_bundle
```

Scroll up, (if necessary), start selecting from BEGIN CERTIFICATE, and stop when you hit the last END CERTIFICATE.

```
10.2.0.102 - PuTTY
DYjUhuGuWFC8PA0HPnaZ3yx1RmGH5Ez2hNSENoc/10h9k78/Qn1Ujmx6xtCGj1txk
cFJCihFU6oyFumyJ/DjFoDHnaB2JBCMguWIoE50INmt58jw6HH0b7ozBXZNYdjnr
aAd27pjkTfI60F8jobr1NFsJmZXQCBzQTqNULNMeuuoU1WkD6Ff+d19kJCABHQkT
mxnUs8EPH2hcPwcAMU897UBS1MTDDvj5K680Qdfq211p0ItrrUkx10MihBFb6z5S
qGUCjI20ueiG/ALE+dJ5UbZvHkmWdK4GLX6oFzQa0W1uXUsxSDHTCnWvk7+TL12q
qopz1RM0EioIsE1uPmYUvh9F3Kx8J8UAcce39wphrP4X73R6WESN18A6uXMgnrJnjj
Re9HHY0U+wawcvNLxzE03FIxChQSmWeTg66Ak6L41jT/k8TpWNL0UoXJmPgPYLIw
c7GJCqaCWN2qSczohmPAgB10pd3EEx9qXkTDQForZ1Xh4wXFU10D7jgm0adsnbZr
FRUX5fbUwb9m6sFLvAa2T/9qGXamc5KeCLE7AYwS6rT7n57EYUqxgJxRunUwq/yN
KkgU20zU7poAF0Ux0asiXBHxDI5JL0518U1RFIzA9wIDAQABMA0GCsQGS1b3DQEB
BQUAA4ICAQB//y3AoaR3h1ouMaLUad4+rQURiMUrqog9873goZiis/nMewNjRghr
PKCN/gcXXNsK/E4hEaa+xQzTUHDoh7Hp40D7noRJ5S3c8QcciRegc8d8Lu2PECw3
+9bYbA2X2en6wdJk0hwGvMFWH/Gzbs5FT4BIA9DpbY34FotTqy5crAngw2me1o5Y
CvjByCm0ISHhIBKHizC9n6588m1S8o5aGyc6CoXQUU13Hc8yo6CbntXaudSP14rI
Ozy56oIrWKNu91UPpsBs1UrBJ9UXWmqowSAB6C2nmWnpIu9ZP2AA4gNPYP1TKb
yew8pre2z919xXkwI8JwSsc6QkMiגעgcyMFCabrDukbHsslQrwt2PQShy6oYajtj
NMRwsMFnQJxxSKF59wL649zW0k5UaSm58JbJR1eoY8CvnJK0JzicPk7BBR/bfDDA
E1RRzzTjptTKU02RrRrdMss9bF3g3Hfhr10UrwYFugNG0aGQ4bZsSbMrUNU6m36cJ
hXftZrJx1FFEJ0eche0tYzStU0IjTN1nRRkDMk241xGJixigU7Np4Q4Fu+iUTuz6
w7gg/dY2+/rpF1AesjyDbwntcLr06v1Q6/UqFZScUnnJcpWHCSGcQ3+GxJa6CZ3M
4rD3ao6dbi70BM6RFtGzsXq+xDS+RPUod28F0AUcwRm/cmQD31qzQ==
-----END CERTIFICATE-----
root@OPENUPNAS:/usr/local/openssl/scripts#
```

The CA bundle is now copied to the clipboard. Open up a text editor, paste the contents into the editor, and then save the file as **ca.pem**.

5.3.4.3 Private Key file:

Execute command: `./confdba -gk cs.priv_key`

```
10.2.0.102 - PuTTY
root@OPENUPNAS:/usr/local/openssl/scripts# ./confdba -gk cs.priv_key
```

Scroll up, (if necessary), start selecting from BEGIN RSA PRIVATE KEY, and stop when you hit END RSA PRIVATE KEY.

```
10.2.0.102 - PuTTY
jMPT3s0rK8pS+EUDA/dzITDcNUIjZpJA/K+gGBB6HrCFPnX8b6SdMzKRT/BGKb2H
0uXqpDm4Np/Ih2/MYD35G/k/B17K5SW1FgbQ9e2Fx5jka5IFUUBo84tE1popj6UP
xNhZy40PXk0sJXoo4X01ua7IGCqT/sERrEL0mHGPxqmZWIxzYmfYqzyhSgb8YbwU
DE7bS+eB3vWo4Yp6K7gKUewguRe16sHk/pod+1+/e/42FKyLmwY6BurQFY01ynss
L0oxiLS9CpMd7YG0YL3CA6Fgz4kpxeDtHrnJG+UYBwKCAQBRQjbf3Ye2HMUPUSdx
xWPQyvg2B9wPGmo3ooEU8y1W/NOhNZDeW2/Rg8i7Scm7FgQo9XpYFU7vk/WRrKBC
4MahCfhgPiZrNSZD/LU4kEkNR17jJua/MhGB8zo+UQA+Z6x1nYWP035ZRjPEJiUP
YzzDoFWeS1jLkeUG3jm95mGbbUWexIhc22te17U6jqJ9dAKgnR9FBYapwqFXZL+E
9Dhc0UFoJ+6DThCe1M3ACKDbxjy8I7ed8b2s1Wrsu8Bh6gb4xP3AxpWuRMKeW1+p
SJM0nUv1xBmXoWN4npz4euvwtQBm0LSeYiM/xDpPkTrbjAgj8462FPxM2U+gF8g/k
8IpxAoIBABp3ebn+gdePEskiDYJDFqr5mPYaor0yCRzrmVffNr17An65LLekoQuj
MUCUaRvW0God/Z+5+TuysP0kIsb1TH0RP7fepS5rTmJkvGmUm0Em3zUyQUzu6IX5
Gwn0y4pURE2B2k60w1qbo3Bi63APvAqo+B2UCtzzhEUqBkICqbmU0Y26s4gUyrnW
Fvk5qmy910MNAagJtRsit1c8GBQc0kCmNQTJ1+XnryPjP25T8vdc+k9dn8KwJ2wv
B1djWoCnbqUjWbv3ewgX5QEt6RinAzhbjgGmuu+MmULF0FDkuGIJokcG0bc/DZGH
dv20mUSWoxpSy79ZScn1a/my7eLUwYECggEBANzqSG07E9YB1RR5U20UDnAsFCg
UHAY1CgP5JfQMXdxtSNYgaD1TeSUL2L/gqovQ1gXW4JXu5b+4/1hKPNP+JsdGWBg
SnDI0NED01d91d6IH9bw0/aINz1Z2HexEi4Nar18N6a0y0Li+zqUE8QjioyPo4DQ
7mz2ZMDsnu6ieQ7Mfw38kA9sW1U45nA30uNnnouj1EiJXWpddbZ3GI+oX6+/UNAG
Gy1zhIoR1CJRG0xUESlpozxeBEcASKNHPJWD04rRHZ81jwyd81PzDf+cT1FIHQB
7dxAb0AIZDAXcsu78aeJ25ba62qop/osEU230FOXp+w26DaoU1H47aW/ZZA=
-----END RSA PRIVATE KEY-----
root@OPENUPNAS:/usr/local/openssl/scripts#
```

The Private Key is now copied to the clipboard. Open up a text editor, paste the contents into the editor, and then save the file as **server.key**.

5.3.4.4 Server certificate file:

Execute command: `./confdba -gk cs.cert`

```
10.2.0.102 - PuTTY
root@OPENUPNAS:/usr/local/openssl/scripts# ./confdba -gk cs.cert
```

Scroll up, (if necessary), start selecting from BEGIN CERTIFICATE, and stop when you hit END CERTIFICATE.

```
10.2.0.102 - PuTTY
DYjUhuGuWFC8PA0HPnaZ3yx1RmGH5Ez2hNSEnoc/10h9k78/Qn1Ujmx6xtCGj1txk
cFJCihFU6oyFumyJ/DjFoDHnaB2JBCMguWlOe501Nmt58jw6HH0b7ozBXZNYdjnr
aAd27pjktFI60F8jobr1NfsJmZXQCBzQTqNULNMeuuoU1WkD6Ff+d19kJCABHQKT
mxnUs8EPH2hcPwcAMU897UBS1MTDDvj5K680Qdfq211p0ItrrUkx10MihBF6z5S
qGUCjI20ueiG/ALE+dJ5UbZvHkmWdK4GLX6oFzQa0W1uXUsxSDHTCnWvk7+TL1Zq
qopz1RM0EioIsEluPmYUhh9F3Kx8J8UAcce39wphrP4X73R6WESN18A6uXMgnrJnjj
Re9HHY0U+wawcVNLxzE03F1XChQSmWeTg66Ak6L41jT/k8TpWNL0UoXJmPgPYLiw
c7GJCqaCWn2qSczohmPAgB10pd3EEx9qXkTDQForZ1Xh4wXFU10D7jgm0adsnbZr
FRUX5fbUwb9m6sFLvAa2T/9qGXamc5KeCle7AYwS6rT7n57EYUqxgJxRunUwq/yN
KkgUz0zU7poAF0Ux0asiXBHxDI5JL0518U1RF1zA9wIDAQABMA0GCsGCS1b3DQEB
BQUAA4ICAQB//y3AoaR3h1ouMaLUad4+rQURiMUrqog9873goZiis/nMewNjRghr
PKCN/gcXXNsK/E4hEaa+xQzTUHDoh7Hp40D7noRJ5S3c8QcciRegc8d8Lu2PECw3
+9bYbA2X2en6wdJk0hwGvMFwH/Gzbs5FT4BIA9DpbY34FotTqy5crAngw2me1o5Y
CvJyBcm0ISHhIBKHizC9n6588m1S8o5aGyC6CoXQUU13Hc8yo6CbntXaudSP14rI
Ozy56oIrWKNu91UPpsBs1UrBJ9UXWmqowSAB6C2nmWnpIu9ZP2AA4gNPYP1TKb
yew8pre2z919xXkwI8JwSsc6QkMigegcymFCabrDukbHsslQrwt2PQShy6oYajtj
NMRwsMFnQJxxSKF59L649zW0k5UaSm58JbJR1eoY8CvNJk0ZjicPk7BBR/bfDDA
E1RRzzTjptTKU02RrRdMss9bF3g3Hfhr10UrvWYFugNG0aGQ4bZsSbMrUNU6m36cJ
hXftZrJk1FFEJ0eche0tYzStU0IjTN1nRkkDMk241xGJixigU7Np4Q4Fu+iUTuz6
w7gg/dy2+/rpF1AesjyDbwntcLr06v1Q6/UqFZScUvnJcpWHCSGcQ3+GxJa6CZ3M
4rD3ao6bdB170BM6RF7GzsXq+xDS+RPUod28F0AUcwRm/cmQD31qzQ==
-----END CERTIFICATE-----
root@OPENUPNAS:/usr/local/openssl/scripts#
```

The Server Certificate is now copied to the clipboard. Open up a text editor, paste the contents into the editor, and then save the file as **server.crt**. You now have a backup of the files as they were submitted to the Access Server originally when the certificates were installed.

5.3.4.5 To install these files back onto an Access Server

You can follow the procedure via the Admin UI: [How to install an SSL certificate in Access Server via the Admin UI](#)

Or you can install via the command line interface: [How to install an SSL certificate in Access Server via the command line interface](#)

6 Support

Access Server and client programs that are developed and supported by the [OpenVPN Inc.](#) company with 24/7 support available via the online support ticket system.

6.1 Getting support for the OpenVPN Access Server

The commercial OpenVPN Access Server product has a dedicated support ticket system with professionals standing by 24/7 across the world to answer any questions you may have. To reach our support ticket system you need an account on our main website [openvpn.net](#). Making an account on our main website is free and gives you the option to purchase license keys online on our website, and to contact our support ticket system. Technically we only provide support for paying customers of our OpenVPN Access Server product, but generally don't mind answering questions for people trying out our product and needing some help to get it configured. And if you have any comments or suggestions regarding our website you're welcome to contact us here

as well. The open source version of OpenVPN is something we do not support here though. See the [getting support for the open source OpenVPN project](#) section for more information.

The recommended and guaranteed method of getting support for OpenVPN Access Server is to follow these steps:

1. Go to our website to [register for a free account](#).
2. Validate your email address and then [sign in to our website](#).
3. Once signed in click 'support' link at the top of the page and [open and submit a support ticket](#).

Once you have visited the support ticket system at least once with your registered account, the support ticket system will also accept emails you send to support@openvpn.net directly. Response times vary a bit depending on how busy things are and whether it's in the weekend or a holiday, or just an ordinary work day, but usually you will have a response within a few hours or sooner.

Additionally, there is a forum specifically for users of OpenVPN Access Server that contains a lot of useful information and lets you submit questions and receive answers there as well. The forum however is part of the open source community sites and while visited by OpenVPN Inc. personnel it is not as good as contacting us directly via the support ticket system. To visit the forum and read messages there you do not need an account. But if you want to participate you need a community account. This is a separate account from the main openvpn.net website.

To use and participate in the forum discussions follow these steps:

1. Go to the community website and [register for a free account](#).
2. [Sign in to the forum website](#)
3. [Visit the OpenVPN Access Server forum](#)

There is also an IRC channel on the Freenode network called #openvpn-as where power users lounge and can sometimes answer questions and help you out. This can sometimes be useful if you need a bit more guidance. Some official OpenVPN Inc. personnel members are also in this channel but are often idling and won't respond immediately if a question is asked. Again this option is not as good as contacting us directly via the support ticket system.

To join the IRC channel use the web chat client below and follow these steps:

- [Freenode IRC web chat client](#)
- Enter your name and validate the captcha and connect
- Ask your question in the channel and wait

6.2 OpenVPN Connect Client for Windows and macOS

These clients are part of the OpenVPN Access Server product, and as such, support for it is available there as well. See the section [getting support for the OpenVPN Access Server](#).

6.3 OpenVPN Connect for iOS and Android

The client available on the Apple App Store titled [OpenVPN Connect](#), and the client available on the Google Play Store titled [OpenVPN Connect](#), are the official OpenVPN clients developed and supported by the OpenVPN Inc. company. There are other clients available, and other related OpenVPN programs, but those other clients are not supported by the OpenVPN Inc. company.

The way to get support for OpenVPN Connect for iOS and Android is to follow these steps:

1. Go to our website to [register for a free account](#).
2. Validate your email address and then [sign in to our website](#).
3. Once signed in click 'support' link at the top of the page and [open and submit a support ticket](#).

Once you have visited the support ticket system at least once with your registered account, the support ticket system will also accept emails you send to ios@openvpn.net or android@openvpn.net directly. Response times vary a bit depending on how busy things are and whether it's in the weekend or a holiday, or just an ordinary work day, but usually you will have a response within a few hours or sooner.

7 Access Server Resources

- Access Server can be installed on Amazon Linux2 following instructions at our website <https://openvpn.net/vpn-software-packages/>
- Access Server Administration manual, demonstration videos, and other resources can be found at <https://openvpn.net/resource-center/>



7901 Stoneridge Drive, Suite 540
Pleasanton, CA
United States 94588
Sales: sales@openvpn.net

OpenVPN is a registered trademark of OpenVPN Inc.
All other marks mentioned herein may be trademarks of their
respective companies.

OVASWP1802v1