axway

SecureTransport

Version 5.4

2 April 2024

AWS Installation Guide

This documentation describes the following Axway software:

Axway SecureTransport 5.4

Revision history

The following changes are added to the SecureTransport 5.4 AWS Installation Guide:

SecureTransport version	Document revision number	Topics updated
5.4	5.4.01 – initial version	
5.4	5.4.02 – current version	New topic added: *Deploy MS SQL database in Amazon RDS* on page 45

Contents

Introduction 1

This document provides a detailed overview and detailed instructions to set up SecureTransport in the Amazon Web Services (AWS) Virtual Private Cloud (VPC).

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the Amazon Web Services (AWS) cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

Learn more about Amazon VPC.

About SecureTransport

SecureTransport is part of the Axway family of managed file transfer (MFT) products. SecureTransport allows organizations to adeptly control and manage the transfer of files inside and outside of the corporate firewall in support of mission-critical business processes, while satisfying policy and regulatory compliance requirements. SecureTransport serves as a hub and router for moving files between humans, systems and more. SecureTransport also completes tasks related to moving files (push or pull), hosting files in mailboxes or "FTP-like" folders, and provides portal access with configurable workflow for file handling and routing. SecureTransport delivers user-friendly governance and configuration capabilities, including delegated administration and pre-defined and configurable workflows, while providing the highest possible level of security.

For a complete description of SecureTransport features and components, refer to the *SecureTransport Administrator's Guide*.

SecureTransport in Amazon Virtual Private Cloud 2

Currently, deployment of SecureTransport on AWS VPC has been verified for Red Hat 7.4 or later implementations only in the following setup:

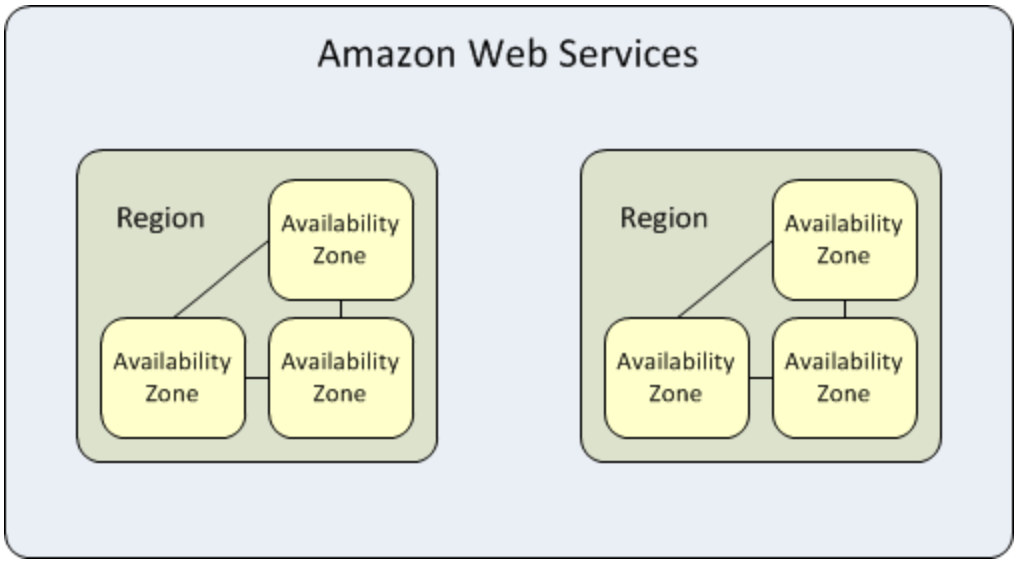
- Enterprise Cluster of two servers with streaming to two edges
- Oracle 12 Database Engine and Microsoft SQL Server 2017 Engine
- GlusterFS file system with two servers
- A Load Balancer (optional)
- A NAT Gateway (optional)
- An Administration Host (optional)
- Four private and two public subnets
- A VPN Connection (optional)
- Instances are assigned to five Security Groups
- All this located in two availability zones

Note For a multiple Availability Zone deployment, SecureTransport is supported in Enterprise Cluster mode only. SecureTransport over AWS is not supported in Standard Cluster mode when cluster nodes are deployed across multiple Availability zones. For a Standard Cluster setup in AWS, all nodes must be located in the same Availability Zone.

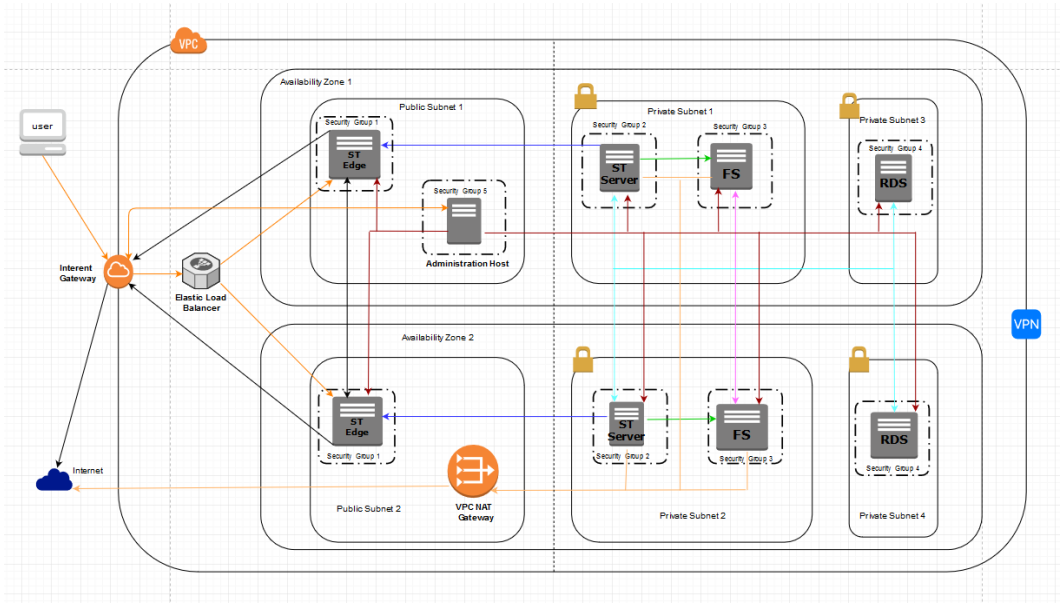
AWS cloud computing resources are housed in highly available data center facilities. To provide additional scalability and reliability, these data center facilities are distributed in different physical locations, categorized by *regions* and *Availability Zones*.

Regions are large and widely dispersed into separate geographic locations. Availability Zones are distinct locations within a region that are engineered to be isolated from failures in other Availability Zones and provide inexpensive, low latency network connectivity to other Availability Zones in the same region.

Each region is completely independent. Each Availability Zone is isolated, but the Availability Zones in a region are connected through low-latency links. The following diagram illustrates the relationship between regions and Availability Zones.



The goal is to provide a highly-available, fault-tolerant and secure setup of SecureTransport in the Amazon Web Services Cloud.



The diagram illustrates a SecureTransport setup of an Enterprise Cluster in streaming mode with two Edge servers, each deployed in a different Availability Zone. Amazon RDS service is used to set up the Oracle / MS SQL database depending on your setup. Amazon Relational Database Service (Amazon RDS) is a managed service that makes it easy to set up, operate, and scale a relational database in the cloud.

The setup is dispersed into two Availability Zones from one region in AWS. Each Availability zone contains public and private subnets within your Virtual Private Cloud. The public subnets host the SecureTransport Edge servers and the private subnets host the SecureTransport servers and the external file system. The database instances are located in separate private subnets in the two Availability Zones. The Edge servers are in a constant synchronization connection. The SecureTransport servers are in an Enterprise Cluster setup and in a constant synchronization

connection as well. The SecureTransport servers are connected in streaming with the Edge server as each server establishes streaming connections with both Edge servers. Internet-facing load balancer distributes requests to the Edge servers in the two Availability Zones.

In case of a system failure in one of the Availability Zones, the other zone remains fully functional with up to date system configuration. The SecureTransport Edge and server in the functional zone continue to process requests.

The public subnets in AWS Cloud are the equivalent of the DMZ (demilitarized zone) in a classic on-premise deployment. Amazon Web Services cloud provides security tools like Security Groups and Network Access Control Lists (ACLs) to protect the public and private subnets in your VPC. These tools act as the firewalls in the classic on-premise deployments and control both inbound and outbound traffic at an instance and subnet level. There is a Network ACL between each tier of the SecureTransport setup – public subnets, private subnets and private database subnets. The Network ACL is a stateless firewall that works at the subnet level. Each group (Administration host, Edge servers, Servers, File System, Database) and the Load Balancer have a corresponding security group. The security group is a stateful firewall that works at the unit level (EC2 instance, LoadBalancer, RDS).

There is a host placed in one of the public subnets called "Administration host" also known as "Bastion host" in the networking terminology. Bastion hosts are instances that typically reside within your public subnet and are usually accessed using SSH or RDP. Once remote connectivity is established with the bastion host, it then acts as a 'jump' server, allowing you to use SSH or RDP to login to other publicly inaccessible instances. When properly configured through the use of security groups and Network ACLs, the bastion host essentially acts as a bridge to your private instances via the Internet. The Administration host is used to perform maintenance and administration tasks on the servers in the SecureTransport setup. You should always consider the resiliency and high availability of your services in cloud deployments and the best practice is to have an Administration host in each availability zone in case one of the zones goes down. For minimal security risks, you should stop your Administration host instances for the duration of no maintenance work and start them when you need access to your servers in AWS again.

You can connect directly to your VPC using VPN in case you don't want to have an Administration host in your setup (see VPN section in this guide).

The installation of SecureTransport on Red Hat instances in Amazon Web Services (AWS) cloud follows the same flow as with the installation on a regular Red Hat machine, including the required installation prerequisites. This process has already been described in the SecureTransport Installation Guide.

You will pass through Amazon Web Services (AWS) cloud specific setup stages and configuration until you can proceed with your SecureTransport installation.

To set up SecureTransport in Amazon Web Services VPC, you must pass through the following steps:

1. Create a VPC.
2. Create Security Groups and Network Access Lists.
3. Create public and private subnets in two availability zones.
4. Internet Gateway and public subnets routing setup.
5. NAT Gateway and private subnets routing setup.
6. Amazon RDS service: Oracle or MS SQL database setup.

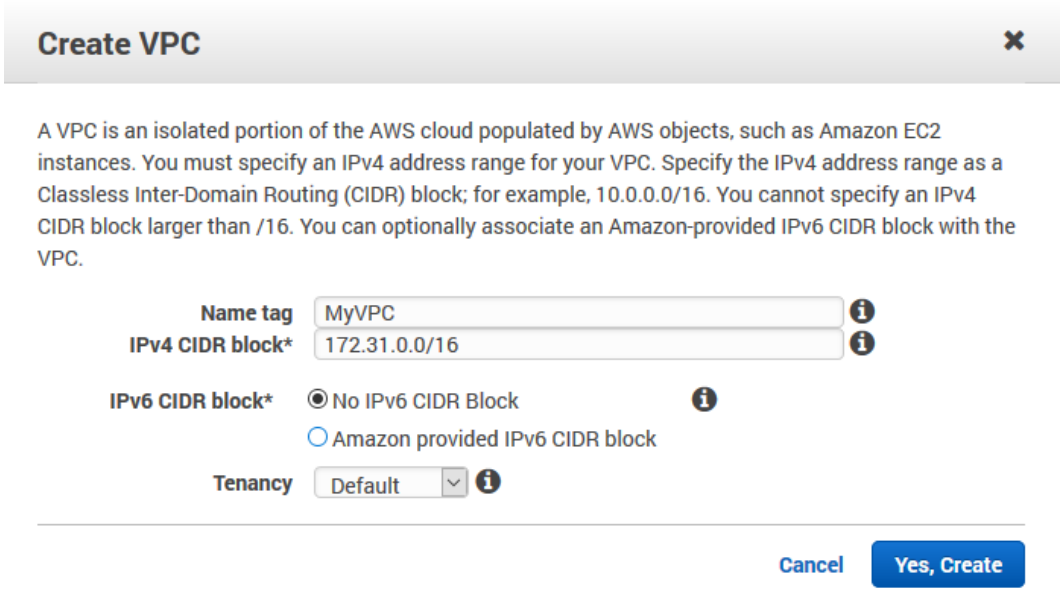
7. Amazon EC2 Red Hat instances launch.
    a. Launch an instance for an Administration Host.
    b. Launch instances in the public subnets for SecureTransport Edge installations.
    c. Launch instances in the private subnets for SecureTransport Server installation.
    d. Launch instances in the private subnets for external GlusterFS file system.
8. Establish VPN connection to your Amazon VPC.
9. Configure the SecureTransport Enterprise Cluster setup.
10. Set up Classic Load Balancer.

Create a VPC 3

To create a VPC, follow these steps:

1. Log in to AWS and navigate to the AWS console -> Services.
2. Under Networking & Content Delivery section, choose VPC.
3. Navigate to Your VPCs and then click Create VPC.
4. Fill in the settings and click Yes, Create.

Note This is an example CIDR block size. You can configure the CIDR block according to your needs.



Create Security Groups and Network Access Lists

4

Amazon VPC provides the following tools to help you increase the security of your VPC:

Security Groups on page 12 – these act as a firewall for associated Amazon EC2 instances, controlling both inbound and outbound traffic at the instance level.

Network Access Lists on page 23 – also called ACLs, these act as a firewall for associated subnets, controlling both inbound and outbound traffic at the subnet level.

Flow logs - Capture information about the IP traffic going to and from network interfaces in your VPC. AWS provides two features that you can use to increase security in your VPC: security groups and network ACLs. Security groups control inbound and outbound traffic for your instances, and network ACLs control inbound and outbound traffic for your subnets. For more information, see VPC Security.

For more details on security options, check the AWS Security Best Practices guide.

Note The suggested configurations with Security Groups and Network Access Lists as stated in the current guide do not include outbound rules. For more information on best practices for the outbound rules to apply for your setup, please refer to the SecureTransport Administrator's guide.

Security Groups

An EC2 instance is a virtual machine in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure.

When you launch an EC2 instance in a VPC, you can associate it with one or more security groups that you've created. Each instance in your VPC could belong to a different set of security groups. If you don't specify a security group when you launch an instance, the instance automatically belongs to the default security group for the VPC. For more information about security groups, see Security Groups for Your VPC.

We recommend you create the security groups you need for your SecureTransport infrastructure in AWS as a first stage before proceeding with rest of the setup. You need to group (assign) the instances and components into the following security groups:

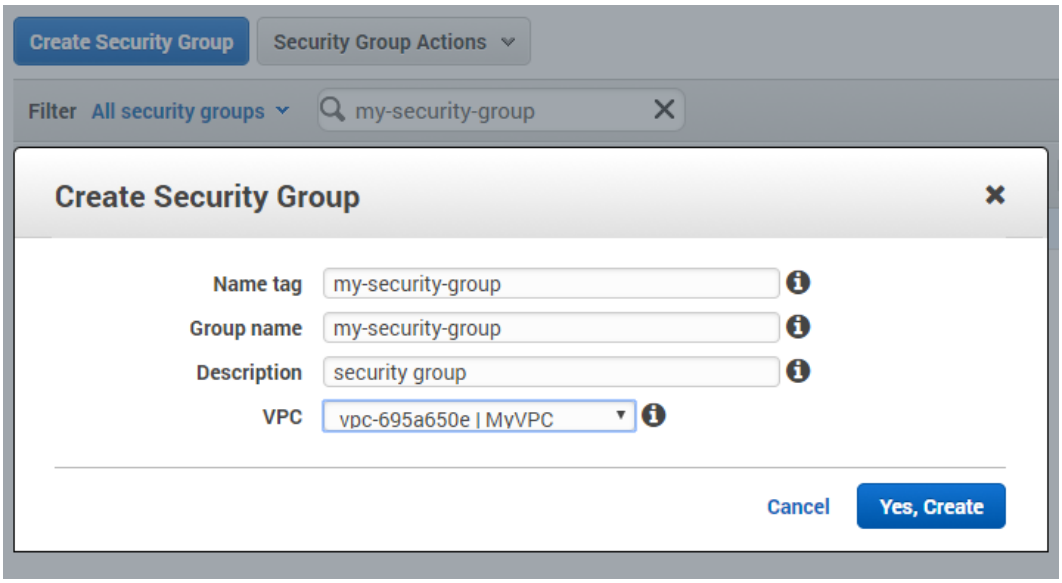
- SecureTransport Edge security group
- SecureTransport Server security group
- External Database security group: Oracle or MS SQL
- External File System security group

- Load Balancer security group
- Administration Host security group

Later, during the launch process of the instances and the creation of different components recommended for your SecureTransport setup, you will just select the security group you need from the list. You can always create the Security Groups on a later stage but we suggest you adhere to the flow as described.

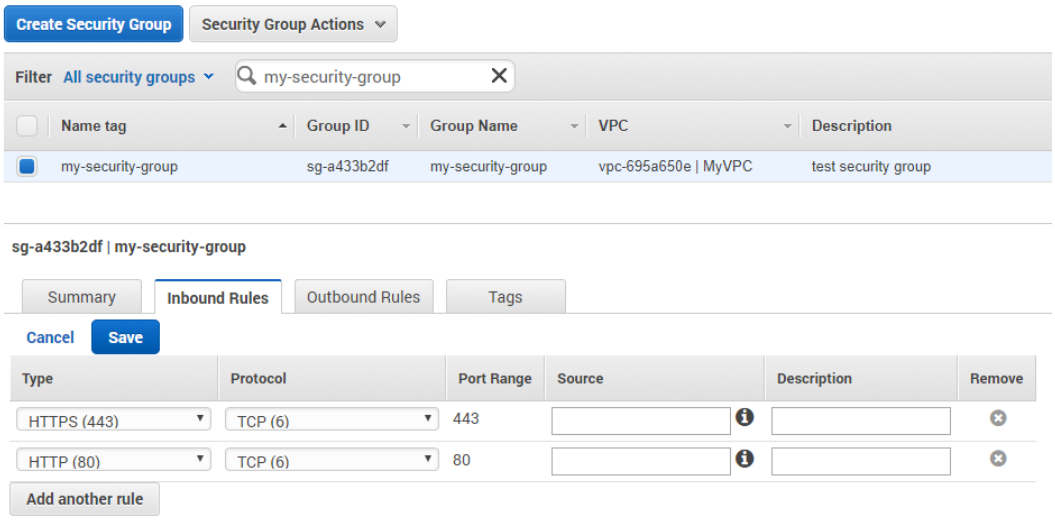
To create a security group in Amazon VPC:

1. Navigate to the AWS console->Services.
2. Under the Networking & Content Delivery section, choose VPC.
3. Navigate to Virtual Private Cloud->Security->Security Groups and click **Create Security Group**.
4. On the newly opened dialog box, enter the required information and click **Yes, Create**.



5. Select your security group from the "All security groups" table.
6. Edit the "Inbound Rules"/"Outbound Rules" section of the security group.

7. Use the **Add another rule** button to enable or disable access to your instances on specific ports or sources.



You must add the necessary inbound/outbound rules to the security groups and the instances assigned to each group will have the required for the group level of connectivity and security. Please refer to the firewall specific information already provided in SecureTransport Administrator's Guide.

Note The rules defined in the following Security Groups cover the most basic security scenario. If you would like more restrictive rules, you can add more Inbound and Outbound rules.

Note The described rules use default ports or ports specific for the test setup. Please change/add rules according to your specific setup. Check the FTP does not work through the firewall section in the SecureTransport Administrator's Guide if you want to configure FTP.

SecureTransport Edge Security Group

Allow inbound traffic according to the *Firewall rules* section as described in the *SecureTransport 5.4 Administrator's Guide*.

- **Inbound traffic** – this setup refers to traffic from the load balancer to the Edge serves.

Type	Protocol	Port / Port Range	Source	Description
HTTP	TCP	80	Load Balancer Security Group	HTTP
HTTPS	TCP	443	Load Balancer Security Group	HTTPS
Custom TCP Rule	TCP	10022	Load Balancer Security Group	SSH (SFTP and SCP)

Type	Protocol	Port / Port Range	Source	Description
Custom TCP Rule	TCP	21	Load Balancer Security Group	FTP (secure and non-secure) control channel (For secure connections: the firewall mustallow bidirectional communication)
Custom TCP Rule	TCP	20	Load Balancer Security Group	FTP (secure and non-secure) active-mode data channel
Custom TCP Rule	TCP	User-defined range	Load Balancer Security Group	FTP (secure and non-secure) passive-mode data channel
Custom TCP Rule	TCP	10080	Load Balancer Security Group	AS2(non-SSL)
Custom TCP Rule	TCP	10443	Load Balancer Security Group	AS2 (SSL)
Custom TCP Rule	TCP	17617	Load Balancer Security Group	PeSIT (non-SSL)
Custom TCP Rule	TCP	17627	Load Balancer Security Group	PeSIT over secure socket (non-Transfer CFT compatible)
Custom TCP Rule	TCP	17637	Load Balancer Security Group	PeSIT over secure socket (CFT compatible)
Custom TCP Rule	TCP	19617	Load Balancer Security Group	PeSIT over pTCP plain socket
Custom TCP Rule	TCP	19627	Load Balancer Security Group	PeSIT over pTCP Secured Socket

- **Streaming** – this setup refers to traffic from the ST servers to the Edge serves.

Type	Protocol	Port / Range	Source	Description
Custom TCP Rule	TCP	20080	SecureTransport Server Security Group	Streaming HTTP Server
Custom TCP Rule	TCP	20022	SecureTransport Server Security Group	Streaming SSH Server
Custom TCP Rule	TCP	20021	SecureTransport Server Security Group	Streaming FTP Server
Custom TCP Rule	TCP	21080	SecureTransport Server Security Group	Streaming AS2 Server
Custom TCP Rule	TCP	20444	SecureTransport Server Security Group	Streaming Administration Tool Server
Custom TCP Rule	TCP	27617	SecureTransport Server Security Group	Streaming PeSIT Server

- **Internal communication** – refers to traffic between the Edge serves and traffic from the Administration host.

Type	Protocol	Port or Port Range	Source	Description
Custom TCP Rule	TCP	33060	Administration Host SG	Database Administration
Custom TCP Rule	TCP	33060	SecureTransport Edge SG	MySQL communication
Custom TCP Rule	TCP	22	Administration Host Security Group	SSH for Administration
Custom TCP Rule	TCP	444	Administration Host Security Group	Administration Tool (HTTPS)
Custom TCP Rule	TCP	444	SecureTransport Edge Security Group	Cluster Synchronization

Type	Protocol	Port or Port Range	Source	Description
Custom TCP Rule	TCP	8005	SecureTransport Edge Security Group	Tomcat shutdown
Custom TCP Rule	TCP	8006	SecureTransport Edge Security Group	AS2 shutdown
Custom TCP Rule	TCP	7800-7802	SecureTransport Edge SG	Hibernate second level cache

SecureTransport Server Security Group

Allow inbound traffic according to the *Firewall rules* section as described in the *SecureTransport 5.4 Administrator's Guide*.

Note Inbound traffic from 8088-8093 range should be allowed for both TCP and UDP protocols from one SecureTransport Server to another (SecureTransport Server Security Group).

Type	Protocol	Port or Port Range	Source	Description
Custom TCP Rule	TCP	162	SecureTransport Server Security Group	SNMP
Custom UDP Rule	TCP	8088-8093	SecureTransport Server Security Group	Cluster cache management
Custom UDP Rule	UDP	8088-8093	SecureTransport Server Security Group	Cluster cache management
Custom TCP Rule	TCP	444	Administration Host Security Group	Administration Tool (HTTPS)
Custom TCP Rule	TCP	44431	SecureTransport Server Security Group	Cluster Listener
Custom TCP Rule	TCP	9999	SecureTransport Server Security Group	TM JMX Port

Type	Protocol	Port or Port Range	Source	Description
Custom ICMP Rule	IPv4	Echo Reply	SecureTransport Server Security Group	Echo Reply
SSH	SSH	22	Administration Host Security Group	SSH for Administration
Custom TCP Rule	TCP	8005	SecureTransport Server Security Group	Tomcat shutdown port
Custom TCP Rule	TCP	8009	SecureTransport Server Security Group	Tomcat JK connector
Custom TCP Rule	TCP	20444	SecureTransport Server Security Group	Administration Tool
Custom TCP Rule	TCP	7	SecureTransport Server Security Group	Coherence
Custom TCP Rule	TCP	7800-7802	SecureTransport Server Security Group	Hibernate second-level cache
Custom ICMP Rule	IPv4	Echo Request	SecureTransport Server Security Group	Echo Request

External Database Security Group

Allow inbound traffic depending on your selected database: Oracle or MS SQL.

Oracle Database Security Group

Type	Protocol	Port or Port Range	Source	Description
Oracle	RDS	1521	Administration Host Security Group SecureTransport Server Security Group	Non-SSL access to the database
Custom TCP Rule	TCP	2484	Administration Host Security Group SecureTransport Server Security Group	SSL access to the database

MS SQL Database Security Group

Type	Protocol	Port or Port Range	Source	Description
MS SQL	TCP	1433	Administration Host Security Group SecureTransport Server Security Group	Access to the database

GlusterFS Security Group

Allow the following inbound traffic:

Type	Protocol	Port or Port Range	Source	Description
SSH	SSH	22	Administration Host Security Group	Administration Host Security Group
Custom TCP Rule	TCP	24007	GlusterFS Security Group SecureTransport Server Security Group	Gluster Daemon
Custom TCP Rule	TCP	111	GlusterFS Security Group SecureTransport Server Security Group	Portmapper
Custom TCP Rule	TCP	49152-49251	GlusterFS Security Group SecureTransport Server Security Group	Each brick for every volume on your host requires its own port
Custom TCP Rule	TCP	2049	GlusterFS Security Group SecureTransport Server Security Group	NFS

Load Balancer Security Group

Allow the following inbound traffic:

Type	Protocol	Port or Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	HTTP
HTTPS	TCP	443	0.0.0.0/0	HTTPS
Custom TCP Rule	TCP	10022	0.0.0.0/0	SSH (SFTP and SCP)

Type	Protocol	Port or Port Range	Source	Description
Custom TCP Rule	TCP	21	0.0.0.0/0	FTP (secure and non-secure) control channel (For secure connections: the firewall must allow bidirectional communication)
Custom TCP Rule	TCP	20	0.0.0.0/0	FTP (secure and non-secure) active-mode data channel
Custom TCP Rule	TCP	User-defined range	0.0.0.0/0	FTP (secure and non-secure) passive-mode data channel
Custom TCP Rule	TCP	10080	0.0.0.0/0	AS2(non-SSL)
Custom TCP Rule	TCP	10443	0.0.0.0/0	AS2 (SSL)
Custom TCP Rule	TCP	17617	0.0.0.0/0	PeSIT (non-SSL)
Custom TCP Rule	TCP	17627	0.0.0.0/0	PeSIT over secure socket (Transfer CFT compatible)
Custom TCP Rule	TCP	17637	0.0.0.0/0	PeSIT over secure socket (CFT compatible)
Custom TCP Rule	TCP	19617	0.0.0.0/0	PeSIT over pTCP plain socket
Custom TCP Rule	TCP	19627	0.0.0.0/0	PeSIT over pTCP Secured Socket

Administration Host Security Group

Allow the following inbound traffic:

Type	Protocol	Port or Port Range	Source	Description
RDP	TCP	3389	Your IP address	Remote connection to the Administration Host

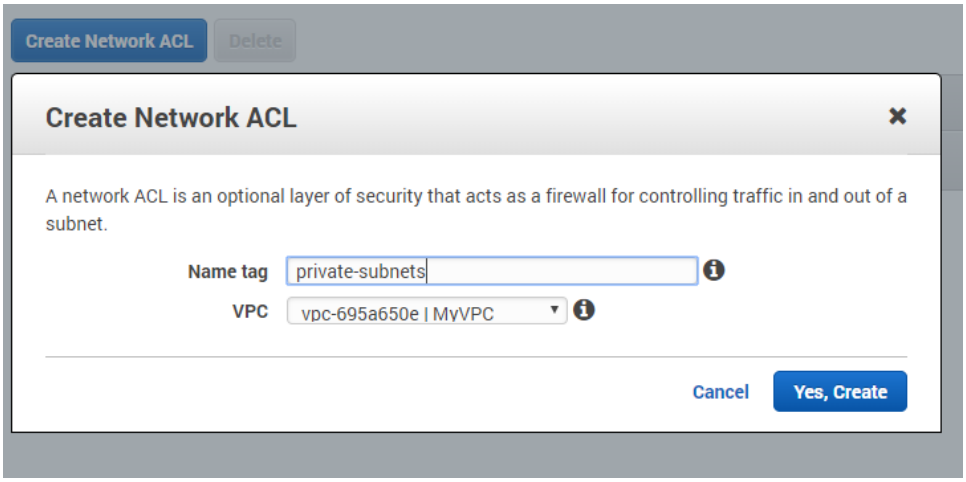
Network Access Lists

You can secure your VPC instances using only security groups and in general they are sufficient to secure your subnets, but we recommend you to add network ACLs as a second layer of defense. For more information about network ACLs, see Network ACLs.

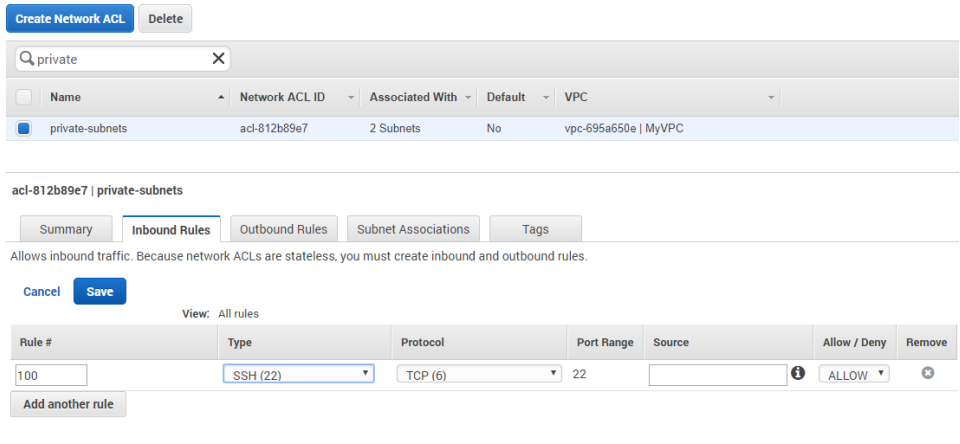
- Create one Network ACL for your public subnets hosting the SecureTransport Edge servers and add inbound/outbound rules limiting the access to the required minimum for the relevant subnets.
- Create one Network ACL for your private subnets hosting the SecureTransport servers and add inbound/outbound rules limiting the access to the required minimum for the relevant subnets.
- Create one Network ACL for the private subnets hosting the database instances and add inbound/outbound rules limiting the access to the required minimum for the relevant subnets. Later after you create the subnets, associate them with their corresponding Network ACL.

To create Network Access Lists in Amazon VPC:

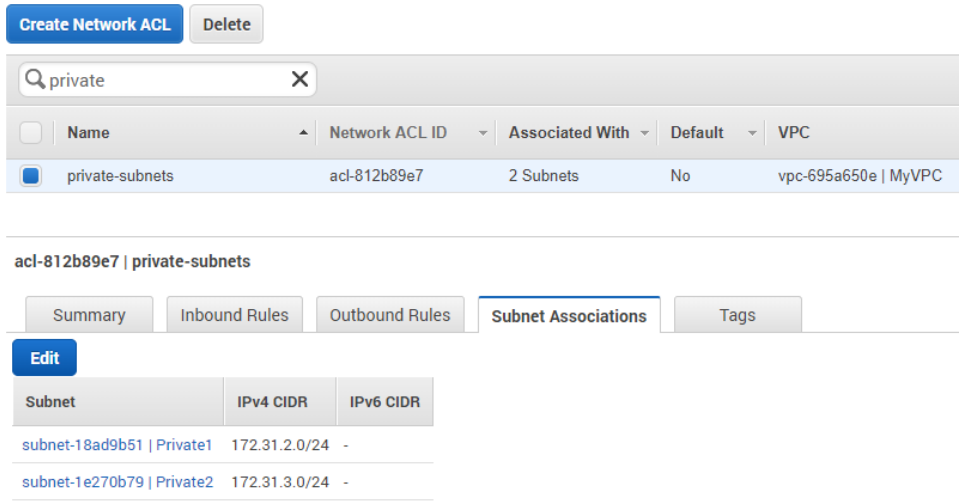
1. Navigate to the AWS console -> Services.
2. Under the Networking & Content Delivery section, choose **VPC**.
3. Navigate to Virtual Private Cloud -> Security -> Network ACLs.
4. Select Create Network ACL, enter a meaningful name, choose your VPC and confirm by clicking **Yes, Create**.



5. Enter Inbound/Outbound rules relevant for the subnets that will be associated with this Network ACL and then click **Save**.



6. Associate the corresponding subnets with this Network ACL and save.



Access your servers using Administration host

When designing the Administration host for your AWS infrastructure, you should use it only for maintenance and administration tasks and avoid opening unnecessary security holes. You need to keep it locked down as much as possible. You could look into hardening your chosen operating system for even tighter security. You should stop your Administration host instances for the duration with no maintenance work and start them when you need access to your servers in AWS in order to minimize the security risks.

Here are the basic steps for creating a bastion host for your AWS infrastructure (see section Launch an instance for Administration Host):

- Launch an EC2 instance.
- Apply your OS hardening as required.

- Set up the appropriate security groups (SG).
- Implement either SSH-Agent Forwarding (Linux connectivity) or Remote Desktop Gateway (Windows connectivity).
- Deploy an AWS bastion host in each of the Availability Zones you're using.

Find more information about the Bastion host architecture in AWS in the Linux Bastion Architecture documentation. See also the Linux Bastion hosts on the AWS cloud guide.

See also the How to record ssh sessions established through a bastion host topic for closer monitoring over this host.

Security groups are essential for maintaining tight security and play a big part in making this solution work. First, you need to create a security group or update an existing security group that will be used to allow connectivity from the Administration host for your existing private instances (see the *SecureTransport Server Security Group* in the *Security* section of the guide). This SG should only accept SSH or RDP inbound requests from your Administration hosts across your Availability Zones. Apply this group to all your private instances that require connectivity.

Next, create a security group to be applied to your Administration host. Inbound and outbound traffic must be restricted at the protocol level as much as possible. The inbound rule base should accept SSH or RDP connections only from specific IP addresses (usually those of your administrators' work computers). See the *Administration Host Security Group* in the *Security Groups* section of this guide. Your outbound connection should again be restricted to SSH or RDP access to the private instances of your AWS infrastructure. An easy way to do this is to populate the 'Destination' field with the ID of the security group you are using for your private instances.

SSH and RDP connections require private and public key access to authenticate. This does not pose a problem when you are trying to connect to your Administration host from a local machine, as you can easily store the private key locally. However, once you have connected to your Administration host, logging in to your private instances from this host would require having their private keys on the Administration host (storing private keys on remote instances is not a good security practice).

As a result, you should implement either Remote Desktop Gateway (for connecting to Windows instances) or SSH-agent forwarding (for Linux instances). Both of these solutions eliminate the need for storing private keys on the Administration host. AWS provides detailed documentation on how to implement Windows Remote Desktop Gateway and SSH-agent forwarding.

Replacing the Administration Host with Amazon EC2 Systems Manager

There is an alternative solution to having an Administration host with access to your private servers in the VPC and it is called Amazon EC2 Systems Manager.

Systems Manager allows you to remotely execute commands on managed hosts without using an Administration host. A host-based agent polls Systems Manager to determine if there is a command awaiting execution.

You can find more information about this option in the Replacing a Bastion Host with Amazon EC2 Systems Manager topic.

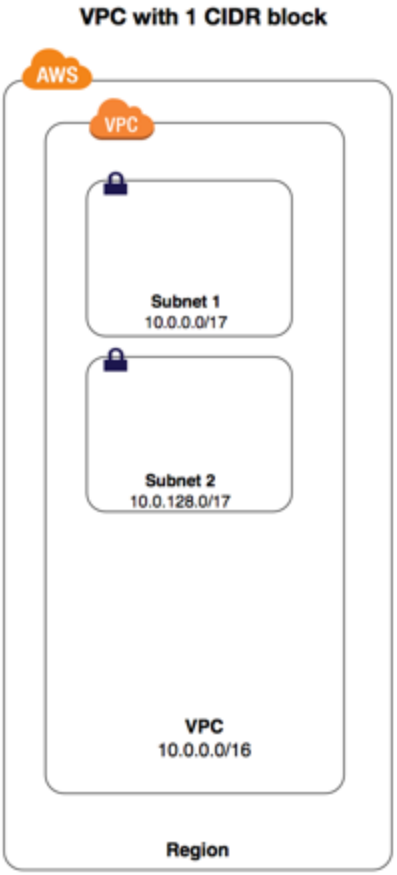
There is no cost to use Systems Manager but you are responsible for the costs of the resources that use Systems Manager, such as the EC2 instances, SNS messages, and S3 storage.

Subnets

After creating a VPC, you can add one or more subnets in each Availability Zone. Public Subnets have access to the Internet, while private do not. When you create a subnet, you specify the CIDR block for the subnet, which is a subset of the VPC CIDR block. Each subnet must reside entirely within one Availability Zone and cannot span zones. Availability Zones are distinct locations that are engineered to be isolated from failures in other Availability Zones. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.

Learn more about VPCs and Subnets.

In the following example, the VPC on the left has a single CIDR block (10.0.0.0/16) and two subnets.



Main route table

Destination	Target
10.0.0.0/16	local

Create subnets

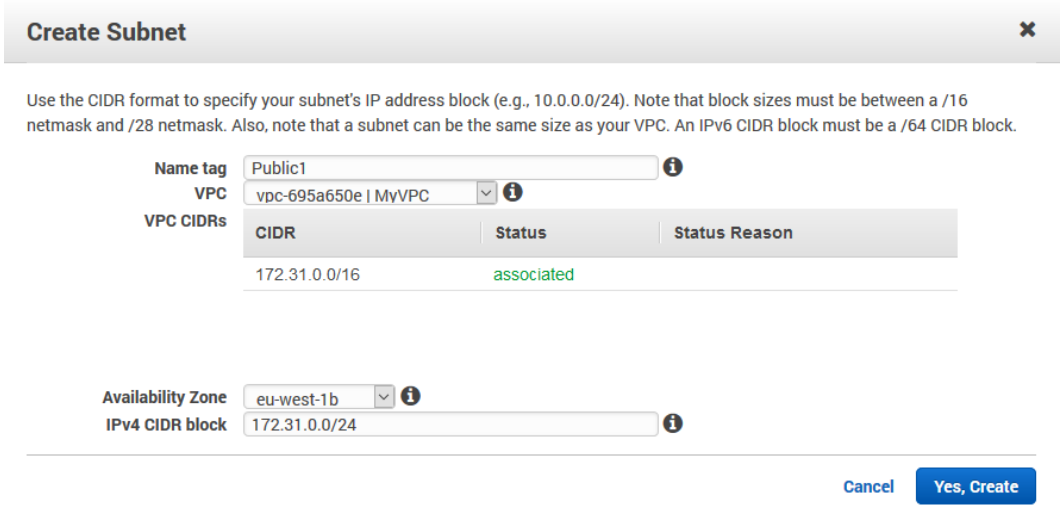
Create six subnets (four private and two public) as follows:

Two private and one public subnets per availability zone.

1. Navigate to the AWS console Services.
2. Under Networking & Content Delivery section choose.
3. Navigate toSubnets and Create Subnet.
4. Fill the settings and click **Yes, Create**.
    - Name tag: specify unique name for each subnet
    - VPC: select you VPC for all subnets
    - Availability Zone: create each subnet in a different zone
      For example: the first public and the first and third private subnets are in eu-west-1 b.
      The second public and the second private subnets are in eu-west-1 c.
    - IPv4 CIDR block: for each subnet specify a different block
      For example:
        - 172.31.0.0/24 - Public1
        - 172.31.3.0/24 - Public2
        - 172.31.1.0/24 - Private1
        - 172.31.2.0/24 - Private2
        - 172.31.4.0/24 - Private3
        - 172.31.5.0/24 - Private4

Note This is an example CIDR block size. You can configure the CIDR block according to your needs.

5. Repeat the steps for the rest subnets.



Internet Gateway and public subnets routing

An Internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the Internet. It therefore imposes no availability risks or bandwidth constraints on your network traffic.

An Internet gateway serves two purposes: to provide a target in your VPC route tables for Internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

In order for the resources in a VPC to send and receive traffic from the Internet, the following conditions must be met:

- An Internet gateway must be attached to the VPC.
- The route tables associated with your public subnet (including custom route tables) must have a route to the Internet gateway.
- The security groups associated with your VPC must allow traffic to flow to and from the Internet.
- Any instances in the VPC must either have a public IP address or an attached Elastic IP address.

You can find instructions for each of these steps at Creating a VPC with an Internet Gateway.

Attach an Internet gateway

Follow these steps to attach an Internet gateway to your VPC to enable communication of the public subnets with the Internet:

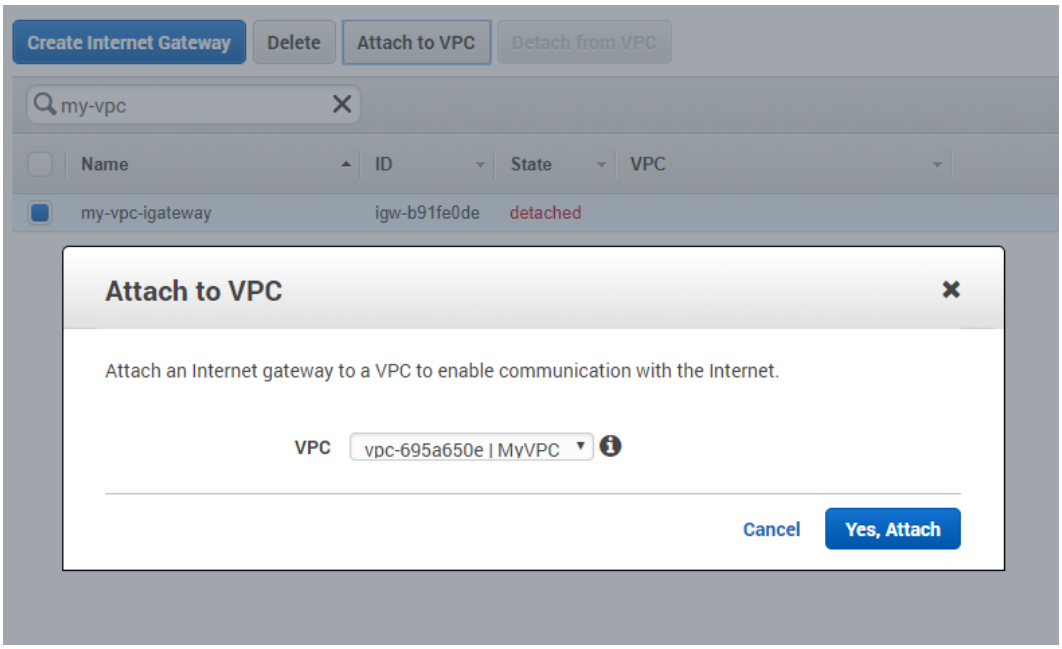
1. Navigate to the AWS console -> Services.
2. Under the Networking & Content Delivery section, choose VPC.
3. Navigate to Virtual Private Cloud -> Internet Gateways.
4. Click **Create Internet Gateway**.

5. Type a name in the Name tag text box and click **Yes, Create**.

The internet gateway just created is in a "detached" state. Your next step is to attach it to your VPC.

6. Click **Attach to VPC**.

7. Select your VPC from the Name tag drop-down list and click **Yes, Attach**.



On success, the state of the internet gateway changes to "attached".

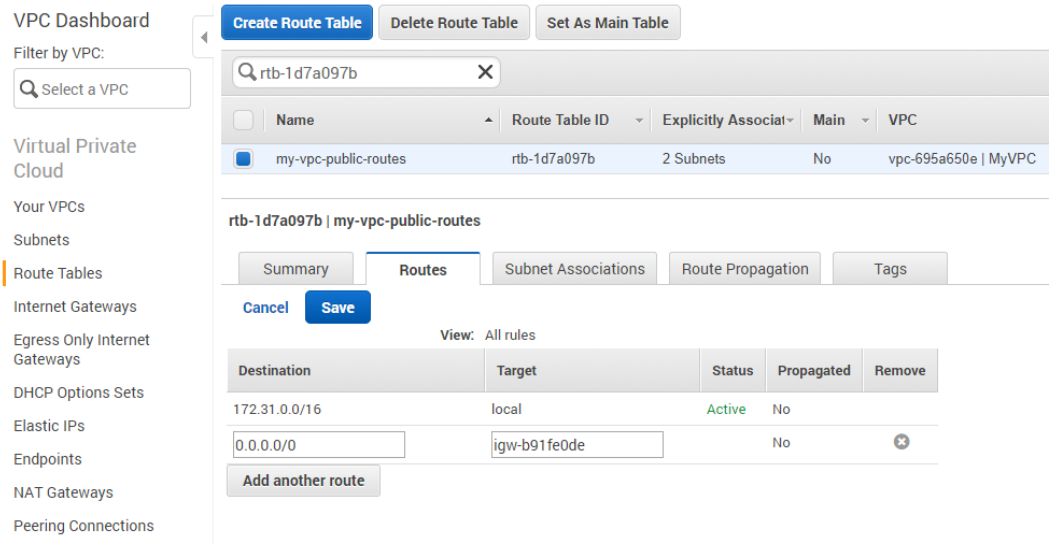
Routing of public subnets

Now you need to configure the routing for your public subnets. Enable traffic from your public subnets to Internet by using the internet gateway attached to the VPC.

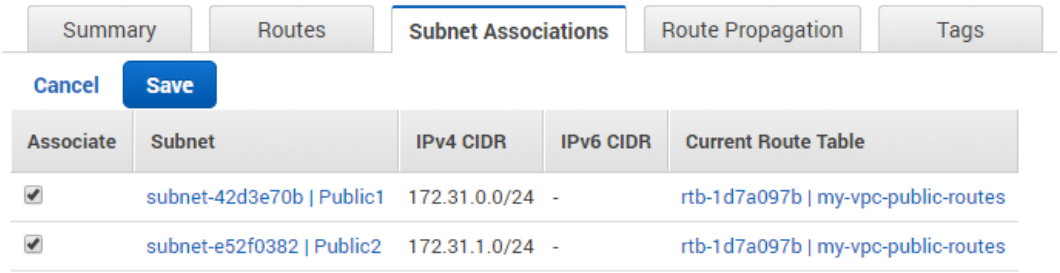
Configure the public subnets Route Table:

1. Navigate to VPC Dashboard -> Subnets.
2. Select your first public subnet from the list and navigate to its Summary section.
3. Click on the name of the Route Table of the subnet.
4. You are then redirected to the Route Table in the Virtual Private Cloud -> Route Tables section.
5. Add two routes for the Route Table - one for the traffic to the Internet to be routed using the Internet Gateway.

6. Add new rules: for destination type 0.0.0.0/0 (all packets for the internet) and for target choose the Internet Gateway you have created as in the previous subtopic.



7. Save the rules.
8. Navigate to "Subnet Associations" tab and associate your public subnets to the route table and save the changes.



Now traffic from instances in the public subnets destined to the Internet will be redirected to the Internet Gateway.

NAT Gateway and private subnets routing

NAT Gateway in AWS can provide your private instances with access to the Internet for essential software updates while blocking incoming traffic from the outside world.

The private subnets in your VPC should have access to Internet only through an AWS feature called NAT Gateway. The NAT Gateway configuration is optional and you can skip it if you want your instances in the private subnets to be completely restricted from accessing the Internet.

You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the Internet or other AWS services, but prevent the internet from initiating a connection with those instances.

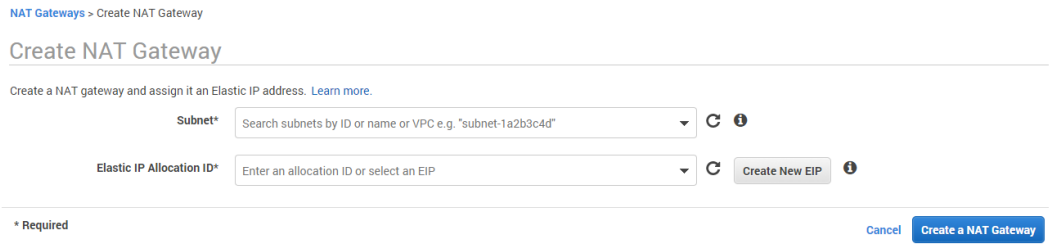
Learn more about NAT.

To create a NAT gateway, you must specify a subnet and an Elastic IP address. Make sure that the Elastic IP address is currently not associated with an instance or a network interface.

Create NAT Gateway

Configure NAT Gateway for private subnets in your VPC:

1. Navigate to the AWS console -> Services.
2. Go to the Networking & Content Delivery section and click **VPC**.
3. Navigate to Virtual Private Cloud -> NAT Gateways -> Create NAT Gateway.

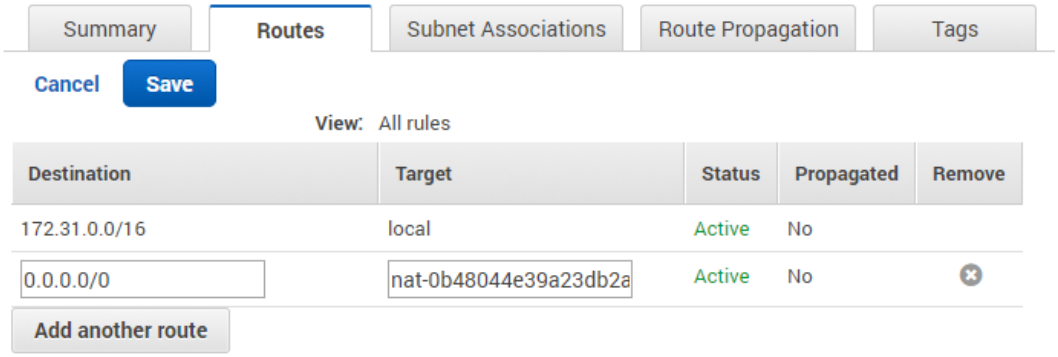


4. On the newly opened page select a public subnet from the **Subnet** drop-down list in which to create the NAT gateway.
5. Assign an Elastic IP Address to the NAT Gateway.
6. Click **Create a NAT Gateway**.

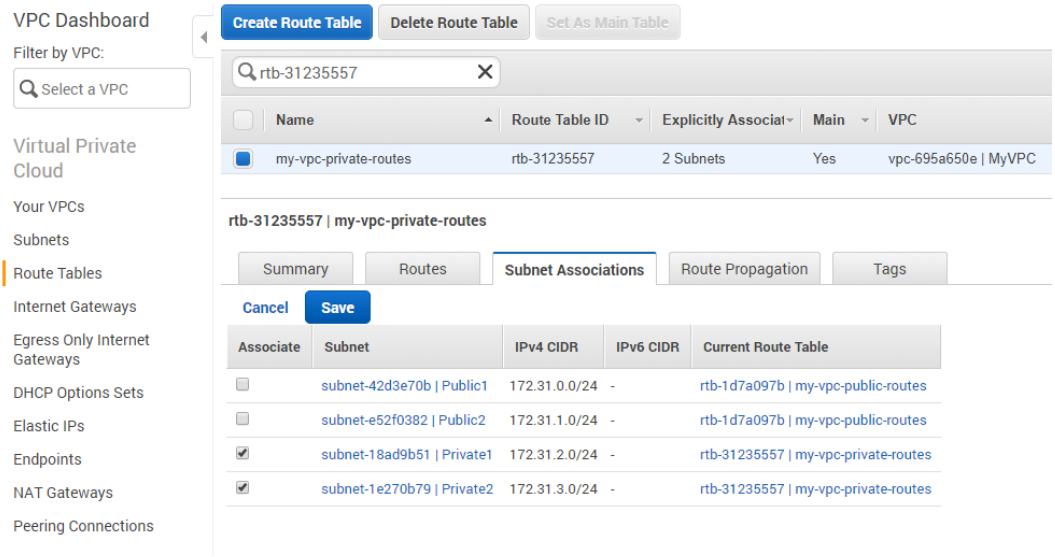
Now you need to configure the routing for your private subnets. Enable traffic from your private subnets to Internet by using the NAT Gateway you have created.

Configure private subnets route table

1. Navigate to VPC Dashboard -> Subnets.
2. Select one of your private subnets from the list and navigate to its Summary section.
3. Click on the name of the Route Table of the subnet.
4. You are then redirected to your Route Table in the Virtual Private Cloud -> Route Tables subsection.
5. Add a new rule and for destination type 0.0.0.0/0 (all packets for the internet) and for target choose the NAT Gateway you created as in the previous subtopic.
6. Save the rules and now the traffic from instances in the private subnets destined to the Internet will be redirected to the NAT Gateway.



7. Save the rules.
8. Navigate to the Subnet Associations tab and associate your private subnets with the route table and save the changes.



Now traffic from instances in the private subnets destined for the Internet will be redirected to the NAT Gateway.

Amazon RDS 5

Amazon Relation Database Service (Amazon RDS) is a web service that allows youto set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

The DB instance is the basic building block of Amazon RDS and is defined as an isolated database environment in the cloud. A DB instance can contain multiple user-created databases, and you can access it by using the same tools and applications that you use with a stand-alone database instance.

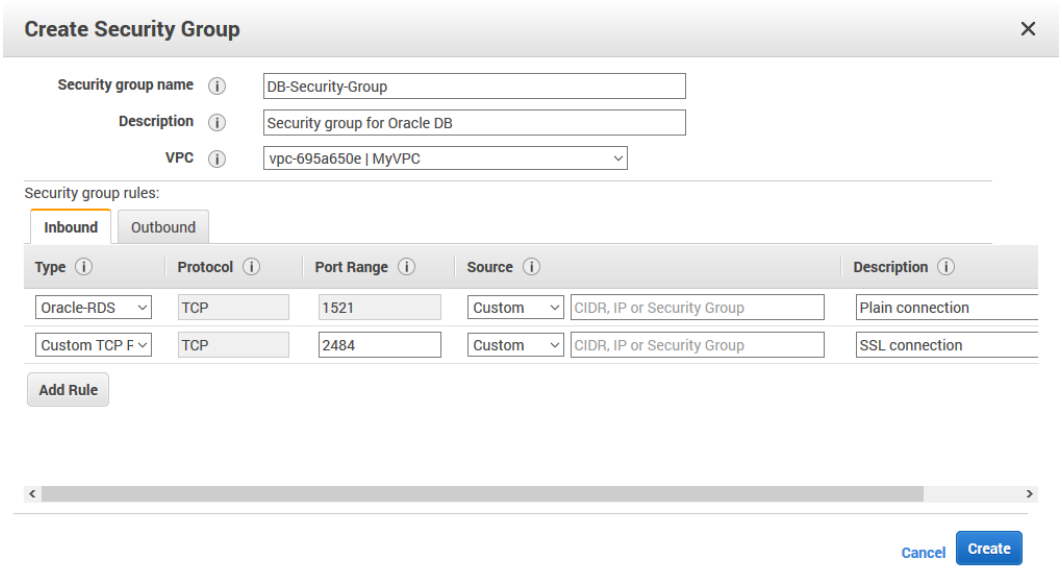
Each DB instance runs a DB engine. For the list of supported database engines in the Amazon RDS environment, refer to *Axway and third-party software support* on page 1.

Deploy Oracle database in Amazon RDS

Create database Security Group

First, create a Security Group for your database:

1. Navigate to AWS console -> Services.
2. Go to the Compute section and select **EC2**.
3. Go to the Network & Security section and select **Security Groups**.
4. Click **Create Security Group**.

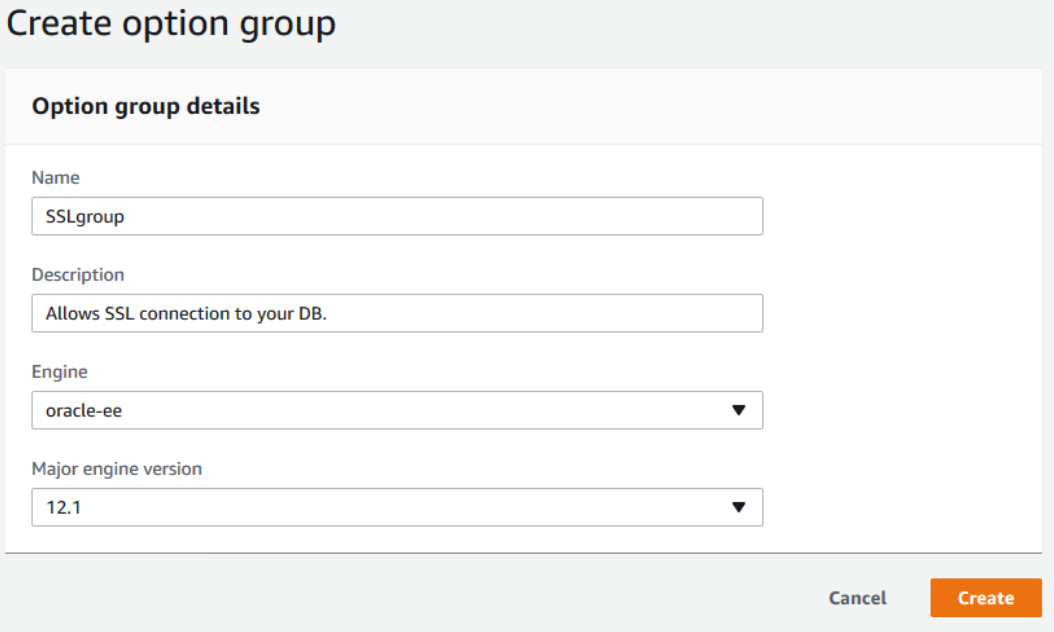


5. After created, select the Security Group and go to Actions -> Add/Edit Tags.
6. In the **Key** text box type *Name*, and in the **Value** text box enter *OracleDB*.
7. Click **Save**.

Create option group

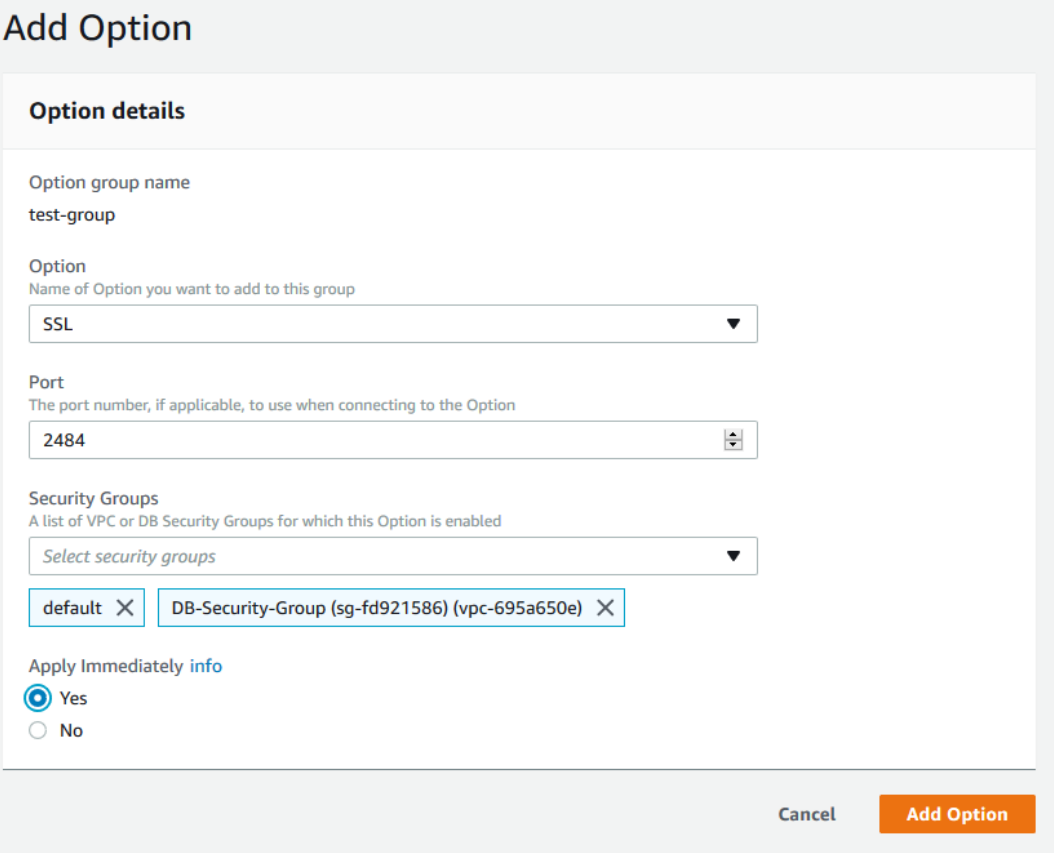
If you would like to make a SSL connection to your Oracle DB, you should first create an **Option group**:

1. Navigate to AWS console -> Services.
2. Go to the Database section and select **RDS**.
3. Go to Option groups and click **Create group**.



4. After the group is created, select it and click **Add option**.
5. On the **Option** drop-down list, select SSL.
6. Specify the SSL Port.
7. Select Security Group for which this option is enabled - select the previously created Database Security Group.

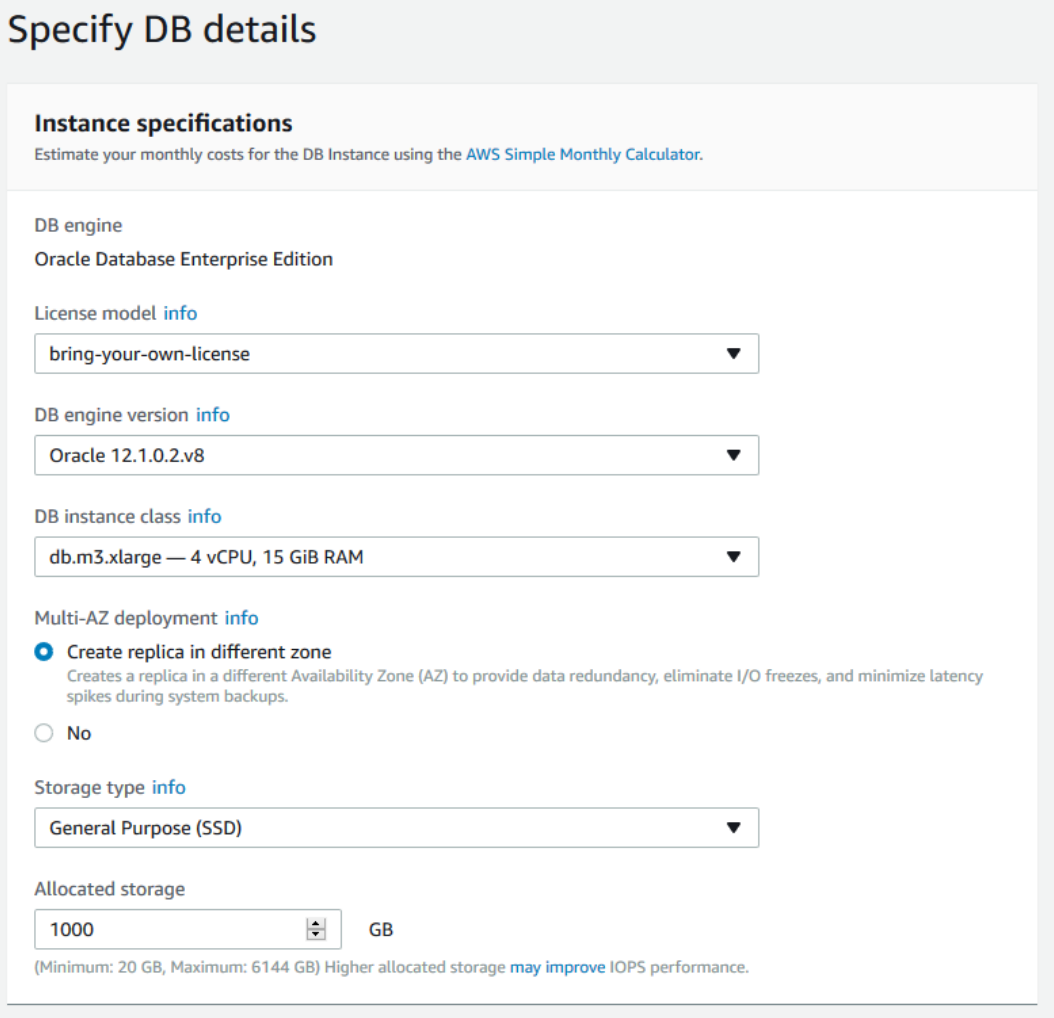
8. On the Apply immediately options, select **Yes**.



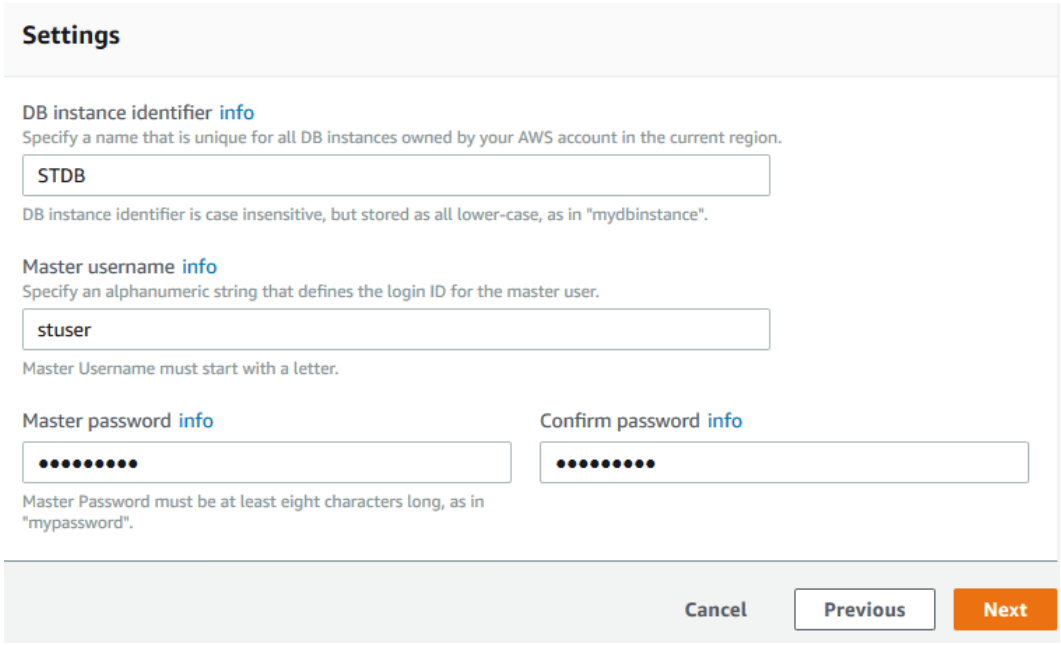
Create Oracle database

1. Navigate to AWS Console -> Services.
2. Go to the Database section and select **RDS**.
3. Go to Instances and click **Launch DB Instance**.
4. Select Oracle -> Enterprise Edition.
5. For Use Case, select **Production**.

6. Specify the DB Details.



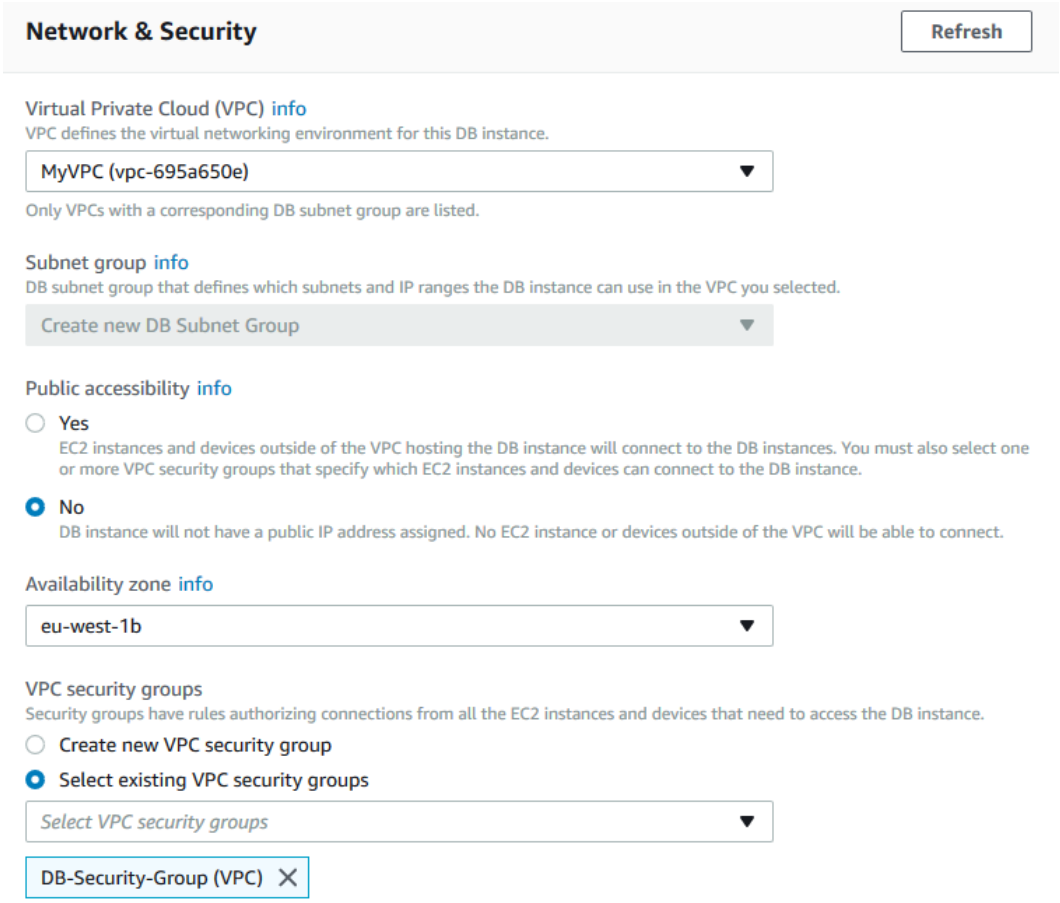
Add your DB Instance Identifier and Master user credentials.



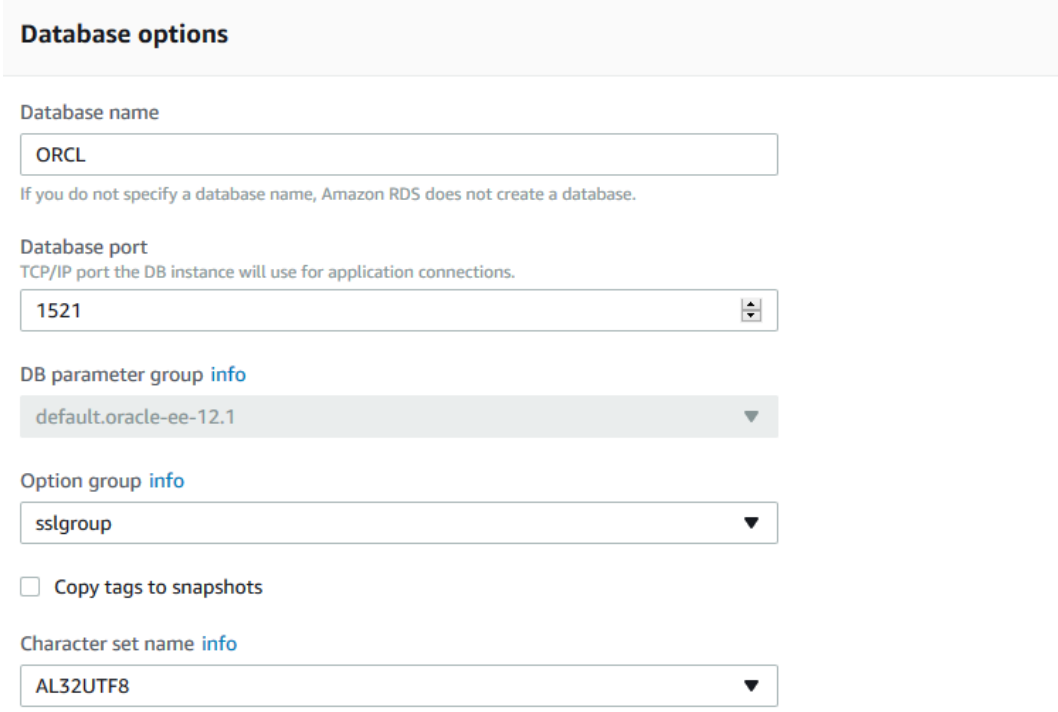
7. Click **Next**.
8. Configure the Advanced Settings:
    - Launch the Database in your VPC.
    - Choose whether your database to be publicly accessible or no.
      Select Yes if you want to allow EC2 instances and devices outside the VPC that hosts the DB instance to connect to this DB instance. If you select No, Amazon RDS will not assign a public IP address to the DB instance, and no EC2 instance or devices outside of the VPC will be able to connect. If you select Yes, you must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance. Click here to learn more.
    - Select the Availability Zone from the current region in which you want the DB instance created.

    Note: For high availability and fault tolerance, we recommend you to create a DB replica in different zones in the previous step.

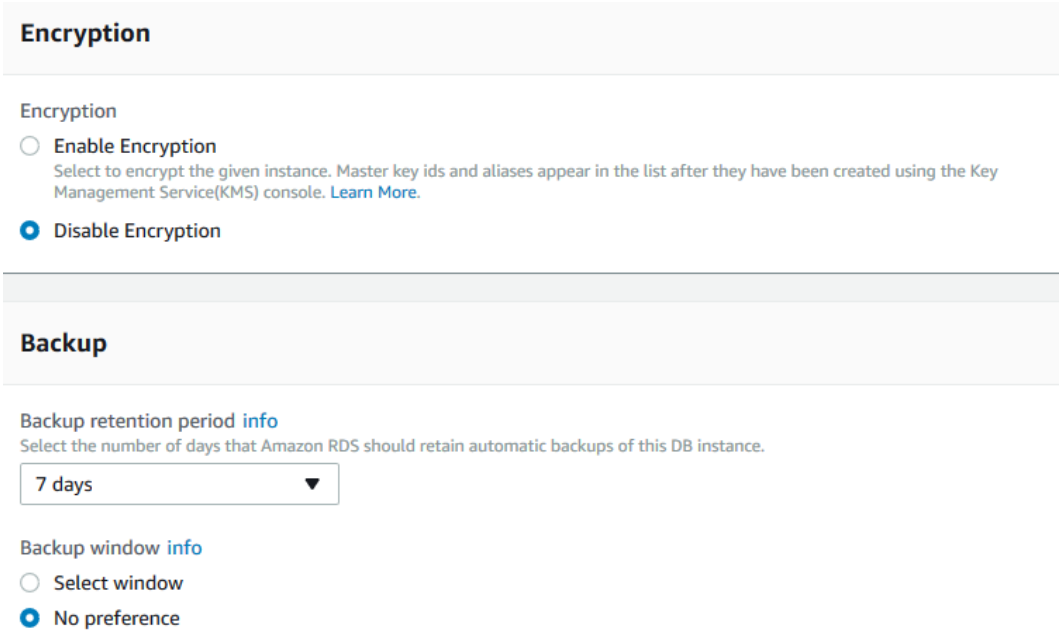
    - Select the Database Security Group you created as described in the Create database Security Group subtopic.



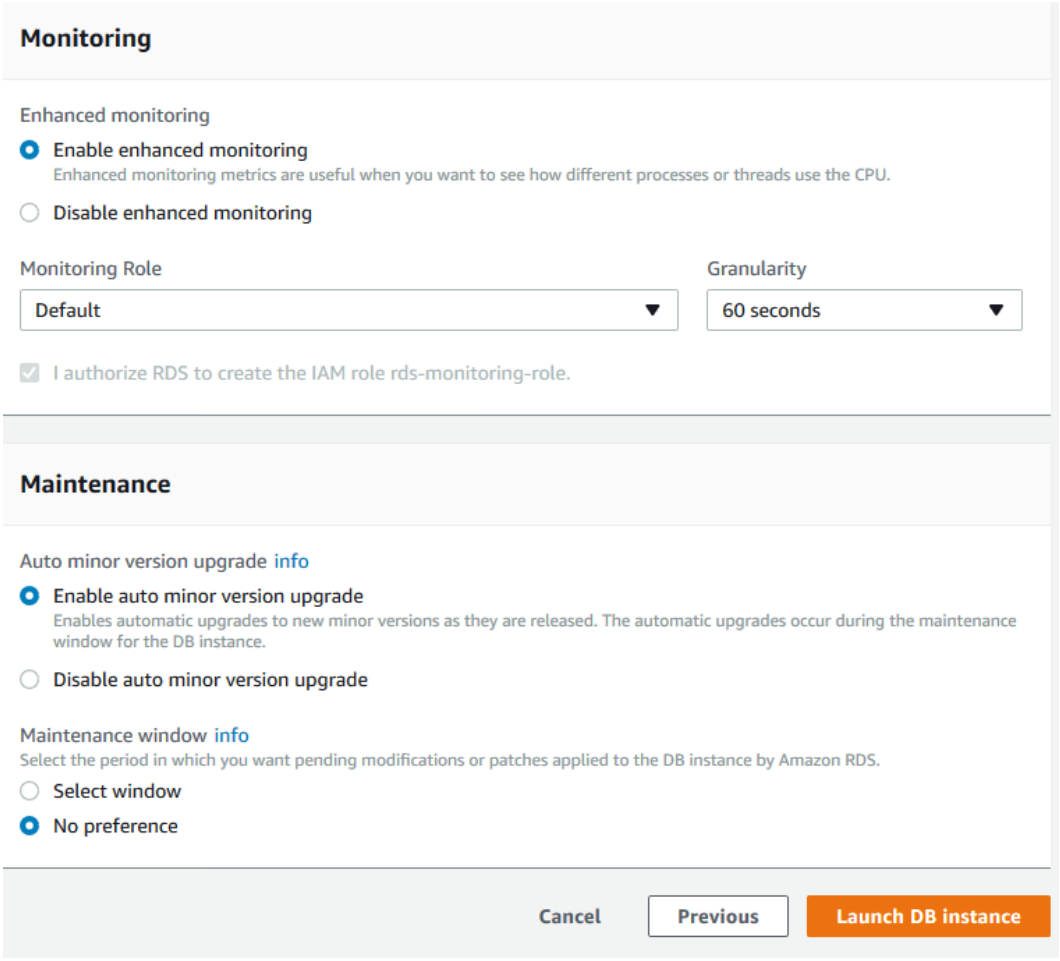
9. Add your Database Options:
    - Set a Database name.
    - Specify a Database port.
    - Select the previously created SSL Option Group.
    - Leave Character set name to the default value: AL32UTF8.



- Select **Enable Encryption** and follow the instructions to supply your Master key IDs and aliases.



- Add your preferences for Monitoring and Maintenance.



10. When you finish with your setup, click **Launch DB instance**.

Parameter Groups

You manage your DB engine configuration through the use of parameters in a DB parameter group.

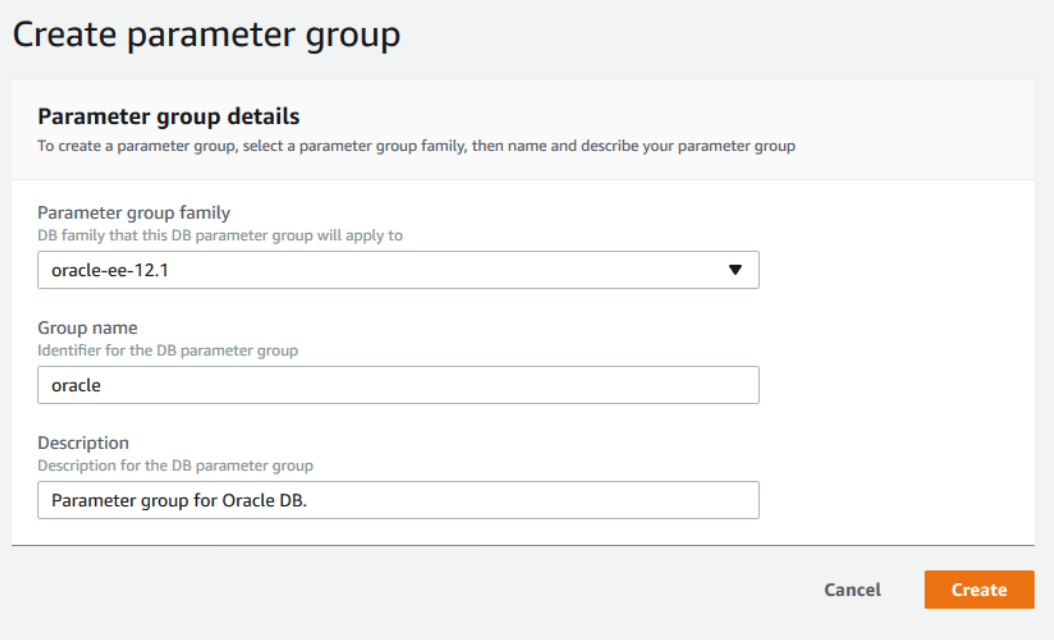
DB parameter groups act as a *container* for engine configuration values that are applied to one or more DB instances.

You cannot modify the parameter settings of a default DB parameter group; you must create your own DB parameter group to change parameter settings from their default value according to the *Requirements for Oracle Databases* section in the*SecureTransport5.4 Installation Guide*.

Create a parameter group

1. Navigate to AWS console -> Services.
2. Go to the Database section and select **RDS**.
3. Go to Parameter groups and click **Create parameter group**.

4. Fill in the fields and click **Create**.



5. After creation, select the parameter group and go to Parameter group actions and click **Edit**.
6. Find the parameter that you would like to change and click **Edit parameters**.
7. Insert the desired value and then click **Save changes**.
8. Change the following parameters according to the database requirements:
    - db_cache_size: 1GB or larger
    - open_cursors: at least 1000
    - processes: 1000 or more

Learn more about DB Parameter Groups.

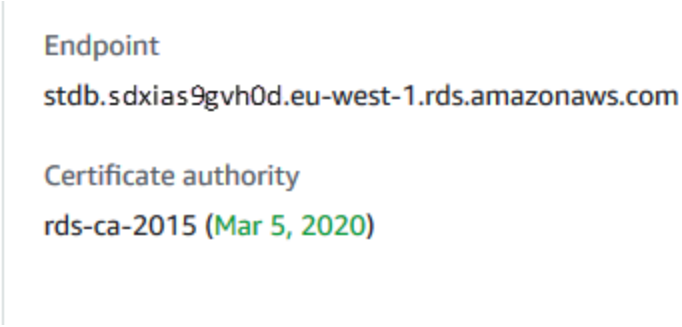
Assign the parameter group to your database

1. Navigate to RDS Instances.
2. Select your database and then go to Instance actions -> Modify.
3. Go to the Database options section and select your **DB parameter group** from the drop-down list.
4. Save and apply changes immediately.
5. Restart your database.

Connect to your Oracle database

1. Navigate to AWS console -> Services.
2. Go to the Database section and select **RDS**.

3. Navigate to **Instances**.
4. Select your newly created Database and then go to Instance Actions ->See Details.
5. Under the Security and network section, see Endpoint: you will need to copy & paste it in the next step.



6. Use Oracle SQL Developer on your Administration Host.
   Add new connection and provide values for the following options:
   - **Connection Name**: type a name that describes the connection
   - **Username**: name of the database administrator
   - **Password**: password for the database administrator
   - **Hostname**: paste the Endpoint
   - **Port**: 1521
   - **SID**: ORCL

For further reference, see Connecting to Oracle DB.

Create tables and set ownership of the Oracle database

Use the following script:

```
CREATE SMALLFILE TABLESPACE "ST_DATA"
DATAFILE SIZE 5000M AUTOEXTEND ON NEXT 12K MAXSIZE 8000M
LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
CREATE SMALLFILE TABLESPACE "ST_FILETRACKING"
DATAFILE SIZE 5000M AUTOEXTEND ON NEXT 12K MAXSIZE 8000M
LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
CREATE SMALLFILE TABLESPACE "ST_SERVERLOG"
DATAFILE SIZE 5000M AUTOEXTEND ON NEXT 12K MAXSIZE 8000M
LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
CREATE USER ST IDENTIFIED BY ST;

         grant connect to ST;

         grant create operator to ST;

         grant create procedure to ST;
```

```
grant create sequence to ST;
grant create session to ST;
grant create table to ST;
alter user ST quota unlimited on ST_DATA;
alter user ST quota unlimited on ST_FILETRACKING;
alter user ST quota unlimited on ST_SERVERLOG;
alter user ST quota unlimited on USERS;
```

Obtain the Database certificate and a Distinguished Name

Execute the following command from one of your RHEL Instances which have access to the database:

```
openssl s_client -connect <host>:<ssl_port>
```

where `<host>` is the Endpoint.

Create directories for the exported files

If you would like to run Maintenance applications and export logs, you will have to create a directory on the RDS service. This directory will contain the exported files.

To create a new directory, you can use the Amazon RDS procedure `dsadmin.rdsadmin_util.create_directory`.

The following example creates a new directory named `ST_DMPDIR`:

```
exec rdsadmin.rdsadmin_util.create_directory(p_directory_name => 'ST_DMPDIR');
```

You can list the directories by querying `DBA_DIRECTORIES`. The system chooses the actual host pathname automatically. The following example gets the directory path for the directory named `ST_DMPDIR`:

```
select DIRECTORY_PATH from DBA_DIRECTORIES where DIRECTORY_NAME='ST_DMPDIR';
```

The master user for the DB instance has read and write privileges in the new directory, and can grant access to other users. You will need to grant read and write privileges to your SecureTransport user.

Execute privileges are not available for directories on a DB instance. Directories are created in your main data storage space and will consume space and I/O bandwidth.

List Files in a DB Instance Directory

You can use the Amazon RDS procedure `rdsadmin.rds_file_util.listdir` to list the files in a directory.

The following example lists the files in the directory named `ST_DMPDIR`:

```
select * from table (rdsadmin.rds_file_util.listdir(p_directory => 'ST_DMPDIR'));
```

Learn more about Creating directories in RDS.

Deploy MS SQL database in Amazon RDS

Create database Security Group

First, create a Security Group for your database:

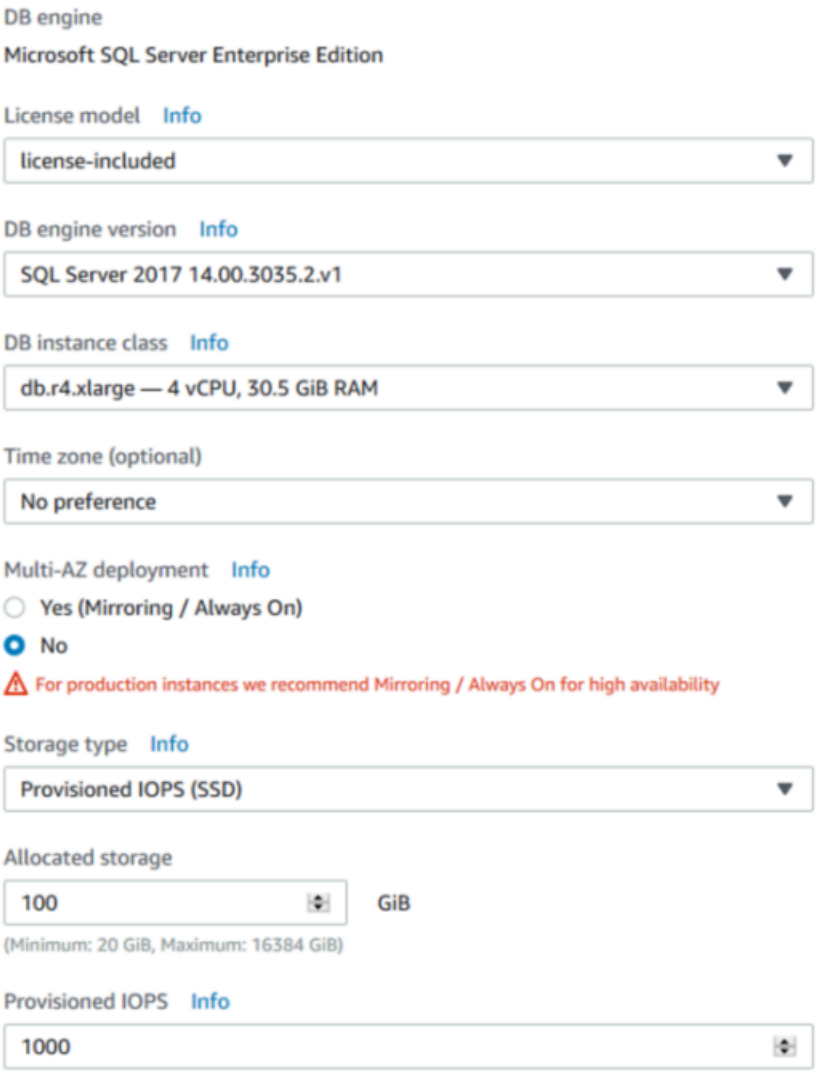
1. Navigate to AWS console -> Services.
2. Go to the Compute section and select **EC2**.
3. Go to the Network & Security section and select **Security Groups**.
4. Click **Create Security Group** and provide your values for the inbound rules.

5. After created, select the Security Group and go to Actions -> Add/Edit Tags.
6. In the **Key** text box type *Name*, and in the **Value** text box enter *MS SQL DB*.
7. Click **Save**.

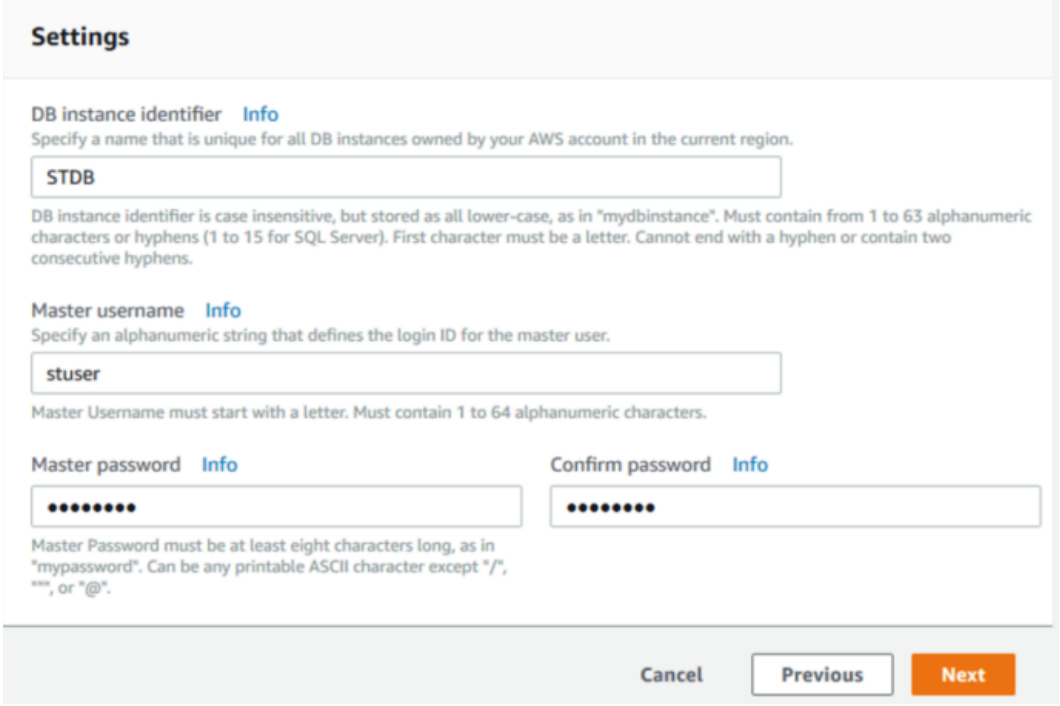
Create MS SQL database

1. Navigate to AWS Console -> Services.
2. Go to the Database section and select **RDS**.
3. Go to Instances and click **Launch DB Instance**.
4. Select Microsoft SQL Server -> SQL Server Enterprise Edition.
5. For Use Case, select **Production**.

6. Specify the DB Details.

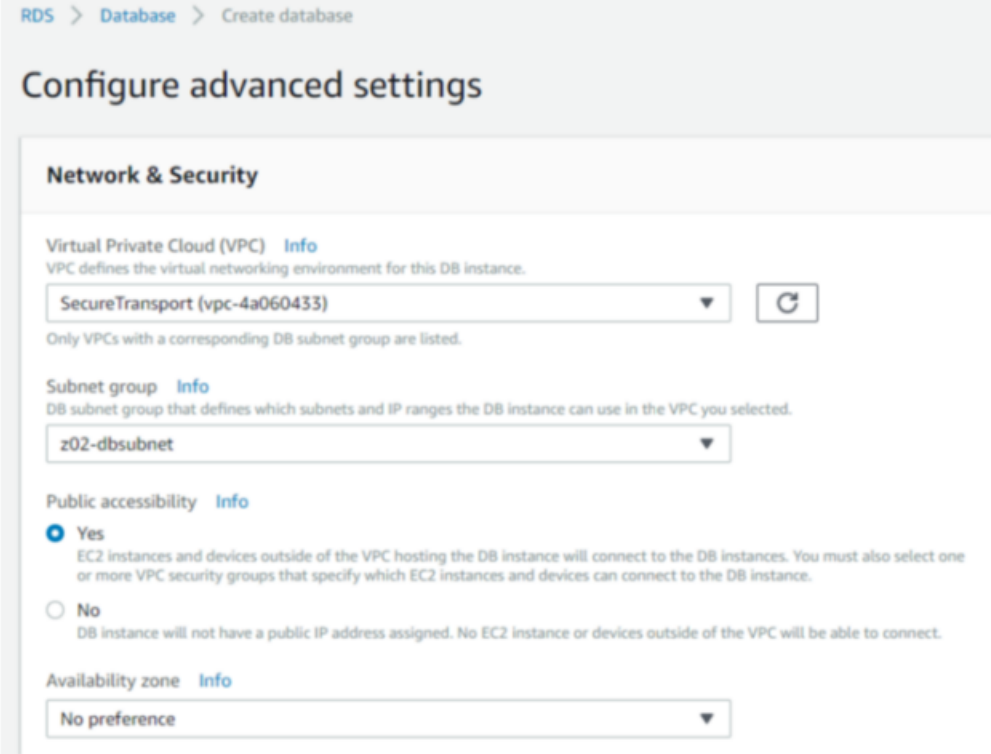


7. Add your DB Instance Identifier and Master user credentials.

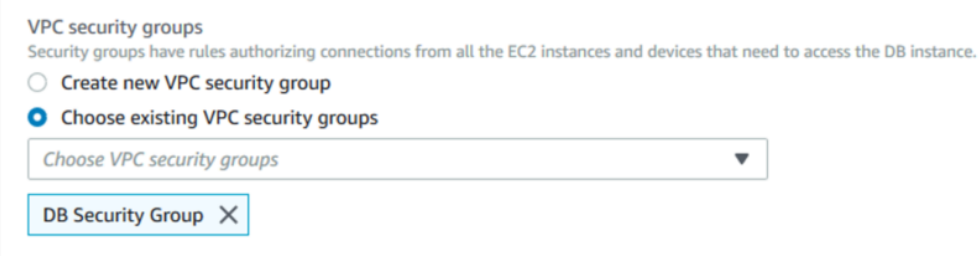


8. Click **Next**.
9. Configure Advanced Settings:
    - Launch the Database in your VPC.
    - Choose whether your database to be publicly accessible or no.
      Select Yes if you want to allow EC2 instances and devices outside the VPC that hosts the DB instance to connect to this DB instance. If you select No, Amazon RDS will not assign a public IP address to the DB instance, and no EC2 instance or devices outside of the VPC will be able to connect. If you select Yes, you must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.
      Click here to learn more.
    - Select the Availability Zone from the current region in which you want the DB instance created.

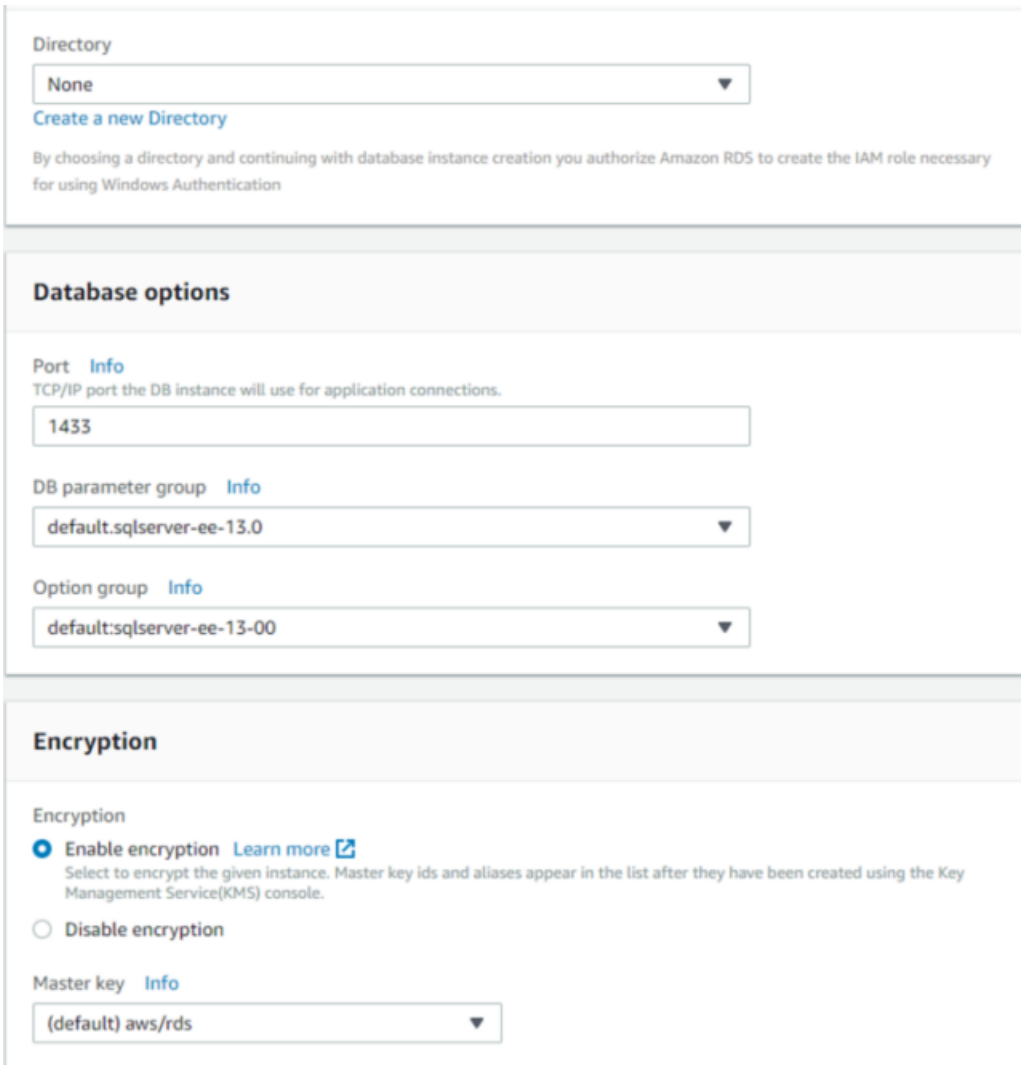
    Note: For high availability and fault tolerance, we recommend you to create a DB replica in different zones in the previous step.



10. Select the Database Security Group you created as described in the Create database Security Group subtopic.

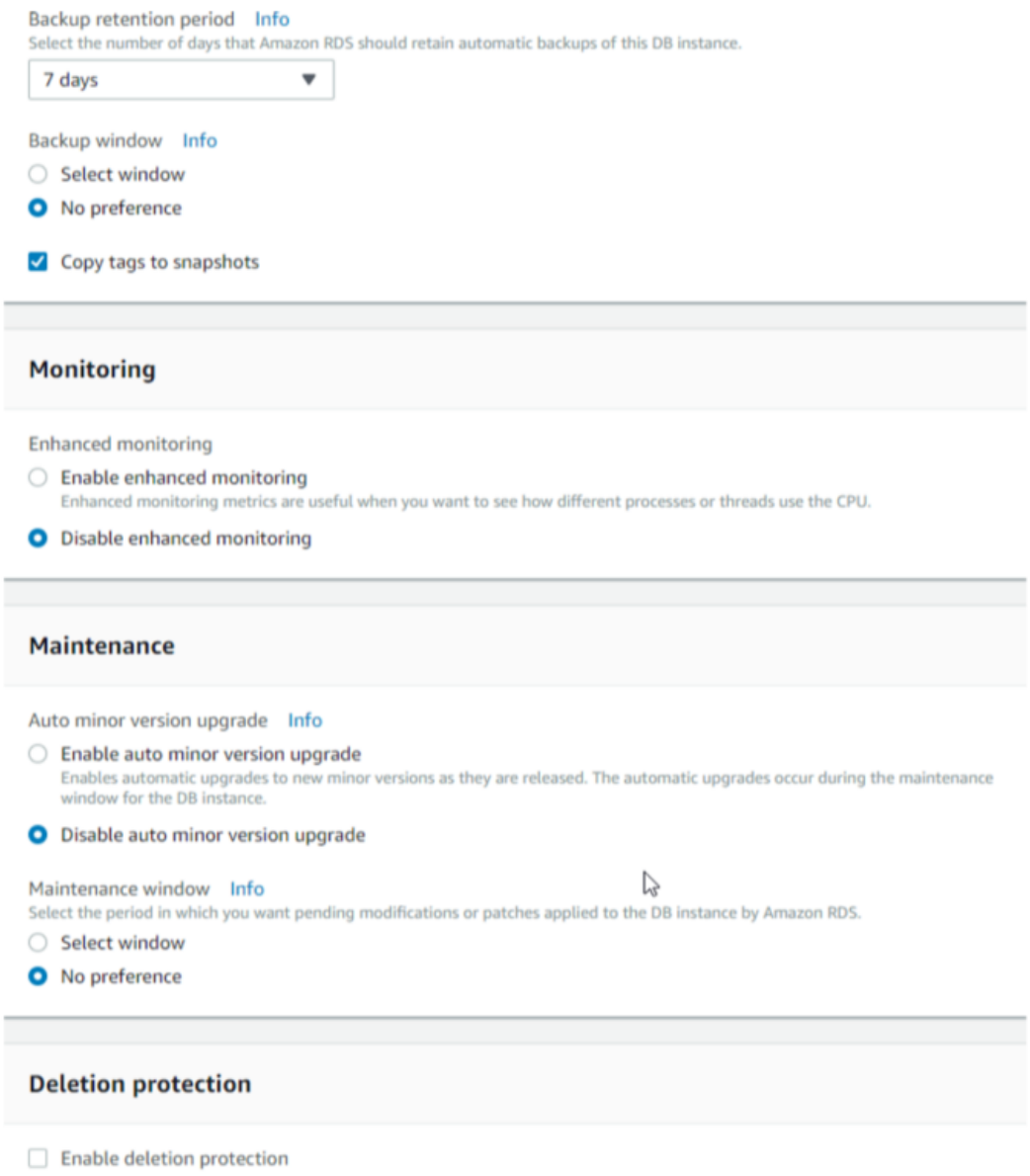


11. Add your Database Options:
    - Set a Database name.
    - Specify a Database port.
    - Select the previously created Parameter Group (only if you would like to force the SSL connections)
    - Leave the default Option Group.
    - Select the Encryption – enable or disable it.



- Select Backup retention period (not mandatory).
- Select Monitoring (not mandatory) – enable or disable it.
- Select Maintenance (not mandatory) – enable or disable it (not mandatory).

12. Click on **Create Database**.



13. When you finish with your setup, click **Launch DB instance**.

Using SSL with Microsoft SQL Server database

You can use Secure Sockets Layer (SSL) to encrypt connections between your client applications and your Amazon RDS DB instances running Microsoft SQL Server. SSL support is available in all AWS regions for all supported SQL Server editions.

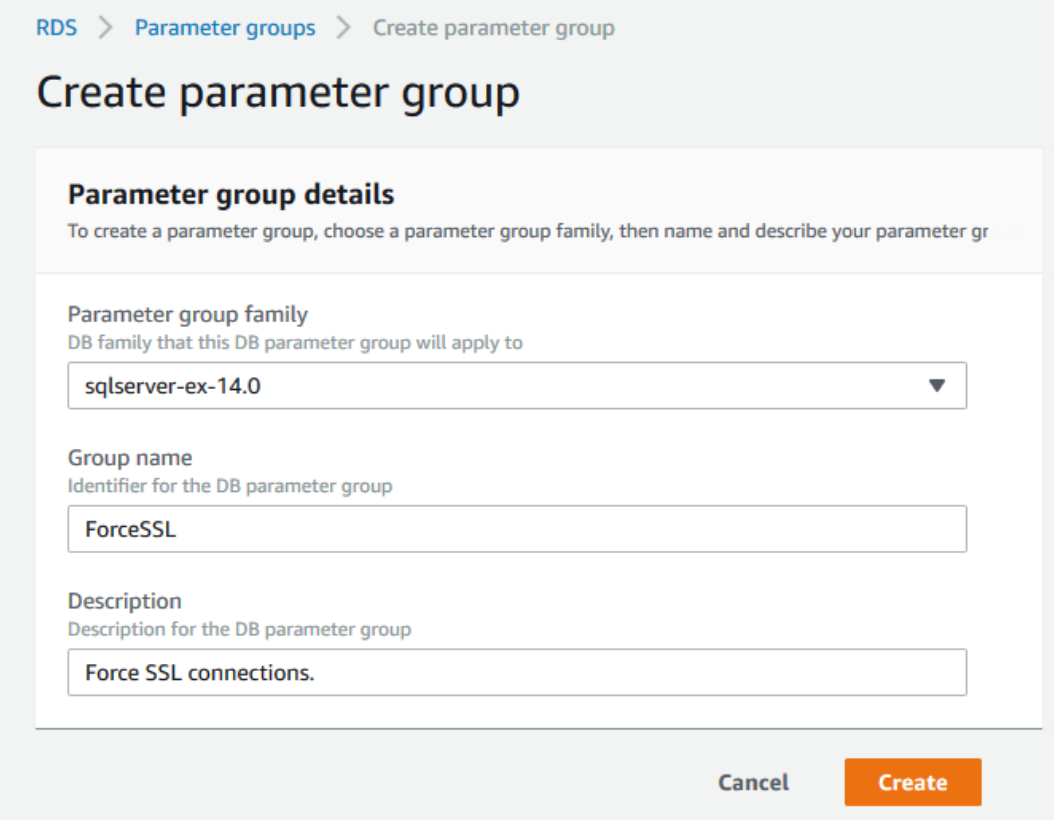
When you create a SQL Server DB instance, Amazon RDS creates an SSL certificate for it. The SSL certificate includes the DB instance endpoint as the Common Name (CN) for the SSL certificate to guard against spoofing attacks.

Force all SSL connections

The first way is to secure your database is to Force SSL for all connection – this happens transparently to the client, and the client does not have to do any work to use SSL.

You could do this by creating a Parameter Group:

1. Navigate to AWS console -> Services.
2. Go to the Database section and select RDS.
3. Go to Parameter groups and click Create parameter group.
4. Select the values in the following fields:
    - Parameter group family
    - Group name
    - Description (optional)



5. When you finish with your selections, click **Create**.
6. After creation, select the parameter group and go to Parameter group actions and click Edit.
7. Find the parameter that you would like to change and click Edit parameters.
    - Set the `rds.force_ssl` parameter to true to force connections to use SSL. The `rds.force_ssl` parameter is static, so after you change the value, you must reboot your DB instance for the change to take effect.

8. Insert the desired value and then click **Save changes**.

Assign the parameter group to your database

1. Navigate to RDS Instances.
2. Select your database and then go to Instance actions -> Modify.
3. Go to the Database options section and select your **DB parameter group** from the drop-down list.
4. Save and apply changes immediately.
5. Restart your database.

Encrypt Specific Connections

To use SSL from a specific client, you must obtain certificate, upload it to the client computer and then specify it during the SecureTransport installation.

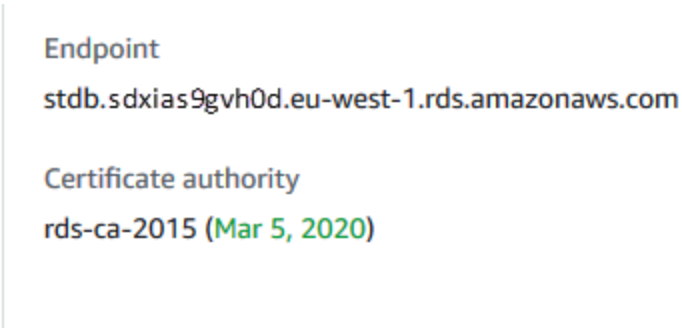
You could execute the following command to obtain the certificate file and certificate CN:

`openssl s_client -connect <host>:<ssl_port>` where `<host>` is the database Endpoint.

Learn more about Using SSL with a Microsoft SQL Server DB Instance.

Connect to your MS SQL database

1. Navigate to AWS console -> Services.
2. Go to the Database section and select **RDS**.
3. Navigate to **Instances**.
4. Select your newly created Database and then go to Instance Actions ->See Details.
5. Under the Security and network section, see Endpoint: you will need to copy & paste it in the next step.



6. Use Microsoft SQL Management Studio on your Administration Host.
   Add new connection and provide values for the following options:
   - **Server type** – select Database Engine
   - **Server name** – paste the newly created endpoint from the RDS MS SQL Database
   - **Authentication** – select SQL Server Authentication
   - **Login** – database administrator login name
   - **Password** – database administrator password

Create tables and set ownership of the MS SQL database

Use the following script to create a table:

```
USE master;

GO

CREATE DATABASE STDB ON PRIMARY (NAME=STDB1, FILENAME ='D:\RDSDBDATA\DATA\STDB.mdf',
MAXSIZE = 4GB, FILEGROWTH = 5MB);

ALTER DATABASE STDB ADD FILEGROUP ST_DATA;

ALTER DATABASE STDB ADD FILEGROUP ST_FILETRACKING;

ALTER DATABASE STDB ADD FILEGROUP ST_SERVERLOG;

ALTER DATABASE STDB ADD FILE (NAME='ST_DATA_STDB', FILENAME = 'D:\RDSDBDATA\DATA\ST_
DATA_STDB.ndf', SIZE = 200MB, FILEGROWTH=50MB) TO FILEGROUP ST_DATA;

ALTER DATABASE STDB ADD FILE (NAME='ST_FILETRACKING_STDB',
FILENAME='D:\RDSDBDATA\DATA\ST_FILETRACKING_STDB.ndf', SIZE = 50MB, FILEGROWTH=10MB) TO
FILEGROUP ST_FILETRACKING;

ALTER DATABASE STDB ADD FILE (NAME='ST_SERVERLOG_STDB', FILENAME='D:\RDSDBDATA\DATA\ST_
SERVERLOG_STDB.ndf', SIZE = 200MB, FILEGROWTH=10MB) TO FILEGROUP ST_SERVERLOG;

ALTER DATABASE STDB SET READ_COMMITTED_SNAPSHOT ON;

GO
```

Use the following script to create user login:

```
USE STDB;

CREATE LOGIN STDB WITH PASSWORD='STDB', DEFAULT_DATABASE=STDB, CHECK_POLICY=OFF, CHECK_
EXPIRATION=OFF;

GO
USE STDB;

EXEC sp_grantdbaccess 'STDB', 'STDB';

EXEC sp_addrolemember 'db_ddladmin', 'STDB';
```

```
EXEC sp_addrolemember 'db_datareader', 'STDB';
EXEC sp_addrolemember 'db_datawriter', 'STDB';
GO
```

Alternative to RDS Service

If you would prefer not to use the AWS RDS Service, you can easily replace it with two RHEL Instances with an Oracle 12 EE or MS SQL Server 2017 EE database setup on each.

The Oracle or MS SQL RHEL instances should follow the same rules:

1. One instance in each availability zone.
2. Each instance in a separate private subnet.
3. Both instances in one Database Security Group as defined in *Security Groups and Network Access Lists*.

Launch RHEL instances 6

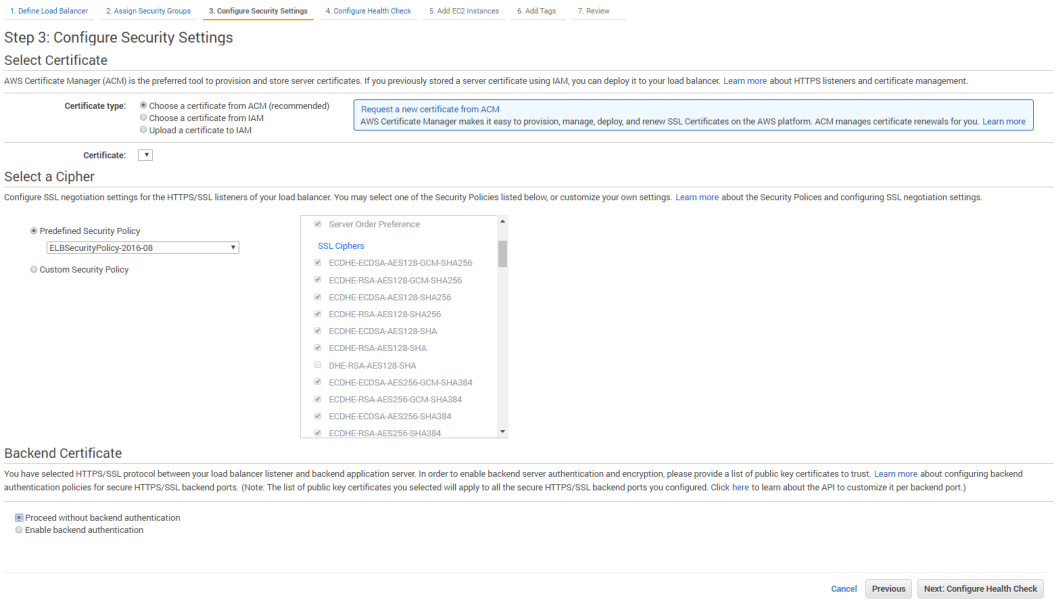
For the purpose of our setup, we need seven RHEL instances in total:

- Two SecureTransport Servers – one in each Private Subnet
- Two SecureTransport Edges – one in each Public Subnet
- Two GlusterFS Servers – one in each Private Subnet
- One instance to administer and access the above instances – Administration Host in one of the public subnets

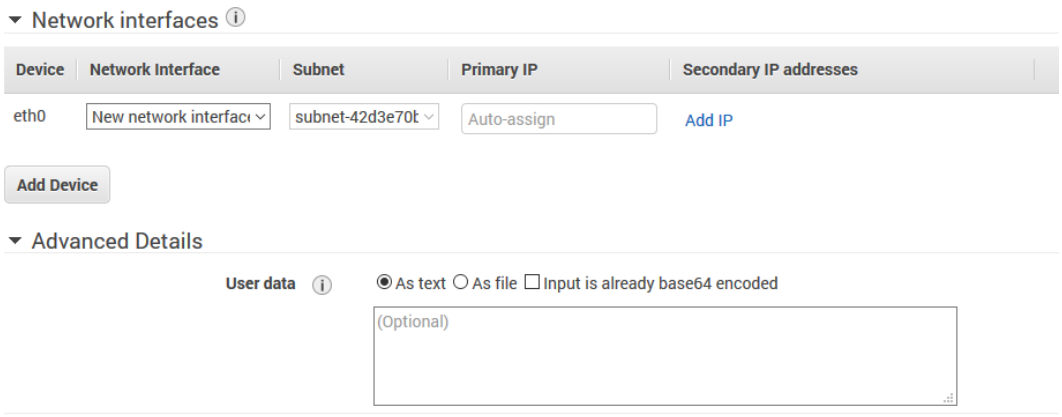
To launch a RHEL instance, follow these steps:

1. Navigate to AWS console -> Services.
2. Go to the Compute section and select **EC2**.
3. Select Instances and then click **Launch Instance**.
4. Choose an Amazon Machine Image and select Red Hat Enterprise Linux 7.4 (HVM), SSD Volume Type - ami-bb9a6bc2 (64 bit) .
5. Choose an Instance Type according to the *Minimum UNIX hardware requirements* in the *SecureTransport5.4 Installation Guide*. We recommend a minimum setup of mx3.large with 4 cores & 15GB RAM & 2x40GB SSD.
   Learn more about Instance Types.
6. Configure Instance Details by providing the following options:
   - **Number of instances:** 1
     Launch the instances one by one because of the different settings.
   - **Network:** select your VPC from the drop-down list.
   - **Subnet:** select a subnet from the drop-down list.
   - **Auto-assign Public IP:** Requests a public IP address from Amazon's public IP address pool to make your instance accessible from the Internet.
   - For instances in the public subnets, select **Enabled**.
   - **IAM role:** An IAM role automatically deploys AWS credentials to resources that assume it. Select the instance profile that contains the required IAM role.
     Learn more about IAM Roles.
   - **Shutdown behavior:** Specify the instance behavior when an OS-level shutdown is performed. Instances can be either terminated or stopped.
   - **Enable termination protection:** You can protect instances from being accidentally terminated.
   - **Monitoring:** Enables you to monitor, collect, and analyze metrics about your instances through Amazon CloudWatch.

- **EBS-optimized instance:** Enables additional, dedicated throughput between Amazon EC2 and Amazon EBS, and therefore improved performance for your Amazon EBS volumes.
- **Tenancy:** You can choose to run your instances on physical servers fully dedicated for your use.

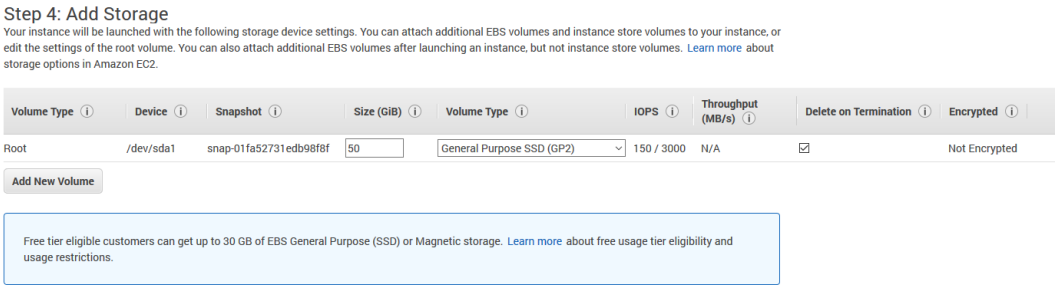


- **Network Interfaces:** You can attach one more network interface to your instance during launch.
- **Advanced Details:** You can specify user data to configure an instance or run a configuration script during launch.

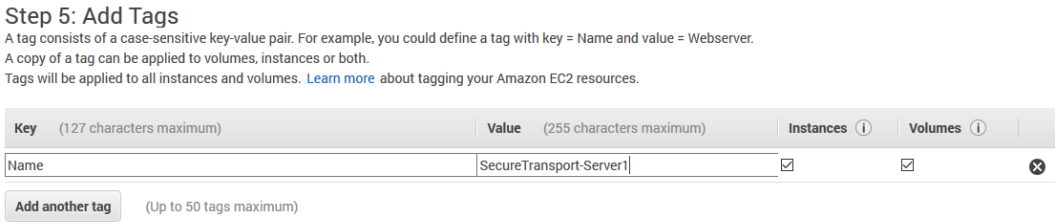


Learn more about Security Groups.

7. Add Storage – specify storage device settings.

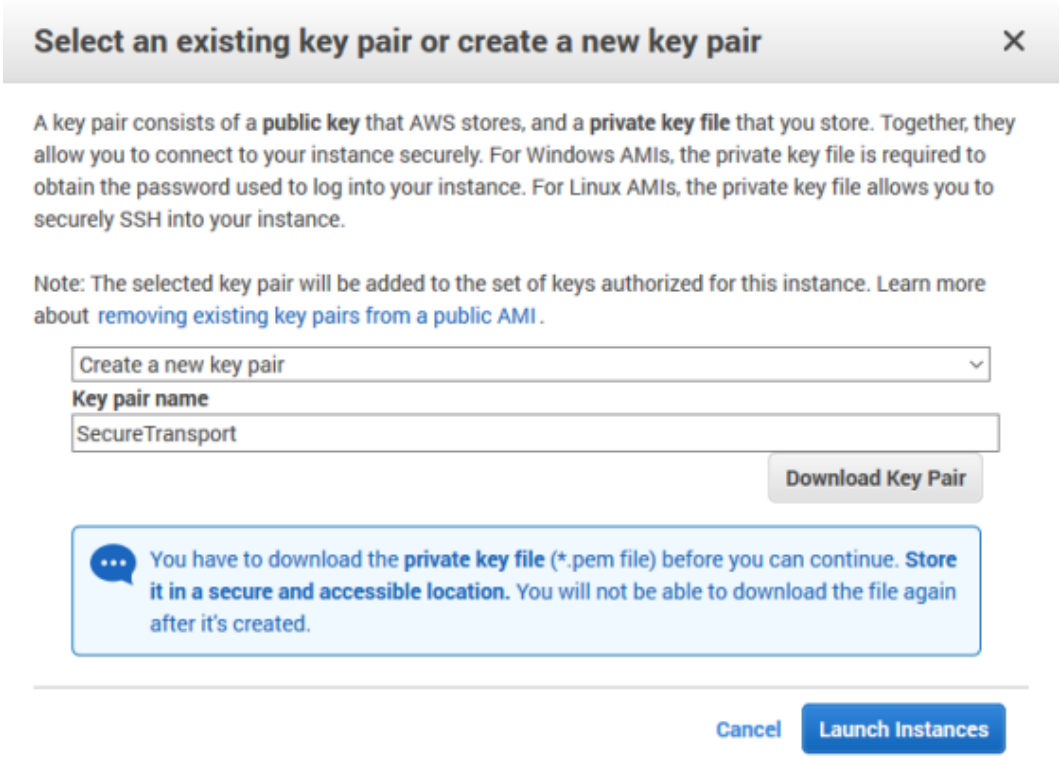


8. Add Tags – a tag consists of a case-sensitive key-value pair.
   For example, you could define a tag with key = Name and value = SecureTransport-Server1.



9. Configure Security Group – create new or choose an existing one.
   A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance.
   Note: SecureTransport Servers, SecureTransport Edges, GlusterFS Servers and Administration Host have specific security groups.
   Learn more about Security Groups.
10. Review and Launch your instance.
11. Select an existing key pair or create a new one.
    - Create a new key pair – type a key pair name and then click **Download Key Pair**.
    - Choose an existing key pair – select a key pair from the drop-down menu and select the **I acknowledge...** check-box.

    After download, you will be able to click **Launch Instances**.



Launch an instance for the Administration Host

The Administration Host is an instance of a virtual machine you launch in one of the public subnets in your VPC. You can use this machine to perform the following activities:

- Perform SecureTransport installations on launched instances.
- Add configurations on your launched instances.
- Perform maintenance and administration tasks on your instances.

Follow the steps for launching RHEL Instance with the following specifics:

1. Choose an Amazon Machine Image and select the operating system that will be most suitable for your administering needs.
2. Choose an Instance Type that will be most suitable for your administering needs.
3. Go to Instance Details -> Subnet.
4. Choose one of the public subnets from one of the availability zones.
5. Go to Configure Instance Details and select to enable Auto-assign Public IP (depending on the connectivity you would like to have to this host).
6. Add storage – enter a value for storage size sufficient for your needs.
7. Add Tags – specify unique names, so you can easily distinguish between instances.

8. Security Groups – assign the instance to the previously created Administration Host Security Group as described in the *Create Security Groups and Network Access Lists on page 12* section.

Launch SecureTransport Edge instances

Follow the steps for launching RHEL Instance with the following specifics:

1. Go to Instance Details -> Subnet.
2. First Server – choose the public subnet from availability zone eu-west-1 b.
3. Second Server – choose the public subnet from availability zone eu-west-1 c.
4. Go to Configure Instance Details and select to enable Auto-assign Public IP (if you need your Edge servers to be directly accessible from the Internet).
5. Add storage.
6. Enter a value for storage size sufficient for your SecureTransport Edge needs.
7. Add Tags.
8. Specify unique names, so you can easily distinguish between instances.
9. Specify the Security Groups.

Assign the instances to the previously created SecureTransport Edge Security Group as described in the *Create Security Groups and Network Access Lists on page 12* section.

Launch SecureTransport Server Instances

Follow the steps for launching RHEL Instance with the following specifics:

1. Configure Instance Details -> Subnet.
2. First Server - choose the private subnet from availability zone eu-west-1 b.
   Second Server - choose the private subnet from availability zone eu-west-1 c.
3. Configure Instance Details - > Auto-assign Public IP: Disabled.
4. Add storage.
5. The default value of 10GB would be insufficient, so you can increase it to 20GB or more.
6. Add Tags.
7. Specify unique names, so you can easily distinguish between instances.
8. Specify the Security Groups.
9. Assign all SecureTransport Servers to the previously created Security Group as described in the *Create Security Groups and Network Access Lists on page 12* section.

Set up GlusterFS Servers

Follow the steps for launching RHEL Instances with the following specifics:

1. Configure Instance Details -> Subnet.
2. First Server - choose the private subnet from availability zone eu-west-1 b.
3. Second Server - choose the private subnet from availability zone eu-west-1 c.
4. Configure Instance Details - > Auto-assign Public IP: Disabled.
5. Add Tags.
6. Specify unique names, so you can easily distinguish between instances.
7. Specify the Security Groups.
8. Assign both GlusterFS servers to the previously created GlusterFS Security Group.

Attach additional volumes

You will need to attach additional volumes to the GlusterFS instances, as they need two or more virtual disks.

Create two volumes:

1. Navigate to AWS console -> Services.
2. Go to the Compute section and select **EC2**.
3. Go to the Elastic Block Store and select **Volumes**.
4. Click **Create a volume**.
5. Leave the default settings except the Availability Zone.
6. Place each volume in the same zone as the GlusterFS server it will be attached to.
7. Attach each volume to a GlusterFS server.
8. Select the volume and then go to Actions -> Attach volume to select the GlusterFS Server ID.
9. Click **Attach**.

Install GlusterFS

Follow the instructions for installing GlusterFS in the GlusterFS Documentation.

Connect to your VPC 7

There are a few ways to connect to a VPC, and the right one for you depends on your use case and preferences. You can use the following protocols or services to connect to a VPC:

- VPN
- AWS Direct Connect
- VPC peering
- VPC endpoints
- EC2 ClassicLink

VPN

A virtual private network (VPN) connection is established to an AWS-managed virtual private gateway (VPG). For more information, see the *Set up VPN connection on page 62* topic.

A virtual private gateway is the VPN device on the AWS side of the VPN connection. After you have created your VPN, you can download the IPsec VPN configuration from the Amazon VPC console to configure the firewall or device in your local network that will connect to the VPN.

AWS offers a managed VPN service, but you can also use a third-party software VPN solution. The latter is suitable if you need to have full access and management of the AWS side of your connection.

For more information about VPN connections, see VPN Connections.

AWS Direct Connect

Direct Connect creates a direct, private connection from your on-premises data center to AWS, letting you establish a 1-gigabit or 10-gigabit dedicated network connection using Ethernet fiber-optic cable. For more information, see What is Direct Connect?

Direct Connect is priced per port-hour, with additional data transfer rates that vary by region. For more detailed pricing information, see the Direct Connect pricing page.

VPC peering

VPC peering allows you to connect two VPCs using each VPC's private IP address. This makes it appear as if the 2 VPCs are on the same network.

This option is recommended for connecting VPCs within a region or across AWS accounts. Because these connections do not rely on physical hardware, they are not subject to issues with single-point of failure or network bandwidth bottlenecks. You can find out more at VPC Peering.

VPC endpoints

VPC endpoints enable you to create a private connection between your VPC and another AWS service, without the need for Internet access. A VPC endpoint enables instances in your VPC to use private IP addresses to communicate with resources in other services. For more information, see VPC Endpoints.

EC2 ClassicLink

ClassicLink allows you to link an EC2-Classic instance to a VPC in your account within same region, without using public IP addresses or Elastic IP addresses to enable communication between instances. You can associate VPC security groups with the EC2-Classic instance and enable a connection between the EC2-Classic instance and instances in your VPC by using a private IP address.

This option is available to users with accounts that support the EC2-Classic platform and can be used with any EC2-Classic instance. For more information, see ClassicLink.

Set up VPN connection

A VPN configuration allows you to create a secure connection from a client computer to your AWS virtual private network. Amazon VPC provides different VPN connection options to your VPC depending on your needs. You can find an overview of the supported VPN connectivity options in the Amazon Documentation.

Click here for detailed information about the AWS VPC Connectivity Options.

One of the connectivity options, described in this guide, is to configure an AWS hardware VPN. In this case you create the following items from the AWS console:

- **Customer gateway**

The VPN endpoint on your network. Here you specify your customer gateway device public IP address and autonomous system number (ASN) if you intend to use the Border Gateway Protocol (BGP) or dynamic routing.

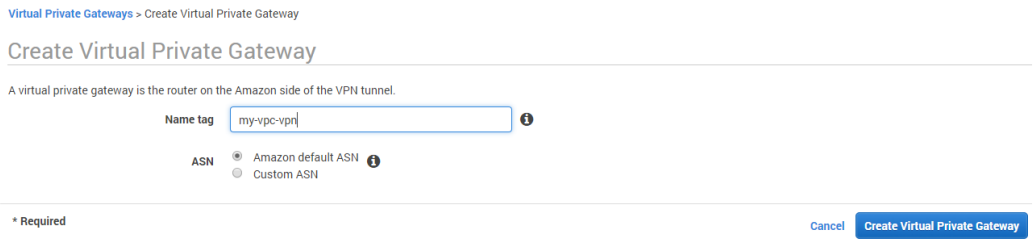
- **Virtual private gateway**

The VPN endpoint on your AWS VPC.

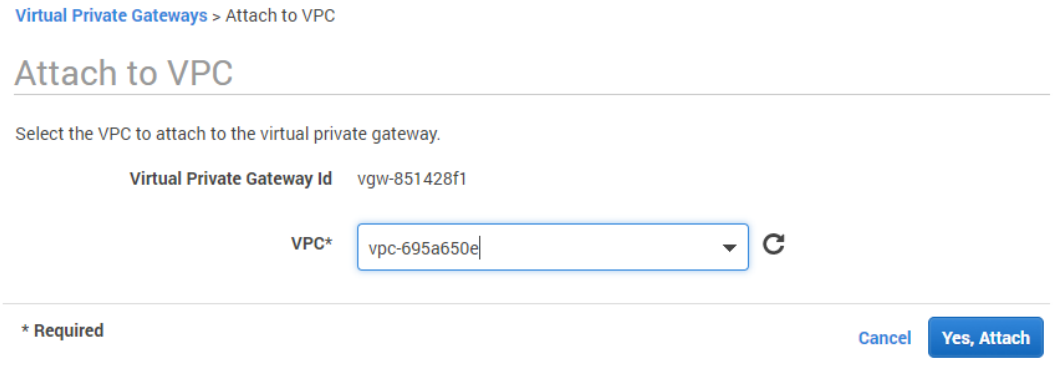
- **VPN connection**

The connection between your network and your AWS VPC. You can automate configuration of the customer gateway device for your network with a configuration file that is generated when you create your Customer Gateway and Virtual Private Gateway.

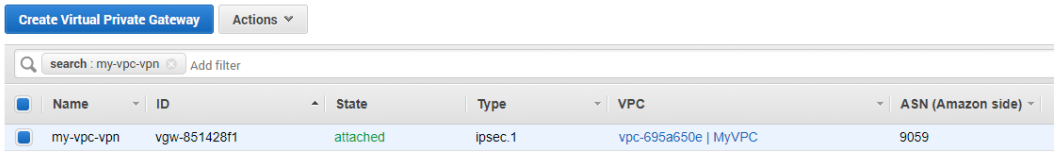
1. Create a customer gateway:
    a. Open the Amazon VPC console.
    b. In the navigation pane, under VPN Connections, choose Customer Gateways.
    c. Choose Create Customer Gateway.
        - Enter a meaningful name for the customer gateway.
        - Choose an option for Static or Dynamic routing.
        - Enter the public IP address of your customer gateway device.
        - Enter your BGP ASN if you selected the option for dynamic routing.
    d. Click **Yes, Create**.



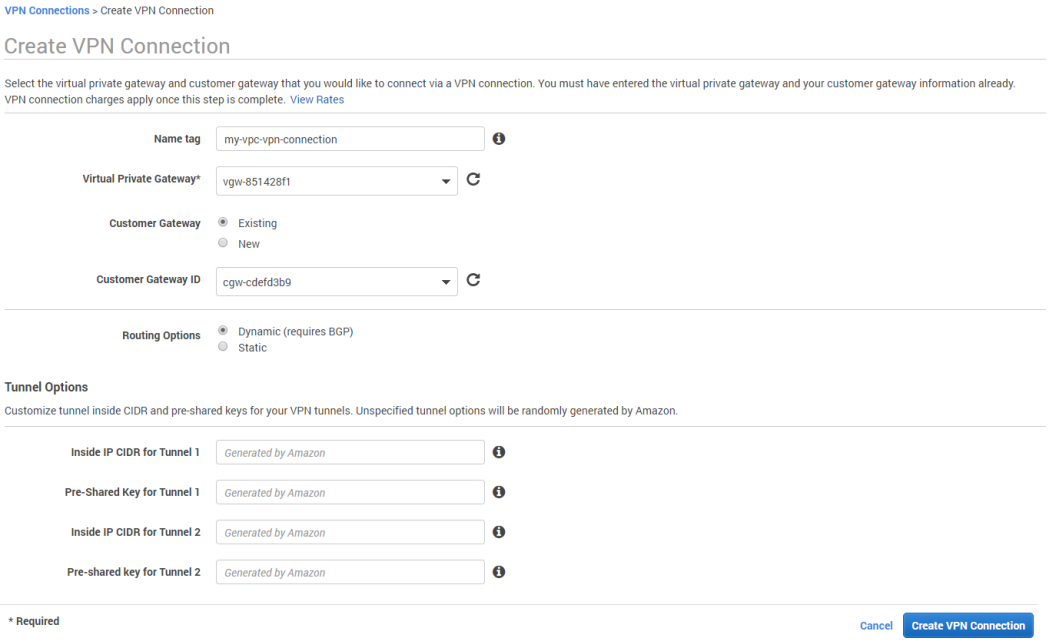
2. Create a virtual private gateway:
    a. In the VPC console, under VPN Connections, choose Virtual Private Gateways.
    b. Choose Create Virtual Private Gateway.
    c. Enter a meaningful name for the virtual private gateway.
    d. Click **Yes, Create**.
    e. Select the new virtual private gateway and open the context menu (using right-button click with the mouse). Click **Attach to VPC**.
    On the newly opened screen, select your VPC and then click **Yes, attach**.



Initially, the virtual private gateway is in "attaching" state for some time and then after success the state is updated to "attached".



3. Create a VPN connection:
    a. In the VPC console, under VPN Connections, choose VPN Connections.
    b. Select Create VPN Connection.
        - Enter a meaningful name for the VPN connection.
        - For Virtual Private Gateway, select the virtual private gateway you just created.
        - For Customer Gateway, select the entry you just created.
        - For Routing Options, choose Dynamic or Static. If you choose static routing, specify the Static IP Prefixes of the appropriate private network(s) on your LAN.
        - Click **Yes, Create**.



4. Get the VPN connection configuration and configure your customer gateway:
    a. In the VPC console, under VPN Connections, select VPN Connection.
    b. Select the VPN connection you created, and then choose Download Configuration.
    c. In the Download Configuration dialog box, select the vendor for the customer gateway, the platform, and the software version, and then click **Yes, Download**.
    d. Save the text file that contains the VPN configuration and give it to your network administrator, along with the Amazon VPC Network Administrator Guide. The VPN won't work until the network administrator configures the customer gateway.
5. Test your VPN connection as described in the Amazon Documentation.

Set up Enterprise Cluster with streaming 8

Prerequisites

The following list outlines the prerequisites necessary for the described SecureTransport Enterprise cluster setup.

1. Two RHEL Instances for SecureTransport Servers.
2. Two RHEL Instances for SecureTransport Edges.
3. Two RHEL Instances with GlusterFS Servers installed.
4. One Administration Host (optional).
5. One Microsoft SQL Server.

For correct setup of all your instances, please refer to the *Prerequisites -> UNIX-based platforms* topic in the *SecureTransport 5.4 Installation Guide*.

Note: Make sure to have the following prerequisite installation package added:

```
ld-linux.so.2 library package
```

Install SecureTransport

1. Install SecureTransportServers as described in the *SecureTransportInstallation Guide: Install SecureTransport Server in an Enterprise Cluster or to use an external database.*
2. Install SecureTransportEdges as described in the *SecureTransport Installation Guide: Install SecureTransport Server to use the embedded database* and in the SecureTransport Administrator's Guide: *SecureTransport Edge synchronization*
3. Install GlusterFS client on both SecureTransport Servers and mount the GlusterFS volume on the client side.

For reference see GlusterFS Documentation.

Set up Classic Load Balancer 9

Amazon Web Services Cloud provides three types of Load Balancers – Application, Network and Classic. The deployment of SecureTransport on Amazon VPC has been verified with the Classic Load Balancer. You can load balance HTTP/HTTPS applications and use Layer 7-specific features, such as X-Forwarded and sticky sessions. You can also use strict Layer 4 load balancing for applications that rely purely on the TCP protocol. The load balancer takes requests from clients and distributes them across the EC2 instances that are registered with the load balancer.

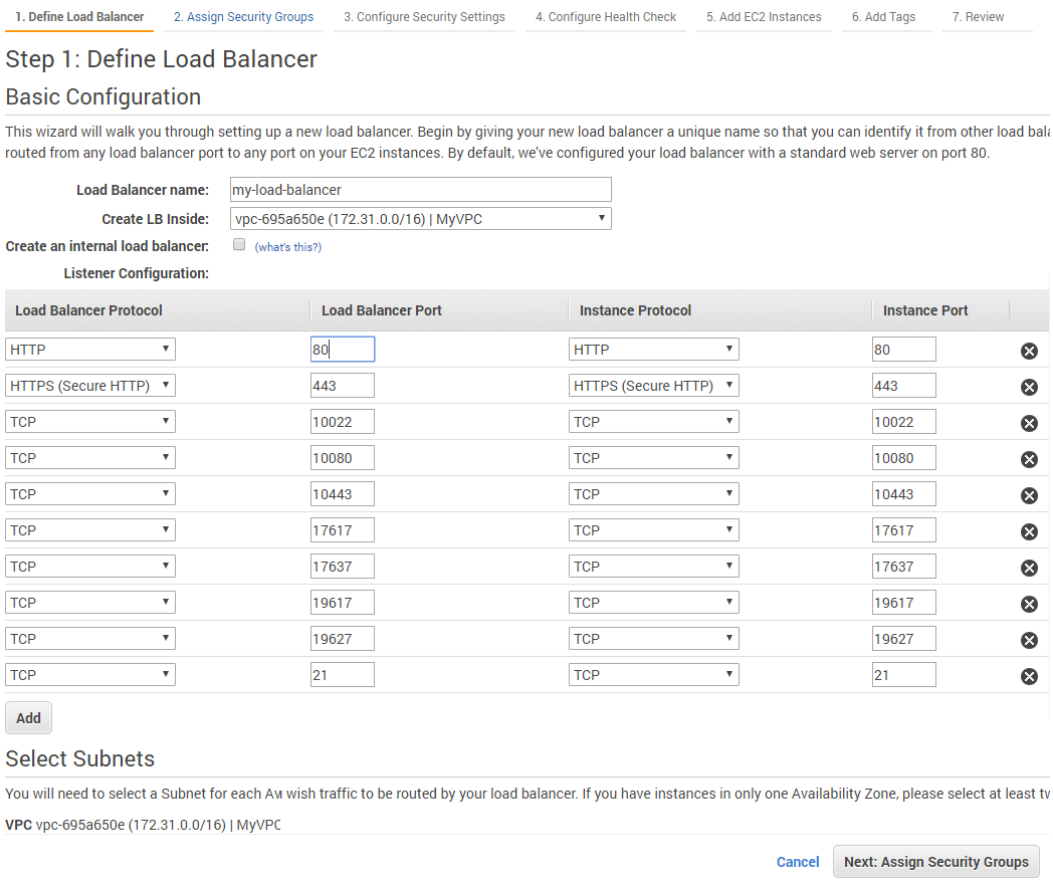
When you create a load balancer in a VPC, you can either make it an internal load balancer, or an Internet-facing load balancer. You create an Internet-facing load balancer in a public subnet.

When you create your load balancer, you must configure listeners, health checks, and register back-end instances. You configure a listener by specifying a protocol and a port for front-end (client to load balancer) connections, and a protocol and a port for back-end (load balancer to back-end instances) connections. You can configure multiple listeners for your load balancer.

If you specify that the HTTPS listener sends requests to the instances on port 80, the load balancer terminates the requests and communication from the load balancer to the instances is not encrypted. If the HTTPS listener sends requests to the instances on port 443, communication from the load balancer to the instances is encrypted.

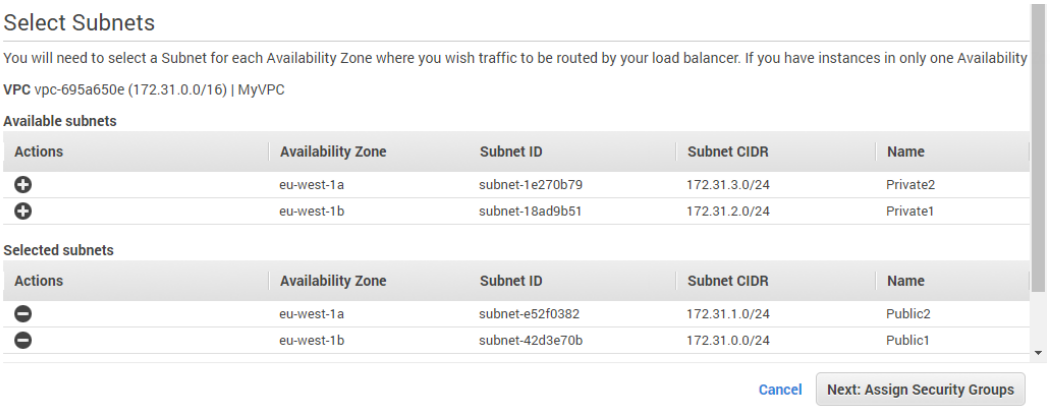
Follow these steps to create Classic Load Balancer redirecting requests to SecureTransport instances in Amazon VPC:

1. Log in to AWS.
2. Navigate to Services->EC2->Load Balancing->Load Balancers.
3. Click the **Create Load Balancer** located in the top of the page.
4. Choose **Classic Load Balancer**.
5. Add **Basic Configuration**:
    a. Enter Load Balancer name.
    b. Create the LB inside and choose your VPC.
    c. Leave the "Create an internal load balancer" options unselected.
    d. In the "Listener Configuration" add listeners by specifying a protocol and a port for front-end (client to load balancer) connections, and a protocol and a port for back-end (load balancer to back-end instances) connections.
    Note that the listeners in the image are configured with default ports or ports specific for the test setup. Please change/add listeners according to your specific setup.
    Check the FTP does not work through the firewall section in the SecureTransport Administrator's Guide if you want to configure FTP listeners.



e. Navigate to the "Select Subnets" section on the same page.

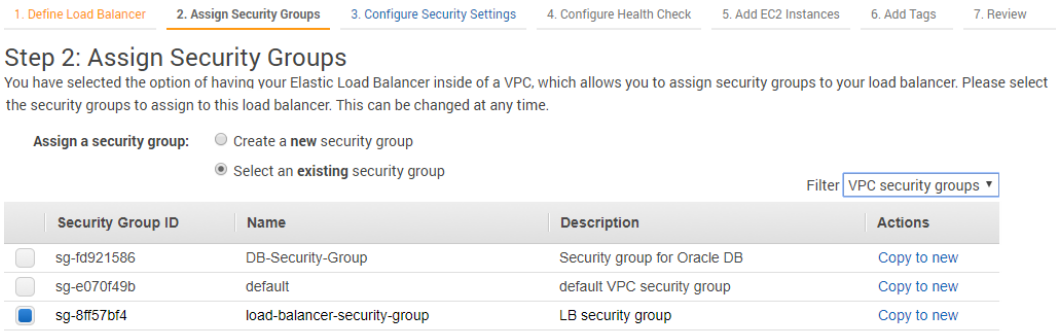
f. Choose your public subnets from both availability zones.



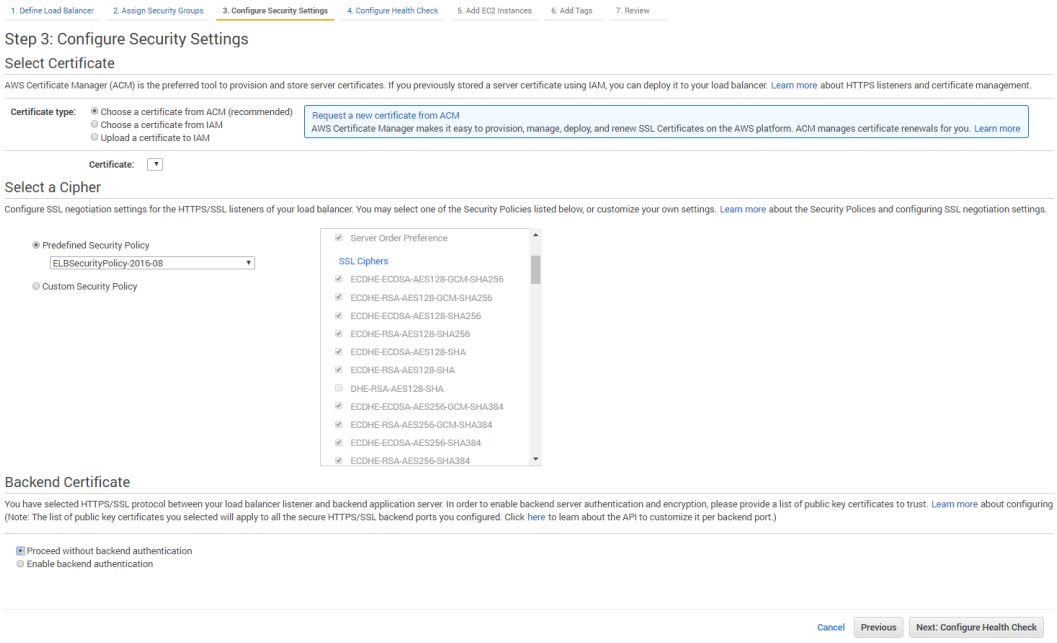
g. Choose "Assign Security Groups" and move to the next phase.

6. On the next step "Assign Security Groups"-> go to "Select an existing security group" and choose the security group you prepared previously for your load balancer, described in the *Security*

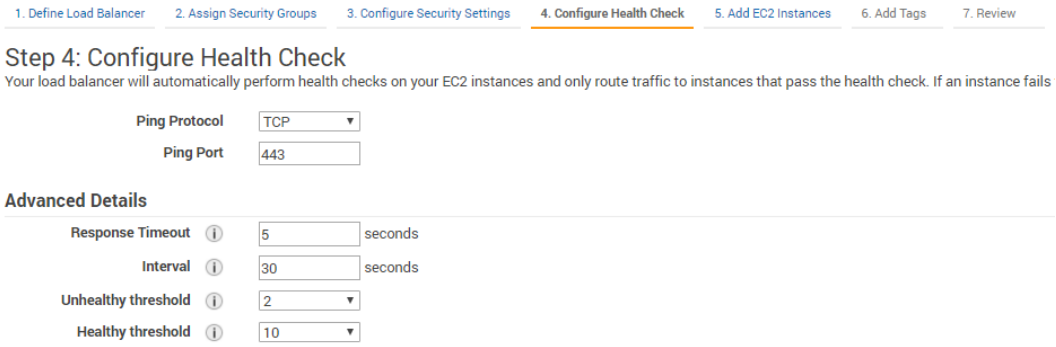
Groups on page 12 section in this guide.



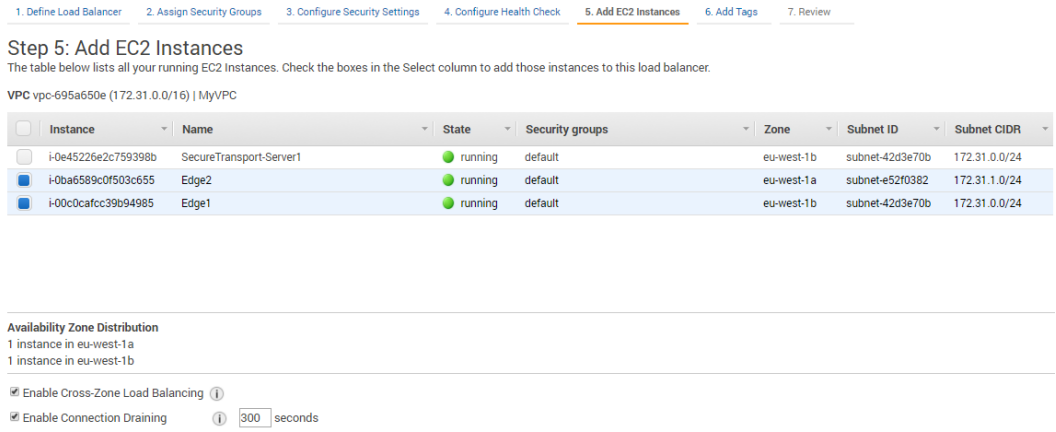
7. On the next stage **"Configure Security Settings"** choose certificate for the HTTPS listener - Choose a certificate from ACM (recommended)/Choose a certificate from IAM and leave default values for the rest of the security settings and proceed to the next stage (modify the security settings according to your needs).



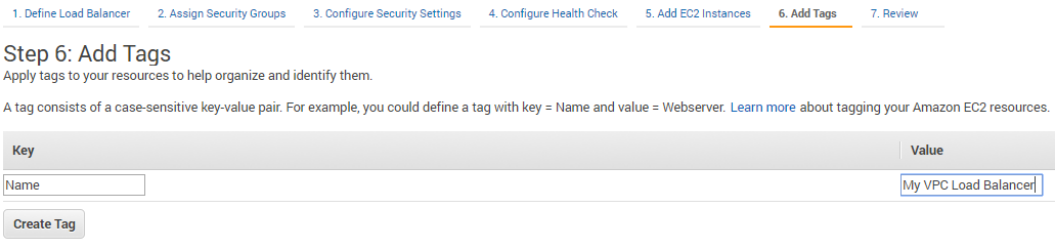
8. **Configure Health Check** on the next step - configure ping protocol and port and choose **Next: Add EC2 instances**.
   Learn more about HealthCheck in the Amazon Documentation.



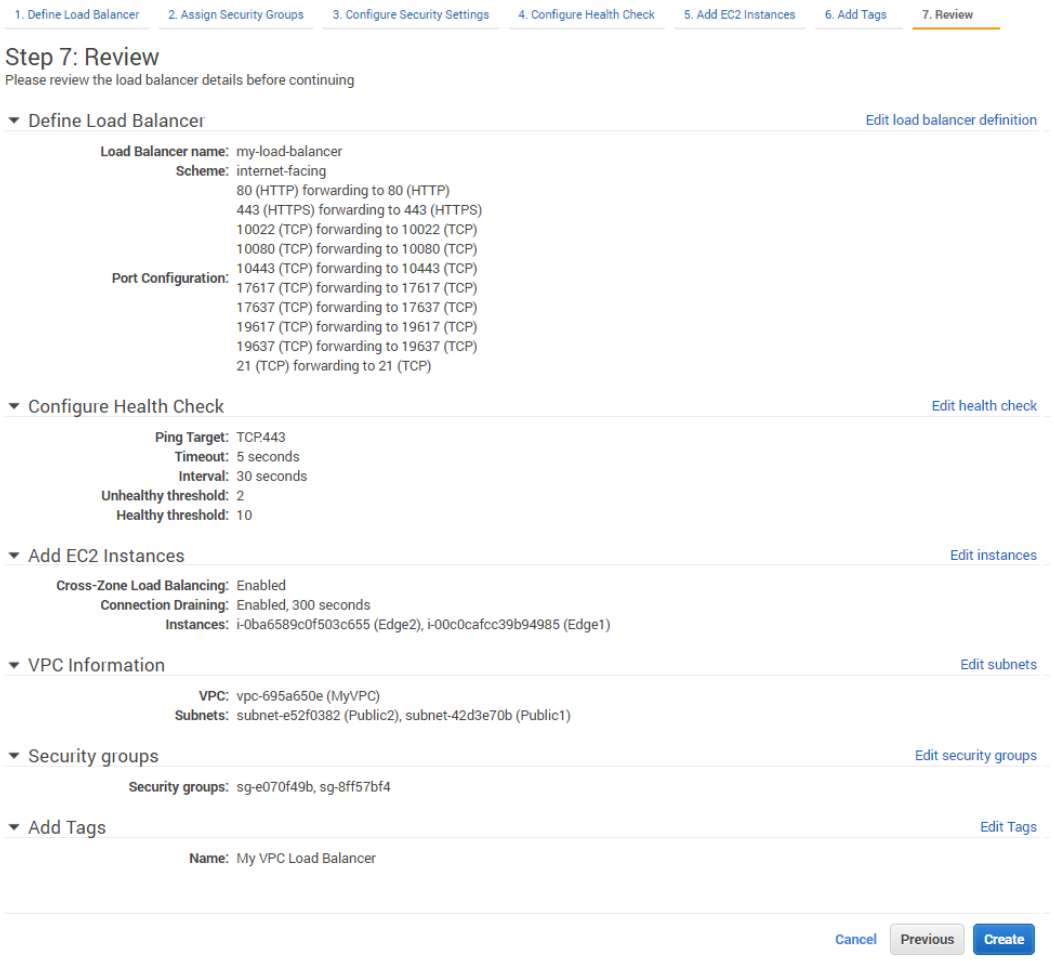
9. **Add EC2 Instances**: choose your two SecureTransport Edge instances and choose **Next: Add Tags**.



10. On the next stage **Add Tags** add a tag for example with key **Name** and a value something unique and meaningful.



11. Review your load balancer settings and choose **Create**.



12. After successful creation of the load balancer you are redirected to a page confirming your load balancer creation status.

13. Configure Stickiness. If you have HTTP/HTTPS listeners in your load balancer configuration you need to edit one more setting called Cookie Stickiness. Edit the setting after the load balancer is successfully created.
    As per AWS Documentation:
    By default, a Classic Load Balancer routes each request independently to the registered instance with the smallest load. However, you can use the *sticky session* feature (also known as *session affinity*), which enables the load balancer to bind a user's session to a specific instance. This ensures that all requests from the user during the session are sent to the same instance.
    The key to managing sticky sessions is to determine how long your load balancer should consistently route the user's request to the same instance. If your application has its own session cookie, then you can configure Elastic Load Balancing so that the session cookie follows the duration specified by the application's session cookie. If your application does not have its own session cookie, then you can configure Elastic Load Balancing to create a session cookie by specifying your own stickiness duration.
    Check the Configure Sticky Sessions for Your Classic Load Balancer in the AWS documentation for more details.

To Configure Stickiness, follow these steps:

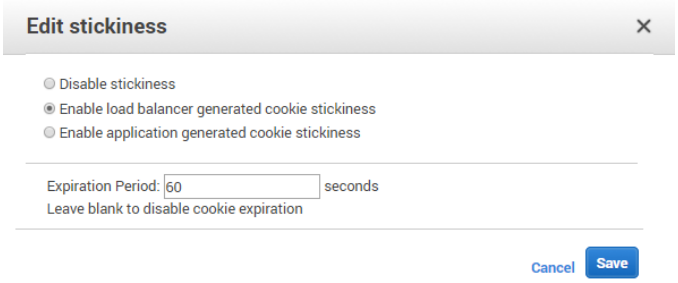
a. Log in to AWS.

b. Navigate to **Services->EC2->Load Balancing->Load Balancers**.

c. Choose your load balancer from the list.

d. In the Description section for your load balancer choose "Edit Stickiness" for HTTP and HTTPS Port Configuration.

e. In the Edit stickiness window displayed choose **"Enable load balancer generated cookie stickiness"** and enter expiration period in seconds.
(SecureTransport works with Load Balancer generated cookie stickiness).



14. **Instance HealthCheck -** your instances should pass the health check you defined in order for your load balancer to become operational.

    1. Log in to AWS console.
    2. Navigate to Services->EC2->Load Balancing->Load Balancers.
    3. Choose your load balancer from the list.
    4. In the **Instances** section for your load balancer check the Status of your instances.
    5. During the health check execution their Status is "OutOfService".
    6. If the check passes successfully the instances' Status changes to "InService".

Now, your load balancer is ready to fetch requests and distribute them among your SecureTransport Edge instances in the two availability zones in Amazon Web Services Cloud.
If you would like to use a friendly DNS name to access your load balancer, instead of the default DNS name automatically assigned to your load balancer, you can create a custom domain name and associate it with the DNS name for your load balancer - for more information check the Amazon Web Services Documentation.

Criteria for a successful setup

10

For a successful setup, make sure you have the following minimum criteria covered:

1. Both SecureTransport Server and Edge Clusters are synchronized and work as expected.
2. Successful end-user login over the desired protocols via the Load Balancer or SecureTransport Edge.
3. Successful file operations and transfers over the desired protocols.