Working with a DB instance in a VPC

Archived: 2026-04-05 15:37:42 UTC

Your DB instance is in a virtual private cloud (VPC). A VPC is a virtual network that is logically isolated from other virtual networks in the AWS Cloud. Amazon VPC makes it possible for you to launch AWS resources, such as an Amazon RDS DB instance or Amazon EC2 instance, into a VPC. The VPC can either be a default VPC that comes with your account or one that you create. All VPCs are associated with your AWS account.

Your default VPC has three subnets that you can use to isolate resources inside the VPC. The default VPC also has an internet gateway that can be used to provide access to resources inside the VPC from outside the VPC.

For a list of scenarios involving Amazon RDS DB instances in a VPC and outside of a VPC, see Scenarios for accessing a DB instance in a VPC.

Topics

In the following tutorials, you can learn to create a VPC that you can use for a common Amazon RDS scenario:

- Tutorial: Create a VPC for use with a DB instance (IPv4 only)
- Tutorial: Create a VPC for use with a DB instance (dual-stack mode)

Working with a DB instance in a VPC

Here are some tips on working with a DB instance in a VPC:

- Your VPC must have at least two subnets. These subnets must be in two different Availability Zones in the AWS Region where you want to deploy your DB instance. A *subnet* is a segment of a VPC's IP address range that you can specify and that you can use to group DB instances based on your security and operational needs.

For Multi-AZ deployments, defining a subnet for two or more Availability Zones in an AWS Region allows Amazon RDS to create a new standby in another Availability Zone as needed. Make sure to do this even for Single-AZ deployments, just in case you want to convert them to Multi-AZ deployments at some point.

Note

The DB subnet group for a Local Zone can have only one subnet.

- If you want your DB instance in the VPC to be publicly accessible, make sure to turn on the VPC attributes *DNS hostnames* and *DNS resolution*.
- Your VPC must have a DB subnet group that you create. You create a DB subnet group by specifying the subnets you created. Amazon RDS chooses a subnet and an IP address within that subnet group to associate with your DB instance. The DB instance uses the Availability Zone that contains the subnet.
- Your VPC must have a VPC security group that allows access to the DB instance.

  For more information, see Scenarios for accessing a DB instance in a VPC.
- The CIDR blocks in each of your subnets must be large enough to accommodate spare IP addresses for Amazon RDS to use during maintenance activities, including failover and compute scaling. For example, a range such as 10.0.0.0/24 and 10.0.1.0/24 is typically large enough.
- A VPC can have an *instance tenancy* attribute of either *default* or *dedicated*. All default VPCs have the instance tenancy attribute set to default, and a default VPC can support any DB instance class.

  If you choose to have your DB instance in a dedicated VPC where the instance tenancy attribute is set to dedicated, the DB instance class of your DB instance must be one of the approved Amazon EC2 dedicated instance types. For example, the r5.large EC2 dedicated instance corresponds to the db.r5.large DB instance class. For information about instance tenancy in a VPC, see Dedicated instances in the *Amazon Elastic Compute Cloud User Guide*.

  For more information about the instance types that can be in a dedicated instance, see Amazon EC2 dedicated instances on the Amazon EC2 pricing page.

Note

When you set the instance tenancy attribute to dedicated for a DB instance, it doesn't guarantee that the DB instance will run on a dedicated host.

- When an option group is assigned to a DB instance, it's associated with the DB instance's VPC. This linkage means that you can't use the option group assigned to a DB instance if you attempt to restore the DB instance into a different VPC.
- If you restore a DB instance into a different VPC, make sure to either assign the default option group to the DB instance, assign an option group that is linked to that VPC, or create a new option group and assign it to the DB instance. With persistent or permanent options, such as Oracle TDE, you must create a new

option group that includes the persistent or permanent option when restoring a DB instance into a different VPC.

VPC encryption control

VPC encryption controls allow you to enforce encryption-in-transit for all network traffic within your VPCs. Use encryption control to meet regulatory compliance requirements by ensuring that only encryption-capable Nitro-based hardware can be provisioned in designated VPCs. Encryption control also catches compatibility issues at API request time rather than during provisioning. Your existing workloads continue operating and only new incompatible requests are blocked.

Set your VPC encryption controls by setting the VPC control mode to :

- *disabled* (default)
- *monitor*
- *enforced*

To check the current control mode for your VPC, use the AWS Management Console or DescribeVpcs CLI or API command.

If your VPC enforces encryption, you can only provision Nitro-based DB instances that support encryption in transit in that VPC. For more information see, DB instance class types. For information about Nitro instances, see Instances built on the AWS Nitro System in the *Amazon EC2 User Guide*.

Note

If you try to provision incompatible DB instances in an encryption-enforced VPC, Amazon RDS returns a `VpcEncryptionControlViolationException` exception.

Working with DB subnet groups

Subnets are segments of a VPC's IP address range that you designate to group your resources based on security and operational needs. A *DB subnet group* is a collection of subnets (typically private) that you create in a VPC and that you then designate for your DB instances. By using a DB subnet group, you can specify a particular VPC when creating DB instances using the AWS CLI or RDS API. If you use the console, you can choose the VPC and subnet groups you want to use.

Each DB subnet group should have subnets in at least two Availability Zones in a given AWS Region. When creating a DB instance in a VPC, you choose a DB subnet group for it. From the DB subnet group, Amazon RDS chooses a subnet and an IP address within that subnet to associate with the DB instance. The DB uses the Availability Zone that contains the subnet. Amazon RDS always assigns an IP address from a subnet that has free IP address space.

If the primary DB instance of a Multi-AZ deployment fails, Amazon RDS can promote the corresponding standby and later create a new standby using an IP address of the subnet in one of the other Availability Zones.

The subnets in a DB subnet group are either public or private. The subnets are public or private, depending on the configuration that you set for their network access control lists (network ACLs) and routing tables. For a DB instance to be publicly accessible, all of the subnets in its DB subnet group must be public. If a subnet that's associated with a publicly accessible DB instance changes from public to private, it can affect DB instance availability.

To create a DB subnet group that supports dual-stack mode, make sure that each subnet that you add to the DB subnet group has an Internet Protocol version 6 (IPv6) CIDR block associated with it. For more information, see Amazon RDS IP addressing and Migrating to IPv6 in the *Amazon VPC User Guide.*

Note

The DB subnet group for a Local Zone can have only one subnet.

When Amazon RDS creates a DB instance in a VPC, it assigns a network interface to your DB instance by using an IP address from your DB subnet group. However, we strongly recommend that you use the Domain Name System (DNS) name to connect to your DB instance. We recommend this because the underlying IP address changes during failover.

Note

For each DB instance that you run in a VPC, make sure to reserve at least one address in each subnet in the DB subnet group for use by Amazon RDS for recovery actions.

You can create a DB instance in a shared VPC.

Some considerations to keep in mind while using shared VPCs:

- You can move a DB instance from a shared VPC subnet to a non-shared VPC subnet and vice-versa.
- Participants in a shared VPC must create a security group in the VPC to allow them to create a DB instance.
- Owners and participants in a shared VPC can access the database by using SQL queries. However, only the creator of a resource can make any API calls on the resource.

Amazon RDS IP addressing

IP addresses enable resources in your VPC to communicate with each other, and with resources over the internet. Amazon RDS supports both IPv4 and IPv6 addressing protocols. By default, Amazon RDS and Amazon VPC use the IPv4 addressing protocol. You can't turn off this behavior. When you create a VPC, make sure to specify an IPv4 CIDR block (a range of private IPv4 addresses). You can optionally assign an IPv6 CIDR block to your VPC and subnets, and assign IPv6 addresses from that block to DB instances in your subnet.

Support for the IPv6 protocol expands the number of supported IP addresses. By using the IPv6 protocol, you ensure that you have sufficient available addresses for the future growth of the internet. New and existing RDS resources can use IPv4 and IPv6 addresses within your VPC. Configuring, securing, and translating network traffic between the two protocols used in different parts of an application can cause operational overhead. You can standardize on the IPv6 protocol for Amazon RDS resources to simplify your network configuration.

IPv4 addresses

When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a CIDR block, such as `10.0.0.0/16`. A *DB subnet group* defines the range of IP addresses in this CIDR block that a DB instance can use. These IP addresses can be private or public.

A private IPv4 address is an IP address that's not reachable over the internet. You can use private IPv4 addresses for communication between your DB instance and other resources, such as Amazon EC2 instances, in the same VPC. Each DB instance has a private IP address for communication in the VPC.

A public IP address is an IPv4 address that's reachable from the internet. You can use public addresses for communication between your DB instance and resources on the internet, such as a SQL client. You control whether your DB instance receives a public IP address.

Amazon RDS uses Public Elastic IPv4 addresses from EC2's public IPv4 address pool for publicly accessible database instances. These IP addresses are visible in your AWS account when using the `describe-addresses` CLI, API or viewing the Elastic IPs (EIP) section in the AWS Management Console. Each RDS-managed IP address is marked with a `service_managed` attribute set to `"rds"`.

While these IPs are visible in your account, they remain fully managed by Amazon RDS and cannot be modified or released. Amazon RDS releases IPs back into the public IPv4 address pool when no longer in use.

CloudTrail logs API calls related to RDS's EIP, such as the `AllocateAddress`. These API calls are invoked by the Service Principal `rds.amazonaws.com`.

Note

IPs allocated by Amazon RDS do not count against your account's EIP limits.

For a tutorial that shows you how to create a VPC with only private IPv4 addresses that you can use for a common Amazon RDS scenario, see Tutorial: Create a VPC for use with a DB instance (IPv4 only).

IPv6 addresses

You can optionally associate an IPv6 CIDR block with your VPC and subnets, and assign IPv6 addresses from that block to the resources in your VPC. Each IPv6 address is globally unique.

The IPv6 CIDR block for your VPC is automatically assigned from Amazon's pool of IPv6 addresses. You can't choose the range yourself.

When connecting to an IPv6 address, make sure that the following conditions are met:

- The client is configured so that client to database traffic over IPv6 is allowed.
- RDS security groups used by the DB instance are configured correctly so that client to database traffic over IPv6 is allowed.
- The client operating system stack allows traffic on the IPv6 address, and operating system drivers and libraries are configured to choose the correct default DB instance endpoint (either IPv4 or IPv6).

For more information about IPv6, see IP Addressing in the *Amazon VPC User Guide*.

Dual-stack mode

A DB instance runs in dual-stack mode when it can communicate over both IPv4 and IPv6 addressing protocols. Resources can then communicate with the DB instance using either IPv4, IPv6, or both protocols. Private dual-stack mode DB instances have IPv6 endpoints that RDS restricts to VPC access only, ensuring your IPv6 endpoints remain private. Public dual-stack mode DB instances provide both IPv4 and IPv6 endpoints that you can access from the internet.

Topics

- Dual-stack mode and DB subnet groups
- Working with dual-stack mode DB instances
- Modifying IPv4-only DB instances to use dual-stack mode
- Region and version availability
- Limitations for dual-stack network DB instances

For a tutorial that shows you how to create a VPC with both IPv4 and IPv6 addresses that you can use for a common Amazon RDS scenario, see Tutorial: Create a VPC for use with a DB instance (dual-stack mode).

Dual-stack mode and DB subnet groups

To use dual-stack mode, make sure that each subnet in the DB subnet group that you associate with the DB instance has an IPv6 CIDR block associated with it. You can create a new DB subnet group or modify an existing DB subnet group to meet this requirement. After a DB instance is in dual-stack mode, clients can connect to it normally. Make sure that client security firewalls and RDS DB instance security groups are accurately configured to allow traffic over IPv6. To connect, clients use the DB instance's endpoint. Client applications can specify which protocol is preferred when connecting to a database. In dual-stack mode, the DB instance detects the client's preferred network protocol, either IPv4 or IPv6, and uses that protocol for the connection.

If a DB subnet group stops supporting dual-stack mode because of subnet deletion or CIDR disassociation, there's a risk of an incompatible network state for DB instances that are associated with the DB subnet group. Also, you can't use the DB subnet group when you create a new dual-stack mode DB instance.

To determine whether a DB subnet group supports dual-stack mode by using the AWS Management Console, view the **Network type** on the details page of the DB subnet group. To determine whether a DB subnet group supports dual-stack mode by using the AWS CLI, run the describe-db-subnet-groups command and view `SupportedNetworkTypes` in the output.

Read replicas are treated as independent DB instances and can have a network type that's different from the primary DB instance. If you change the network type of a read replica's primary DB instance, the read replica isn't affected. When you are restoring a DB instance, you can restore it to any network type that's supported.

Working with dual-stack mode DB instances

When you create or modify a DB instance, you can specify dual-stack mode to allow your resources to communicate with your DB instance over IPv4, IPv6, or both.

When you use the AWS Management Console to create or modify a DB instance, you can specify dual-stack mode in the **Network type** section. The following image shows the **Network type** section in the console.

When you use the AWS CLI to create or modify a DB instance, set the `--network-type` option to `DUAL` to use dual-stack mode. When you use the RDS API to create or modify a DB instance, set the `NetworkType` parameter to `DUAL` to use dual-stack mode. When you are modifying the network type of a DB instance, downtime is possible. If dual-stack mode isn't supported by the specified DB engine version or DB subnet group, the `NetworkTypeNotSupported` error is returned.

For more information about creating a DB instance, see Creating an Amazon RDS DB instance. For more information about modifying a DB instance, see Modifying an Amazon RDS DB instance.

To determine whether a DB instance is in dual-stack mode by using the console, view the **Network type** on the **Connectivity & security** tab for the DB instance.

Modifying IPv4-only DB instances to use dual-stack mode

You can modify an IPv4-only DB instance to use dual-stack mode. To do so, change the network type of the DB instance. The modification might result in downtime.

It is recommended that you change the network type of your Amazon RDS DB instances during a maintenance window. Currently, setting the network type of new instances to dual-stack mode isn't supported. You can set network type manually by using the `modify-db-instance` command.

Before modifying a DB instance to use dual-stack mode, make sure that its DB subnet group supports dual-stack mode. If the DB subnet group associated with the DB instance doesn't support dual-stack mode, specify a different DB subnet group that supports it when you modify the DB instance. Modifying the DB subnet group of a DB instance can cause downtime.

If you modify the DB subnet group of a DB instance before you change the DB instance to use dual-stack mode, make sure that the DB subnet group is valid for the DB instance before and after the change.

For RDS for PostgreSQL, RDS for MySQL, RDS for Oracle, and RDS for MariaDB Single-AZ instances, we recommend that you run the modify-db-instance command with only the `--network-type` parameter set to `DUAL` to change the network to dual-stack mode. Adding other parameters along with the `--network-type` parameter in the same API call could result in downtime. To modify multiple parameters, ensure that the network type modification is successfully completed before sending another `modify-db-instance` request with other parameters.

Network type modifications for RDS for PostgreSQL, RDS for MySQL, RDS for Oracle, and RDS for MariaDB Multi-AZ DB instances cause a brief downtime and trigger a failover if you only use the `--network-type` parameter or if you combine parameters in a modify-db-instance command.

Network type modifications on RDS for SQL Server Single-AZ or Multi-AZ DB instances cause downtime if you only use the `--network-type` parameter or if you combine parameters in a `modify-db-instance` command. Network type modifications cause failover in an SQL Server Multi-AZ instance.

If you can't connect to the DB instance after the change, make sure that the client and database security firewalls and route tables are accurately configured to allow traffic to the database on the selected network (either IPv4 or IPv6). You might also need to modify operating system parameter, libraries, or drivers to connect using an IPv6 address.

When you modify a DB instance to use dual-stack mode, there can't be a pending change from a Single-AZ deployment to a Multi-AZ deployment, or from a Multi-AZ deployment to a Single-AZ deployment.

To modify an IPv4-only DB instance to use dual-stack mode

1. Modify a DB subnet group to support dual-stack mode, or create a DB subnet group that supports dual-stack mode:
    1. Associate an IPv6 CIDR block with your VPC.

       For instructions, see Add an IPv6 CIDR block to your VPC in the *Amazon VPC User Guide*.
    2. Attach the IPv6 CIDR block to all of the subnets in your the DB subnet group.

       For instructions, see Add an IPv6 CIDR block to your subnet in the *Amazon VPC User Guide*.
    3. Confirm that the DB subnet group supports dual-stack mode.

       If you are using the AWS Management Console, select the DB subnet group, and make sure that the **Supported network types** value is **Dual, IPv4**.

       If you are using the AWS CLI, run the describe-db-subnet-groups command, and make sure that the `SupportedNetworkType` value for the DB instance is `Dual, IPv4` .
2. Modify the security group associated with the DB instance to allow IPv6 connections to the database, or create a new security group that allows IPv6 connections.

For instructions, see Security group rules in the *Amazon VPC User Guide*.

3. Modify the DB instance to support dual-stack mode. To do so, set the **Network type** to **Dual-stack mode**.

   If you are using the console, make sure that the following settings are correct:

   - **Network type** – **Dual-stack mode**
   - **DB subnet group** – The DB subnet group that you configured in a previous step
   - **Security group** – The security that you configured in a previous step

   If you are using the AWS CLI, make sure that the following settings are correct:

   - `--network-type` – `dual`
   - `--db-subnet-group-name` – The DB subnet group that you configured in a previous step
   - `--vpc-security-group-ids` – The VPC security group that you configured in a previous step

   For example:

   ```
   aws rds modify-db-instance --db-instance-identifier my-instance --network-type "DUAL"
   ```

4. Confirm that the DB instance supports dual-stack mode.

   If you are using the console, choose the **Connectivity & security** tab for the DB instance. On that tab, make sure that the **Network type** value is **Dual-stack mode**.

   If you are using the AWS CLI, run the describe-db-instances command, and make sure that the `NetworkType` value for the DB instance is `dual`.

   Run the `dig` command on the DB instance endpoint to identify the IPv6 address associated with it.

   ```
   dig  db-instance-endpoint  AAAA
   ```

   Use the DB instance endpoint, not the IPv6 address, to connect to the DB instance.

Region and version availability

Feature availability and support varies across specific versions of each database engine, and across AWS Regions. For more information on version and Region availability with dual-stack mode, see Supported Regions and DB engines for dual-stack mode in Amazon RDS.

Limitations for dual-stack network DB instances

The following limitations apply to dual-stack network DB instances:

- DB instances can't use the IPv6 protocol exclusively. They can use IPv4 exclusively, or they can use the IPv4 and IPv6 protocol (dual-stack mode).
- Amazon RDS doesn't support native IPv6 subnets.
- For RDS for SQL Server, dual-stack mode DB instances that use Always On AGs availability group listener endpoints only present IPv4 addresses.
- You can't use RDS Proxy with dual-stack mode DB instances.
- You can't use dual-stack mode with RDS on AWS Outposts DB instances.
- You can't use dual-stack mode with DB instances in a Local Zone.

Hiding a DB instance in a VPC from the internet

One common Amazon RDS scenario is to have a VPC in which you have an Amazon EC2 instance with a public-facing web application and a DB instance with a database that isn't publicly accessible. For example, you can create a VPC that has a public subnet and a private subnet. EC2 instances that function as web servers can be deployed in the public subnet. The DB instances are deployed in the private subnet. In such a deployment, only the web servers have access to the DB instances. For an illustration of this scenario, see A DB instance in a VPC accessed by an Amazon EC2 instance in the same VPC.

When you launch a DB instance inside a VPC, the DB instance has a private IP address for traffic inside the VPC. This private IP address isn't publicly accessible. You can use the **Public access** option to designate whether the DB instance also has a public IP address in addition to the private IP address. If the DB instance is designated as publicly accessible, its DNS endpoint resolves to the private IP address from within the VPC. It resolves to the public IP address from outside of the VPC. Access to the DB instance is ultimately controlled by the security group it uses. That public access is not permitted if the security group assigned to the DB instance doesn't include inbound rules that permit it. In addition, for a DB instance to be publicly accessible, the subnets in its DB subnet group must have an internet gateway. For more information, see Can't connect to Amazon RDS DB instance

You can modify a DB instance to turn on or off public accessibility by modifying the **Public access** option. The following illustration shows the **Public access** option in the **Additional connectivity configuration** section. To set the option, open the **Additional connectivity configuration** section in the **Connectivity** section.

For information about modifying a DB instance to set the **Public access** option, see Modifying an Amazon RDS DB instance.

Creating a DB instance in a VPC

The following procedures help you create a DB instance in a VPC. To use the default VPC, you can begin with step 2, and use the VPC and DB subnet group have already been created for you. If you want to create an additional VPC, you can create a new VPC.

Note

If you want your DB instance in the VPC to be publicly accessible, you must update the DNS information for the VPC by enabling the VPC attributes *DNS hostnames* and *DNS resolution*. For information about updating the DNS information for a VPC instance, see Updating DNS support for your VPC.

Follow these steps to create a DB instance in a VPC:

- Step 1: Create a VPC
- Step 2: Create a DB subnet group
- Step 3: Create a VPC security group
- Step 4: Create a DB instance in the VPC

Step 1: Create a VPC

Create a VPC with subnets in at least two Availability Zones. You use these subnets when you create a DB subnet group. If you have a default VPC, a subnet is automatically created for you in each Availability Zone in the AWS Region.

For more information, see Create a VPC with private and public subnets, or see Create a VPC in the *Amazon VPC User Guide*.

Step 2: Create a DB subnet group

A DB subnet group is a collection of subnets (typically private) that you create for a VPC and that you then designate for your DB instances. A DB subnet group allows you to specify a particular VPC when you create DB instances using the AWS CLI or RDS API. If you use the console, you can just choose the VPC and subnets you want to use. Each DB subnet group must have at least one subnet in at least two Availability Zones in the AWS Region. As a best practice, each DB subnet group should have at least one subnet for every Availability Zone in the AWS Region.

For Multi-AZ deployments, defining a subnet for all Availability Zones in an AWS Region enables Amazon RDS to create a new standby replica in another Availability Zone if necessary. You can follow this best practice even for Single-AZ deployments, because you might convert them to Multi-AZ deployments in the future.

For a DB instance to be publicly accessible, the subnets in the DB subnet group must have an internet gateway. For more information about internet gateways for subnets, see Connect to the internet using an internet gateway in the *Amazon VPC User Guide*.

Note

The DB subnet group for a Local Zone can have only one subnet.

When you create a DB instance in a VPC, you can choose a DB subnet group. Amazon RDS chooses a subnet and an IP address within that subnet to associate with your DB instance. If no DB subnet groups exist, Amazon RDS creates a default subnet group when you create a DB instance. Amazon RDS creates and associates an Elastic

Network Interface to your DB instance with that IP address. The DB instance uses the Availability Zone that contains the subnet.

For Multi-AZ deployments, defining a subnet for two or more Availability Zones in an AWS Region allows Amazon RDS to create a new standby in another Availability Zone should the need arise. You need to do this even For Single-AZ deployments, just in case you want to convert them to Multi-AZ deployments at some point.

In this step, you create a DB subnet group and add the subnets that you created for your VPC.

To create a DB subnet group

1. Open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the navigation pane, choose **Subnet groups**.
3. Choose **Create DB Subnet Group**.
4. For **Name**, type the name of your DB subnet group.
5. For **Description**, type a description for your DB subnet group.
6. For **VPC**, choose the default VPC or the VPC that you created.
7. In the **Add subnets** section, choose the Availability Zones that include the subnets from **Availability Zones**, and then choose the subnets from **Subnets**.

   Note

   If you have enabled a Local Zone, you can choose an Availability Zone group on the **Create DB subnet group** page. In this case, choose the **Availability Zone group**, **Availability Zones**, and **Subnets**.
8. Choose **Create**.

   Your new DB subnet group appears in the DB subnet groups list on the RDS console. You can choose the DB subnet group to see details, including all of the subnets associated with the group, in the details pane at the bottom of the window.

Step 3: Create a VPC security group

Before you create your DB instance, you can create a VPC security group to associate with your DB instance. If you don't create a VPC security group, you can use the default security group when you create a DB instance. For instructions on how to create a security group for your DB instance, see Create a VPC security group for a private DB instance, or see Control traffic to resources using security groups in the *Amazon VPC User Guide*.

Step 4: Create a DB instance in the VPC

In this step, you create a DB instance and use the VPC name, the DB subnet group, and the VPC security group you created in the previous steps.

Note

If you want your DB instance in the VPC to be publicly accessible, you must enable the VPC attributes *DNS hostnames* and *DNS resolution*. For more information, see DNS attributes for your VPC in the *Amazon VPC User Guide*.

For details on how to create a DB instance, see Creating an Amazon RDS DB instance.

When prompted in the **Connectivity** section, enter the VPC name, the DB subnet group, and the VPC security group.
