**My Research Work**

**AMI**

An Amazon Machine Image (**AMI**) is a special type of virtual appliance that is used to create a virtual machine within the Amazon Elastic Compute Cloud ("**EC2**"). It serves as the basic unit of deployment for services delivered using **EC2**.

Like all [virtual appliances](https://en.wikipedia.org/wiki/Virtual_appliance), the main component of an AMI is a read-only filesystem image that includes an operating system (e.g., [Linux](https://en.wikipedia.org/wiki/Linux), [Unix](https://en.wikipedia.org/wiki/Unix), or [Windows](https://en.wikipedia.org/wiki/Windows)) and any additional software required to deliver a service or a portion of it.[[2]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-2)

An AMI includes the following:

* A template for the root volume for the instance (for example, an operating system, an application server, and applications)
* Launch permissions that control which AWS accounts can use the AMI to launch instances
* A block device mapping that specifies the volumes to attach to the instance when it's launched

The AMI filesystem is compressed, encrypted, signed, split into a series of 10 MB chunks and uploaded into [Amazon S3](https://en.wikipedia.org/wiki/Amazon_S3) for storage. An XML manifest file stores information about the AMI, including name, version, architecture, default kernel id, decryption key and digests for all of the filesystem chunks.

An AMI does not include a kernel image, only a pointer to the default kernel id, which can be chosen from an approved list of safe kernels maintained by Amazon and its partners (e.g., [Red Hat](https://en.wikipedia.org/wiki/Red_Hat), [Canonical](https://en.wikipedia.org/wiki/Canonical_Ltd.), [Microsoft](https://en.wikipedia.org/wiki/Microsoft)). Users may choose kernels other than the default when booting an AMI.[[3]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-3)

## Operating systems

When it launched in August 2006, the EC2 service offered [Linux](https://en.wikipedia.org/wiki/Linux) and later [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems)' [OpenSolaris](https://en.wikipedia.org/wiki/OpenSolaris) and [Solaris Express Community Edition](https://en.wikipedia.org/wiki/Solaris_(operating_system)). In October 2008, EC2 added the [Windows Server 2003](https://en.wikipedia.org/wiki/Windows_Server_2003) and [Windows Server 2008](https://en.wikipedia.org/wiki/Windows_Server_2008) operating systems to the list of available [operating systems](https://en.wikipedia.org/wiki/Operating_system).[[4]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-4)[[5]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-5) As of December 2010, it has also been reported to run [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD);[[6]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-6) in March 2011, [NetBSD](https://en.wikipedia.org/wiki/NetBSD) AMIs became available.[[7]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-7) In November 2012, [Windows Server 2012](https://en.wikipedia.org/wiki/Windows_Server_2012) support was added.[[8]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-8)

### Amazon Linux AMI

Amazon has its own Linux distribution that is largely binary compatible with [Red Hat Enterprise Linux](https://en.wikipedia.org/wiki/Red_Hat_Enterprise_Linux), and therefore [CentOS](https://en.wikipedia.org/wiki/CentOS)[[9]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-9). This offering has been in production since September 2011, and in development since 2010.[[10]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-10)[[11]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-11)[[12]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-12)[[13]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-13)[[14]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-14) The most recent release is version 2018.03[[15]](https://en.wikipedia.org/wiki/Amazon_Machine_Image#cite_note-15) and uses version 4.14 of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel)

## Types of images

* **Public**: an AMI that can be used by anyone.
* **Paid**: a for-pay AMI that is registered with Amazon DevPay and can be used by anyone who subscribes for it. DevPay allows developers to mark-up Amazon's usage fees and optionally add monthly subscription fees.
* **Shared**: a private AMI that can only be used by Amazon EC2 users who are allowed access to it by the developer.

**KERNEL**

The **Linux kernel** is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source), [monolithic](https://en.wikipedia.org/wiki/Monolithic_kernel), [Unix-like](https://en.wikipedia.org/wiki/Unix-like) [operating system](https://en.wikipedia.org/wiki/Operating_system) [kernel](https://en.wikipedia.org/wiki/Kernel_(operating_system)). The [Linux](https://en.wikipedia.org/wiki/Linux) family of operating systems is based on this kernel and deployed on both traditional computer systems such as [personal computers](https://en.wikipedia.org/wiki/Personal_computer) and [servers](https://en.wikipedia.org/wiki/Server_(computing)), usually in the form of [Linux distributions](https://en.wikipedia.org/wiki/Linux_distribution),[[7]](https://en.wikipedia.org/wiki/Linux_kernel#cite_note-7) and on various [embedded devices](https://en.wikipedia.org/wiki/Embedded_device) such as [routers](https://en.wikipedia.org/wiki/Router_(computing)), [wireless access points](https://en.wikipedia.org/wiki/Wireless_access_point), [PBXes](https://en.wikipedia.org/wiki/Private_branch_exchange), [set-top boxes](https://en.wikipedia.org/wiki/Set-top_box), [FTA receivers](https://en.wikipedia.org/wiki/FTA_receiver), [smart TVs](https://en.wikipedia.org/wiki/Smart_TV), [PVRs](https://en.wikipedia.org/wiki/Personal_video_recorder), and [NAS appliances](https://en.wikipedia.org/wiki/NAS_appliance). While the adoption of the Linux kernel in [desktop computer](https://en.wikipedia.org/wiki/Desktop_computer) operating system is low, Linux-based operating systems dominate nearly every other segment of computing, from mobile devices to [mainframes](https://en.wikipedia.org/wiki/Mainframe_computer). As of November 2017, all of the [world's 500 most powerful supercomputers](https://en.wikipedia.org/wiki/TOP500) run Linux.[[8]](https://en.wikipedia.org/wiki/Linux_kernel#cite_note-top500stats-8) The [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) operating system for [tablet computers](https://en.wikipedia.org/wiki/Tablet_computer), [smartphones](https://en.wikipedia.org/wiki/Smartphone), and [smartwatches](https://en.wikipedia.org/wiki/Smartwatch) also uses the Linux kernel.

**Enhanced Snapshots**

## Product Overview

Enhanced Snapshots, from Sungard Availability Services | Labs, manages EBS snapshots and performs data deduplication to S3 for greater cost savings. The product will be useful for AWS users who want to: reduce the cost of storing snapshots, reduce the time IT engineers spend on routine snapshot management tasks, and schedule recurring snapshots. Deduplication is run across all enabled snapshots in a region, which decreases the amount of total data stored. Furthermore, deduplicated blocks are stored in S3 at a much lower cost than AWS' standard EBS snapshots. This tool provides users with a great way to pay less for long-term retention of snapshot based data in AWS. Using an intuitive interface, users can easily automate routine tasks like the creation of snapshots and the deletion of old backups. Since these tasks are automated, risks associated with human error are minimized. Enhanced Snapshots is open sourced and licensed under Apache v 2.0. Use of Enhanced Snapshots software is free and you only pay for the underlying infrastructure required to support it.

### Highlights

* Scheduling and Cost savings over AWS standard EBS snapshots due to compression & Deduplication.
* Automate routine tasks such as snapshot and deletion of resources.
* Zero load on protected instances and volumes during snapshot

# Security

Amazon Virtual Private Cloud provides features that you can use to increase and monitor the security for your virtual private cloud (VPC):

* **Security groups**: Security groups act as a firewall for associated Amazon EC2 instances, controlling both inbound and outbound traffic at the instance level. When you launch an instance, 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 is automatically associated the default security group for the VPC. For more information, see [Security Groups for Your VPC](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html).
* **Network access control lists (ACLs)**: Network ACLs act as a firewall for associated subnets, controlling both inbound and outbound traffic at the subnet level. For more information, see [Network ACLs](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html).
* **Flow logs**: Flow logs capture information about the IP traffic going to and from network interfaces in your VPC. You can create a flow log for a VPC, subnet, or individual network interface. Flow log data is published to CloudWatch Logs or Amazon S3, and can help you diagnose overly restrictive or overly permissive security group and network ACL rules. For more information, see [VPC Flow Logs](https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html).

You can use AWS Identity and Access Management to control who in your organization has permission to create and manage security groups, network ACLs, and flow logs. For example, you can give only your network administrators that permission, but not personnel who only need to launch instances. For more information, see [Controlling Access to Amazon VPC Resources](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_IAM.html).

Amazon security groups and network ACLs don't filter traffic to or from link-local addresses (169.254.0.0/16) or AWS-reserved IPv4 addresses—these are the first four IPv4 addresses of the subnet (including the Amazon DNS server address for the VPC). Similarly, flow logs do not capture IP traffic to or from these addresses. These addresses support the services: Domain Name Services (DNS), Dynamic Host Configuration Protocol (DHCP), Amazon EC2 instance metadata, Key Management Server (KMS—license management for Windows instances), and routing in the subnet. You can implement additional firewall solutions in your instances to block network communication with link-local addresses.

## Comparison of Security Groups and Network ACLs

The following table summarizes the basic differences between security groups and network ACLs.

|  |  |
| --- | --- |
| **Security group** | **Network ACL** |
| Operates at the instance level | Operates at the subnet level |
| Supports allow rules only | Supports allow rules and deny rules |
| Is stateful: Return traffic is automatically allowed, regardless of any rules | Is stateless: Return traffic must be explicitly allowed by rules |
| We evaluate all rules before deciding whether to allow traffic | We process rules in number order when deciding whether to allow traffic |
| Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on | Automatically applies to all instances in the subnets it's associated with (therefore, an additional layer of defense if the security group rules are too permissive) |

The following diagram illustrates the layers of security provided by security groups and network ACLs. For example, traffic from an Internet gateway is routed to the appropriate subnet using the routes in the routing table. The rules of the network ACL associated with the subnet control which traffic is allowed to the subnet. The rules of the security group associated with an instance control which traffic is allowed to the instance.


        Traffic is controlled using security groups and network ACLs
      

You can secure your instances using only security groups; however, you can add network ACLs as an additional layer of defense. For an example, see [Example: Controlling Access to Instances in a Subnet](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html#nacl-examples).

n Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans. EBS volumes persist independently from the running life of an EC2 instance.

After a volume is attached to an instance, you can use it like any other physical hard drive. EBS volumes are flexible. For current-generation volumes attached to current-generation instance types, you can dynamically increase size, modify the provisioned IOPS capacity, and change volume type on live production volumes.

Amazon EBS provides the following volume types: General Purpose SSD (gp2), Provisioned IOPS SSD (io1), Throughput Optimized HDD (st1), Cold HDD (sc1), and Magnetic (standard, a previous-generation type). They differ in performance characteristics and price, allowing you to tailor your storage performance and cost to the needs of your applications. For more information, see [Amazon EBS Volume Types](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html).

**Contents**

* [Benefits of Using EBS Volumes](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumes.html#EBSFeatures)
* [Amazon EBS Volume Types](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html)
* [Constraints on the Size and Configuration of an EBS Volume](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/volume_constraints.html)
* [Creating an Amazon EBS Volume](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-volume.html)
* [Restoring an Amazon EBS Volume from a Snapshot](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-restoring-volume.html)
* [Attaching an Amazon EBS Volume to an Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-attaching-volume.html)
* [Making an Amazon EBS Volume Available for Use on Linux](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html)
* [Viewing Information about an Amazon EBS Volume](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-describing-volumes.html)
* [Monitoring the Status of Your Volumes](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html)
* [Detaching an Amazon EBS Volume from an Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-detaching-volume.html)
* [Deleting an Amazon EBS Volume](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-deleting-volume.html)

## Benefits of Using EBS Volumes

EBS volumes provide several benefits that are not supported by instance store volumes.

* **Data availability**

When you create an EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to failure of any single hardware component. After you create a volume, you can attach it to any EC2 instance in the same Availability Zone. After you attach a volume, it appears as a native block device similar to a hard drive or other physical device. At that point, the instance can interact with the volume just as it would with a local drive. The instance can format the EBS volume with a file system, such as ext3, and then install applications.

An EBS volume can be attached to only one instance at a time, but multiple volumes can be attached to a single instance. If you attach multiple volumes to a device that you have named, you can stripe data across the volumes for increased I/O and throughput performance.

An EBS volume and the instance to which it attaches must be in the same Availability Zone.

You can get monitoring data for your EBS volumes, including root device volumes for EBS-backed instances, at no additional charge. For more information about monitoring metrics, see [Amazon CloudWatch Metrics for Amazon EBS](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using_cloudwatch_ebs.html). For information about tracking the status of your volumes, see [Amazon CloudWatch Events for Amazon EBS](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-cloud-watch-events.html).

* **Data persistence**

An EBS volume is off-instance storage that can persist independently from the life of an instance. You continue to pay for the volume usage as long as the data persists.

EBS volumes that are attached to a running instance can automatically detach from the instance with their data intact when the instance is terminated if you uncheck the **Delete on Termination** checkbox when you configure EBS volumes for your instance on the EC2 console. The volume can then be reattached to a new instance, enabling quick recovery. If the checkbox for **Delete on Termination** is checked, the volume(s) will delete upon termination of the EC2 instance. If you are using an EBS-backed instance, you can stop and restart that instance without affecting the data stored in the attached volume. The volume remains attached throughout the stop-start cycle. This enables you to process and store the data on your volume indefinitely, only using the processing and storage resources when required. The data persists on the volume until the volume is deleted explicitly. The physical block storage used by deleted EBS volumes is overwritten with zeroes before it is allocated to another account. If you are dealing with sensitive data, you should consider encrypting your data manually or storing the data on a volume protected by Amazon EBS encryption. For more information, see [Amazon EBS Encryption](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html).

By default, the root EBS volume that is created and attached to an instance at launch is deleted when that instance is terminated. You can modify this behavior by changing the value of the flag DeleteOnTermination to false when you launch the instance. This modified value causes the volume to persist even after the instance is terminated, and enables you to attach the volume to another instance.

By default, additional EBS volumes that are created and attached to an instance at launch are not deleted when that instance is terminated. You can modify this behavior by changing the value of the flag DeleteOnTermination to true when you launch the instance. This modified value causes the volumes to be deleted when the instance is terminated.

* **Data encryption**

For simplified data encryption, you can create encrypted EBS volumes with the Amazon EBS encryption feature. All EBS volume types support encryption. You can use encrypted EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256) and an Amazon-managed key infrastructure. The encryption occurs on the server that hosts the EC2 instance, providing encryption of data-in-transit from the EC2 instance to Amazon EBS storage. For more information, see [Amazon EBS Encryption](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html).

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted EBS volume in a region, a default master key is created for you automatically. This key is used for Amazon EBS encryption unless you select a customer master key (CMK) that you created separately using AWS KMS. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the [AWS Key Management Service Developer Guide](https://docs.aws.amazon.com/kms/latest/developerguide/).

* **Snapshots**

Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon S3, where it is stored redundantly in multiple Availability Zones. The volume does not need to be attached to a running instance in order to take a snapshot. As you continue to write data to a volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes. These snapshots can be used to create multiple new EBS volumes or move volumes across Availability Zones. Snapshots of encrypted EBS volumes are automatically encrypted.

When you create a new volume from a snapshot, it's an exact copy of the original volume at the time the snapshot was taken. EBS volumes that are restored from encrypted snapshots are automatically encrypted. By optionally specifying a different Availability Zone, you can use this functionality to create a duplicate volume in that zone. The snapshots can be shared with specific AWS accounts or made public. When you create snapshots, you incur charges in Amazon S3 based on the volume's total size. For a successive snapshot of the volume, you are only charged for any additional data beyond the volume's original size.

Snapshots are incremental backups, meaning that only the blocks on the volume that have changed after your most recent snapshot are saved. If you have a volume with 100 GiB of data, but only 5 GiB of data have changed since your last snapshot, only the 5 GiB of modified data is written to Amazon S3. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

To help categorize and manage your volumes and snapshots, you can tag them with metadata of your choice. For more information, see [Tagging Your Amazon EC2 Resources](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html).

* **Flexibility**

EBS volumes support live configuration changes while in production. You can modify volume type, volume size, and IOPS capacity without service interruptions.

# lastic IP Addresses

An Elastic IP address is a static IPv4 address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.

An Elastic IP address is a public IPv4 address, which is reachable from the internet. If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet; for example, to connect to your instance from your local computer.

We currently do not support Elastic IP addresses for IPv6.

**Contents**

* [Elastic IP Address Basics](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#eip-basics)
* [Working with Elastic IP Addresses](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#working-with-eips)
* [Using Reverse DNS for Email Applications](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#Using_Elastic_Addressing_Reverse_DNS)
* [Elastic IP Address Limit](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#using-instance-addressing-limit)

## Elastic IP Address Basics

The following are the basic characteristics of an Elastic IP address:

* To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance or a network interface.
* When you associate an Elastic IP address with an instance or its primary network interface, the instance's public IPv4 address (if it had one) is released back into Amazon's pool of public IPv4 addresses. You cannot reuse a public IPv4 address, and you cannot convert a public IPv4 address to an Elastic IP address. For more information, see [Public IPv4 Addresses and External DNS Hostnames](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-instance-addressing.html#concepts-public-addresses).
* You can disassociate an Elastic IP address from a resource, and reassociate it with a different resource. Any open connections to an instance continue to work for a time even after you disassociate its Elastic IP address and reassociate it with another instance. We recommend that you reopen these connections using the reassociated Elastic IP address.
* A disassociated Elastic IP address remains allocated to your account until you explicitly release it.
* To ensure efficient use of Elastic IP addresses, we impose a small hourly charge if an Elastic IP address is not associated with a running instance, or if it is associated with a stopped instance or an unattached network interface. While your instance is running, you are not charged for one Elastic IP address associated with the instance, but you are charged for any additional Elastic IP addresses associated with the instance. For more information, see [Amazon EC2 Pricing](https://aws.amazon.com/ec2/pricing/on-demand/#Elastic_IP_Addresses).
* An Elastic IP address is for use in a specific region only.
* When you associate an Elastic IP address with an instance that previously had a public IPv4 address, the public DNS hostname of the instance changes to match the Elastic IP address.
* We resolve a public DNS hostname to the public IPv4 address or the Elastic IP address of the instance outside the network of the instance, and to the private IPv4 address of the instance from within the network of the instance.
* When you allocate an Elastic IP address from an IP address pool that you have brought to your AWS account, it does not count toward your Elastic IP address limits.

## Working with Elastic IP Addresses

The following sections describe how you can work with Elastic IP addresses.

**Tasks**

* [Allocating an Elastic IP Address](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#using-instance-addressing-eips-allocating)
* [Describing Your Elastic IP Addresses](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#using-instance-addressing-eips-describing)
* [Tagging an Elastic IP Address](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#using-instance-addressing-eips-tagging)
* [Associating an Elastic IP Address with a Running Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#using-instance-addressing-eips-associating)
* [Disassociating an Elastic IP Address and Reassociating with a Different Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#using-instance-addressing-eips-associating-different)
* [Releasing an Elastic IP Address](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#using-instance-addressing-eips-releasing)
* [Recovering an Elastic IP Address](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#using-eip-recovering)

### Allocating an Elastic IP Address

You can allocate an Elastic IP address from Amazon's pool of public IPv4 addresses, or from a custom IP address pool that you have brought to your AWS account. For more information about bringing your own IP address range to your AWS account, see [Bring Your Own IP Addresses (BYOIP)](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-byoip.html).

You can allocate an Elastic IP address using the Amazon EC2 console or the command line.

**To allocate an Elastic IP address from Amazon's pool of public IPv4 addresses using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate new address**.
4. For **IPv4 address pool**, choose **Amazon pool**.
5. Choose **Allocate**, and close the confirmation screen.

**To allocate an Elastic IP address from an IP address pool that you own using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate new address**.
4. For **IPv4 address pool**, choose **Owned by me** and then select the IP address pool.

To see the IP address range of the selected address pool and the number of IP addresses already allocated from the address pool, see **Address ranges**.

1. For **IPv4 address**, do one of the following:
   * To let Amazon EC2 select an IP address from the address pool, choose **No preference**.
   * To select a specific IP address from the address pool, choose **Select an address** and then type the IP address.
2. Choose **Allocate**, and close the confirmation screen.

**To allocate an Elastic IP address using the command line**

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [allocate-address](https://docs.aws.amazon.com/cli/latest/reference/ec2/allocate-address.html) (AWS CLI)
* [New-EC2Address](https://docs.aws.amazon.com/powershell/latest/reference/items/New-EC2Address.html) (AWS Tools for Windows PowerShell)

### Describing Your Elastic IP Addresses

You can describe an Elastic IP address using the Amazon EC2 or the command line.

**To describe your Elastic IP addresses using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select a filter from the Resource Attribute list to begin searching. You can use multiple filters in a single search.

**To describe your Elastic IP addresses using the command line**

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [describe-addresses](https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-addresses.html) (AWS CLI)
* [Get-EC2Address](https://docs.aws.amazon.com/powershell/latest/reference/items/Get-EC2Address.html) (AWS Tools for Windows PowerShell)

### Tagging an Elastic IP Address

You can assign custom tags to your Elastic IP addresses to categorize them in different ways, for example, by purpose, owner, or environment. This helps you to quickly find a specific Elastic IP address based on the custom tags you've assigned it.

Note

Cost allocation tracking using Elastic IP address tags is not supported.

**To tag an Elastic IP address using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to tag and choose **Tags**.
4. Choose **Add/Edit Tags**.
5. In the **Add/Edit Tags** dialog box, choose **Create Tag**, and then specify the key and value for the tag.
6. (Optional) Choose **Create Tag** to add additional tags to the Elastic IP address.
7. Choose **Save**.

**To tag an Elastic IP address using the command line**

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [create-tags](https://docs.aws.amazon.com/cli/latest/reference/ec2/create-tags.html) (AWS CLI)

aws ec2 create-tags --resources *eipalloc-12345678* --tags Key=*Owner*,Value=*TeamA*

* [New-EC2Tag](https://docs.aws.amazon.com/powershell/latest/reference/items/New-EC2Tag.html) (AWS Tools for Windows PowerShell)

The New-EC2Tag command needs a Tag parameter, which specifies the key and value pair to be used for the Elastic IP address tag. The following commands create the Tag parameter:

PS C:\> $tag = New-Object Amazon.EC2.Model.Tag

PS C:\> $tag.Key = "*Owner*"

PS C:\> $tag.Value = "*TeamA*"

PS C:\> New-EC2Tag -Resource *eipalloc-12345678* -Tag $tag

### Associating an Elastic IP Address with a Running Instance

You can associate an Elastic IP address to an instance using the Amazon EC2 console or the command line.

If you're associating an Elastic IP address with your instance to enable communication with the internet, you must also ensure that your instance is in a public subnet. For more information, see [Internet Gateways](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html) in the Amazon VPC User Guide.

**To associate an Elastic IP address with an instance using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select an Elastic IP address and choose **Actions**, **Associate address**.
4. Select the instance from **Instance** and then choose **Associate**.

**To associate an Elastic IP address using the command line**

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [associate-address](https://docs.aws.amazon.com/cli/latest/reference/ec2/associate-address.html) (AWS CLI)
* [Register-EC2Address](https://docs.aws.amazon.com/powershell/latest/reference/items/Register-EC2Address.html) (AWS Tools for Windows PowerShell)

### Disassociating an Elastic IP Address and Reassociating with a Different Instance

You can disassociate an Elastic IP address and then reassociate it using the Amazon EC2 console or the command line.

**To disassociate and reassociate an Elastic IP address using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, choose **Actions**, and then select **Disassociate address**.
4. Choose **Disassociate address**.
5. Select the address that you disassociated in the previous step. For **Actions**, choose **Associate address**.
6. Select the new instance from **Instance**, and then choose **Associate**.

**To disassociate an Elastic IP address using the command line**

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [disassociate-address](https://docs.aws.amazon.com/cli/latest/reference/ec2/disassociate-address.html) (AWS CLI)
* [Unregister-EC2Address](https://docs.aws.amazon.com/powershell/latest/reference/items/Unregister-EC2Address.html) (AWS Tools for Windows PowerShell)

**To associate an Elastic IP address using the command line**

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [associate-address](https://docs.aws.amazon.com/cli/latest/reference/ec2/associate-address.html) (AWS CLI)
* [Register-EC2Address](https://docs.aws.amazon.com/powershell/latest/reference/items/Register-EC2Address.html) (AWS Tools for Windows PowerShell)

### Releasing an Elastic IP Address

If you no longer need an Elastic IP address, we recommend that you release it (the address must not be associated with an instance).

**To release an Elastic IP address using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, choose **Actions**, and then select **Release addresses**. Choose **Release** when prompted.

**To release an Elastic IP address using the command line**

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [release-address](https://docs.aws.amazon.com/cli/latest/reference/ec2/release-address.html) (AWS CLI)
* [Remove-EC2Address](https://docs.aws.amazon.com/powershell/latest/reference/items/Remove-EC2Address.html) (AWS Tools for Windows PowerShell)

### Recovering an Elastic IP Address

If you have released your Elastic IP address, you might be able to recover it. The following rules apply:

* You cannot recover an Elastic IP address if it has been allocated to another AWS account, or if it will result in your exceeding your Elastic IP address limit.
* You cannot recover tags associated with an Elastic IP address.
* You can recover an Elastic IP address using the Amazon EC2 API or a command line tool only.

**To recover an Elastic IP address using the command line**

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [allocate-address](https://docs.aws.amazon.com/cli/latest/reference/ec2/allocate-address.html) (AWS CLI) — Specify the IP address using the --address parameter as follows.

aws ec2 allocate-address --domain vpc --address *203.0.113.3*

* [New-EC2Address](https://docs.aws.amazon.com/powershell/latest/reference/items/New-EC2Address.html) (AWS Tools for Windows PowerShell) — Specify the IP address using the -Address parameter as follows.

PS C:\> New-EC2Address -Address *203.0.113.3* -Domain vpc -Region *us-east-1*

## Using Reverse DNS for Email Applications

If you intend to send email to third parties from an instance, we suggest you provision one or more Elastic IP addresses and provide them to us. AWS works with ISPs and internet anti-spam organizations to reduce the chance that your email sent from these addresses will be flagged as spam.

In addition, assigning a static reverse DNS record to your Elastic IP address used to send email can help avoid having email flagged as spam by some anti-spam organizations. Note that a corresponding forward DNS record (record type A) pointing to your Elastic IP address must exist before we can create your reverse DNS record.

If a reverse DNS record is associated with an Elastic IP address, the Elastic IP address is locked to your account and cannot be released from your account until the record is removed.

To remove email sending limits, or to provide us with your Elastic IP addresses and reverse DNS records, go to the [Request to Remove Email Sending Limitations](https://aws.amazon.com/forms/ec2-email-limit-rdns-request) page.

## Elastic IP Address Limit

By default, all AWS accounts are limited to five (5) Elastic IP addresses per region, because public (IPv4) internet addresses are a scarce public resource. We strongly encourage you to use an Elastic IP address primarily for the ability to remap the address to another instance in the case of instance failure, and to use DNS hostnames for all other inter-node communication.

If you feel your architecture warrants additional Elastic IP addresses, complete the [Amazon EC2 Elastic IP Address Request Form](https://console.aws.amazon.com/support/home#/case/create?issueType=service-limit-increase&limitType=service-code-elastic-ips). Describe your use case so that we can understand your need for additional addresses.

**Identity and Access Management (IAM)**

# Controlling Access to Amazon VPC Resources

To allow access to Amazon VPC resources without sharing your security credentials, you must create and attach an IAM policy to the IAM user or the group to which the IAM user belongs. The IAM user must be given permission to use the specific Amazon VPC resources and Amazon EC2 API actions they need. When you attach a policy to a user or group of users, it allows or denies permission to perform the specified tasks on the specified resources. Some API actions support resource-level permissions, which allow you to control the specific resources that users can create or modify.

Important

Currently, not all Amazon EC2 API actions support resource-level permissions. If an Amazon EC2 API action does not support resource-level permissions, you can grant users permission to use the action, but you have to specify a \* for the resource element of your policy statement. For an example of how to do this, see the following example policy: [1. Managing a VPC](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_IAM.html#managingvpciam) We'll add support for additional API actions and ARNs for additional Amazon EC2 resources later. For information about which ARNs you can use with which Amazon EC2 API actions, as well as supported condition keys for each ARN, see [Supported Resources and Conditions for Amazon EC2 API Actions](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-supported-iam-actions-resources.html) in the Amazon EC2 User Guide for Linux Instances.

For more information about creating IAM policies for Amazon EC2, supported resources for EC2 API actions, as well as example policies for Amazon EC2, see [IAM Policies for Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-policies-for-amazon-ec2.html) in the Amazon EC2 User Guide for Linux Instances.

**Simple Notification Service (SNS)**

Amazon Simple Notification Service (**SNS**) is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications. ... Additionally, **SNS** can be used to fan out notifications to end users using mobile push, SMS, and email.

You can get started with Amazon SNS in minutes by using the AWS Management Console, AWS Command Line Interface (CLI), or AWS Software Development Kit (SDK).

## Benefits

### Reliably deliver messages with durability

Amazon SNS uses cross availability zone message storage to provide high message durability. Running within Amazon’s proven network infrastructure and datacenters, Amazon SNS topics are available whenever your applications need them. All messages published to Amazon SNS are stored redundantly across multiple geographically separated servers and data centers. Amazon SNS reliably delivers messages to all valid AWS endpoints, such as [Amazon SQS](https://aws.amazon.com/sqs/) queues and [AWS Lambda](https://aws.amazon.com/lambda/) functions.

### Automatically scale your workload

Amazon SNS leverages the proven AWS cloud to dynamically scale with your application. Amazon SNS is a fully managed service, taking care of the heavy lifting related to capacity planning, provisioning, monitoring, and patching. The service is designed to handle high-throughput, bursty traffic patterns. Moreover, there is no upfront cost, and no need to acquire, install, configure, or upgrade messaging software.

### Simplify your architecture with Message Filtering

Amazon SNS helps you simplify your pub/sub messaging architecture by offloading the message filtering logic from your subscriber systems, and message routing logic from your publisher systems. With Amazon SNS message filtering, subscribing endpoints receive only the messages of interest, instead of all messages published to the topic. [Amazon CloudWatch](https://aws.amazon.com/cloudwatch/) gives visibility into your filtering activity, and [AWS CloudFormation](https://aws.amazon.com/cloudformation/) enables you to deploy subscription filter policies in an automated and secure manner.

### Keep messages private and secure

Amazon SNS topic owners can keep sensitive data secure by setting topic policies that restrict who can publish and subscribe to a topic. Amazon SNS also ensures that data is encrypted in transit by applying [Amazon ATS](https://www.amazontrust.com/) certificates to support its HTTPS API, and can also encrypt data at rest by using [AWS KMS](https://aws.amazon.com/kms/) keys. Additionally, using [AWS PrivateLink](https://aws.amazon.com/privatelink/), you can privately publish messages to Amazon SNS topics from your [Amazon VPC](https://aws.amazon.com/vpc/) subnets without traversing the public Internet. Amazon SNS can also support use cases in regulated markets, and is [in-scope with compliance programs](https://aws.amazon.com/compliance/services-in-scope/), including [HIPAA](https://aws.amazon.com/compliance/hipaa-compliance/), [PCI](https://aws.amazon.com/compliance/pci-dss-level-1-faqs/), [ISO](https://aws.amazon.com/compliance/iso-9001-faqs/), [FIPS](https://aws.amazon.com/compliance/fips/), [SOC](https://aws.amazon.com/compliance/soc-faqs/) and [FedRAMP](https://aws.amazon.com/compliance/fedramp/)[.](https://aws.amazon.com/compliance/hipaa-compliance/)

## How it works

Amazon SNS enables message filtering and fanout to a large number of subscribers, including serverless functions, queues, and distributed systems. Additionally, Amazon SNS fans out notifications to end users via mobile push messages, SMS, and email.

*For figure Ref.* [*https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc*](https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc)

# Changing the AWS Account Root User Password (Password policy)

You must be signed in as the AWS account root user in order to change the password. To learn how to reset a forgotten root user password, see [Resetting Your Lost or Forgotten Passwords or Access Keys](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys_retrieve.html).

**To change the password for the root user**

1. Use your AWS account email address and password to sign in to the [AWS Management Console](https://console.aws.amazon.com/) as the root user.

Note

If you previously signed in to the console with [*IAM user*](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html) credentials, your browser might remember this preference and open your account-specific sign-in page. You cannot use the IAM user sign-in page to sign in with your AWS account root user credentials. If you see the IAM user sign-in page, choose **Sign-in using root account credentials** near the bottom of the page to return to the main sign-in page. From there, you can type your AWS account email address and password.

1. In the upper right corner of the console, choose your account name or number and then choose **My Account**.
2. On the right side of the page, next to the **Account Settings** section, choose **Edit**.
3. On the **Password** line choose **Edit** to change your password.
4. Choose a strong password. Although you can [set an account password policy for IAM users](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html), that policy does not apply to your AWS account root user.

AWS requires that your password meet these conditions:

* + have a minimum of 8 characters and a maximum of 128 characters
  + include a minimum of three of the following mix of character types: uppercase, lowercase, numbers, and ! @ # $ % ^ & \* () <> [] {} | \_+-= symbols
  + not be identical to your AWS account name or email address

Note

AWS is rolling out improvements to the sign-in process. One of those improvements is to enforce a more secure password policy for your account. If your account has been upgraded, you are required to meet the password policy above. If your account has not yet been upgraded, then AWS does not enforce this policy, but highly recommends that you follow its guidelines for a more secure password.

To protect your password, it's important to follow these best practices:

* + Change your password periodically and keep your password private, since anyone who knows your password may access your account.
  + Use a different password on AWS than you use on other sites.
  + Avoid passwords that are easy to guess. These include passwords such as secret, password, amazon, or 123456. They also include things like a dictionary word, your name, email address, or other personal information that can easily be obtained.