



 

STOP AND START :

The instance is preparing to be stopped or stop-hibernated.The instance is shut down and cannot be used. The instance can be started at any time. If your instance fails a status check or is not running your applications as expected, and if the root volume of your instance is an Amazon EBS volume, you can stop and start your instance to try to fix the problem.

When you stop your instance, it enters the stopping state, and then the stopped state. We don't charge usage or data transfer fees for your instance after you stop it, but we do charge for the storage for any Amazon EBS volumes. While your instance is in the stopped state, you can modify certain attributes of the instance, including the instance type.

When you start your instance, it enters the pending state, and we move the instance to a new host computer (though in some cases, it remains on the current host). When you stop and start your instance, you lose any data on the instance store volumes on the previous host computer.

Your instance retains its private IPv4 address, which means that an Elastic IP address associated with the private IPv4 address or network interface is still associated with your instance. If your instance has an IPv6 address, it retains its IPv6 address.

Each time you transition an instance from stopped to running, we charge per second when the instance is running, with a minimum of one minute every time you start your instance.

For more information, see [Stop and start your instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Stop_Start.html).

When you hibernate an instance, we signal the operating system to perform hibernation (suspend-to-disk), which saves the contents from the instance memory (RAM) to your Amazon EBS root volume. We persist the instance's Amazon EBS root volume and any attached Amazon EBS data volumes. When you start your instance, the Amazon EBS root volume is restored to its previous state and the RAM contents are reloaded. Previously attached data volumes are reattached and the instance retains its instance ID.

When you hibernate your instance, it enters the stopping state, and then the stopped state. We don't charge usage for a hibernated instance when it is in the stopped state, but we do charge while it is in the stopping state, unlike when you [stop an instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html#instance-stop-start) without hibernating it. We don't charge usage for data transfer fees, but we do charge for the storage for any Amazon EBS volumes, including storage for the RAM data.

When you start your hibernated instance, it enters the pending state, and we move the instance to a new host computer (though in some cases, it remains on the current host).

Your instance retains its private IPv4 address, which means that an Elastic IP address associated with the private IPv4 address or network interface is still associated with your instance. If your instance has an IPv6 address, it retains its IPv6 address.

REBOOT : You can reboot your instance using the Amazon EC2 console, a command line tool, and the Amazon EC2 API. We recommend that you use Amazon EC2 to reboot your instance instead of running the operating system reboot command from your instance.

Rebooting an instance is equivalent to rebooting an operating system. The instance remains on the same host computer and maintains its public DNS name, private IP address, and any data on its instance store volumes. It typically takes a few minutes for the reboot to complete, but the time it takes to reboot depends on the instance configuration.

Rebooting an instance doesn't start a new instance billing period; per second billing continues without a further one-minute minimum charge.

TEMINATE PUBLIC IP : The instance has been permanently deleted and cannot be started. When you've decided that you no longer need an instance, you can terminate it. As soon as the status of an instance changes to shutting-down or terminated, you stop incurring charges for that instance.

If you enable termination protection, you can't terminate the instance using the console, CLI, or API.

After you terminate an instance, it remains visible in the console for a short while, and then the entry is automatically deleted. You can also describe a terminated instance using the CLI and API. Resources (such as tags) are gradually disassociated from the terminated instance, therefore may no longer be visible on the terminated instance after a short while. You can't connect to or recover a terminated instance.

Each Amazon EBS-backed instance supports the InstanceInitiatedShutdownBehavior attribute, which controls whether the instance stops or terminates when you initiate shutdown from within the instance itself (for example, by using the **shutdown** command on Linux). The default behavior is to stop the instance. You can modify the setting of this attribute while the instance is running or stopped.

Each Amazon EBS volume supports the DeleteOnTermination attribute, which controls whether the volume is deleted or preserved when you terminate the instance it is attached to. The default is to delete the root device volume and preserve any other EBS volumes.

PUBLIC IP : Amazon Elastic Inference (EI) is a resource you can attach to your Amazon EC2 CPU instances to accelerate your deep learning (DL) inference workloads. Amazon EI accelerators come in multiple sizes and are a cost-effective method to build intelligent capabilities into applications running on Amazon EC2 instances.

Amazon EI distributes model operations defined by TensorFlow, Apache MXNet, and the Open Neural Network Exchange (ONNX) format through MXNet between low-cost, DL inference accelerators and the CPU of the instance.

APPLICATION INSTALLED : Bitscape is a global technology consulting company and a software Development Company which improves software delivery through its ALM services backed-up by CMMi practices. From feasibility study to business need analysis to recommendations of development of solution (including but not limited to Agile, scrum, CMMi or SDLC Waterfall method), we consult all and suggest best practices approach for the solution for organization to achieve organizational application life cycle management goal. We focus on critical process improvement visibility and manageability, leading directly to better quality, reliability and ultimately predictability over the entire software delivery process.  
  
Some of the basic functions we perform in ALM services are:  
  
•Requirements analysis.  
•Requirements management.  
•Feature management.  
•Workflow.  
•Modelling.  
•Design.  
•Project management.  
•Software deployment.  
•Software testing.  
•Release management.  
•Change management.  
•Software information management (for ALM tool integration).  
•Build management.  
•Software configuration management.  
•Revision control.  
•Issue management.  
•Monitoring and reporting.