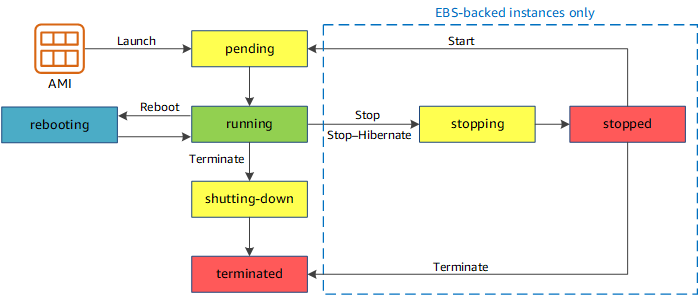
AWS EC2 Instance Lifecycle

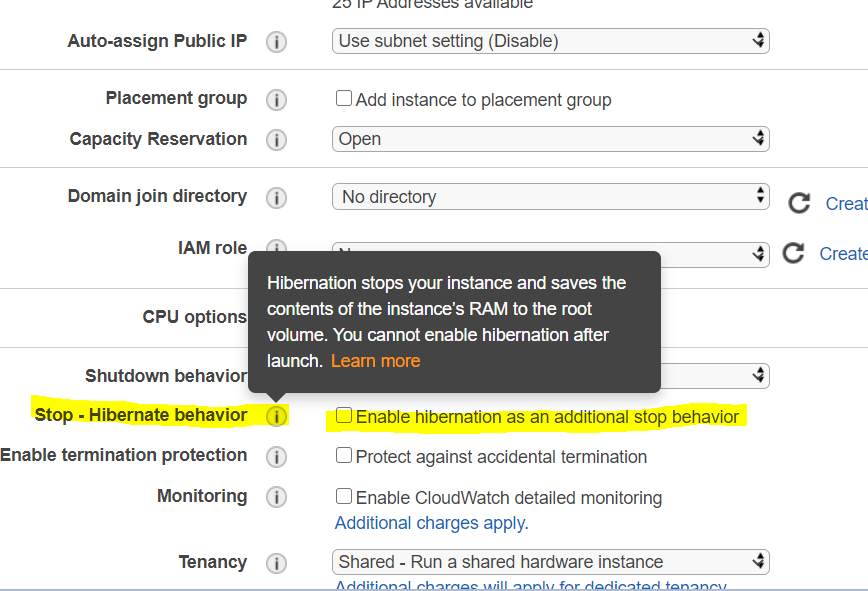
EC2 Instance Lifecycle Overview

* An EC2 instance is a virtual server in the AWS cloud, launched from an Amazon Machine Image (AMI).
* AMI provides the operating system, application server, and applications for the instance.
* When an instance is launched, it receives private DNS name that other instances within the same EC2 network can use to contact the instance. Optionally, it receives a public DNS name that can be used to contact the instance from the Internet.

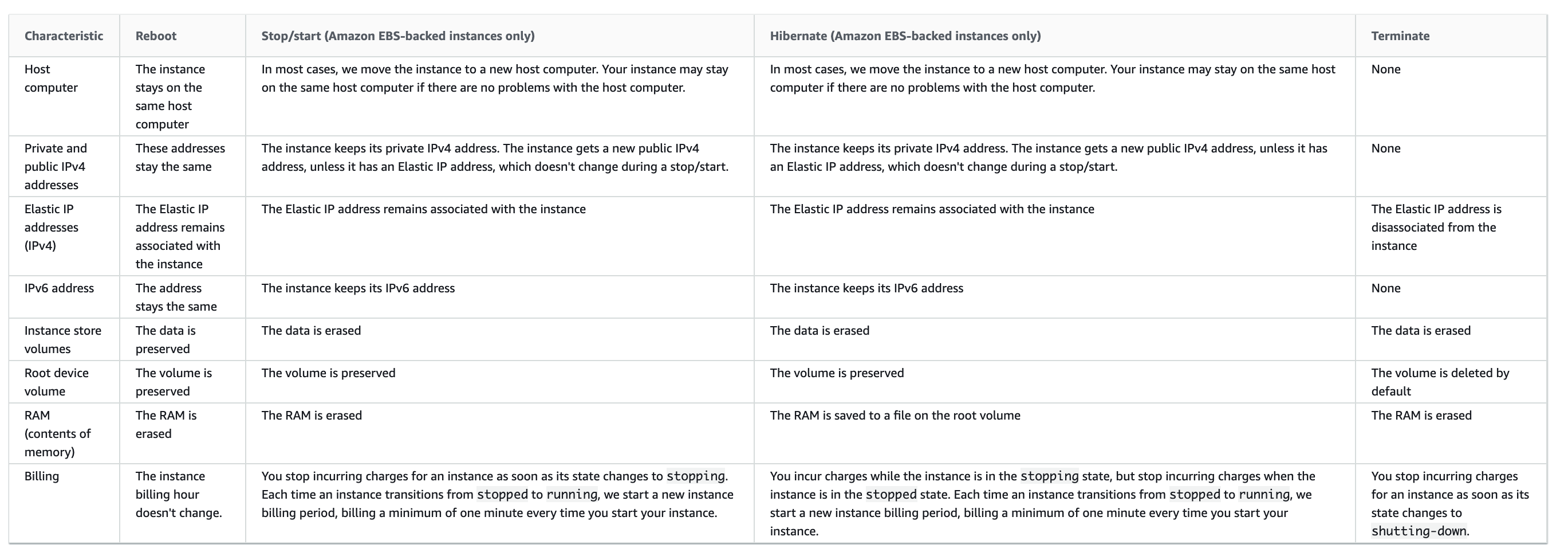
Instance Lifecycle



* Pending
  + When the instance is first launched is enters into the **pending** state
* Running
  + After the instance is launched, it enters into the **running** state
  + Charges are incurred for each second, with a one-minute minimum, that the instance running is running, even if the instance remains idle
* **Start & Stop (EBS-backed instances only)**
  + Only and EBS-backed instance can be stopped and started. Instance store-backed instance cannot be stopped and started
  + An instance can stopped & started in case the instance fails a status check or is not running as expected
  + Stop
    - After the instance is stopped, it enters in stopping state and then to stopped state.
    - Charges are only incurred for the EBS storage and not for the instance hourly charge or data transfer.
    - While the instance is stopped, its root volume can be treated like any other volume, and modify it *for e.g. repair file system problems or update software or change the instance type, user data, EBS optimization attributes etc*
    - Volume can be detached from the stopped instance, and attached to a running instance, modified, detached from the running instance, and then reattached to the stopped instance. It should be reattached using the storage device name that’s specified as the root device in the block device mapping for the instance.
  + Start
    - When the instance is started, it enters into pending state and then into running
    - An instance when stopped and started is launched on a new host
    - Any data on an instance store volume (not root volume) would be lost while data on the EBS volume persists
  + **EC2 instance retains its private IP address as well as the Elastic IP address.**
  + **If the instance has an IPv6 address, it retains its IPv6 address.**
  + **However, the public IP address, if assigned instead of the Elastic IP address, would be released**
  + For each transition of an instance from stopped to running, charges per second are incurred when the instance is running, with a minimum of one minute every time the instance is started
* **Instance Hibernate**
  + Instance hibernation signals the operating system to perform hibernation (suspend-to-disk), which saves the contents from the instance memory (RAM) to the EBS root volume
  + Instance’s EBS root volume and any attached EBS data volumes is persisted
  + When the instance is restarted, the 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.
  + After the instance is hibernated, it enters in stopping state and then to stopped state.
  + When the instance is restarted
    - EBS root volume is restored to its previous state
    - RAM contents are reloaded
    - Processes that were previously running on the instance are resumed
    - Previously attached data volumes are reattached and the instance retains its instance ID



* **Instance reboot**
  + Both EBS-backed and Instance store-backed instances can be rebooted
  + An instance retains it public DNS, public and private IP address during the reboot
  + Data on the EBS and Instance store volume is also retained
  + **Amazon recommends to use Amazon EC2 to reboot the instance instead of running the operating system reboot command from your instance as it performs a hard reboot if the instance does not cleanly shutdown within four minutes also creates an API record in CloudTrail, if enabled.**
* **Instance retirement**
  + An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware hosting the instance.
  + When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS.
  + If the instance root device is an Amazon EBS volume, the instance is stopped, and can be started again at any time.
  + If the instance root device is an instance store volume, the instance is terminated, and cannot be used again.
* **Instance Termination**
  + An instance can be terminated, and it enters into the shutting-down and then the terminated state
  + After an instance is terminated, it can’t be connected and no charges are incurred
  + **Instance Shutdown behaviour**
    - **EBS-backed instance** supports **InstanceInitiatedShutdownBehavior**attribute which determines whether the instance would be stopped or terminated when a shutdown command is initiated from the instance itself *for e.g. shutdown, halt or poweroff command in linux*
    - Default behaviour for the the instance to be stopped.
    - A shutdown command for an Instance store-backed instance will always terminate the instance
  + **Termination protection**
    - Termination protection (DisableApiTermination attribute) can be enabled on the instance to prevent it from being accidentally terminated
    - DisableApiTermination from the Console, CLI or API.
    - Instance can be terminated through EC2 CLI.
    - **Termination protection does not work for instances when**
      * **part of an Autoscaling group**
      * **launched as Spot instances**
      * **terminating an instance by initiating shutdown from the instance**
  + **Data persistence**
    - EBS volume have a DeleteOnTermination attribute which determines whether the volumes would be persisted or deleted when an instance they are associated with are terminated
    - Data on Instance store volume data does not persist
    - Data on EBS root volumes, have the DeleteOnTermination flag set to true, would be deleted by default
    - Additional EBS volumes attached have the DeleteOnTermination flag set to false are not deleted but just detached from the instance



EC2 Hibernate

* EC2 instance hibernate signals the operating system to perform hibernation (suspend-to-disk).
* Hibernation saves the contents from the instance memory (RAM) to the EBS root volume.
* Instance’s EBS root volume and any attached EBS data volumes are persisted
* Hibernation prerequisites
  + Supported instance families – C3, C4, C5, M3, M4, M5, R3, R4, R5, & T2
  + Instance RAM size – must be less than 150 GB.
  + Instance size – not supported for bare metal instances.
  + Supported AMIs must be an HVM AMI that supports hibernation
  + Root volume type – must be EBS volume
  + EBS root volume size – must be large enough to store the RAM contents
  + EBS root volume encryption must be encrypted to ensure the protection of sensitive content that is in memory at the time of hibernation
  + Enable hibernation at launch is not supported
  + Purchasing options – Only On-Demand Instances and Reserved Instances supported
* Limitations or Unsupported Actions
  + Changing the instance type or size of a hibernated instance
  + Creating snapshots or AMIs from hibernated instances or instances for which hibernation is enabled
  + instance store-backed instances can’t stop or hibernate
  + can’t hibernate an instance that has more than 150 GB of RAM.
  + can’t hibernate an instance that is in an Auto Scaling group or used by ECS. If the instance is in an Auto Scaling group is hibernated, the EC2 Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance.
  + instance cannot be hibernated for more than 60 days.

AWS Certification Exam Practice Questions

1. What does Amazon EC2 provide?
   1. **Virtual servers in the Cloud**
   2. A platform to run code (Java, PHP, Python), paying on an hourly basis.
   3. Computer Clusters in the Cloud.
   4. Physical servers, remotely managed by the customer.
2. A user has enabled termination protection on an EC2 instance. The user has also set Instance initiated shutdown behavior to terminate. When the user shuts down the instance from the OS, what will happen?
   1. The OS will shutdown but the instance will not be terminated due to protection
   2. **It will terminate the instance**
   3. It will not allow the user to shutdown the instance from the OS
   4. It is not possible to set the termination protection when an Instance initiated shutdown is set to Terminate
3. A user has launched an EC2 instance and deployed a production application in it. The user wants to prohibit any mistakes from the production team to avoid accidental termination. How can the user achieve this?
   1. **The user can the set DisableApiTermination attribute to avoid accidental termination**
   2. It is not possible to avoid accidental termination
   3. The user can set the Deletion termination flag to avoid accidental termination
   4. The user can set the InstanceInitiatedShutdownBehavior flag to avoid accidental termination
4. You have been doing a lot of testing of your VPC Network by deliberately failing EC2 instances to test whether instances are failing over properly. Your customer who will be paying the AWS bill for all this asks you if he being charged for all these instances. You try to explain to him how the billing works on EC2 instances to the best of your knowledge. What would be an appropriate response to give to the customer in regards to this?
   1. Billing commences when Amazon EC2 AMI instance is completely up and billing ends as soon as the instance starts to shutdown.
   2. **Billing commences when Amazon EC2 initiates the boot sequence of an AMI instance and billing ends when the instance shuts down.**
   3. Billing only commences only after 1 hour of uptime and billing ends when the instance terminates.
   4. Billing commences when Amazon EC2 initiates the boot sequence of an AMI instance and billing ends as soon as the instance starts to shutdown