AWS STUDY NOTES FOR SA & SYSOPS

By www.HadoopExam.com Version 1.0.0

Note: This study notes are useful for AWS certifications (However, it is expected you have very well and in depth knowledge to clear the exam. AWS certification does not asks direct questions, they ask questions based on fundamentals and assume you have good hands-on AWS services. This study notes/book will help you to revise your concepts at the last moment of your exam and does not contain any material from the real exam directly. Only concepts are included, which is very important to answer real exam questions, especially for the multi-answer questions. (This study guide will help you in all AWS certification exams). So start right away. Don't forget to create account on our site to get access to more material made only for free registered user.

Contents

EC2 Introduction:

1. EC2 Virtual Servers also known as instances.
2. **Instance types**: It is decided based on CPU, memory, storage, and networking capacity of instances.
3. If you want to login EC2 instances remotely, you have to use key pairs (AWS stores the public key, and you store the private key in a secure place).
4. **Instance Store (Temporary):** It is a temporary data store in EC2, that's deleted when you stop or terminate your instance.
5. **Persistent storage**: Storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as Amazon EBS volumes.
8. **Cloudwatch**: To monitor basic statistics for your instances and Amazon EBS volumes, you can use Amazon CloudWatch.
9. **CloudTrail**: To monitor the calls made to the Amazon EC2 API for your account, including calls made by the AWS Management Console, command line tools, and other services, use AWS CloudTrail.

10. **RDS** : Although you can set up or install your own database on an EC2 instance e.g. Oracle, MySQL etc. Use Amazon RDS which offers the advantage of handling your database management tasks, such as patching the software, backing up, and storing the backups.

11. **On-Demand instances**: Pay for the instances that you use by the hour, with no long-term commitments or up-front payments.

12. **Reserved Instances**: Make a low, one-time, up-front payment for an instance, reserve it for a one or three-year term, and pay a significantly lower hourly rate for these instances.

13. **Spot instances (Instance bidding e.g. Stock Bid):** Specify the maximum hourly price that you are willing to pay to run a particular instance type. The Spot price fluctuates based on supply and demand, but you never pay more than the maximum price you specified. If the Spot price moves higher than your maximum price, Amazon EC2 shuts down your Spot instances.

AMI: Amazon Machine Image

1. An AMI includes the following:
   - A template for the root volume for the instance (for example, an operating system, an application server, and applications)
   - It also contains the Launch permissions that control which AWS accounts can use the AMI to launch instances (AMI may or may not be used by everybody). So only people who have permissions can use your AMI. You can have public AMI as well (Use for anybody).
   - A block device mapping that specifies the volumes to attach to the instance when it's launched.

2. **AMI Across regions**: You can copy an AMI to the same region or to different regions.

3. When you are finished launching an instance from an AMI, you can deregister the AMI.

4. **Custom AMI**: You can customize the instance that you launch from a public AMI and then save that configuration as a custom AMI for your own use.

5. **Instance Type and AMI**: You can launch different types of instances from a single AMI.

6. You can use sudo to run commands that require root privileges.

7. The root device for your instance contains the image used to boot the instance.

7A: EC2 instances support two types for block level storage
   - **Elastic Block Store (EBS):** It is, somewhat like network attached storage.
   - **Instance Store** (It is physically attached to EC2 instance)

7B. EC2 Instances can be launched using either Elastic Block Store (EBS) or Instance Store volume as root volumes and additional volumes.

7C. EC2 instances can be launched by choosing between AMIs backed by Amazon EC2 instance store and AMIs backed by Amazon EBS. However, AWS recommends use of AMIs backed by Amazon EBS, because they launch faster and use persistent storage.

7D. **Instance Store**: Also known as Ephemeral storage.

7E. Instance store volumes accesses storage from disks that are physically attached to the host computer.

7F. When an Instance stored instance is launched, the image that is used to boot the instance is copied to the root volume (typically sda1).

7G. Instance store provides temporary block-level storage for instances.

8. Your instance may include local storage volumes, known as instance store volumes, which you can configure at launch time with block device mapping.

8A. Key points for Instance store backed Instance

- Boot time is slower then EBS backed volumes and usually less than 5 min
- Can be selected as Root Volume and attached as additional volumes
- Instance store backed Instances can be of maximum 10GiB volume size
- Instance store volume can be attached as additional volumes only when is the Instance is being launched and cannot be attached once the Instance is up and running
- Instance store backed Instances cannot be stopped as one of the main reason being when stopped and started AWS does not guarantee the Instance would be launched in the same host.
- Data on Instance store volume is LOST in following scenarios :-
    - Failure of an underlying drive
    - Stopping an EBS-backed instance where instance store are additional volumes
    - Termination of the Instance
- Data on Instance store volume is NOT LOST when the instance is rebooted
- Instance store backed Instances cannot be upgraded
- When you launch an Amazon EC2 instance store-backed AMI, all the parts have to be retrieved from Amazon S3 before the instance is available.

9. An EBS volume behaves like a raw, unformatted, external block device that you can attach to a single instance and are not physically attached to the Instance host computer (more like a network attached storage).

9A. Key points for EBS backed Instance

- Boot time is very fast usually less than a min
- Can be selected as Root Volume and attached as additional volumes
- EBS backed Instances can be of maximum 16TiB volume size depending upon the OS
- EBS volume can be attached as additional volumes when the Instance is launched and even when the Instance is up and running
- Data on the EBS volume is LOST only if the Root Volume is EBS backed and the Delete On Termination flag is enabled (This is default behavior)

Data on EBS volume is NOT LOST in following scenarios :-

- Reboot on the Instance
- Stopping an EBS-backed instance

- Termination of the Instance for the additional EBS volumes. Additional EBS volumes are detached with their data intact
- When EBS-backed instance is in a stopped state, various instance and volume-related tasks can be done for e.g. you can modify the properties of the instance, you can change the size of your instance or update the kernel it is using, or you can attach your root volume to a different running instance for debugging or any other purpose
- **EBS volumes are tied to a single AZ in which they are created.**
- EBS volumes are automatically replicated within that zone to prevent data loss due to failure of any single hardware component
- EBS backed Instances can be upgraded for instance type, Kernel, RAM disk and user data
- With an Amazon EBS-backed AMI, parts are lazily loaded and only the parts required to boot the instance need to be retrieved from the snapshot before the instance is available.
- However, the performance of an instance that uses an Amazon EBS volume for its root device is slower for a short time while the remaining parts are retrieved from the snapshot and loaded into the volume.
- Review the rules in your security groups regularly, and ensure that you apply the principle of least privilege—only open up permissions that you require

10. Consider creating a **bastion security group** that allows external logins, and keep the remainder of your instances in a group that does not allow external logins.

11. Disable password-based logins for instances launched from your AMI. Passwords can be found or cracked, and are a security risk.

12. When an instance is in a stopped state, you can attach or detach Amazon EBS volumes. You can also create an AMI from the instance, and you can change the kernel, RAM disk, and instance type.

13. Instances with Amazon EBS volumes for the root device default to stop, and instances with instance-store root devices are always terminated as the result of an instance shutdown.

14. All AMIs are categorized as either backed by Amazon EBS, which means that the root device for an instance launched from the AMI is an Amazon EBS volume, or backed by instance store, which means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.

Regions and AZs:

1. Resources aren't replicated across regions unless you do so specifically.

2. Each region is completely independent.

3. Each Availability Zone is isolated, but the Availability Zones in a region are connected through low-latency links

4. Amazon EC2 resources are either global, tied to a region, or tied to an Availability Zone.

5. When you view your resources, you'll only see the resources tied to the region you've specified. This is because regions are isolated from each other, and we don't replicate resources across regions automatically.

6. When you launch an instance, you must select an AMI that's in the same region. If the AMI is in another region, you can copy the AMI to the region you're using.

7. All communication between regions is across the public Internet. Therefore, you should use the appropriate encryption methods to protect your data. Data transfer between regions is charged at the Internet data transfer rate for both the sending and the receiving instance.

8. You can also use Elastic IP addresses to mask the failure of an instance in one Availability Zone by rapidly remapping the address to an instance in another Availability Zone.

9. An Availability Zone is represented by a region code followed by a letter identifier; for example, us-east-1a.

10. Ensure that resources are distributed across the Availability Zones for a region, we independently map Availability Zones to identifiers for each account. For example, your Availability Zone us-east-1a might not be the same location as us-east-1a for another account. There's no way for you to coordinate Availability Zones between accounts.

11. As Availability Zones grow over time, our ability to expand them can become constrained. If this happens, we might restrict you from launching an instance in a constrained Availability Zone unless you already have an instance in that Availability Zone. Eventually, we might also remove the constrained Availability Zone from the list of Availability Zones for new customers. Therefore, your account might have a different number of available Availability Zones in a region than another account.

12. An AWS account provides multiple regions so that you can launch Amazon EC2 instances in locations that meet your requirements. For example, you might want to launch instances in Europe to be closer to your European customers or to meet legal requirements.

13. When you launch an instance, you can optionally specify an Availability Zone in the region that you are using.

15. If you need to, you can migrate an instance from one Availability Zone to another. For example, if you are trying to modify the instance type of your instance and we can't launch an instance of the new instance type in the current Availability Zone, you could migrate the instance to an Availability Zone where we can launch an instance of that instance type. The migration process involves creating an AMI from the original instance, launching an instance in the new Availability Zone, and updating the configuration of the new instance

Auto Scaling and ELB:

1. When you have Auto Scaling enabled it will automatically increase the number of EC2 instances number of request to service increases, and decrease the number of EC2 instances when number of requests reduces.

2. If you have a higher load on your servers, use ELB service which will distribute the incoming web traffic load automatically among all the running EC2 instances in Auto scaling group.

3. Load balancers also help us to monitor the incoming traffic. All the traffic comes via only ELB (single point of contact) for all incoming traffic to the instances in an Auto Scaling group.

5. You can attach more than one ELB to your auto scaling group. Once you attach an ELB to Auto scaling group it will automatically registers the instances from the group and balance the incoming traffic across the instances.

7. ELB uses IP address of the instances to register them and routes requests to the primary IP address of the primary interface (eth0) of the instance.

8. You can also detach the ELB from scaling group, once you detach it will be having Removing state till deregistering of the instances in the group.

9. If connection draining is enabled, ELB waits for in-flight requests to complete before deregistering the instances.

10. Impact of Deregistering ELB on Instances: They will remain in their state like running instance will remain in running state only.

11. Suppose you have suspended Load Balancer and few more instances are added to scaling group , during the load balancer in suspended state instances launched during the suspension period are not added to the load balancer, after resumption we have to manually register them again.

12. Once you create an auto scaling group, you can attach ELB to them and all the instances running in scaling group will be automatically attached to ELB, you can have more than one ELB to group.

13. We can have auto scaling across AZs in single and same region. Similarly if you have attached an ELB to that auto scaling group, then it will also distribute the traffic across Azs.

14. Both the ELB and Scaling group separately check the health status for each instances separately. Scaling group check instance health using the instance status check. If instance is in failed status they will marked as unhealthy. ELB also perform the health status of instances using ping. However, Auto scaling is not depend on status check is done by ELB.

15. ELB does its own health check to make sure that traffic is routed to healthy instances.

16. Once an ELB is registered with a Scaling group, it can be configured to use the results of the ELB health check in addition to the EC2 instance status checks to determine the health of the EC2 instances in your Auto Scaling group.

17. You can also use Cloudwatch monitor matrix to monitor the EC2 instances and scaling group.

18. Once you have registered the ELB with scaling group, Scaling group can be configured to use ELB metrics e.g. such as request latency or request count to scale the application automatically.

19. Using the auto scaling, we can make sure that minimum number of required instances always available. However, we can attach some policy to auto scaling group. Which can help launch and terminate EC2 instances to handle any increase or decrease in demand on your application.

20. Auto Scaling attempts to distribute instances evenly between the Availability Zones that are enabled for your Auto Scaling group. Auto Scaling does this by attempting to launch new instances in the Availability Zone with the fewest instances. If the attempt fails, however, Auto Scaling attempts to launch the instances in another Availability Zone until it succeeds

21. Launch Configuration: You can create a template known as launch configuration which has information e.g. AMI, Instance type, their key pairs, security group and block device mapping.

22. Once you created launch configuration, you cannot change it. If you want change please create new one. Also same launch configuration can be attached to multiple scaling group.

23. You have to have following for auto scaling group:

- Launch Configuration, Desired Capacity, Availability Zones or Subnets, Metrics & Health Checks

24. Auto Scaling groups cannot span multiple regions

25. After changing the launch configuration of Auto Scaling group, any new instances are launched using with this new configuration parameters, but existing instances are not affected.

26. While status check of EC2 instance, scaling group finds state other than RUNNING or IMPAIRED it will be considered as unhealthy and will launch new instance as replacement.

27. If you have configured scaling group to use instance status using both instance status as well as status reported by load balancer then it will mark instance as unhealthy if instance status is other than RUNNING /IMPAIRED or reported by ELB as OutOfService. Once is marked unhealthy it will be scheduled to be replaced and will not be automatically recovers its health.

28. When your instance is terminated, any associated Elastic IP addresses are disassociated and are not automatically associated with the new instance. You must associate these Elastic IP addresses with the new instance manually. Similarly, when your instance is terminated, its attached EBS volumes are detached. You must attach these EBS volumes to the new instance manually.

29. Attaching/Detaching of an EC2 instance can be done only if

- Instance is in the running state.
- AMI used to launch the instance must still exist.
- Instance is not a member of another Auto Scaling group.
- Instance is in the same Availability Zone as the Auto Scaling group.
- If the Auto Scaling group is associated with a load balancer, the instance and the load balancer must both be in the same VPC.

30. Scaling based on a schedule allows you to scale your application in response to predictable load changes. For e.g. last day of the month, last day of an financial year

31. Auto Scaling guarantees the order of execution for scheduled actions within the same group, but not for scheduled actions across groups

32. Multiple Scheduled Actions can be specified but should have unique time value and they cannot have overlapping time scheduled which will lead to its rejection

33. Dynamic Scaling: Allows you to scale automatically in response to the changing demand for e.g. scale out in case CPU utilization of the instance goes above 70% and scale in when the CPU utilization goes below 30%

34. Auto Scaling group uses a combination of alarms and policies to determine when the conditions for scaling are met.

35. An Auto Scaling group can have more than one scaling policy attached to it any given time.

36. Each Auto Scaling group would have at least two policies: one to scale your architecture out and another to scale your architecture in.

37. If an Auto Scaling group has multiple policies, there is always a chance that both policies can instruct the Auto Scaling to Scale Out or Scale In at the same time. When these situations occur, Auto Scaling chooses the policy that has the greatest impact on the Auto Scaling group. For e.g. if two policies are triggered at the same time and Policy 1 instructs to scale out the instance by 1 while Policy 2 instructs to scale out the instances by 2, Auto Scaling will use the Policy 2 and scale out the instances by 2 as it has a greater impact

38. Termination policy helps the Auto Scaling to decide which instances it should terminate first when you have Auto Scaling automatically scale in. Auto Scaling specifies a default termination policy and also allows you to create a customized one

39. Instance protection controls whether Auto Scaling can terminate a particular instance or not.
Instance protection can be enabled on an Auto Scaling group or an individual instance as well, at any time

40. Instance protection does not protect for the below cases

- Manual termination through the Amazon EC2 console, the terminate-instances command, or the Terminate Instances API.
- Termination if it fails health checks and must be replaced.
- Spot instances in an Auto Scaling group from interruption.

41. Auto Scaling allows you to put the In-service instance in the Standby state during which the instance is still a part of the Auto Scaling group but does not serve any requests. This can be used to either troubleshoot an instance or update and instance and return the instance back to service

42. If a load balancer is associated with Auto Scaling, the instance is automatically deregistered when the instance is in Standby state and registered again when the instance exits the Standby state.

EC2 Root Device Volume:

1. /dev/sda it's 100% your internal drive. Your external drive may be sdb, sdc or another one.

- The disk names in Linux are alphabetical. /dev/sda is the first hard drive (the primary master)
- Root device volume contains the image used to boot the instance.

2. Amazon EC2 instance store: which means the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.
3. Backed by Amazon EBS: This means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot.
4. We recommend that you use AMIs backed **by Amazon EBS**, because they launch faster and use persistent storage.
5. Instances that use instance stores for the root device automatically have one or more instance store volumes available, with one volume serving as the root device volume.
6. When an instance is launched, the image that is used to boot the instance is copied to the root volume. Note that you can optionally use additional instance store volumes, depending on the instance type.
7. Any data on the instance store volumes persists as long as the instance is running, but this data is deleted when the instance is terminated
8. **Instance store-backed instances do not support the Stop action**
9. **Instance Store (ephemeral storage):** Temporary block level storage. It is physically attached to your instance.
10. Instance stores cost is part of EC2 instance only.
11. If you plan to use Amazon EC2 instance store-backed instances, we highly recommend that you distribute the data on your instance stores across multiple Availability Zones.
12. You should also back up critical data on your instance store volumes to persistent storage on a regular basis.
13. Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached.
14. When you launch an Amazon EBS-backed instance, we create an Amazon EBS volume for each Amazon EBS snapshot referenced by the AMI you use.
15. An Amazon EBS-backed instance can be stopped and later restarted without affecting data stored in the attached volumes.
16. There are various instance and volume-related tasks you can do when an Amazon EBS-backed instance is in a stopped state. For example, you can modify the properties of the instance, you can change the size of your instance or update the kernel it is using, or you can attach your root volume to a different running instance for debugging or any other purpose.
17. If an Amazon EBS-backed instance fails, you can restore your session
18. By default, the root device volume for an AMI backed by Amazon EBS is deleted when the instance terminates.

19. Use the modify-instance-attribute command to preserve the root volume by including a block device mapping that sets its DeleteOnTermination attribute to false. (Hence, you can change this parameter event instance is running).

Setup:

1. VPCs are specific to a region, so you should select the same region in which you created your key pair.
2. Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level.
3. Note that if you plan to launch instances in multiple regions, you'll need to create a security group in each region.
4. Security Group: Source is set to anywhere using IP 0.0.0.0/0.
5. Security Group: To specify an individual IP address in CIDR notation, add the routing prefix /32. For example, if your IP address is 203.0.113.25, specify 203.0.113.25/32. If your company allocates addresses from a range, specify the entire range, such as 203.0.113.0/24.
6. Don't select the Proceed without a key pair option. If you launch your instance without a key pair, then you can't connect to it.

AWS Admin Permissions

1. AWS provides the root or system privileges only for a limited set of services, which includes Elastic Cloud Compute (EC2), Elastic MapReduce (EMR),Elastic BeanStalk, Opswork
2. AWS does not provide root privileges for managed services like RDS, DynamoDB, S3, Glacier etc

AWS Config:

1. AWS Config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance
2. In cases where several configuration changes are made to a resource in quick succession (i.e., within a span of few minutes), AWS Config will only record the latest configuration of that resource; this represents the cumulative impact of that entire set of changes
3. Use AWS Config for supported services and use an automated process via APIs for unsupported services

Consolidate billing

1. Allows receiving a combined view of charges incurred by all the associated accounts as well as each of the accounts.
2. This account is strictly an accounting and billing feature.

3. This is not a method for controlling accounts, or provisioning resources for accounts.
4. Payer account cannot access data belonging to the linked account owners
5. However, access to the Payer account users can be granted through Cross Account Access roles
6. Volume Pricing Discounts
    - For billing purposes, AWS treats all the accounts on the consolidated bill as if they were one account.
    - AWS combines the usage from all accounts to determine which volume pricing tiers to apply, giving you a lower overall price whenever possible.
7. Linked accounts receive the cost benefit from other's Reserved Instances only if instances are launched in the same Availability Zone where the Reserved Instances were purchased
8. Capacity reservation only applies to the product platform, instance type, and Availability Zone specified in the purchase
    - For e.g., Amit and Rakesh each have an account on Amit's consolidated bill. Rakesh has 5 Reserved Instances of the same type, and Amit has none. During one particular hour, Rakesh uses 3 instances and Amit uses 6, for a total of 9 instances used on Amit's consolidated bill. AWS will bill 5 as Reserved Instances, and the remaining 4 as normal instances.
9. Paying account should be used solely for billing purposes
10. Master (Payee) account can view only the AWS billing details of the linked accounts
11. Any single instance from all the three accounts can get the benefit of AWS RI pricing if they are running in the same zone and are of the same size
12. The payee account will send a request to the linked account to be a part of consolidated billing.

Best Practices for Amazon EC2:

1. Regularly patch, update, and secure the operating system and applications on your instance.
2. Implement the least permissive rules for your security group.
3. Use separate Amazon EBS volumes for the operating system versus your data. Ensure that the volume with your data persists after instance termination.
4. If you use instance store for database storage, ensure that you have a cluster with a replication factor that ensures fault tolerance.
5. Design your applications to handle dynamic IP addressing when your instance restarts.
6. The best practice for securing your web server is to install support for HTTPS (HTTP Secure), which protects your data with SSL/TLS encryption.

Important Notes:

1. It is strongly recommended that you associate an Elastic IP address (EIP) to the instance you are using to host a WordPress blog. This prevents the public DNS address for your instance from changing and breaking your installation. If you own a domain name and you want to use it for

your blog, you can update the DNS record for the domain name to point to your EIP address. You can have **one EIP address associated with a running instance at no charge**.

2. You can register a domain name with Amazon Route 53 and associate your instance's EIP address with your domain name.
3. Move your MySQL database to Amazon RDS to take advantage of the service's ability to scale automatically.
4. If you have Joomla CMS installation, which is automatically configured using the public DNS address for your EC2 instance. Once you stop and restart the instance, the **public DNS address changes (unless it is associated with an Elastic IP address**) and your website will not work anymore because it references resources at an address that no longer exists (or is assigned to another EC2 instance).
5. You can launch multiple EC2 instances from your AMI and then use Elastic Load Balancing to distribute incoming traffic for your application across these EC2 instances.
6. You can use Auto Scaling to maintain a minimum number of running instances for your application at all times. Auto Scaling can detect when your instance or application is unhealthy and replace it automatically to maintain the availability of your application. You can also use Auto Scaling to scale your Amazon EC2 capacity up or down automatically based on demand, using criteria that you specify.
7. Auto Scaling with Elastic Load Balancing to ensure that you maintain a specified number of healthy EC2 instances behind your load balancer. Note that these instances do not need public IP addresses, because traffic goes to the load balancer and is then routed to the instances.
8. **subnet can be across Az**
9. Increase the Availability of Your Application on Amazon EC2 , create a VPC with one public subnet in two or more Availability Zones.
10. When you use ELB and Auto scaling, **pre-requisite is an AMI** which will be used by Auto-scaling to launch new instance based on AMI.
11. If you have some scripts, which needs to be executed as soon as your instance started. **Please add this script in User data, while configuring Auto-scaling**.
12. When you are using load balancer in front of your instances, then it is needed that in your security group, you must allow HTTP traffic and health checks from the load balancer.
13. You must assign the IAM role when you create the new instance. **You can't assign a role to an instance that is already running**. For existing instances, you must create an image of the instance, launch an instance from that image, and assign the IAM role as you launch the instance.
14. Instances require an AWS Identity and Access Management (IAM) role that enables the instance to communicate with Amazon EC2 Simple Systems Manager (SSM).

AMI Types:

1. **Boot time** -> Amazon EBS-Backed: Usually less than 1 minute. Amazon Instance Store-Backed: Usually less than 5 minutes
2. **Upgrading** -> Amazon EBS-Backed: The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped. Amazon Instance Store-Backed: Instance attributes are fixed for the life of an instance.
3. **Ephemeral disk** is a temporary storage that it is added to your instance, and depending on your instance type the bigger is such storage.
4. **Instance store is dedicated to a particular instance.**
5. **You can't attach instance store volumes to an instance after you've launched it.**
6. After you launch the instance, you must ensure that the instance store volumes for your instance are formatted and mounted before you can use them.
7. Note that the root volume of an instance store-backed instance is mounted automatically.
8. When you launch an Amazon EC2 instance store-backed AMI, all the parts have to be retrieved from Amazon S3 before the instance is available. With an Amazon EBS-backed AMI, only the parts required to boot the instance need to be retrieved from the snapshot before the instance is available.
9. For the best performance, we recommend that you use current generation instance types and HVM AMIs when you launch your instances.
10. Linux Amazon Machine Images use one of two types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). The main difference between PV and HVM AMIs is the way in which they boot and whether they can take advantage of special hardware extensions (CPU, network, and storage) for better performance.
11. Unlike PV guests, HVM guests can take advantage of hardware extensions that provide fast access to the underlying hardware on the host system.
12. Amazon can't vouch for the integrity or security of AMIs shared by other Amazon EC2 users. Therefore, you should treat shared AMIs as you would any foreign code that you might consider deploying in your own data center and perform the appropriate due diligence.
13. Amazon's public images have an aliased owner, which appears as amazon in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.
14. Before you use a shared AMI, take the following steps to confirm that there are no pre-installed credentials that would allow unwanted access to your instance by a third party and no pre-configured remote logging that could transmit sensitive data to a third party.

15. To ensure that you don't accidentally lose access to your instance, we recommend that you initiate two SSH sessions and keep the second session open until you've removed credentials that you don't recognize and confirmed that you can still log into your instance using SSH.
16. To prevent preconfigured remote logging, you should delete the existing configuration file and restart the rsyslog service.
17. AMIs are a regional resource. Therefore, sharing an AMI makes it available in that region. To make an AMI available in a different region, copy the AMI to the region and then share it.
18. If an AMI has a product code, you can't make it public. You must share the AMI with only specific AWS accounts.
19. **You do not need to share the Amazon EBS snapshots that an AMI references in order** to share the AMI. Only the AMI itself needs to be shared; the system automatically provides the instance access to the referenced Amazon EBS snapshots for the launch.
20. If you have created a public AMI, or shared an AMI with another AWS user, **you can create a bookmark** that allows a user to access your AMI and launch an instance in their own account immediately. This is an easy way to share AMI references, so users don't have to spend time finding your AMI in order to use it.
21. For AMIs backed by instance store, we recommend that your AMIs download and upgrade the Amazon EC2 AMI creation tools during startup. This ensures that new AMIs based on your shared AMIs have the latest AMI tools.
22. Using a fixed root password for a public AMI is a security risk that can quickly become known. Even relying on users to change the password after the first login opens a small window of opportunity for potential abuse. To solve this problem, **disable password-based remote logins for** the root user.
23. When you work with shared AMIs, a best practice is to disable direct root logins.
24. If you plan to share an AMI derived from a public AMI, remove the existing SSH host key pairs located in /etc/ssh. This forces SSH to generate new unique SSH key pairs when someone launches an instance using your AMI, improving security and reducing the likelihood of "man-in-the-middle" attacks.
25. If you forget to remove the existing SSH host key pairs from your public AMI, our routine auditing process notifies you and all customers running instances of your AMI of the potential security risk. After a short grace period, we mark the AMI private.
26. Currently, there is no easy way to know who provided a shared AMI, because each AMI is represented by an account ID. We recommend that you post a description of your AMI, and the AMI ID, in the Amazon EC2 forum. This provides a convenient central location for users who are interested in trying new shared AMIs. You can also post the AMI to the Amazon Machine Images (AMIs) page.

27. We recommend using the --exclude directory option on ec2-bundle-vol to skip any directories and subdirectories that contain secret information that you would not like to include in your bundle. In particular, exclude all user-owned SSH public/private key pairs and SSH authorized_keys files when bundling the image. The Amazon public AMIs store these in /root/.ssh for the root account, and /home/user_name/.ssh/ for regular user accounts.
28. **Always delete the shell history before bundling**. If you attempt more than one bundle upload in the same AMI, the shell history contains your secret access key.
29. Bundling a running instance requires your private key and X.509 certificate. Put these and other credentials in a location that is not bundled (such as the instance store).
30. The developer of a paid AMI can enable you to purchase a paid AMI that isn't listed in AWS Marketplace. The developer provides you with a link that enables you to purchase the product through Amazon
31. You can retrieve the AWS Marketplace product code for your instance using its instance metadata.
32. During the AMI-creation process, Amazon EC2 creates snapshots of your instance's root volume and any other EBS volumes attached to your instance. If any volumes attached to the instance are encrypted, the new AMI only launches successfully on instances that support Amazon EBS encryption.
33. In the navigation pane, choose Instances and select your instance. Choose Actions, Image, and Create Image. If this option is disabled, your instance isn't an Amazon EBS-backed instance.
34. You can convert an instance store-backed Linux AMI that you own to an Amazon EBS-backed Linux AMI.
35. You can't convert an instance store-backed Windows AMI to an Amazon EBS-backed Windows AMI and you cannot convert an AMI that you do not own.
36. AMIs that are backed by Amazon EBS snapshots can take advantage of Amazon EBS encryption. Snapshots of both data and root volumes can be encrypted and attached to an AMI.
37. The CopyImage action can be used to create an AMI with encrypted snapshots from an AMI with unencrypted snapshots.
38. You can create an AMI from a running Amazon EC2 instance (with or without encrypted volumes) using either the Amazon EC2 console or the command line
39. This scenario starts with an AMI backed by a root-volume snapshot (encrypted to key #1), and finishes with an AMI that has two additional data-volume snapshots attached (encrypted to key #2 and key #3). The CopyImage action cannot apply more than one encryption key in a single operation. However, you can create an AMI from an instance that has multiple attached volumes encrypted to different keys. The resulting AMI has snapshots encrypted to those keys and any instance launched from this new AMI also has volumes encrypted to those keys.

40. You can copy an Amazon Machine Image (AMI) within or across an AWS region using the AWS Management Console, the command line, or the Amazon EC2 API, all of which support the CopyImage action. Both Amazon EBS-backed AMIs and instance store-backed AMIs can be copied.
41. The source AMI can be changed or deregistered with no effect on the target AMI. The reverse is also true.
42. AWS does not copy launch permissions, user-defined tags, or Amazon S3 bucket permissions from the source AMI to the new AMI.
43. You can copy an AMI across AWS accounts. This includes AMIs with encrypted snapshots, but does not include encrypted AMIs.
44. **You can't copy an encrypted AMI between accounts**. Instead, if the underlying snapshot and encryption key have been shared with you, you can copy the snapshot to another account while re-encrypting it with a key of your own, and then register this privately owned snapshot as a new AMI.
45. When you launch an instance from an AMI, it resides in the same region where the AMI resides. If you make changes to the source AMI and want those changes to be reflected in the AMIs in the target regions, you must recopy the source AMI to the target regions.
46. When you first copy an instance store-backed AMI to a region, we create an Amazon S3 bucket for the AMIs copied to that region. All instance store-backed AMIs that you copy to that region are stored in this bucket.
47. Prior to copying an AMI, you must ensure that the contents of the source AMI are updated to support running in a different region. For example, you should update any database connection strings or similar application configuration data to point to the appropriate resources. Otherwise, instances launched from the new AMI in the destination region may still use the resources from the source region, which can impact performance and cost.
48. **Encrypting during copying applies only to Amazon EBS-backed AMIs**. Because an instance-store-backed AMIs does not rely on snapshots, the CopyImage action cannot be used to change its encryption status.
49. The following table shows encryption support for various scenarios. Note that while it is possible to copy an unencrypted snapshot to yield an encrypted snapshot, you cannot copy an encrypted snapshot to yield an unencrypted one.

Scenario	Description	Supported
1	Unencrypted-to-unencrypted	Yes
2	Encrypted-to-encrypted	Yes
3	Unencrypted-to-encrypted	Yes
4	Encrypted-to-unencrypted	No

50. You can deregister an AMI when you have finished using it. After you deregister an AMI, you can't use it to launch new instances.
51. When you deregister an AMI, it doesn't affect any instances that you've already launched from the AMI.
52. When you deregister an Amazon EBS-backed AMI, it doesn't affect the snapshot that was created for the root volume of the instance during the AMI creation process. You'll continue to incur storage costs for this snapshot. Therefore, if you are finished with the snapshot, you should delete it.
53. When you deregister an instance store-backed AMI, it doesn't affect the files that you uploaded to Amazon S3 when you created the AMI. You'll continue to incur usage costs for these files in Amazon S3. Therefore, if you are finished with these files, you should delete them.
54. Micro instances can only be launched by with EBS volumes.
55. You can detach an EBS volume from an instance and attach it to another instance. But you cannot do the same with Instance store.
56. If your data can easily re-generated (and you are less worried about data loss, then you should use instance store backed instances).
57. If you want to create databases with high IO like nosql databases MongoDB, Cassandra etc. Then use Instance store backed **SSD** instance.

Instance Types

1. EBS are network attached storages thus it will consume your network bandwidth to connect the the volumes, so if you have a network intensive application on your EC2 instance it might affect the connection between your instance to your EBS volumes. Optimized instances are just instances with a "dedicated" connection to EBS volumes, that means that you will have a dedicated amount of bandwidth just for your EBS volumes. Each instance has a limit of bandwidth for their connections to the EBS volumes.
2. To obtain additional, dedicated capacity for Amazon EBS I/O, you can launch some instance types as EBS–optimized instances (Bandwidth between EC2 instance and EBS volumes). Some instance types are EBS–optimized by default.
3. To maximize the networking and bandwidth performance of your instance type, you can do the following:
    - Launch supported instance types into a placement group: Instances in a common placement group can benefit from high-bandwidth (10 Gbps), low-latency networking. Instance types that **support 10 Gbps network speeds** can only take advantage of those network speeds when launched in a placement group.

- Enable enhanced networking for supported current generation instance types to get significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies.

4. T2 instances are lowest cost instance type, with a **burstable** CPU.
5. T2 instances are designed to provide moderate baseline performance (minimum CPU all time) and the capability to burst to significantly higher performance as required by your workload. They are intended for workloads that don't use the full CPU often or consistently, but occasionally need to burst. T2 instances are well suited for general purpose workloads, such as web servers, developer environments, and small databases.
6. You must launch your T2 instances using an EBS volume as the root device.
7. T2 instances are available as On-Demand instances and Reserved Instances, but they are not available **as Spot instances, Scheduled Instances, or Dedicated instances**. They are also not supported on a Dedicated Host.
8. There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types.
9. C4 instances are ideal for **compute-bound** applications that benefit from high performance processors. C4 instances are well suited for the following applications:
   - Batch processing workloads
   - Media transcoding
   - High-traffic web servers, massively multiplayer online (MMO) gaming servers, and ad serving engines
   - High performance computing (HPC) and other compute-intensive applications
10. C4 instances are **EBS-optimized by default**, and deliver dedicated block storage throughput to Amazon EBS ranging from 500 Mbps to 4,000 Mbps at no additional cost. EBS-optimized instances enable you to get **consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic** from your C4 instance.
11. You can cluster C4 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances **within a single Availability Zone**.
12. Accelerated computing instance families use hardware accelerators, or co-processors, to perform some functions, such as floating point number calculation and graphics processing, more efficiently than is possible in software running on CPUs.
13. I2 instances are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications. They are well suited for the following scenarios:
    - NoSQL databases (for example, Cassandra and MongoDB)
    - Clustered databases
    - Online transaction processing (OLTP) systems

14. I2 Instance Features
    - The primary data storage is SSD-based instance storage. Like all instance storage, these volumes persist only for the life of the instance.
15. D2 instances are designed for workloads that require high sequential read and write access to very large data sets on local storage. D2 instances are well suited for the following applications:
    - Massive parallel processing (MPP) data warehouse
    - MapReduce and Hadoop distributed computing
    - Log or data processing applications
16. The primary data storage for D2 instances is HDD-based instance storage.
17. EC2-VPC, the private IP is retained but public is released and Elastic IP addresses are also retained.
18. for EC2-Classic instances, public and private IP addresses are released when the instance is stopped, new addresses are assigned when restarted and the Elastic IP addresses are disassociated
19. Because Amazon EC2 instance store-backed AMIs can't be stopped, they're either running or terminated.
20. If the root device for your instance is an EBS volume, you can change the size of the instance simply by changing its **instance type**, which is known as resizing it (**Changing the instance type)**.
21. **Important Points about re-sizing** :
    a. While re-sizing actually you want migrate either bigger or smaller hardware and keeping all the software installed on your existing machine should remain as it is.
    b. You must be running an EBS boot instance (not instance-store or S3-based AMI). Any files on ephemeral storage (e.g., /mnt) will be lost.
    c. You can only move to a different instance type of the same architecture (32-bit or 64-bit).
    d. The private and initial public IP addresses of the instance will be different when it is running on the new hardware. Use an Elastic IP Address to keep the public IP address the same.
    e. There will be a short outage while the instance is moved to new hardware (roughly equivalent to the reboot time of normal hardware).
22. If the root device for your instance is an instance store volume and want to resize it, you must migrate your application to a new instance with the instance type that you want.
23. When you resize an instance, the resized instance usually has the same **number of instance store** volumes that you specified when you launched the original instance. If you want to add **instance store volumes**, you must migrate your application to a completely new instance with the instance type and instance store volumes that you want. **An exception to this rule is when**

you resize to a storage-intensive instance type that by default contains a higher number of volumes.

24. You can't resize an instance that was launched from a PV AMI to an instance type that is HVM only. (Virtualization compatibility is required)
25. You must stop your Amazon EBS–backed instance before you can change its instance type.
26. When you stop and start an instance, be aware of the following:
    - We move the instance to new hardware; however, the instance ID does not change.
    - If your instance is running in a VPC and has a public IP address, we release the address and give it a new public IP address. **The instance retains its private IP addresses and any Elastic IP addresses**.
    - If your instance is in an Auto Scaling group, the Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. To prevent this, **you can suspend the Auto Scaling processes for the group while you're resizing your instance**.
27. To ensure that your users can continue to use the applications that you're hosting on your instance uninterrupted, you must take any Elastic IP address that you've associated with your original instance and associate it with the new instance.
28. **Resizing in-compatible instance (Changing the instance type)**: If the current configuration of your instance is incompatible with the new instance type that you want, then you can't resize the instance to that instance type. Instead, you have to migrate your application to a new instance with a configuration that is compatible with the new instance type that you want.
29. Amazon EC2 provides the following purchasing options to enable you to optimize your costs based on your needs:
    - **On-Demand instances** — pay, by the hour, for the instances that you launch.
    - **Reserved Instances** — Purchase, at a significant discount, instances that are always available, for a term from one to three years.
    - **Scheduled Instances** — Purchase instances that are always available on the specified recurring schedule, for a one-year term.
    - **Spot instances** — Bid on unused instances, which can run as long as they are available and your bid is above the Spot price, at a significant discount. (Like you buy shares in stock market, based on bid price)
    - **Dedicated hosts** — Pay for a physical host that is fully dedicated to running your instances, and bring your existing per-socket, per-core, or per-VM software licenses to reduce costs.
    - **Dedicated instances** — Pay, by the hour, for instances that run on single-tenant hardware.
30. If you require a capacity reservation, consider Reserved Instances or Scheduled Instances.
31. Spot instances are a cost-effective choice if you can be flexible about when your applications run and if they can be interrupted.

32. Dedicated hosts can help you **address compliance requirements and reduce costs by using your existing server-bound software licenses**. If you have any existing software licenses which are bound to server e.g. Simulator Keys of www.HadoopExam.com , then you can use dedicated **hosts**, whose hardware does not changes.
33. Amazon EC2 launches the instances and then terminates them three minutes before the time period ends.
34. Reserved Instances provide you with a significant discount compared to On-Demand instance pricing. In addition, when Reserved Instances are assigned to a specific Availability Zone, they provide a capacity reservation.
35. **Each Reserved Instance that is specific to an Availability Zone can also provide a capacity reservation**.
36. Generally speaking, you can save more money choosing Reserved Instances with a higher upfront payment. There are three payment options (No Upfront, Partial Upfront, and All Upfront) and two term lengths (one-year or three-years).
37. You can find Reserved Instances offered by third-party sellers at shorter term lengths and lower prices as well.
38. When you purchase a Reserved Instance, the reservation is automatically applied to running instances that match your specified parameters.
39. **Reserved Instances do not renew automatically**; you can continue using the EC2 instance without interruption, but you will be charged On-Demand rates.
40. You can use Auto Scaling or other AWS services to launch the On-Demand instances that use your Reserved Instance benefits.
41. Both Standard and Convertible Reserved Instances can be purchased to apply to instances in a specific Availability Zone, or to instances in a region. **Reserved Instances purchased for a specific Availability Zone can be modified to apply to a region—but doing so removes the associated capacity reservation**.
42. Convertible Reserved Instances can be exchanged for other Convertible Reserved Instances with entirely different configurations, including instance type, platform, or tenancy. It is not possible to exchange Standard Reserved Instances in this way.
43. Convertible Reserved Instances— Only Amazon EC2 Standard Reserved Instances can be sold in the Reserved Instance Marketplace. Convertible Reserved Instances cannot be sold.
44. Other AWS Reserved Instances, such as Amazon RDS and Amazon ElastiCache Reserved Instances cannot be sold in the Reserved Instance Marketplace.
45. Convertible Reserved Instances are not available for purchase in the Reserved Instance Marketplace.

46. When your computing needs change, you can modify your Standard Reserved Instances and continue to benefit from your capacity reservation. Convertible Reserved Instances can be modified using the exchange process.
47. Scheduled Instances are a good choice for workloads that do not run continuously, but do run on a regular schedule. For example, you can use Scheduled Instances for an application that runs during business hours or for batch processing that runs at the end of the week.
48. Scheduled Reserved Instances (Scheduled Instances) enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified start time and duration, for a one-year term.
49. If you are flexible about when your instances run, Spot instances might meet your needs and decrease costs.
50. Amazon EC2 sets aside pools of EC2 instances in each Availability Zone for use as Scheduled Instances. Each pool supports a specific combination of instance type, operating system, and network (**EC2-Classic: this means shared network or EC2-VPC: This means private network for each account**).
51. **You can't stop or reboot Scheduled Instances**, but you can terminate them manually as needed.
52. If you terminate a Scheduled Instance before its current scheduled time period ends, you can launch it again after a few minutes. Otherwise, you must wait until the next scheduled time period.
53. Scheduled Instances are subject to the following limits:
    - The following are the only supported instance types: C3, C4, M4, and R3.
    - The required term is 365 days (one year).
    - The minimum required utilization is 1,200 hours per year.
    - You can purchase a Scheduled Instance up to three months in advance.
54. Spot instances enable you to bid on unused EC2 instances, which can lower your Amazon EC2 costs significantly. The hourly price for a Spot instance (of each instance type in each Availability Zone) is set by Amazon EC2, and fluctuates depending on the supply of and demand for Spot instances. Your Spot instance runs whenever your bid exceeds the current market price.
55. Spot instances are a cost-effective choice if you can be flexible about when your applications run and if your applications can be interrupted. For example, Spot instances are well-suited for data analysis, batch jobs, background processing, and optional tasks
56. Spot instance interruption—Amazon EC2 terminates your Spot instance when the Spot price exceeds your bid price or there are no longer any unused EC2 instances. Amazon EC2 marks the Spot instance for termination and provides a Spot instance termination notice, which gives the instance a two-minute warning before it terminates.
57. You can create launch configurations with a bid price so that Auto Scaling can launch Spot instances.

58. There are scenarios where it can be useful to run Spot instances in an Amazon EMR cluster.
59. AWS CloudFormation (**form your infrastructure in cloud**: **everything define in a template file and AWS create infrastructure for you**) enables you to create and manage a collection of AWS resources using a template in JSON format. AWS CloudFormation templates can include a Spot price.
60. When you use Spot instances, you must be prepared for interruptions. Amazon EC2 can interrupt your Spot instance when the Spot price rises above your bid price, when the demand for Spot instances rises, or when the supply of Spot instances decreases.
61. **Note that you can't stop and start an Amazon EBS-backed instance if it is a Spot instance, but you can reboot or terminate it**.
62. Note that you can't stop and start an Amazon EBS-backed instance if it is a Spot instance, but you can reboot or terminate it.
    - Shutting down a Spot instance on OS-level results in the Spot instance being terminated. It is not possible to change this behavior.
63. Specify an Availability Zone group in your Spot instance request to tell the Spot service to launch a set of Spot instances in the same Availability Zone. Note that Amazon EC2 need not terminate all instances in an Availability Zone group at the same time. If Amazon EC2 must terminate one of the instances in an Availability Zone group, the others remain running.
64. A Spot fleet is a collection, or fleet, of Spot instances. The Spot fleet attempts to launch the number of Spot instances that are required to meet the target capacity that you specified in the Spot fleet request. The Spot fleet also attempts to maintain its target capacity fleet if your Spot instances are interrupted due to a change in Spot prices or available capacity.
    - A Spot instance pool is a set of unused EC2 instances with the same instance type, operating system, Availability Zone, and network platform (EC2-Classic or EC2-VPC).
65. The following instance types are not supported for Spot:
    - T2 , HS1
66. By default, there is an account limit of 20 Spot instances per region.
67. **Spot instances EBS encryption is not supported: You can specify encrypted EBS volumes in the launch specification for your Spot instances, but these volumes are not encrypted**.
68. An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Dedicated Hosts allow you to use your existing per-socket, per-core, or per-VM software licenses
69. **Dedicated Hosts and Dedicated Instances are different things**: Dedicated Hosts and Dedicated Instances can both be used to launch Amazon EC2 instances onto physical servers that are dedicated for your use.

- There are no performance, security, or physical differences between Dedicated Instances and instances on Dedicated Hosts. **However, Dedicated Hosts give you additional visibility and control over how instances are placed on a physical server**.

70. When you use Dedicated Hosts, you have control over **instance placement on the host using the Host Affinity and Instance Auto-placement settings**.
71. With Dedicated Instances, you don't have control over which host your instance launches and runs on (Underline hardware changes: e.g. Simulator Keys of www.HadoopExam.com , then you cannot use dedicated instances, whose hardware does changes). If your organization wants to use AWS, but has an existing software license with hardware compliance requirements, this allows visibility into the host's hardware so you can meet those requirements.
72. Dedicated Host Reservations provide a billing discount compared to running On-Demand Dedicated Hosts. Reservations are available in three payment options:
    - No Upfront—No Upfront Reservations provide you with a discount on your Dedicated Host usage over a term and do not require an upfront payment. Available for a one-year term only.
    - Partial Upfront—A portion of the reservation must be paid upfront and the remaining hours in the term are billed at a discounted rate. Available in one-year and three-year terms.
    - All Upfront (Highest cost saving)—Provides the lowest effective price. Available in one-year and three-year terms and covers the entire cost of the term upfront, with no additional charges going forward.
73. To use a Dedicated Host, you first allocate hosts for use in your account. You then launch instances onto the hosts by specifying host tenancy for the instance. The instance auto-placement setting allows you to control whether an instance can launch onto a particular host. When an instance is stopped and restarted, **the Host affinity setting** determines whether it's restarted on the same, or a different, host.

Monitoring

Automated Monitoring:

1. There are two types of status to check the health of your instance. System Status and Instance Status checks. (Learn with this video : https://www.youtube.com/watch?v=YPf7wSEq9d0)
2. **System Status Checks** - These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue or you can resolve it yourself (for example, by stopping and restarting or terminating and replacing an instance). Examples of problems that cause system status checks to fail include:
    a. Loss of network connectivity
    b. Loss of system power

   c. Software issues on the physical host
   d. Hardware issues on the physical host

3. **Instance Status Checks** - monitor the software and network configuration of your **individual instance**. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example by rebooting the instance or by making modifications in your operating system). Examples of problems that may cause instance status checks to fail include:
   a. Failed system status checks
   b. Misconfigured networking or startup configuration
   c. Exhausted memory
   d. Corrupted file system
   e. Incompatible kernel
4. **Amazon EC2 Monitoring Scripts (Best for custom monitoring)** - Perl scripts that can monitor **memory, disk, and swap** file usage in your instances.
5. **Amazon CloudWatch Logs** - monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, or other sources.

Manual monitoring:

6. Using Amazon EC2 Dashboard you can check:
   a. Service Health and Scheduled Events by region
   b. Instance state
   c. Status checks
   d. Alarm status
   e. Instance metric details (In the navigation pane click Instances, select an instance, and then click the Monitoring tab)
   f. Volume metric details (In the navigation pane click Volumes, select a volume, and then click the Monitoring tab)
7. CloudWatch is an AWS service that automatically collects a wide range of performance and health data about your AWS resources. This data is available through an API, and also can be viewed as graphs on the AWS console. However the graphs are located on the separate console pages for each type of resource (e.g. EC2, RDS, load balancer, etc)
8. Amazon CloudWatch Dashboard shows:
   a. Current alarms and status
   b. Graphs of alarms and resources
9. Service health status
   a. You can monitor the status of your instances by
      i. viewing status checks and

ii. scheduled events for your instances

b. You can also see status on specific events scheduled for your instances. Events provide information about upcoming activities such as rebooting or retirement that are planned for your instances, along with the scheduled start and end time of each event.

10. A **status check** gives you the information that results from automated checks performed by Amazon EC2.
    a. With **instance status monitoring**, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications.
    b. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues.
    c. AWS Cloudwatch monitoring : **CPU utilization, network traffic, and disk activity (not memory)**
    d. Status checks are performed every minute and each returns a pass or a fail status. If all checks pass, the overall status of the instance is OK. If one or more checks fail, the overall status is impaired.
    e. Status checks are built into Amazon EC2, so they cannot be disabled or deleted.
    f. You can, however create or delete alarms that are triggered based on the result of the status checks. For example, you can create an alarm to warn you if status checks fail on a specific instance.

10. Amazon EC2 supports the following types of scheduled events for your instances:
    a. Instance stop: The instance will be stopped. When you start it again, it's migrated to a new host computer. Applies only to instances backed by Amazon EBS.
    b. Instance retirement: The instance will be stopped or terminated.
    c. Reboot: Either the instance will be rebooted (**instance reboot**) or the host computer for the instance will be rebooted (**system reboot**).
    d. System maintenance: The instance might be temporarily affected by network maintenance or power maintenance.

11. When AWS detects irreparable failure of the underlying host computer for your instance, it schedules the instance to stop or terminate, depending on the type of root device for the instance. If the root device is an EBS volume, the instance is scheduled to stop. If the root device is an instance store volume, the instance is scheduled to terminate.
    a. **Actions for Instances Backed by Amazon EBS**: You can wait for the maintenance to occur as scheduled. Alternatively, you can stop and start the instance, which migrates it to a new host computer.
    b. **Actions for Instances Backed by Instance Store**: You can wait for the maintenance to occur as scheduled. Alternatively, if you want to maintain normal operation during a scheduled maintenance window, you can launch a replacement instance from your most

recent AMI, migrate all necessary data to the replacement instance before the scheduled maintenance window, and then terminate the original instance.

12. By default, Amazon EC2 sends metric data to CloudWatch in 5-minute periods. To send metric data for your instance to CloudWatch in 1-minute periods, you can enable detailed monitoring on the instance.
    a. **Basic**: Data is available automatically in 5-minute periods at no charge.
    b. **Detailed**: Data is available in 1-minute periods for an additional cost. To get this level of data, you must specifically enable it for the instance. For the instances where you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances.

13. Aggregate statistics are available for the instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates.
    a. **In addition, Amazon CloudWatch does not aggregate data across regions. Therefore, metrics are completely separate between regions**.
    b. Because no dimension is specified, CloudWatch returns statistics for all dimensions in the AWS/EC2 namespace.
    c. This technique for retrieving all dimensions across an AWS namespace does not work for custom namespaces that you publish to Amazon CloudWatch. With custom namespaces, you must specify the complete set of dimensions that are associated with any given data point to retrieve statistics that include the data point.
    d. You can aggregate statistics for the EC2 instances in an Auto Scaling group. **Note that Amazon CloudWatch cannot aggregate data across regions. Metrics are completely separate between regions.**
    e. After you launch an instance, you can open the Amazon EC2 console and view the monitoring graphs for an instance on the Monitoring tab. Each graph is based on one of the available Amazon EC2 metrics.
14. The following graphs are available:
    - Average CPU Utilization (Percent)
    - Average Disk Reads (Bytes)
    - Average Disk Writes (Bytes)
    - Maximum Network In (Bytes)
    - Maximum Network Out (Bytes)
    - Summary Disk Read Operations (Count)
    - Summary Disk Write Operations (Count)
    - Summary Status (Any)
    - Summary Status Instance (Count)

- Summary Status System (Count)

15. You can create a CloudWatch alarm that monitors CloudWatch metrics for one of your instances. CloudWatch will automatically send you a notification when the metric reaches a threshold you specify. You can create a CloudWatch alarm using the Amazon EC2 console, or using the more advanced options provided by the CloudWatch console.
16. You can use the stop or terminate actions to help you save money when you no longer need an instance to be running. You can use the reboot and recover actions to automatically reboot those instances or recover them onto new hardware if a system impairment occurs.
17. You can add the stop, terminate, reboot, or recover actions to any alarm that is set on an Amazon EC2 per-instance metric, including basic and detailed monitoring metrics provided by Amazon CloudWatch (in the AWS/EC2 namespace), as well as any custom metrics that include the "InstanceId=" dimension, as long as the InstanceId value refers to a valid running Amazon EC2 instance.
18. If you want to use an IAM role to stop, terminate, or reboot an instance using an alarm action, you can only use the **EC2ActionsAccess role**. Other IAM roles are not supported. If you are using another IAM role, you cannot stop, terminate, or reboot the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
19. If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.
20. You can create an alarm that stops an Amazon EC2 instance when a certain threshold has been met. For example, you may run development or test instances and occasionally forget to shut them off. You can create an alarm that is triggered when the average CPU utilization percentage has been lower than 10 percent for 24 hours, signaling that it is idle and no longer in use. You can adjust the threshold, duration, and period to suit your needs, plus you can add an Amazon Simple Notification Service (Amazon SNS) notification, so that you will receive an email when the alarm is triggered.
21. You can create an alarm that terminates an EC2 instance automatically when a certain threshold has been met **(as long as termination protection is not enabled for the instance**). For example, you might want to terminate an instance when it has completed its work, and you don't need the instance again. If you might want to use the instance later, you should stop the instance instead of terminating it.
22. You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically reboots the instance. The **reboot alarm action** is recommended for **Instance Health Check failures** (as opposed to the **recover alarm action**, which is suited for **System Health Check failures**). An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. **When you reboot an instance, it**

remains on the same physical host, so your instance keeps its public DNS name, private IP address, and any data on its instance store volumes.

23. Rebooting an instance doesn't start a new instance billing hour, unlike stopping and restarting your instance.
24. You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. **Terminated instances cannot be recovered**. **A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata**.
25. When the **StatusCheckFailed_System** alarm is triggered, and the recover action is initiated, you will be notified by the Amazon SNS topic that you chose when you created the alarm and associated the recover action. During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost. When the process is complete, information is published to the SNS topic you've configured for the alarm. Anyone who is subscribed to this SNS topic will receive an email notification that includes the status of the recovery attempt and any further instructions. You will notice an instance reboot on the recovered instance. If your instance has a public IP address, it retains the public IP address after recovery.
26. To avoid a race condition between the **reboot and recover actions**, we recommend that you set the alarm threshold to 2 for 1 minute when creating alarms that recover an Amazon EC2 instance.
27. You can view alarm and action history in the Amazon CloudWatch console. Amazon CloudWatch keeps the last two weeks' worth of alarm and action history.
28. **The MemoryUtilization metric is a custom metric (By default it is not available)**. In order to use the MemoryUtilization metric, you must install the Perl scripts for Linux instances.

Security:

1. If you access Amazon EC2 using the command line tools or an API, you'll need your access key ID and secret access key.
2. Instances can fail or terminate for reasons outside of your control. If an instance fails and you launch a replacement instance, the replacement has a different public IP address than the original.
3. However, if your application needs a static IP address, you can use an Elastic IP address.
4. You can use **security groups** to control who can access your instances. These are analogous to an inbound network firewall that enables you to specify the protocols, ports, and source IP ranges that are allowed to reach your instances.

5. You can create multiple security groups and assign different rules to each group. You can then assign each instance to one or more security groups, and we use the rules to determine which traffic is allowed to reach the instance. You can configure a security group so that only specific IP addresses or specific security groups have access to the instance.
6. Amazon EC2 stores the public key only, and you store the private key (ssh keys, to connect your EC2 instance).
7. Anyone who possesses your private key can decrypt your login information, so it's important that you store your private keys in a secure place.
8. You can have up **to five thousand key pairs** per region.
9. When you launch an instance, you should specify the name of the key pair you plan to use to connect to the instance. If you don't specify the name of an existing key pair when you launch an instance, **you won't be able to connect to the instance**.
10. **Amazon EC2 doesn't keep a copy of your private key; therefore, if you lose a private key, there is no way to recover it.**
11. If you lose the private key for an instance store-backed instance, you can't access the instance; you should terminate the instance and launch another instance using a new key pair.
12. **If you lose the private key for an EBS-backed Linux instance, you can regain access to your instance**.
13. If you have several users that require access to a single instance, you can add user accounts to your instance. (By sharing private keys among users)
14. You can create a key pair for each user, and add the public key information from each key pair to the .ssh/authorized_keys file for each user on your EC2 instance.
15. **You can then distribute the private key files to your users. That way, you do not have to distribute the same private key file that's used for the root account to multiple users**.
16. Amazon EC2 does not accept DSA keys. Make sure your key generator is set up to create RSA keys. **Supported lengths: 1024, 2048, and 4096**.
17. **The public key that you specified when you launched an instance is also available to you through its instance metadata**. To view the public key that you specified when launching the instance, use the following command from your instance:
18. GET http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
19. if you change the key pair that you use to connect to the instance, we don't update the instance metadata to show the new public key; you'll continue to see the public key for the key pair you specified when you launched the instance in the instance metadata.
20. **When you delete a key pair, you are only deleting Amazon EC2's copy of the public key. Deleting a key pair doesn't affect the private key on your computer or the public key on any instances already launched using that key pair. You can't launch a new instance using**

a deleted key pair, but you can continue to connect to any instances that you launched using a deleted key pair, as long as you still have the private key (.pem) file.

21. If you're using an Auto Scaling group (for example, in an Elastic Beanstalk environment), ensure that the **key pair you're deleting is not specified in your launch configuration**. Auto Scaling launches a replacement instance if it detects an unhealthy instance; however, the instance launch fails if the key pair cannot be found.
22. If you create a Linux AMI from an instance, and then use the AMI to launch a new instance in a different region or account, the new instance includes the public key from the original instance. This enables you to connect to the new instance using the same private key file as your original instance. You can remove this public key from your instance by removing its entry from the **.ssh/authorized_keys** file using a text editor of your choice.
23. If you lose the private key for an EBS-backed instance, you can regain access to your instance. You must stop the instance, detach its root volume and attach it to another instance as a data volume, modify the authorized_keys file, move the volume back to the original instance, and restart the instance.
24. This procedure isn't supported for instance store-backed instances. If the root device is an instance store volume, you must have the private key in order to connect to the instance.

Security Groups:

1. You can modify the rules for a security group at any time; **the new rules are automatically applied to all instances that are associated with the security group** (no need to re-start instances). When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.
2. If you have requirements that aren't met by security groups, **you can maintain your own firewall on any of your instances in addition to using security groups.**
3. You can't specify a security group that you created for a VPC when you launch an instance in EC2-Classic.
4. **After you launch an instance in EC2-Classic, you can't change its security groups**. However, you can add rules to or remove rules from a security group, and those changes are automatically applied to all instances that are associated with the security group.
5. If you're using EC2-VPC, **you must use security groups created specifically for your VPC**. When you launch an instance in a VPC, you must specify a security group for that VPC. You can't specify a security group that you created for EC2-Classic when you launch an instance in a VPC.
6. **After you launch an instance in a VPC, you can change its security groups**. Security groups are associated with network interfaces. Changing an instance's security group's changes the security groups associated with the primary network interface (eth0).
7. You can also change the security groups associated with any other network interface.

8. You can change the rules of a security group, and those changes are automatically applied to all instances that are associated with the security group.
9. **In EC2-VPC, you can associate a network interface with up to 5 security groups and add up to 50 rules to a security group**.
10. When you specify a security group for a non-default VPC to the CLI or the API actions, you must **use the security group ID and not the security group name to identify the security group**.
11. Security groups for EC2-VPC have additional capabilities that aren't supported by security groups for EC2-Classic.
12. **The rules of a security group control the inbound traffic that's allowed to reach the instances that are associated with the security group and the outbound traffic that's allowed to leave them.**
13. **By default, security groups allow all outbound traffic.**
14. **Security group rules are always permissive; you can't create rules that deny access.**
15. You can add and remove rules at any time. You can't change the outbound rules for EC2-Classic. If you're using the Amazon EC2 console, you can modify existing rules, and you can copy the rules from an existing security group to a new security group.
16. Security groups are state-ful — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. For VPC security groups, this also means that responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.
17. If your instance (host A) initiates traffic to host B and uses a protocol other than TCP, UDP, or ICMP, your instance's firewall only tracks the IP address and protocol number for the purpose of allowing response traffic from host B. If host B initiates traffic to your instance in a separate request within 600 seconds of the original request or response, your instance accepts it regardless of inbound security group rules, because it's regarded as response traffic. For VPC security groups, you can control this by modifying your security group's outbound rules to permit only certain types of outbound traffic. **Alternatively, you can use a network ACL for your subnet — network ACLs are stateless and therefore do not automatically allow response traffic**.
18. An individual IP address, in CIDR notation. Be sure to use the /32 prefix after the IP address; if you use the /0 prefix after the IP address, this opens the port to everyone. For example, specify the IP address 203.0.113.1 as 203.0.113.1/32.
19. When you specify a security group as the source or destination for a rule, the rule affects all instances associated with the security group. **Incoming traffic is allowed based on the private IP addresses of the instances that are associated with the source security group (and not the public IP or Elastic IP addresses).**

20. If your security group rule references a security group in a peer VPC, and the referenced security group or VPC peering connection is deleted, the rule is marked as stale.
21. **If there is more than one rule for a specific port, we apply the most permissive rule. For example, if you have a rule that allows access to TCP port 22 (SSH) from IP address 203.0.113.1 and another rule that allows access to TCP port 22 from everyone, everyone has access to TCP port 22.**
22. When you associate multiple security groups with an instance, the rules from each security group are effectively aggregated to create one set of rules. We use this set of rules to determine whether to allow access.
23. Because you can assign multiple security groups to an instance, an instance can have hundreds of rules that apply. This might cause problems when you access the instance. Therefore, we recommend that you condense your rules as much as possible.
24. Your security groups use connection tracking to track information about traffic to and from the instance. Rules are applied based on the connection state of the traffic to determine if the traffic is allowed or denied.
25. This allows security groups to be stateful — responses to inbound traffic are allowed to flow out of the instance regardless of outbound security group rules, and vice versa. For example, if you initiate an ICMP ping command to your instance from your home computer, and your inbound security group rules allow ICMP traffic, information about the connection (including the port information) is tracked. Response traffic from the instance for the ping command is not tracked as new request, but rather as an established connection and is allowed to flow out of the instance, even if your outbound security group rules restrict outbound ICMP traffic.
26. Not all flows of traffic are tracked. If a security group rule permits TCP or UDP flows for all traffic/ip (0.0.0.0/0) and there is a corresponding rule in the other direction that permits the response traffic, then that flow of traffic is not tracked. The response traffic is therefore allowed to flow based on the inbound or outbound rule that permits the response traffic, and not on tracking information.
27. An existing flow of traffic that is tracked may not be interrupted when you remove the security group rule that enables that flow. Instead, the flow is interrupted when it's stopped by you or the other host for at least a few minutes (or up to 5 days for established TCP connections). For UDP, this may require terminating actions on the remote side of the flow. An untracked flow of traffic is immediately interrupted if the rule that enables the flow is removed or modified. For example, if you remove a rule that allows all inbound SSH traffic (0.0.0.0/0) to the instance, then your existing SSH connections to the instance are immediately dropped.
28. If you want to ensure that traffic is immediately interrupted when you remove a security group rule, you can use a network ACL for your subnet — network ACLs are stateless and therefore do not automatically allow response traffic.

29. Your AWS account automatically has a default security group per region for EC2-Classic. When you create a VPC, we automatically create a default security group for the VPC. If you don't specify a different security group when you launch an instance, the instance is automatically associated with the appropriate default security group.
30. A default security group is named default, and it has an ID assigned by AWS. The following are the initial settings for each default security group:
    - Allow inbound traffic only from other instances associated with the default security group
    - Allow all outbound traffic from the instance
31. The default security group specifies itself as a source security group in its inbound rules. This is what allows instances associated with the default security group to communicate with other instances associated with the default security group.
32. You can change the rules for a default security group. For example, you can add an inbound rule to allow SSH connections so that specific hosts can manage the instance.
33. You can't delete a default security group.
34. **After you launch an instance in EC2-Classic, you can't change its security groups. After you launch an instance in a VPC, you can change its security groups**.
35. When you add a rule to a security group, the new rule is automatically applied to any instances associated with the security group.
36. When you delete a rule from a security group, the change is automatically applied to any instances associated with the security group.
37. You can't delete a security group that is associated with an instance. You can't delete the default security group. You can't delete a security group that is referenced by a rule in another security group. If your security group is referenced by one of its own rules, you must delete the rule before you can delete the security group.

AWS Resource Access

1. You can use IAM to control how other users use resources in your AWS account, and you can use security groups to control access to your Amazon EC2 instances.
2. IAM enables you to do the following:
    - Create users and groups under your AWS account
    - Assign unique security credentials to each user under your AWS account
    - Control each user's permissions to perform tasks using AWS resources
    - Allow the users in another AWS account to share your AWS resources
    - Create roles for your AWS account and define the users or services that can assume them
    - Use existing identities for your enterprise to grant permissions to perform tasks using AWS resources

3. By default, IAM users don't have permission to create or modify Amazon EC2 resources, or perform tasks using the Amazon EC2 API.
4. To allow IAM users to create or modify resources and perform tasks, you must create IAM policies that grant IAM users permission to use the specific resources and API actions they'll need, and then attach those policies to the IAM users or groups that require those permissions.
5. When you attach a policy to a user or group of users, **it allows or denies the users permission to perform the specified tasks on the specified resources**.
6. An IAM policy must **grant or deny** permission to use one or more Amazon EC2 actions. It must also specify the resources that can be used with the action, which can be all resources, or in some cases, specific resources. The policy can also include conditions that you apply to the resource.
7. Amazon EC2 partially supports resource-level permissions. This means that for some EC2 API actions, you cannot specify which resource a user is allowed to work with for that action; instead, you have to allow users to work with all resources for that action.
8. Each IAM policy statement applies to the resources that you specify using their ARNs.
9. An ARN has the following general syntax:
    a. arn:aws:[service]:[region]:[account]:resourceType/resourcePath
10. In a policy statement, you can optionally specify conditions that control when it is in effect. Each condition contains one or more key-value pairs. Condition keys are not case sensitive. We've defined AWS-wide condition keys, plus additional service-specific condition keys.
11. If you specify multiple conditions, or multiple keys in a single condition, we evaluate them using a logical AND operation. If you specify a single condition with multiple values for one key, we evaluate the condition using a logical OR operation. For permission to be granted, all conditions must be met.
12. Amazon EC2 implements the AWS-wide condition keys, plus the following service-specific condition keys.
13. Many condition keys are specific to a resource, and some API actions use multiple resources. If you write a policy with a condition key, use the Resource element of the statement to specify the resource to which the condition key applies. If not, the policy may prevent users from performing the action at all, because the condition check fails for the resources to which the condition key does not apply. If you do not want to specify a resource, or if you've written the Action element of your policy to include multiple API actions, then you must use the ...IfExists condition type to ensure that the condition key is ignored for resources that do not use it.
14. It can take several minutes for policy changes to propagate before they take effect. Therefore, we recommend that you allow five minutes to pass before you test your policy updates.

Select the Package or Products from the Below Combination and Get Great Discount

Customize Your Package : Following Products are available. You can customize your own package. Please select Products from below list and send an email to hadoopexam@gmail.com . So, our team will reply with discounted price.

Recorded Trainings (With Hands On Lab) 1. Hadoop BigData Professional Training (3500INR/$79) 2. HBase (NoSQL) Professional Training (3500INR/$79) 3. Apache Spark Professional Training (3900INR/$89 for a week 3500INR/$75) 4. Apache OOzie (Hadoop Workflow Engine) Training	5. Apache Spark Oreilly Developer Certification Hortonworks Certification : 6. HDPCD : No Java	EMC :Data Science Certification Practice Questions 7. EMC E20-007 8. EMC E20-065
Cloudera Certification Practice Question Bank 9. CCA50X : Hadoop Administrator 10. CCA-175 Cloudera® (Hadoop and Spark Developer) 11. CCP:DE575 : Cloudera® Data Engineer Certification 12. CCA159 : Cloudera® Data Analyst Certification	AWS : Amazon WebService Certification Practice Question 13. AWS Solution Architect : Associate 14. AWS Solution Architect: Professional 15. AWS Developer : Associate 16. AWS Sysops Admin : Associate	SAS Certffication Practice Questions 17. SAS Base A00-211 18. SAS Advanced A00-212 19. SAS Analytics : A00-240 20. SAS Administrator : A00-250
MapR Hadoop/BigData Certification Practice Questions 21. MapR Hadoop Developer Certification 22. MapR HBase NoSQL Certifcation 23. MapR Spark Developer Certification	Microsoft Azure Cloud Certification 24. Azure 70-532 25. Azure 70-533	Oracle Certfication Practice Questions 26. Java 1z0-808 27. 1z0-060 (Oracle 12c) 28. 1z0-061 (Oracle 12c)

DON'T FORGET TO SUBSCRIBE FOR UPDATE ON PRODUCTS

Subscribe

Request an Exam or Training

Cloudera Certification	Hortonworks Certification	Data Science	Cloud Computing	Analytics	SAS Certification	Java /J2EE Training	NoSQL	Deep/Machine Learning
Python Programming	Scala Programming	SQL for Finance	Azure Training	Cassandra NoSQL	Salesforce	Tableau	QlickView	Many More

HadoopExam Learning Resource provides the following material for the Advanced Technologies.
Please visit www.HadoopExam.com for more detail this is just a few products from portfolio.

Price start for training with Just $79/3500INR

Apache Spark Professional Training with HandsOn Session + Certification Material

Hadoop Professional Training with HandsOn Sessioon + Certification Material

HBase Professional Training with HandsOn Session + Certification Material

Certification Material

Certification Material

Certification Material

Certification Material

Certification Material

Certification Material

Certification Material