**Real-time Interview Crackers! Part # 7**

**AWS-related interview:**

Deeper understanding

the key aws services

and features!

**Questionnaire**

**1. What are the types of virtualization in AWS**

**2. What is the difference between t2.micro and t3.micro instance types?**

**3. What are the different types of EBS volume types and their use cases?**

**4. Why do we associate an IAM role with an EC2 instance during its creation?**

**5. What are the use cases of tags in AWS?**

**6. What is the advantage of attaching an IAM role to an EC2 instance instead of using access keys for users?**

**7. What are the security options available for controlling access to EC2 instances and other AWS resources?**

**8. What is the difference between EBS and Instance Store as root device types?**

**9. What is the t2/t3 unlimited feature?**

**10. How do you check the hypervisor type of an EC2 instance?**

**11. What are the different types of Auto Scaling in AWS?**

**12. What is the difference between AWS Lambda and EC2?**

**13. How does AWS CloudTrail help with security and auditing?**

**14. What is AWS Cost Explorer and how does it help in cost management?**

**15. What are AWS Security Groups and Network ACLs, and how do they differ?**

**16. What are VPC Peering and Transit Gateway, and when would you use them?**

**17. What is the difference between RDS and DynamoDB?**

**18. What is AWS Elastic Load Balancing (ELB), and what are its types?**

**19. What is the purpose of AWS IAM Policies, and how do they work?**

**20. How can you use AWS Config for compliance management?**

**In Detail:**

**1. What are the types of virtualization in AWS?**

**Answer:** AWS supports **two** types of virtualization for EC2 instances:

* **HVM (Hardware Virtual Machine)**: This type of virtualization uses hardware-assisted virtualization and provides better performance as it directly uses the host’s hardware.
* **PV (Para-Virtualization)**: This is a software-assisted virtualization where the guest operating system is aware of the hypervisor and runs with less overhead. It's used for **older** instances and provides **lower** performance compared to HVM.

**Scenario:**

* **Use Case for HVM**: For a **high**-performance application like a **database** or a **web** server with high traffic, HVM instances would provide better performance and scalability.
* **Use Case for PV**: PV instances could be used for **legacy** applications where the OS might not be compatible with hardware-assisted virtualization.

**2. What is the difference between t2.micro and t3.micro instance types?**

**Answer:**

* **t2.micro**: Part of the T2 instance family. It offers a baseline CPU performance with the ability to **burst** when needed.
* **t3.micro**: Part of the T3 instance family, which provides **better** CPU performance compared to T2 instances, with a higher baseline and **more** burst capability.

**Scenario:**

* **Use Case for t2.micro**: Suitable for lightweight applications like **testing** environments where occasional bursts are needed.
* **Use Case for t3.micro**: Ideal for **production** workloads that require slightly better performance than t2.micro, such as small web applications, dev/test environments, or small databases.

**3. What are the different types of EBS volume types and their use cases?**

**Answer:**

* **General Purpose SSD (gp3)**: Offers a balance between **price** and performance for most applications.
* **Provisioned IOPS SSD (io1)**: For workloads that require high performance with low latency, like large **databases**.
* **Magnetic (standard)**: Older generation storage offering lower performance, ideal for **archival** and infrequent access data.
* **Cold HDD (sc1)**: Best for large data sets with low access frequency, such as logs and **backups**.
* **Throughput Optimized HDD (st1)**: Ideal for frequently accessed, throughput-intensive workloads like **big data** and data **warehousing**.

**Scenario:**

* **Use Case for General Purpose SSD**: Perfect for a simple EC2 instance running a small **web** application where performance is balanced between cost and speed.
* **Use Case for Provisioned IOPS SSD**: Use this for an EC2 instance running a high-performance database such as MySQL or PostgreSQL, which requires consistent and low-latency I/O performance.
* **Use Case for Magnetic**: For low-cost storage of infrequently accessed data, such as logs or **backups**.
* **Use Case for Cold HDD**: Ideal for storing large amounts of data that **don’t** need to be accessed often, such as old **archives** or infrequently accessed files.
* **Use Case for Throughput Optimized HDD**: Best used for big data applications or log storage, where large amounts of throughput are needed, but latency is less critical.

**4. Why do we associate an IAM role with an EC2 instance during its creation?**

**Answer:** An IAM (Identity and Access Management) role is associated with an EC2 instance to allow the instance to perform actions on AWS resources on behalf of the user. For example, if the EC2 instance needs to upload files to an S3 bucket, the IAM role grants it the necessary permissions.

**Scenario:**

* **Use Case**: If your EC2 instance is running a backup script that needs to upload files to an S3 bucket, associating an IAM role with the necessary S3 permissions (e.g., s3:**PutObject**) eliminates the need to configure individual AWS credentials on the instance, improving security and manageability.

**5. What are the use cases of tags in AWS?**

**Answer:** Tags are metadata attached to AWS resources to help organize and manage them. Tags can help with resource **identification**, cost allocation, and automation.

**Scenario:**

* **Use Case for EC2 Instance Tags**: Suppose you have multiple EC2 instances running in a development environment, and you want to track resource usage. You can tag instances with information like "Environment: Development," "Application: Web Server," and "Owner: Dev Team." This helps with cost allocation, identifying underutilized resources, and reporting.

**6. What is the advantage of attaching an IAM role to an EC2 instance instead of using access keys for users?**

**Answer:** Attaching an IAM role to an EC2 instance removes the need to configure and manage access keys, which can be compromised or misused. It also avoids the complexities of rotating credentials for individual users and enhances security by ensuring that permissions are attached directly to the instance.

**Scenario:**

* **Use Case**: For a scenario where an EC2 instance needs to access multiple AWS resources, such as an S3 bucket and DynamoDB, attaching an IAM role provides seamless access without embedding sensitive access keys within the instance, which could lead to accidental exposure.

**7. What are the security options available for controlling access to EC2 instances and other AWS resources?**

**Answer:**

* **Security Groups**: Control inbound and outbound traffic to EC2 instances.
* **Network ACLs**: Provide stateless filtering at the **subnet** level.
* **IAM Roles and Policies**: Control access permissions for users and services.
* **Tags**: Tags can also help in organizing and **securing** resources by tracking ownership and **purpose**.

**Scenario:**

* **Use Case for Security Groups**: If you're hosting a web application on an EC2 instance, you would configure a security group that allows inbound traffic on port 80 (HTTP) and port 443 (HTTPS) but restricts all other traffic.
* **Use Case for IAM Policies**: When using an EC2 instance to interact with other AWS services (e.g., S3, DynamoDB), IAM policies can limit the permissions granted to the instance, ensuring that it only has the necessary access to resources.

**8. What is the difference between EBS and Instance Store as root device types?**

**Answer:**

* **EBS (Elastic Block Store)**: Persistent storage that can be attached to EC2 instances. Data **remains** intact even after the instance is stopped or terminated.
* **Instance Store**: Temporary storage directly attached to the **physical** host. Data is **lost** when the instance is **stopped** or terminated.

**Scenario:**

* **Use Case for EBS**: If you need to **retain** data after an instance is terminated or stopped (e.g., application data or database files), EBS is the best choice.
* **Use Case for Instance Store**: Ideal for **temporary** storage that doesn't need to persist after an instance is stopped, such as **cache** data or temporary files.

**9. What is the t2/t3 unlimited feature?**

**Answer:** T2 and T3 instances support the **Unlimited mode** feature, which allows the instance to **burst** beyond its baseline CPU performance for as long as needed. This mode is helpful when your application occasionally requires more CPU than the baseline can provide.

**Scenario:**

* **Use Case for t2/t3 Unlimited**: Suppose you're running a small application that **occasionally** experiences spikes in traffic (e.g., a marketing campaign or product launch). With T2 or T3 Unlimited, your instance can burst beyond its baseline performance, handling the additional load without requiring you to manually scale up.

**Real-time Interview Crackers! Part # 8**

**AWS-related interview:**

Deeper understanding

the key aws services

and features!

**10. How do you check the hypervisor type of an EC2 instance?**

**Answer:** To check the hypervisor type, navigate to the EC2 instance details page in the AWS Console and check the "Hypervisor" field. AWS uses two hypervisors:

* **Xen**: The default hypervisor for older EC2 instance types (e.g., t2).
* **Nitro**: The hypervisor used by newer instance types (e.g., t3, m5) for better performance and security.

**Scenario:**

* **Use Case for Xen Hypervisor**: Older EC2 instances with Xen may be suitable for applications that are not as resource-intensive or for legacy systems that haven't yet been upgraded.
* **Use Case for Nitro Hypervisor**: Newer applications that require high performance and security, like machine learning or high-performance computing, benefit from the Nitro hypervisor’s improved capabilities.

**11. What are the different types of Auto Scaling in AWS?**

**Answer:**

* **Dynamic Scaling**: Adjusts capacity based on traffic demands. It increases or decreases the number of EC2 instances automatically based on CloudWatch alarms.
* **Predictive Scaling**: Uses machine learning to predict future traffic and scale your environment accordingly before the traffic spikes.
* **Scheduled Scaling**: Scales the infrastructure at a scheduled time, allowing you to pre-define scaling actions.

**Scenario:**

* **Use Case for Dynamic Scaling**: In an e-commerce application, where user traffic fluctuates throughout the day, dynamic scaling ensures that resources scale up during peak hours and scale down during off-peak hours.
* **Use Case for Predictive Scaling**: If you expect a surge in traffic for an event (e.g., Black Friday sales), predictive scaling can proactively add resources based on historical trends.
* **Use Case for Scheduled Scaling**: If you're running a batch job that needs to run every evening, scheduled scaling can automatically increase EC2 instances to handle the batch process and then scale down afterward.

**12. What is the difference between AWS Lambda and EC2?**

**Answer:**

* **AWS Lambda**: A serverless compute service that automatically manages the infrastructure for you. You only pay for the compute time used, and there is no need to manage servers.
* **EC2 (Elastic Compute Cloud)**: A virtual server in the cloud that provides full control over the operating system and application stack. You pay for the instances based on the type, size, and runtime.

**Scenario:**

* **Use Case for AWS Lambda**: If you're running event-driven functions like processing files in S3 or handling real-time data streams from Kinesis or DynamoDB, Lambda is ideal as you don't need to manage servers.
* **Use Case for EC2**: If you need to run a web server, a database, or any long-running application where you require full control over the environment (e.g., installing custom software or complex configurations), EC2 instances are better.

**13. How does AWS CloudTrail help with security and auditing?**

**Answer:** AWS CloudTrail records API calls made on your AWS resources. It provides a history of AWS API calls for your account, which can be used for auditing, security analysis, and compliance verification.

**Scenario:**

* **Use Case for Security**: If you suspect that an unauthorized user has accessed your AWS account, CloudTrail logs allow you to investigate the actions taken by that user, including which resources they accessed, when, and from which IP address.
* **Use Case for Auditing**: If you're required to provide an audit trail for compliance (e.g., SOC 2 or GDPR), CloudTrail provides detailed logs that show who made each change, enhancing your ability to demonstrate compliance.

**14. What is AWS Cost Explorer and how does it help in cost management?**

**Answer:** AWS Cost Explorer helps you visualize and analyze your AWS spending and usage patterns over time. You can set budgets, track usage across different accounts, services, and resources, and gain insights into cost-saving opportunities.

**Scenario:**

* **Use Case for Cost Optimization**: You can use AWS Cost Explorer to analyze your usage of EC2 instances and identify underutilized instances, which can be downsized or terminated to save costs.
* **Use Case for Budget Tracking**: If you're working on a project with a defined budget, you can create custom cost and usage reports in Cost Explorer to ensure you stay within budget by tracking the actual spend versus projections.

**15. What are AWS Security Groups and Network ACLs, and how do they differ?**

**Answer:**

* **Security Groups**: They act as a virtual firewall for EC2 instances to control incoming and outgoing traffic. Security groups are stateful, meaning that if you allow inbound traffic, the response traffic is automatically allowed, regardless of outbound rules.
* **Network ACLs (Access Control Lists)**: Provide an additional layer of security at the subnet level. They are stateless, meaning inbound and outbound traffic must be explicitly defined in both directions.

**Scenario:**

* **Use Case for Security Groups**: You can configure a security group for an EC2 instance to allow HTTP and HTTPS traffic while blocking SSH access from unauthorized IP addresses.
* **Use Case for Network ACLs**: You can set up a network ACL to block all incoming traffic from a specific IP range to protect against malicious attacks at the subnet level, regardless of individual instance settings.

**16. What are VPC Peering and Transit Gateway, and when would you use them?**

**Answer:**

* **VPC Peering**: A network connection between two VPCs that allows traffic to flow between them using private IPs. It’s suitable for direct communication between VPCs, but it becomes difficult to manage if you have multiple VPCs.
* **Transit Gateway**: A hub-and-spoke model for interconnecting multiple VPCs and on-premises networks. It simplifies network management and scales better than VPC Peering in large environments.

**Scenario:**

* **Use Case for VPC Peering**: You can use VPC Peering for a simple, low-cost connection between two VPCs in the same region where there’s no need for centralized management.
* **Use Case for Transit Gateway**: In a complex, multi-VPC environment with several accounts, using a Transit Gateway would allow you to manage inter-VPC communication centrally and reduce the complexity of managing many peering connections.

**17. What is the difference between RDS and DynamoDB?**

**Answer:**

* **RDS (Relational Database Service)**: A managed relational database service that supports SQL-based databases such as MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server. It’s ideal for structured data and complex queries.
* **DynamoDB**: A fully managed NoSQL database service designed for high-performance and scalability with key-value and document data models. It’s ideal for unstructured data or applications requiring high throughput with low latency.

**Scenario:**

* **Use Case for RDS**: Use RDS for applications requiring complex queries, joins, and transactional consistency, like an e-commerce backend or a financial application.
* **Use Case for DynamoDB**: Ideal for high-throughput applications with simple query patterns, like session storage, IoT data collection, or gaming leaderboards.

**18. What is AWS Elastic Load Balancing (ELB), and what are its types?**

**Answer:** AWS Elastic Load Balancing automatically distributes incoming application traffic across multiple targets (e.g., EC2 instances) to ensure high availability and fault tolerance. The types of ELB are:

* **Application Load Balancer (ALB)**: Best suited for HTTP/HTTPS traffic with advanced routing capabilities (e.g., path-based or host-based routing).
* **Network Load Balancer (NLB)**: Handles TCP traffic at high throughput with low latency, ideal for real-time applications.
* **Classic Load Balancer**: The original version of the ELB, supporting both HTTP and TCP traffic but with fewer features compared to ALB and NLB.

**Scenario:**

* **Use Case for ALB**: For a microservices-based application where you need to route traffic to different services based on the URL path (e.g., /api for the backend service and /frontend for the web service).
* **Use Case for NLB**: For applications requiring high-speed, low-latency TCP connections, such as gaming servers or real-time communication systems.
* **Use Case for Classic Load Balancer**: Suitable for legacy applications that do not require the advanced features provided by ALB or NLB.

**19. What is the purpose of AWS IAM Policies, and how do they work?**

**Answer:** IAM Policies are JSON documents that define permissions for users, groups, or roles. They specify what actions are allowed or denied on specific AWS resources.

**Scenario:**

* **Use Case**: If you want to allow a user to only read from an S3 bucket but not modify the contents, you would attach a policy that grants s3:GetObject permission while denying s3:PutObject.

**20. How can you use AWS Config for compliance management?**

**Answer:** AWS Config provides a detailed inventory of your AWS resources and their configuration history, allowing you to track configuration changes over time. It can evaluate your resources against compliance rules to ensure they conform to your organizational policies.

**Scenario:**

* **Use Case**: If your organization requires that all EC2 instances be encrypted, you can use AWS Config to monitor the compliance of instances, alerting you if any instance is found without encryption enabled.

These will provide a well-rounded set of AWS-related interviews or deepen their understanding of key AWS services and features.