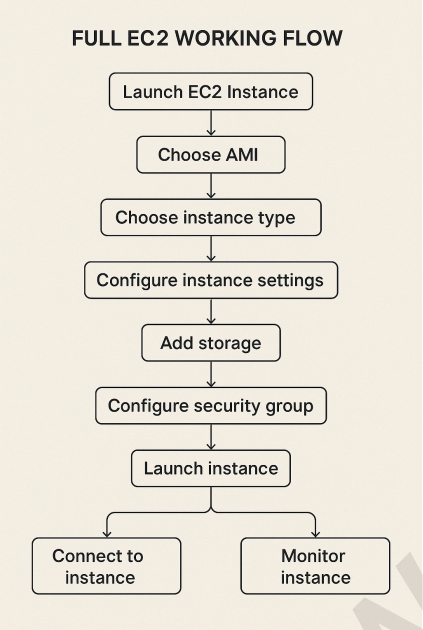
AWS EC2 Short Notes



1. What is EC2?

- **EC2 = Elastic Compute Cloud.**
- **It gives virtual computers (called instances) that you can use anytime.**
- **You can run websites, apps, software, etc. on these instances.**
- **EC2 is like renting a computer in the cloud.**

---

2. Main Features of EC2

- **Elastic: Easily increase or decrease computers as needed.**
- **Pay as you go: Pay only for the time you use.**
- **Secure: You control who can access it.**
- **Flexible: Choose different types of computers (CPU, RAM, storage).**
- **Global: Available in many parts of the world (called Regions).**

3. Key Terms

Term	Meaning
Instance	**A virtual server in EC2.**
AMI	**Amazon Machine Image (a template to launch instances).**
Instance Type	**Hardware configuration (like how much CPU, RAM, etc.).**
Key Pair	**Password for logging into instance.**
Security Group	**Like a firewall to allow or block traffic.**
Elastic IP	**A static (fixed) public IP address.**
Volume	**Storage (like hard drive) for your instance.**
EBS	**Elastic Block Store, gives storage to EC2.**
User Data	**Script you can run automatically when an instance starts.**

4. How EC2 Works (Simple Steps)

1. **Choose an AMI (template) → e.g., Ubuntu, Windows.**
2. **Choose Instance Type → e.g., t2.micro (free tier), m5.large, etc.**
3. **Configure Instance Settings → e.g., number of instances, network settings.**
4. **Add Storage → Attach disk space (EBS Volume).**
5. **Add Tags → Labels for easy identification (e.g., Name = WebServer).**
6. **Configure Security Group → Allow ports like SSH (22), HTTP (80).**
7. **Launch Instance → Start your virtual server.**
8. **Connect to Instance → Using SSH or RDP.**

---

5. Types of EC2 Instances

Family	Use Case	Example
General Purpose	Balanced CPU, Memory	t2, t3, m5
Compute Optimized	High CPU	c5, c6g
Memory Optimized	High RAM	r5, x1e
Storage Optimized	High storage	i3, d2
Accelerated Computing	GPU based (machine learning)	p4, g4ad

---

6. EC2 Pricing Options

Option	Meaning
On-Demand	Pay hourly/secondly. No long commitment.
Reserved Instances	1-year or 3-year commitment. Big discount.
Spot Instances	Buy unused capacity at cheap price.
Savings Plan	Flexible discounts without locking to specific instance.
Dedicated Hosts	Rent physical server, full control (good for license-heavy apps).

---

7. Storage in EC2

- **EBS Volume: Block storage for instance (like C drive/D drive).**
    - **Types: gp2 (General Purpose SSD), io1 (High IOPS), st1 (HDD), etc.**
- **Instance Store: Temporary storage (data lost if instance stops).**
- **EFS (Elastic File System): Shared storage (like Google Drive but for servers).**

---

8. Security in EC2

- **Key Pair: Private key (saved by you) and public key (on instance) for SSH login.**
- **Security Group: Set firewall rules.**
    - **Example: Allow SSH (port 22) from only your IP.**

- **IAM Role: Give permissions to instance to access AWS services (e.g., S3).**

---

9. Networking in EC2

- **VPC (Virtual Private Cloud): Private network for EC2.**
- **Subnet: A part inside VPC.**
- **Internet Gateway: Allows internet access.**
- **Elastic IP: Fixed IP to your instance.**
- **Private IP: Only accessible inside VPC.**

---

10. Load Balancing and Auto Scaling

- **Elastic Load Balancer (ELB): Distributes traffic to many instances.**
- **Auto Scaling Group (ASG): Automatically add/remove instances based on load.**
    - **Example: Add 2 more instances if CPU usage > 80%.**

---

11. Monitoring and Logging

- **CloudWatch:**
    - **Monitors instance performance (CPU, Disk, Network).**
    - **Set alarms (e.g., send alert if CPU > 90%).**
- **CloudTrail:**
    - **Records all API activities (who did what).**

---

12. EC2 Lifecycle

State	Meaning
Pending	Starting up.
Running	Running normally.
Stopping	Shutting down.
Stopped	Fully shut down, can restart later.
Terminated	Permanently deleted. Cannot recover.

---

13. EC2 Best Practices

- Always create a Key Pair and keep it safe.
- Use Security Groups properly (don't open all ports).
- Choose instance type according to workload.
- Use IAM Roles for permissions (don't hardcode access keys).
- Always take Snapshots (backup) of EBS volumes.
- Use Auto Scaling for handling traffic automatically.
- Set up monitoring and alerts in CloudWatch.

---

14. Useful EC2 Commands (Linux CLI)

Command	Purpose
`sudo yum update`	Update packages (Amazon Linux).
`sudo apt-get update`	Update packages (Ubuntu).
`ssh -i key.pem ec2-user@IP`	Connect to EC2 instance.
`df -h`	Check disk usage.
`top`	Check CPU usage.
`sudo reboot`	Restart server.

Final Tip:

👉 **EC2 is the heart of AWS.**
👉 **Master it first, then learn Auto Scaling, Load Balancer, VPC, etc.**
👉 **Always think about security, cost control, and backup.**

EC2 Interview Question and Answers

Q1. What is EC2 in AWS?

EC2 stands for **Elastic Compute Cloud**. It is a service provided by AWS that lets you run **virtual servers** (called instances) in the cloud. Instead of buying physical servers, you can launch EC2 instances anytime and pay only for what you use. You can choose the size (like

small or large), operating system (Linux or Windows), and control the server fully—just like a computer in your office, but on the cloud.

Q2. What are different types of EC2 instance?

There are many types (or families) of EC2 instances based on different needs:

- **T family (General Purpose)** – Good for low-cost, basic use (T2, T3, T4g).
- **M family (General Purpose)** – Balanced compute, memory, and networking.
- **C family (Compute Optimized)** – For high CPU tasks like gaming or batch processing.
- **R family (Memory Optimized)** – For apps that need a lot of RAM, like databases.
- **I family (Storage Optimized)** – For high-speed storage apps.
- **P & G family (GPU Instances)** – For machine learning or graphics.
- **D/H family (Dense Storage)** – For data warehousing or heavy storage needs.

Q3. What is AMI?

AMI stands for **Amazon Machine Image**. It is like a blueprint or template of an EC2 instance. An AMI contains the operating system, software, and settings. When you launch an EC2 instance, you choose an AMI. You can also create your own AMI from a running instance, so you can launch more instances with the exact same setup.

Q4. What are instance purchasing options in EC2?

AWS offers 4 main ways to buy EC2:

1. **On-Demand** – Pay per hour or second. No commitment. Good for short-term use.
2. **Reserved Instances** – Commit for 1 or 3 years and get up to 75% discount.
3. **Spot Instances** – Very cheap (up to 90% off), but AWS can stop it anytime.
4. **Savings Plans** – Like reserved instances but more flexible.

Q5. Difference between Stop, Terminate & Hibernate.

- **Stop** – Instance is shut down, but EBS volume (storage) remains. You can start it later.
- **Terminate** – Instance is deleted permanently. All data on instance storage is lost.
- **Hibernate** – Instance is paused, and RAM content is saved. When restarted, it resumes where it left off.

Q6. What is default EC2 storage - explain EBS, EFS, Instance Store, FSx.

- **EBS (Elastic Block Store)** – Like a hard disk for EC2. It stores data even if EC2 is stopped.
- **Instance Store** – Temporary storage. Data is lost if instance stops or crashes.
- **EFS (Elastic File System)** – Like a shared folder. Many EC2s can access it at once.
- **FSx** – Managed Windows file system or high-performance file system for special needs.

Q7. Explain Security Group.

A Security Group is like a **virtual firewall** for EC2. It controls **which traffic is allowed** to enter (inbound) or leave (outbound) your EC2. You can allow or block traffic based on IP address, port number, or protocol. By default, everything is blocked except what you allow.

Q8. What is a key pair & explain its types.

A key pair is used for **secure login** to an EC2 instance using SSH (Linux) or RDP (Windows). It has two parts:

- **Public Key** – Stored in EC2 by AWS.
- **Private Key** – Downloaded by you. Needed to login. Types:

- **RSA (default)** – Older standard, widely used.
- **ED25519** – Newer, more secure and faster.

Q9. How do you connect EC2 to system?

To connect a Linux EC2:

1. Use **SSH client** on your system (like terminal or PuTTY).
2. Use your **private key (.pem file)**.
3. Run command:
   ```
   ssh -i "key.pem" ec2-user@<public-ip>
   ```
   For Windows EC2:
4. Use **Remote Desktop (RDP)**.
5. Decrypt the admin password using the key pair.
6. Login with username and password.

Q10. What is difference between Private, Public & Elastic IP?

- **Private IP** – Used for communication inside AWS (within VPC). Not reachable from internet.
- **Public IP** – Given automatically by AWS. Changes if instance is stopped and started.
- **Elastic IP** – Static public IP. Doesn't change even if EC2 is stopped or restarted. Good for servers that need a permanent IP.

Q11. What is Placement Group & Tenancy?

- **Placement Group**: A way to place EC2 instances for better performance.
    - **Cluster** – All instances close together for high speed.

    - **Spread** – Instances on separate hardware for high availability.
    - **Partition** – Like spread, but with groups.
- **Tenancy**: Where EC2 runs.
    - **Shared** – On shared hardware (default).
    - **Dedicated** – On separate physical hardware (for compliance or security).

Q12. How do you monitor EC2 instance?

You can monitor EC2 using:

- **CloudWatch** – For CPU, memory, disk, and network usage.
- **CloudWatch Alarms** – To get alerts if something is wrong.
- **Status Checks** – AWS checks for hardware or software issues.
- **CloudTrail** – For tracking API calls and user actions.

Q13. Explain all types of instances - Spot, Reserved, On-Demand.

- **On-Demand**: Pay for what you use. Best for short-term or testing.
- **Reserved**: Book in advance (1 or 3 years). Cheaper for long-term use.
- **Spot**: Very cheap. Use extra AWS capacity. Can be stopped anytime.

Q14. How can you automate creation of EC2?

You can automate EC2 launch using:

- **Launch Templates** – Predefined settings for EC2.
- **Auto Scaling** – Automatically creates or removes instances.
- **CloudFormation** – Infrastructure as code.

- **Terraform** – Another tool to automate resources.
- **CI/CD tools** – Like GitHub Actions or Jenkins.

Q15. Explain what happens in backend when EC2 instance Reboot, Terminate, Start, Hibernate.

- **Reboot** – OS restarts, but instance stays on same hardware. No data loss.
- **Start** – EC2 boots up. If previously stopped, it may get a new public IP.
- **Stop** – Shuts down instance. Data on EBS is safe, but public IP is lost.
- **Terminate** – Deletes instance and attached instance store. EBS also deleted if delete option is selected.
- **Hibernate** – RAM content is saved to EBS. When started, it resumes from the same state.

Q16. How would you implement high availability using EC2?

To make EC2 highly available, you need to make sure that if one instance or zone fails, your application keeps running. Here's how:

- **Use Auto Scaling Group (ASG) with EC2 instances in multiple Availability Zones (AZs).**
- **Add a Load Balancer (ELB) to distribute traffic across healthy instances.**
- **Use CloudWatch to monitor and trigger new instances if something fails.**
- **Store data in EBS, EFS, or RDS, which are also highly available. This setup makes sure your website or app is always up and running.**

Q17. Difference between EC2 Instance and Lambda.

- **EC2 is a virtual server. You manage everything: OS, updates, and scaling.**

- **Lambda is serverless. You only write code, and AWS runs it when needed. Key differences:**
- **EC2 is always running unless you stop it; Lambda runs only when triggered.**
- **EC2 charges per time; Lambda charges per request and execution time.**
- **EC2 is good for long-running apps; Lambda is good for small, fast tasks.**

Q18. Role of EC2 in 3-tier architecture

In 3-tier architecture:

1. **Presentation Layer (Front-end) – EC2 can run web servers (like Apache/NGINX).**
2. **Application Layer – EC2 runs the main app logic (backend).**
3. **Database Layer – EC2 can host the database or connect to RDS. Each tier is usually on separate EC2 instances, making the app organized and scalable.**

Q19. How do you secure EC2 instance?

You can secure EC2 by:

- **Using Security Groups to allow only required ports (e.g., port 22 or 80).**
- **Using Key Pair to allow only trusted users to log in.**
- **Disabling root access and using non-root users.**
- **Regularly updating OS and applications.**
- **Using IAM roles instead of access keys.**
- **Enabling CloudWatch Logs, CloudTrail, and GuardDuty for monitoring.**
- **Encrypting EBS volumes and backups.**

Q20. How do you resize an EC2 instance?

To resize (change type or size):

1. **Stop the instance.**
2. **Go to Actions → Instance Settings → Change Instance Type.**
3. **Choose a new instance type (e.g., from t2.micro to t3.medium).**
4. **Start the instance again. Make sure the new instance type is supported in your availability zone.**

Q21. How do you schedule EC2 start & stop automatically?

Use AWS Lambda + CloudWatch Events (Scheduler):

1. **Create a Lambda function that starts/stops EC2.**
2. **Use CloudWatch rule to trigger it at specific times (e.g., 9 AM start, 6 PM stop). Or use Instance Scheduler – a solution provided by AWS to handle start/stop based on tags.**

Q22. What is Launch Template & Launch Configuration?

Both are used with Auto Scaling, but:

- **Launch Configuration is older. You can't edit it once created.**
- **Launch Template is newer and better. You can version it, support newer features like T3 instances, and use it in EC2 and ASG. Launch Templates are recommended for new setups.**

Q23. How do you migrate on-premises to AWS?

Steps:

1. **Assess current infrastructure.**

2. **Use AWS Migration Tools like:**
    - **Server Migration Service (SMS) for VMs.**
    - **Database Migration Service (DMS) for databases.**
3. **Create VPC, subnets, and security setup on AWS.**
4. **Transfer data using AWS Snowball (large) or AWS Transfer (small).**
5. **Test and then switch production to AWS.**

Q24. Difference between User Data & Metadata

- **User Data: Script or commands you give to EC2 when it starts (e.g., install Apache). Runs only once on first boot.**
- **Metadata: Information about the instance (e.g., instance ID, IP, IAM role) available from inside the instance using special URL `http://169.254.169.254`.**

Q25. How do you implement Blue-Green deployment using EC2?

Blue-Green means running two environments:

- **Blue = current live version.**
- **Green = new version. Steps:**

1. **Launch Green EC2 with updated app.**
2. **Test Green.**
3. **If okay, switch Load Balancer to Green.**
4. **Terminate Blue after final check. This reduces downtime and risk.**

Q26. How do you host high-traffic website?

Use this setup:

- **EC2 instances in Auto Scaling Group (multiple AZs).**
- **Use Application Load Balancer (ALB) to distribute load.**
- **Store static files in S3.**
- **Use CloudFront for global caching.**
- **Database in RDS or Aurora (Multi-AZ).**
- **Monitor with CloudWatch.**
- **Enable Elastic IP if public server.**

Q27. Best practices for securing EC2.

- **Use only required ports in Security Groups.**
- **Keep EC2 and packages updated.**
- **Use IAM roles, not hardcoded keys.**
- **Enable CloudWatch Logs, GuardDuty, CloudTrail.**
- **Encrypt EBS, snapshots, and data in transit.**
- **Disable root login.**
- **Use Bastion Host or Session Manager for SSH.**

Q28. How do you handle instance failover?

If one instance fails:

- **Use Auto Scaling Group to replace it automatically.**
- **Load Balancer detects unhealthy instance and routes traffic to healthy ones.**
- **Store data in EBS or external services (RDS, S3) to avoid data loss.**

- **Use CloudWatch Alarms to notify you.**

Q29. How do you backup & restore EC2 instance?

- **Take EBS Snapshots regularly. It saves disk data.**
- **You can also create AMI of EC2 for full backup. To restore:**
- **Launch a new EC2 using snapshot or AMI.**
- **Attach snapshot volume if only storage is needed.**

Q30. Steps to troubleshoot non-responding EC2 instance.

1. **Check System and Instance Status Checks.**
2. **Use CloudWatch metrics – high CPU/memory?**
3. **Try SSH or RDP login.**
4. **Check Security Group, Network ACL, and route tables.**
5. **Review logs (e.g., `/var/log/messages`).**
6. **Stop and Start instance.**
7. **If all fails, create AMI, launch new EC2 and test.**

Q31. Can an instance have multiple Elastic IPs? If yes, then why?

Yes, an EC2 instance can have more than one Elastic IP, but only if:

- **The instance is in a VPC (not older EC2-Classic).**
- **The instance has more than one network interface (ENI).**

Why use multiple Elastic IPs?

- **If the server hosts multiple websites needing different IPs.**
- **For failover setup, where one IP is backup. But remember: AWS charges for unused or extra Elastic IPs.**

Q32. How do you encrypt EC2 instance?

You can't encrypt the EC2 itself, but you can encrypt its EBS volumes (disk):

1. **While creating a volume or EC2, select "Enable encryption".**
2. **AWS uses KMS keys to encrypt it.**
3. **If the EC2 is already running:**
    - **Take a snapshot of the volume.**
    - **Copy the snapshot with encryption.**
    - **Create a new volume from the encrypted snapshot.**
    - **Attach it to a new or same EC2.**

Q33. How do you reduce EC2 cost?

- **Use Spot Instances for temporary or flexible tasks.**
- **Use Reserved Instances for long-term usage.**
- **Auto Stop instances not in use (like dev/test).**
- **Use Auto Scaling to add/remove EC2 based on need.**
- **Use Smaller Instance Types or right-size using CloudWatch.**
- **Use Savings Plans or Compute Optimizer.**
- **Store data in S3, not on EC2 if not required.**

Q34. How do you choose right EC2 instance?

Ask:

- **How much CPU, RAM, storage, and network speed does your app need?**
- **Use instance types:**
    - **t3/t4 – general purpose**
    - **m5 – balanced**
    - **c5 – compute-heavy**
    - **r5 – RAM-heavy**
    - **g4/p4 – GPU tasks Check with AWS Instance Selector, or test with trial and monitor using CloudWatch.**

Q35. What tools do you use to monitor instance?

- **CloudWatch: for CPU, memory, disk, network, alarms.**
- **CloudTrail: logs all user/API activity.**
- **AWS Config: tracks resource changes.**
- **AWS Systems Manager: monitor, run commands, patch, automate.**
- **Third-party: Datadog, New Relic, Prometheus + Grafana.**

Q36. Can we attach multiple EBS volumes? If yes, then why?

Yes, you can attach multiple EBS volumes to one EC2.
Why?

- **To separate OS and data (cleaner backups).**
- **For faster I/O by using RAID (striping).**
- **For storing different data types (e.g., logs, app files).**

- For better organization and easier recovery.

Q37. How do you attach & mount EBS volume?

Steps:

1. Go to EC2 → Volumes → Attach Volume.
2. Select EC2 and device name (e.g., `/dev/xvdf`).
3. Connect to EC2 via SSH.
4. Format the volume:
   ```
   sudo mkfs -t ext4 /dev/xvdf
   ```
5. Create mount point:
   ```
   sudo mkdir /data
   ```
6. Mount it:
   ```
   sudo mount /dev/xvdf /data
   ```
7. Add to `/etc/fstab` for auto-mount on reboot.

Q38. How do you encrypt EBS volume?

1. While creating a new EBS, select "Encryption".
2. If the volume is already created:
   - Take a snapshot.
   - Copy it with encryption enabled.
   - Create a new volume from encrypted snapshot.
3. Attach encrypted volume to EC2. Encryption is done using AWS KMS keys, and all data is encrypted at rest.

Q39. What is EBS snapshot? How do you use it for backup & reduce cost?

- **A snapshot is a backup of your EBS volume.**
- **You can restore it anytime by creating a new volume. To reduce cost:**
- **Use incremental snapshots (AWS does this by default – only changes are saved).**
- **Use lifecycle policies to delete old snapshots.**
- **Store less-frequent backups to S3 Glacier for long-term, cheap storage.**

Q40. How do you use S3 with EC2 storage?

You can use S3 to:

- **Store static files, logs, backups from EC2.**
- **Mount S3 like a drive using s3fs-fuse or AWS DataSync.**
- **Backup EBS snapshots or app data. Example: Copy log files to S3 daily using a cron job or Lambda.**

Q41. How do you restore terminated instance?

If instance is terminated, you can't bring it back. But if you had:

- **AMI backup – create new EC2 from that AMI.**
- **EBS snapshot – create volume and attach to new EC2. Tip: Always create AMIs or snapshots before major changes to restore easily.**

Q42. What is Nitro System?

Nitro is the hardware + software system used in new EC2 instances:

- **Offers better security, faster performance, and lower cost.**

- **Gives full hardware isolation.**
- **Enables features like bare-metal instances, fast networking, and enhanced monitoring. New instance families (e.g., C5, M5) use Nitro.**

Q43. What are ENI and ENA in AWS?

- **ENI (Elastic Network Interface): A virtual network card. One EC2 can have multiple ENIs (for different IPs or subnets).**
- **ENA (Elastic Network Adapter): Used for high-performance networking, up to 100 Gbps. Mostly used in large-scale apps.**

Q44. How do you move EC2 from one region to another?

1. **Create an AMI of your EC2.**
2. **Copy the AMI to the target region.**
3. **In that region, launch EC2 from that AMI.**
4. **Move EBS snapshots if needed. Note: IPs and endpoints will change, so update DNS or config.**

Q45. How do you log in if key is lost?

If private key is lost, you can't SSH. But you can recover:

1. **Stop the instance.**
2. **Detach its root volume.**
3. **Attach to another EC2 as secondary volume.**
4. **Log in to the second EC2, edit `authorized_keys`, and add a new key.**
5. **Detach the volume and reattach it as root to the original instance.**
6. **Start the original EC2 and log in with the new key.**

Scenario-Based AWS EC2 Interview Questions

1. Scenario: You need to deploy a web application that must be highly available across different geographic locations. How would you architect the solution using AWS EC2?

- **Answer: In this scenario, the goal is to ensure high availability and low latency for users across different geographic locations. Here's how you can architect the solution using EC2:**

    1. **Use Multiple EC2 Instances Across Multiple Regions: Deploy EC2 instances in multiple AWS regions, ideally in regions that are geographically close to your user base. This will reduce latency and improve the availability of your application.**

    2. **Set Up Load Balancing: Use Elastic Load Balancing (ELB) to distribute incoming traffic across EC2 instances in different regions. ELB will ensure that traffic is routed to healthy instances and balance the load evenly across them.**

    3. **Use Auto Scaling: Set up Auto Scaling to automatically add or remove EC2 instances based on traffic demand. This ensures that you have enough capacity during traffic spikes and saves costs when demand is low.**

    4. **Use Amazon Route 53 for DNS Routing: Use Amazon Route 53, AWS's DNS service, to route users to the nearest region based on their geographic location. Route 53 supports latency-based routing to direct users to the EC2 instances with the lowest latency.**

    5. **Replication for Data Consistency: If your application relies on databases, consider using Amazon RDS with Multi-AZ deployments or Amazon DynamoDB Global Tables for cross-region data replication to ensure data consistency and availability.**

---

2. Scenario: You are hosting a critical application on EC2 that needs to be running 24/7. However, the cost of running EC2 instances on-demand is too high. How would you optimize costs without compromising availability?

- **Answer: To reduce the cost of running EC2 instances 24/7 while maintaining availability, you can do the following:**

1. **Use Reserved Instances: Purchase Reserved Instances for a 1 or 3-year term. Reserved Instances offer a significant discount (up to 75%) compared to On-Demand Instances, and you get the guarantee that your instances will be available when you need them.**

2. **Consider Spot Instances for Non-Critical Workloads: If you have certain tasks that are not time-sensitive or can be interrupted (e.g., batch processing), consider using Spot Instances. These are much cheaper than On-Demand Instances, and you can bid for unused EC2 capacity. However, Spot Instances can be terminated by AWS with little notice, so they are best for flexible workloads.**

3. **Optimize EC2 Instance Size: Regularly monitor the performance of your EC2 instances using CloudWatch and resize them to ensure you're not over-provisioning resources. For example, if your instance is underutilized, you can downgrade to a smaller instance to save costs.**

4. **Use Auto Scaling: Set up Auto Scaling so that you only pay for the EC2 instances you need. Auto Scaling will dynamically adjust the number of instances based on the load, ensuring that you only pay for capacity during peak times.**

---

3. Scenario: Your EC2 instance is running out of storage space, and you need to expand it. What are your options?

- **Answer: If your EC2 instance is running out of storage space, you can expand the storage in the following ways:**

    1. **Increase EBS Volume Size: If your EC2 instance is using Elastic Block Store (EBS) for storage, you can increase the size of the EBS volume without stopping the instance. You can do this through the AWS Management Console or the AWS CLI. Once the volume is expanded, you may need to extend the file system on the instance to utilize the new space.**

    2. **Add Additional EBS Volumes: If you're running out of space on a single EBS volume, you can add additional EBS volumes to your EC2 instance. After adding the new volumes, you can mount them to the instance and start using them for additional storage.**

    3. **Use Instance Store: If your instance is equipped with instance store (ephemeral storage), you can use it for temporary data storage. However, note that instance store data is lost when the instance is stopped or terminated.**

4. **Use Amazon S3 for Object Storage: For large amounts of unstructured data, consider moving some of your data to Amazon S3. S3 offers scalable object storage and is much cheaper than EBS for storing large volumes of data.**

---

4. Scenario: Your EC2 instance is running fine, but users report that the application is slow. How do you troubleshoot and resolve the performance issue?

- **Answer: When users report that an application on EC2 is slow, you should troubleshoot the issue systematically:**

    1. **Check Instance CPU Utilization: Use CloudWatch to monitor CPU utilization. If the CPU usage is high, it may indicate that your instance type is not powerful enough for your workload. Consider resizing the instance to a larger type or optimizing your application for better CPU efficiency.**

    2. **Monitor Memory and Disk I/O: Check the memory usage and disk I/O of your EC2 instance in CloudWatch. If your instance is running out of memory or experiencing high disk activity, you may need to either resize the instance or optimize your application's memory and disk usage.**

    3. **Check Network Traffic: High network traffic could be a reason for poor application performance. You can check the network metrics in CloudWatch to see if there's excessive inbound or outbound traffic that could be causing slowdowns.**

    4. **Analyze Application Logs: Check your application logs for any errors, bottlenecks, or issues. Sometimes the application itself might be the cause of the slowdown, not the EC2 instance.**

    5. **Scale Your EC2 Instance: If you are consistently hitting high usage levels, consider scaling vertically (choosing a larger instance type) or horizontally (adding more instances with Auto Scaling) to handle the load.**

    6. **Use Load Balancer and Auto Scaling: If your application is receiving more traffic than a single EC2 instance can handle, implement an Elastic Load Balancer (ELB) to distribute the load across multiple instances. Also, set up Auto Scaling to dynamically scale the number of instances as needed.**

---

5. Scenario: You need to make sure that your EC2 instance is always available, even in case of a hardware failure. What strategies would you use?

- **Answer: To ensure high availability for your EC2 instance even in the case of a hardware failure, you can use the following strategies:**

    1. **Use EC2 in Multiple Availability Zones (AZs): Launch EC2 instances in multiple Availability Zones within a region. Each Availability Zone is a physically isolated data center, so if one zone experiences a failure, your application can continue running from another zone.**

    2. **Set Up Auto Scaling: Use Auto Scaling to automatically replace unhealthy instances and scale your application in response to demand. This helps ensure that there is always the right amount of capacity, even if an instance becomes unavailable.**

    3. **Elastic Load Balancing (ELB): Use ELB to distribute traffic across instances in multiple AZs. If one EC2 instance fails, ELB will route the traffic to the remaining healthy instances, maintaining application availability.**

    4. **Backup with Amazon EBS Snapshots: Regularly take snapshots of your EBS volumes and store them in S3. In case of a failure, you can restore the data from the snapshot to a new EC2 instance.**

    5. **Use Amazon RDS with Multi-AZ Deployment: If your EC2 application depends on a database, consider using Amazon RDS with Multi-AZ deployments. This ensures that your database is replicated to a standby instance in another AZ, providing fault tolerance and high availability.**