1. **How do you implement rolling updates in ECS?**

What: Rolling updates let ECS **gradually replace old tasks with new ones** during deployment.
Why: To ensure **zero downtime** and **continuous availability** while updating your app.

🔄 **How It Works:**

- When you update the task definition or container image, ECS replaces tasks in **batches**.
- It keeps **minimum healthy tasks** running using deploymentConfiguration.

🔧 **Example:**

json
"deploymentConfiguration": {
"maximumPercent": 200,
"minimumHealthyPercent": 50
}
➡️ If desired tasks = 4, ECS keeps at least 2 old tasks running and can launch up to 8 tasks during the update.

---

2. How do you handle secrets and environment variables in ECS?
There are 3 main ways:
✅ **Basic Environment Variables**

- Define name and value in Task Definition JSON/YAML
- Not encrypted

✅ **AWS Secrets Manager**

- Store secrets securely
- Use in ECS like this:

json

"secrets": [
{
"name": "DB_PASSWORD",
"valueFrom": "arn:aws:secretsmanager:region:account-id:secret:secret-name"

```
 }
]
```
✅ **AWS Systems Manager (SSM) Parameter Store**

- Similar to Secrets Manager, used for config values

🟢 Best practice: Use **Secrets Manager** or **SSM** for any sensitive info (DB credentials, API keys).

---

3. How do you implement logging and monitoring for ECS tasks?

✅ **Logging with CloudWatch:**

- Configure each container to use awslogs driver:

json

```
"logConfiguration": {
 "logDriver": "awslogs",
 "options": {
  "awslogs-group": "/ecs/my-app",
  "awslogs-region": "ap-south-1",
  "awslogs-stream-prefix": "ecs"
 }
}
```

✅ **Monitoring:**

- **CloudWatch Metrics**: CPU, memory, task count, etc.
- **CloudWatch Alarms**: Trigger on high CPU, low health, etc.
- **Container Insights**: Deeper visibility (enabled in CloudWatch > ECS > Enable Insights)
- **X-Ray**: Optional tracing support for performance debugging

---

4. What are ECS Exec capabilities and how do you use them?

ECS Exec allows you to **open a shell inside a running container**, just like docker exec.

How to use:

1. Enable Exec in Task Definition and Service
2. Attach proper IAM policy (e.g., ssmmessages, ssm:StartSession)
3. Run command:

bash

```
aws ecs execute-command \
 --cluster my-cluster \
 --task my-task-id \
 --container my-container \
 --command "/bin/sh" \
 --interactive
```

🔐 **Secure**: Uses **SSM Session Manager** — no open SSH ports required.

---

5. How do you implement multi-container task definitions?

What: ECS lets you run **multiple containers in a single task** using a task definition.
Why: To group related containers that **must run together** on the same instance or Fargate task.

When: Use it when containers share resources or need **tight coupling** (e.g., log agent, proxy, sidecar).

🔧 **Common Use Cases:**

- App + **Nginx reverse proxy**
- App + **log shipper** (e.g., FluentBit)
- App + **monitoring agent**

🔧 **Key Configuration:**

- Each container has its own: name, image, portMappings, cpu, memory
- Use dependsOn, links, and **shared volumes** for communication/order
- For inter-container communication:
  - Use localhost with **bridge mode**
  - Use **private IPs** with awsvpc mode

---

6. What are the best practices for ECS security?

Area	Best Practice
IAM	Use least privilege; create per-task IAM roles
Secrets	Use AWS Secrets Manager or SSM Parameter Store
Networking	Use VPC, private subnets, NAT Gateway
Security Groups	Allow only needed ports/IPs
ECS Exec	Enable only for debugging, restrict with IAM
Image Security	Use scanned, signed images; avoid latest tag
OS/AMI Updates	Regularly patch EC2 AMIs (if using EC2 launch type)

---

7. How do you troubleshoot ECS service deployment issues?

✅ **Troubleshooting checklist:**

1. **Check Service Events** (ECS Console > Service):
   - Shows task failures, placement errors, or image pull errors.
2. **View Task Logs**:
   - CloudWatch logs (if enabled)
   - aws ecs describe-tasks → for recent failures
3. **Task Definitions**:
   - Validate image, port mapping, memory/CPU limits
4. **Networking Issues**:
   - Check subnet and security group configs
   - Verify awsvpc mode permissions (for Fargate)
5. **IAM Roles**:
   - Ensure correct task execution roles and permissions
6. **Target Group Health**:
   - Check ALB → Target Group → Health Checks
   - Update health check path/port if mismatched

🛠️ Use aws ecs describe-services, describe-tasks, and logs CLI for debugging.

1. What is the main purpose of rolling updates in ECS?
A. Update all tasks at once
B. Gradually replace old tasks with new ones to avoid downtime
C. Scale tasks based on load
D. Delete unused services

B. **Gradually replace old tasks with new ones to avoid downtime** ✅

---

2. What is the purpose of ECS Exec?
A. Restart ECS services remotely
B. SSH into EC2 hosts
C. Open a shell inside a running container securely
D. Execute IAM policies

C. **Open a shell inside a running container securely** ✅

---

3. When should you use multi-container ECS task definitions?
A. When running only one container
B. When related containers must run together (e.g., app + logging agent)
C. For database backups
D. To increase CPU

B. **When related containers must run together (e.g., app + logging agent)** ✅

---

4. Which of the following is a best practice for ECS security?
A. Use latest container tags
B. Use one IAM role for all services
C. Open all ports for ECS Exec
D. Use least-privilege IAM roles and private networking

D. **Use least-privilege IAM roles and private networking** ✅

---

5. If an ECS task fails to start, what should you check first?
A. VPC peering
B. Lambda logs
C. Service events, CloudWatch logs, and task definition parameters
D. EBS volume size

C. **Service events, CloudWatch logs, and task definition parameters** ✅