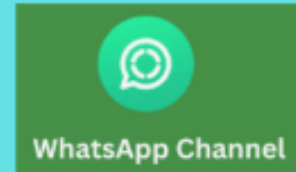


@devopschallengehub



## How do you implement automated vulnerability scanning in ECR?

**Answer:**

Amazon ECR provides **image scanning** using Amazon Inspector (v2).

To enable automated scanning:

- Enable "**scan on push**" for the repository.
  - This automatically triggers a vulnerability scan every time a new image is pushed.
  - You can also run **manual scans** using the AWS CLI:  
`aws ecr start-image-scan --repository-name <name> --image-id imageTag=<tag>`
  - Integrate scan results with **SNS notifications**, **EventBridge**, or **Security Hub** for alerting and automation.
- 

## What are the best practices for ECR repository naming conventions?

**Answer:**

- Use **lowercase, hyphen-separated** names (e.g., team-service-env).
  - Include relevant metadata like project, service, and environment (e.g., inventory-api-prod).
  - Avoid special characters or uppercase letters.
  - Versioning should be handled via **tags**, not in the repository name.
  - Maintain a naming convention doc for team-wide consistency.
- 

## How do you handle ECR repository cleanup and cost optimization?

**Answer:**

- Enable **lifecycle policies** to automatically delete untagged or old images (e.g., older than 30 days or keep only last 10).

- Use **aws ecr list-images** with filters to find and delete outdated images manually or via cron jobs.
- Use **image immutability** to prevent overwrites.
- Set up **CloudWatch metrics and billing alerts** to track storage usage and control costs.

---

## 16. How do you implement cross-account ECR access?

Answer:

- Use **resource-based policies** on the ECR repository to allow another AWS account to pull images:

json

-----

```
{
  "Effect": "Allow",
  "Principal": { "AWS": "arn:aws:iam::<account-id>:root" },
  "Action": [
    "ecr:BatchCheckLayerAvailability",
    "ecr:GetDownloadUrlForLayer",
    "ecr:BatchGetImage"
  ]
}
```

- Authenticate using **aws ecr get-login-password** in the target account and **Docker login**.
- Optionally, set up **AWS Resource Access Manager (RAM)** or **PrivateLink** for secure VPC-level access.

---

## 17. What are the networking considerations for ECR in VPC?

Answer:

- ECR is a **regional public service**, but you can access it privately using **VPC endpoints (interface type)**.
- Create a **VPC endpoint for ECR API and ECR DKR** (Docker registry).
- This keeps traffic within the AWS network and avoids NAT Gateway costs.
- Ensure **route tables and security groups** allow traffic to the endpoint.

---


## 18. How do you monitor ECR usage and performance?

Answer:

- Enable **CloudTrail** for auditing API calls to ECR (push, pull, delete).
  - Use **Amazon CloudWatch** metrics like:
    - StorageSize
    - ImageScanFindingsCount
  - Set up **CloudWatch Alarms** for large storage or failed scans.
  - Periodically review usage reports via **Cost Explorer** or **AWS Budgets**.
-


**What is a best practice for naming ECR repositories?**

- a. Use camelCase with environment tags inside the name
- b. Include version numbers in the repository name
- c. Use lowercase, hyphen-separated names with metadata like service and environment
- d. Always include special characters for easier sorting

 **Correct Answer: c.** Use lowercase, hyphen-separated names with metadata like service and environment

**How can you manage cleanup and reduce costs in Amazon ECR?**

- a. Enable image overwrite to reduce image count
- b. Manually delete repositories every month
- c. Enable lifecycle policies to remove untagged or old images
- d. Move images to S3 for cheaper storage

 **Correct Answer: c.** Enable lifecycle policies to remove untagged or old images