

Service Name

Amazon Elastic Container Registry (ECR)

Service Overview

Amazon Elastic Container Registry (ECR) is an AWS managed container image registry service that provides secure, scalable, and reliable storage for Docker images and Open Container Initiative (OCI) compatible artifacts. ECR supports both private repositories with resource-based permissions using AWS IAM for controlled access, and public repositories for worldwide distribution. Users can push, pull, and manage container images using preferred CLI tools, with deep integration into AWS container orchestration services like Amazon ECS, Amazon EKS, and AWS Fargate for streamlined development-to-production workflows.

Key Use Cases

- **Container image storage and distribution:** Centralized repository for storing and distributing Docker container images within organizations or for public consumption
- **Continuous integration and deployment (CI/CD):** Integrated automated building, testing, and deployment of containerized applications with various CI/CD tools
- **Microservices architecture:** Supporting containerized microservices deployments across AWS container orchestration platforms
- **Hybrid and multi-cloud deployments:** Container image management across different deployment environments
- **Globally distributed applications:** Cross-region replication for consistent global deployments with reduced latency
- **Security and compliance:** Image vulnerability scanning, access control, and managed signing for enhanced container security

Features

- **Lifecycle Policies:** Automated cleanup of unused images based on defined rules for image age or tag criteria
- **Image Scanning:** Vulnerability scanning for container images with scan-on-push capability using Clair CVE database
- **Cross-Region Replication:** Automatic replication of images across AWS regions for high availability and reduced latency
- **Pull Through Cache:** Cache repositories from upstream registries with periodic synchronization to ensure up-to-date images
- **Managed Signing:** Automatic cryptographic signature generation for pushed images using AWS Signer

- **Repository Creation Templates:** Define settings for automatically created repositories including encryption, policies, and tags
- **Encryption:** Image encryption at rest using AWS KMS and in-transit encryption for secure image storage
- **IAM Integration:** Fine-grained access control using AWS IAM roles and policies for repository and image access

Value Proposition

Amazon ECR eliminates the need to operate and manage container registries or underlying infrastructure, providing a fully managed solution with deep AWS integration. ECR offers enterprise-grade security with encryption at rest and in-transit, vulnerability scanning, and IAM-based access controls:

- The service provides high availability across multiple availability zones with automatic scaling and 99.9% uptime SLA
- ECR's native integration with Amazon ECS, EKS, and Fargate streamlines container workflows, while cross-region replication and pull-through caching optimize global deployments
- Cost-effective pricing with no upfront fees, pay-as-you-use model, and AWS Free Tier inclusion make it economical for organizations of all sizes

Service Integration

Amazon ECR integrates natively with core AWS container orchestration services including Amazon Elastic Container Service (ECS), Amazon Elastic Kubernetes Service (EKS), and AWS Fargate for seamless container deployment workflows. ECR works with AWS Lambda for serverless container functions and integrates with AWS CodePipeline, CodeBuild, and CodeDeploy for CI/CD automation:

- The service leverages AWS IAM for access management, AWS KMS for encryption, Amazon CloudWatch for monitoring and logging, and AWS CloudTrail for audit trails
- ECR also integrates with third-party tools like Jenkins, GitHub Actions, GitLab, and other popular CI/CD platforms through standard Docker Registry API compatibility

Onboarding and Offboarding

Getting started with Amazon ECR requires an AWS account and the latest AWS CLI version installed. Users begin by creating a repository using the AWS Management Console, CLI, or SDKs, then authenticate Docker CLI to ECR using the `get-authorization-token` command:

- The basic workflow involves building Docker images locally, tagging them with the ECR repository URI, and pushing images to ECR using standard Docker commands
- ECR provides comprehensive getting started guides with step-by-step tutorials for creating repositories, pushing first images, and integrating with ECS or EKS

- The service offers both console-based and programmatic access methods to accommodate different user preferences and automation requirements

Getting Started Guide:

<https://docs.aws.amazon.com/AmazonECR/latest/userguide/getting-started-cli.html>

Data Removal

Offboarding from Amazon ECR involves deleting container images and repositories to remove all stored data. Users can delete individual images using the console, CLI, or SDKs, which removes both the image and associated tags. Repository deletion requires the `--force` flag if images exist, permanently removing all contained images and metadata. Lifecycle policies can automate cleanup before offboarding. Cross-region replicated images must be deleted from each region separately. Once repositories are deleted, all associated data including images, tags, scan results, and policies are permanently removed from AWS infrastructure.

Backup/Restore and Disaster Recovery

Amazon ECR provides built-in backup and disaster recovery through cross-region replication, automatically copying images to secondary regions for redundancy. Images stored in ECR are replicated across multiple availability zones within a region for high availability. ECR supports native multi-region replication configured at the registry level, enabling automated disaster recovery scenarios. The service integrates with AWS Application Recovery Controller for automated failover orchestration. ECR's durability is enhanced by AWS S3-backed storage with 99.999999999% (11 9's) durability, providing enterprise-grade data protection without additional backup infrastructure.

Backup Capabilities

Amazon ECR includes built-in backup capabilities through cross-region replication that automatically copies container images to designated regions. While ECR doesn't directly integrate with AWS Backup service, its native replication provides automated backup functionality:

- Images are stored with high durability using AWS S3 infrastructure, ensuring data persistence
- Organizations can implement additional backup strategies using ECR APIs to programmatically copy images to multiple regions or accounts
- ECR's lifecycle policies can be configured to maintain image versions while cleaning up older copies for cost optimization

Disaster Recovery

Amazon ECR supports disaster recovery through cross-region and cross-account replication capabilities that can be configured at the registry level. While ECR doesn't

directly integrate with AWS Elastic Disaster Recovery service, it provides robust disaster recovery through automated image replication and high availability architecture:

- ECR repositories can be replicated to multiple regions simultaneously, enabling rapid failover scenarios
- The service works with AWS Application Recovery Controller to orchestrate complete application disaster recovery including container image availability
- ECR's multi-AZ architecture within regions provides automatic failover for high availability

Recovery Objectives

Amazon ECR helps customers achieve aggressive RPO and RTO objectives through real-time cross-region replication that continuously synchronizes images with near-zero data loss. ECR's multi-AZ architecture provides RTO of minutes within a region through automatic failover. Cross-region replication enables RPO of seconds to minutes depending on image size and network conditions. Integration with AWS Application Recovery Controller enables automated disaster recovery orchestration with RTO objectives measured in minutes. ECR's high availability design and global replication capabilities support enterprise-grade recovery requirements for containerized applications.

Service Constraints

Service Quotas: <https://docs.aws.amazon.com/AmazonECR/latest/userguide/service-quotas.html>

FAQ: <https://aws.amazon.com/ecr/faqs/>

Technical Requirements

Service Documentation:

<https://docs.aws.amazon.com/AmazonECR/latest/userguide/what-is-ecr.html>

User Guide: <https://docs.aws.amazon.com/AmazonECR/latest/userguide/>

API Reference: <https://docs.aws.amazon.com/AmazonECR/latest/APIReference/>

Pricing Overview

Please see the AWS UK G Cloud 15 Pricing Document accompanying this service in the Digital Marketplace.

Public Pricing: <https://aws.amazon.com/ecr/pricing/>

Cost Optimization

Amazon ECR offers several unique cost optimization strategies including lifecycle policies that automatically delete unused or old images based on configurable rules for age, count,

or tags. The service provides separate pricing for private and public repositories, with public repositories offering significant bandwidth allowances:

- ECR's pay-as-you-use model charges only for storage used and data transferred, with no upfront costs or minimum commitments
- Cross-region replication can be optimized by replicating only necessary images to reduce storage and transfer costs
- The AWS Free Tier includes 500 MB monthly storage for private repositories and 50 GB for public repositories, enabling cost-effective experimentation and small-scale deployments

Savings Considerations

Key cost savings considerations for Amazon ECR include implementing lifecycle policies to automatically remove unused images, preventing storage cost accumulation over time. Using public repositories for open-source or widely distributed images eliminates storage charges while leveraging ECR's global content delivery:

- Organizations can optimize data transfer costs by strategically placing repositories in regions closest to container workloads and using VPC endpoints to eliminate internet gateway charges
- Consolidating repositories and using image layer caching reduces storage through deduplication
- Regular auditing using ECR usage reports helps identify cost optimization opportunities, while leveraging AWS Free Tier allowances and volume discounts for larger deployments can significantly reduce overall container registry costs