



Panduan Pengguna

Amazon ECR



Versi API 2015-09-21

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon ECR: Panduan Pengguna

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan milik dari pemiliknya masing-masing, yang mungkin berafiliasi dengan, terhubung ke, atau disponsori oleh Amazon.

Table of Contents

Apa itu Amazon ECR	1
Komponen Amazon ECR	1
Fitur-Fitur Amazon ECR	2
Cara memulai dengan Amazon ECR	3
Harga untuk Amazon ECR	3
Menyiapkan	4
Mendaftar Akun AWS	4
Membuat pengguna administratif	5
Memulai	6
Menggunakan AWS CLI	8
Prasyarat	8
Instal AWS CLI	8
Instal Docker	8
Langkah 1: Buat citra Docker	10
Langkah 2: Autentikasi ke registrasi default	12
Langkah 3: Buat repositori	13
Langkah 4: Dorong citra ke Amazon ECR	13
Langkah 5: Menarik citra dari Amazon ECR	14
Langkah 6: Menghapus citra	15
Langkah 7: Menghapus repositori	15
Registri pribadi	17
Konsep registri	17
Otentikasi registri	17
Menggunakan Amazon ECR credential helper	18
Menggunakan token otorisasi	18
Menggunakan autentikasi HTTP API	19
Pengaturan registri	20
Izin registri	21
Menyetel pernyataan izin registri	21
Menghapus pernyataan izin registri	24
Contoh kebijakan registri	25
Repositori pribadi	28
Konsep repositori	28
Membuat repositori	29

Melihat detail repositori	30
Mengedit repositori	31
Menghapus repositori	32
Kebijakan repositori	33
Kebijakan repositori vs kebijakan IAM	33
Menetapkan pernyataan kebijakan repositori	35
Menghapus pernyataan kebijakan repositori	36
Contoh kebijakan repositori	37
Penandaan repositori	42
Dasar tanda	42
Penandaan sumber daya Anda	43
Batasan tanda	43
Penandaan sumber daya Anda untuk penagihan	44
Cara menggunakan tanda dengan menggunakan konsol	44
Cara menggunakan tanda dengan menggunakan AWS CLI atau API	45
Citra privat	48
Mendorong citra	48
Izin IAM yang diperlukan	49
Mendorong gambar Docker	50
Mendorong citra multi-arsitektur	52
Mendorong grafik Helm	54
Menandatangani gambar	56
Pertimbangan	56
Prasyarat	56
Konfigurasi otentikasi untuk klien Notaris	57
Menandatangani gambar	57
Verifikasi gambar secara lokal	58
Menghapus tanda tangan	60
Melihat detail citra	61
Menarik citra	61
Menggunakan aturan pull through cache	63
Pertimbangan untuk menggunakan pull through cache	64
Izin IAM yang diperlukan	66
Langkah selanjutnya	68
Membuat aturan pull through cache	68
Mengelola template pembuatan repositori	76

Bekerja dengan aturan pull through cache	92
Menghapus aturan pull through cache	95
Menyimpan kredensi repositori upstream Anda	96
Pemecahan masalah tarik melalui masalah cache	101
Menghapus citra	103
Menandai ulang citra	104
Replikasi citra	107
Pertimbangan untuk replikasi citra pribadi	107
Mengonfigurasi replikasi	108
Melihat status replikasi	113
Kebijakan siklus hidup	114
Cara kerja kebijakan siklus hidup	114
Templat kebijakan siklus hidup	117
Parameter kebijakan siklus hidup	117
Membuat pratinjau kebijakan siklus hidup	121
Membuat kebijakan siklus hidup	123
Contoh kebijakan siklus hidup	124
Tanda ketetapan citra	133
Pemindaian citra	135
Menggunakan filter	136
Pemindaian yang ditingkatkan	137
Pemindaian dasar	149
Memecahkan masalah pemindaian citra	156
Format manifes citra kontainer	157
Konversi manifes citra Amazon ECR	158
Menggunakan citra Amazon ECR dengan Amazon ECS	159
Izin IAM yang diperlukan	159
Menentukan gambar Amazon ECR dalam definisi tugas	161
Menggunakan Citra Amazon ECR dengan Amazon EKS	161
Menginstal grafik Helm yang di-host Amazon ECR dengan Amazon EKS	162
Citra kontainer Amazon Linux	164
Keamanan	166
Identity and Access Management	166
Audiens	167
Mengautentikasi dengan identitas	168
Mengelola akses menggunakan kebijakan	171

Bagaimana Amazon Elastic Container Registry bekerja dengan IAM	173
AWS kebijakan terkelola untuk Amazon ECR	179
Menggunakan peran terkait layanan	187
Pencegahan confused deputy lintas layanan	193
Contoh kebijakan berbasis identitas	195
Menggunakan Kontrol Akses Berbasis Tanda	199
Pemecahan Masalah	201
Perlindungan data	203
Enkripsi diam	204
Validasi kepatuhan	212
Keamanan Infrastruktur	213
VPC endpoint Antarmuka (AWS PrivateLink)	214
Memantau	224
Memvisualisasikan Kuota Layanan Anda dan Mengatur Alarm	225
Metrik Penggunaan	226
Laporan Penggunaan	227
Metrik repositori	228
Mengaktifkan metrik CloudWatch	228
Metrik dan dimensi yang tersedia	228
Melihat metrik Amazon ECR	229
Acara dan EventBridge	229
Contoh kejadian dari Amazon ECR	230
Mencatat Tindakan dengan AWS CloudTrail	234
Informasi Amazon ECR di CloudTrail	235
Memahami entri file log Amazon ECR	236
Kuota layanan	247
Mengelola kuota layanan Amazon ECR Anda di AWS Management Console	252
Membuat CloudWatch alarm untuk memantau metrik penggunaan API	253
Pemecahan Masalah	254
Mengaktifkan output debug Docker	254
Mengaktifkan AWS CloudTrail	254
Mengoptimalkan performa untuk Amazon ECR	255
Memecahkan masalah kesalahan dengan perintah Docker saat menggunakan Amazon ECR .	256
Kesalahan: "Verifikasi Sistem Berkas Gagal" atau "404: Citra Tidak Ditemukan" saat menarik citra dari repositori Amazon ECR	257

Kesalahan: "Verifikasi Lapisan Sistem Berkas Gagal" saat menarik citra dari Amazon ECR	258
Kesalahan HTTP 403 atau kesalahan "no basic auth credentials" ketika mendorong ke repositori	258
Memecahkan masalah pesan kesalahan Amazon ECR	259
HTTP 429: Terlalu Banyak Permintaan atau ThrottleException	259
HTTP 403: "Pengguna [arn] tidak memiliki otorisasi untuk melakukan [operasi]"	260
HTTP 404: kesalahan "Repositori Tidak Ada"	260
Kesalahan: Tidak dapat melakukan login interaktif dari perangkat non TTY	261
Riwayat dokumen	262
AWSGlosarium	267
.....	cclxviii

Apa Itu Amazon Elastic Container Registry?

Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri citra kontainer terkelola milik AWS yang aman, dapat diskalakan, dan andal. Amazon ECR mendukung repositori pribadi dengan izin berbasis sumber daya menggunakan IAM. AWS Hal ini agar pengguna tertentu atau instans Amazon EC2 dapat mengakses repositori kontainer dan citra. Anda dapat menggunakan CLI pilihan Anda untuk mendorong, menarik, dan mengelola citra Docker, citra Open Container Initiative (OCI), dan artefak yang kompatibel dengan OCI.

Note

Amazon ECR juga mendukung repositori citra kontainer publik. Untuk informasi selengkapnya, lihat [Apa itu Amazon ECR Public](#) di Panduan Pengguna Amazon ECR Publik.

Tim layanan AWS kontainer memelihara peta jalan publik di GitHub. Ini berisi informasi tentang pekerjaan yang dilakukan tim dan memungkinkan semua pelanggan AWS untuk memberikan umpan balik langsung. Untuk informasi selengkapnya, lihat [AWS Roadmap Kontainer](#).

Komponen Amazon ECR

Amazon ECR mengandung komponen berikut:

Registri

Registri pribadi Amazon ECR disediakan untuk setiap AWS akun; Anda dapat membuat satu atau lebih repositori di registri Anda dan menyimpan gambar Docker, gambar Open Container Initiative (OCI), dan artefak yang kompatibel dengan OCI di dalamnya. Untuk informasi selengkapnya, lihat [Registri pribadi Amazon ECR](#).

Token otorisasi

Klien Anda harus mengautentikasi ke registri pribadi Amazon ECR sebagai AWS pengguna sebelum dapat mendorong dan menarik gambar. Untuk informasi selengkapnya, lihat [Autentikasi registri pribadi](#).

Repositori

Repositori Amazon ECR berisi gambar Docker Anda, gambar Open Container Initiative (OCI), dan artefak yang kompatibel dengan OCI. Untuk informasi selengkapnya, lihat [Repositori pribadi Amazon ECR](#).

Kebijakan repositori

Anda dapat mengontrol akses ke repositori Anda dan konten di dalamnya dengan kebijakan repositori. Untuk informasi selengkapnya, lihat [Kebijakan repositori pribadi](#).

Citra

Anda dapat mendorong dan menarik citra kontainer ke repositori Anda. Anda dapat menggunakan citra ini secara lokal pada sistem pengembangan Anda, atau Anda dapat menggunakannya dalam definisi tugas Amazon ECS dan spesifikasi pod Amazon EKS. Untuk informasi lebih lanjut, lihat [Menggunakan citra Amazon ECR dengan Amazon ECS](#) dan [Menggunakan Citra Amazon ECR dengan Amazon EKS](#).

Fitur-Fitur Amazon ECR

Amazon ECR menyediakan fitur-fitur berikut:

- Kebijakan siklus hidup membantu mengelola siklus hidup citra di repositori Anda. Anda menentukan aturan yang mengakibatkan pembersihan citra yang tidak terpakai. Anda dapat menguji aturan sebelum menerapkannya ke repositori Anda. Untuk informasi selengkapnya, lihat [Kebijakan siklus hidup](#).
- Pemindaian citra membantu dalam mengidentifikasi kerentanan dalam citra kontainer Anda. Setiap repositori dapat dikonfigurasi untuk pindai saat mendorong. Hal ini memastikan bahwa setiap citra baru yang didorong ke repositori dipindai. Anda kemudian dapat mengambil hasil pemindaian citra. Untuk informasi selengkapnya, lihat [Pemindaian citra](#).
- Replikasi antar wilayah dan antar akun memudahkan Anda untuk memiliki citra di mana Anda membutuhkannya. Ini dikonfigurasi sebagai pengaturan registri dan berdasarkan per Wilayah. Untuk informasi selengkapnya, lihat [Pengaturan registri pribadi](#).
- Tarik melalui aturan cache menyediakan cara untuk menyimpan repositori di registri upstream di registri Amazon ECR pribadi Anda. Menggunakan aturan cache pull through, Amazon ECR akan secara berkala menjangkau registri upstream untuk memastikan gambar yang di-cache di registri pribadi Amazon ECR Anda mutakhir. Untuk informasi selengkapnya, lihat [Menggunakan aturan pull through cache](#).

Cara memulai dengan Amazon ECR

Untuk menggunakan Amazon ECR, Anda harus bersiap untuk menginstal AWS Command Line Interface dan Docker. Untuk informasi lebih lanjut, lihat [Menyiapkan Amazon ECR](#) dan [Menggunakan Amazon ECR dengan AWS CLI](#).

Harga untuk Amazon ECR

Dengan Amazon ECR, Anda hanya membayar untuk jumlah data yang Anda simpan di repositori Anda dan untuk transfer data dari dorongan dan tarikan citra Anda. Untuk informasi lebih lanjut, lihat [Harga Amazon ECR](#).

Menyiapkan Amazon ECR

Jika Anda telah mendaftar AWS dan telah menggunakan Amazon Elastic Container Service (Amazon ECS) atau Amazon Elastic Kubernetes Service (Amazon EKS), Anda akan segera dapat menggunakan Amazon ECR. Proses penyiapan untuk kedua layanan tersebut serupa, karena Amazon ECR adalah perpanjangan dari kedua layanan tersebut. Saat menggunakan AWS CLI dengan Amazon ECR, kami sarankan Anda menggunakan versi AWS CLI yang mendukung fitur Amazon ECR terbaru. Jika Anda tidak melihat dukungan untuk fitur Amazon ECR di AWS CLI, Anda harus melakukan upgrade ke versi terbaru. Untuk informasi selengkapnya, lihat <http://aws.amazon.com/cli/>.

Selesaikan tugas-tugas berikut ini untuk persiapan mendorong citra kontainer ke Amazon ECR untuk pertama kalinya. Jika Anda telah menyelesaikan salah satu langkah ini, Anda dapat melewatinya dan melanjutkan ke langkah berikutnya.

Mendaftar Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar Akun AWS

1. Buka <https://portal.aws.amazon.com/billing/signup>.
2. Ikuti petunjuk secara online.

Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

Saat Anda mendaftar Akun AWS, Pengguna root akun AWS akan dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, [tetapkan akses administratif ke pengguna administratif](#), dan hanya gunakan pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

AWS akan mengirimkan email konfirmasi kepada Anda setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun saat ini dan mengelola akun dengan mengunjungi <https://aws.amazon.com/> dan memilih Akun Saya.

Membuat pengguna administratif

Setelah mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat sebuah pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Mengamankan Pengguna root akun AWS Anda

1. Masuk ke [AWS Management Console](#) sebagai pemilik akun dengan memilih Pengguna root dan memasukkan alamat email Akun AWS Anda. Di halaman berikutnya, masukkan kata sandi Anda.

Untuk bantuan masuk menggunakan pengguna root, lihat [Masuk sebagai pengguna root](#) dalam Panduan Pengguna AWS Sign-In.

2. Aktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

Untuk petunjuknya, silakan lihat [Mengaktifkan perangkat MFA virtual untuk pengguna root Akun AWS Anda \(konsol\)](#) dalam Panduan Pengguna IAM.

Membuat pengguna administratif

1. Aktifkan Pusat Identitas IAM.

Untuk mendapatkan petunjuk, silakan lihat [Mengaktifkan AWS IAM Identity Center](#) di Panduan Pengguna AWS IAM Identity Center.

2. Di Pusat Identitas IAM, berikan akses administratif ke sebuah pengguna administratif.

Untuk mendapatkan tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, silakan lihat [Mengonfigurasi akses pengguna dengan Direktori Pusat Identitas IAM default](#) di Panduan Pengguna AWS IAM Identity Center.

Masuk sebagai pengguna administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email Anda saat Anda membuat pengguna Pusat Identitas IAM.

Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat [Masuk ke portal akses AWS](#) dalam Panduan Pengguna AWS Sign-In.

Memulai Amazon ECR dengan menggunakan AWS Management Console

Memulai Amazon ECR dengan membuat repositori di konsol Amazon ECR. Konsol Amazon ECR memandu Anda melalui suatu proses untuk mulai membuat repositori pertama Anda.

Sebelum memulai, pastikan Anda telah menyelesaikan langkah-langkah dalam [Menyiapkan Amazon ECR](#).

Membuat repositori citra

Repositori adalah tempat Anda menyimpan citra Docker atau Open Container Initiative (OCI) di Amazon ECR. Setiap kali Anda mendorong atau menarik citra dari Amazon ECR, Anda menentukan repositori dan lokasi registrasi yang menginformasikan tempat mendorong citra atau tempat menarik citra tersebut.

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Pilih Memulai.
3. Untuk pengaturan Visibilitas, pilih Pribadi.
4. Untuk nama Repository, tentukan nama untuk repositori.
5. Untuk Ketentuan tanda, pilih pengaturan ketentuan tanda untuk repositori. Repositori yang dikonfigurasi dengan tanda tetap akan mencegah tanda citra ditimpa. Untuk informasi selengkapnya, lihat [Tanda ketentuan citra](#).
6. Untuk Pindai saat mendorong, pilih pengaturan pemindaian citra untuk repositori. Repositori yang terkonfigurasi untuk pindai saat mendorong akan memulai pemindaian citra setiap kali citra didorong, jika tidak maka pemindaian citra perlu dimulai secara manual.

Important

Mengkonfigurasi pemindaian gambar di tingkat repositori telah usang demi mengonfigurasinya di tingkat registri. Untuk informasi selengkapnya, lihat [Pemindaian citra](#).

7. Untuk enkripsi KMS, pilih apakah akan mengaktifkan enkripsi sisi server menggunakan AWS KMS kunci yang disimpan dalam AWS Key Management Service layanan. Untuk informasi selengkapnya tentang fitur ini, lihat [Enkripsi diam](#).

8. Pilih Buat repositori.

Membuat, menandai, dan mendorong citra Docker

Di bagian wizard ini, Anda menggunakan Docker CLI untuk menandai citra lokal yang tersedia (bahwa Anda telah membangunnya dari Dockerfile atau ditarik dari registrasi lain, seperti Docker Hub) dan kemudian mendorong citra yang ditandai ke registrasi Amazon ECR Anda. Untuk langkah-langkah lebih rinci tentang menggunakan Docker CLI, lihat [Menggunakan Amazon ECR dengan AWS CLI](#).

1. Pilih repositori yang Anda buat dan pilih Lihat perintah dorong untuk melihat langkah-langkah untuk mendorong citra ke repositori baru Anda.
2. Jalankan perintah login yang mengautentikasi klien Docker Anda ke registrasi Anda dengan menggunakan perintah dari konsol ke jendela terminal. Perintah ini menyediakan token otorisasi yang berlaku selama 12 jam.
3. (Opsional) Jika Anda memiliki Dockerfile untuk citra saat mendorong, bangunlah citra dan tanda untuk repositori baru Anda. Menggunakan `docker build` perintah dari konsol ke jendela terminal. Pastikan bahwa Anda berada di direktori yang sama seperti Dockerfile Anda.
4. Tandai citra dengan URI registrasi Amazon ECR dan repositori baru Anda dengan menempelkan docker tag dari konsol ke jendela terminal. Perintah konsol mengasumsikan bahwa citra Anda dibangun dari Dockerfile pada langkah sebelumnya. Jika Anda tidak membangun citra Anda dari Dockerfile, gantikan instans pertama `repository:latest` dengan ID citra atau nama citra lokal Anda untuk mendorong.
5. Dorong gambar yang baru ditandai ke repositori Anda dengan menggunakan `docker push` perintah di jendela terminal.
6. Pilih Tutup.

Menggunakan Amazon ECR dengan AWS CLI

Langkah-langkah berikut memandu Anda dalam melakukan langkah-langkah yang diperlukan untuk mendorong kontainer citra ke repositori Amazon ECR privat untuk pertama kalinya dengan menggunakan Docker CLI AWS CLI.

[Untuk informasi selengkapnya tentang alat lain yang tersedia untuk mengelola AWS sumber daya Anda, termasuk AWS SDK yang berbeda, toolkit IDE, dan alat baris PowerShell perintah Windows, lihat <http://aws.amazon.com/tools/>.](http://aws.amazon.com/tools/)

Prasyarat

Sebelum memulai, pastikan Anda telah menyelesaikan langkah-langkah dalam [Menyiapkan Amazon ECR](#).

Jika Anda belum memiliki AWS CLI yang terbaru serta Docker yang terinstal dan siap digunakan, lakukan langkah-langkah berikut untuk menginstal kedua alat ini.

Instal AWS CLI

Anda dapat menggunakan alat baris AWS perintah untuk mengeluarkan perintah di baris perintah Anda untuk melakukan Amazon ECR dan AWS tugas lainnya. Ini mungkin lebih cepat dan nyaman dibandingkan jika menggunakan konsol. Alat baris perintah juga berguna jika Anda ingin membangun skrip yang melakukan tugas AWS.

Untuk menggunakan AWS CLI dengan Amazon ECR, instal AWS CLI versi terbaru. Untuk selengkapnya, lihat [Menginstal AWS Command Line Interface](#) di Panduan AWS Command Line Interface Pengguna.

Instal Docker

Docker tersedia dalam banyak sistem operasi yang berbeda, termasuk sebagian besar distribusi Linux modern, seperti Ubuntu, dan bahkan macOS dan Windows. Untuk informasi lebih lanjut tentang cara menginstal Docker pada sistem operasi tertentu Anda, kunjungi situs web [panduan penginstalan Docker](#).

Anda tidak memerlukan sistem pengembangan lokal untuk menggunakan Docker. Jika Anda sudah menggunakan Amazon EC2, Anda dapat meluncurkan instans Amazon Linux 2023 dan menginstal Docker untuk memulai.

Jika Anda sudah menginstal Docker, langsung ke [Langkah 1: Buat citra Docker](#).

Untuk menginstal Docker pada instans Amazon EC2 menggunakan AMI Amazon Linux 2023

1. Luncurkan instance dengan AMI Amazon Linux 2023 terbaru. Untuk informasi selengkapnya, lihat [Meluncurkan instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
2. Connect ke instans Anda. Untuk informasi lebih lanjut, lihat [Connect ke Instans Linux Anda](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.
3. Memperbarui paket yang diinstal dan paket cache pada instans Anda.

```
sudo yum update -y
```

4. Instal paket Edisi Komunitas Docker terbaru.

```
sudo yum install docker
```

5. Mulai layanan Docker.

```
sudo service docker start
```

6. Tambahkan `ec2-user` ke grup `docker` sehingga Anda dapat menjalankan perintah Docker tanpa menggunakan `sudo`.

```
sudo usermod -a -G docker ec2-user
```

7. Keluar dan masuk kembali untuk mengambil izin grup `docker` yang baru. Anda dapat melakukannya dengan menutup jendela terminal SSH Anda saat ini dan menghubungkan kembali ke instans Anda yang baru. Sesi SSH baru Anda akan memiliki izin grup `docker` yang sesuai.
8. Verifikasi bahwa `ec2-user` dapat menjalankan perintah Docker tanpa `sudo`.

```
docker info
```

Note

Dalam beberapa kasus, Anda mungkin perlu melakukan booting ulang pada instans Anda untuk memberikan izin bagi `ec2-user` untuk mengakses daemon Docker. Coba me-reboot instans Anda jika Anda melihat kesalahan berikut:

Cannot connect to the Docker daemon. Is the docker daemon running on this host?

Langkah 1: Buat citra Docker

Pada langkah ini, Anda membuat gambar Docker dari aplikasi web sederhana, dan mengujinya di sistem lokal Anda atau instans Amazon EC2.

Untuk membuat citra Docker dari aplikasi web sederhana

1. Buat file bernama `Dockerfile`. Dockerfile adalah manifes yang menjelaskan citra dasar yang akan digunakan untuk citra Docker Anda dan apa yang ingin Anda instal dan jalankan di atasnya. Untuk informasi selengkapnya tentang Dockerfiles, buka [Referensi Dockerfile](#).

```
touch Dockerfile
```

2. Edit Dockerfile yang baru saja Anda buat dan tambahkan konten berikut.

```
FROM public.ecr.aws/amazonlinux/amazonlinux:latest

# Install dependencies
RUN yum update -y && \
    yum install -y httpd

# Install apache and write hello world message
RUN echo 'Hello World!' > /var/www/html/index.html

# Configure apache
RUN echo 'mkdir -p /var/run/httpd' >> /root/run_apache.sh && \
    echo 'mkdir -p /var/lock/httpd' >> /root/run_apache.sh && \
    echo '/usr/sbin/httpd -D FOREGROUND' >> /root/run_apache.sh && \
    chmod 755 /root/run_apache.sh

EXPOSE 80

CMD /root/run_apache.sh
```

Dockerfile ini menggunakan gambar Amazon Linux 2 publik yang dihosting di Amazon ECR Public. RUNInstruksi memperbarui cache paket, menginstal beberapa paket perangkat lunak untuk server web, dan kemudian menulis "Hello World!" konten ke root dokumen server web. Instruksi EXPOSE mengekspos port 80 pada kontainer, dan instruksi CMD memulai server web.

3. Membangun citra Docker dari Dockerfile Anda.

Note

Beberapa versi Docker mungkin memerlukan jalur lengkap ke Dockerfile Anda dalam perintah berikut, bukan jalur relatif yang ditunjukkan di bawah ini.

```
docker build -t hello-world .
```

4. Buat daftar gambar kontainer Anda.

```
docker images --filter reference=hello-world
```

Output:

REPOSITORY	TAG	IMAGE ID	CREATED
hello-world	latest	e9ffedc8c286	4 minutes ago
SIZE			
194MB			

5. Jalankan citra yang baru dibuat. Opsi `-p 80:80` memetakan port 80 yang terbuka pada kontainer ke port 80 pada sistem host. Untuk informasi lebih lanjut tentang docker run, buka [Referensi menjalankan Docker](#).

```
docker run -t -i -p 80:80 hello-world
```

Note

Output dari server web Apache ditampilkan di jendela terminal. Anda dapat mengabaikan pesan "Could not reliably determine the fully qualified domain name".

6. Buka peramban dan arahkan ke server yang menjalankan Docker dan meng-host kontainer Anda.
 - Jika Anda menggunakan instans EC2, nilai ini adalah nilai DNS Publik untuk server, yang merupakan alamat yang sama yang Anda gunakan untuk terhubung ke instans dengan SSH. Pastikan bahwa grup keamanan untuk instans Anda mengizinkan lalu lintas masuk pada port 80.
 - Jika Anda menjalankan Docker secara lokal, arahkan peramban Anda ke <http://localhost/>.
 - Jika Anda menggunakan docker-machine pada komputer Windows atau Mac, temukan alamat IP VirtualBox VM yang menghosting Docker dengan docker-machine ip perintah, ganti nama mesin dengan *nama mesin* docker yang Anda gunakan.

```
docker-machine ip machine-name
```

Anda akan melihat halaman web dengan pernyataan “Hello World!” .

7. Hentikan kontainer Docker dengan mengetik Ctrl + c.

Langkah 2: Autentikasi ke registrasi default

Setelah Anda menginstal dan mengonfigurasi AWS CLI, lakukan autentikasi Docker CLI ke registrasi default Anda. Dengan cara tersebut, perintah docker dapat mendorong dan menarik citra dengan Amazon ECR. AWS CLI menyediakan perintah get-login-password untuk menyederhanakan proses autentikasi.

Untuk mengautentikasi Docker ke registri Amazon ECR dengan get-login-password, jalankan perintah `aws ecr get-login-password` Ketika meneruskan token otorisasi ke perintah docker login, gunakan nilai AWS untuk nama pengguna dan tentukan URI registri Amazon ECR yang ingin Anda autentikasi. Jika melakukan autentikasi untuk beberapa registri, Anda harus mengulangi perintah tersebut untuk setiap registri.

Important

Jika Anda menerima pesan kesalahan, instal atau upgrade ke versi terbaru AWS CLI. Untuk informasi selengkapnya, lihat [Menginstal AWS Command Line Interface](#) dalam Panduan Pengguna Amazon EKS AWS Command Line Interface.

- [get-login-password](#) (AWS CLI)

```
aws ecr get-login-password --region region | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

- [LoginCommandDapatkan-ECR](#) () AWS Tools for Windows PowerShell

```
(Get-ECRLoginCommand).Password | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

Langkah 3: Buat repositori

Kini Anda memiliki citra yang akan didorong ke Amazon ECR, Anda harus membuat repositori untuk menahannya. Dalam contoh ini, Anda membuat repositori yang disebut `hello-repository` yang kemudian Anda mendorong citra `hello-world:latest`. Untuk membuat repositori, jalankan perintah berikut:

```
aws ecr create-repository \  
  --repository-name hello-repository \  
  --region region
```

Langkah 4: Dorong citra ke Amazon ECR

Sekarang Anda dapat mendorong citra Anda ke repositori Amazon ECR yang Anda buat di bagian sebelumnya. Anda menggunakan docker CLI untuk mendorong citra, tetapi ada beberapa persyaratan yang harus dipenuhi agar tugas ini dapat diselesaikan dengan baik:

- Versi minimum docker diinstal: 1.7
- Token otorisasi Amazon ECR telah terkonfigurasi dengan docker login.
- Amazon ECR repositori tersedia dan pengguna memiliki akses untuk mendorong ke repositori.

Setelah persyaratan tersebut terpenuhi, Anda dapat mendorong citra Anda ke repositori yang baru dibuat di registrasi default untuk akun Anda.

Untuk membuat tanda dan mendorong citra ke Amazon ECR

1. Cantumkan citra yang telah Anda simpan secara lokal untuk mengidentifikasi citra yang akan ditandai dan didorong.

```
docker images
```

Output:

REPOSITORY	TAG	IMAGE ID	CREATED
hello-world	latest	e9ffedc8c286	4 minutes ago
241MB			

2. Tandai citra untuk mendorongnya ke repositori Anda.

```
docker tag hello-world:latest aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository
```

3. Mendorong citra

```
docker push aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository
```

Output:

```
The push refers to a repository [aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository] (len: 1)
e9ae3c220b23: Pushed
a6785352b25c: Pushed
0998bf8fb9e9: Pushed
0a85502c06c9: Pushed
latest: digest: sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636EXAMPLE
size: 6774
```

Langkah 5: Menarik citra dari Amazon ECR

Setelah citra Anda didorong ke repositori Amazon ECR Anda, Anda dapat menariknya dari lokasi lain. Gunakan docker CLI untuk menarik citra, tetapi ada beberapa persyaratan yang harus dipenuhi agar tugas ini dapat diselesaikan dengan baik:

- Versi minimum docker diinstal: 1.7
- Token otorisasi Amazon ECR telah terkonfigurasi dengan docker login.
- Amazon ECR repositori tersedia dan pengguna memiliki akses untuk melakukan penarikan dari repositori.

Setelah prasyarat tersebut terpenuhi, Anda bisa menarik citra Anda. Untuk menarik citra contoh Anda dari Amazon ECR, jalankan perintah berikut:

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository:latest
```

Output:

```
latest: Pulling from hello-repository
0a85502c06c9: Pull complete
0998bf8fb9e9: Pull complete
a6785352b25c: Pull complete
e9ae3c220b23: Pull complete
Digest: sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636EXAMPLE
Status: Downloaded newer image for aws_account_id.dkr.region.amazonaws.com/hello-
repository:latest
```

Langkah 6: Menghapus citra

Jika Anda tidak lagi membutuhkan atau menginginkan citra tertentu di salah satu repositori Anda, maka Anda dapat menghapusnya dengan perintah `batch-delete-image`. Untuk menghapus citra, Anda harus menentukan bahwa repositori tersebut menyimpannya dan termasuk salah satu dari nilai `imageTag` atau `imageDigest` citra. Contoh di bawah ini menghapus citra di repositori `hello-repository` dengan tanda citra `latest`.

```
aws ecr batch-delete-image \
  --repository-name hello-repository \
  --image-ids imageTag=latest \
  --region region
```

Langkah 7: Menghapus repositori

Jika Anda tidak lagi membutuhkan atau menginginkan seluruh repositori citra, Anda dapat menghapus repositori. Secara default, Anda tidak dapat menghapus repositori yang berisi citra;

Namun, tanda `--force` mengizinkan penghapusan ini. Untuk menghapus repositori yang berisi citra (dan semua citra di dalamnya), jalankan perintah berikut.

```
aws ecr delete-repository \  
  --repository-name hello-repository \  
  --force \  
  --region region
```

Registri pribadi Amazon ECR

Registri pribadi Amazon ECR menyimpan gambar kontainer Anda dalam arsitektur yang sangat tersedia dan dapat diskalakan. Anda dapat menggunakan registri pribadi Anda untuk mengelola repositori citra pribadi yang terdiri dari citra dan artefak Docker dan Open Container Initiative (OCI). Setiap akun AWS disediakan registri pribadi default Amazon ECR. Untuk informasi lebih lanjut tentang registri publik Amazon ECR, lihat [Registri publik](#) di Panduan pengguna Amazon Elastic Container Registry.

Konsep registri pribadi

- URL untuk registri pribadi default Anda adalah `https://aws_account_id.dkr.ecr.us-west-2.amazonaws.com`.
- Secara default, akun Anda telah membaca dan menulis akses ke repositori di registri pribadi Anda. Namun, pengguna memerlukan izin untuk melakukan panggilan ke Amazon ECR API dan untuk mendorong atau menarik gambar ke dan dari repositori pribadi Anda. Amazon ECR menyediakan beberapa kebijakan yang dikelola untuk mengontrol akses pengguna pada berbagai tingkat. Untuk informasi selengkapnya, lihat [Contoh kebijakan berbasis identitas Amazon Elastic Container Registry](#).
- Anda harus mengautentikasi klien Docker ke registri pribadi Anda sehingga Anda dapat menggunakan perintah `docker push` dan `docker pull` untuk mendorong dan menarik citra ke dan dari repositori dalam registri tersebut. Untuk informasi selengkapnya, lihat [Autentikasi registri pribadi](#).
- Repositori pribadi dapat dikontrol dengan kebijakan akses pengguna dan kebijakan repositori. Untuk informasi lebih lanjut tentang kebijakan repositori, lihat [Kebijakan repositori pribadi](#).
- Repositori di registri pribadi Anda dapat direplikasi di seluruh wilayah di registri pribadi Anda sendiri dan di akun terpisah dengan mengonfigurasi replikasi untuk registri pribadi Anda. Untuk informasi selengkapnya, lihat [Replikasi citra pribadi](#).

Autentikasi registri pribadi

Anda dapat menggunakan AWS Management Console, SDK AWS CLI, atau AWS SDK untuk membuat dan mengelola repositori pribadi. Anda juga dapat menggunakan metode tersebut untuk melakukan beberapa tindakan pada citra, seperti mendaftar atau menghapusnya. Klien ini

menggunakan metode AWS otentikasi standar. Meskipun Anda dapat menggunakan API Amazon ECR untuk mendorong dan menarik citra, Anda lebih cenderung menggunakan Docker CLI atau perpustakaan Docker khusus bahasa.

Docker CLI tidak mendukung metode otentikasi IAM asli. Langkah-langkah tambahan harus diambil sehingga Amazon ECR dapat mengautentikasi dan mengotorisasi permintaan dorongan dan tarikan Docker.

Metode autentikasi registri yang dirinci dalam bagian berikut tersedia.

Menggunakan Amazon ECR credential helper

Amazon ECR menyediakan Docker credential helper yang membuatnya lebih mudah untuk menyimpan dan menggunakan kredensial Docker saat mendorong dan menarik citra ke Amazon ECR. Untuk langkah-langkah instalasi dan konfigurasi, lihat [Amazon ECR Docker Credential Helper](#).

Note

Pembantu kredensial Amazon ECR Docker saat ini tidak mendukung otentikasi multi-faktor (MFA).

Menggunakan token otorisasi

Lingkup perizinan token otorisasi sesuai dengan IAM utama yang digunakan untuk mengambil token autentikasi. Token otentikasi digunakan untuk mengakses registri Amazon ECR di mana IAM utama Anda memiliki akses dan berlaku selama 12 jam. Untuk mendapatkan token otorisasi, Anda harus menggunakan operasi [GetAuthorizationToken](#) API untuk mengambil token otorisasi berenkode base64 yang berisi nama pengguna dan kata sandi yang dikodekan. AWS CLI `get-login-password` perintah menyederhanakan ini dengan mengambil dan mendekode token otorisasi yang kemudian dapat Anda salurkan ke perintah untuk mengautentikasi. `docker login`

Untuk mengautentikasi Docker ke registri pribadi Amazon ECR dengan CLI

Untuk mengautentikasi Docker ke registri Amazon ECR dengan `get-login-password`, jalankan perintah `aws ecr get-login-password`. Ketika meneruskan token otorisasi ke perintah `docker login`, gunakan nilai AWS untuk nama pengguna dan tentukan URI registri Amazon ECR yang ingin Anda autentikasi. Jika melakukan autentikasi untuk beberapa registri, Anda harus mengulangi perintah tersebut untuk setiap registri.

⚠ Important

Jika Anda menerima pesan kesalahan, instal atau upgrade ke versi terbaru AWS CLI. Untuk informasi selengkapnya, lihat [Menginstal AWS Command Line Interface](#) dalam Panduan Pengguna Amazon EKS AWS Command Line Interface .

- [get-login-password](#) (AWS CLI)

```
aws ecr get-login-password --region region | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

- [LoginCommandDapatkan-ECR](#) ()AWS Tools for Windows PowerShell

```
(Get-ECRLoginCommand).Password | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

Menggunakan autentikasi HTTP API

Amazon ECR mendukung [HTTP API Registri Docker](#). Namun, karena Amazon ECR adalah registri pribadi, Anda harus memberikan token otorisasi pada setiap permintaan HTTP. Anda dapat menambahkan header otorisasi HTTP menggunakan -H opsi untuk curl dan meneruskan token otorisasi yang disediakan oleh perintah. get-authorization-token AWS CLI

Untuk mengautentikasi dengan HTTP API Amazon ECR

1. Ambil token otorisasi dengan AWS CLI dan atur ke variabel lingkungan.

```
TOKEN=$(aws ecr get-authorization-token --output text --query 'authorizationData[].authorizationToken')
```

2. Untuk mengautentikasi ke API, berikan variabel \$TOKEN ke opsi -H dari curl. Misalnya, perintah berikut mencantumkan tanda citra dalam repositori Amazon ECR. Untuk informasi lebih lanjut, lihat dokumentasi referensi [HTTP API Registri Docker](#).

```
curl -i -H "Authorization: Basic $TOKEN" https://aws_account_id.dkr.ecr.region.amazonaws.com/v2/amazonlinux/tags/list
```

Outputnya adalah sebagai berikut:

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8
Date: Thu, 04 Jan 2018 16:06:59 GMT
Docker-Distribution-Api-Version: registry/2.0
Content-Length: 50
Connection: keep-alive

{"name":"amazonlinux","tags":["2017.09","latest"]}
```

Pengaturan registri pribadi

Amazon ECR menggunakan pengaturan registri pribadi untuk mengonfigurasi fitur di tingkat registri. Pengaturan registri pribadi dikonfigurasi secara terpisah untuk setiap Wilayah. Anda dapat menggunakan pengaturan registri pribadi untuk mengonfigurasi fitur-fitur berikut.

- Izin registri —Kebijakan izin registri memberikan kontrol atas replikasi dan menarik izin cache. Untuk informasi selengkapnya, lihat [Izin Registri Pribadi](#).
- Tarik aturan cache —Aturan pull through cache digunakan untuk menyimpan gambar dari registri upstream di registri pribadi Amazon ECR Anda. Untuk informasi selengkapnya, lihat [Menggunakan aturan pull through cache](#).
- Konfigurasi replikasi —Konfigurasi replikasi digunakan untuk mengontrol apakah repositori Anda disalin di seluruh Wilayah atau akun. AWS Lihat informasi yang lebih lengkap di [Replikasi citra pribadi](#)
- Templat pembuatan repositori —Templat pembuatan repositori digunakan untuk menentukan pengaturan standar yang akan diterapkan saat repositori baru dibuat oleh Amazon ECR atas nama Anda. Misalnya, repositori yang dibuat oleh tindakan pull through cache. Untuk informasi selengkapnya, lihat [Kelola templat pembuatan repositori Anda](#).
- Konfigurasi pemindaian —Secara default, registri Anda diaktifkan untuk pemindaian dasar. Anda dapat mengaktifkan pemindaian yang disempurnakan yang menyediakan mode pemindaian otomatis dan berkelanjutan yang memindai kerentanan paket sistem operasi dan bahasa pemrograman. Untuk informasi selengkapnya, lihat [Pemindaian citra](#).

Izin Registri Pribadi

Amazon ECR menggunakan kebijakan registri untuk memberikan izin kepada AWS kepala sekolah di tingkat registri pribadi. Izin ini digunakan untuk cakupan akses ke replikasi dan menarik melalui fitur cache.

Amazon ECR hanya memberlakukan izin berikut di tingkat registri pribadi. Jika ada tindakan tambahan yang ditambahkan ke kebijakan registri, kesalahan akan terjadi.

- `ecr:ReplicateImage`— Memberikan izin ke akun lain, yang disebut sebagai registri sumber, untuk mereplikasi gambarnya ke registri Anda. Ini hanya digunakan untuk replikasi lintas akun.
- `ecr:BatchImportUpstreamImage`— Memberikan izin untuk mengambil gambar eksternal dan mengimpornya ke registri pribadi Anda.
- `ecr:CreateRepository`— Memberikan izin untuk membuat repositori di registri pribadi. Izin ini diperlukan jika repositori yang menyimpan gambar yang direplikasi atau di-cache belum ada di registri pribadi.

Note

Meskipun dimungkinkan untuk menambahkan `ecr:*` tindakan ke kebijakan izin registri pribadi, dianggap praktik terbaik untuk hanya menambahkan tindakan spesifik yang diperlukan berdasarkan fitur yang Anda gunakan daripada menggunakan wildcard.

Topik

- [Menetapkan pernyataan izin registri pribadi](#)
- [Menghapus pernyataan izin registri pribadi](#)
- [Contoh kebijakan registri pribadi](#)

Menetapkan pernyataan izin registri pribadi

Anda dapat menambahkan atau memperbarui kebijakan izin untuk registri Anda dengan menggunakan langkah-langkah berikut. Anda dapat menambahkan beberapa pernyataan kebijakan per registri. Untuk contoh kebijakan, lihat [Contoh kebijakan registri pribadi](#).

Topik

- [Izin registri pribadi untuk replikasi](#)
- [Izin registri pribadi untuk menarik cache](#)

Izin registri pribadi untuk replikasi

Jenis kebijakan lintas akun digunakan untuk memberikan izin kepada AWS prinsipal, memungkinkan replikasi repositori dari registri sumber ke registri Anda. Secara default, Anda memiliki izin untuk mengonfigurasi replikasi antar wilayah dalam registri Anda sendiri. Anda hanya perlu mengonfigurasi kebijakan registri jika Anda memberikan izin akun lain untuk mereplikasi konten ke registri Anda.

Kebijakan registri harus memberikan izin untuk tindakan API `ecr:ReplicateImage`. API ini adalah API Amazon ECR internal yang dapat mereplikasi citra antar Wilayah atau akun. Anda juga dapat memberikan izin untuk `ecr:CreateRepository`, yang memungkinkan Amazon ECR untuk membuat repositori di registri Anda jika belum ada. Jika izin `ecr:CreateRepository` tidak diberikan, repositori dengan nama yang sama dengan repositori sumber harus dibuat secara manual di registri Anda. Jika keduanya tidak dilakukan, replikasi gagal. Setiap tindakan yang gagal `CreateRepository` atau `ReplicateImage` API muncul di CloudTrail.

Untuk mengonfigurasi kebijakan izin untuk replikasi ()AWS Management Console

Untuk mengonfigurasi kebijakan izin replikasi untuk registri pribadi ()AWS Management Console

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah untuk mengonfigurasi kebijakan registri Anda.
3. Di panel navigasi, pilih Registri pribadi, Izin registri.
4. Pada halaman Izin registri, pilih Hasilkan pernyataan.
5. Menyelesaikan langkah-langkah berikut untuk menentukan pernyataan kebijakan Anda menggunakan kebijakan generator.
 - a. Untuk jenis Kebijakan, pilih Kebijakan lintas akun.
 - b. Untuk ID Pernyataan, masukkan ID pernyataan unik. Bidang ini digunakan sebagai `Sid` pada kebijakan registri.
 - c. Untuk Akun, masukkan ID akun untuk setiap akun yang ingin Anda berikan izin. Saat menentukan beberapa ID akun, pisahkan dengan koma.
6. Perluas bagian Pernyataan kebijakan pratinjau untuk meninjau pernyataan kebijakan izin registri.
7. Setelah pernyataan kebijakan dikonfirmasi, pilih Tambahkan ke kebijakan untuk menyimpan kebijakan ke registri Anda.

Untuk mengonfigurasi kebijakan izin untuk replikasi (AWS CLI)

Untuk mengonfigurasi kebijakan izin untuk registri pribadi (AWS CLI)

1. Buat file bernama `registry_policy.json` dan isi dengan kebijakan registri.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReplicationAccessCrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::source_account_id:root"
      },
      "Action": [
        "ecr:CreateRepository",
        "ecr:ReplicateImage"
      ],
      "Resource": [
        "arn:aws:ecr:us-west-2:your_account_id:repository/*"
      ]
    }
  ]
}
```

2. Buat kebijakan registri menggunakan file kebijakan.

```
aws ecr put-registry-policy \
  --policy-text file://registry_policy.json \
  --region us-west-2
```

3. Ambil kebijakan untuk registri Anda untuk mengonfirmasi.

```
aws ecr get-registry-policy \
  --region us-west-2
```

Izin registri pribadi untuk menarik cache

Izin registri pribadi Amazon ECR dapat digunakan untuk mencakup izin entitas IAM individu untuk menggunakan cache pull through. Jika entitas IAM memiliki lebih banyak izin yang diberikan oleh kebijakan IAM daripada yang diberikan oleh kebijakan izin registri, kebijakan IAM akan diutamakan.

Untuk membuat kebijakan izin registri pribadi ()AWS Management Console

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah untuk mengonfigurasi pernyataan izin registri pribadi Anda.
3. Di panel navigasi, pilih Registri pribadi, Izin registri.
4. Pada halaman Izin registri, pilih Hasilkan pernyataan.
5. Untuk setiap pull through pernyataan kebijakan izin cache yang ingin Anda buat, lakukan hal berikut.
 - a. Untuk jenis Policy, pilih Pull through cache policy.
 - b. Untuk id Pernyataan, berikan nama untuk kebijakan pernyataan cache tarik melalui.
 - c. Untuk entitas IAM, tentukan pengguna, grup, atau peran yang akan disertakan dalam kebijakan.
 - d. Untuk namespace Repositori, pilih aturan pull through cache untuk mengaitkan kebijakan dengan.
 - e. Untuk nama Repositori, tentukan nama dasar repositori untuk menerapkan aturan. Misalnya, jika Anda ingin menentukan repositori Amazon Linux di Amazon ECR Public, nama repositori akan menjadi. `amazonlinux`

Menghapus pernyataan izin registri pribadi

Anda dapat menghapus semua pernyataan kebijakan izin untuk registri Anda dengan menggunakan langkah-langkah berikut.

Untuk mengonfigurasi kebijakan izin untuk registri pribadi (AWS Management Console)

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah untuk mengonfigurasi kebijakan izin registri Anda.
3. Di panel navigasi, pilih Registri.
4. Pada halaman Registri, pilih registri Pribadi Anda dan pilih Edit.

5. Pada Izin Registri Pribadi, pilih Hapus pernyataan.
6. Pada layar konfirmasi Hapus kebijakan registri, pilih Hapus kebijakan.

Untuk mengonfigurasi kebijakan izin untuk registri pribadi (AWS CLI)

1. Hapus kebijakan registri.

```
aws ecr delete-registry-policy \  
  --region us-west-2
```

2. Ambil kebijakan untuk registri Anda untuk mengonfirmasi.

```
aws ecr get-registry-policy \  
  --region us-west-2
```

Contoh kebijakan registri pribadi

Contoh berikut menunjukkan pernyataan kebijakan izin registri yang dapat Anda gunakan untuk mengontrol izin yang dimiliki pengguna atas registri Amazon ECR Anda.

Note

Dalam setiap contoh, jika `ecr:CreateRepository` tindakan dihapus dari pernyataan izin registri Anda, replikasi masih dapat terjadi. Namun, agar replikasi berhasil, Anda perlu membuat repositori dengan nama yang sama dalam akun Anda.

Contoh: Izinkan pengguna root dari akun sumber untuk mereplikasi semua repositori

```
{  
  "Version":"2012-10-17",  
  "Statement":[  
    {  
      "Sid":"ReplicationAccessCrossAccount",  
      "Effect":"Allow",  
      "Principal":{  
        "AWS":"arn:aws:iam::source_account_id:root"  
      },  
      "Action":[
```



```

        "ecr:CreateRepository",
        "ecr:ReplicateImage"
    ],
    "Resource": [
        "arn:aws:ecr:us-west-2:your_account_id:repository/*"
    ]
}
]
}

```

Contoh: Izinkan beberapa akun

```

{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"ReplicationAccessCrossAccount",
      "Effect":"Allow",
      "Principal":{
        "AWS":"arn:aws:iam::source_account_id:root"
      },
      "Action":[
        "ecr:CreateRepository",
        "ecr:ReplicateImage"
      ],
      "Resource": [
        "arn:aws:ecr:us-west-2:your_account_id:repository/*"
      ]
    },
    {
      "Sid":"ReplicationAccessCrossAccount",
      "Effect":"Allow",
      "Principal":{
        "AWS":"arn:aws:iam::source_account_id:root"
      },
      "Action":[
        "ecr:CreateRepository",
        "ecr:ReplicateImage"
      ],
      "Resource": [
        "arn:aws:ecr:us-west-2:your_account_id:repository/*"
      ]
    }
  ]
}

```

```
]
}
```

Contoh: Izinkan pengguna root dari akun sumber untuk mereplikasi semua repositori dengan awalan. **prod-**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReplicationAccessCrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::source_account_id:root"
      },
      "Action": [
        "ecr:CreateRepository",
        "ecr:ReplicateImage"
      ],
      "Resource": [
        "arn:aws:ecr:us-west-2:your_account_id:repository/prod-*"
      ]
    }
  ]
}
```

Repository pribadi Amazon ECR

Amazon Elastic Container Registry (Amazon ECR) menyediakan operasi API untuk membuat, memantau, dan menghapus repository citra dan mengatur izin yang mengontrol siapa yang dapat mengaksesnya. Anda dapat melakukan tindakan yang sama di bagian Repository di konsol Amazon ECR. Amazon ECR juga terintegrasi dengan Docker CLI, sehingga Anda mendorong dan menarik gambar dari lingkungan pengembangan Anda ke repository Anda.

Topik

- [Konsep repository pribadi](#)
- [Membuat repository pribadi](#)
- [Melihat detail repository pribadi](#)
- [Mengedit repository pribadi](#)
- [Menghapus repository pribadi](#)
- [Kebijakan repository pribadi](#)
- [Menandai repository pribadi](#)

Konsep repository pribadi

- Secara default, akun Anda telah membaca dan menulis akses ke repository di registri default Anda (`aws_account_id.dkr.ecr.region.amazonaws.com`). Namun, pengguna memerlukan izin untuk melakukan panggilan ke Amazon ECR API dan untuk mendorong atau menarik gambar ke dan dari repository Anda. Amazon ECR menyediakan beberapa kebijakan yang dikelola untuk mengontrol akses pengguna pada berbagai tingkat. Untuk informasi selengkapnya, lihat [Contoh kebijakan berbasis identitas Amazon Elastic Container Registry](#).
- Repository dapat dikontrol dengan kebijakan akses pengguna dan kebijakan repository individu. Untuk informasi selengkapnya, lihat [Kebijakan repository pribadi](#).
- Nama repository dapat mendukung namespace, yang dapat Anda gunakan untuk mengelompokkan repository serupa. Misalnya, jika ada beberapa tim yang menggunakan registri yang sama, Tim A dapat menggunakan namespace `team-a`, dan Tim B dapat menggunakan namespace `team-b`. Dengan melakukan ini, setiap tim memiliki citra mereka sendiri yang disebut `web-app` pada setiap citra yang diawali dengan namespace tim. Konfigurasi ini memungkinkan citra-citra pada setiap tim ini untuk digunakan secara bersamaan tanpa gangguan. Citra Tim A adalah `team-a/web-app`, dan citra Tim B adalah `team-b/web-app`.

- Citra Anda dapat direplikasi ke repositori lain di seluruh Wilayah di registri Anda sendiri dan di seluruh akun. Anda dapat melakukan ini dengan menentukan konfigurasi replikasi dalam pengaturan registri Anda. Untuk informasi selengkapnya, lihat [Pengaturan registri pribadi](#).

Membuat repositori pribadi

Gambar kontainer Anda disimpan di repositori Amazon ECR. Gunakan langkah-langkah berikut untuk membuat repositori pribadi menggunakan file. AWS Management Console Untuk langkah-langkah untuk membuat repositori menggunakan AWS CLI, lihat [Langkah 3: Buat repositori](#)

Untuk membuat repositori () AWS Management Console

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah untuk membuat repositori Anda.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih tab Private, dan kemudian pilih Create repository.
5. Untuk pengaturan Visibilitas, verifikasi bahwa Private dipilih.
6. Untuk Nama repositori, masukkan nama yang unik untuk repositori Anda. Nama repositori dapat ditentukan sendiri (misalnya `nginx-web-app`). Atau, dapat ditambahkan dengan namespace untuk mengelompokkan repositori ke dalam suatu kategori (misalnya `project-a/nginx-web-app`).

Note

Nama repositori dapat berisi karakter maksimum. 256 Nama harus dimulai dengan huruf dan hanya dapat berisi huruf kecil, angka, tanda hubung, garis bawah, titik dan garis miring ke depan. Menggunakan tanda hubung ganda, garis bawah ganda, atau garis miring ganda tidak didukung.

7. Untuk Ketetapan tanda, pilih pengaturan ketetapan tanda untuk repositori. Repositori yang dikonfigurasi dengan tanda tetap mencegah tanda citra ditimpa. Untuk informasi selengkapnya, lihat [Tanda ketetapan citra](#).
8. Untuk Scan on push, sementara Anda dapat menentukan pengaturan pemindaian di tingkat repositori untuk pemindaian dasar, praktik terbaik adalah menentukan konfigurasi pemindaian di tingkat registri pribadi. Tentukan pengaturan pemindaian di registri pribadi memungkinkan Anda mengaktifkan pemindaian yang ditingkatkan atau pemindaian dasar serta menentukan

- filter untuk menentukan repositori mana yang dipindai. Untuk informasi selengkapnya, lihat [Pemindaian citra](#).
9. Untuk Enkripsi KMS, pilih apakah akan mengaktifkan enkripsi citra dalam repositori menggunakan AWS Key Management Service. Secara default, ketika enkripsi KMS diaktifkan, Amazon ECR menggunakan Kunci yang dikelola AWS (kunci KMS) dengan alias `aws/ecr`. Kunci ini dibuat di akun Anda saat pertama kali Anda membuat repositori dengan enkripsi KMS diaktifkan. Untuk informasi selengkapnya, lihat [Enkripsi diam](#).
 10. Saat enkripsi KMS diaktifkan, pilih Pengaturan enkripsi pelanggan (lanjutan) untuk memilih kunci KMS Anda sendiri. Kunci KMS harus berada di Wilayah yang sama dengan kluster. Pilih Buat kunci AWS KMS untuk menavigasi ke konsol AWS KMS untuk membuat kunci Anda sendiri.
 11. Pilih Buat repositori.
 12. (Opsional) Pilih repositori yang Anda buat dan pilih Lihat perintah dorongan untuk melihat langkah-langkah untuk mendorong citra ke repositori baru Anda. Untuk informasi selengkapnya tentang mendorong gambar ke repositori Anda, lihat [Mendorong citra](#)

Melihat detail repositori pribadi

Setelah Anda membuat repositori, Anda dapat melihat detail tentang repositori di: AWS Management Console

- Citra mana yang disimpan dalam repositori
- Detail tentang setiap gambar yang disimpan dalam repositori, termasuk ukuran dan intisari SHA untuk setiap gambar
- Frekuensi pemindaian yang ditentukan untuk isi repositori
- Apakah repositori memiliki aturan pull through cache aktif yang terkait dengannya
- Pengaturan enkripsi untuk repositori

Note

Dimulai dengan Docker versi 1.9, client Docker mengompresi lapisan citra sebelum mendorongnya ke registri V2 Docker. Output dari perintah `docker images` menunjukkan ukuran citra yang tidak terkompresi. Oleh karena itu, perlu diingat bahwa Docker mungkin mengembalikan gambar yang lebih besar daripada citra yang ditampilkan dalam AWS Management Console.

Untuk melihat informasi repositori (AWS Management Console)

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah yang berisi repositori untuk dilihat.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih tab Private dan kemudian repositori untuk dilihat.
5. Pada halaman detail repositori, konsol default ke tampilan Gambar. Gunakan menu navigasi untuk melihat informasi lain tentang repositori.

- Pilih Ringkasan untuk melihat detail repositori dan tarik data hitungan untuk repositori.
- Pilih Gambar untuk melihat informasi tentang tag gambar di repositori. Untuk melihat informasi lebih lanjut tentang gambar, pilih tag gambar. Untuk informasi selengkapnya, lihat [Melihat detail citra](#).

Jika ada citra yang tidak ditandai yang ingin Anda hapus, Anda dapat memilih kotak di sebelah kiri repositori untuk menghapus dan pilih Hapus. Untuk informasi selengkapnya, lihat [Menghapus citra](#).

- Pilih Izin untuk melihat kebijakan repositori yang diterapkan ke repositori. Untuk informasi selengkapnya, lihat [Kebijakan repositori pribadi](#).
- Pilih Kebijakan siklus hidup untuk melihat aturan kebijakan siklus hidup yang diterapkan ke repositori. Histori peristiwa siklus hidup juga dilihat di sini. Untuk informasi selengkapnya, lihat [Kebijakan siklus hidup](#).
- Pilih Tanda untuk melihat tanda metadata yang diterapkan ke repositori.

Mengedit repositori pribadi

Repositori yang ada dapat diedit untuk mengubah ketentuan tanda pada citra dan pengaturan pemindaian citra.

Untuk mengedit repositori () AWS Management Console

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah yang berisi repositori untuk diedit.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih tab Private dan kemudian pilih repositori yang akan diedit dan pilih Edit.

5. Untuk Ketetapan tanda, pilih pengaturan ketetapan tanda untuk repositori. Repositori yang dikonfigurasi dengan tanda tetap mencegah tanda citra ditimpa. Untuk informasi selengkapnya, lihat [Tanda ketetapan citra](#).
6. Untuk pengaturan pemindaian Gambar, sementara Anda dapat menentukan pengaturan pemindaian di tingkat repositori untuk pemindaian dasar, praktik terbaik adalah menentukan konfigurasi pemindaian di tingkat registri pribadi. Tentukan pengaturan pemindaian di registri pribadi memungkinkan Anda mengaktifkan pemindaian yang ditingkatkan atau pemindaian dasar serta menentukan filter untuk menentukan repositori mana yang dipindai. Untuk informasi selengkapnya, lihat [Pemindaian citra](#).
7. Untuk pengaturan Enkripsi, ini adalah bidang tampilan saja karena pengaturan enkripsi untuk repositori tidak dapat diubah setelah repositori dibuat.
8. Pilih Simpan untuk memperbarui pengaturan repositori.

Menghapus repositori pribadi

Jika Anda selesai menggunakan repositori, Anda dapat menghapusnya. Ketika Anda menghapus repositori di AWS Management Console, semua citra yang ada dalam repositori juga dihapus; ini tidak dapat dibatalkan.

Untuk menghapus repositori () AWS Management Console

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah yang berisi repositori untuk dihapus.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih tab Private dan kemudian pilih repositori yang akan dihapus dan pilih Delete.
5. Di jendela Hapus **nama_repositori**, verifikasi bahwa repositori yang dipilih akan dihapus dan pilih Hapus.

Important

Semua citra dalam repositori yang dipilih juga dihapus.

Kebijakan repositori pribadi

Amazon ECR menggunakan izin berbasis sumber daya untuk mengontrol akses ke repositori. Izin berbasis sumber daya memungkinkan Anda menentukan pengguna atau peran mana yang memiliki akses ke repositori dan tindakan apa yang dapat mereka lakukan padanya. Secara default, hanya AWS akun yang membuat repositori yang memiliki akses ke repositori. Anda dapat menerapkan dokumen kebijakan yang memungkinkan izin tambahan ke repositori Anda.

Topik

- [Kebijakan repositori vs kebijakan IAM](#)
- [Menyetel pernyataan kebijakan repositori pribadi](#)
- [Menghapus pernyataan kebijakan repositori pribadi](#)
- [Contoh kebijakan repositori pribadi](#)

Kebijakan repositori vs kebijakan IAM

Kebijakan repositori Amazon ECR adalah bagian dari kebijakan IAM yang memiliki lingkup untuk, dan secara khusus digunakan untuk, mengendalikan akses ke repositori Amazon ECR individu. Kebijakan IAM umumnya digunakan untuk menerapkan izin untuk seluruh layanan Amazon ECR tetapi juga dapat digunakan untuk mengontrol akses ke sumber daya tertentu.

Kebijakan repositori Amazon ECR dan kebijakan IAM digunakan saat menentukan tindakan yang mungkin dilakukan pengguna atau peran tertentu pada repositori. Jika pengguna atau peran diperbolehkan untuk melakukan tindakan melalui kebijakan repositori tetapi tidak diberi izin melalui kebijakan IAM (atau sebaliknya) maka tindakan akan ditolak. Agar tindakan diizinkan, pengguna atau peran hanya perlu diberi izin untuk tindakan baik melalui kebijakan repositori atau kebijakan IAM, tetapi tidak keduanya.

Important

Amazon ECR mengharuskan pengguna memiliki izin untuk melakukan panggilan ke API `ecr:GetAuthorizationToken` melalui kebijakan IAM sebelum mereka dapat mengautentikasi ke registri dan mendorong atau menarik citra dari repositori Amazon ECR. Amazon ECR menyediakan beberapa kebijakan yang dikelola untuk mengontrol akses pengguna pada berbagai tingkat; untuk informasi lebih lanjut, lihat [Contoh kebijakan berbasis identitas Amazon Elastic Container Registry](#).

Anda dapat menggunakan salah satu jenis kebijakan ini untuk mengontrol akses ke repositori Anda, seperti yang ditampilkan dalam contoh berikut.

Contoh ini menunjukkan kebijakan repositori Amazon ECR, yang memungkinkan pengguna tertentu untuk mendeskripsikan repositori dan gambar dalam repositori.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ECRRepositoryPolicy",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::account-id:user/username"},
      "Action": [
        "ecr:DescribeImages",
        "ecr:DescribeRepositories"
      ]
    }
  ]
}
```

Contoh ini menunjukkan kebijakan IAM yang mencapai tujuan yang sama seperti di atas, dengan lingkup kebijakan pada repositori (ditentukan oleh ARN penuh repositori) menggunakan parameter sumber daya. Untuk informasi lebih lanjut tentang format Amazon Resource Name (ARN), lihat [Sumber daya](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDescribeRepoImage",
      "Effect": "Allow",
      "Action": [
        "ecr:DescribeImages",
        "ecr:DescribeRepositories"
      ],
      "Resource": ["arn:aws:ecr:region:account-id:repository/repository-name"]
    }
  ]
}
```

Menyetel pernyataan kebijakan repositori pribadi

Anda dapat menambahkan pernyataan kebijakan akses ke repositori di AWS Management Console dengan mengikuti langkah-langkah di bawah ini. Anda dapat menambahkan beberapa pernyataan kebijakan per repositori. Untuk contoh kebijakan, lihat [Contoh kebijakan repositori pribadi](#).

Important

Amazon ECR mengharuskan pengguna memiliki izin untuk melakukan panggilan ke API `ecr:GetAuthorizationToken` melalui kebijakan IAM sebelum mereka dapat mengautentikasi ke registri dan mendorong atau menarik citra dari repositori Amazon ECR. Amazon ECR menyediakan beberapa kebijakan yang dikelola untuk mengontrol akses pengguna pada berbagai tingkat; untuk informasi lebih lanjut, lihat [Contoh kebijakan berbasis identitas Amazon Elastic Container Registry](#).

Untuk menetapkan pernyataan kebijakan repositori

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah yang berisi repositori yang akan ditetapkan pernyataan kebijakan.
3. Di panel navigasi, pilih Repositori.
4. Pada Repositori, pilih repositori yang akan ditetapkan pernyataan kebijakan untuk melihat isi repositori.
5. Dari tampilan daftar citra repositori, di panel navigasi, pilih Izin, Edit.

Note

Jika Anda tidak melihat opsi Izin di panel navigasi, pastikan bahwa Anda berada di tampilan daftar citra repositori.

6. Pada halaman Edit izin, pilih Tambah pernyataan.
7. Untuk Nama pernyataan, masukkan nama untuk pernyataan tersebut.
8. Untuk Efek, pilih apakah pernyataan kebijakan tersebut akan menghasilkan izin atau penolakan eksplisit.
9. Untuk Utama, pilih ruang lingkup untuk penerapan pernyataan kebijakan. Untuk informasi lebih lanjut, lihat [Elemen Kebijakan JSON AWS: Utama](#) dalam Panduan Pengguna IAM.

- Anda dapat menerapkan pernyataan untuk semua pengguna AWS yang sudah diautentikasi dengan memilih kotak centang Semua orang (*).
- Untuk Prinsipal layanan, tentukan nama prinsipal layanan (misalnya, `ecs.amazonaws.com`) untuk menerapkan pernyataan ke layanan tertentu.
- Untuk ID Akun AWS, tentukan nomor akun AWS (misalnya, 111122223333) untuk menerapkan pernyataan untuk semua pengguna di bawah akun AWS tertentu. Beberapa akun dapat ditentukan dengan menggunakan daftar yang dipisahkan koma.

Important

Akun yang Anda berikan izin harus memiliki Wilayah yang Anda beri kebijakan repositori yang diaktifkan, jika tidak, kesalahan akan terjadi.

- Untuk Entitas IAM, pilih peran atau pengguna dengan akun AWS Anda untuk diterapkan pernyataan.

Note

Untuk kebijakan repositori yang lebih rumit yang saat ini tidak didukung di AWS Management Console, Anda dapat menerapkan kebijakan dengan perintah [set-repository-policy](#) AWS CLI.

10. Untuk Tindakan, pilih ruang lingkup operasi API Amazon ECR yang akan diterapkan pernyataan kebijakan dari daftar operasi API individu.
11. Setelah selesai, pilih Simpan untuk menetapkan kebijakan.
12. Ulangi langkah sebelumnya untuk setiap kebijakan repositori yang ditambahkan.

Menghapus pernyataan kebijakan repositori pribadi

Jika Anda tidak lagi menginginkan pernyataan kebijakan repositori yang ada untuk diterapkan ke repositori, Anda dapat menghapusnya.

Untuk menghapus pernyataan kebijakan repositori

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.

2. Dari bilah navigasi, pilih Wilayah yang berisi repositori yang akan dihapus pernyataan kebijakannya.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih repositori yang akan dihapus pernyataan kebijakannya.
5. Di panel navigasi, pilih Izin, Edit.
6. Pada halaman Edit izin, pilih Tambah pernyataan.

Contoh kebijakan repositori pribadi

Important

Contoh kebijakan repositori di halaman ini dimaksudkan untuk diterapkan ke repositori pribadi Amazon ECR. Mereka tidak akan berfungsi dengan baik jika digunakan dengan prinsipal IAM secara langsung kecuali dimodifikasi untuk menentukan repositori Amazon ECR sebagai sumber daya. Untuk informasi selengkapnya tentang pengaturan kebijakan repositori, lihat.

[Menyetel pernyataan kebijakan repositori pribadi](#)

Kebijakan repositori Amazon ECR adalah bagian dari kebijakan IAM yang memiliki lingkup untuk, dan secara khusus digunakan untuk, mengendalikan akses ke repositori Amazon ECR individu. Kebijakan IAM umumnya digunakan untuk menerapkan izin untuk seluruh layanan Amazon ECR tetapi juga dapat digunakan untuk mengontrol akses ke sumber daya tertentu. Untuk informasi selengkapnya, lihat [Kebijakan repositori vs kebijakan IAM](#).

Contoh kebijakan repositori berikut menunjukkan pernyataan izin yang dapat Anda gunakan untuk mengontrol akses ke repositori pribadi Amazon ECR Anda.

Important

Amazon ECR mengharuskan pengguna memiliki izin untuk melakukan panggilan ke API `ecr:GetAuthorizationToken` melalui kebijakan IAM sebelum mereka dapat melakukan autentikasi ke registrasi dan mendorong atau menarik citra dari repositori Amazon ECR. Amazon ECR menyediakan beberapa kebijakan IAM terkelola untuk mengontrol akses pengguna pada berbagai tingkat; untuk informasi lebih lanjut, lihat [Contoh kebijakan berbasis identitas Amazon Elastic Container Registry](#).

Contoh: Izinkan satu atau lebih pengguna

Kebijakan repositori berikut memungkinkan satu atau lebih pengguna untuk mendorong dan menarik gambar ke dan dari repositori.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPushPull",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::account-id:user/push-pull-user-1",
          "arn:aws:iam::account-id:user/push-pull-user-2"
        ]
      },
      "Action": [
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability",
        "ecr:CompleteLayerUpload",
        "ecr:GetDownloadUrlForLayer",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ]
    }
  ]
}
```

Contoh: Izinkan akun lain

Kebijakan repositori berikut memungkinkan akun tertentu untuk mendorong citra.

Important

Akun yang Anda berikan izin harus memiliki Wilayah yang Anda beri kebijakan repositori yang diaktifkan, jika tidak, kesalahan akan terjadi.

```
{
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Sid": "AllowCrossAccountPush",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::account-id:root"
    },
    "Action": [
      "ecr:BatchCheckLayerAvailability",
      "ecr:CompleteLayerUpload",
      "ecr:InitiateLayerUpload",
      "ecr:PutImage",
      "ecr:UploadLayerPart"
    ]
  }
]
}

```

Kebijakan repositori berikut memungkinkan beberapa pengguna untuk menarik gambar (pull-user-1 dan pull-user-2) sambil memberikan akses penuh ke yang lain (admin-user).

Note

Untuk kebijakan repositori yang lebih rumit yang saat ini tidak didukung di AWS Management Console, Anda dapat menerapkan kebijakan dengan perintah [set-repository-policy](#) AWS CLI.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPull",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::account-id:user/pull-user-1",
          "arn:aws:iam::account-id:user/pull-user-2"
        ]
      },
      "Action": [
        "ecr:BatchGetImage",

```

```

        "ecr:GetDownloadUrlForLayer"
    ],
},
{
    "Sid": "AllowAll",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::account-id:user/admin-user"
    },
    "Action": [
        "ecr:*"
    ]
}
]
}

```

Contoh: Tolak semua

Kebijakan repositori berikut menolak semua pengguna di semua akun untuk menarik citra.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyPull",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ]
        }
    ]
}

```

Contoh: Membatasi akses ke alamat IP tertentu

Contoh berikut menolak izin kepada pengguna mana pun untuk melakukan operasi ECR Amazon apa pun saat diterapkan ke repositori dari rentang alamat tertentu.

Kondisi dalam pernyataan ini mengidentifikasi 54.240.143.* kisaran alamat IP Internet Protocol versi 4 (IPv4) yang diizinkan.

ConditionBlok menggunakan NotIpAddress kondisi dan kunci `aws:SourceIp` kondisi, yang merupakan kunci kondisi AWS -wide. Untuk informasi lebih lanjut tentang kunci syarat ini, lihat [AWS Kunci Konteks Syarat Global](#). Nilai IPv4 `aws:sourceIp` menggunakan notasi CIDR standar. Untuk informasi lebih lanjut, lihat [Operator Syarat Alamat IP](#) dalam Panduan Pengguna IAM.

```
{
  "Version": "2012-10-17",
  "Id": "ECRPolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "ecr:*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": "54.240.143.0/24"
        }
      }
    }
  ]
}
```

Contoh: Izinkan AWS layanan

Kebijakan repositori berikut memungkinkan akses AWS CodeBuild ke tindakan API Amazon ECR yang diperlukan untuk integrasi dengan layanan tersebut. Saat menggunakan contoh berikut, Anda harus menggunakan kunci `aws:SourceArn` dan `aws:SourceAccount` kondisi untuk cakupan sumber daya mana yang dapat mengasumsikan izin ini. Untuk informasi selengkapnya, lihat [contoh Amazon ECR CodeBuild](#) di Panduan AWS CodeBuild Pengguna.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": [
        "ecr:BatchGetImage",

```



```
        "ecr:GetDownloadUrlForLayer"
    ],
    "Condition":{
        "ArnLike":{
            "aws:SourceArn":"arn:aws:codebuild:region:123456789012:project/project-
name"
        },
        "StringEquals":{
            "aws:SourceAccount":"123456789012"
        }
    }
}
]
```

Menandai repositori pribadi

Untuk membantu Anda mengelola repositori Amazon ECR, Anda dapat menetapkan metadata Anda sendiri ke setiap repositori dalam bentuk tag sumber daya. AWS Topik ini menjelaskan tag AWS sumber daya dan menunjukkan cara membuatnya.

Dasar tanda

Tanda merupakan sebuah label yang Anda tetapkan ke sebuah sumber daya AWS. Setiap tag terdiri atas sebuah kunci dan sebuah nilai, yang keduanya Anda tentukan.

Tanda memungkinkan Anda untuk mengategorikan sumber daya AWS Anda dengan berbagai cara, misalnya, berdasarkan tujuan, pemilik, atau lingkungan. Hal ini berguna ketika Anda memiliki banyak sumber daya dengan jenis yang sama—Anda dapat dengan cepat mengidentifikasi sumber daya tertentu berdasarkan tanda yang telah Anda tetapkan. Misalnya, Anda dapat menentukan sekumpulan tag untuk repositori Amazon ECR akun Anda yang membantu Anda melacak pemilik setiap repositori.

Kami menyarankan agar Anda merancang serangkaian kunci tanda yang memenuhi kebutuhan Anda. Penggunaan serangkaian kunci tanda akan mempermudah Anda dalam mengelola sumber daya Anda. Anda dapat mencari dan memfilter sumber daya berdasarkan tanda yang Anda tambahkan.

Tanda tidak memiliki makna semantik bagi Amazon ECR dan diterjemahkan sebagai serangkaian karakter saja. Selain itu, tanda tidak secara otomatis ditetapkan ke sumber daya Anda. Anda dapat

mengedit kunci dan nilai tanda, dan Anda dapat membuang tanda dari sumber daya kapan saja. Anda dapat mengatur nilai tanda menjadi sebuah string kosong, tetapi Anda tidak dapat mengatur nilai tanda menjadi nol. Jika Anda menambahkan tag yang memiliki kunci yang sama dengan tag yang ada pada sumber daya tersebut, nilai yang baru akan menimpa nilai yang lama. Jika Anda menghapus sebuah sumber daya, semua tanda untuk sumber daya tersebut juga dihapus.

Anda dapat bekerja dengan tanda menggunakan AWS Management Console, AWS CLI, dan API Amazon ECR.

Jika Anda menggunakan AWS Identity and Access Management (IAM), Anda dapat mengontrol pengguna mana dalam akun AWS Anda yang memiliki izin untuk membuat, mengedit, atau menghapus tanda.

Penandaan sumber daya Anda

Anda dapat menandai repositori Amazon ECR baru atau yang sudah ada.

Jika Anda menggunakan konsol Amazon ECR, Anda dapat menerapkan tanda ke sumber daya baru ketika dibuat atau sumber daya yang ada dengan menggunakan opsi Tanda pada panel navigasi kapan saja.

Jika Anda menggunakan Amazon ECR API, atau AWS SDK/AWS CLI, Anda dapat menerapkan tag ke repositori baru menggunakan `tags` parameter pada tindakan API atau menggunakan tindakan `CreateRepository` API untuk menerapkan tag `TagResource` ke sumber daya yang ada. Untuk informasi lebih lanjut, lihat [TagResource](#).

Selain itu, jika tanda tidak dapat diterapkan selama pembuatan repositori, kami akan mengulang kembali proses pembuatan repositori. Hal ini untuk memastikan bahwa sumber daya dibuat dengan tanda atau tidak dibuat sama sekali, dan tidak akan ada sumber daya yang dibiarkan tidak ditandai. Dengan melakukan penandaan sumber daya pada saat pembuatan, Anda dapat menghilangkan kebutuhan untuk menjalankan skrip penandaan kustom setelah pembuatan sumber daya.

Batasan tanda

Batasan dasar berikut berlaku untuk tanda:

- Jumlah maksimum tanda per repositori — 50
- Untuk setiap sumber daya, setiap kunci tanda harus unik, dan setiap kunci tanda hanya dapat memiliki satu nilai.

- Panjang kunci maksimum – 128 karakter Unicode dalam UTF-8
- Panjang nilai maksimum – 256 karakter Unicode dalam UTF-8
- Jika skema penandaan Anda digunakan di beberapa layanan dan sumber daya, ingatlah bahwa layanan lain mungkin memiliki pembatasan pada karakter yang diizinkan. Karakter-karakter yang diizinkan secara umum adalah: huruf, angka, dan spasi yang dapat diwakili dalam UTF-8, beserta karakter-karakter berikut: + - = . _ : / @.
- Kunci dan nilai tanda bersifat peka terhadap huruf besar dan kecil.
- Jangan gunakan prefiks `aws :` untuk kunci ataupun nilai; itu disediakan untuk penggunaan AWS. Anda tidak dapat mengedit atau menghapus kunci atau nilai tanda dengan prefiks ini. Tanda dengan prefiks ini tidak dihitung, berlawanan dengan tanda milik Anda per batas sumber daya.

Penandaan sumber daya Anda untuk penagihan

Tanda yang Anda tambahkan ke repositori Amazon ECR Anda sangat membantu ketika meninjau alokasi biaya setelah tanda diaktifkan dalam Laporan Biaya & Penggunaan Anda. Untuk informasi selengkapnya, lihat [Laporan penggunaan Amazon ECR](#).

Untuk melihat biaya sumber daya gabungan, Anda dapat mengatur informasi penagihan berdasarkan sumber daya yang memiliki nilai kunci tanda yang sama. Misalnya, Anda dapat memberi tanda pada beberapa sumber daya dengan nama aplikasi tertentu, lalu organisir informasi penagihan Anda untuk melihat total biaya aplikasi tersebut pada beberapa layanan. Untuk informasi selengkapnya tentang pengaturan laporan alokasi biaya dengan tanda, lihat [Laporan Alokasi Biaya Bulanan](#) dalam Panduan Pengguna AWS Billing.

Note

Jika Anda baru saja mengaktifkan pelaporan, data untuk bulan yang berjalan dapat dilihat setelah 24 jam.

Cara menggunakan tanda dengan menggunakan konsol

Dengan menggunakan konsol Amazon ECR, Anda dapat mengelola tanda yang terkait dengan repositori baru atau yang sudah ada.

Bila Anda memilih repositori tertentu di konsol Amazon ECR, Anda dapat melihat tanda dengan memilih Tanda di panel navigasi.

Untuk menambahkan tag ke repositori () AWS Management Console

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah untuk digunakan.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih kotak centang di sebelah repositori yang ingin Anda tandai.
5. Dari menu Tindakan, pilih Tag repositori.
6. Pada halaman Repository tags, pilih Add tags, Add tag.
7. Pada halaman Edit tag repositori, tentukan kunci dan nilai untuk setiap tag, lalu pilih Simpan.

Untuk menghapus tag dari sumber daya individu (AWS Management Console)

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah untuk digunakan.
3. Pada halaman Repositori, pilih kotak centang di sebelah repositori tempat Anda ingin menghapus tag.
4. Dari menu Tindakan, pilih Tag repositori.
5. Pada halaman Repository tags, pilih Edit.
6. Pada halaman Edit tag repositori, pilih Hapus untuk setiap tag yang ingin Anda hapus, lalu pilih Simpan.

Cara menggunakan tanda dengan menggunakan AWS CLI atau API

Gunakan yang berikut ini untuk menambahkan, memperbarui, membuat daftar, dan menghapus tanda untuk sumber daya Anda. Dokumentasi yang sesuai memberikan contoh.

Dukungan pemberian tag untuk sumber daya Amazon ECR

Tugas	AWS CLI	Tindakan API
Tambahkan atau timpa satu atau beberapa tanda.	tag-sumber	TagResource
Hapus satu atau beberapa tanda.	untag-sumber	UntagResource

Contoh berikut menunjukkan cara mengelola tanda dengan menggunakan AWS CLI.

Contoh 1: Beri tanda pada repositori yang ada

Perintah berikut memberi tanda pada repositori yang ada.

```
aws ecr tag-resource \  
  --resource-arn arn:aws:ecr:region:account_id:repository/repository_name \  
  --tags Key=stack,Value=dev
```

Contoh 2: Beri tanda pada repositori yang ada dengan beberapa tanda

Perintah berikut memberi tanda pada repositori yang ada.

```
aws ecr tag-resource \  
  --resource-arn arn:aws:ecr:region:account_id:repository/repository_name \  
  --tags Key=key1,Value=value1 Key=key2,Value=value2 Key=key3,Value=value3
```

Contoh 3: Hapus tanda dari repositori yang ada

Perintah berikut menghapus tanda dari repositori yang ada.

```
aws ecr untag-resource \  
  --resource-arn arn:aws:ecr:region:account_id:repository/repository_name \  
  --tag-keys tag_key
```

Contoh 4: Buat daftar tanda untuk repositori

Perintah berikut membuat daftar tanda yang terkait dengan repositori yang ada.

```
aws ecr list-tags-for-resource \  
  --resource-arn arn:aws:ecr:region:account_id:repository/repository_name
```

Contoh 5: Buat repositori dan terapkan tanda

Perintah berikut membuat repositori bernama test-repo dan menambahkan tanda dengan kunci team dan nilai devs.

```
aws ecr create-repository \  
  --repository-name test-repo \  
  --tags Key=team,Value=devs
```

```
--tags Key=team,Value=devs
```

Citra privat

Amazon Elastic Container Registry (Amazon ECR) menyimpan gambar Docker, gambar Open Container Initiative (OCI), dan artefak yang kompatibel dengan OCI di repositori pribadi. Anda dapat menggunakan CLI Docker, atau klien pilihan Anda, untuk mendorong dan menarik gambar ke dan dari repositori Anda.

Topik

- [Mendorong citra](#)
- [Menandatangani gambar](#)
- [Melihat detail citra](#)
- [Menarik citra](#)
- [Menggunakan aturan pull through cache](#)
- [Menghapus citra](#)
- [Menandai ulang citra](#)
- [Replikasi citra pribadi](#)
- [Kebijakan siklus hidup](#)
- [Tanda ketetapan citra](#)
- [Pemindaian citra](#)
- [Format manifes citra kontainer](#)
- [Menggunakan citra Amazon ECR dengan Amazon ECS](#)
- [Menggunakan Citra Amazon ECR dengan Amazon EKS](#)
- [Citra kontainer Amazon Linux](#)

Mendorong citra

Anda dapat mendorong gambar Docker, daftar manifes, dan gambar Open Container Initiative (OCI) dan artefak yang kompatibel ke repositori pribadi Anda. Halaman berikut mendeskripsikan secara lebih mendetail.

Amazon ECR juga menyediakan cara untuk mereplikasi gambar Anda ke repositori lain, di seluruh Wilayah di registri Anda sendiri dan di berbagai akun, dengan menentukan konfigurasi replikasi di pengaturan registri pribadi Anda. Untuk informasi selengkapnya, lihat [Pengaturan registri pribadi](#).

Topik

- [Izin IAM yang diperlukan untuk mendorong gambar](#)
- [Mendorong gambar Docker](#)
- [Mendorong citra multi-arsitektur](#)
- [Mendorong grafik Helm](#)

Izin IAM yang diperlukan untuk mendorong gambar

Amazon ECR mengharuskan pengguna memiliki izin berikut untuk mendorong gambar. Mengikuti praktik terbaik pemberian hak istimewa paling sedikit, Anda dapat memasukkan izin ini ke repositori tertentu atau Anda dapat memberikan izin untuk semua repositori. Pengguna harus mengautentikasi ke setiap registri Amazon ECR yang ingin mereka dorong gambar dengan meminta token otorisasi. Amazon ECR menyediakan beberapa kebijakan IAM terkelola untuk mengontrol akses pengguna pada berbagai tingkat; untuk informasi lebih lanjut, lihat [Contoh kebijakan berbasis identitas Amazon Elastic Container Registry](#).

Kebijakan IAM berikut memberikan izin yang diperlukan untuk mendorong gambar tanpa pelingkupan ke repositori tertentu.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:CompleteLayerUpload",
        "ecr:GetAuthorizationToken",
        "ecr:UploadLayerPart",
        "ecr:InitiateLayerUpload",
        "ecr:BatchCheckLayerAvailability",
        "ecr:PutImage"
      ],
      "Resource": "*"
    }
  ]
}
```

Kebijakan IAM berikut memberikan izin yang diperlukan untuk mendorong gambar dan cakupan ke repositori tertentu. Repositori harus ditentukan sebagai Nama Sumber Daya Amazon (ARN) lengkap.


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:CompleteLayerUpload",
        "ecr:UploadLayerPart",
        "ecr:InitiateLayerUpload",
        "ecr:BatchCheckLayerAvailability",
        "ecr:PutImage"
      ],
      "Resource": "arn:aws:ecr:region:111122223333:repository/repository-name"
    },
    {
      "Effect": "Allow",
      "Action": "ecr:GetAuthorizationToken",
      "Resource": "*"
    }
  ]
}
```

Mendorong gambar Docker

Anda dapat mendorong gambar kontainer Anda ke repositori Amazon ECR dengan perintah. `docker push` Amazon ECR juga dapat menciptakan dan mendorong daftar manifes Docker, yang digunakan untuk citra multi-arsitektur. Setiap citra yang direferensikan dalam daftar manifes harus sudah didorong ke repositori Anda. Untuk informasi selengkapnya, lihat [Mendorong citra multi-arsitektur](#).

Untuk mendorong gambar Docker ke repositori Amazon ECR

Repositori Amazon ECR harus tersedia sebelum Anda mendorong citra. Untuk informasi selengkapnya, lihat [the section called "Membuat repositori"](#).

1. Autentikasi klien Docker Anda ke registrasi Amazon ECR di mana Anda berniat untuk mendorong citra Anda. Token autentikasi harus diperoleh untuk setiap registrasi yang digunakan, dan token berlaku selama 12 jam. Untuk informasi selengkapnya, lihat [Autentikasi registri pribadi](#).

Untuk mengautentikasi Docker ke registri Amazon ECR, jalankan perintah. `aws ecr get-login-password` Ketika meneruskan token otorisasi ke perintah `docker login`, gunakan nilai AWS untuk

nama pengguna dan tentukan URI registri Amazon ECR yang ingin Anda autentikasi. Jika melakukan autentikasi untuk beberapa registri, Anda harus mengulangi perintah tersebut untuk setiap registri.

Important

Jika Anda menerima pesan kesalahan, instal atau upgrade ke versi terbaru AWS CLI. Untuk informasi selengkapnya, lihat [Menginstal AWS Command Line Interface](#) dalam Panduan Pengguna AWS Command Line Interface .

```
aws ecr get-login-password --region region | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

2. Jika repositori citra Anda tidak tersedia di registrasi yang ingin Anda dorong, maka buatlah. Untuk informasi selengkapnya, lihat [Membuat repositori pribadi](#).
3. Identifikasi gambar lokal untuk didorong. Jalankan docker images perintah untuk membuat daftar gambar kontainer di sistem Anda.

```
docker images
```

Anda dapat mengidentifikasi citra dengan *repositori:tag* nilai atau ID citra dalam output perintah yang dihasilkan.

4. Tandai citra Anda dengan registrasi Amazon ECR, repositori, dan kombinasi nama tanda citra opsional untuk digunakan. Format registrasi adalah *aws_account_id.dkr.ecr.us-west-2.amazonaws.com*. Nama repositori harus sesuai dengan repositori yang Anda buat untuk citra Anda. Jika Anda menghilangkan tanda citra, maka kami berasumsi bahwa tandanya adalah `latest`.

Contoh berikut menandai gambar lokal dengan ID *e9ae3c220b23* sebagai.

```
aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository:tag
```

```
docker tag e9ae3c220b23 aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository:tag
```

5. Mendorong citra menggunakan perintah docker push:

```
docker push aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository:tag
```

6. (Opsional) Terapkan tanda tambahan untuk citra Anda dan dorong tanda tersebut ke Amazon ECR dengan mengulangi [Step 4](#) dan [Step 5](#).

Mendorong citra multi-arsitektur

Amazon ECR dapat menciptakan dan mendorong daftar manifes Docker, yang digunakan untuk citra multi-arsitektur. Daftar manifes adalah daftar citra yang dibuat dengan menentukan satu atau lebih nama citra. Dalam kebanyakan kasus, daftar manifes dibuat dari citra yang memiliki fungsi yang sama tetapi untuk sistem operasi atau arsitektur yang berbeda. Daftar manifes yang tidak diperlukan. Untuk informasi selengkapnya, lihat [manifes docker](#).

Important

CLI Docker Anda harus mengaktifkan fitur eksperimental untuk menggunakan fitur ini. Untuk informasi selengkapnya, lihat [Fitur eksperimental](#).

Sebuah daftar manifes dapat ditarik atau direferensikan dalam definisi tugas Amazon ECS atau Amazon EKS pod spec seperti citra Amazon ECR lainnya.

Langkah-langkah berikut dapat dilakukan untuk membuat dan mendorong daftar manifes Docker ke repositori Amazon ECR. Anda harus sudah memiliki citra yang didorong ke repositori Anda untuk referensi dalam manifes Docker. Untuk informasi tentang cara mendorong citra, lihat [Mendorong gambar Docker](#).

Untuk mendorong gambar Docker multi-arsitektur ke repositori Amazon ECR

Repositori Amazon ECR harus tersedia sebelum Anda mendorong citra. Untuk informasi selengkapnya, lihat [the section called "Membuat repositori"](#).

1. Autentikasi klien Docker Anda ke registrasi Amazon ECR di mana Anda berniat untuk mendorong citra Anda. Token autentikasi harus diperoleh untuk setiap registrasi yang digunakan, dan token berlaku selama 12 jam. Untuk informasi selengkapnya, lihat [Autentikasi registri pribadi](#).

Untuk mengautentikasi Docker ke registri Amazon ECR, jalankan perintah `aws ecr get-login-password`. Ketika meneruskan token otorisasi ke perintah `docker login`, gunakan nilai AWS untuk nama pengguna dan tentukan URI registri Amazon ECR yang ingin Anda autentikasi. Jika melakukan autentikasi untuk beberapa registri, Anda harus mengulangi perintah tersebut untuk setiap registri.

⚠ Important

Jika Anda menerima pesan kesalahan, instal atau upgrade ke versi terbaru AWS CLI. Untuk informasi selengkapnya, lihat [Menginstal AWS Command Line Interface](#) dalam Panduan Pengguna AWS Command Line Interface .

```
aws ecr get-login-password --region region | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

2. Masukkan citra di repositori Anda, konfirmasi tanda citra.

```
aws ecr describe-images --repository-name my-repository
```

3. Buat daftar manifes Docker. Perintah `manifest create` memverifikasi bahwa citra yang direferensikan sudah berada di repositori Anda dan menciptakan manifes lokal.

```
docker manifest create aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository:image_one_tag aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository:image_two
```

4. (Opsional) Periksa daftar manifes Docker. Hal ini memungkinkan Anda untuk mengonfirmasi ukuran dan digest untuk setiap manifes citra yang direferensikan dalam daftar manifes.

```
docker manifest inspect aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository
```

5. Dorong daftar manifes Docker ke repositori Amazon ECR Anda.

```
docker manifest push aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository
```

Mendorong grafik Helm

Amazon ECR mendukung dorongan Open Container Initiative (OCI) artefak ke repositori Anda. Untuk menampilkan fungsionalitas ini, lakukan langkah-langkah berikut untuk mendorong grafik Helm ke Amazon ECR.

Untuk informasi lebih lanjut tentang menggunakan Amazon ECR yang menghost grafik Helm dengan Amazon EKS, lihat [Menginstal grafik Helm yang di-host Amazon ECR dengan Amazon EKS](#).

Untuk mendorong grafik Helm ke repositori Amazon ECR

1. Instal versi terbaru klien Helm. Langkah-langkah ini ditulis menggunakan versi 3.8.2 Helm. Untuk informasi selengkapnya, lihat [Helm](#).
2. Lakukan langkah-langkah berikut untuk membuat grafik tes Helm. Untuk informasi lebih lanjut, lihat [Helm Docs - Memulai](#).
 - a. Buat grafik Helm bernama `helm-test-chart` dan hapus isi direktori `templates`.

```
helm create helm-test-chart  
rm -rf ./helm-test-chart/templates/*
```

- b. Buat ConfigMap di `templates` folder.

```
cd helm-test-chart/templates  
cat <<EOF > configmap.yaml  
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: helm-test-chart-configmap  
data:  
  myvalue: "Hello World"  
EOF
```

3. Package grafik. Outputnya akan berisi nama file bagan paket yang Anda gunakan saat mendorong bagan Helm.

```
cd ../../  
helm package helm-test-chart
```

Output

```
Successfully packaged chart and saved it to: /Users/username/helm-test-chart-0.1.0.tgz
```

4. Buat repositori untuk menyimpan grafik Helm Anda. Nama repositori Anda harus sesuai dengan nama yang Anda gunakan saat membuat bagan Helm di langkah 2. Untuk informasi selengkapnya, lihat [Membuat repositori pribadi](#).

```
aws ecr create-repository \  
  --repository-name helm-test-chart \  
  --region us-west-2
```

5. Autentikasi Helm client Anda ke registrasi Amazon ECR di mana Anda berniat untuk mendorong citra Anda. Token autentikasi harus diperoleh untuk setiap registrasi yang digunakan, dan token berlaku selama 12 jam. Untuk informasi selengkapnya, lihat [Autentikasi registri pribadi](#).

```
aws ecr get-login-password \  
  --region us-west-2 | helm registry login \  
  --username AWS \  
  --password-stdin aws_account_id.dkr.ecr.us-west-2.amazonaws.com
```

6. Dorong bagan Helm menggunakan helm push perintah. Outputnya harus mencakup URI repositori Amazon ECR dan SHA digest.

```
helm push helm-test-chart-0.1.0.tgz oci://aws_account_id.dkr.ecr.us-west-2.amazonaws.com/
```

7. Deskripsikan grafik Helm Anda.

```
aws ecr describe-images \  
  --repository-name helm-test-chart \  
  --region us-west-2
```

Outputnya, verifikasi bahwa parameter `artifactMediaType` menunjukkan jenis artefak yang tepat.

```
{  
  "imageDetails": [  
    {  
      "registryId": "aws_account_id",  
      "repositoryName": "helm-test-chart",
```

```
    "imageDigest":
      "sha256:dd8aebdda7df991a0ffe0b3d6c0cf315fd582cd26f9755a347a52adEXAMPLE",
      "imageTags": [
        "0.1.0"
      ],
      "imageSizeInBytes": 1620,
      "imagePushedAt": "2021-09-23T11:39:30-05:00",
      "imageManifestMediaType": "application/vnd.oci.image.manifest.v1+json",
      "artifactMediaType": "application/vnd.cncf.helm.config.v1+json"
    }
  ]
}
```

8. (Opsional) Untuk langkah tambahan, instal helm configmap dan mulai dengan Amazon EKS. Untuk informasi selengkapnya, lihat [Menginstal grafik Helm yang di-host Amazon ECR dengan Amazon EKS](#).

Menandatangani gambar

Amazon ECR terintegrasi dengan AWS Signer untuk menyediakan cara bagi Anda untuk menandatangani gambar kontainer Anda. Anda dapat menyimpan gambar kontainer dan tanda tangan di repositori pribadi Anda.

Pertimbangan

Berikut ini harus dipertimbangkan saat menggunakan penandatanganan gambar Amazon ECR.

- Tanda tangan yang disimpan di repositori Anda dihitung terhadap kuota layanan untuk jumlah maksimum gambar per repositori. Untuk informasi selengkapnya, lihat [Kuota layanan Amazon ECR](#).
- Saat menggunakan kebijakan siklus hidup Amazon ECR, tindakan apa pun menurut aturan untuk kedaluwarsa atau menghapus indeks gambar OCI akan mengakibatkan Amazon ECR menghapus tanda tangan apa pun yang direferensikan oleh indeks gambar tersebut dalam waktu 24 jam.

Prasyarat

Sebelum Anda mulai, Prasyarat berikut harus dipenuhi.

- Instal dan konfigurasi versi terbaru dari file AWS CLI. Untuk informasi selengkapnya, lihat [Menginstal atau memperbarui versi terbaru](#) dari Panduan AWS Command Line Interface Pengguna. AWS CLI
- Instal CLI Notasi dan plugin untuk AWS Signer Notasi. Untuk informasi selengkapnya, lihat [Prasyarat untuk menandatangani gambar kontainer](#) di Panduan Pengembang.AWS Signer
- Miliki gambar kontainer yang disimpan di repositori pribadi Amazon ECR untuk ditandatangani. Untuk informasi selengkapnya, lihat [Mendorong citra](#).

Konfigurasi otentikasi untuk klien Notaris

Sebelum Anda dapat membuat tanda tangan menggunakan CLI Notasi, Anda harus mengonfigurasi klien sehingga dapat mengautentikasi ke Amazon ECR. Jika Anda menginstal Docker pada host yang sama tempat Anda menginstal klien Notation, maka Notation akan menggunakan kembali metode otentikasi yang sama yang Anda gunakan untuk klien Docker. Docker login dan logout perintah akan memungkinkan Notasi sign dan verify perintah untuk menggunakan kredensial yang sama, dan Anda tidak perlu mengautentikasi Notasi secara terpisah. Untuk informasi selengkapnya tentang mengonfigurasi klien Notasi Anda untuk otentikasi, lihat Mengautentikasi dengan pendaftar yang sesuai [dengan OCI di dokumentasi Proyek Notaris](#)

Jika Anda tidak menggunakan Docker atau alat lain yang menggunakan kredensial Docker, sebaiknya gunakan Amazon ECR Docker Credential Helper sebagai toko kredensial Anda. Untuk informasi selengkapnya tentang cara menginstal dan mengonfigurasi Amazon ECR Credential Helper, lihat [Amazon ECR Docker Credential Helper](#).

Menandatangani gambar

Langkah-langkah berikut dapat digunakan untuk membuat sumber daya yang diperlukan untuk menandatangani gambar kontainer dan menyimpan tanda tangan di repositori pribadi Amazon ECR. Notasi menandatangani gambar menggunakan intisari.

Untuk menandatangani gambar

1. Buat profil AWS Signer penandatanganan menggunakan platform Notation-OCI-SHA384-ECDSA penandatanganan. Anda dapat secara opsional menentukan periode validitas tanda tangan menggunakan parameter. `--signature-validity-period` Nilai ini dapat ditentukan menggunakan `DAYS`, `MONTHS`, atau `YEARS`. Jika tidak ada periode validitas yang ditentukan, nilai default 135 bulan digunakan.


```
aws signer put-signing-profile --profile-name ecr_signing_profile --platform-id
Notation-OCI-SHA384-ECDSA
```

Note

Nama profil penandatanganan hanya mendukung karakter alfanumerik dan garis bawah (`.`, `_`).

2. Otentikasi klien Notasi ke registri default Anda. Contoh berikut menggunakan AWS CLI untuk mengautentikasi CLI Notasi ke registri pribadi Amazon ECR.

```
aws ecr get-login-password --region region | notation login --username AWS --
password-stdin 111122223333.dkr.ecr.region.amazonaws.com
```

3. Gunakan CLI Notasi untuk menandatangani gambar, menentukan gambar menggunakan nama repositori dan intisari SHA. Ini membuat tanda tangan dan mendorongnya ke repositori pribadi Amazon ECR yang sama dengan gambar yang ditandatangani.

Dalam contoh berikut, kami menandatangani gambar di `curl` repositori dengan SHA digest. `sha256:ca78e5f730f9a789ef8c63bb55275ac12dfb9e8099e6EXAMPLE`

```
notation
sign 111122223333.dkr.ecr.region.amazonaws.com/
curl@sha256:ca78e5f730f9a789ef8c63bb55275ac12dfb9e8099e6EXAMPLE --plugin
"com.amazonaws.signer.notation.plugin" --id "arn:aws:signer:region:111122223333:/
signing-profiles/ecrSigningProfileName"
```

Verifikasi gambar secara lokal setelah penandatanganan

Setelah Anda menandatangani gambar kontainer menggunakan AWS Signer dan Notasi, Anda atau anggota resmi tim Anda dapat memverifikasi asal dan integritas gambar dengan cara kriptografi.

Selesaikan langkah-langkah berikut untuk memverifikasi bahwa gambar valid dengan Notasi.

Untuk memverifikasi gambar

1. Toko kepercayaan diperlukan untuk verifikasi. Jika Anda menggunakan penginstal untuk AWS Signer plugin dan Notasi, toko kepercayaan disiapkan secara otomatis dan disediakan dengan sertifikat root.
2. Siapkan kebijakan kepercayaan yang mirip dengan yang di bawah ini, modifikasi sesuai kebutuhan nama profil penandatanganan yang Anda gunakan untuk memverifikasi gambar.

```
{
  "version": "1.0",
  "trustPolicies": [
    {
      "name": "aws-signer-tp",
      "registryScopes": [
        "*"
      ],
      "signatureVerification": {
        "level": "strict"
      },
      "trustStores": [
        "signingAuthority:aws-signer-ts"
      ],
      "trustedIdentities": [
        "arn:aws:signer:region:111122223333:/signing-profiles/ecr_signing_profile",
        "arn:aws:signer:region:111122223333:/signing-profiles/ecr_signing_profile2"
      ]
    }
  ]
}
```

3. Impor kebijakan ke dalam Notasi.

```
$ notation policy import mypolicy.json
```

Output:

```
Existing trust policy configuration found, do you
want to overwrite it? [y/N] y
```

```
Trust policy configuration imported successfully.
```

4. Verifikasi tanda tangan, tentukan tanda tangan menggunakan nama repositori dan intisari SHA.

```
$ notation verify 111122223333.dkr.ecr.region.amazonaws.com/curl@SHA256_digest
```

Output:

```
Successfully verified signature for 111122223333.dkr.ecr.us-west-2.amazonaws.com/curl@SHA256_digest
```

Menghapus tanda tangan

Saat Anda membuat dan mendorong tanda tangan menggunakan CLI Notasi, indeks gambar OCI juga dibuat di repositori Amazon ECR Anda. Amazon ECR API tidak mendukung penghapusan artefak atau gambar yang dirujuk oleh indeks gambar OCI, jadi berikut ini adalah opsi yang tersedia untuk membersihkan artefak ini.

- (Disarankan) Anda dapat menggunakan CLI ORAS untuk menghapus artefak dan ORAS akan menangani pembaruan atau penghapusan indeks gambar.
- Anda dapat menggunakan Amazon ECR API atau konsol untuk menghapus indeks gambar OCI terlebih dahulu dan kemudian artefak yang direferensikan seperti tanda tangan.

Saat menggunakan klien ORAS untuk menghapus tanda tangan dan artefak tipe referensi lainnya, ORAS mengelola indeks gambar OCI. ORAS pertama-tama akan menghapus referensi ke artefak dari indeks, dan kemudian akan menghapus manifes. `oras manifest delete` Perintah dapat digunakan, merujuk indeks artefak tanda tangan. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi klien ORAS, lihat [Instalasi](#) di dokumentasi ORAS.

Contoh perintah berikut dapat digunakan untuk menghapus tanda tangan.

```
oras manifest  
delete 111122223333.dkr.ecr.region.amazonaws.com/  
repository_name@sha256:ca78e5f730f9a789ef8c63bb55275ac12dfb9e8099e6EXAMPLE
```

Melihat detail citra

Setelah Anda mendorong citra ke repositori Anda, maka Anda dapat melihat informasinya di AWS Management Console. Detail yang disertakan adalah sebagai berikut:

- URI citra
- Tanda citra
- Tipe media artefak
- Tipe manifes citra
- Status pemindaian
- Ukuran citra dalam MB
- Ketika citra telah didorong ke repositori
- Status replikasi

Untuk melihat detail citra (AWS Management Console)

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah yang berisi repositori berisi citra Anda.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih repositori yang akan dilihat.
5. Pada halaman Repositori: ***repository_name***, pilih citra untuk melihat detailnya.

Menarik citra

Jika Anda ingin menjalankan gambar Docker yang tersedia di Amazon ECR, Anda dapat menariknya ke lingkungan lokal Anda dengan perintah `docker pull`. Anda dapat melakukan ini dari registri default Anda atau dari registri yang terkait dengan AWS akun lain. Untuk menggunakan image Amazon ECR dalam definisi tugas Amazon ECS, lihat. [Menggunakan citra Amazon ECR dengan Amazon ECS](#)

Important

Amazon ECR mengharuskan pengguna memiliki izin untuk melakukan panggilan ke API `ecr:GetAuthorizationToken` melalui kebijakan IAM sebelum mereka dapat

mengautentikasi ke registrasi dan mendorong atau menarik citra dari repositori Amazon ECR. Amazon ECR menyediakan beberapa kebijakan IAM terkelola untuk mengontrol akses pengguna di berbagai tingkat; untuk informasi lebih lanjut, lihat [Contoh kebijakan berbasis identitas Amazon Elastic Container Registry](#).

Untuk menarik gambar Docker dari repositori Amazon ECR

1. Autentikasi klien Docker Anda ke registrasi Amazon ECR di mana Anda berniat untuk menarik citra Anda. Token autentikasi harus diperoleh untuk setiap registrasi yang digunakan, dan token berlaku selama 12 jam. Untuk informasi selengkapnya, lihat [Autentikasi registri pribadi](#).
2. (Opsional) Identifikasi citra untuk menarik.
 - Anda dapat memasukkan repositori di registrasi dengan perintah `aws ecr describe-repositories`:

```
aws ecr describe-repositories
```

Contoh registrasi di atas memiliki sebuah repositori bernama `amazonlinux`.

- Anda dapat mendeskripsikan citra dalam repositori dengan perintah `aws ecr describe-images`:

```
aws ecr describe-images --repository-name amazonlinux
```

Contoh repositori di atas memiliki citra yang ditandai sebagai `latest` dan `2016.09`, dengan digest

```
sha256:f1d4ae3f7261a72e98c6ebefe9985cf10a0ea5bd762585a43e0700ed99863807
```

citra.

3. Tarik citra menggunakan perintah `docker pull`. Format nama citra harus `registry/repository[:tag]` untuk menariknya dengan tanda atau `registry/repository[@digest]` untuk menariknya dengan digest.

```
docker pull aws_account_id.dkr.ecr.us-west-2.amazonaws.com/amazonlinux:latest
```

⚠ Important

Jika Anda menerima pesan kesalahan `repository-url not found: does not exist or no pull access`, Anda mungkin perlu mengautentikasi klien Docker Anda dengan Amazon ECR. Untuk informasi selengkapnya, lihat [Autentikasi registri pribadi](#).

Menggunakan aturan pull through cache

Dengan aturan pull through cache, Anda dapat menyinkronkan konten registri upstream dengan registri pribadi Amazon ECR Anda. Amazon ECR saat ini mendukung pembuatan aturan cache tarik untuk pendaftar hulu berikut.

- Docker Hub, Microsoft Azure Container Registry, dan GitHub Container Registry (Memerlukan otentikasi)
- Amazon ECR Public, registri image container Kubernetes, dan Quay (Tidak memerlukan autentikasi)

Untuk registrasi hulu yang memerlukan otentikasi, Anda harus menyimpan kredensialnya secara rahasia. AWS Secrets Manager Konsol Amazon ECR memudahkan Anda membuat rahasia Secrets Manager untuk setiap registri upstream yang diautentikasi. Untuk informasi selengkapnya tentang membuat rahasia Secrets Manager menggunakan konsol Secrets Manager, lihat [Menyimpan kredensi repositori upstream Anda secara rahasia AWS Secrets Manager](#).

Setelah Anda membuat aturan pull through cache untuk registri upstream, cukup tarik gambar dari registri upstream menggunakan URI registri pribadi Amazon ECR Anda. Amazon ECR kemudian membuat repositori dan menyimpan gambar itu di registri pribadi Anda. Pada permintaan tarik Anda berikutnya dari gambar yang di-cache dengan tag yang diberikan, Amazon ECR memeriksa registri hulu untuk melihat apakah ada versi baru gambar dengan tag spesifik tersebut dan mencoba memperbarui gambar di registri pribadi Anda setidaknya sekali setiap 24 jam.

Amazon ECR telah menambahkan dukungan untuk template pembuatan repositori, saat ini dalam pratinjau, yang memberi Anda kontrol untuk menentukan konfigurasi awal untuk repositori baru yang dibuat oleh Amazon ECR atas nama Anda menggunakan aturan pull through cache. Setiap template berisi awalan namespace repositori yang digunakan untuk mencocokkan repositori baru dengan template tertentu. Template dapat menentukan konfigurasi untuk semua pengaturan repositori

termasuk kebijakan akses berbasis sumber daya, kekekalan tag, enkripsi, dan kebijakan siklus hidup. Pengaturan dalam template pembuatan repositori hanya diterapkan selama pembuatan repositori dan tidak berpengaruh pada repositori atau repositori yang ada yang dibuat menggunakan metode lain. Untuk informasi selengkapnya, lihat [Kelola templat pembuatan repositori Anda](#).

Pertimbangan untuk menggunakan pull through cache

Berikut ini harus dipertimbangkan saat menggunakan Amazon ECR pull through aturan cache.

- Membuat aturan pull through cache tidak didukung di Wilayah berikut.
 - China (Beijing) (`cn-north-1`)
 - China (Ningxia) (`cn-northwest-1`)
 - AWS GovCloud (AS-Timur) (`us-gov-east-1`)
 - AWS GovCloud (AS-Barat) (`us-gov-west-1`)
- AWS Lambda tidak mendukung penarikan gambar kontainer dari Amazon ECR menggunakan aturan cache tarik.
- Saat menarik gambar menggunakan cache tarik, titik akhir layanan Amazon ECR FIPS tidak didukung saat pertama kali gambar ditarik. Menggunakan titik akhir layanan Amazon ECR FIPS berfungsi pada tarikan berikutnya.
- Saat gambar yang di-cache ditarik melalui URI registri pribadi Amazon ECR, penarikan gambar dimulai oleh alamat IP. AWS Ini memastikan bahwa penarikan gambar tidak dihitung terhadap kuota tingkat tarik apa pun yang diterapkan oleh registri hulu.
- Saat gambar yang di-cache ditarik melalui URI registri pribadi Amazon ECR, Amazon ECR memeriksa repositori upstream setidaknya sekali setiap 24 jam untuk memverifikasi apakah gambar yang di-cache adalah versi terbaru. Jika ada gambar yang lebih baru di registri hulu, Amazon ECR mencoba memperbarui gambar yang di-cache. Timer ini didasarkan pada tarikan terakhir dari gambar yang di-cache.
- Jika Amazon ECR tidak dapat memperbarui gambar dari registri hulu karena alasan apa pun dan gambar ditarik, gambar cache terakhir akan tetap ditarik.
- Saat membuat rahasia Secrets Manager yang berisi kredensial registri hulu, nama rahasia harus menggunakan awalan. `ecr-pullthroughcache/` Rahasiannya juga harus berada di akun dan Wilayah yang sama tempat aturan pull through cache dibuat.
- Saat gambar multi-arsitektur ditarik menggunakan aturan cache pull through, daftar manifes dan setiap gambar yang direferensikan dalam daftar manifes ditarik ke repositori Amazon ECR. Jika

- Anda hanya ingin menarik arsitektur tertentu, Anda dapat menarik gambar menggunakan intisari gambar atau tag yang terkait dengan arsitektur daripada tag yang terkait dengan daftar manifes.
- Amazon ECR menggunakan peran IAM terkait layanan, yang menyediakan izin yang diperlukan Amazon ECR untuk membuat repositori, mengambil nilai rahasia Secrets Manager untuk otentikasi, dan mendorong gambar yang di-cache atas nama Anda. Peran IAM terkait layanan dibuat secara otomatis saat aturan pull through cache dibuat. Untuk informasi selengkapnya, lihat [Peran terkait layanan Amazon ECR untuk menarik cache](#).
 - Secara default, prinsipal IAM yang menarik gambar yang di-cache memiliki izin yang diberikan kepada mereka melalui kebijakan IAM mereka. Anda dapat menggunakan kebijakan izin registri pribadi Amazon ECR untuk cakupan lebih lanjut izin entitas IAM. Untuk informasi selengkapnya, lihat [Menggunakan izin registri](#).
 - Repositori Amazon ECR yang dibuat menggunakan alur kerja pull through cache diperlakukan seperti repositori ECR Amazon lainnya. Semua fitur repositori, seperti replikasi dan pemindaian gambar didukung.
 - Saat Amazon ECR membuat repositori baru atas nama Anda menggunakan tindakan pull through cache, pengaturan default berikut diterapkan ke repositori kecuali ada template pembuatan repositori yang cocok. Anda dapat menggunakan template pembuatan repositori untuk menentukan pengaturan yang diterapkan ke repositori yang dibuat oleh Amazon ECR atas nama Anda. Untuk informasi selengkapnya, lihat [Kelola templat pembuatan repositori Anda](#).
 - Kekekalan tag - Dimatikan, tag dapat berubah dan dapat ditimpa.
 - Enkripsi — AES256 Enkripsi default digunakan.
 - Izin repositori - Dihilangkan, tidak ada kebijakan izin repositori yang diterapkan.
 - Kebijakan siklus hidup - Dihilangkan, tidak ada kebijakan siklus hidup yang diterapkan.
 - Tag sumber daya - Dihilangkan, tidak ada tag sumber daya yang diterapkan.
 - Mengaktifkan kekekalan tag gambar untuk repositori menggunakan aturan cache tarik melalui akan mencegah Amazon ECR memperbarui gambar menggunakan tag yang sama.
 - Ketika gambar ditarik menggunakan aturan pull through cache untuk pertama kalinya rute ke internet mungkin diperlukan. Ada keadaan tertentu di mana rute ke internet diperlukan sehingga yang terbaik adalah mengatur rute untuk menghindari kegagalan. Jadi, jika Anda telah mengonfigurasi Amazon ECR untuk AWS PrivateLink menggunakan titik akhir VPC antarmuka, maka Anda perlu memastikan tarikan pertama memiliki rute ke internet. Salah satu cara untuk melakukannya adalah dengan membuat subnet publik di VPC yang sama, dengan gateway internet, dan kemudian merutekan semua lalu lintas keluar ke internet dari subnet pribadi mereka ke subnet publik. Penarikan gambar berikutnya menggunakan aturan pull through cache tidak

memerlukan ini. Untuk informasi selengkapnya, lihat [Contoh opsi perutean](#) di Panduan Pengguna Amazon Virtual Private Cloud.

Izin IAM yang diperlukan

Selain izin Amazon ECR API yang diperlukan untuk mengautentikasi ke registri pribadi dan untuk mendorong dan menarik gambar, izin tambahan berikut diperlukan untuk menggunakan aturan cache tarik melalui secara efektif.

- `ecr:CreatePullThroughCacheRule`— Memberikan izin untuk membuat aturan cache pull through. Izin ini harus diberikan melalui kebijakan IAM berbasis identitas.
- `ecr:BatchImportUpstreamImage`— Memberikan izin untuk mengambil gambar eksternal dan mengimpornya ke registri pribadi Anda. Izin ini dapat diberikan dengan menggunakan kebijakan izin registri pribadi, kebijakan IAM berbasis identitas, atau dengan menggunakan kebijakan izin repositori berbasis sumber daya. Untuk informasi selengkapnya tentang menggunakan izin repositori, lihat. [Kebijakan repositori pribadi](#)
- `ecr:CreateRepository`— Memberikan izin untuk membuat repositori di registri pribadi. Izin ini diperlukan jika repositori yang menyimpan gambar yang di-cache belum ada. Izin ini dapat diberikan oleh kebijakan IAM berbasis identitas atau kebijakan izin registri pribadi.
- `ecr:TagResource`— Memberikan izin untuk menambahkan tag metadata ke sumber daya Amazon ECR. Izin ini hanya diperlukan jika Anda menarik gambar yang menggunakan aturan cache tarik melalui yang memiliki template pembuatan repositori terkait yang dikonfigurasi untuk menambahkan tag sumber daya ke repositori. Izin ini harus diberikan melalui kebijakan IAM berbasis identitas.

Menggunakan izin registri

Izin registri pribadi Amazon ECR dapat digunakan untuk mencakup izin entitas IAM individu untuk menggunakan cache pull through. Jika entitas IAM memiliki lebih banyak izin yang diberikan oleh kebijakan IAM daripada yang diberikan oleh kebijakan izin registri, kebijakan IAM akan diutamakan. Misalnya, jika pengguna memiliki `ecr:*` izin yang diberikan, tidak ada izin tambahan yang diperlukan di tingkat registri.

Untuk membuat kebijakan izin registri pribadi ()AWS Management Console

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah untuk mengonfigurasi pernyataan izin registri pribadi Anda.

3. Di panel navigasi, pilih Registri pribadi, Izin registri.
4. Pada halaman Izin registri, pilih Hasilkan pernyataan.
5. Untuk setiap pull through pernyataan kebijakan izin cache yang ingin Anda buat, lakukan hal berikut.
 - a. Untuk jenis Kebijakan, pilih Tarik kebijakan cache.
 - b. Untuk id Pernyataan, berikan nama untuk kebijakan pernyataan cache tarik melalui.
 - c. Untuk entitas IAM, tentukan pengguna, grup, atau peran yang akan disertakan dalam kebijakan.
 - d. Untuk namespace Repositori, pilih aturan pull through cache untuk mengaitkan kebijakan dengan.
 - e. Untuk nama Repositori, tentukan nama dasar repositori untuk menerapkan aturan. Misalnya, jika Anda ingin menentukan repositori Amazon Linux di Amazon ECR Public, nama repositori akan menjadi. `amazonlinux`

Untuk membuat kebijakan izin registri pribadi ()AWS CLI

Gunakan AWS CLI perintah berikut untuk menentukan izin registri pribadi menggunakan. AWS CLI

1. Buat file lokal bernama `ptc-registry-policy.json` dengan isi kebijakan registri Anda. Contoh berikut memberikan `ecr-pull-through-cache-user` izin untuk membuat repositori dan menarik gambar dari Amazon ECR Public, yang merupakan sumber upstream yang terkait dengan aturan cache pull through yang dibuat sebelumnya.

```
{
  "Sid": "PullThroughCacheFromReadOnlyRole",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/ecr-pull-through-cache-user"
  },
  "Action": [
    "ecr:CreateRepository",
    "ecr:BatchImportUpstreamImage"
  ],
  "Resource": "arn:aws:ecr:us-east-1:111122223333:repository/ecr-public/*"
}
```

⚠ Important

`ecr-CreateRepository` hanya diperlukan jika repositori yang menyimpan gambar yang di-cache belum ada. Misalnya, jika tindakan pembuatan repositori dan tindakan tarik gambar sedang dilakukan oleh prinsipal IAM terpisah seperti administrator dan pengembang.

- Gunakan `put-registry-policy` perintah untuk mengatur kebijakan registri.

```
aws ecr put-registry-policy \  
  --policy-text file://ptc-registry.policy.json
```

Langkah selanjutnya

Setelah Anda siap untuk mulai menggunakan aturan pull through cache, berikut ini adalah langkah selanjutnya.

- Buat aturan cache pull through. Untuk informasi selengkapnya, lihat [Membuat aturan pull through cache](#).
- Buat template pembuatan repositori. Template pembuatan repositori memberi Anda kontrol untuk menentukan pengaturan yang akan digunakan untuk repositori baru yang dibuat oleh Amazon ECR atas nama Anda selama tindakan pull through cache. Untuk informasi selengkapnya, lihat [Kelola templat pembuatan repositori Anda](#).
- Pelajari cara bekerja dengan aturan pull through cache. Untuk informasi selengkapnya, lihat [Bekerja dengan aturan pull through cache](#).

Membuat aturan pull through cache

Untuk setiap registri upstream yang berisi gambar yang ingin Anda cache di registri pribadi Amazon ECR Anda, Anda harus membuat aturan cache pull through. Untuk registrasi upstream yang memerlukan otentikasi, Anda harus menyimpan kredensialnya dalam rahasia Secrets Manager. Anda dapat membuat rahasia Secrets Manager di konsol Amazon ECR atau Secrets Manager.

Sebelum Anda mulai membuat aturan cache pull through, pastikan Anda memiliki izin IAM yang tepat. Untuk informasi selengkapnya, lihat [Izin IAM yang diperlukan](#).

Untuk membuat aturan pull through cache (AWS Management Console)

Langkah-langkah berikut menunjukkan cara membuat aturan cache pull through dan rahasia Secrets Manager menggunakan konsol Amazon ECR. Untuk informasi selengkapnya tentang membuat rahasia menggunakan konsol Secrets Manager, lihat [Menyimpan kredensi repositori upstream Anda secara rahasia AWS Secrets Manager](#).


Untuk Amazon ECR Public, registri kontainer Kubernetes, atau Quay

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah untuk mengonfigurasi pengaturan registri pribadi Anda.
3. Di panel navigasi, pilih Registri pribadi, Tarik cache.
4. Pada halaman konfigurasi Tarik melalui cache, pilih Tambahkan aturan.
5. Pada Langkah 1: Tentukan halaman sumber, untuk Registry, pilih Amazon ECR Public, Kubernetes, atau Quay dari daftar registri upstream dan kemudian pilih Next.
6. Pada Langkah 2: Tentukan halaman tujuan, untuk awalan repositori Amazon ECR, tentukan awalan namespace repositori yang akan digunakan saat menyimpan gambar yang ditarik dari registri publik sumber dan kemudian pilih Berikutnya. Secara default, namespace diisi tetapi namespace khusus dapat ditentukan juga.
7. Pada Langkah 3: Tinjau dan buat halaman, tinjau konfigurasi aturan pull through cache dan kemudian pilih Create.
8. Ulangi langkah sebelumnya untuk setiap pull through cache yang ingin Anda buat. Aturan pull through cache dibuat secara terpisah untuk setiap Wilayah.

Untuk Docker Hub

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah untuk mengonfigurasi pengaturan registri pribadi Anda.
3. Di panel navigasi, pilih Registri pribadi, Tarik cache.
4. Pada halaman konfigurasi Tarik melalui cache, pilih Tambahkan aturan.
5. Pada Langkah 1: Tentukan halaman sumber, untuk Registry, pilih Docker Hub, Next.
6. Pada Langkah 2: Konfigurasi halaman otentikasi, untuk kredensi Upstream, Anda harus menyimpan kredensi otentikasi Anda untuk Docker Hub secara rahasia. AWS Secrets Manager Anda dapat menentukan rahasia yang ada atau menggunakan konsol Amazon ECR untuk membuat rahasia baru.

- a. Untuk menggunakan rahasia yang ada, pilih Gunakan AWS rahasia yang ada. Untuk nama Rahasia gunakan drop-down untuk memilih rahasia yang ada, lalu pilih Berikutnya. Untuk informasi selengkapnya tentang membuat rahasia Secrets Manager menggunakan konsol Secrets Manager, lihat [Menyimpan kredensi repositori upstream Anda secara rahasia AWS Secrets Manager](#).

 Note

AWS Management Console Satu-satunya menampilkan rahasia Secrets Manager dengan nama menggunakan `ecr-pullthroughcache/` awalan. Rahasiannya juga harus berada di akun dan Wilayah yang sama tempat aturan pull through cache dibuat.

- b. Untuk membuat rahasia baru, pilih Buat AWS rahasia, lakukan hal berikut, lalu pilih Berikutnya.
 - i. Untuk nama Rahasia, tentukan nama deskriptif untuk rahasia tersebut. Nama rahasia harus berisi 1-512 karakter Unicode.
 - ii. Untuk nama pengguna Docker Hub, tentukan nama pengguna Docker Hub Anda.
 - iii. Untuk token akses Docker Hub, tentukan token akses Docker Hub Anda. Untuk informasi selengkapnya tentang membuat token akses Docker Hub, lihat [Membuat dan mengelola token akses](#) dalam dokumentasi Docker.
7. Pada Langkah 3: Tentukan halaman tujuan, untuk awalan repositori Amazon ECR, tentukan namespace repositori yang akan digunakan saat menyimpan gambar yang ditarik dari registri publik sumber dan kemudian pilih Berikutnya.


Secara default, namespace diisi tetapi namespace khusus dapat ditentukan juga.
8. Pada Langkah 4: Tinjau dan buat halaman, tinjau konfigurasi aturan pull through cache dan kemudian pilih Create.
9. Ulangi langkah sebelumnya untuk setiap pull through cache yang ingin Anda buat. Aturan pull through cache dibuat secara terpisah untuk setiap Wilayah.

Untuk Registri GitHub Kontainer

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah untuk mengonfigurasi pengaturan registri pribadi Anda.

3. Di panel navigasi, pilih Registri pribadi, Tarik cache.
4. Pada halaman konfigurasi Tarik melalui cache, pilih Tambahkan aturan.
5. Pada Langkah 1: Tentukan halaman sumber, untuk Registry, pilih Registry GitHub Container, Next.
6. Pada Langkah 2: Konfigurasi halaman otentikasi, untuk kredensi Upstream, Anda harus menyimpan kredensi otentikasi Anda untuk Container Registry secara rahasia. GitHub AWS Secrets Manager Anda dapat menentukan rahasia yang ada atau menggunakan konsol Amazon ECR untuk membuat rahasia baru.
 - a. Untuk menggunakan rahasia yang ada, pilih Gunakan AWS rahasia yang ada. Untuk nama Rahasia gunakan drop-down untuk memilih rahasia yang ada, lalu pilih Berikutnya. Untuk informasi selengkapnya tentang membuat rahasia Secrets Manager menggunakan konsol Secrets Manager, lihat [Menyimpan kredensi repositori upstream Anda secara rahasia AWS Secrets Manager](#).
 - b. Untuk membuat rahasia baru, pilih Buat AWS rahasia, lakukan hal berikut, lalu pilih Berikutnya.
 - i. Untuk nama Rahasia, tentukan nama deskriptif untuk rahasia tersebut. Nama rahasia harus berisi 1-512 karakter Unicode.
 - ii. Untuk nama pengguna GitHub Container Registry, tentukan nama pengguna GitHub Container Registry Anda.
 - iii. Untuk token akses GitHub Container Registry, tentukan token akses GitHub Container Registry Anda. Untuk informasi selengkapnya tentang membuat token GitHub akses, lihat [Mengelola token akses pribadi Anda](#) dalam GitHub dokumentasi.
7. Pada Langkah 3: Tentukan halaman tujuan, untuk awalan repositori Amazon ECR, tentukan namespace repositori yang akan digunakan saat menyimpan gambar yang ditarik dari registri publik sumber dan kemudian pilih Berikutnya.

Secara default, namespace diisi tetapi namespace khusus dapat ditentukan juga.

 Note

AWS Management Console Satu-satunya menampilkan rahasia Secrets Manager dengan nama menggunakan `ecr-pullthroughcache/` awalan. Rahasiannya juga harus berada di akun dan Wilayah yang sama tempat aturan pull through cache dibuat.

8. Pada Langkah 4: Tinjau dan buat halaman, tinjau konfigurasi aturan pull through cache dan kemudian pilih Create.
9. Ulangi langkah sebelumnya untuk setiap pull through cache yang ingin Anda buat. Aturan pull through cache dibuat secara terpisah untuk setiap Wilayah.

Untuk Microsoft Azure Container Registry

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah untuk mengonfigurasi pengaturan registri pribadi Anda.
3. Di panel navigasi, pilih Registri pribadi, Tarik cache.
4. Pada halaman konfigurasi Tarik melalui cache, pilih Tambahkan aturan.
5. Pada Langkah 1: Tentukan halaman sumber, lakukan hal berikut.
 - a. Untuk Registry, pilih Microsoft Azure Container Registry
 - b. Untuk URL registri Sumber, tentukan nama registri kontainer Microsoft Azure Anda, lalu pilih Berikutnya.

Important

Anda hanya perlu menentukan awalan, karena `.azurecr.io` akhiran diisi atas nama Anda.

6. Pada Langkah 2: Konfigurasi halaman otentikasi, untuk kredensial Upstream, Anda harus menyimpan kredensial otentikasi Anda untuk Microsoft Azure Container Registry secara rahasia. AWS Secrets Manager Anda dapat menentukan rahasia yang ada atau menggunakan konsol Amazon ECR untuk membuat rahasia baru.
 - a. Untuk menggunakan rahasia yang ada, pilih Gunakan AWS rahasia yang ada. Untuk nama Rahasia gunakan drop-down untuk memilih rahasia yang ada, lalu pilih Berikutnya. Untuk informasi selengkapnya tentang membuat rahasia Secrets Manager menggunakan konsol Secrets Manager, lihat [Menyimpan kredensi repositori upstream Anda secara rahasia AWS Secrets Manager](#).

Note

AWS Management Console Satu-satunya menampilkan rahasia Secrets Manager dengan nama menggunakan `ecr-pullthroughcache/` awalan. Rahasiannya juga

harus berada di akun dan Wilayah yang sama tempat aturan pull through cache dibuat.

- b. Untuk membuat rahasia baru, pilih Buat AWS rahasia, lakukan hal berikut, lalu pilih Berikutnya.
 - i. Untuk nama Rahasia, tentukan nama deskriptif untuk rahasia tersebut. Nama rahasia harus berisi 1-512 karakter Unicode.
 - ii. Untuk nama pengguna Microsoft Azure Container Registry, tentukan nama pengguna Microsoft Azure Container Registry Anda.
 - iii. Untuk token akses Microsoft Azure Container Registry, tentukan token akses Microsoft Azure Container Registry Anda. Untuk informasi selengkapnya tentang cara membuat token akses Microsoft Azure Container Registry, lihat [Membuat token - portal](#) di dokumentasi Microsoft Azure.
7. Pada Langkah 3: Tentukan halaman tujuan, untuk awalan repositori Amazon ECR, tentukan namespace repositori yang akan digunakan saat menyimpan gambar yang ditarik dari registri publik sumber dan kemudian pilih Berikutnya.

Secara default, namespace diisi tetapi namespace khusus dapat ditentukan juga.
8. Pada Langkah 4: Tinjau dan buat halaman, tinjau konfigurasi aturan pull through cache dan kemudian pilih Create.
9. Ulangi langkah sebelumnya untuk setiap pull through cache yang ingin Anda buat. Aturan pull through cache dibuat secara terpisah untuk setiap Wilayah.

Untuk membuat aturan pull through cache (AWS CLI)

Gunakan AWS CLI perintah [create-pull-through-cache-rule](#) untuk membuat aturan cache pull through untuk registri pribadi Amazon ECR. Untuk registrasi upstream yang memerlukan otentikasi, Anda harus menyimpan kredensialnya dalam rahasia Secrets Manager. Untuk informasi selengkapnya tentang membuat rahasia menggunakan konsol Secrets Manager, lihat [Menyimpan kredensi repositori upstream Anda secara rahasia AWS Secrets Manager](#).

Contoh berikut disediakan untuk setiap registri upstream yang didukung.

Untuk Amazon ECR Publik

Contoh berikut membuat aturan cache pull through untuk registri Publik Amazon ECR. Ini menentukan awalan repositori `ecr-public`, yang menghasilkan setiap repositori yang

dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. `ecr-public/upstream-repository-name`

```
aws ecr create-pull-through-cache-rule \  
  --ecr-repository-prefix ecr-public \  
  --upstream-registry-url public.ecr.aws \  
  --region us-east-2
```

Untuk registri kontainer Kubernetes

Contoh berikut membuat aturan pull through cache untuk registri publik Kubernetes. Ini menentukan awalan `repositorikubernetes`, yang menghasilkan setiap repositori yang dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. `kubernetes/upstream-repository-name`

```
aws ecr create-pull-through-cache-rule \  
  --ecr-repository-prefix kubernetes \  
  --upstream-registry-url registry.k8s.io \  
  --region us-east-2
```

Untuk Quay

Contoh berikut membuat aturan pull through cache untuk registri publik Quay. Ini menentukan awalan `repositoriquay`, yang menghasilkan setiap repositori yang dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. `quay/upstream-repository-name`

```
aws ecr create-pull-through-cache-rule \  
  --ecr-repository-prefix quay \  
  --upstream-registry-url quay.io \  
  --region us-east-2
```

Untuk Docker Hub

Contoh berikut membuat aturan pull through cache untuk registri Docker Hub. Ini menentukan awalan `repositoridocker-hub`, yang menghasilkan setiap repositori yang dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. `docker-hub/upstream-repository-name` Anda harus menentukan Nama Sumber Daya Amazon (ARN) lengkap dari rahasia yang berisi kredensial Docker Hub Anda.

```
aws ecr create-pull-through-cache-rule \  
  --ecr-repository-prefix docker-hub \  
  --upstream-registry-url registry.docker.com \  
  --region us-east-2
```

```
--upstream-registry-url registry-1.docker.io \
--credential-arn arn:aws:secretsmanager:us-east-2:111122223333:secret:ecr-
pullthroughcache/example1234 \
--region us-east-2
```

Untuk Registri GitHub Kontainer

Contoh berikut membuat aturan pull through cache untuk registri Registry GitHub Container. Ini menentukan awalan repositoridocker-hub, yang menghasilkan setiap repositori yang dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. `github/upstream-repository-name` Anda harus menentukan Nama Sumber Daya Amazon (ARN) lengkap dari rahasia yang berisi kredensyal Registri GitHub Penampung Anda.

```
aws ecr create-pull-through-cache-rule \
--ecr-repository-prefix github \
--upstream-registry-url ghcr.io \
--credential-arn arn:aws:secretsmanager:us-east-2:111122223333:secret:ecr-
pullthroughcache/example1234 \
--region us-east-2
```

Untuk Microsoft Azure Container Registry

Contoh berikut membuat aturan pull through cache untuk registri Microsoft Azure Container Registry. Ini menentukan awalan repositoriazure, yang menghasilkan setiap repositori yang dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. `azure/upstream-repository-name` Anda harus menentukan Nama Sumber Daya Amazon (ARN) lengkap dari rahasia yang berisi kredensyal Microsoft Azure Container Registry Anda.

```
aws ecr create-pull-through-cache-rule \
--ecr-repository-prefix azure \
--upstream-registry-url myregistry.azurecr.io \
--credential-arn arn:aws:secretsmanager:us-east-2:111122223333:secret:ecr-
pullthroughcache/example1234 \
--region us-east-2
```

Langkah selanjutnya

Setelah Anda membuat aturan cache pull through, berikut ini adalah langkah selanjutnya.

- Buat template pembuatan repositori. Template pembuatan repositori memberi Anda kontrol untuk menentukan pengaturan yang akan digunakan untuk repositori baru yang dibuat oleh Amazon ECR

atas nama Anda selama tindakan pull through cache. Untuk informasi selengkapnya, lihat [Kelola templat pembuatan repositori Anda](#).

- Validasi tarik Anda melalui aturan cache. Saat memvalidasi aturan pull through cache, Amazon ECR membuat koneksi jaringan dengan registri upstream, memverifikasi bahwa ia dapat mengakses rahasia Secrets Manager yang berisi kredensial untuk registri upstream, dan otentikasi itu berhasil. Untuk informasi selengkapnya, lihat [Memvalidasi aturan pull through cache Anda](#).
- Mulai gunakan aturan cache pull through Anda. Untuk informasi selengkapnya, lihat [Menarik gambar dengan aturan cache pull through](#).

Kelola templat pembuatan repositori Anda

Fitur template pembuatan repositori ada dalam rilis pratinjau untuk Amazon ECR dan dapat berubah sewaktu-waktu. Selama pratinjau publik ini, hanya yang AWS Management Console dapat digunakan untuk mengelola template pembuatan repositori Anda.

Templat pembuatan repositori Amazon ECR memberi Anda kontrol untuk menentukan pengaturan yang akan digunakan untuk repositori baru yang dibuat oleh Amazon ECR atas nama Anda selama tindakan pull through cache. Pengaturan dalam template pembuatan repositori hanya diterapkan selama pembuatan repositori dan tidak berpengaruh pada repositori atau repositori yang ada yang dibuat menggunakan metode lain.

Templat pembuatan repositori tidak didukung di Wilayah berikut.

- China (Beijing) (cn-north-1)
- China (Ningxia) (cn-northwest-1)
- AWS GovCloud (AS-Timur) (us-gov-east-1)
- AWS GovCloud (AS-Barat) (us-gov-west-1)

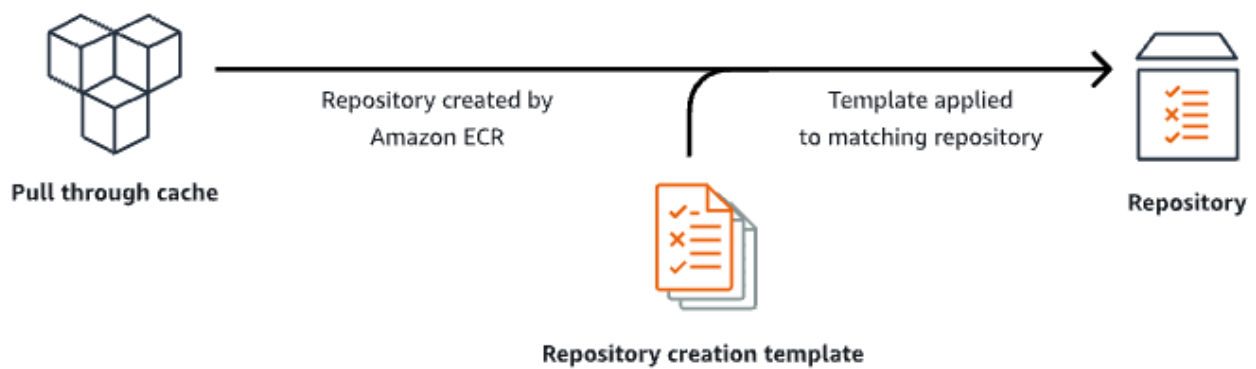
Cara kerja template pembuatan repositori

Ada kalanya Amazon ECR perlu membuat repositori pribadi baru atas nama Anda. Misalnya, pertama kali Anda menggunakan aturan pull through cache untuk mengambil konten repositori upstream dan menyimpannya di registri pribadi Amazon ECR Anda. Jika tidak ada template pembuatan repositori yang cocok dengan aturan cache pull through Anda, Amazon ECR menggunakan pengaturan default

untuk repositori baru. Pengaturan default ini termasuk mematikan kekekalan tag, menggunakan AES-256 enkripsi, dan tidak menerapkan kebijakan repositori atau siklus hidup apa pun.

Menggunakan template pembuatan repositori dengan awalan yang cocok dengan aturan pull through cache memberi Anda kemampuan untuk menentukan pengaturan Amazon ECR berlaku untuk repositori baru yang dibuat melalui tindakan pull through cache. Anda dapat menentukan kekekalan tag, konfigurasi enkripsi, izin repositori, kebijakan siklus hidup, dan tag sumber daya untuk repositori baru.

Diagram berikut menunjukkan alur kerja yang digunakan Amazon ECR saat template pembuatan repositori digunakan.



Berikut ini menjelaskan setiap parameter dalam template pembuatan repositori secara rinci.

Prefiks

Awalan adalah awalan namespace repositori untuk dikaitkan dengan template. Semua repositori yang dibuat menggunakan awalan ini akan memiliki pengaturan yang diterapkan yang ditentukan dalam template ini. Misalnya, awalan `prod` akan berlaku untuk semua repositori yang dimulai dengan `prod/`. Demikian pula, awalan `prod/team` akan berlaku untuk semua repositori yang dimulai dengan `prod/team/`.

Untuk menerapkan template ke semua repositori di registri Anda yang tidak memiliki template pembuatan terkait, Anda dapat menggunakan `ROOT` sebagai awalan.

Important

Selalu ada asumsi `/` diterapkan pada akhir awalan. Jika Anda menentukan `ecr-public` sebagai awalan, Amazon ECR memperlakukannya sebagai `ecr-public/`. Saat

menggunakan aturan pull through cache, awalan repositori yang Anda tentukan selama pembuatan aturan adalah apa yang harus Anda tentukan sebagai awalan template pembuatan repositori Anda juga.

Deskripsi

Deskripsi template ini bersifat opsional dan digunakan untuk menjelaskan tujuan template pembuatan repositori.

Versi template

Versi template pembuatan repositori yang akan digunakan. Saat ini, hanya versi TV1 template yang didukung.

Versi konfigurasi

Versi konfigurasi repositori template yang akan digunakan. Setiap template harus menyertakan konfigurasi repositori. Versi konfigurasi default adalah CV1 dan terdiri dari mutabilitas tag gambar, kebijakan repositori, dan setelan kebijakan siklus hidup.

Tanda ketetapan citra

Pengaturan mutabilitas tag yang akan digunakan untuk repositori yang dibuat menggunakan template. Jika parameter ini dihilangkan, pengaturan default MUTABLE akan digunakan yang akan memungkinkan tag gambar untuk ditimpa. Ini adalah pengaturan yang disarankan untuk digunakan untuk template yang digunakan untuk repositori yang dibuat dengan tindakan pull through cache. Ini memastikan bahwa Amazon ECR dapat memperbarui gambar yang di-cache saat tag sama.

Jika IMMUTABLE ditentukan, semua tag gambar dalam repositori akan kekal yang akan mencegahnya ditimpa.

Konfigurasi enkripsi

Konfigurasi enkripsi yang digunakan untuk repositori yang dibuat menggunakan template.

Jika Anda menggunakan jenis enkripsi KMS, isi repositori akan dienkripsi menggunakan enkripsi sisi server dengan kunci yang disimpan di AWS Key Management Service AWS KMS. Saat Anda menggunakan AWS KMS untuk mengenkripsi data, Anda dapat menggunakan AWS KMS kunci AWS terkelola default untuk Amazon ECR, atau menentukan AWS KMS kunci Anda sendiri, yang sudah Anda buat. Untuk informasi selengkapnya, lihat [Melindungi data menggunakan enkripsi sisi](#)

[server dengan AWS Key Management Service kunci yang disimpan di AWS Key Management Service \(SSE-KMS\)](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Jika Anda menggunakan jenis enkripsi AES256, Amazon ECR menggunakan enkripsi sisi server dengan kunci enkripsi yang dikelola Amazon S3 yang mengenkripsi gambar di repositori menggunakan algoritma enkripsi AES-256. Untuk informasi selengkapnya, lihat [Melindungi data menggunakan enkripsi sisi server dengan kunci enkripsi terkelola Amazon S3 \(SSE-S3\)](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Izin repositori

Kebijakan repositori untuk diterapkan ke repositori yang dibuat menggunakan template. Kebijakan repositori menggunakan izin berbasis sumber daya untuk mengontrol akses ke repositori. Izin berbasis sumber daya memungkinkan Anda menentukan pengguna atau peran IAM mana yang memiliki akses ke repositori dan tindakan apa yang dapat mereka lakukan di situ. Secara default, hanya AWS akun yang membuat repositori yang memiliki akses ke repositori. Anda dapat menerapkan dokumen kebijakan yang memberikan atau menolak izin tambahan ke repositori Anda. Untuk informasi selengkapnya, lihat

[Amazon ECR menggunakan izin berbasis sumber daya untuk mengontrol akses ke repositori. Izin berbasis sumber daya memungkinkan Anda menentukan pengguna atau peran mana yang memiliki akses ke repositori dan tindakan apa yang dapat mereka lakukan padanya. Secara default, hanya AWS akun yang membuat repositori yang memiliki akses ke repositori. Anda dapat menerapkan dokumen kebijakan yang memungkinkan izin tambahan ke repositori Anda.](#)

Topik

- [Kebijakan repositori vs kebijakan IAM](#)
- [Menyetel pernyataan kebijakan repositori pribadi](#)
- [Menghapus pernyataan kebijakan repositori pribadi](#)
- [Contoh kebijakan repositori pribadi](#)

Kebijakan repositori vs kebijakan IAM

[Kebijakan repositori Amazon ECR adalah bagian dari kebijakan IAM yang memiliki lingkup untuk, dan secara khusus digunakan untuk, mengendalikan akses ke repositori Amazon ECR individu. Kebijakan IAM umumnya digunakan untuk menerapkan izin untuk seluruh layanan Amazon ECR tetapi juga dapat digunakan untuk mengontrol akses ke sumber daya tertentu.](#)

Kebijakan repositori Amazon ECR dan kebijakan IAM digunakan saat menentukan tindakan yang mungkin dilakukan pengguna atau peran tertentu pada repositori. Jika pengguna atau peran diperbolehkan untuk melakukan tindakan melalui kebijakan repositori tetapi tidak diberi izin melalui kebijakan IAM (atau sebaliknya) maka tindakan akan ditolak. Agar tindakan diizinkan, pengguna atau peran hanya perlu diberi izin untuk tindakan baik melalui kebijakan repositori atau kebijakan IAM, tetapi tidak keduanya.

⚠ Important

Amazon ECR mengharuskan pengguna memiliki izin untuk melakukan panggilan ke API `ecr:GetAuthorizationToken` melalui kebijakan IAM sebelum mereka dapat

mengautentikasi ke registri dan mendorong atau menarik citra dari repositori Amazon

ECR. Amazon ECR menyediakan beberapa kebijakan yang dikelola untuk mengontrol

akses pengguna pada berbagai tingkat; untuk informasi lebih lanjut, lihat [Contoh kebijakan berbasis identitas Amazon Elastic Container Registry](#).

Anda dapat menggunakan salah satu jenis kebijakan ini untuk mengontrol akses ke repositori Anda, seperti yang ditampilkan dalam contoh berikut.

Contoh ini menunjukkan kebijakan repositori Amazon ECR, yang memungkinkan pengguna tertentu untuk mendeskripsikan repositori dan gambar dalam repositori.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ECRRepositoryPolicy",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::account-id:user/username"},
      "Action": [
        "ecr:DescribeImages",
```

menggunakan parameter sumber daya. Untuk informasi lebih lanjut tentang format Amazon Resource Name (ARN), lihat [Sumber daya](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDescribeRepoImage",
      "Effect": "Allow",
      "Action": [
        "ecr:DescribeImages",
        "ecr:DescribeRepositories"
      ],
      "Resource": ["arn:aws:ecr:region:account-id:repository/repository-name"]
    }
  ]
}
```

Menyetel pernyataan kebijakan repositori pribadi

Anda dapat menambahkan pernyataan kebijakan akses ke repositori di AWS Management Console dengan mengikuti langkah-langkah di bawah ini. Anda dapat menambahkan beberapa pernyataan kebijakan per repositori. Untuk contoh kebijakan, lihat [Contoh kebijakan repositori pribadi](#).

Important

Menetapkan pernyataan kebijakan repositori

Amazon ECR mengharuskan pengguna memiliki izin untuk melakukan panggilan ke API `ecr:GetAuthorizationToken` melalui kebijakan IAM sebelum mereka dapat

Versi API 2015-09-21 81

2. Dari bilah navigasi, pilih Wilayah yang berisi repositori yang akan ditetapkan pernyataan kebijakan.
3. Di panel navigasi, pilih Repositori.
4. Pada Repositori, pilih repositori yang akan ditetapkan pernyataan kebijakan untuk melihat isi repositori.
5. Dari tampilan daftar citra repositori, di panel navigasi, pilih Izin, Edit.

 Note

Jika Anda tidak melihat opsi Izin di panel navigasi, pastikan bahwa Anda berada di tampilan daftar citra repositori.

6. Pada halaman Edit izin, pilih Tambah pernyataan.
7. Untuk Nama pernyataan, masukkan nama untuk pernyataan tersebut.
8. Untuk Efek, pilih apakah pernyataan kebijakan tersebut akan menghasilkan izin atau penolakan eksplisit.
9. Untuk Utama, pilih ruang lingkup untuk penerapan pernyataan kebijakan. Untuk informasi lebih lanjut, lihat [Elemen Kebijakan JSON AWS: Utama](#) dalam Panduan Pengguna IAM.
 - Anda dapat menerapkan pernyataan untuk semua pengguna AWS yang sudah diautentikasi dengan memilih kotak centang Semua orang (*).
 - Untuk Prinsipal layanan, tentukan nama prinsipal layanan (misalnya, `ecs.amazonaws.com`) untuk menerapkan pernyataan ke layanan tertentu.
 - Untuk ID Akun AWS, tentukan nomor akun AWS (misalnya, 111122223333) untuk menerapkan pernyataan untuk semua pengguna di bawah akun AWS tertentu. Beberapa akun dapat ditentukan dengan menggunakan daftar yang dipisahkan koma.

 Important

Akun yang Anda berikan izin harus memiliki Wilayah yang Anda beri kebijakan repositori yang diaktifkan, jika tidak, kesalahan akan terjadi.

- Untuk Entitas IAM, pilih peran atau pengguna dengan akun AWS Anda untuk diterapkan pernyataan.

Note

Untuk kebijakan repositori yang lebih rumit yang saat ini tidak didukung di AWS Management Console, Anda dapat menerapkan kebijakan dengan perintah `set-repository-policy` AWS CLI.

10. Untuk Tindakan, pilih ruang lingkup operasi API Amazon ECR yang akan diterapkan pernyataan kebijakan dari daftar operasi API individu.
11. Setelah selesai, pilih Simpan untuk menetapkan kebijakan.
12. Ulangi langkah sebelumnya untuk setiap kebijakan repositori yang ditambahkan.

Menghapus pernyataan kebijakan repositori pribadi

Jika Anda tidak lagi menginginkan pernyataan kebijakan repositori yang ada untuk diterapkan ke repositori, Anda dapat menghapusnya.

Untuk menghapus pernyataan kebijakan repositori

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah yang berisi repositori yang akan dihapus pernyataan kebijakannya.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih repositori yang akan dihapus pernyataan kebijakannya.
5. Di panel navigasi, pilih Izin, Edit.
6. Pada halaman Edit izin, pilih Tambah pernyataan.

Contoh kebijakan repositori pribadi

Important

Contoh kebijakan repositori di halaman ini dimaksudkan untuk diterapkan ke repositori pribadi Amazon ECR. Mereka tidak akan berfungsi dengan baik jika digunakan

dengan prinsipal IAM secara langsung kecuali dimodifikasi untuk menentukan

repositori Amazon ECR sebagai sumber daya. Untuk informasi selengkapnya tentang pengaturan kebijakan repositori, lihat [Menyetel pernyataan kebijakan repositori pribadi](#)

Kebijakan repositori Amazon ECR adalah bagian dari kebijakan IAM yang memiliki lingkup untuk, dan secara khusus digunakan untuk, mengendalikan akses ke repositori Amazon ECR individu. Kebijakan IAM umumnya digunakan untuk menerapkan izin untuk seluruh layanan Amazon ECR tetapi juga dapat digunakan untuk mengontrol akses ke sumber daya tertentu. Untuk informasi selengkapnya, lihat [Kebijakan repositori vs kebijakan IAM](#).

Contoh kebijakan repositori berikut menunjukkan pernyataan izin yang dapat Anda gunakan untuk mengontrol akses ke repositori pribadi Amazon ECR Anda.

Important

Amazon ECR mengharuskan pengguna memiliki izin untuk melakukan panggilan ke API `ecr:GetAuthorizationToken` melalui kebijakan IAM sebelum mereka dapat

melakukan autentikasi ke registrasi dan mendorong atau menarik citra dari repositori

Amazon ECR. Amazon ECR menyediakan beberapa kebijakan IAM terkelola untuk

mengontrol akses pengguna pada berbagai tingkat; untuk informasi lebih lanjut, lihat [Contoh kebijakan berbasis identitas Amazon Elastic Container Registry](#).

Contoh: Izinkan satu atau lebih pengguna

Kebijakan repositori berikut memungkinkan satu atau lebih pengguna untuk mendorong dan menarik gambar ke dan dari repositori.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
```

```
    {
```

```
      "Sid": "AllowPushPull",
```

```
      "Effect": "Allow",
```

```

    },
    "Action": [
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability",
        "ecr:CompleteLayerUpload",
        "ecr:GetDownloadUrlForLayer",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
    ]
}
]
}

```

Contoh: Izinkan akun lain

Kebijakan repositori berikut memungkinkan akun tertentu untuk mendorong citra.

Important

Akun yang Anda berikan izin harus memiliki Wilayah yang Anda beri kebijakan repositori yang diaktifkan, jika tidak, kesalahan akan terjadi.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCrossAccountPush",

```

```
}

```

Kebijakan repositori berikut memungkinkan beberapa pengguna untuk menarik gambar (pull-user-1 dan pull-user-2) sambil memberikan akses penuh ke yang lain (admin-user).

 Note

Untuk kebijakan repositori yang lebih rumit yang saat ini tidak didukung di AWS Management Console, Anda dapat menerapkan kebijakan dengan perintah `set-repository-policy` AWS CLI.

```
{

```

```
  "Version": "2012-10-17",

```

```
  "Statement": [

```

```
    {

```

```
      "Sid": "AllowPull",

```

```
      "Effect": "Allow",

```

```
      "Principal": {

```

```
        "AWS": [

```

```
          "arn:aws:iam::account-id:user/pull-user-1",

```

```
          "arn:aws:iam::account-id:user/pull-user-2"

```

```
        ]

```

```
      },

```

```
      "Action": [

```

```
        "ecr:BatchGetImage",

```

```
        "ecr:GetDownloadUrlForLayer"

```

```
    ]

```

```
  },

```

```
}

```

Contoh: Tolak semua

Kebijakan repositori berikut menolak semua pengguna di semua akun untuk menarik citra.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyPull",
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer"
      ]
    }
  ]
}
```

Contoh: Membatasi akses ke alamat IP tertentu

Contoh berikut menolak izin kepada pengguna mana pun untuk melakukan operasi ECR Amazon apa pun saat diterapkan ke repositori dari rentang alamat tertentu.

Kondisi dalam pernyataan ini mengidentifikasi 54.240.143.* kisaran alamat IP Internet Protocol versi 4 (IPv4) yang diizinkan.

Contoh kebijakan repositori **ConditionBlock menggunakan NotIpAddress kondisi dan kunci aws:SourceIp kondisi** Versi: April 2019 09-21 87

yang merupakan kunci kondisi AWS -wide. Untuk informasi lebih lanjut tentang kunci syarat ini, lihat [AWS Kunci Konteks Syarat Global](#). Nilai IPv4 `aws:sourceIp` menggunakan notasi CIDR standar. Untuk informasi lebih lanjut, lihat [Operator Syarat Alamat IP dalam Panduan](#)

```

    "Sid": "IPAllow",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "ecr:*",
    "Condition": {
        "NotIpAddress": {
            "aws:SourceIp": "54.240.143.0/24"
        }
    }
}
]
}

```

Contoh: Izinkan AWS layanan

Kebijakan repositori berikut memungkinkan akses AWS CodeBuild ke tindakan API Amazon ECR yang diperlukan untuk integrasi dengan layanan tersebut. Saat menggunakan contoh berikut, Anda harus menggunakan kunci `aws:SourceArn` dan `aws:SourceAccount` kondisi untuk cakupan sumber daya mana yang dapat mengasumsikan izin ini. Untuk informasi selengkapnya, lihat [contoh Amazon ECR CodeBuild](#) di Panduan AWS CodeBuild Pengguna.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      }
    }
  ]
}

```

```
}  
}
```

```
]
```

```
}
```

Kebijakan siklus hidup repositori

Kebijakan siklus hidup yang akan digunakan untuk repositori yang dibuat menggunakan template. Kebijakan siklus hidup memberikan kontrol lebih besar atas pengelolaan siklus hidup gambar dalam repositori pribadi. Sebuah kebijakan siklus hidup berisi satu atau lebih aturan, di mana setiap aturan mendefinisikan sebuah tindakan untuk Amazon ECR. Ini menyediakan cara untuk mengotomatiskan pembersihan gambar kontainer Anda dengan kedaluwarsa gambar berdasarkan usia atau hitungan. Untuk informasi selengkapnya, lihat [Kebijakan siklus hidup](#).

Tanda sumber daya

Tag sumber daya adalah metadata untuk diterapkan ke repositori untuk membantu Anda mengkategorikan dan mengaturnya. Setiap tanda terdiri atas sebuah kunci dan sebuah nilai opsional, yang keduanya Anda tentukan.

Izin IAM yang diperlukan untuk membuat templat pembuatan repositori

Izin berikut diperlukan untuk prinsipal IAM untuk mengelola template pembuatan repositori. Izin ini harus diberikan menggunakan kebijakan IAM berbasis identitas.

- `ecr:CreateRepositoryCreationTemplate`— Memberikan izin untuk membuat template pembuatan repositori.
- `ecr>DeleteRepositoryCreationTemplate`— Memberikan izin untuk menghapus template pembuatan repositori.
- `ecr:PutLifecyclePolicy`— Memberikan izin untuk membuat kebijakan siklus hidup dan menerapkannya ke repositori. Izin ini hanya diperlukan jika template pembuatan repositori menyertakan kebijakan siklus hidup.
- `ecr:SetRepositoryPolicy`— Memberikan izin untuk membuat kebijakan izin untuk repositori. Izin ini hanya diperlukan jika template pembuatan repositori menyertakan kebijakan repositori.
- `ecr:TagResource`— Memberikan izin untuk menambahkan tag metadata ke sumber daya. Izin ini hanya diperlukan jika template pembuatan repositori menyertakan tag sumber daya.

Membuat template pembuatan repositori

Anda dapat membuat template pembuatan repositori untuk menentukan pengaturan yang akan digunakan untuk repositori yang dibuat oleh Amazon ECR atas nama Anda selama tindakan pull through cache. Setelah template pembuatan repositori dibuat, semua repositori baru yang dibuat akan memiliki pengaturan yang diterapkan. Ini tidak berpengaruh pada repositori yang dibuat sebelumnya.

Untuk membuat template pembuatan repositori () AWS Management Console

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah untuk membuat template pembuatan repositori di.
3. Di panel navigasi, pilih Registri pribadi, Template pembuatan repositori.
4. Pada halaman template pembuatan repositori, pilih Buat template.
5. Pada Langkah 1: Tentukan halaman template, untuk detail Template, pilih awalan khusus untuk menerapkan template ke awalan namespace repositori tertentu atau pilih Awalan apa pun di registri ECR Anda untuk menerapkan templat ke semua repositori yang tidak cocok dengan templat lain di Wilayah.
 - a. Jika Anda memilih awalan tertentu, untuk Awalan tentukan awalan namespace repositori untuk menerapkan template ke. Selalu ada asumsi / diterapkan pada akhir awalan. Misalnya, awalan `prod` akan berlaku untuk semua repositori yang dimulai dengan `prod/`. Demikian pula, awalan `prod/team` akan berlaku untuk semua repositori yang dimulai dengan `prod/team/`.
 - b. Jika Anda memilih awalan apa pun di registri ECR Anda, Awalan akan diatur ke `ROOT`.
6. Untuk deskripsi Template, tentukan deskripsi opsional untuk template dan kemudian pilih Berikutnya.
7. Pada Langkah 2: Tambahkan halaman konfigurasi pembuatan repositori, tentukan konfigurasi pengaturan repositori untuk diterapkan ke repositori yang dibuat menggunakan templat.
 - a. Untuk mutabilitas tag Gambar, pilih pengaturan mutabilitas tag yang akan digunakan. Untuk informasi selengkapnya, lihat [Tanda ketetapan citra](#).


Ketika Mutable dipilih, tag gambar dapat ditimpa. Ini adalah pengaturan yang disarankan untuk digunakan untuk template yang digunakan untuk repositori yang dibuat dengan tindakan pull through cache. Ini memastikan bahwa Amazon ECR dapat memperbarui gambar yang di-cache saat tag sama.

Ketika Immutable dipilih, tag gambar dicegah untuk ditimpa. Setelah repositori dikonfigurasi untuk tag yang tidak dapat diubah, `ImageTagAlreadyExistsException` kesalahan dikembalikan jika ada upaya untuk mendorong gambar dengan tag yang sudah ada di repositori. Ketika kekekalan tag diaktifkan untuk repositori, ini memengaruhi semua tag dan Anda tidak dapat membuat beberapa tag tidak dapat diubah sementara yang lain tidak.

- b. Untuk konfigurasi Enkripsi, pilih pengaturan enkripsi yang akan digunakan. Untuk informasi selengkapnya, lihat [Enkripsi diam](#).

Saat AES-256 dipilih, Amazon ECR menggunakan enkripsi sisi server dengan kunci enkripsi yang dikelola Amazon Simple Storage Service yang mengenkripsi data Anda saat istirahat menggunakan algoritme enkripsi AES-256 standar industri. Ini ditawarkan tanpa biaya tambahan.

Saat AWSKMS dipilih, Amazon ECR menggunakan enkripsi sisi server dengan kunci yang disimpan di (). AWS Key Management Service AWS KMS Saat Anda menggunakan AWS KMS untuk mengenkripsi data, Anda dapat menggunakan kunci AWS terkelola default, yang dikelola oleh Amazon ECR, atau menentukan AWS KMS kunci Anda sendiri, yang disebut sebagai kunci yang dikelola pelanggan.

 Note

Pengaturan enkripsi untuk repositori tidak dapat diubah setelah repositori dibuat.

- c. Untuk izin Repositori, tentukan kebijakan izin repositori yang akan diterapkan ke repositori yang dibuat menggunakan templat ini. Anda dapat menggunakan drop-down secara opsional untuk memilih salah satu sampel JSON untuk kasus penggunaan yang paling umum. Untuk informasi selengkapnya, lihat [Kebijakan repositori pribadi](#).
- d. Untuk kebijakan siklus hidup Repositori, tentukan kebijakan siklus hidup repositori yang akan diterapkan ke repositori yang dibuat menggunakan templat ini. Anda dapat menggunakan drop-down secara opsional untuk memilih salah satu sampel JSON untuk kasus penggunaan yang paling umum. Untuk informasi selengkapnya, lihat [Kebijakan siklus hidup](#).
- e. Untuk AWStag Repositori, tentukan metadata, dalam bentuk pasangan kunci-nilai, untuk dikaitkan dengan repositori yang dibuat menggunakan templat ini dan kemudian pilih Berikutnya. Untuk informasi selengkapnya, lihat [Menandai repositori pribadi](#).

8. Pada Langkah 3: Tinjau dan buat halaman, tinjau pengaturan yang Anda tentukan untuk template pembuatan repositori. Pilih opsi Edit untuk membuat perubahan. Pilih Buat setelah Anda selesai.

Menghapus template pembuatan repositori

Anda dapat menghapus template pembuatan repositori jika Anda selesai menggunakannya. Setelah template pembuatan repositori dihapus, setiap repositori baru yang dibuat selama tindakan pull through cache akan memiliki pengaturan default yang diterapkan.

Untuk menghapus template pembuatan repositori () AWS Management Console

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah tempat template pembuatan repositori yang akan dihapus.
3. Di panel navigasi, pilih Registri pribadi, Template pembuatan repositori.
4. Pada halaman template pembuatan repositori, pilih template pembuatan repositori untuk dihapus.
5. Dari menu tarik-turun Tindakan, pilih Hapus.

Bekerja dengan aturan pull through cache

Setelah aturan cache pull through dibuat untuk registri upstream, untuk registri upstream yang memerlukan otentikasi, Anda dapat memvalidasi aturan tersebut. Anda kemudian dapat menarik gambar hulu menggunakan URI repositori Amazon ECR Anda dan gambar di-cache di registri pribadi Amazon ECR Anda.

Sebelum Anda mulai bekerja dengan aturan cache pull through Anda, pastikan Anda memiliki izin IAM yang tepat. Untuk informasi selengkapnya, lihat [Izin IAM yang diperlukan](#).

Memvalidasi aturan pull through cache Anda

Setelah Anda membuat aturan cache pull through, Anda dapat memvalidasi bahwa aturan tersebut berfungsi dengan baik. Saat memvalidasi aturan pull through cache, Amazon ECR membuat koneksi jaringan dengan registri upstream, memverifikasi bahwa ia dapat mengakses rahasia Secrets Manager yang berisi kredensial untuk registri upstream, dan otentikasi itu berhasil.

Untuk memvalidasi aturan pull through cache ()AWS Management Console

Langkah-langkah berikut menunjukkan cara memvalidasi aturan pull through cache menggunakan konsol Amazon ECR.

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah yang berisi aturan pull through cache untuk memvalidasi.
3. Di panel navigasi, pilih Registri pribadi, Tarik cache.
4. Pada halaman konfigurasi Pull through cache, pilih aturan pull through cache untuk memvalidasi. Kemudian, gunakan menu tarik-turun Tindakan dan pilih Lihat detail.
5. Pada halaman detail aturan cache pull through, gunakan menu tarik-turun Tindakan dan pilih Verifikasi otentikasi. Amazon ECR akan menampilkan spanduk dengan hasilnya.
6. Ulangi langkah-langkah ini untuk setiap pull through aturan cache yang ingin Anda validasi.

Untuk memvalidasi aturan pull through cache ()AWS CLI

AWS CLI Perintah [validate-pull-through-cache-rule](#) digunakan untuk memvalidasi aturan cache pull through untuk registri pribadi Amazon ECR. Contoh berikut menggunakan awalan `ecr-public` namespace. Ganti nilai itu dengan nilai awalan untuk aturan pull through cache untuk memvalidasi.

```
aws ecr validate-pull-through-cache-rule \  
  --ecr-repository-prefix ecr-public \  
  --region us-east-2
```

Dalam tanggapan, `isValid` parameter menunjukkan apakah validasi berhasil atau tidak. Jika `true`, Amazon ECR dapat mencapai registri hulu dan otentikasi berhasil. Jika `false`, ada masalah dan validasi gagal. `failureParameter` menunjukkan penyebabnya.

Menarik gambar dengan aturan cache pull through

Contoh berikut menunjukkan sintaks perintah untuk digunakan saat menarik gambar menggunakan aturan pull through cache. Jika Anda menerima kesalahan saat menarik gambar upstream menggunakan aturan cache pull through, lihat [Pemecahan masalah tarik melalui masalah cache](#) kesalahan yang paling umum dan cara mengatasinya.

Note

Contoh berikut menggunakan nilai namespace repositori Amazon ECR default yang digunakan. AWS Management Console Pastikan Anda menggunakan URI repositori pribadi Amazon ECR yang telah dikonfigurasi.

Untuk Amazon ECR Publik

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/ecr-public/repository_name/  
image_name:tag
```

Registri kontainer Kubernetes

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/kubernetes/repository_name/  
image_name:tag
```

dermaga

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/quay/repository_name/  
image_name:tag
```

Hub Docker

Untuk gambar resmi Docker Hub:

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/docker-hub/  
library/image_name:tag
```

Note

Untuk gambar resmi Docker Hub, `/library` awalan harus disertakan. Untuk semua repositori Docker Hub lainnya, Anda harus menghilangkan awalan. `/library`

Untuk semua gambar Docker Hub lainnya:

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/docker-hub/repository_name/  
image_name:tag
```

GitHub Registri Kontainer

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/github/repository_name/  
image_name:tag
```

Registri Kontainer Microsoft Azure

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/azure/repository_name/  
image_name:tag
```

Menghapus aturan pull through cache

Anda dapat menghapus aturan pull through cache untuk menghentikan perilaku caching. Menghapus aturan pull through cache tidak berpengaruh pada repositori atau gambar yang di-cache, itu hanya menghentikan perilaku caching future.

Untuk menghapus aturan pull through cache (AWS Management Console)

Untuk menghapus aturan pull through cache (AWS Management Console)

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah yang berisi aturan pull through cache untuk dihapus.
3. Di panel navigasi, pilih Registri pribadi, Tarik cache.
4. Pada halaman konfigurasi Tarik melalui cache, pilih aturan tarik cache untuk dihapus, lalu gunakan menu tarik-turun Tindakan dan pilih Aturan Hapus.
5. Di panel navigasi, pilih Registri pribadi, Izin.
6. (Opsional) Pada halaman izin registri, tinjau pernyataan kebijakan izin registri yang ada. Anda dapat menghapus pernyataan kebijakan izin registri apa pun yang terkait dengan namespace repositori untuk aturan pull through cache yang dihapus.

Untuk menghapus aturan pull through cache (AWS CLI)

Gunakan perintah berikut untuk menghapus aturan pull through cache menggunakan AWS CLI.

- [delete-pull-through-cache-aturan](#) ()AWS CLI

Contoh berikut menghapus aturan pull through cache yang menggunakan awalan `ecr-public` repositori.

```
aws ecr delete-pull-through-cache-rule \  
  --ecr-repository-prefix ecr-public \  
  --region us-east-2
```

Menyimpan kredensi repositori upstream Anda secara rahasia AWS Secrets Manager

Saat membuat aturan cache pull through untuk repositori upstream yang memerlukan otentikasi, Anda harus menyimpan kredensialnya dalam rahasia Secrets Manager. Mungkin ada biaya untuk menggunakan rahasia Secrets Manager. Untuk informasi selengkapnya, lihat [harga AWS Secrets Manager](#).

Prosedur berikut memandu Anda melalui cara membuat rahasia Secrets Manager untuk setiap repositori upstream yang didukung. Anda dapat secara opsional menggunakan alur kerja aturan create pull through cache di konsol Amazon ECR untuk membuat rahasia alih-alih membuat rahasia menggunakan konsol Secrets Manager. Untuk informasi selengkapnya, lihat [Membuat aturan pull through cache](#).


Docker Hub

Untuk membuat rahasia Secrets Manager untuk kredensial Docker Hub Anda (AWS Management Console)

1. Buka konsol Secrets Manager di <https://console.aws.amazon.com/secretsmanager/>.
2. Pilih Store a new secret (Simpan rahasia baru).
3. Pada halaman Pilih jenis rahasia, lakukan hal berikut.
 - a. Untuk Tipe rahasia, pilih Tipe rahasia lainnya.
 - b. Pada pasangan kunci/nilai, buat dua baris untuk kredensial Docker Hub Anda. Anda dapat menyimpan hingga 65536 byte secara rahasia.
 - i. Untuk pasangan kunci/nilai pertama, tentukan username sebagai kunci dan nama pengguna Docker Hub Anda sebagai nilainya.
 - ii. Untuk pasangan kunci/nilai kedua, tentukan accessToken sebagai kunci dan token akses Docker Hub Anda sebagai nilainya. Untuk informasi selengkapnya tentang


membuat token akses Docker Hub, lihat [Membuat dan mengelola token akses](#) dalam dokumentasi Docker.

- c. Untuk kunci Enkripsi, pertahankan AWS KMS key nilai `aws/secretsmanager` default lalu pilih Berikutnya. Tidak ada biaya untuk menggunakan kunci ini. Untuk informasi selengkapnya, lihat [Enkripsi dan dekripsi rahasia di Secrets Manager](#) di AWS Secrets Manager Panduan Pengguna.

 Important

Anda harus menggunakan kunci `aws/secretsmanager` enkripsi default untuk mengenkripsi rahasia Anda. Amazon ECR tidak mendukung penggunaan kunci dikelola pelanggan (CMK) untuk ini.

4. Pada halaman Konfigurasi rahasia, lakukan hal berikut.
 - a. Masukkan nama Rahasia deskriptif dan Deskripsi. Nama rahasia harus berisi 1-512 karakter Unicode dan diawali dengan `ecr-pullthroughcache/`

 Important

Amazon ECR AWS Management Console hanya menampilkan rahasia Secrets Manager dengan nama menggunakan `ecr-pullthroughcache/` awalan.

- b. (Opsional) Di bagian Tag, tambahkan tag ke rahasia Anda. Untuk menandai strategi, lihat [Tag Secrets Manager secrets](#) di Panduan AWS Secrets Manager Pengguna. Jangan menyimpan informasi sensitif dalam tag karena tidak dienkripsi.
 - c. (Opsional) Di Izin sumber daya, untuk menambahkan kebijakan sumber daya ke rahasia Anda, pilih Edit izin. Untuk informasi selengkapnya, lihat [Melampirkan kebijakan izin ke rahasia Secrets Manager](#) di Panduan AWS Secrets Manager Pengguna.
 - d. (Opsional) Dalam rahasia Replikasi, untuk mereplikasi rahasia Anda ke yang lain Wilayah AWS, pilih Replikasi rahasia. Anda dapat mereplikasi rahasia Anda sekarang atau kembali dan mereplikasi nanti. Untuk informasi selengkapnya, lihat [Mereplikasi rahasia ke Wilayah lain](#) di Panduan AWS Secrets Manager Pengguna.
 - e. Pilih Berikutnya.
5. (Opsional) Pada halaman Konfigurasi rotasi, Anda dapat mengaktifkan rotasi otomatis. Anda juga dapat mematikan rotasi untuk saat ini dan kemudian menyalakannya nanti. Untuk

informasi selengkapnya, lihat [Memutar rahasia Secrets Manager](#) di Panduan AWS Secrets Manager Pengguna. Pilih Berikutnya.

6. Pada halaman Ulasan, tinjau detail rahasia Anda, lalu pilih Store.

Secrets Manager kembali ke daftar rahasia. Jika rahasia baru Anda tidak muncul, pilih tombol refresh.

GitHub Container Registry

Untuk membuat rahasia Secrets Manager untuk kredensi Registry GitHub Container Anda ()AWS Management Console

1. Buka konsol Secrets Manager di <https://console.aws.amazon.com/secretsmanager/>.
2. Pilih Store a new secret (Simpan rahasia baru).
3. Pada halaman Pilih jenis rahasia, lakukan hal berikut.
 - a. Untuk Tipe rahasia, pilih Tipe rahasia lainnya.
 - b. Dalam pasangan kunci/nilai, buat dua baris untuk kredensial Anda GitHub . Anda dapat menyimpan hingga 65536 byte secara rahasia.
 - i. Untuk pasangan kunci/nilai pertama, tentukan username sebagai kunci dan GitHub nama pengguna Anda sebagai nilainya.
 - ii. Untuk pasangan kunci/nilai kedua, tentukan accessToken sebagai kunci dan token GitHub akses Anda sebagai nilainya. Untuk informasi selengkapnya tentang membuat token GitHub akses, lihat [Mengelola token akses pribadi Anda](#) dalam GitHub dokumentasi.
 - c. Untuk kunci Enkripsi, pertahankan AWS KMS key nilai aws/secretsmanager default lalu pilih Berikutnya. Tidak ada biaya untuk menggunakan kunci ini. Untuk informasi selengkapnya, lihat [Enkripsi dan dekripsi rahasia di Secrets Manager](#) di AWS Secrets Manager Panduan Pengguna.

Important

Anda harus menggunakan kunci aws/secretsmanager enkripsi default untuk mengenkripsi rahasia Anda. Amazon ECR tidak mendukung penggunaan kunci terkelola pelanggan (CMK) untuk ini.

4. Pada halaman Konfigurasi rahasia, lakukan hal berikut:
 - a. Masukkan nama Rahasia deskriptif dan Deskripsi. Nama rahasia harus berisi 1-512 karakter Unicode dan diawali dengan `ecr-pullthroughcache/`
 - b. (Opsional) Di bagian Tag, tambahkan tag ke rahasia Anda. Untuk menandai strategi, lihat [Tag Secrets Manager secrets](#) di Panduan AWS Secrets Manager Pengguna. Jangan menyimpan informasi sensitif dalam tag karena tidak dienkripsi.
 - c. (Opsional) Di Izin sumber daya, untuk menambahkan kebijakan sumber daya ke rahasia Anda, pilih Edit izin. Untuk informasi selengkapnya, lihat [Melampirkan kebijakan izin ke rahasia Secrets Manager](#) di Panduan AWS Secrets Manager Pengguna.
 - d. (Opsional) Dalam rahasia Replikasi, untuk mereplikasi rahasia Anda ke yang lain Wilayah AWS, pilih Replikasi rahasia. Anda dapat mereplikasi rahasia Anda sekarang atau kembali dan mereplikasi nanti. Untuk informasi selengkapnya, lihat [Mereplikasi rahasia ke Wilayah lain](#) di Panduan AWS Secrets Manager Pengguna.
 - e. Pilih Berikutnya.
5. (Opsional) Pada halaman Konfigurasi rotasi, Anda dapat mengaktifkan rotasi otomatis. Anda juga dapat mematikan rotasi untuk saat ini dan kemudian menyalakannya nanti. Untuk informasi selengkapnya, lihat [Memutar rahasia Secrets Manager](#) di Panduan AWS Secrets Manager Pengguna. Pilih Berikutnya.
 6. Pada halaman Ulasan, tinjau detail rahasia Anda, lalu pilih Store.


Secrets Manager kembali ke daftar rahasia. Jika rahasia baru Anda tidak muncul, pilih tombol refresh.

Microsoft Azure Container Registry

Untuk membuat rahasia Secrets Manager untuk kredensi Microsoft Azure Container Registry Anda ()AWS Management Console


1. Buka konsol Secrets Manager di <https://console.aws.amazon.com/secretsmanager/>.
2. Pilih Store a new secret (Simpan rahasia baru).

3. Pada halaman Pilih jenis rahasia, lakukan hal berikut.
 - a. Untuk Tipe rahasia, pilih Tipe rahasia lainnya.
 - b. Dalam pasangan kunci/nilai, buat dua baris untuk kredensial Microsoft Azure Anda. Anda dapat menyimpan hingga 65536 byte secara rahasia.
 - i. Untuk pasangan kunci/nilai pertama, tentukan `username` sebagai kunci dan nama pengguna Microsoft Azure Container Registry Anda sebagai nilainya.
 - ii. Untuk pasangan kunci/nilai kedua, tentukan `accessToken` sebagai kunci dan token akses Microsoft Azure Container Registry Anda sebagai nilainya. Untuk informasi selengkapnya tentang cara membuat token akses Microsoft Azure, lihat [Membuat token - portal](#) di dokumentasi Microsoft Azure.
 - c. Untuk kunci Enkripsi, pertahankan AWS KMS key nilai `aws/secretsmanager` default lalu pilih Berikutnya. Tidak ada biaya untuk menggunakan kunci ini. Untuk informasi selengkapnya, lihat [Enkripsi dan dekripsi rahasia di Secrets Manager](#) di AWS Secrets Manager Panduan Pengguna.

 Important

Anda harus menggunakan kunci `aws/secretsmanager` enkripsi default untuk mengenkripsi rahasia Anda. Amazon ECR tidak mendukung penggunaan kunci terkelola pelanggan (CMK) untuk ini.

4. Pada halaman Konfigurasi rahasia, lakukan hal berikut:
 - a. Masukkan nama Rahasia deskriptif dan Deskripsi. Nama rahasia harus berisi 1-512 karakter Unicode dan diawali dengan `ecr-pullthroughcache/`

 Important

Amazon ECR AWS Management Console hanya menampilkan rahasia Secrets Manager dengan nama menggunakan `ecr-pullthroughcache/` awalan.

- b. (Opsional) Di bagian Tag, tambahkan tag ke rahasia Anda. Untuk menandai strategi, lihat [Tag Secrets Manager secrets](#) di Panduan AWS Secrets Manager Pengguna. Jangan menyimpan informasi sensitif dalam tag karena tidak dienkripsi.

- c. (Opsional) Di Izin sumber daya, untuk menambahkan kebijakan sumber daya ke rahasia Anda, pilih Edit izin. Untuk informasi selengkapnya, lihat [Melampirkan kebijakan izin ke rahasia Secrets Manager](#) di Panduan AWS Secrets Manager Pengguna.
 - d. (Opsional) Dalam rahasia Replikasi, untuk mereplikasi rahasia Anda ke yang lain Wilayah AWS, pilih Replikasi rahasia. Anda dapat mereplikasi rahasia Anda sekarang atau kembali dan mereplikasi nanti. Untuk informasi selengkapnya, lihat [Mereplikasi rahasia ke Wilayah lain](#) di Panduan AWS Secrets Manager Pengguna.
 - e. Pilih Berikutnya.
5. (Opsional) Pada halaman Konfigurasi rotasi, Anda dapat mengaktifkan rotasi otomatis. Anda juga dapat mematikan rotasi untuk saat ini dan kemudian menyalakannya nanti. Untuk informasi selengkapnya, lihat [Memutar rahasia Secrets Manager](#) di Panduan AWS Secrets Manager Pengguna. Pilih Berikutnya.
 6. Pada halaman Ulasan, tinjau detail rahasia Anda, lalu pilih Store.

Secrets Manager kembali ke daftar rahasia. Jika rahasia baru Anda tidak muncul, pilih tombol refresh.

Pemecahan masalah tarik melalui masalah cache

Saat menarik gambar upstream menggunakan aturan pull through cache, berikut ini adalah kesalahan paling umum yang mungkin Anda terima.

Repositori tidak ada

Kesalahan yang menunjukkan bahwa repositori tidak ada paling sering disebabkan oleh repositori yang tidak ada di registri pribadi Amazon ECR Anda atau `ecr:CreateRepository` izin yang tidak diberikan kepada prinsipal IAM yang menarik gambar hulu. Untuk mengatasi kesalahan ini, Anda harus memverifikasi bahwa URI repositori dalam perintah tarik Anda sudah benar, izin IAM yang diperlukan diberikan kepada prinsipal IAM yang menarik gambar upstream, atau bahwa repositori untuk gambar upstream yang akan didorong dibuat di registri pribadi Amazon ECR Anda sebelum melakukan penarikan gambar upstream. Untuk informasi selengkapnya tentang izin IAM yang diperlukan, lihat [Izin IAM yang diperlukan](#)

Berikut ini adalah contoh kesalahan ini.

```
Error response from daemon: repository 111122223333.dkr.ecr.us-east-1.amazonaws.com/
ecr-public/amazonlinux/amazonlinux not found: name unknown: The repository with
```

```
name 'ecr-public/amazonlinux/amazonlinux' does not exist in the registry with id '111122223333'
```

Gambar yang diminta tidak ditemukan

Kesalahan yang menunjukkan bahwa gambar tidak dapat ditemukan paling sering disebabkan oleh gambar yang tidak ada di registri hulu atau `ecr:BatchImportUpstreamImage` izin yang tidak diberikan kepada kepala sekolah IAM yang menarik gambar hulu tetapi repositori sudah dibuat di registri pribadi Amazon ECR Anda. Untuk mengatasi kesalahan ini, Anda harus memverifikasi gambar hulu dan nama tag gambar sudah benar dan bahwa itu ada dan izin IAM yang diperlukan diberikan kepada prinsipal IAM yang menarik gambar hulu. Untuk informasi selengkapnya tentang izin IAM yang diperlukan, lihat [Izin IAM yang diperlukan](#)

Berikut ini adalah contoh kesalahan ini.

```
Error response from daemon: manifest for 111122223333.dkr.ecr.us-east-1.amazonaws.com/ecr-public/amazonlinux/amazonlinux:latest not found: manifest unknown: Requested image not found
```

403 Terlarang saat menarik dari repositori Docker Hub

Saat menarik dari repositori Docker Hub yang ditandai sebagai Gambar Resmi Docker, Anda harus menyertakan URI yang `/library/` Anda gunakan. Misalnya, `aws_account_id.dkr.ecr.region.amazonaws.com/docker-hub/library/image_name:tag`. Jika Anda menghilangkan gambar `/library/` Resmi untuk Docker Hub, 403 Forbidden kesalahan akan dikembalikan saat Anda mencoba menarik gambar menggunakan aturan cache tarik. Untuk informasi selengkapnya, lihat [Menarik gambar dengan aturan cache pull through](#).

Berikut ini adalah contoh kesalahan ini.

```
Error response from daemon: failed to resolve reference "111122223333.dkr.ecr.us-west-2.amazonaws.com/docker-hub/amazonlinux:2023": pulling from host 111122223333.dkr.ecr.us-west-2.amazonaws.com failed with status code [manifests 2023]: 403 Forbidden
```

Menghapus citra

Jika Anda selesai menggunakan citra, Anda dapat menghapusnya dari repositori. Jika Anda selesai dengan repositori, Anda dapat menghapus seluruh repositori dan semua citra di dalamnya. Untuk informasi selengkapnya, lihat [Menghapus repositori pribadi](#).

Sebagai alternatif untuk menghapus citra secara manual, Anda dapat membuat kebijakan siklus hidup repositori yang memberikan kontrol lebih atas manajemen siklus hidup citra dalam repositori Anda. Kebijakan siklus hidup mengotomatisasi proses ini untuk Anda. Untuk informasi selengkapnya, lihat [Kebijakan siklus hidup](#).

Untuk menghapus citra (AWS Management Console)

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah yang berisi citra yang akan dihapus.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih repositori yang berisi citra yang akan dihapus.
5. Pada halaman Repositori: **repository_name**, pilih kotak di sebelah kiri citra untuk dihapus kemudian pilih Hapus.
6. Di kotak dialog Hapus citra, verifikasi bahwa citra yang dipilih perlu dihapus kemudian pilih Hapus.

Untuk menghapus citra (AWS CLI)

1. Masukkan citra di repositori Anda. Citra yang ditandai akan memiliki kedua digest citra serta daftar tanda terkait. Citra yang tidak tertagged hanya akan memiliki digest citra.

```
aws ecr list-images \  
  --repository-name my-repo
```

2. (Opsional) Hapus tanda yang tidak diinginkan untuk citra tersebut dengan menentukan tanda yang terkait dengan citra yang ingin Anda hapus. Bila tanda terakhir dihapus dari sebuah citra, maka citra tersebut juga akan dihapus.

```
aws ecr batch-delete-image \  
  --repository-name my-repo \  
  --image-ids imageTag=tag1 imageTag=tag2
```

3. Hapus citra yang ditandai atau tidak ditandai dengan menentukan digest citra. Ketika Anda menghapus citra dengan referensi digestnya, maka citra dan semua tandanya akan dihapus.

```
aws ecr batch-delete-image \
  --repository-name my-repo \
  --image-ids imageDigest=sha256:4f70ef7a4d29e8c0c302b13e25962d8f7a0bd304EXAMPLE
```

Untuk menghapus beberapa citra, Anda dapat menentukan beberapa tanda citra atau digest citra dalam permintaan tersebut.

```
aws ecr batch-delete-image \
  --repository-name my-repo \
  --image-ids imageDigest=sha256:4f70ef7a4d29e8c0c302b13e25962d8f7a0bd304EXAMPLE
  imageDigest=sha256:f5t0e245ssffc302b13e25962d8f7a0bd304EXAMPLE
```

Menandai ulang citra

Dengan citra Docker Image Manifest V2 Skema 2, Anda dapat menggunakan opsi `--image-tag` perintah `put-image` untuk menandai ulang citra yang tersedia. Anda dapat menandai ulang tanpa menarik atau mendorong citra dengan Docker. Untuk citra yang lebih besar, proses ini menghemat sejumlah besar bandwidth jaringan dan waktu yang diperlukan untuk menandai ulang citra.

Untuk menandai ulang citra (AWS CLI)

Untuk menandai ulang gambar dengan AWS CLI

1. Gunakan `batch-get-image` perintah untuk mendapatkan manifes gambar untuk gambar untuk retag dan menulis ke file. *Dalam contoh ini, manifes untuk gambar dengan tag, latest, dalam repositori, amazonlinux, ditulis ke variabel lingkungan bernama MANIFEST.*

```
MANIFEST=$(aws ecr batch-get-image --repository-name amazonlinux --image-ids
  imageTag=latest --output text --query 'images[].imageManifest')
```

2. Gunakan opsi `--image-tag` perintah `put-image` untuk menempatkan manifes citra ke Amazon ECR dengan tanda baru. Dalam contoh ini, citra ditandai sebagai *2017.03*.

Note

Jika `--image-tag` opsi tidak tersedia di versi Anda AWS CLI, tingkatkan ke versi terbaru. Untuk informasi selengkapnya, lihat [Menginstal AWS Command Line Interface](#) dalam Panduan Pengguna AWS Command Line Interface .

```
aws ecr put-image --repository-name amazonlinux --image-tag 2017.03 --image-manifest "$MANIFEST"
```

3. Verifikasi bahwa tanda citra baru Anda terpasang pada citra Anda. Pada output berikut, citra memiliki tanda `latest` dan `2017.03`.

```
aws ecr describe-images --repository-name amazonlinux
```

Outputnya adalah sebagai berikut:

```
{
  "imageDetails": [
    {
      "imageSizeInBytes": 98755613,
      "imageDigest":
"sha256:8d00af8f076eb15a33019c2a3e7f1f655375681c4e5be157a26EXAMPLE",
      "imageTags": [
        "latest",
        "2017.03"
      ],
      "registryId": "aws_account_id",
      "repositoryName": "amazonlinux",
      "imagePushedAt": 1499287667.0
    }
  ]
}
```


Untuk menandai ulang citra (AWS Tools for Windows PowerShell)

Untuk menandai ulang gambar dengan AWS Tools for Windows PowerShell

- Gunakan cmdlet `Get-ECRIImageBatch` untuk mendapatkan deskripsi citra yang akan ditandai ulang dan tuliskan ke variabel lingkungan. Dalam contoh ini, citra dengan tanda, *terbaru*, dalam repositori, *amazonlinux*, ditulis ke variabel lingkungan, *\$Image*.

Note

Jika Anda tidak memiliki cmdlet `Get-ECRIImageBatch` tersedia di sistem Anda, lihat [Menyiapkan AWS Tools for Windows PowerShell](#) dalam Panduan Pengguna AWS Tools for Windows PowerShell .

```
$Image = Get-ECRIImageBatch -ImageId @{ imageTag="latest" } -
RepositoryName amazonlinux
```

- Menulis manifes dari citra ke variabel lingkungan *Manifest\$*.

```
$Manifest = $Image.Images[0].ImageManifest
```

- Gunakan opsi `-ImageTag` cmdlet `Write-ECRIImage` untuk menempatkan manifes citra ke Amazon ECR dengan tanda baru. Dalam contoh ini, citra ditandai sebagai *2017.09*.

```
Write-ECRIImage -RepositoryName amazonlinux -ImageManifest $Manifest -
ImageTag 2017.09
```

- Verifikasi bahwa tanda citra baru Anda terpasang pada citra Anda. Pada output berikut, citra memiliki tanda `latest` dan `2017.09`.

```
Get-ECRIImage -RepositoryName amazonlinux
```

Outputnya adalah sebagai berikut:

```
ImageDigest                                     ImageTag
-----
sha256:359b948ea8866817e94765822787cd482279eed0c17bc674a7707f4256d5d497 latest
sha256:359b948ea8866817e94765822787cd482279eed0c17bc674a7707f4256d5d497 2017.09
```

Replikasi citra pribadi

Anda dapat mengonfigurasi registri pribadi Amazon ECR Anda untuk mendukung replikasi repositori Anda. Amazon ECR mendukung replikasi lintas wilayah dan lintas akun. Agar replikasi lintas akun terjadi, akun tujuan harus mengonfigurasi kebijakan izin registri agar replikasi dari registri sumber terjadi. Untuk informasi selengkapnya, lihat [Izin Registri Pribadi](#).

Topik

- [Pertimbangan untuk replikasi citra pribadi](#)
- [mengonfigurasi replikasi citra pribadi](#)
- [Melihat status replikasi](#)

Pertimbangan untuk replikasi citra pribadi

Hal-hal berikut ini harus dipertimbangkan ketika menggunakan replikasi citra pribadi.

- Hanya konten repositori yang didorong ke repositori setelah replikasi dikonfigurasi direplikasi. Konten apa pun yang sudah ada sebelumnya dalam repositori tidak direplikasi. Setelah replikasi dikonfigurasi untuk repositori, Amazon ECR menjaga tujuan dan sumber disinkronkan.
- Nama repositori akan tetap sama di seluruh Wilayah dan akun saat replikasi telah terjadi. Amazon ECR tidak mendukung perubahan nama repositori selama replikasi.
- Pertama kali Anda mengonfigurasi registri pribadi untuk replikasi, Amazon ECR membuat peran IAM terkait layanan atas nama Anda. Peran IAM terkait layanan memberikan layanan replikasi Amazon ECR izin yang diperlukan untuk membuat repositori dan mereplikasi gambar di registri Anda. Untuk informasi selengkapnya, lihat [Menggunakan Peran Terkait Layanan untuk Amazon ECR](#).
- Agar replikasi lintas akun terjadi, tujuan registri pribadi harus memberikan izin untuk mengizinkan registri sumber mereplikasi gambarnya. Ini dilakukan dengan menetapkan kebijakan izin registri pribadi. Untuk informasi selengkapnya, lihat [Izin Registri Pribadi](#).
- Jika kebijakan izin untuk registri pribadi diubah untuk menghapus izin, setiap replikasi yang sedang berlangsung yang sebelumnya diberikan dapat diselesaikan.
- Agar replikasi lintas wilayah terjadi, akun sumber dan tujuan harus ikut serta dalam Wilayah sebelum tindakan replikasi apa pun yang terjadi di dalam atau ke Wilayah tersebut. Untuk informasi selengkapnya, lihat [Mengelola Wilayah AWS](#) di Referensi Umum Amazon Web Services.

- Replikasi Lintas Wilayah tidak didukung antar AWS partisi. Misalnya, repositori di `us-west-2` dapat direplikasi ke `cn-north-1`. Untuk informasi selengkapnya tentang AWS partisi, lihat format [ARN](#) di Referensi Umum AWS.
- Konfigurasi replikasi untuk registri pribadi dapat berisi hingga 25 tujuan unik di semua aturan, dengan maksimal 10 aturan total. Setiap aturan dapat berisi hingga 100 filter. Ini memungkinkan untuk menentukan aturan terpisah untuk repositori yang berisi gambar yang digunakan untuk produksi dan pengujian, misalnya.
- Konfigurasi replikasi mendukung pemfilteran repositori mana dalam registri pribadi direplikasi dengan menentukan awalan repositori. Sebagai contoh, lihat [Contoh: Mengonfigurasi replikasi lintas wilayah menggunakan filter repositori](#).
- Sebuah tindakan replikasi hanya terjadi sekali per dorongan citra. Sebagai contoh, jika Anda mengonfigurasi replikasi lintas wilayah dari `us-west-2` ke `us-east-1` dan dari `us-east-1` ke `us-east-2`, citra yang didorong ke `us-west-2` bereplikasi hanya ke `us-east-1`, ia tidak mereplikasi lagi ke `us-east-2`. Perilaku ini berlaku untuk replikasi lintas wilayah maupun lintas akun.
- Mayoritas gambar mereplikasi dalam waktu kurang dari 30 menit, tetapi dalam kasus yang jarang terjadi replikasi mungkin memakan waktu lebih lama.
- Replikasi registri tidak melakukan tindakan menghapus apa pun. Citra dan repositori yang direplikasi dapat dihapus secara manual saat tidak lagi digunakan.
- Kebijakan repositori, termasuk kebijakan IAM, dan kebijakan siklus hidup tidak direplikasi dan tidak memiliki efek apa pun selain pada repositori yang ditetapkan untuknya.
- Pengaturan repositori tidak direplikasi. Pengaturan ketetapan tanda, pemindaian citra, dan enkripsi dinonaktifkan secara default pada semua repositori yang dibuat karena tindakan replikasi. Pengaturan ketetapan tanda dan pemindaian citra dapat diubah setelah repositori dibuat. Namun, pengaturan hanya berlaku untuk citra yang didorong setelah pengaturan berubah.
- Jika ketetapan tanda diaktifkan pada repositori dan sebuah citra direplikasi yang menggunakan tanda yang sama sebagai citra yang ada, citra direplikasi tetapi tidak akan berisi duplikasi tandanya. Hal ini dapat mengakibatkan citra yang tidak ditandai.

mengonfigurasi replikasi citra pribadi

Pengaturan replikasi untuk registri pribadi Anda dikonfigurasi secara terpisah untuk setiap Wilayah. Gunakan langkah-langkah berikut untuk mengkonfigurasi replikasi untuk registri pribadi Anda menggunakan salah satu AWS Management Console atau AWS CLI. Untuk contoh bagaimana replikasi umumnya digunakan, lihat [Contoh replikasi citra pribadi](#).

Untuk mengonfigurasi pengaturan replikasi registri (AWS Management Console)

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah untuk mengonfigurasi pengaturan replikasi registri.
3. Di panel navigasi, pilih Registri pribadi.
4. Pada halaman registri pribadi, pada bagian Replikasi, pilih Edit.
5. Pada halaman Replikasi, pilih Tambahkan aturan replikasi.
6. Pada halaman Jenis tujuan, pilih apakah akan mengaktifkan replikasi lintas wilayah, replikasi lintas akun, atau keduanya, lalu pilih Berikutnya.
7. Jika replikasi lintas wilayah diaktifkan, maka untuk Konfigurasi wilayah tujuan, pilih satu atau beberapa wilayah Tujuan, lalu pilih Berikutnya.
8. Jika replikasi lintas akun diaktifkan, maka untuk replikasi Cross-account, pilih pengaturan replikasi lintas akun untuk registri. Untuk akun Tujuan, masukkan ID akun untuk akun tujuan dan satu atau beberapa wilayah Tujuan untuk direplikasi. Pilih Akun tujuan+untuk mengonfigurasi akun tambahan sebagai tujuan replikasi.

Important

Agar replikasi lintas akun terjadi, akun tujuan harus mengonfigurasi kebijakan izin registri untuk memungkinkan replikasi terjadi. Untuk informasi selengkapnya, lihat [Izin Registri Pribadi](#).

9. (Opsional) Pada halaman Tambahkan filter, tentukan satu atau beberapa filter untuk aturan replikasi lalu pilih Tambah. Ulangi langkah ini untuk setiap filter yang ingin Anda kaitkan dengan tindakan replikasi. Filter harus ditentukan sebagai awalan nama repositori. Jika tidak ada filter yang ditambahkan, isi semua repositori direplikasi. Pilih Berikutnya setelah semua filter telah ditambahkan.
10. Pada halaman Tinjau dan kirim, tinjau konfigurasi aturan replikasi lalu pilih Aturan Kirim.

Untuk mengonfigurasi pengaturan replikasi registri (AWS CLI)

1. Buat file JSON yang berisi aturan replikasi untuk menentukan registri Anda. Konfigurasi replikasi dapat berisi hingga 10 aturan, dengan hingga 25 tujuan unik di semua aturan dan 100 filter per setiap aturan. Untuk mengonfigurasi replikasi lintas wilayah dalam akun Anda sendiri, Anda menentukan ID akun Anda sendiri. Untuk contoh lainnya, lihat [Contoh replikasi citra pribadi](#).

```
{
  "rules": [{
    "destinations": [{
      "region": "destination_region",
      "registryId": "destination_accountId"
    }],
    "repositoryFilters": [{
      "filter": "repository_prefix_name",
      "filterType": "PREFIX_MATCH"
    }]
  }]
}
```

2. Membuat konfigurasi replikasi untuk registri Anda.

```
aws ecr put-replication-configuration \
  --replication-configuration file://replication-settings.json \
  --region us-west-2
```

3. Konfirmasikan pengaturan registri Anda.

```
aws ecr describe-registry \
  --region us-west-2
```

Contoh replikasi citra pribadi

Contoh berikut menunjukkan kasus penggunaan umum untuk replikasi gambar pribadi.

Contoh: mengonfigurasi replikasi lintas wilayah untuk satu Wilayah tujuan

Berikut ini adalah contoh untuk mengonfigurasi replikasi lintas wilayah dalam satu registri. Contoh ini mengasumsikan bahwa ID akun Anda adalah 111122223333 dan bahwa Anda menentukan konfigurasi replikasi ini di wilayah selain `us-west-2`.

```
{
  "rules": [
    {
      "destinations": [
        {
          "region": "us-west-2",
```

```

        "registryId": "111122223333"
      }
    ]
  }
}

```

Contoh: Mengonfigurasi replikasi lintas wilayah menggunakan filter repositori

Berikut ini menunjukkan contoh untuk mengkonfigurasi replikasi lintas wilayah untuk repositori yang cocok dengan nilai nama awalan. Contoh ini mengasumsikan ID akun Anda 111122223333 dan bahwa Anda menentukan konfigurasi replikasi ini di Wilayah selain `us-west-1` dan memiliki repositori dengan awalan `prod`

```

{
  "rules": [{
    "destinations": [{
      "region": "us-west-1",
      "registryId": "111122223333"
    }],
    "repositoryFilters": [{
      "filter": "prod",
      "filterType": "PREFIX_MATCH"
    }]
  }]
}

```

Contoh: mengonfigurasi replikasi lintas wilayah untuk beberapa Wilayah tujuan

Berikut ini adalah contoh untuk mengonfigurasi replikasi lintas wilayah dalam satu registri. Contoh ini mengasumsikan bahwa ID akun Anda adalah 111122223333 dan bahwa Anda menentukan konfigurasi replikasi ini di wilayah selain `us-west-1` atau `us-west-2`.

```

{
  "rules": [
    {
      "destinations": [
        {
          "region": "us-west-1",
          "registryId": "111122223333"
        },
        {

```

```
        "region": "us-west-2",
        "registryId": "111122223333"
      }
    ]
  }
}
```

Contoh: mengonfigurasi replikasi lintas akun

Berikut ini adalah contoh untuk mengonfigurasi replikasi lintas akun untuk registri Anda. Contoh ini mengonfigurasi replikasi ke akun 444455556666 dan ke Wilayah us-west-2.

⚠ Important

Agar replikasi lintas akun terjadi, akun tujuan harus mengonfigurasi kebijakan izin registri untuk memungkinkan replikasi terjadi. Untuk informasi selengkapnya, lihat [Izin Registri Pribadi](#).

```
{
  "rules": [
    {
      "destinations": [
        {
          "region": "us-west-2",
          "registryId": "444455556666"
        }
      ]
    }
  ]
}
```

Contoh: Menentukan beberapa aturan dalam konfigurasi

Berikut ini menunjukkan contoh untuk mengkonfigurasi beberapa aturan replikasi untuk registri Anda. Contoh ini mengonfigurasi replikasi untuk akun **111122223333** dengan satu aturan yang mereplikasi repositori dengan awalan ke Wilayah dan repositori dengan awalan prod ke Wilayah. us-west-2 test us-east-2 Konfigurasi replikasi dapat berisi hingga 10 aturan, dengan setiap aturan menentukan hingga 25 tujuan.

```
{
  "rules": [{
    "destinations": [{
      "region": "us-west-2",
      "registryId": "111122223333"
    }],
    "repositoryFilters": [{
      "filter": "prod",
      "filterType": "PREFIX_MATCH"
    }]
  }],
  {
    "destinations": [{
      "region": "us-east-2",
      "registryId": "111122223333"
    }],
    "repositoryFilters": [{
      "filter": "test",
      "filterType": "PREFIX_MATCH"
    }]
  }
]
}
```

Melihat status replikasi

Setelah Anda mengkonfigurasi registri pribadi Anda untuk replikasi, Anda dapat melihat status replikasi untuk isi repositori Anda. Status replikasi gambar kontainer individu dapat dilihat dengan melakukan kueri menggunakan tag gambar atau intisari gambar.

Memeriksa status replikasi () AWS Management Console

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah yang merupakan sumber registri yang direplikasi Anda.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih repositori untuk memeriksa status replikasi.
5. Pada halaman detail repositori, pilih tag Gambar untuk memeriksa status replikasi.
6. Untuk status replikasi Gambar, verifikasi status replikasi. Anda dapat melihat status replikasi berdasarkan tag gambar atau intisari gambar.

Memeriksa status replikasi () AWS CLI

- Status replikasi isi repositori dapat dilihat berdasarkan tag gambar menggunakan perintah berikut.

```
aws ecr describe-image-replication-status \  
  --repository-name repository_name \  
  --image-id imageTag=image_tag \  
  --region us-west-2
```

- Status replikasi isi repositori dapat dilihat berdasarkan intisari gambar menggunakan perintah berikut.

```
aws ecr describe-image-replication-status \  
  --repository-name repository_name \  
  --image-id imageDigest=image_digest \  
  --region us-west-2
```

Kebijakan siklus hidup

Kebijakan siklus hidup Amazon ECR memberikan kontrol lebih atas manajemen siklus hidup citra dalam repositori pribadi. Sebuah kebijakan siklus hidup berisi satu atau lebih aturan, di mana setiap aturan mendefinisikan sebuah tindakan untuk Amazon ECR. Ini menyediakan cara untuk mengotomatiskan pembersihan gambar kontainer Anda dengan kedaluwarsa gambar berdasarkan usia atau hitungan. Anda harus mengharapkan bahwa gambar menjadi kedaluwarsa dalam waktu 24 jam setelah memenuhi kriteria kedaluwarsa sesuai kebijakan siklus hidup Anda. Saat Amazon ECR melakukan tindakan berdasarkan kebijakan siklus hidup, tindakan ini akan diambil sebagai peristiwa di AWS CloudTrail Untuk informasi selengkapnya, lihat [Mencatat tindakan Amazon ECR dengan AWS CloudTrail](#).

Cara kerja kebijakan siklus hidup

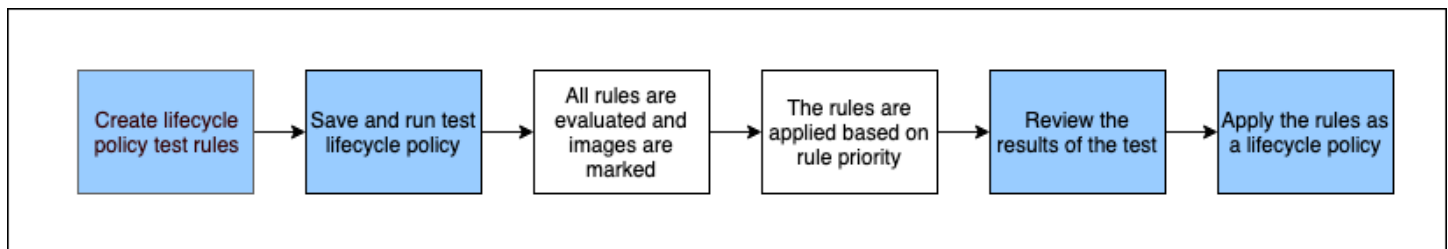
Kebijakan siklus hidup terdiri dari satu atau lebih aturan yang menentukan citra mana dalam repositori yang akan kedaluwarsa. Saat mempertimbangkan penggunaan kebijakan siklus hidup, penting untuk menggunakan pratinjau kebijakan siklus hidup untuk mengonfirmasi citra mana yang kebijakan siklus hidupnya kedaluwarsa sebelum menerapkannya ke repositori. Setelah kebijakan siklus hidup diterapkan ke repositori, Anda akan mengharapkan bahwa gambar menjadi kedaluwarsa dalam waktu 24 jam setelah memenuhi kriteria kedaluwarsa. Saat Amazon ECR melakukan tindakan

berdasarkan kebijakan siklus hidup, tindakan ini akan diambil sebagai peristiwa di AWS CloudTrail. Untuk informasi selengkapnya, lihat [Mencatat tindakan Amazon ECR dengan AWS CloudTrail](#).

Note

Jika Anda menggunakan replikasi Amazon ECR untuk membuat salinan repositori di berbagai Wilayah atau akun, ingatlah bahwa kebijakan siklus hidup hanya dapat mengambil tindakan pada repositori di Wilayah tempat ia dibuat. Oleh karena itu, jika replikasi diaktifkan, Anda mungkin ingin membuat kebijakan siklus hidup di setiap Wilayah dan akun tempat Anda mereplikasi repositori Anda.

Diagram berikut menunjukkan alur kerja kebijakan siklus hidup.



1. Buat satu atau lebih aturan pengujian.
2. Simpan aturan pengujian dan jalankan pratinjau.
3. Evaluator kebijakan siklus hidup membaca semua aturan dan menandai citra yang terpengaruh oleh setiap aturan.
4. Penilai kebijakan siklus hidup kemudian menerapkan aturan, berdasarkan prioritas aturan, dan menampilkan citra mana dalam repositori yang ditetapkan sebagai kedaluwarsa.
5. Tinjau hasil pengujian untuk memastikan bahwa citra yang ditandai kedaluwarsa sesuai keinginan Anda.
6. Terapkan aturan pengujian sebagai kebijakan siklus hidup untuk repositori.
7. Setelah kebijakan siklus hidup dibuat, Anda akan mengharapkan bahwa gambar akan kedaluwarsa dalam waktu 24 jam setelah memenuhi kriteria kedaluwarsa.

Aturan evaluasi kebijakan siklus hidup

Evaluator kebijakan siklus hidup bertanggung jawab menguraikan JSON plaintext dari kebijakan siklus hidup, mengevaluasi semua aturan, kemudian menerapkan aturan tersebut berdasarkan

prioritas aturan pada citra dalam repositori. Berikut ini penjelasan logika evaluator kebijakan siklus hidup secara lebih rinci. Sebagai contoh, lihat [Contoh kebijakan siklus hidup](#).

- Semua aturan dievaluasi pada saat yang sama tanpa memperhatikan prioritas aturan. Setelah dievaluasi, semua aturan kemudian diterapkan berdasarkan prioritas aturan.
- Sebuah citra kedaluwarsa oleh tepat satu atau nol aturan.
- Sebuah citra yang sesuai dengan persyaratan penandaan aturan tidak dapat kedaluwarsa oleh aturan dengan prioritas yang lebih rendah.
- Aturan tidak pernah dapat menandai citra yang ditandai dengan aturan prioritas yang lebih tinggi, tetapi masih dapat mengidentifikasi citra tersebut seolah-olah belum kedaluwarsa.
- Kumpulan aturan harus berisi sekumpulan prefiks tanda yang unik.
- Hanya satu aturan yang diizinkan untuk memilih citra yang tidak ditandai.
- Jika gambar direferensikan oleh daftar manifes, gambar tidak dapat kedaluwarsa tanpa daftar manifes dihapus terlebih dahulu.
- Kedaluwarsa selalu dipesan oleh `pushed_at_time`, dan selalu dilakukan pada citra yang lebih lama sebelum citra yang lebih baru.
- Aturan kebijakan siklus hidup dapat menentukan salah satu `tagPatternList` atau `tagPrefixList`, tetapi tidak keduanya. Namun, kebijakan siklus hidup mungkin berisi beberapa aturan di mana aturan yang berbeda menggunakan daftar pola dan awalan.
- `tagPrefixListParameter` `tagPatternList` atau hanya dapat digunakan jika `tagStatus` `adatagged`.
- Saat menggunakan `tagPatternList`, gambar berhasil dicocokkan jika cocok dengan filter wildcard. Misalnya, jika filter diterapkan, itu akan cocok dengan repositori yang namanya dimulai dengan `prod` seperti `prod`, `prod1`, atau `prod* production-team1`. Demikian pula, jika filter `*prod*` diterapkan, itu akan cocok dengan repositori yang namanya berisi `prod` seperti `repo-production` atau `prod-team`.

 Important

Ada batas maksimum empat wildcard (*) per string. Misalnya, `["*test*1*2*3", "test*1*2*3*"]` valid tetapi `["test*1*2*3*4*5*6"]` tidak valid.

- Saat menggunakan `tagPrefixList`, gambar berhasil dicocokkan jika semua tag dalam `tagPrefixList` nilai dicocokkan dengan salah satu tag gambar.
- Parameter `countUnit` hanya digunakan jika `countType` adalah `sinceImagePushed`.

- Dengan `countType = imageCountMoreThan`, citra diurutkan dari yang termuda hingga tertua berdasarkan `pushed_at_time` dan kemudian semua citra yang lebih besar dari hitungan yang ditentukan menjadi kedaluwarsa.
- Dengan `countType = sinceImagePushed`, semua citra yang `pushed_at_time` lebih tua dari jumlah hari yang ditentukan berdasarkan `countNumber` menjadi kedaluwarsa.

Templat kebijakan siklus hidup

Isi kebijakan siklus hidup Anda dievaluasi sebelum dikaitkan dengan repositori. Berikut ini adalah templat sintaks JSON untuk kebijakan siklus hidup. Untuk contoh kebijakan siklus hidup, lihat [Contoh kebijakan siklus hidup](#).

```
{
  "rules": [
    {
      "rulePriority": integer,
      "description": "string",
      "selection": {
        "tagStatus": "tagged"|"untagged"|"any",
        "tagPatternList": list<string>,
        "tagPrefixList": list<string>,
        "countType": "imageCountMoreThan"|"sinceImagePushed",
        "countUnit": "string",
        "countNumber": integer
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}
```

Parameter kebijakan siklus hidup

Kebijakan siklus hidup dibagi ke dalam beberapa bagian berikut:

Topik

- [Prioritas aturan](#)
- [Deskripsi](#)

- [Status tanda](#)
- [Daftar pola tag](#)
- [Daftar prefiks tanda](#)
- [Jenis hitungan](#)
- [Unit hitungan](#)
- [Jumlah hitungan](#)
- [Tindakan](#)

Prioritas aturan

`rulePriority`

Jenis: integer

Wajib: ya

Menetapkan urutan aturan yang diterapkan, terendah ke tertinggi. Aturan kebijakan siklus hidup dengan prioritas 1 akan diterapkan pertama, aturan dengan prioritas 2 diterapkan berikutnya, dan seterusnya. Ketika Anda menambahkan aturan ke kebijakan siklus hidup, Anda harus memberi nilai unik pada masing-masing aturan untuk `rulePriority`. Nilai tidak perlu berurutan di sepanjang aturan dalam kebijakan. Aturan dengan nilai `tagStatus` any harus memiliki nilai tertinggi untuk `rulePriority` dan dievaluasi terakhir.

Deskripsi

`description`

Jenis: string

Wajib: tidak

(Opsional) Menjelaskan tujuan sebuah aturan dalam kebijakan siklus hidup.

Status tanda

`tagStatus`

Jenis: string

Wajib: ya

Menentukan apakah aturan kebijakan siklus hidup yang Anda tambahkan menentukan tanda untuk sebuah citra. Pilihan yang dapat diterima adalah `tagged`, `untagged`, atau `any`. Jika Anda menentukan aturan `any`, maka semua citra dievaluasi oleh aturan tersebut. Jika Anda menentukan `tagged`, maka Anda juga harus menentukan nilai `tagPrefixList`. Jika Anda menentukan `untagged`, maka Anda harus menghilangkan `tagPrefixList`.

Daftar pola tag

`tagPatternList`

Jenis: daftar [string]

Wajib: ya, jika `tagStatus` disetel ke `tag` dan `tagPrefixList` tidak ditentukan

Saat membuat kebijakan siklus hidup untuk gambar yang diberi tag, sebaiknya gunakan a untuk menentukan tag yang akan `tagPatternList` kedaluwarsa. Anda menentukan daftar pola tag gambar yang dipisahkan koma yang mungkin berisi wildcard (*) untuk mengambil tindakan dengan kebijakan siklus hidup Anda. Misalnya, jika gambar Anda ditandai sebagaiprod,, prod1prod2, dan seterusnya, Anda akan menggunakan daftar pola tag `prod*` untuk menentukan semuanya. Jika Anda menentukan beberapa tanda, hanya citra dengan semua tanda yang ditentukan yang dipilih.

Important

Ada batas maksimum empat wildcard (*) per string. Misalnya, `["*test*1*2*3", "test*1*2*3*"]` valid tetapi `["test*1*2*3*4*5*6"]` tidak valid.

Daftar prefiks tanda

`tagPrefixList`

Jenis: daftar [string]

Wajib: ya, jika `tagStatus` disetel ke `tag` dan `tagPatternList` tidak ditentukan

Hanya digunakan jika Anda menentukan `"tagStatus": "tagged"` dan Anda tidak menentukan. `tagPatternList` Anda harus menentukan daftar prefiks tanda citra yang

dipisahkan koma untuk mengambil tindakan dengan kebijakan siklus hidup Anda. Misalnya, jika citra Anda ditandai sebagai `prod`, `prod1`, `prod2`, dan seterusnya, Anda akan menggunakan prefiks `prod` untuk menentukan semuanya. Jika Anda menentukan beberapa tanda, hanya citra dengan semua tanda yang ditentukan yang dipilih.

Jenis hitungan

`countType`

Jenis: `string`

Wajib: ya

Tentukan jenis jumlah untuk diterapkan pada citra.

Jika `countType` diatur ke `imageCountMoreThan`, Anda juga menentukan `countNumber` untuk membuat aturan yang menetapkan batas pada jumlah citra yang ada di repositori Anda. Jika `countType` diatur ke `sinceImagePushed`, Anda juga menentukan `countUnit` dan `countNumber` untuk menentukan batas waktu pada citra yang ada di repositori Anda.

Unit hitungan

`countUnit`

Jenis: `string`

Wajib: ya, hanya jika `countType` diatur ke `sinceImagePushed`

Tentukan unit hitungan `days` untuk menunjukkan bahwa sebagai unit waktu, di samping `countNumber`, yang merupakan jumlah hari.

Ini harus ditentukan hanya ketika `countType` adalah `sinceImagePushed`; kesalahan akan terjadi jika Anda menentukan unit hitungan saat `countType` adalah nilai lain.

Jumlah hitungan

`countNumber`

Jenis: `integer`

Wajib: ya

Tentukan jumlah hitungan. Nilai yang dapat diterima adalah integer positif (nilai 0 tidak diterima).

Jika `countType` yang digunakan adalah `imageCountMoreThan`, maka nilainya adalah jumlah maksimum citra yang ingin Anda pertahankan di repositori Anda. Jika `countType` yang digunakan adalah `sinceImagePushed`, maka nilainya adalah batas usia maksimum untuk citra Anda.

Tindakan

`type`

Tipe string

Wajib: ya

Tentukan jenis tindakan. Nilai yang didukung adalah `expire`.


Membuat pratinjau kebijakan siklus hidup

Pratinjau kebijakan siklus hidup memberikan cara untuk melihat dampak kebijakan siklus hidup pada repositori citra sebelum Anda menerapkannya. Hal ini dianggap sebagai praktik terbaik untuk melakukan pratinjau sebelum menerapkan kebijakan siklus hidup pada repositori. Prosedur berikut ini menunjukkan cara membuat pratinjau kebijakan siklus hidup.

Untuk membuat pratinjau kebijakan siklus hidup (AWS Management Console)

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah yang berisi repositori untuk melakukan pratinjau kebijakan siklus hidup.
3. Di panel navigasi, di bawah Registri pribadi, pilih Repositori.
4. Pada halaman Private repositories, pilih repositori dan yang menggunakan drop down Actions untuk memilih kebijakan Siklus Hidup.
5. Pada halaman aturan kebijakan siklus hidup untuk repositori, pilih Edit aturan pengujian, Buat aturan.
6. Tentukan detail berikut untuk setiap aturan kebijakan siklus hidup pengujian.

- a. Untuk Prioritas aturan, ketikkan nomor untuk prioritas aturan. Prioritas aturan menentukan urutan aturan kebijakan siklus hidup yang diterapkan.
- b. Untuk Deskripsi aturan, ketikkan deskripsi untuk aturan kebijakan siklus hidup.
- c. Untuk status Gambar, pilih Tagged (pencocokan wildcard), Tagged (pencocokan awalan), Untagged, atau Any.
- d. Jika Anda memilih Tagged (pencocokan wildcard) untuk status Gambar, lalu untuk Menentukan tag untuk pencocokan wildcard, Anda dapat menentukan daftar tag gambar dengan wildcard (*) untuk mengambil tindakan dengan kebijakan siklus hidup Anda. Misalnya, jika gambar Anda ditandai sebagaiprod,, prod1prod2, dan seterusnya, Anda akan menentukan prod* untuk mengambil tindakan pada semuanya. Jika Anda menentukan beberapa tanda, hanya citra dengan semua tanda yang ditentukan yang dipilih.

 Important

Ada batas maksimum empat wildcard (*) per string. Misalnya, ["*test*1*2*3", "test*1*2*3*"] valid tetapi ["test*1*2*3*4*5*6"] tidak valid.

- e. Jika Anda memilih Tagged (pencocokan awalan) untuk status Gambar, lalu untuk Menentukan tag untuk pencocokan awalan, Anda dapat menentukan daftar tag gambar yang akan diambil tindakan dengan kebijakan siklus hidup Anda.
 - f. Untuk Kriteria kecocokan, pilih salah satu Karena gambar didorong atau Jumlah gambar lebih dari lalu tentukan nilainya.
 - g. Pilih Simpan.
7. Buat aturan kebijakan siklus hidup pengujian tambahan dengan mengulangi langkah 5—7.
 8. Untuk menjalankan pratinjau kebijakan siklus hidup, pilih Simpan dan jalankan tes.
 9. Di bawah citra sesuai untuk aturan pengujian siklus hidup, tinjau dampak pratinjau kebijakan siklus hidup Anda.
 10. Jika Anda puas dengan hasil pratinjau, pilih Terapkan sebagai kebijakan siklus hidup untuk membuat kebijakan siklus hidup dengan aturan yang ditentukan. Harap diketahui bahwa setelah menerapkan kebijakan siklus hidup, citra yang terpengaruh akan kedaluwarsa dalam waktu 24 jam.
 11. Jika Anda tidak puas dengan hasil pratinjau, Anda dapat menghapus satu atau beberapa aturan siklus hidup pengujian dan membuat satu atau beberapa aturan untuk menggantinya, lalu ulangi pengujian.

Membuat kebijakan siklus hidup

Kebijakan siklus hidup menyediakan cara bagi Anda untuk membuat seperangkat aturan yang kedaluwarsa gambar repositori yang tidak digunakan. Prosedur berikut ini menunjukkan cara membuat kebijakan siklus hidup. Harap diketahui bahwa setelah membuat kebijakan siklus hidup, citra yang terpengaruh akan kedaluwarsa dalam waktu 24 jam.

Important

Merupakan praktik terbaik untuk membuat pratinjau kebijakan siklus hidup untuk memastikan bahwa gambar yang terpengaruh oleh aturan kebijakan siklus hidup Anda adalah yang Anda inginkan. Untuk informasi selengkapnya, lihat [Membuat pratinjau kebijakan siklus hidup](#).

Untuk membuat kebijakan siklus hidup (AWS Management Console)

Untuk membuat kebijakan siklus hidup menggunakan konsol

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah yang berisi repositori untuk membuat kebijakan siklus hidup.
3. Di panel navigasi, di bawah Registri pribadi, pilih Repositori.
4. Pada halaman Private repositories, pilih repositori dan yang menggunakan drop down Actions untuk memilih kebijakan Siklus Hidup.
5. Pada halaman aturan kebijakan siklus hidup untuk repositori, pilih Buat aturan.
6. Masukkan detail berikut untuk aturan kebijakan siklus hidup Anda.
 - a. Untuk Prioritas aturan, ketikkan nomor untuk prioritas aturan. Prioritas aturan menentukan urutan aturan kebijakan siklus hidup yang diterapkan.
 - b. Untuk Deskripsi aturan, ketikkan deskripsi untuk aturan kebijakan siklus hidup.
 - c. Untuk status Gambar, pilih Tagged (pencocokan wildcard), Tagged (pencocokan awalan), Untagged, atau Any.
 - d. Jika Anda memilih Tagged (pencocokan wildcard) untuk status Gambar, lalu untuk Menentukan tag untuk pencocokan wildcard, Anda dapat menentukan daftar tag gambar dengan wildcard (*) untuk mengambil tindakan dengan kebijakan siklus hidup Anda. Misalnya, jika gambar Anda ditandai sebagaiprod,, prod1prod2, dan seterusnya, Anda akan menentukan prod* untuk mengambil tindakan pada semuanya. Jika Anda menentukan beberapa tanda, hanya citra dengan semua tanda yang ditentukan yang dipilih.

⚠ Important

Ada batas maksimum empat wildcard (*) per string. Misalnya, ["*test*1*2*3", "test*1*2*3*"] valid tetapi ["test*1*2*3*4*5*6"] tidak valid.

- e. Jika Anda memilih Tagged (pencocokan awalan) untuk status Gambar, lalu untuk Menentukan tag untuk pencocokan awalan, Anda dapat menentukan daftar tag gambar yang akan diambil tindakan dengan kebijakan siklus hidup Anda.
 - f. Untuk Kriteria kecocokan, pilih salah satu Karena gambar didorong atau Jumlah gambar lebih dari lalu tentukan nilainya.
 - g. Pilih Simpan.
7. Buat aturan kebijakan siklus hidup tambahan dengan mengulangi langkah 5-7.

Untuk membuat kebijakan siklus hidup (AWS CLI)

Untuk membuat kebijakan siklus hidup menggunakan AWS CLI

1. Dapatkan nama repositori untuk membuat kebijakan siklus hidup.

```
aws ecr describe-repositories
```

2. Buat file lokal bernama `policy.json` dengan isi kebijakan siklus hidup. Untuk contoh kebijakan siklus hidup, lihat [Contoh kebijakan siklus hidup](#).
3. Buat kebijakan siklus hidup dengan menentukan nama repositori dan referensi file JSON kebijakan siklus hidup yang Anda buat.

```
aws ecr put-lifecycle-policy \  
  --repository-name repository-name \  
  --lifecycle-policy-text file://policy.json
```

Contoh kebijakan siklus hidup

Berikut ini adalah contoh kebijakan siklus hidup, yang menunjukkan sintaks.

Topik

- [Memfilter usia citra](#)

- [Memfilter jumlah citra](#)
- [Memfilter beberapa aturan](#)
- [Memfilter beberapa tanda dalam satu aturan](#)
- [Memfilter semua citra](#)

Memfilter usia citra

Contoh berikut menunjukkan sintaks kebijakan siklus hidup untuk kebijakan yang kedaluwarsa gambar dengan tag yang dimulai prod dengan menggunakan tagPatternList dari prod* yang juga lebih lama dari hari. 14

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Expire images older than 14 days",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["prod*"],
        "countType": "sinceImagePushed",
        "countUnit": "days",
        "countNumber": 14
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}
```

Memfilter jumlah citra

Contoh berikut menunjukkan sintaks kebijakan siklus hidup untuk kebijakan yang hanya menyimpan satu gambar yang tidak diberi tag dan kedaluwarsa semua gambar lainnya.

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Keep only one untagged image, expire all others",
```

```

        "selection": {
            "tagStatus": "untagged",
            "countType": "imageCountMoreThan",
            "countNumber": 1
        },
        "action": {
            "type": "expire"
        }
    }
]
}

```

Memfilter beberapa aturan

Contoh berikut menggunakan beberapa aturan dalam kebijakan siklus hidup. Contoh repositori dan kebijakan siklus hidup diberikan bersama dengan penjelasan dari hasilnya.

Contoh A

Isi repositori:

- citra A, Taglist: ["beta-1", "prod-1"], Didorong: 10 hari yang lalu
- citra B, Taglist: ["beta-2", "prod-2"], Didorong: 9 hari yang lalu
- citra C, Taglist: ["beta-3"], Didorong: 8 hari yang lalu

Teks kebijakan siklus hidup:

```

{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Rule 1",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["prod*"],
        "countType": "imageCountMoreThan",
        "countNumber": 1
      },
      "action": {
        "type": "expire"
      }
    },
  ],
}

```

```
{
  {
    "rulePriority": 2,
    "description": "Rule 2",
    "selection": {
      "tagStatus": "tagged",
      "tagPatternList": ["beta*"],
      "countType": "imageCountMoreThan",
      "countNumber": 1
    },
    "action": {
      "type": "expire"
    }
  }
}
```

Logika kebijakan siklus hidup ini adalah:

- Aturan 1 mengidentifikasi citra yang ditandai dengan prefiks prod. Ini harus menandai citra, dimulai dengan yang tertua, sampai ada satu atau lebih sedikit citra yang tersisa yang sesuai. Ini menandai citra A untuk kedaluwarsa.
- Aturan 2 mengidentifikasi citra yang ditandai dengan prefiks beta. Ini harus menandai citra, dimulai dengan yang tertua, sampai ada satu atau lebih sedikit citra yang tersisa yang sesuai. Ini menandai citra A dan citra B untuk kedaluwarsa. Namun, citra A sudah terlihat oleh Aturan 1 dan jika citra B kedaluwarsa, maka akan melanggar Aturan 1 dan dengan demikian dilewati.
- Hasil: citra A kedaluwarsa.

Contoh B

Ini adalah repositori yang sama seperti contoh sebelumnya tetapi urutan prioritas aturan diubah untuk mengcitrakan hasilnya.

Isi repositori:

- citra A, Taglist: ["beta-1", "prod-1"], Didorong: 10 hari yang lalu
- citra B, Taglist: ["beta-2", "prod-2"], Didorong: 9 hari yang lalu
- citra C, Taglist: ["beta-3"], Didorong: 8 hari yang lalu

Teks kebijakan siklus hidup:

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Rule 1",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["beta*"],
        "countType": "imageCountMoreThan",
        "countNumber": 1
      },
      "action": {
        "type": "expire"
      }
    },
    {
      "rulePriority": 2,
      "description": "Rule 2",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["prod*"],
        "countType": "imageCountMoreThan",
        "countNumber": 1
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}
```

Logika kebijakan siklus hidup ini adalah:

- Aturan 1 mengidentifikasi citra yang ditandai dengan prefiks beta. Ini harus menandai citra, dimulai dengan yang tertua, sampai ada satu atau lebih sedikit citra yang tersisa yang sesuai. Ini melihat ketiga citra dan akan menandai citra A dan citra B untuk kedaluwarsa.
- Aturan 2 mengidentifikasi citra yang ditandai dengan prefiks prod. Ini harus menandai citra, dimulai dengan yang tertua, sampai ada satu atau lebih sedikit citra yang tersisa yang sesuai. Ini tidak akan melihat citra karena semua citra yang tersedia sudah dilihat oleh Aturan 1 dan dengan demikian akan menandai tidak ada citra tambahan.
- Hasil: citra A dan B kedaluwarsa.

Memfilter beberapa tanda dalam satu aturan

Contoh berikut menentukan sintaks kebijakan siklus hidup untuk beberapa pola tag dalam satu aturan. Contoh repositori dan kebijakan siklus hidup diberikan bersama dengan penjelasan dari hasilnya.

Contoh A

Ketika beberapa pola tag ditentukan pada satu aturan, gambar harus cocok dengan semua pola tag yang terdaftar.

Isi repositori:

- citra A, Taglist: ["alpha-1"], Didorong: 12 hari yang lalu
- citra B, Taglist: ["beta-1"], Didorong: 11 hari yang lalu
- citra C, Taglist: ["alpha-2", "beta-2"], Didorong: 10 hari yang lalu
- citra D, Taglist: ["alpha-3"], Didorong: 4 hari yang lalu
- citra E, Taglist: ["beta-3"], Didorong: 3 hari yang lalu
- citra F, Taglist: ["alpha-4", "beta-4"], Didorong: 2 hari yang lalu

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Rule 1",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["alpha*", "beta*"],
        "countType": "sinceImagePushed",
        "countNumber": 5,
        "countUnit": "days"
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}
```

Logika kebijakan siklus hidup ini adalah:

- Aturan 1 mengidentifikasi gambar yang ditandai dengan awalan alpha dan. beta Ini melihat citra C dan F. Ini harus menandai citra yang lebih tua dari lima hari, yang akan menjadi citra C.
- Hasil: citra C kedaluwarsa.

Contoh B

Contoh berikut mengcitrakan bahwa tanda tidak bersifat eksklusif.

Isi repositori:

- citra A, Taglist: ["alpha-1", "beta-1", "gamma-1"], Didorong: 10 hari yang lalu
- citra B, Taglist: ["alpha-2", "beta-2"], Didorong: 9 hari yang lalu
- citra C, Taglist: ["alpha-3", "beta-3", "gamma-2"], Didorong: 8 hari yang lalu

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Rule 1",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["alpha*", "beta*"],
        "countType": "imageCountMoreThan",
        "countNumber": 1
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}
```

Logika kebijakan siklus hidup ini adalah:

- Aturan 1 mengidentifikasi gambar yang ditandai dengan awalan alpha dan. beta Ini melihat semua citra. Ini harus menandai citra, dimulai dengan yang tertua, sampai ada satu atau lebih sedikit citra yang tersisa yang sesuai. Ini menandai citra A dan B untuk kedaluwarsa.
- Hasil: citra A dan B kedaluwarsa.

Memfilter semua citra

Contoh kebijakan siklus hidup berikut ini menentukan semua citra dengan filter yang berbeda. Contoh repositori dan kebijakan siklus hidup diberikan bersama dengan penjelasan dari hasilnya.

Contoh A

Berikut ini adalah sintaks kebijakan siklus hidup untuk kebijakan yang berlaku untuk semua aturan tetapi hanya mempertahankan satu citra dan membuat semua lainnya kedaluwarsa.

Isi repositori:

- citra A, Taglist: ["alpha-1"], Didorong: 4 hari yang lalu
- citra B, Taglist: ["beta-1"], Didorong: 3 hari yang lalu
- citra C, Taglist: [], Didorong: 2 hari yang lalu
- citra D, Taglist: ["alpha-2"], Didorong: 1 hari yang lalu

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Rule 1",
      "selection": {
        "tagStatus": "any",
        "countType": "imageCountMoreThan",
        "countNumber": 1
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}
```

Logika kebijakan siklus hidup ini adalah:

- Aturan 1 mengidentifikasi semua citra. Ini melihat citra A, B, C, dan D. Ini membuat semua citra selain yang terbaru kedaluwarsa. Ini menandai citra A, B, dan C untuk kedaluwarsa.
- Hasil: citra A, B, dan C kedaluwarsa.

Contoh B

Contoh berikut ini mengcitakan kebijakan siklus hidup yang menggabungkan semua jenis aturan dalam satu kebijakan.

Isi repositori:

- Citra A, Taglist: ["alpha-1", "beta-1"], Didorong: 4 hari yang lalu
- Citra B, Taglist: [], Didorong: 3 hari yang lalu
- Citra C, Taglist: ["alpha-2"], Didorong: 2 hari yang lalu
- Citra D, Taglist: ["git hash"], Didorong: 1 hari yang lalu
- Citra E, Taglist: [], Didorong: 1 hari yang lalu

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Rule 1",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["alpha"],
        "countType": "imageCountMoreThan",
        "countNumber": 1
      },
      "action": {
        "type": "expire"
      }
    },
    {
      "rulePriority": 2,
      "description": "Rule 2",
      "selection": {
        "tagStatus": "untagged",
        "countType": "sinceImagePushed",
        "countUnit": "days",
        "countNumber": 1
      },
      "action": {
        "type": "expire"
      }
    }
  ],
}
```

```
{
  "rulePriority": 3,
  "description": "Rule 3",
  "selection": {
    "tagStatus": "any",
    "countType": "imageCountMoreThan",
    "countNumber": 1
  },
  "action": {
    "type": "expire"
  }
}
]
```

Logika kebijakan siklus hidup ini adalah:

- Aturan 1 mengidentifikasi citra yang ditandai dengan prefiks `alpha`. Ini mengidentifikasi citra A dan C. Ini mempertahankan citra terbaru dan menandai sisanya untuk kedaluwarsa. Ini menandai citra A untuk kedaluwarsa.
- Aturan 2 mengidentifikasi citra yang tidak ditandai. Ini mengidentifikasi citra B dan E. Ini menandai semua citra yang lebih tua dari satu hari untuk kedaluwarsa. Ini menandai citra B untuk kedaluwarsa.
- Aturan 3 mengidentifikasi semua citra. Ini mengidentifikasi citra A, B, C, D, dan E. Ini mempertahankan citra terbaru dan menandai sisanya untuk kedaluwarsa. Namun, ini tidak dapat menandai citra A, B, C, atau E karena citra tersebut diidentifikasi oleh aturan prioritas yang lebih tinggi. Ini menandai citra D untuk kedaluwarsa.
- Hasil: Citra A, B, dan D kedaluwarsa.

Tanda ketetapan citra

Anda dapat mengonfigurasi repositori untuk mengaktifkan kekekalan tag untuk mencegah tag gambar ditimpa. Setelah repositori dikonfigurasi untuk tanda tetap, sebuah pesan kesalahan `ImageTagAlreadyExistsException` akan kembali muncul jika Anda mencoba untuk mendorong citra dengan tanda yang sudah tersedia dalam repositori. Ketika kekekalan tag diaktifkan untuk repositori, ini memengaruhi semua tag dan Anda tidak dapat membuat beberapa tag tidak dapat diubah sementara yang lain tidak.

Anda dapat menggunakan AWS CLI atau AWS Management Console dan untuk mengatur mutabilitas tag gambar untuk repositori baru selama pembuatan atau untuk repositori yang ada kapan saja. Untuk langkah-langkah konsol, lihat [Membuat repositori pribadi](#) dan [Mengedit repositori pribadi](#).

Untuk membuat repositori dengan tanda tetap terkonfigurasi

Gunakan salah satu dari perintah berikut untuk membuat repositori citra baru dengan tanda tetap terkonfigurasi.

- [create-repository](#) (AWS CLI)

```
aws ecr create-repository --repository-name name --image-tag-mutability IMMUTABLE --region us-east-2
```

- [New-ECRRepository](#) (AWS Tools for Windows PowerShell)

```
New-ECRRepository -RepositoryName name -ImageTagMutability IMMUTABLE -Region us-east-2 -Force
```

Untuk memperbarui pengaturan ketetapan tanda citra untuk repositori yang tersedia

Gunakan salah satu dari perintah berikut untuk memperbarui pengaturan ketetapan tanda citra untuk repositori yang tersedia.

- [put-image-tag-mutability](#) (AWS CLI)

```
aws ecr put-image-tag-mutability --repository-name name --image-tag-mutability IMMUTABLE --region us-east-2
```

- [ImageTagMutabilityTulis-ECR](#) (AWS Tools for Windows PowerShell)

```
Write-ECRImageTagMutability -RepositoryName name -ImageTagMutability IMMUTABLE -Region us-east-2 -Force
```

Pemindaian citra

Fitur pemindaian dasar yang ditingkatkan ada dalam rilis pratinjau untuk Amazon ECR dan dapat berubah sewaktu-waktu. Selama pratinjau publik ini, Anda hanya dapat menggunakan AWS Management Console untuk ikut serta untuk versi pemindaian dasar yang ditingkatkan.

Pemindaian citra Amazon ECR membantu dalam mengidentifikasi kelemahan perangkat lunak dalam citra kontainer Anda. Jenis pemindaian berikut ditawarkan.

Important

Beralih antara pemindaian yang ditingkatkan, pemindaian dasar, dan versi pemindaian dasar yang ditingkatkan akan menyebabkan pemindaian yang telah dibuat sebelumnya tidak lagi tersedia. Anda harus mengatur pemindaian Anda lagi. Namun, jika Anda beralih kembali ke versi pemindaian sebelumnya, pemindaian yang sudah ada akan tersedia.

- Pemindaian yang disempurnakan —Amazon ECR terintegrasi dengan Amazon Inspector untuk menyediakan pemindaian otomatis dan berkelanjutan dari repositori Anda. Gambar kontainer Anda dipindai untuk kedua sistem operasi dan kerentanan paket bahasa pemrograman. Saat kerentanan baru muncul, hasil pemindaian diperbarui dan Amazon Inspector memancarkan acara EventBridge untuk memberi tahu Anda. Pemindaian yang disempurnakan memberikan yang berikut:
 - Kerentanan paket OS dan bahasa pemrograman.
 - Dua frekuensi pemindaian: Pindai pada push dan scan terus menerus.
- Pemindaian dasar —Amazon ECR menyediakan dua versi pemindaian dasar yang menggunakan database Common Vulnerabilities and Exposures (CVEs); versi GA saat ini yang menggunakan proyek Clair open-source dan versi pemindaian dasar yang baru ditingkatkan (dalam pratinjau) yang menggunakan teknologi asli kami. AWS Dengan pemindaian dasar, Anda mengonfigurasi repositori untuk memindai saat push atau Anda dapat melakukan pemindaian manual dan Amazon ECR menyediakan daftar temuan pemindaian. Pemindaian dasar menyediakan yang berikut:
 - Pemindaian OS.
 - Dua frekuensi pemindaian: Manual dan pemindaian saat push.

⚠ Important

Versi baru pemindaian dasar tidak mendukung `imageScanFindingsSummary` dan `imageScanStatus` di `DescribeImages` API. Untuk melihat ini, gunakan `DescribeImageScanFindings` API.

Menggunakan filter

Ketika pemindaian gambar dikonfigurasi untuk registri pribadi Anda, Anda dapat menentukan bahwa semua repositori dipindai atau Anda dapat menentukan filter untuk cakupan repositori mana yang dipindai.

Saat pemindaian dasar digunakan, Anda dapat menentukan pemindaian pada filter push untuk menentukan repositori mana yang diatur untuk melakukan pemindaian gambar saat gambar baru didorong. Setiap repositori yang tidak cocok dengan pemindaian pemindaian dasar pada filter push akan diatur ke frekuensi pemindaian manual yang berarti untuk melakukan pemindaian, Anda harus memicu pemindaian secara manual.

Saat pemindaian yang disempurnakan digunakan, Anda dapat menentukan filter terpisah untuk pemindaian saat push dan pemindaian berkelanjutan. Repositori apa pun yang tidak cocok dengan filter pemindaian yang disempurnakan akan menonaktifkan pemindaian. Jika Anda menggunakan pemindaian yang disempurnakan dan menentukan filter terpisah untuk pemindaian pada push dan pemindaian berkelanjutan di mana beberapa filter cocok dengan repositori yang sama, maka Amazon ECR memberlakukan filter pemindaian berkelanjutan melalui pemindaian pada filter push untuk repositori itu.

Ketika filter ditentukan, filter tanpa wildcard akan cocok dengan semua nama repositori yang berisi filter. Filter dengan wildcard (*) cocok dengan nama repositori mana pun di mana wildcard menggantikan nol atau lebih karakter dalam nama repositori. Tabel berikut memberikan contoh di mana nama repositori dinyatakan pada sumbu horizontal dan contoh filter ditentukan pada sumbu vertikal.

	prod	repo-prod	prod-repo	repo-prod-repo	prodrepo	
prod		Y 	Y 	Y 	Y 	Ya
*prod		Y 	Y 	T 	T 	Tidak
prod*		Y 	T 	Y 	T 	Ya
prod		Y 	Y 	Y 	Y 	Ya
prod*repo		T 	T 	Y 	T 	Ya

Topik

- [Pemindaian yang ditingkatkan](#)
- [Pemindaian dasar](#)
- [Memecahkan masalah pemindaian citra](#)

Pemindaian yang ditingkatkan

Pemindaian Amazon ECR yang disempurnakan adalah integrasi dengan Amazon Inspector yang menyediakan pemindaian kerentanan untuk gambar kontainer Anda. Gambar kontainer Anda dipindai untuk sistem operasi dan kerentanan paket bahasa pemrograman. Anda dapat melihat temuan pemindaian dengan Amazon ECR dan dengan Amazon Inspector secara langsung. Untuk

informasi selengkapnya tentang Amazon Inspector, lihat [Memindai gambar kontainer dengan Amazon Inspector](#) di Panduan Pengguna Amazon Inspector.

Dengan pemindaian yang disempurnakan, Anda dapat memilih repositori mana yang dikonfigurasi untuk pemindaian otomatis dan berkelanjutan dan mana yang dikonfigurasi untuk pemindaian saat push. Ini dilakukan dengan mengatur filter pemindaian.

Pertimbangan untuk pemindaian yang ditingkatkan

Berikut ini harus dipertimbangkan saat mengaktifkan pemindaian Amazon ECR yang ditingkatkan.

- Tidak ada biaya tambahan dari Amazon ECR untuk menggunakan fitur ini, namun ada biaya dari Amazon Inspector untuk memindai gambar Anda. Untuk informasi selengkapnya, lihat [harga Amazon Inspector](#).
- Pemindaian yang disempurnakan tidak didukung di Wilayah berikut:
 - Timur Tengah (UEA) (me-central-1)
 - Asia Pasifik (Hyderabad) () ap-south-2
 - Israel (Tel Aviv) (il-central-1)
 - Asia Pasifik (Melbourne) (ap-southeast-4)
 - Eropa (Spanyol) (eu-south-2)
- Amazon Inspector mendukung pemindaian untuk sistem operasi tertentu. Untuk daftar selengkapnya, lihat [Sistem operasi yang didukung - Pemindaian Amazon ECR](#) di Panduan Pengguna Amazon Inspector.
- Amazon Inspector menggunakan peran IAM terkait layanan, yang menyediakan izin yang diperlukan untuk menyediakan pemindaian yang disempurnakan untuk repositori Anda. Peran IAM terkait layanan dibuat secara otomatis oleh Amazon Inspector saat pemindaian yang disempurnakan diaktifkan untuk registri pribadi Anda. Untuk informasi selengkapnya, lihat [Menggunakan peran terkait layanan untuk Amazon Inspector](#) di Panduan Pengguna Amazon Inspector.
- Saat Anda awalnya mengaktifkan pemindaian yang disempurnakan untuk registri pribadi Anda, Amazon Inspector hanya mengenali gambar yang didorong ke Amazon ECR dalam 30 hari terakhir, berdasarkan stempel waktu push gambar, atau ditarik dalam 90 hari terakhir. Gambar yang lebih tua akan memiliki status SCAN_ELIGIBILITY_EXPIRED pemindaian. Jika Anda ingin gambar-gambar ini dipindai oleh Amazon Inspector, Anda harus mendorongnya lagi ke repositori Anda.

- Semua gambar yang didorong ke Amazon ECR setelah pemindaian yang ditingkatkan diaktifkan terus dipindai selama durasi yang dikonfigurasi. Secara default, durasinya adalah Seumur Hidup. Pengaturan ini dapat dikonfigurasi menggunakan konsol Amazon Inspector. Untuk informasi selengkapnya, lihat [Mengubah durasi pemindaian yang ditingkatkan](#).
- Saat pemindaian yang disempurnakan diaktifkan untuk registri pribadi Amazon ECR Anda, repositori yang cocok dengan filter pemindaian dipindai hanya menggunakan pemindaian yang disempurnakan. Repositori apa pun yang tidak cocok dengan filter akan memiliki frekuensi Off pemindaian dan tidak akan dipindai. Pemindaian manual menggunakan pemindaian yang disempurnakan tidak didukung. Untuk informasi selengkapnya, lihat [Menggunakan filter](#).
- Jika Anda menentukan filter terpisah untuk pemindaian pada push dan pemindaian berkelanjutan di mana beberapa filter cocok dengan repositori yang sama, maka Amazon ECR memberlakukan filter pemindaian berkelanjutan melalui pemindaian pada filter push untuk repositori tersebut.
- Saat pemindaian yang ditingkatkan diaktifkan, Amazon ECR mengirimkan peristiwa ke EventBridge saat frekuensi pemindaian untuk repositori diubah. Amazon Inspector memancarkan peristiwa EventBridge saat pemindaian awal selesai dan saat temuan pemindaian gambar dibuat, diperbarui, atau ditutup.

Izin IAM yang diperlukan

Pemindaian Amazon ECR yang disempurnakan memerlukan peran IAM terkait layanan Amazon Inspector dan bahwa prinsipal IAM yang mengaktifkan dan menggunakan pemindaian yang disempurnakan memiliki izin untuk memanggil Amazon Inspector API yang diperlukan untuk pemindaian. Peran IAM terkait layanan Amazon Inspector dibuat secara otomatis oleh Amazon Inspector ketika pemindaian yang disempurnakan diaktifkan untuk registri pribadi Anda. Untuk informasi selengkapnya, lihat [Menggunakan peran terkait layanan untuk Amazon Inspector](#) di Panduan Pengguna Amazon Inspector.

Kebijakan IAM berikut memberikan izin yang diperlukan untuk mengaktifkan dan menggunakan pemindaian yang disempurnakan. Ini termasuk izin yang diperlukan untuk Amazon Inspector untuk membuat peran IAM terkait layanan serta izin Amazon Inspector API yang diperlukan untuk mengaktifkan dan menonaktifkan pemindaian yang disempurnakan dan mengambil temuan pemindaian.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Effect": "Allow",
    "Action": [
        "inspector2:Enable",
        "inspector2:Disable",
        "inspector2:ListFindings",
        "inspector2:ListAccountPermissions",
        "inspector2:ListCoverage"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "iam:AWSServiceName": [
          "inspector2.amazonaws.com"
        ]
      }
    }
  }
]
}

```

Menggunakan pemindaian yang disempurnakan

AWS Management Console

Untuk mengaktifkan pemindaian yang disempurnakan untuk registri pribadi Anda (AWS Management Console)

Konfigurasi pemindaian didefinisikan pada tingkat registri pribadi berdasarkan Per-wilayah.

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah untuk mengatur konfigurasi pemindaian.
3. Di panel navigasi, pilih Registri pribadi, Pemindaian.
4. Pada halaman konfigurasi Scanning, untuk jenis Scan pilih Enhanced scanning.
5. (Opsional) Secara default, saat pemindaian yang ditingkatkan dipilih, semua repositori Anda diatur untuk Pemindaian berkelanjutan. Anda dapat mengubah konfigurasi pemindaian default dengan membatalkan pilihan kotak Continuous scan all repositories. Anda kemudian

dapat mengonfigurasi semua repositori untuk pemindaian saat push atau Anda dapat menentukan filter pemindaian terpisah untuk terus menerus dan memindai saat push. Saat filter pemindaian disetel, Anda dapat memilih kecocokan repositori pratinjau untuk memverifikasi repositori mana di registri Anda yang cocok dengan filter yang ditentukan.

Important

Filter tanpa wildcard akan cocok dengan semua nama repositori yang berisi filter. Filter dengan wildcard (*) cocok dengan nama repositori di mana wildcard menggantikan nol atau lebih karakter dalam nama repositori.

6. Pilih Simpan.
7. Ulangi langkah-langkah ini di setiap Wilayah di mana Anda ingin mengaktifkan pemindaian yang disempurnakan.

AWS CLI

Gunakan AWS CLI perintah berikut untuk mengaktifkan pemindaian yang disempurnakan untuk registri pribadi Anda menggunakan AWS CLI. Anda dapat menentukan filter pemindaian menggunakan `rules` objek.

- [put-registry-scanning-configuration](#) (AWS CLI)

Contoh berikut mengaktifkan pemindaian yang disempurnakan untuk registri pribadi Anda. Secara default, ketika tidak `rules` ditentukan, Amazon ECR menyetel konfigurasi pemindaian ke pemindaian berkelanjutan untuk semua repositori.

```
aws ecr put-registry-scanning-configuration \
  --scan-type ENHANCED \
  --region us-east-2
```

Contoh berikut mengaktifkan pemindaian yang disempurnakan untuk registri pribadi Anda dan menentukan filter pemindaian. Filter pemindaian dalam contoh mengaktifkan pemindaian terus menerus untuk semua repositori dengan `prod` namanya.

```
aws ecr put-registry-scanning-configuration \
  --scan-type ENHANCED \
  --rules '[{"repositoryFilters" : [{"filter": "prod", "filterType" :  
"WILDCARD"}], "scanFrequency" : "CONTINUOUS_SCAN"}]' \
```

```
--region us-east-2
```

Contoh berikut mengaktifkan pemindaian yang disempurnakan untuk registri pribadi Anda dan menentukan beberapa filter pemindaian. Filter pemindaian dalam contoh mengaktifkan pemindaian berkelanjutan untuk semua repositori dengan prod namanya dan mengaktifkan pemindaian pada push hanya untuk semua repositori lainnya.

```
aws ecr put-registry-scanning-configuration \  
  --scan-type ENHANCED \  
  --rules '[{"repositoryFilters" : [{"filter": "prod", "filterType" :  
"WILDCARD"}], "scanFrequency" : "CONTINUOUS_SCAN"}, {"repositoryFilters" :  
[{"filter": "*", "filterType" : "WILDCARD"}], "scanFrequency" : "SCAN_ON_PUSH"}]' \  
  --region us-west-2
```

Mengubah durasi pemindaian yang ditingkatkan

Amazon Inspector mendukung konfigurasi durasi yang terus dipantau oleh repositori pribadi Anda. Secara default, ketika pemindaian yang disempurnakan diaktifkan untuk registri pribadi Amazon ECR Anda, layanan Amazon Inspector terus memantau repositori Anda hingga gambar dihapus atau pemindaian yang ditingkatkan dinonaktifkan. Durasi Amazon Inspector memindai gambar Anda dapat diubah menggunakan pengaturan Amazon Inspector. Durasi pemindaian yang tersedia adalah Seumur Hidup (default), 180 hari, dan 30 hari. Ketika durasi pemindaian untuk repositori berlalu, status pemindaian ditampilkan saat mencantumkan kerentanan `SCAN_ELIGIBILITY_EXPIRED` pemindaian Anda. Untuk informasi selengkapnya, lihat [Mengubah durasi pemindaian ulang otomatis Amazon ECR](#) di Panduan Pengguna Amazon Inspector.

Untuk mengubah pengaturan durasi pemindaian yang disempurnakan

1. [Buka konsol Amazon Inspector di https://console.aws.amazon.com/inspector/v2/home](https://console.aws.amazon.com/inspector/v2/home).
2. Di navigasi kiri, perluas Pengaturan dan kemudian pilih Umum.
3. Pada halaman Pengaturan, di bawah durasi pemindaian ulang ECR pilih pengaturan, lalu pilih Simpan.

EventBridge acara

Saat pemindaian yang ditingkatkan diaktifkan, Amazon ECR mengirimkan peristiwa ke EventBridge saat frekuensi pemindaian untuk repositori diubah. Amazon Inspector memancarkan peristiwa

EventBridge saat pemindaian awal selesai dan saat temuan pemindaian gambar dibuat, diperbarui, atau ditutup.

Acara untuk perubahan frekuensi pemindaian repositori

Ketika pemindaian yang disempurnakan diaktifkan untuk registri Anda, peristiwa berikut dikirim oleh Amazon ECR ketika ada perubahan dengan sumber daya yang telah ditingkatkan pemindaian diaktifkan. Ini termasuk repositori baru yang sedang dibuat, frekuensi pemindaian untuk repositori yang diubah, atau ketika gambar dibuat atau dihapus di repositori dengan pemindaian yang ditingkatkan diaktifkan. Untuk informasi selengkapnya, lihat [Pemindaian citra](#).

```
{
  "version": "0",
  "id": "0c18352a-a4d4-6853-ef53-0abEXAMPLE",
  "detail-type": "ECR Scan Resource Change",
  "source": "aws.ecr",
  "account": "123456789012",
  "time": "2021-10-14T20:53:46Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "action-type": "SCAN_FREQUENCY_CHANGE",
    "repositories": [{
      "repository-name": "repository-1",
      "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-1",
      "scan-frequency": "SCAN_ON_PUSH",
      "previous-scan-frequency": "MANUAL"
    },
    {
      "repository-name": "repository-2",
      "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-2",
      "scan-frequency": "CONTINUOUS_SCAN",
      "previous-scan-frequency": "SCAN_ON_PUSH"
    },
    {
      "repository-name": "repository-3",
      "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-3",
      "scan-frequency": "CONTINUOUS_SCAN",
      "previous-scan-frequency": "SCAN_ON_PUSH"
    }
  ],
  "resource-type": "REPOSITORY",
  "scan-type": "ENHANCED"
}
```

```
}  
}
```

Acara untuk pemindaian gambar awal (pemindaian yang disempurnakan)

Saat pemindaian yang disempurnakan diaktifkan untuk registri Anda, peristiwa berikut akan dikirim oleh Amazon Inspector saat pemindaian gambar awal selesai. Parameter `finding-severity-counts` hanya akan mengembalikan nilai untuk suatu tingkat keparahan jika ada. Contohnya, jika citra tidak mengandung temuan di tingkat CRITICAL, maka tidak ada hitungan kritis yang dikembalikan. Untuk informasi selengkapnya, lihat [Pemindaian yang ditingkatkan](#).

Pola acara:

```
{  
  "source": ["aws.inspector2"],  
  "detail-type": ["Inspector2 Scan"]  
}
```

Contoh output:

```
{  
  "version": "0",  
  "id": "739c0d3c-4f02-85c7-5a88-94a9EXAMPLE",  
  "detail-type": "Inspector2 Scan",  
  "source": "aws.inspector2",  
  "account": "123456789012",  
  "time": "2021-12-03T18:03:16Z",  
  "region": "us-east-2",  
  "resources": [  
    "arn:aws:ecr:us-east-2:123456789012:repository/amazon/amazon-ecs-sample"  
  ],  
  "detail": {  
    "scan-status": "INITIAL_SCAN_COMPLETE",  
    "repository-name": "arn:aws:ecr:us-east-2:123456789012:repository/amazon/  
amazon-ecs-sample",  
    "finding-severity-counts": {  
      "CRITICAL": 7,  
      "HIGH": 61,  
      "MEDIUM": 62,  
      "TOTAL": 158  
    }  
  },  
}
```

```

    "image-digest":
      "sha256:36c7b282abd0186e01419f2e58743e1bf635808231049bbc9d77e5EXAMPLE",
    "image-tags": [
      "latest"
    ]
  }
}

```

Acara untuk pembaruan pencarian pemindaian gambar (pemindaian yang disempurnakan)

Saat pemindaian yang disempurnakan diaktifkan untuk registri Anda, peristiwa berikut akan dikirim oleh Amazon Inspector saat temuan pemindaian gambar dibuat, diperbarui, atau ditutup. Untuk informasi selengkapnya, lihat [Pemindaian yang ditingkatkan](#).

Pola acara:

```

{
  "source": ["aws.inspector2"],
  "detail-type": ["Inspector2 Finding"]
}

```

Contoh output:

```

{
  "version": "0",
  "id": "42dbea55-45ad-b2b4-87a8-afaEXAMPLE",
  "detail-type": "Inspector2 Finding",
  "source": "aws.inspector2",
  "account": "123456789012",
  "time": "2021-12-03T18:02:30Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ecr:us-east-2:123456789012:repository/amazon/amazon-ecs-sample/
sha256:36c7b282abd0186e01419f2e58743e1bf635808231049bbc9d77eEXAMPLE"
  ],
  "detail": {
    "awsAccountId": "123456789012",
    "description": "In libssh2 v1.9.0 and earlier versions, the SSH_MSG_DISCONNECT
logic in packet.c has an integer overflow in a bounds check, enabling an attacker to
specify an arbitrary (out-of-bounds) offset for a subsequent memory read. A crafted
SSH server may be able to disclose sensitive information or cause a denial of service
condition on the client system when a user connects to the server.",

```



```
"findingArn": "arn:aws:inspector2:us-east-2:123456789012:finding/
be674aadd0f75ac632055EXAMPLE",
"firstObservedAt": "Dec 3, 2021, 6:02:30 PM",
"inspectorScore": 6.5,
"inspectorScoreDetails": {
  "adjustedCvss": {
    "adjustments": [],
    "cvssSource": "REDHAT_CVE",
    "score": 6.5,
    "scoreSource": "REDHAT_CVE",
    "scoringVector": "CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N",
    "version": "3.0"
  }
},
"lastObservedAt": "Dec 3, 2021, 6:02:30 PM",
"packageVulnerabilityDetails": {
  "cvss": [
    {
      "baseScore": 6.5,
      "scoringVector": "CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N",
      "source": "REDHAT_CVE",
      "version": "3.0"
    },
    {
      "baseScore": 5.8,
      "scoringVector": "AV:N/AC:M/Au:N/C:P/I:N/A:P",
      "source": "NVD",
      "version": "2.0"
    },
    {
      "baseScore": 8.1,
      "scoringVector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:H",
      "source": "NVD",
      "version": "3.1"
    }
  ],
  "referenceUrls": [
    "https://access.redhat.com/errata/RHSA-2020:3915"
  ],
  "source": "REDHAT_CVE",
  "sourceUrl": "https://access.redhat.com/security/cve/CVE-2019-17498",
  "vendorCreatedAt": "Oct 16, 2019, 12:00:00 AM",
  "vendorSeverity": "Moderate",
  "vulnerabilityId": "CVE-2019-17498",
```

```

    "vulnerablePackages": [
      {
        "arch": "X86_64",
        "epoch": 0,
        "name": "libssh2",
        "packageManager": "OS",
        "release": "12.amzn2.2",
        "sourceLayerHash":
"sha256:72d97abdfae3b3c933ff41e39779cc72853d7bd9dc1e4800c5294dEXAMPLE",
        "version": "1.4.3"
      }
    ],
    "remediation": {
      "recommendation": {
        "text": "Update all packages in the vulnerable packages section to
their latest versions."
      }
    },
    "resources": [
      {
        "details": {
          "awsEcrContainerImage": {
            "architecture": "amd64",
            "imageHash":
"sha256:36c7b282abd0186e01419f2e58743e1bf635808231049bbc9d77e5EXAMPLE",
            "imageTags": [
              "latest"
            ],
            "platform": "AMAZON_LINUX_2",
            "pushedAt": "Dec 3, 2021, 6:02:13 PM",
            "registry": "123456789012",
            "repositoryName": "amazon/amazon-ecs-sample"
          }
        },
        "id": "arn:aws:ecr:us-east-2:123456789012:repository/amazon/amazon-ecs-
sample/sha256:36c7b282abd0186e01419f2e58743e1bf635808231049bbc9d77EXAMPLE",
        "partition": "N/A",
        "region": "N/A",
        "type": "AWS_ECR_CONTAINER_IMAGE"
      }
    ],
    "severity": "MEDIUM",
    "status": "ACTIVE",

```

```
    "title": "CVE-2019-17498 - libssh2",
    "type": "PACKAGE_VULNERABILITY",
    "updatedAt": "Dec 3, 2021, 6:02:30 PM"
  }
}
```

Mengambil temuan pemindaian citra

Anda dapat mengambil temuan pemindaian untuk pemindaian citra yang telah diselesaikan terakhir. Daftar temuan berdasarkan tingkat kepelikan kelemahan perangkat lunak yang ditemukan, berdasarkan basis data Common Vulnerabilities and Exposures (CVE).

Untuk detail pemecahan masalah untuk beberapa masalah umum saat memindai citra, lihat [Memecahkan masalah pemindaian citra](#).

Untuk mengambil temuan pemindaian citra (AWS Management Console)

AWS Management Console

Lakukan langkah-langkah berikut untuk mengambil temuan pemindaian citra menggunakan AWS Management Console.

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah tempat repositori Anda ada.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih repositori yang berisi citra yang akan diambil hasil pemindaian.
5. Pada halaman Gambar, di bawah kolom Kerentanan, pilih Lihat temuan untuk gambar untuk mengambil temuan pemindaian.
6. Saat melihat Temuan, nama kerentanan di kolom Nama adalah tautan ke konsol Amazon Inspector tempat Anda dapat melihat detail selengkapnya.

AWS CLI

Gunakan AWS CLI perintah berikut untuk mengambil temuan pemindaian gambar menggunakan AWS CLI. Anda dapat menentukan citra menggunakan `imageTag` atau `imageDigest`, yang keduanya dapat diperoleh dengan menggunakan perintah CLI [list-images](#).

- [describe-image-scan-findings](#) (AWS CLI)

Contoh berikut menggunakan tanda citra.

```
aws ecr describe-image-scan-findings \  
  --repository-name name \  
  --image-id imageTag=tag_name \  
  --region us-east-2
```

Contoh berikut menggunakan digest citra.

```
aws ecr describe-image-scan-findings \  
  --repository-name name \  
  --image-id imageDigest=sha256_hash \  
  --region us-east-2
```


Pemindaian dasar

Fitur pemindaian dasar yang ditingkatkan ada dalam rilis pratinjau untuk Amazon ECR dan dapat berubah sewaktu-waktu. Selama pratinjau publik ini, Anda hanya dapat menggunakan AWS Management Console untuk ikut serta untuk versi pemindaian dasar yang ditingkatkan.

Amazon ECR menyediakan dua versi pemindaian dasar yang menggunakan database Common Vulnerabilities and Exposures (CVEs); versi GA saat ini yang menggunakan proyek Clair open-source dan versi pemindaian dasar yang baru ditingkatkan (dalam pratinjau) yang menggunakan teknologi asli kami. AWS Dengan salah satu versi pemindaian dasar Amazon ECR diaktifkan pada registri pribadi Anda, Anda dapat mengonfigurasi filter repositori untuk menentukan repositori mana yang diatur untuk dipindai saat push atau Anda dapat melakukan pemindaian manual. Amazon ECR menyediakan daftar temuan pemindaian. Setiap citra kontainer dapat dipindai sekali per 24 jam. Anda dapat meninjau temuan pemindaian untuk informasi tentang keamanan gambar kontainer yang sedang digunakan menggunakan DescribeImageScanFindings API atau di dalam konsol. Untuk informasi lebih lanjut tentang Clair, lihat [Clair](#) di GitHub

Amazon ECR menggunakan tingkat kepelikan untuk CVE dari sumber distribusi hulu jika tersedia, jika tidak, kita menggunakan skor Common Vulnerability Scoring System (CVSS). Skor CVSS dapat digunakan untuk mengetahui tingkat kepelikan kelemahan NVD. Untuk informasi selengkapnya, lihat [Tingkat kepelikan kelemahan NVD](#).

Setiap repositori yang tidak cocok dengan pemindaian pada filter push akan diatur ke frekuensi pemindaian manual yang berarti untuk melakukan pemindaian, Anda harus secara manual memicu pemindaian. Temuan pemindaian citra yang telah diselesaikan terakhir dapat diambil untuk setiap citra. Amazon ECR mengirimkan acara ke Amazon EventBridge (sebelumnya disebut CloudWatch Peristiwa) saat pemindaian gambar selesai. Untuk informasi selengkapnya, lihat [Acara Amazon ECR dan EventBridge](#).

 Important

Versi baru pemindaian dasar didukung di wilayah berikut:

- Asia Pacific (Hong Kong) (ap-east-1)
- Europe (Stockholm) (eu-north-1)
- Middle East (Bahrain) (me-south-1)
- Asia Pacific (Mumbai) (ap-south-1)
- Europe (Paris) (eu-west-3)
- AWS GovCloud (AS-Timur) (us-gov-east-1)
- Africa (Cape Town) (af-south-1)
- Asia Pasifik (Jakarta) (ap-southeast-3)
- Europe (Frankfurt) (eu-central-1)
- Europe (Ireland) (eu-west-1)
- South America (São Paulo) (sa-east-1)
- US East (Ohio) (us-east-2)
- AWS GovCloud (AS-Barat) (us-gov-west-1)
- Asia Pacific (Tokyo) (ap-northeast-1)
- Asia Pacific (Seoul) (ap-northeast-2)
- Asia Pacific (Osaka) (ap-northeast-3)
- Europe (Milan) (eu-south-1)
- Europe (London) (eu-west-2)
- US East (N. Virginia) (us-east-1)
- Asia Pacific (Singapore) (ap-southeast-1)

- Asia Pacific (Sydney) (ap-southeast-2)
- Canada (Central) (ca-central-1)
- US West (N. California) (us-west-1)
- US West (Oregon) (us-west-2)
- Eropa (Zurich) () eu-central-2

Untuk detail pemecahan masalah untuk beberapa masalah umum saat memindai citra, lihat [Memecahkan masalah pemindaian citra](#).

Sebagai praktik keamanan terbaik dan untuk cakupan lanjutan, kami menyarankan Anda untuk terus menggunakan versi sistem operasi yang didukung. Sesuai dengan kebijakan vendor, sistem operasi yang dihentikan tidak lagi diperbarui dengan tambalan dan, dalam banyak kasus, nasihat keamanan baru tidak lagi dirilis untuk mereka. Selain itu, beberapa vendor menghapus penasihat dan deteksi keamanan yang ada dari feed mereka ketika sistem operasi yang terpengaruh mencapai akhir dukungan standar. Setelah distribusi kehilangan dukungan dari vendor, Amazon ECR mungkin tidak lagi mendukung pemindaian kelemahan. Setiap temuan yang dihasilkan Amazon ECR untuk sistem operasi yang dihentikan harus digunakan hanya untuk tujuan informasi. Di bawah ini adalah sistem operasi dan versi yang didukung saat ini.

Sistem Operasi	Versi
Alpine Linux (Alpine)	3.19
Alpine Linux (Alpine)	3.18
Alpine Linux (Alpine)	3.17
Alpine Linux (Alpine)	3.16
Amazon Linux 2 (AL2)	AL2
Amazon Linux 2023 (AL2023)	AL2023
CentOS Linux (CentOS)	7
Server Debian (Kutu Buku)	12
Server Debian (Bullseye)	11

Sistem Operasi	Versi
Server Debian (Buster)	10
Oracle Linux (Oracle)	9
Oracle Linux (Oracle)	8
Oracle Linux (Oracle)	7
Ubuntu (Bulan)	23.04
Ubuntu (Jammy)	22.04 (LTS)
Ubuntu (Fokus)	20.024 (LTS)
Ubuntu (Bionik)	18.04 (ESM)
Ubuntu (Xenial)	16.04 (ESM)
Ubuntu (Terpercaya)	14.04 (ESM)
Red Hat Enterprise Linux (RHEL)	7
Red Hat Enterprise Linux (RHEL)	8
Red Hat Enterprise Linux (RHEL)	9

Menggunakan pemindaian dasar

Basic scanning with Clair

Secara default, Amazon ECR memungkinkan pemindaian dasar pada semua pendaftar pribadi. Akibatnya, kecuali Anda telah mengubah pengaturan pemindaian pada registri pribadi Anda, seharusnya tidak perlu mengaktifkan pemindaian dasar. Anda dapat menggunakan langkah-langkah berikut untuk memverifikasi bahwa pemindaian dasar diaktifkan dan menentukan satu atau lebih pemindaian pada filter push.

Untuk mengaktifkan pemindaian dasar untuk registri pribadi Anda (AWS Management Console)

Konfigurasi pemindaian didefinisikan pada tingkat registri pribadi berdasarkan Per-wilayah.

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah untuk mengatur konfigurasi pemindaian.
3. Di panel navigasi, pilih Registri pribadi, Pemindaian.
4. Pada halaman konfigurasi Pemindaian, Untuk jenis Pindai pilih Pemindaian dasar.
5. Secara default semua repositori Anda diatur untuk pemindaian Manual. Anda dapat secara opsional mengonfigurasi pemindaian saat push dengan menentukan Pindai pada filter push. Anda dapat mengatur pemindaian pada push untuk semua repositori atau repositori individual. Untuk informasi selengkapnya, lihat [Menggunakan filter](#).

Improved basic scanning with AWS native technology (In preview)

Versi baru pemindaian dasar Amazon ECR sekarang tersedia dalam pratinjau.

Untuk mengaktifkan pemindaian dasar yang ditingkatkan untuk registri pribadi Anda (AWS Management Console)

Konfigurasi pemindaian didefinisikan pada tingkat registri pribadi berdasarkan Per-wilayah.

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah untuk mengatur konfigurasi pemindaian.
3. Di panel navigasi, pilih Registri pribadi, Pemindaian.
4. Pada halaman konfigurasi Pemindaian, Untuk jenis Pindai pilih Peningkatan pemindaian dasar (Dalam pratinjau) - baru.
5. Secara default semua repositori Anda diatur untuk pemindaian Manual. Anda dapat secara opsional mengonfigurasi pemindaian saat push dengan menentukan Pindai pada filter push. Anda dapat mengatur pemindaian pada push untuk semua repositori atau repositori individual. Untuk informasi selengkapnya, lihat [Menggunakan filter](#).

Memindai citra secara manual

Anda dapat memulai pemindaian citra secara manual ketika Anda ingin memindai citra di repositori yang tidak dikonfigurasi untuk pindai saat mendorong. Sebuah citra hanya dapat dipindai sekali setiap hari. Batas ini mencakup pemindaian awal pada push, jika dikonfigurasi, dan pemindaian manual apa pun.

Untuk detail pemecahan masalah untuk beberapa masalah umum saat memindai citra, lihat [Memecahkan masalah pemindaian citra](#).

AWS Management Console

Gunakan langkah-langkah berikut untuk memulai pemindaian citra manual menggunakan AWS Management Console.

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah untuk membuat repositori Anda.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih repositori yang berisi citra yang akan dipindai.
5. Pada halaman citra, pilih citra yang akan dipindai dan kemudian pilih Pindai.

AWS CLI

- [start-image-scan](#) (AWS CLI)

Contoh berikut menggunakan tanda citra.

```
aws ecr start-image-scan --repository-name name --image-id imageTag=tag_name --region us-east-2
```

Contoh berikut menggunakan digest citra.

```
aws ecr start-image-scan --repository-name name --image-id imageDigest=sha256_hash --region us-east-2
```

AWS Tools for Windows PowerShell

- [ImageScanFindingDapatkan-ECR](#) ()AWS Tools for Windows PowerShell

Contoh berikut menggunakan tanda citra.

```
Start-ECRImageScan -RepositoryName name -ImageId_ImageTag tag_name -Region us-east-2 -Force
```

Contoh berikut menggunakan digest citra.

```
Start-ECRImageScan -RepositoryName name -ImageId_ImageDigest sha256_hash -  
Region us-east-2 -Force
```

Mengambil temuan pemindaian citra

Anda dapat mengambil temuan pemindaian untuk pemindaian citra yang telah diselesaikan terakhir. Daftar temuan berdasarkan tingkat kepelikan kelemahan perangkat lunak yang ditemukan, berdasarkan basis data Common Vulnerabilities and Exposures (CVE).

Untuk detail pemecahan masalah untuk beberapa masalah umum saat memindai citra, lihat [Memecahkan masalah pemindaian citra](#).

AWS Management Console

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/repositories>.
2. Dari bilah navigasi, pilih Wilayah untuk membuat repositori Anda.
3. Di panel navigasi, pilih Repositori.
4. Pada halaman Repositori, pilih repositori yang berisi citra yang akan diambil hasil pemindaian.
5. Pada halaman Citra, di bawah kolom kelemahan, pilih detail untuk citra yang akan diambil hasil pemindaian.

AWS CLI

Gunakan AWS CLI perintah berikut untuk mengambil temuan pemindaian gambar menggunakan AWS CLI Anda dapat menentukan citra menggunakan `imageTag` atau `imageDigest`, yang keduanya dapat diperoleh dengan menggunakan perintah CLI [list-images](#).

- [describe-image-scan-findings](#) (AWS CLI)

Contoh berikut menggunakan tanda citra.

```
aws ecr describe-image-scan-findings --repository-name name --image-id  
imageTag=tag_name --region us-east-2
```

Contoh berikut menggunakan digest citra.

```
aws ecr describe-image-scan-findings --repository-name name --image-id  
imageDigest=sha256_hash --region us-east-2
```

AWS Tools for Windows PowerShell

- [ImageScanFindingDapatkan-ECR](#) ()AWS Tools for Windows PowerShell

Contoh berikut menggunakan tanda citra.

```
Get-ECRImageScanFinding -RepositoryName name -ImageId_ImageTag tag_name -  
Region us-east-2
```

Contoh berikut menggunakan digest citra.

```
Get-ECRImageScanFinding -RepositoryName name -ImageId_ImageDigest sha256_hash -  
Region us-east-2
```

Memecahkan masalah pemindaian citra

Berikut ini adalah kegagalan pemindaian citra yang umum. Anda dapat melihat kesalahan seperti ini di konsol Amazon ECR dengan menampilkan detail gambar atau melalui API atau AWS CLI dengan menggunakan `DescribeImageScanFindings`

UnsupportedImageError

Anda mungkin mendapatkan `UnsupportedImageError` kesalahan saat mencoba melakukan pemindaian dasar pada gambar yang dibuat menggunakan sistem operasi yang Amazon ECR tidak mendukung pemindaian gambar dasar. Amazon ECR mendukung pemindaian kelemahan paket untuk sebagian besar versi distribusi Amazon Linux, Amazon Linux 2, Debian, Ubuntu, CentOS, Oracle Linux, Alpine, dan RHEL Linux. Setelah distribusi kehilangan dukungan dari vendor, Amazon ECR mungkin tidak lagi mendukung pemindaian kelemahan. Amazon ECR tidak mendukung pemindaian citra yang dibangun dari citra [Scratch Docker](#).

Important

Saat menggunakan pemindaian yang disempurnakan, Amazon Inspector mendukung pemindaian untuk sistem operasi dan jenis media tertentu. Untuk daftar selengkapnya,

lihat [Sistem operasi dan tipe media yang didukung](#) di Panduan Pengguna Amazon Inspector.

Tingkat kepelikan UNDEFINED dikembalikan

Anda mungkin menerima temuan pindaian yang memiliki tingkat kepelikan UNDEFINED. Berikut ini adalah penyebab umum untuk ini:

- Kelemahan tidak ditetapkan sebagai prioritas oleh sumber CVE.
- Kelemahan diberikan prioritas yang tidak dikenali Amazon ECR.

Untuk menentukan tingkat kepelikan dan deskripsi kelemahan, Anda dapat melihat CVE langsung dari sumber.

Memahami status pemindaian **SCAN_ELIGIBILITY_EXPIRED**

Saat pemindaian yang disempurnakan menggunakan Amazon Inspector diaktifkan untuk registri pribadi Anda dan Anda melihat kerentanan pemindaian, Anda mungkin melihat status pemindaian. **SCAN_ELIGIBILITY_EXPIRED** Berikut ini adalah penyebab paling umum dari hal ini.

- Saat Anda awalnya mengaktifkan pemindaian yang disempurnakan untuk registri pribadi Anda, Amazon Inspector hanya mengenali gambar yang didorong ke Amazon ECR dalam 30 hari terakhir, berdasarkan stempel waktu push gambar. Gambar yang lebih tua akan memiliki status **SCAN_ELIGIBILITY_EXPIRED** pemindaian. Jika Anda ingin gambar-gambar ini dipindai oleh Amazon Inspector, Anda harus mendorongnya lagi ke repositori Anda.
- Jika durasi pemindaian ulang ECR diubah di konsol Amazon Inspector dan waktu tersebut berlalu, status pemindaian gambar diubah menjadi `inactive` kode `expired` alasan, dan semua temuan terkait untuk gambar dijadwalkan ditutup. Ini menghasilkan konsol Amazon ECR yang mencantumkan status pemindaian sebagai **SCAN_ELIGIBILITY_EXPIRED**.

Format manifes citra kontainer

Amazon ECR mendukung format manifes citra kontainer berikut:

- Docker Image manifes V2 Skema 1 (digunakan dengan Docker versi 1.9 dan lebih tua)
- Docker Image Manifest V2 Skema 2 (digunakan dengan Docker versi 1.10 dan yang lebih baru)
- Spesifikasi Open Container Initiative (OCI) (v1.0 dan yang lebih tinggi)

Support untuk Docker Image Manifest V2 Skema 2 menyediakan fungsionalitas berikut:

- Kemampuan untuk menggunakan beberapa tanda untuk citra tunggal.
- Support untuk menyimpan citra kontainer Windows. Untuk informasi selengkapnya, lihat [Mendorong Citra Windows ke Amazon ECR](#) dalam Panduan Developer Amazon Elastic Container.

Konversi manifes citra Amazon ECR

Ketika Anda mendorong dan menarik citra ke dan dari Amazon ECR, klien mesin kontainer Anda (misalnya, Docker) berkomunikasi dengan registrasi untuk menyetujui format manifes yang dipahami oleh klien dan registrasi yang akan digunakan untuk citra.

Ketika Anda mendorong citra ke Amazon ECR dengan Docker versi 1.9 atau sebelumnya, format manifes citra disimpan sebagai Docker Image Manifest V2 Skema 1. Ketika Anda mendorong citra ke Amazon ECR dengan Docker versi 1.10 atau yang lebih baru, format manifes citra disimpan sebagai Docker Image Manifest V2 Skema 2.

Ketika Anda menarik citra dari Amazon ECR dengan tanda, maka Amazon ECR mengembalikan format manifes citra yang disimpan dalam repositori. Format akan dikembalikan hanya jika format tersebut dipahami oleh klien. Jika format manifes citra yang disimpan tidak dipahami oleh klien, maka Amazon ECR akan mengkonversi manifes citra tersebut ke dalam format yang dipahami. Sebagai contoh, jika klien Docker 1.9 meminta manifes citra yang disimpan sebagai Docker Image Manifest V2 Skema 2, maka Amazon ECR mengembalikan manifes dalam format Docker Image Manifest V2 Skema 1. Tabel berikut mendeskripsikan konversi yang tersedia dan didukung oleh Amazon ECR ketika citra ditarik dengan tanda:

Skema yang diminta oleh klien	Didorong ke ECR sebagai V2, skema 1	Didorong ke ECR sebagai V2, skema 2	Didorong ke ECR sebagai OCI
V2, skema 1	Tidak memerlukan penerjemahan	Diterjemahkan ke V2, skema 1	Diterjemahkan ke V2, skema 1
V2, skema 2	Tidak ada terjemahan yang tersedia, klien dikembalikan ke V2, skema 1	Tidak memerlukan penerjemahan	Diterjemahkan ke V2, skema 2

Skema yang diminta oleh klien	Didorong ke ECR sebagai V2, skema 1	Didorong ke ECR sebagai V2, skema 2	Didorong ke ECR sebagai OCI
OCI	Penerjemahan tidak tersedia	Diterjemahkan ke OCI	Tidak memerlukan penerjemahan

Important

Jika Anda menarik citra berdasarkan digest, maka tidak ada terjemahan yang tersedia. Klien Anda harus memahami format manifes citra yang disimpan di Amazon ECR. Jika Anda meminta citra Docker Image Manifest V2 Skema 2 berdasarkan digest pada Docker 1.9 atau klien yang lebih lama, maka penarikan citra gagal. Untuk informasi selengkapnya, lihat [Kompabilitas registrasi](#) dalam dokumentasi Docker.

Pada contoh ini, jika Anda meminta citra yang sama berdasarkan tanda, maka Amazon ECR akan menerjemahkan manifes citra ke dalam format yang dapat dipahami klien. Penarikan citra berhasil.

Menggunakan citra Amazon ECR dengan Amazon ECS

Anda dapat menggunakan repositori pribadi Amazon ECR untuk meng-host gambar kontainer dan artefak yang dapat diambil tugas Amazon ECS Anda. Agar ini berfungsi, agen kontainer Amazon ECS, atau Fargate, harus memiliki izin untuk membuat `ecr:BatchGetImage`, `ecr:GetDownloadUrlForLayer`, dan `ecr:GetAuthorizationToken` API.

Izin IAM yang diperlukan

Tabel berikut menunjukkan peran IAM yang akan digunakan, untuk setiap jenis peluncuran, yang menyediakan izin yang diperlukan untuk tugas Anda untuk menarik dari repositori pribadi Amazon ECR. Amazon ECS menyediakan kebijakan IAM terkelola yang menyertakan izin yang diperlukan.

Jenis peluncuran	IAM role	AWSKebijakan IAM terkelola
Instans instans instans Amazon EC2 instans	Gunakan peran IAM instans kontainer, yang dikaitkan	AmazonEC2ContainerServiceforEC2Role

Jenis peluncuran	IAM role	AWSKebijakan IAM terkelola
	dengan instans Amazon EC2 yang terdaftar ke kluster Amazon ECS Anda. Untuk informasi selengkapnya, peran IAM instans kontainer dalam Panduan Developer.	Untuk informasi selengkapnya, lihat AmazonEC2ContainerServiceforEC2Role dalam Panduan Developer Layanan Kontainer Elastis.
ECS di Fargate	Gunakan peran IAM eksekusi tugas yang Anda referensi kan dalam definisi tugas Amazon ECS Anda. Untuk informasi selengkapnya, peran IAM tugas dalam Panduan Developer.	AmazonECSTaskExecutionRolePolicy Untuk informasi selengkapnya, lihat AmazonECSTaskExecutionRolePolicy dalam Panduan Developer.
Instans eksternal	Gunakan peran instans kontainer IAM, yang terkait dengan server on-premise atau mesin virtual (VM) yang terdaftar ke kluster Amazon ECS. Untuk informasi selengkapnya, instans kontainer peran peran instans, dalam Panduan Developer.	AmazonEC2ContainerServiceforEC2Role Untuk informasi selengkapnya, lihat AmazonEC2ContainerServiceforEC2Role dalam Panduan Developer.

Important

Kebijakan IAM yang AWS dikelola berisi izin tambahan yang mungkin tidak Anda perlukan untuk penggunaan Anda. Dalam hal ini, ini adalah izin minimum yang diperlukan untuk menarik dari repositori pribadi Amazon ECR.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```

        "Action": [
            "ecr:BatchGetImage",
            "ecr:GetDownloadUrlForLayer",
            "ecr:GetAuthorizationToken"
        ],
        "Resource": "*"
    }
]
}

```

Penetapan gambar gambar dalam definisi tugas instans

Saat membuat definisi tugas Amazon ECS, Anda dapat menentukan gambar kontainer yang dihosting di repositori pribadi Amazon ECR. Dalam definisi tugas, pastikan Anda menggunakan `registry/repository:tag` penamaan lengkap untuk gambar Amazon ECR Anda. Misalnya, `aws_account_id.dkr.ecr.region.amazonaws.com/my-repository:latest`.

Kutipan definisi tugas berikut menunjukkan sintaks yang akan Anda gunakan untuk menentukan citra kontainer yang dihost oleh Amazon ECR dalam definisi tugas Amazon ECS Anda.

```

{
  "family": "task-definition-name",
  ...
  "containerDefinitions": [
    {
      "name": "container-name",
      "image": "aws_account_id.dkr.ecr.region.amazonaws.com/my-
repository:latest",
      ...
    }
  ],
  ...
}

```

Menggunakan Citra Amazon ECR dengan Amazon EKS

Anda dapat menggunakan citra Amazon ECR Anda dengan Amazon EKS, tetapi Anda perlu memenuhi persyaratan berikut.

- Untuk beban kerja Amazon EKS yang dihosting pada node yang dikelola atau dikelola sendiri, peran IAM (NodeInstanceRole) node pekerja Amazon EKS diperlukan. IAM role simpul pekerja Amazon EKS harus memuat izin kebijakan IAM berikut untuk Amazon ECR.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer",
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

Jika Anda menggunakan `eksctl` atau templat AWS CloudFormation dalam [Memulai dengan Amazon EKS](#) untuk membuat kluster dan grup simpul pekerja Anda, izin IAM ini diterapkan untuk IAM role simpul pekerja Anda secara default.

- Untuk beban kerja Amazon EKS yang di-host AWS Fargate, Anda harus menggunakan peran eksekusi pod Fargate, yang memberikan izin Pod Anda untuk menarik gambar dari repositori Amazon ECR pribadi. Untuk informasi lebih lanjut, lihat [Membuat peran eksekusi pod Fargate](#).
- Ketika mereferensikan citra dari Amazon ECR, Anda harus menggunakan penamaan penuh `registry/repository:tag` untuk citra. Misalnya, `aws_account_id.dkr.ecr.region.amazonaws.com/my-repository:latest`.

Menginstal grafik Helm yang di-host Amazon ECR dengan Amazon EKS

Grafik Helm Anda yang di-host Amazon ECR dapat diinstal pada kluster Amazon EKS Anda. Langkah-langkah berikut menunjukkan caranya.

Prasyarat

Sebelum memulai, pastikan langkah-langkah berikut telah selesai.

- Instal versi terbaru. Langkah-langkah ini ditulis menggunakan versi Helm3 . 9 . 0. Untuk informasi selengkapnya, lihat [Helm](#).
- Anda memiliki setidaknya versi 1 . 23 . 9 atau 2 . 6 . 3 AWS CLI diinstal pada komputer Anda. Untuk informasi selengkapnya, lihat [Menginstal atau memperbarui versi terbaru AWS CLI](#).
- Anda telah mendorong grafik Helm untuk repositori Amazon ECR Anda. Untuk informasi selengkapnya, lihat [Mendorong grafik Helm](#).
- Anda telah mengkonfigurasi kubectl untuk menggunakan Amazon EKS. Untuk informasi selengkapnya, lihat [Buat kubeconfig untuk Amazon EKS](#) dalam Panduan Pengguna Amazon EKS. Jika perintah berikut berhasil diterapkan pada kluster Anda, berarti Anda telah melakukan konfigurasi dengan benar.

```
kubectl get svc
```

Menginstal Amazon ECR yang di-host grafik Helm untuk kluster Amazon EKS

1. autentikasi Helm client Anda ke registrasi Amazon ECR bahwa grafik Helm Anda di-host. Token autentikasi harus diperoleh untuk setiap registrasi yang digunakan, dan token berlaku selama 12 jam. Untuk informasi selengkapnya, lihat [Autentikasi registri pribadi](#).

```
aws ecr get-login-password \
  --region us-west-2 | helm registry login \
  --username AWS \
  --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

2. Instal bagan. Ganti *helm-test-chart* dengan repositori Anda dan *0.1.0* dengan tag bagan Helm Anda.

```
helm install ecr-chart-demo oci://aws_account_id.dkr.ecr.region.amazonaws.com/helm-test-chart --version 0.1.0
```

Outputnya akan terlihat serupa dengan ini:

```
NAME: ecr-chart-demo
LAST DEPLOYED: Tue May 31 17:38:56 2022
NAMESPACE: default
STATUS: deployed
```

```
REVISION: 1
TEST SUITE: None
```

3. Verifikasi instalasi grafik.

```
helm list -n default
```

Contoh keluaran:

NAME	NAMESPACE	REVISION	UPDATED
STATUS	CHART	APP VERSION	
ecr-chart-demo	default	1	2022-06-01 15:56:40.128669157 +0000
UTC deployed	helm-test-chart-0.1.0	1.16.0	

4. (Opsional) Lihat bagan Helm yang diinstalConfigMap.

```
kubectl describe configmap helm-test-chart-configmap
```

5. Setelah selesai, Anda dapat menghapus rilis grafik dari kluster Anda.

```
helm uninstall ecr-chart-demo
```

Citra kontainer Amazon Linux

Citra kontainer Amazon Linux dibangun dari komponen perangkat lunak yang sama yang termasuk dalam Amazon Linux AMI. Ini tersedia untuk digunakan dalam lingkungan apa pun sebagai citra dasar untuk beban kerja Docker. Jika Anda menggunakan Amazon Linux AMI untuk aplikasi di Amazon EC2, Anda dapat menyimpan aplikasi Anda dengan citra kontainer Amazon Linux.

Anda dapat menggunakan image container Amazon Linux di lingkungan pengembangan lokal Anda dan kemudian mendorong aplikasi Anda untuk AWS menggunakan Amazon ECS. Untuk informasi selengkapnya, lihat [Menggunakan citra Amazon ECR dengan Amazon ECS](#).

Gambar wadah Amazon Linux tersedia di Amazon ECR Public dan di [Docker Hub](#). Support untuk citra kontainer Amazon Linux dapat ditemukan dengan mengunjungi [Forum Developer AWS](#).

Untuk menarik gambar wadah Amazon Linux dari Amazon ECR Public

1. Otentikasi klien Docker Anda ke registri Publik Amazon Linux. Token autentikasi berlaku selama 12 jam. Untuk informasi selengkapnya, lihat [Autentikasi registri pribadi](#).

Note

ecr-publicPerintah tersedia di AWS CLI awal dengan versi 1.18.1.187, namun kami sarankan untuk menggunakan versi terbaru dari versi AWS CLI. Untuk informasi selengkapnya, lihat [Menginstal AWS Command Line Interface](#) dalam Panduan Pengguna AWS Command Line Interface .

```
aws ecr-public get-login-password --region us-east-1 | docker login --username AWS
--password-stdin public.ecr.aws
```

Output adalah sebagai berikut:

```
Login succeeded
```

2. Tarik citra kontainer Amazon Linux menggunakan perintah docker pull. Untuk melihat gambar penampungan Amazon Linux di Galeri Publik Amazon ECR, lihat Galeri Publik [Amazon ECR - amazonlinux](#).

```
docker pull public.ecr.aws/amazonlinux/amazonlinux:latest
```

3. (Opsional) Jalankan kontainer secara lokal.

```
docker run -it public.ecr.aws/amazonlinux/amazonlinux /bin/bash
```

Untuk menarik citra kontainer Amazon Linux dari Docker Hub

1. Tarik citra kontainer Amazon Linux menggunakan perintah docker pull.

```
docker pull amazonlinux
```

2. (Opsional) Jalankan kontainer secara lokal.

```
docker run -it amazonlinux:latest /bin/bash
```

Amazon Elastic Container Registry

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model tanggung jawab bersama](#) menjelaskan hal ini sebagai keamanan dari cloud dan keamanan di cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ke tiga menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [Program kepatuhan AWS](#) secara berkala. Untuk mempelajari tentang program kepatuhan yang berlaku di Amazon ECR, lihat [Layanan AWS dalam Cakupan melalui Program Kepatuhan](#).
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon ECR. Topik berikut menunjukkan kepada Anda cara mengonfigurasi Amazon ECR untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Amazon ECR Anda.

Topik

- [Identity and Access Management untuk Amazon Elastic Container Registry](#)
- [Perlindungan data dalam Amazon ECR](#)
- [Validasi kepatuhan Amazon Elastic Container Registry](#)
- [Keamanan Infrastruktur di Amazon Elastic Container Registry](#)

Identity and Access Management untuk Amazon Elastic Container Registry

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang

dapat diautentikasi (masuk) dan diotorisasi (mendapatkan izin) untuk menggunakan sumber daya Amazon ECR. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Bagaimana Amazon Elastic Container Registry bekerja dengan IAM](#)
- [AWS kebijakan terkelola untuk Amazon Elastic Container Registry](#)
- [Menggunakan Peran Terkait Layanan untuk Amazon ECR](#)
- [Pencegahan confused deputy lintas layanan](#)
- [Contoh kebijakan berbasis identitas Amazon Elastic Container Registry](#)
- [Menggunakan Kontrol Akses Berbasis Tanda](#)
- [Pemecahan Masalah Identitas dan Akses Amazon Elastic Container Registry](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Amazon ECR.

Pengguna layanan – Jika Anda menggunakan layanan Amazon ECR untuk melakukan tugas Anda, administrator Anda akan memberikan kredensial dan izin yang dibutuhkan. Saat Anda menggunakan lebih banyak fitur Amazon ECR untuk melakukan pekerjaan, Anda mungkin memerlukan izin tambahan. Memahami bagaimana cara mengelola akses dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Amazon ECR, lihat [Pemecahan Masalah Identitas dan Akses Amazon Elastic Container Registry](#).

Administrator layanan – Jika Anda bertanggung jawab atas sumber daya Amazon ECR di perusahaan Anda, maka Anda mungkin memiliki akses penuh ke Amazon ECR. Tugas Anda adalah menentukan fitur dan sumber daya Amazon ECR mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari lebih lanjut tentang cara perusahaan Anda dapat menggunakan IAM dengan Amazon ECR, lihat [Bagaimana Amazon Elastic Container Registry bekerja dengan IAM](#).

Administrator IAM – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih mendetail tentang cara Anda dapat menulis kebijakan untuk mengelola akses ke Amazon ECR. Untuk melihat contoh kebijakan berbasis identitas Amazon ECR yang dapat Anda gunakan di IAM, lihat [Contoh kebijakan berbasis identitas Amazon Elastic Container Registry](#).

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas gabungan, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat [Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS](#).

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani permintaan AWS API](#) di Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat [Autentikasi multi-faktor](#) dalam Panduan Pengguna AWS IAM Identity Center dan [Menggunakan autentikasi multi-faktor \(MFA\) di AWS](#) dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk

membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari Anda. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya andalkan kredensial sementara daripada membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami sarankan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat [Rotasikan kunci akses secara rutin untuk kasus penggunaan yang memerlukan kredensial jangka panjang](#) dalam Panduan Pengguna IAM.

[Grup IAM](#) adalah identitas yang menentukan kumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, silakan lihat [Kapan harus membuat pengguna IAM \(bukan peran\)](#) dalam Panduan Pengguna IAM.

Peran IAM

[Peran IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan [beralih peran](#). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat [Menggunakan peran IAM](#) dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas gabungan, Anda dapat membuat peran dan menentukan izin untuk peran tersebut. Saat identitas terfederasi mengautentikasi, identitas tersebut akan dikaitkan dengan peran dan diberi izin yang ditentukan oleh peran tersebut. Untuk informasi tentang peran-peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) dalam Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda perlu mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengorelasikan izin yang diatur ke peran dalam IAM. Untuk informasi tentang rangkaian izin, lihat [Rangkaian izin](#) dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (pengguna utama tepercaya) dengan akun berbeda untuk mengakses sumber daya yang ada di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
 - Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Saat Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memulai tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat permintaan FAS, lihat [Teruskan sesi akses](#).
- Peran layanan – Peran layanan adalah [peran IAM](#) yang diambil oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus

peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan dapat menggunakan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensial sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2](#) dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat [Kapan harus membuat peran IAM \(bukan pengguna\)](#) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan dapat menentukan permintaan yang diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan konten dokumen kebijakan JSON, lihat [Gambaran umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat menjalankan peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk operasi. Sebagai contoh, anggap saja Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan pengguna dan peran, di sumber daya mana, dan dengan ketentuan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat dilampirkan ke beberapa pengguna, grup, dan peran dalam akun AWS. Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, lihat [Memilih antara kebijakan terkelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan tersebut, kebijakan ini menentukan jenis tindakan yang dapat dilakukan oleh pengguna utama tertentu di sumber daya tersebut dan apa ketentuannya. Anda harus [menentukan pengguna utama](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau Layanan AWS.

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Jenis kebijakan lainnya

AWS mendukung jenis kebijakan tambahan yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda berdasarkan tipe kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan di mana Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan secara eksplisit terhadap salah satu kebijakan ini akan mengesampingkan izin tersebut. Untuk informasi selengkapnya tentang batasan izin, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) — SCP adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di sebuah organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat [Cara kerja SCP](#) dalam Panduan Pengguna AWS Organizations .
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda teruskan sebagai parameter saat Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Beberapa jenis kebijakan

Ketika beberapa jenis kebijakan berlaku untuk sebuah permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat [Logika evaluasi kebijakan](#) di Panduan Pengguna IAM.

Bagaimana Amazon Elastic Container Registry bekerja dengan IAM

Sebelum menggunakan IAM untuk mengelola akses ke Amazon ECR, Anda harus memahami fitur IAM mana yang tersedia untuk digunakan dengan Amazon ECR. Untuk mendapatkan tampilan tingkat tinggi tentang cara Amazon ECR dan AWS layanan lainnya bekerja dengan IAM, lihat [AWS Layanan yang Bekerja dengan IAM di Panduan Pengguna IAM](#).

Topik

- [Kebijakan berbasis Identitas Amazon ECR](#)
- [Kebijakan berbasis sumber daya Amazon ECR](#)
- [Otorisasi berdasarkan tag Amazon ECR](#)
- [Peran Amazon ECR IAM](#)

Kebijakan berbasis Identitas Amazon ECR

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, serta kondisi di mana tindakan tersebut diperbolehkan atau ditolak. Amazon ECR support tindakan, sumber daya, dan kunci syarat tertentu. Untuk mempelajari semua elemen yang Anda gunakan dalam kebijakan JSON, lihat [Referensi Elemen kebijakan IAM JSON](#) dalam Panduan Pengguna IAM.

Tindakan

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin guna melakukan operasi terkait.

Tindakan kebijakan di Amazon ECR menggunakan prefiks berikut sebelum tindakan: `ecr:` Misalnya, untuk memberikan izin kepada seseorang untuk menggambarkan instans DB dengan operasi `CreateRepository` API Amazon ECR, Anda menyertakan tindakan `ecr:CreateRepository` dalam kebijakan mereka. Pernyataan kebijakan harus memuat elemen `Action` atau `NotAction`. Amazon ECR menentukan serangkaian tindakannya sendiri yang menjelaskan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma seperti berikut:

```
"Action": [
```

```
"ecr:action1",  
"ecr:action2"
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Misalnya, untuk menentukan semua tindakan yang dimulai dengan kata Describe, sertakan tindakan berikut:

```
"Action": "ecr:Describe*"
```

Untuk melihat daftar tindakan Amazon ECR, lihat [Tindakan, Sumber Daya, dan kunci syarat untuk Amazon Elastic Elastic](#) di Panduan Pengguna IAM.

Sumber daya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek atau beberapa objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak support izin tingkat sumber daya, seperti operasi daftar, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan berlaku untuk semua sumber daya.

```
"Resource": "*" 
```

Sumber daya repositori Amazon ECR memiliki ARN berikut:

```
arn:${Partition}:ecr:${Region}:${Account}:repository/${Repository-name}
```

Untuk informasi selengkapnya tentang format ARN, lihat [Nama Sumber Daya Amazon \(ARN\) dan Ruang Nama AWS Layanan](#).

Misalnya, untuk menentukan wilayah `us-east-1` `my-repo` repositori dalam pernyataan Anda, gunakan ARN berikut:

```
"Resource": "arn:aws:ecr:us-east-1:123456789012:repository/my-repo"
```

Untuk menentukan semua repositori milik akun tertentu, gunakan wildcard (*):

```
"Resource": "arn:aws:ecr:us-east-1:123456789012:repository/*"
```

Untuk menentukan beberapa sumber daya dalam pernyataan tunggal, pisahkan ARN dengan koma.

```
"Resource": [  
    "resource1",  
    "resource2"
```

Untuk melihat daftar jenis sumber daya Amazon ECR dan ARN-nya, lihat [Sumber Daya yang Ditetapkan oleh Amazon Elastic Container Registry](#) dalam Panduan Pengguna IAM. Untuk mempelajari tindakan mana yang dapat menentukan ARN setiap sumber daya, lihat [Tindakan yang Ditentukan oleh Amazon Elastic Container Registry](#).

Kunci syarat

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen Condition (atau blok Condition) memungkinkan Anda menentukan kondisi di mana suatu pernyataan akan diterapkan. Elemen Condition bersifat opsional. Anda dapat membuat ekspresi kondisional yang menggunakan [operator kondisi](#), misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen Condition dalam satu pernyataan, atau beberapa kunci dalam satu elemen Condition, AWS akan mengevaluasinya dengan menggunakan operasi AND logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan OR operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Misalnya, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat [Elemen kebijakan IAM: variabel dan tag](#) di Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

Amazon ECR menentukan set kunci syaratnya sendiri dan juga support penggunaan beberapa kunci syarat global. Untuk melihat semua kunci kondisi AWS global, lihat [Kunci Konteks Kondisi AWS Global](#) di Panduan Pengguna IAM.

Semua tindakan Amazon ECR support kunci syarat `aws:ResourceTag` dan `ecr:ResourceTag`. Untuk informasi selengkapnya, lihat [Menggunakan Kontrol Akses Berbasis Tanda](#).

Untuk melihat daftar kunci syarat Amazon ECR, lihat [Kondisi Kunci yang Ditetapkan oleh Amazon Elastic Container Registry](#) dalam Panduan Pengguna IAM. Untuk mempelajari tindakan dan sumber daya yang mana Anda diperbolehkan menggunakan kunci syarat, lihat [Tindakan yang Ditentukan oleh Amazon Elastic Container Registry](#).

Contoh-contoh

Untuk melihat contoh kebijakan berbasis identitas Amazon ECR, lihat [Contoh kebijakan berbasis identitas Amazon Elastic Container Registry](#).

Kebijakan berbasis sumber daya Amazon ECR

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang menentukan tindakan yang dapat dilakukan oleh prinsipal tertentu pada sumber daya Amazon ECR dan dengan syarat apa. Amazon ECR support kebijakan izin berbasis sumber daya untuk repositori Amazon ECR. Kebijakan berbasis sumber daya mengizinkan Anda memberikan izin penggunaan ke akun lain berdasarkan penggunaan sumber daya. Anda juga dapat menggunakan kebijakan berbasis sumber daya untuk mengizinkan layanan AWS mengakses repositori Amazon ECR Anda.

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau entitas IAM di akun lain sebagai [prinsipal dalam kebijakan berbasis sumber daya](#). Menambahkan pengguna utama lintas akun ke kebijakan berbasis sumber daya bagian dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berada di AWS akun yang berbeda, Anda juga harus memberikan izin entitas utama untuk mengakses sumber daya. Berikan izin dengan melampirkan kebijakan berbasis identitas ke entitas tersebut. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat [Bagaimana IAM role Berbeda dengan Kebijakan Berbasis Sumber Daya](#) dalam Panduan Pengguna IAM.

Layanan Amazon ECR hanya support satu jenis kebijakan berbasis sumber daya yang disebut kebijakan repositori, yang terlampir pada repositori. Kebijakan ini menentukan entitas prinsipal

(akun, pengguna, peran, dan pengguna gabungan) yang dapat melakukan tindakan pada repositori. Untuk mempelajari cara melampirkan kebijakan berbasis sumber daya ke repositori, lihat [Kebijakan repositori pribadi](#).

Note

Dalam kebijakan repositori Amazon ECR, elemen kebijakan Sid mendukung karakter tambahan dan spasi yang tidak didukung dalam kebijakan IAM.

Contoh-contoh

Untuk melihat contoh kebijakan berbasis sumber daya Amazon ECR, lihat [Contoh kebijakan repositori pribadi](#),

Otorisasi berdasarkan tag Amazon ECR

Anda dapat melampirkan tanda di sumber daya Amazon ECR atau tanda yang lolos dalam permintaan untuk Amazon ECR. Untuk mengontrol akses berdasarkan tandanya, Anda memberikan informasi tanda di [elemen syarat](#) kebijakan dengan menggunakan kunci syarat `ecr:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`. Untuk informasi lebih lanjut tentang penandaan Amazon ECR sumber daya, lihat [Menandai repositori pribadi](#).

Untuk melihat contoh kebijakan berbasis identitas untuk membatasi akses ke sumber daya berdasarkan tanda pada sumber daya tersebut, lihat [Menggunakan Kontrol Akses Berbasis Tanda](#).

Peran Amazon ECR IAM

[Peran IAM](#) adalah entitas dalam AWS akun Anda yang memiliki izin tertentu.

Menggunakan kredensial sementara dengan Amazon ECR

Anda dapat menggunakan kredensial sementara untuk masuk dengan gabungan, menjalankan IAM role, atau menjalankan peran lintas akun. Anda memperoleh kredensial keamanan sementara dengan memanggil operasi AWS STS API seperti [AssumeRole](#) atau [GetFederationToken](#)

Amazon ECR support penggunaan kredensial sementara.

Peran terkait layanan

[Peran terkait AWS layanan](#) memungkinkan layanan mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran terkait layanan muncul di akun IAM Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Amazon ECR mendukung peran terkait layanan. Untuk informasi selengkapnya, lihat [Menggunakan Peran Terkait Layanan untuk Amazon ECR](#).

AWS kebijakan terkelola untuk Amazon Elastic Container Registry

Amazon ECR menyediakan beberapa kebijakan terkelola yang dapat Anda lampirkan ke pengguna IAM atau instans Amazon EC2. Kebijakan ini mengizinkan tingkat kontrol yang berbeda atas akses ke sumber daya Amazon ECR dan operasi API. Anda dapat menerapkan kebijakan ini secara langsung atau menggunakannya sebagai titik awal untuk membuat kebijakan Anda sendiri. Untuk informasi lebih lanjut tentang setiap operasi API yang disebutkan dalam kebijakan ini, lihat [Tindakan](#) di Referensi Amazon Elastic Container Registry.

Topik

- [AmazonEC2ContainerRegistryFullAccess](#)
- [AmazonEC2ContainerRegistryPowerUser](#)
- [AmazonEC2ContainerRegistryReadOnly](#)
- [AWSECRPullThroughCache_ServiceRolePolicy](#)
- [ECRReplicationServiceRolePolicy](#)
- [Amazon ECR memperbarui kebijakan AWS terkelola](#)

AmazonEC2ContainerRegistryFullAccess

Anda dapat melampirkan kebijakan AmazonEC2ContainerRegistryFullAccess ke identitas IAM.

Anda dapat menggunakan kebijakan terkelola ini sebagai titik awal untuk membuat kebijakan IAM Anda sendiri berdasarkan kebutuhan khusus Anda. Misalnya, Anda dapat membuat kebijakan khusus untuk menyediakan pengguna atau peran dengan akses administrator penuh untuk mengelola penggunaan Amazon ECR. Dengan fitur [Kebijakan Siklus Hidup Amazon ECR](#), Anda dapat menentukan manajemen siklus hidup citra dalam repositori. Peristiwa kebijakan siklus hidup dilaporkan sebagai CloudTrail peristiwa. Amazon ECR terintegrasi AWS CloudTrail sehingga

dapat menampilkan peristiwa kebijakan siklus hidup Anda secara langsung di konsol Amazon ECR. kebijakan IAM terkelola AmazonEC2ContainerRegistryFullAccess mencakup izin `cloudtrail:LookupEvents` untuk memfasilitasi perilaku ini.

Detail izin

Kebijakan ini mencakup izin berikut:

- `ecr` – Mengizinkan prinsipal untuk mendapatkan akses penuh ke semua API Amazon ECR.
- `cloudtrail`— Memungkinkan kepala sekolah untuk mencari acara manajemen atau peristiwa AWS CloudTrail Wawasan yang ditangkap oleh CloudTrail

Kebijakan AmazonEC2ContainerRegistryFullAccess adalah sebagai berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:*",
        "cloudtrail:LookupEvents"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": [
            "replication.ecr.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

AmazonEC2ContainerRegistryPowerUser

Anda dapat melampirkan kebijakan AmazonEC2ContainerRegistryPowerUser ke identitas IAM.

Kebijakan ini memberikan izin administratif yang mengizinkan pengguna IAM membaca dan menulis ke repositori, namun tidak mengizinkan mereka menghapus repositori atau mengubah dokumen kebijakan yang diterapkan di dalamnya.

Detail izin

Kebijakan ini mencakup izin berikut:

- `ecr` – Mengizinkan prinsipal untuk membaca dan menulis ke repositori, serta membaca kebijakan siklus hidup. Prinsipal tidak diberikan izin untuk menghapus repositori atau mengubah kebijakan siklus hidup yang diterapkan padanya.

Kebijakan AmazonEC2ContainerRegistryPowerUser adalah sebagai berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:GetRepositoryPolicy",
        "ecr:DescribeRepositories",
        "ecr:ListImages",
        "ecr:DescribeImages",
        "ecr:BatchGetImage",
        "ecr:GetLifecyclePolicy",
        "ecr:GetLifecyclePolicyPreview",
        "ecr:ListTagsForResource",
        "ecr:DescribeImageScanFindings",
        "ecr:InitiateLayerUpload",
        "ecr:UploadLayerPart",
        "ecr:CompleteLayerUpload",
        "ecr:PutImage"
      ],
      "Resource": "*"
    }
  ],
}
```

```
    }  
  ]  
}
```

AmazonEC2ContainerRegistryReadOnly

Anda dapat melampirkan kebijakan AmazonEC2ContainerRegistryReadOnly ke identitas IAM.

Kebijakan ini memberikan izin baca saja untuk Amazon ECR. Ini termasuk kemampuan untuk membuat daftar repositori dan citra dalam repositori. Ini juga mencakup kemampuan untuk menarik citra dari Amazon ECR dengan Docker CLI.

Detail izin

Kebijakan ini memberikan izin berikut:

- `ecr` – mengizinkan prinsipal untuk membaca repositori dan kebijakan siklus hidupnya masing-masing.

Kebijakan AmazonEC2ContainerRegistryReadOnly adalah sebagai berikut.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "ecr:GetAuthorizationToken",  
        "ecr:BatchCheckLayerAvailability",  
        "ecr:GetDownloadUrlForLayer",  
        "ecr:GetRepositoryPolicy",  
        "ecr:DescribeRepositories",  
        "ecr:ListImages",  
        "ecr:DescribeImages",  
        "ecr:BatchGetImage",  
        "ecr:GetLifecyclePolicy",  
        "ecr:GetLifecyclePolicyPreview",  
        "ecr:ListTagsForResource",  
        "ecr:DescribeImageScanFindings"  
      ],  
      "Resource": "*"   
    }  
  ]  
}
```

```
]
}
```

AWSECRPullThroughCache_ServiceRolePolicy

Anda tidak dapat melampirkan kebijakan IAM AWSECRPullThroughCache_ServiceRolePolicy terkelola ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang memungkinkan Amazon ECR untuk mendorong gambar ke repositori Anda melalui alur kerja pull through cache. Untuk informasi selengkapnya, lihat [Peran terkait layanan Amazon ECR untuk menarik cache](#).

ECRReplicationServiceRolePolicy

Anda tidak dapat melampirkan kebijakan IAM ECRReplicationServiceRolePolicy terkelola ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang mengizinkan Amazon ECR melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat [Menggunakan Peran Terkait Layanan untuk Amazon ECR](#).

Amazon ECR memperbarui kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Amazon ECR sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, silakan berlangganan RSS feed pada halaman riwayat Dokumen Amazon ECR.

Perubahan	Deskripsi	Tanggal
AWSECRPullThroughCache_ServiceRolePolicy — Perbaruan ke kebijakan yang sudah ada	Amazon ECR menambahkan izin baru ke kebijakan AWSECRPullThroughCache_ServiceRolePolicy. Izin ini memungkinkan Amazon ECR untuk mengambil konten terenkripsi dari rahasia Secrets Manager. Ini diperlukan saat menggunakan aturan pull through cache untuk menyimpan gambar dari	15 November 2023

Perubahan	Deskripsi	Tanggal
	registri hulu yang memerlukan otentikasi.	
AWSECRPullThroughCache_ServiceRolePolicy – Kebijakan baru	Amazon ECR menambahkan kebijakan baru. Kebijakan ini dikaitkan dengan peran <code>AWSServiceRoleForECRPullThroughCache</code> terkait layanan untuk fitur pull through cache.	29 November 2021
ECR_ReplicationService_RolePolicy — Kebijakan baru	Amazon ECR menambahkan kebijakan baru. Kebijakan ini dikaitkan dengan peran <code>AWSServiceRoleForECRReplication</code> terkait layanan untuk fitur replikasi.	4 Desember 2020
AmazonEC2 Container RegistryFullAccess - Perbarui ke kebijakan yang ada	Amazon ECR menambahkan izin baru ke kebijakan <code>AmazonEC2ContainerRegistryFullAccess</code> . Izin ini mengizinkan prinsipal untuk membuat peran terkait layanan Amazon ECR.	4 Desember 2020
AmazonEC2 Container RegistryReadOnly - Perbarui ke kebijakan yang ada	Amazon ECR menambahkan izin baru ke kebijakan <code>AmazonEC2ContainerRegistryReadOnly</code> yang mengizinkan prinsipal untuk membaca kebijakan siklus hidup, daftar tanda, dan menjelaskan temuan pemindaian untuk citra.	10 Desember 2019

Perubahan	Deskripsi	Tanggal
AmazonEC2 Container RegistryPowerUser - Perbarui ke kebijakan yang ada	Amazon ECR menambahkan izin baru ke kebijakan AmazonEC2ContainerRegistryPowerUser . Mereka mengizinkan prinsipal untuk membaca kebijakan siklus hidup, daftar tanda, dan menjelaskan temuan pemindaian untuk citra.	10 Desember 2019
AmazonEC2 Container RegistryFullAccess - Perbarui ke kebijakan yang ada	Amazon ECR menambahkan izin baru ke kebijakan AmazonEC2ContainerRegistryFullAccess . Mereka memungkinkan kepala sekolah untuk mencari acara manajemen atau peristiwa AWS CloudTrail Wawasan yang ditangkap oleh CloudTrail	10 November 2017
AmazonEC2 Container RegistryReadOnly - Perbarui ke kebijakan yang ada	Amazon ECR menambahkan izin baru ke kebijakan AmazonEC2ContainerRegistryReadOnly . Mereka mengizinkan prinsipal untuk menggambarkan citra Amazon ECR.	11 Oktober 2016

Perubahan	Deskripsi	Tanggal
AmazonEC2 Container RegistryPowerUser - Perbarui ke kebijakan yang ada	Amazon ECR menambahkan izin baru ke kebijakan AmazonEC2ContainerRegistryPowerUser . Mereka mengizinkan prinsipal untuk menggambarkan citra Amazon ECR.	11 Oktober 2016
AmazonEC2 Container RegistryReadOnly - Kebijakan baru	Amazon ECR menambahkan kebijakan baru yang memberikan izin baca saja untuk Amazon ECR. Izin ini mencakup kemampuan untuk membuat daftar repositori dan citra dalam repositori. Izin ini juga termasuk kemampuan untuk menarik citra dari Amazon ECR dengan Docker CLI.	21 Desember 2015
AmazonEC2 Container RegistryPowerUser - Kebijakan baru	Amazon ECR menambahkan kebijakan baru yang memberikan izin administratif yang memungkinkan users membaca dan menulis ke repositori tetapi tidak mengizinkan mereka untuk menghapus repositori atau mengubah dokumen kebijakan yang diterapkan padanya.	21 Desember 2015
AmazonEC2 Container RegistryFullAccess - Kebijakan baru	Amazon ECR menambahkan kebijakan baru. Kebijakan ini juga memberikan akses penuh ke Amazon ECR.	21 Desember 2015

Perubahan	Deskripsi	Tanggal
Amazon ECR mulai melacak perubahan	Amazon ECR mulai melacak perubahan untuk kebijakan AWS terkelola.	24 Juni 2021

Menggunakan Peran Terkait Layanan untuk Amazon ECR

Amazon Elastic Container Registry (Amazon ECR) AWS Identity and Access Management menggunakan peran [terkait layanan \(IAM\)](#) untuk memberikan izin yang diperlukan untuk menggunakan replikasi dan menarik fitur cache. Peran terkait layanan adalah jenis IAM role unik yang terhubung langsung ke Amazon ECR. Peran terkait layanan yang telah ditetapkan oleh Amazon ECR. Ini mencakup semua izin yang diperlukan layanan untuk mendukung replikasi dan menarik fitur cache untuk registri pribadi Anda. Setelah Anda mengonfigurasi replikasi atau menarik cache untuk registri Anda, peran terkait layanan dibuat secara otomatis atas nama Anda. Untuk informasi selengkapnya, lihat [Pengaturan registri pribadi](#).

Peran terkait layanan membuat pengaturan replikasi dan menarik cache dengan Amazon ECR lebih mudah. Hal ini karena, dengan menggunakannya, Anda tidak perlu menambahkan semua izin yang diperlukan secara manual. Amazon ECR menentukan izin peran terkait layanan, dan kecuali ditentukan lain, hanya Amazon ECR yang dapat menjalankan perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin. Kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran terkait layanan yang sesuai hanya setelah menonaktifkan replikasi atau menarik cache pada registri Anda. Ini memastikan bahwa Anda tidak secara tidak sengaja menghapus izin yang diperlukan Amazon ECR untuk fitur-fitur ini.

Untuk informasi tentang layanan lain yang support peran terkait layanan, lihat [Layanan AWS yang bekerja dengan IAM](#). Pada halaman tertaut ini, cari layanan yang memiliki Ya dalam kolom Peran terkait layanan. Pilih Ya dengan tautan untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Topik

- [Wilayah yang Didukung untuk Peran Terkait Layanan Amazon ECR](#)
- [Peran terkait layanan Amazon ECR untuk replikasi](#)
- [Peran terkait layanan Amazon ECR untuk menarik cache](#)

Wilayah yang Didukung untuk Peran Terkait Layanan Amazon ECR

Amazon ECR mendukung penggunaan peran terkait layanan di semua Wilayah tempat layanan ECR Amazon tersedia. Untuk informasi selengkapnya tentang ketersediaan Wilayah ECR Amazon, lihat [AWS Wilayah dan Titik Akhir](#).

Peran terkait layanan Amazon ECR untuk replikasi

Izin peran terkait layanan untuk Amazon ECR

Amazon ECR menggunakan peran terkait layanan bernama - `AWSServiceRoleForECRReplication` memungkinkan Amazon ECR mereplikasi gambar di beberapa akun..

Peran `AWSServiceRoleForECRReplication` terkait layanan mempercayai layanan berikut untuk mengambil peran:

- `replication.ecr.amazonaws.com`

Berikut ini kebijakan izin peran `ECRReplicationServiceRolePolicy` yang mengizinkan Amazon ECR untuk menggunakan tindakan berikut pada sumber daya:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:CreateRepository",
        "ecr:ReplicateImage"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

`ReplicateImage` adalah API internal yang digunakan oleh Amazon ECR untuk mereplikasi dan tidak dapat disebut secara langsung.

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi lebih lanjut, lihat [Izin Peran Terkait Layanan](#) dalam Panduan Pengguna IAM.

Membuat Peran Terkait Layanan untuk Amazon ECR

Anda tidak perlu membuat peran terkait layanan untuk Amazon ECR secara manual. Saat Anda mengonfigurasi pengaturan replikasi untuk registri Anda di AWS Management Console, API AWS CLI, atau AWS API, Amazon ECR akan membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran terkait layanan ini, lalu ingin membuatnya lagi, Anda dapat menggunakan proses yang sama untuk membuat ulang peran tersebut di akun Anda. Saat Anda mengkonfigurasi pengaturan replikasi untuk registrasi Anda, Amazon ECR membuat peran terkait layanan untuk Anda kembali.

Mengedit Peran Terkait Layanan untuk Amazon ECR

Amazon ECR tidak mengizinkan pengeditan peran `AWSServiceRoleForECRReplication` terkait layanan secara manual. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat menyunting penjelasan peran menggunakan IAM. Untuk informasi lebih lanjut, lihat [Mengedit peran terkait layanan](#) dalam Panduan Pengguna IAM.

Menghapus Peran Terkait Layanan untuk Amazon ECR

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami menyarankan Anda menghapus peran tersebut. Dengan begitu Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Namun, Anda harus menghapus konfigurasi replikasi untuk registrasi Anda di setiap Wilayah sebelum Anda dapat menghapus peran terkait layanan secara manual.

Note

Jika Anda mencoba untuk menghapus sumber daya ketika layanan Amazon ECR masih menggunakan peran, tindakan penghapusan yang Anda lakukan mungkin gagal. Jika hal tersebut terjadi, tunggu beberapa menit dan coba lagi.

Untuk menghapus sumber daya Amazon ECR yang digunakan oleh `AWSServiceRoleForECRReplication`

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah konfigurasi di mana replikasi Anda diatur.
3. Di panel navigasi, pilih Registri pribadi.
4. Pada halaman registri pribadi, pada bagian konfigurasi replikasi, pilih Edit.
5. Untuk menghapus semua aturan replikasi Anda, pilih Hapus semua. Langkah ini membutuhkan konfirmasi.

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran `AWSServiceRoleForECRReplication` terkait layanan. Untuk informasi selengkapnya, silakan lihat [Menghapus Peran Terkait Layanan](#) di Panduan Pengguna IAM.

Peran terkait layanan Amazon ECR untuk menarik cache

Amazon ECR menggunakan peran terkait layanan bernama `AWSServiceRoleForECRPullThroughCache` yang memberikan izin kepada Amazon ECR untuk melakukan tindakan atas nama Anda guna menyelesaikan tindakan pull through cache. Untuk informasi selengkapnya tentang pull through cache, lihat [Menggunakan aturan pull through cache](#).

Izin peran terkait layanan untuk Amazon ECR

Peran `AWSServiceRoleForECRPullThroughCache` terkait layanan mempercayai layanan berikut untuk mengambil peran tersebut.

- `pullthroughcache.ecr.amazonaws.com`

Detail izin

Kebijakan `AWSECRPullThroughCache_ServiceRolePolicy` izin dilampirkan ke peran terkait layanan. Kebijakan terkelola ini memberikan izin Amazon ECR untuk melakukan tindakan berikut. Untuk informasi selengkapnya, lihat [AWSECRPullThroughCache_ServiceRolePolicy](#).

- `ecr`— Memungkinkan layanan Amazon ECR untuk mendorong gambar ke repositori pribadi.

- `secretsmanager:GetSecretValue`— Memungkinkan layanan Amazon ECR untuk mengambil konten rahasia yang dienkripsi. AWS Secrets Manager Ini diperlukan saat menggunakan aturan pull through cache untuk menyimpan gambar dari registri upstream yang memerlukan otentikasi di registri pribadi Anda. Izin ini hanya berlaku untuk rahasia dengan awalan `ecr-pullthroughcache/` nama.

AWSECRPullThroughCache_ServiceRolePolicyKebijakan ini berisi JSON berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ECR",
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:InitiateLayerUpload",
        "ecr:UploadLayerPart",
        "ecr:CompleteLayerUpload",
        "ecr:PutImage"
      ],
      "Resource": "*"
    },
    {
      "Sid": "SecretsManager",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:ecr-pullthroughcache/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    }
  ]
}
```

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat [Izin peran terkait layanan](#) dalam Panduan Pengguna IAM.

Membuat Peran Terkait Layanan untuk Amazon ECR

Anda tidak perlu membuat peran terkait layanan Amazon ECR secara manual untuk menarik cache. Saat Anda membuat aturan cache pull through untuk registri pribadi Anda di AWS Management Console, API AWS CLI, atau AWS API, Amazon ECR membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran terkait layanan ini, lalu ingin membuatnya lagi, Anda dapat menggunakan proses yang sama untuk membuat ulang peran tersebut di akun Anda. Saat Anda membuat aturan cache pull through untuk registri pribadi Anda, Amazon ECR akan membuat peran terkait layanan untuk Anda lagi jika belum ada.

Mengedit Peran Terkait Layanan untuk Amazon ECR

Amazon ECR tidak mengizinkan pengeditan peran `AWSServiceRoleForECRPullThroughCache` terkait layanan secara manual. Setelah peran terkait layanan dibuat, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat mengedit deskripsi peran ini menggunakan IAM. Untuk informasi lebih lanjut, lihat [Mengedit peran terkait layanan](#) dalam Panduan Pengguna IAM.

Menghapus Peran Terkait Layanan untuk Amazon ECR

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami menyarankan Anda menghapus peran tersebut. Dengan begitu Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Namun, Anda harus menghapus aturan pull through cache untuk registri Anda di setiap Wilayah sebelum Anda dapat menghapus peran terkait layanan secara manual.

Note

Jika Anda mencoba menghapus sumber daya saat layanan Amazon ECR masih menggunakan peran tersebut, tindakan penghapusan Anda mungkin gagal. Jika hal tersebut terjadi, tunggu beberapa menit dan coba lagi.

Untuk menghapus sumber daya Amazon ECR yang digunakan oleh peran terkait `AWSServiceRoleForECRPullThroughCache` layanan

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Dari bilah navigasi, pilih Wilayah tempat aturan cache tarik Anda dibuat.
3. Di panel navigasi, pilih Registri pribadi.
4. Pada halaman registri pribadi, pada bagian konfigurasi Tarik melalui cache, pilih Edit.
5. Untuk setiap aturan pull through cache yang telah Anda buat, pilih aturan dan kemudian pilih Hapus aturan.

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran `AWSServiceRoleForECRPullThroughCache` terkait layanan. Untuk informasi selengkapnya, silakan lihat [Menghapus Peran Terkait Layanan](#) di Panduan Pengguna IAM.

Pencegahan confused deputy lintas layanan

Masalah confused deputy adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang lebih berhak untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan pengguna utama layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan [aws:SourceArn](#) atau kunci konteks kondisi [aws:SourceAccount](#) global dalam kebijakan sumber daya untuk membatasi izin yang diberikan Amazon ECR layanan lain ke sumber daya. Gunakan `aws:SourceArn` jika Anda hanya ingin satu sumber daya dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Cara paling efektif untuk melindungi dari masalah confused deputy adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan

kunci kondisi konteks `aws:SourceArn` global dengan karakter wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:servicename:region:123456789012:*`.

Jika `aws:SourceArn` nilainya tidak berisi ID akun, seperti ARN bucket Amazon S3, Anda harus menggunakan kedua kunci konteks kondisi global untuk membatasi izin.

Nilai `aws:SourceArn` harus `ResourceDescription`.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan global dalam kebijakan repositori Amazon ECR untuk mengizinkan AWS CodeBuild akses ke tindakan Amazon ECR API yang diperlukan untuk integrasi dengan layanan tersebut sekaligus mencegah masalah deputy yang membingungkan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer"
      ],
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:codebuild:region:123456789012:project/project-  
name"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

Contoh kebijakan berbasis identitas Amazon Elastic Container Registry

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Amazon ECR. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat menjalankan peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Amazon ECR, termasuk format ARN untuk setiap jenis sumber daya, lihat [Kunci tindakan, sumber daya, dan kondisi untuk Amazon Elastic Container Registry](#) di Referensi Otorisasi Layanan.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat Kebijakan pada Tab JSON](#) dalam Panduan Pengguna IAM.

Topik

- [Praktik Terbaik Kebijakan](#)
- [Menggunakan konsol Amazon ECR](#)
- [Izinkan Pengguna untuk Melihat Izin Mereka Sendiri](#)
- [Mengakses Satu Repositori Amazon ECR](#)

Praktik Terbaik Kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Amazon ECR di akun Anda. Tindakan ini dikenai biaya untuk Akun AWS Anda. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi

selengkapnya, lihat [kebijakan yang dikelola AWS](#) atau [kebijakan yang dikelola AWS untuk fungsi pekerjaan](#) di Panduan Pengguna IAM.

- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukan ini dengan menentukan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk menerapkan izin, lihat [Kebijakan dan izin di IAM](#) di Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Syarat](#) di Panduan Pengguna IAM.
- Menggunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda guna memastikan izin yang aman dan berfungsi – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat [validasi kebijakan Analizer Akses IAM](#) di Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk mewajibkan MFA saat operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat [Mengonfigurasi akses API yang dilindungi MFA](#) di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan di IAM](#) di Panduan Pengguna IAM.

Menggunakan konsol Amazon ECR

Untuk mengakses konsol Amazon Elastic Container Registry, Anda harus memiliki rangkaian izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Amazon ECR di akun Anda AWS . Jika Anda membuat kebijakan berbasis identitas

yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Untuk memastikan bahwa entitas tersebut masih dapat menggunakan konsol Amazon ECR, tambahkan kebijakan `AmazonEC2ContainerRegistryReadOnly` AWS terkelola ke entitas. Untuk informasi selengkapnya, lihat [Menambahkan Izin ke Pengguna](#) dalam Panduan Pengguna IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:GetRepositoryPolicy",
        "ecr:DescribeRepositories",
        "ecr:ListImages",
        "ecr:DescribeImages",
        "ecr:BatchGetImage",
        "ecr:GetLifecyclePolicy",
        "ecr:GetLifecyclePolicyPreview",
        "ecr:ListTagsForResource",
        "ecr:DescribeImageScanFindings"
      ],
      "Resource": "*"
    }
  ]
}
```

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai alternatif, hanya izinkan akses ke tindakan yang cocok dengan operasi API yang sedang Anda coba lakukan.

Izinkan Pengguna untuk Melihat Izin Mereka Sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan para pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. `AWS CLI AWS`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

Mengakses Satu Repositori Amazon ECR

Dalam contoh ini, Anda ingin memberikan pengguna di AWS akun Anda akses ke salah satu repositori Amazon ECR Anda, `my-repo` Anda juga ingin mengizinkan pengguna untuk mendorong, menarik, dan memasukkan citra.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

    {
      "Sid": "ListImagesInRepository",
      "Effect": "Allow",
      "Action": [
        "ecr:ListImages"
      ],
      "Resource": "arn:aws:ecr:us-east-1:123456789012:repository/my-repo"
    },
    {
      "Sid": "GetAuthorizationToken",
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ManageRepositoryContents",
      "Effect": "Allow",
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:GetRepositoryPolicy",
        "ecr:DescribeRepositories",
        "ecr:ListImages",
        "ecr:DescribeImages",
        "ecr:BatchGetImage",
        "ecr:InitiateLayerUpload",
        "ecr:UploadLayerPart",
        "ecr:CompleteLayerUpload",
        "ecr:PutImage"
      ],
      "Resource": "arn:aws:ecr:us-east-1:123456789012:repository/my-repo"
    }
  ]
}

```

Menggunakan Kontrol Akses Berbasis Tanda

Tindakan Amazon ECR `CreateRepository` API memungkinkan Anda menentukan tag saat membuat repositori. Untuk informasi selengkapnya, lihat [Menandai repositori pribadi](#).

Untuk mengizinkan pengguna menandai repositori pada pembuatan, mereka harus memiliki izin untuk menggunakan tindakan yang membuat sumber daya (misalnya, `ecr:CreateRepository`). Jika tanda ditentukan dalam aksi pembuatan sumber daya, Amazon melakukan otorisasi tambahan pada tindakan `ecr:CreateRepository` untuk memverifikasi apakah pengguna memiliki izin untuk membuat tanda.

Anda dapat menggunakan kontrol akses berbasis tag melalui kebijakan IAM. Berikut ini adalah beberapa contohnya.

Kebijakan berikut hanya akan mengizinkan pengguna untuk membuat atau menandai repositori sebagai `key=environment,value=dev`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateTaggedRepository",
      "Effect": "Allow",
      "Action": [
        "ecr:CreateRepository"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "dev"
        }
      }
    },
    {
      "Sid": "AllowTagRepository",
      "Effect": "Allow",
      "Action": [
        "ecr:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "dev"
        }
      }
    }
  ]
}
```

```
}
```

Kebijakan berikut akan memungkinkan pengguna mengakses semua repositori kecuali mereka ditandai sebagai `key=environment,value=prod`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ecr:*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": "ecr:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ecr:ResourceTag/environment": "prod"
        }
      }
    }
  ]
}
```

Pemecahan Masalah Identitas dan Akses Amazon Elastic Container Registry

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan mengatasi masalah umum yang mungkin Anda temui saat bekerja menggunakan Amazon ECR dan IAM.

Topik

- [Saya tidak Berwenang untuk Melakukan Tindakan di Amazon ECR](#)
- [Saya Tidak Berwenang untuk Melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Amazon ECR saya](#)

Saya tidak Berwenang untuk Melakukan Tindakan di Amazon ECR

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya `my-example-widget` rekaan, tetapi tidak memiliki izin `ecr:GetWidget` rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
ecr:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya `my-example-widget` dengan menggunakan tindakan `ecr:GetWidget`.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya Tidak Berwenang untuk Melakukan `iam:PassRole`

Jika Anda menerima kesalahan bahwa Anda tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke Amazon ECR.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi saat pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di Amazon ECR. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Amazon ECR saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau pengguna di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi pengguna akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, lihat hal berikut:

- Untuk mempelajari apakah Amazon ECR support fitur ini, lihat [Bagaimana Amazon Elastic Container Registry bekerja dengan IAM](#).
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki](#) di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat [Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Memberikan akses kepada pengguna eksternal yang sah \(federasi identitas\)](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan kebijakan berbasis peran dan sumber daya untuk akses lintas akun, lihat [Bagaimana IAM role berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.

Perlindungan data dalam Amazon ECR

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di Amazon Elastic Container Service. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Amazon ECS atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Topik

- [Enkripsi diam](#)

Enkripsi diam

Amazon ECR menyimpan citra di bucket Amazon S3 yang dikelola Amazon ECR. Secara default, Amazon ECR menggunakan enkripsi server-side dengan kunci enkripsi terkelola Amazon S3 yang mengenkripsi data Anda saat tidak aktif menggunakan algoritme enkripsi AES-256. Ini tidak

memerlukan tindakan apa pun dari bagian Anda dan ditawarkan tanpa biaya tambahan. Untuk informasi selengkapnya, lihat [Melindungi Data Menggunakan Enkripsi Sisi Server dengan Kunci Enkripsi Terkelola Amazon S3 \(SSE-S3\)](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Untuk kontrol lebih lanjut atas enkripsi untuk repositori Amazon ECR Anda, Anda dapat menggunakan enkripsi sisi server dengan kunci KMS yang disimpan di (). AWS Key Management Service AWS KMS Saat Anda menggunakan AWS KMS untuk mengenkripsi data, Anda dapat menggunakan default Kunci yang dikelola AWS, yang dikelola oleh Amazon ECR, atau menentukan kunci KMS Anda sendiri (disebut sebagai kunci yang dikelola pelanggan). Untuk informasi selengkapnya, lihat [Melindungi Data Menggunakan Enkripsi Sisi Server dengan kunci KMS yang Disimpan di AWS KMS \(SSE-KMS\) di](#) Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Setiap repositori Amazon ECR memiliki konfigurasi enkripsi, yang diatur saat repositori tersebut dibuat. Anda dapat menggunakan konfigurasi enkripsi yang berbeda pada setiap repositori. Untuk informasi selengkapnya, lihat [Membuat repositori pribadi](#).

Ketika repositori dibuat dengan AWS KMS enkripsi diaktifkan, kunci KMS digunakan untuk mengenkripsi isi repositori. Selain itu, Amazon ECR menambahkan AWS KMS hibah ke kunci KMS dengan repositori Amazon ECR sebagai pokok penerima hibah.

Berikut ini memberikan pemahaman tingkat tinggi tentang bagaimana Amazon ECR terintegrasi dengan AWS KMS untuk mengenkripsi dan mendekripsi repositori Anda:

1. Saat membuat repositori, Amazon ECR mengirimkan [DescribeKey](#) panggilan AWS KMS untuk memvalidasi dan mengambil Nama Sumber Daya Amazon (ARN) dari kunci KMS yang ditentukan dalam konfigurasi enkripsi.
2. Amazon ECR mengirimkan dua [CreateGrant](#) permintaan AWS KMS untuk membuat hibah pada kunci KMS untuk memungkinkan Amazon ECR mengenkripsi dan mendekripsi data menggunakan kunci data.
3. Saat mendorong gambar, [GenerateDataKey](#) permintaan dibuat untuk AWS KMS yang menentukan kunci KMS yang akan digunakan untuk mengenkripsi lapisan gambar dan manifes.
4. AWS KMS menghasilkan kunci data baru, mengenkripsi di bawah kunci KMS yang ditentukan, dan mengirimkan kunci data terenkripsi untuk disimpan dengan metadata lapisan gambar dan manifes gambar.
5. Saat menarik gambar, permintaan [Dekripsi](#) dibuat untuk AWS KMS, menentukan kunci data terenkripsi.

6. AWS KMS mendekripsi kunci data terenkripsi dan mengirimkan kunci data yang didekripsi ke Amazon S3.
7. Kunci data digunakan untuk mendekripsi layer gambar sebelum layer gambar ditarik.
8. Ketika repositori dihapus, Amazon ECR mengirimkan dua [RetireGrant](#) permintaan untuk menghentikan hibah yang AWS KMS dibuat untuk repositori.

Pertimbangan

Poin-poin berikut harus dipertimbangkan saat menggunakan AWS KMS enkripsi dengan Amazon ECR.

- Jika Anda membuat repositori Amazon ECR Anda dengan enkripsi KMS dan Anda tidak menentukan kunci KMS, Amazon ECR menggunakan Kunci yang dikelola AWS dengan alias secara default. `aws/ecr` Kunci KMS ini dibuat di akun Anda saat pertama kali Anda membuat repositori dengan enkripsi KMS yang diaktifkan.
- Saat Anda menggunakan enkripsi KMS dengan kunci KMS Anda sendiri, kunci harus ada di wilayah yang sama sebagai repositori Anda.
- Bantuan yang dibuat Amazon ECR atas nama Anda tidak harus dicabut. Jika Anda mencabut hibah yang memberikan izin Amazon ECR untuk menggunakan AWS KMS kunci di akun Anda, Amazon ECR tidak dapat mengakses data ini, mengenkripsi gambar baru yang didorong ke repositori, atau mendekripsi ketika ditarik. Ketika Anda mencabut bantuan untuk Amazon ECR, perubahan terjadi segera. Untuk mencabut hak akses, Anda harus menghapus repositori bukan mencabut bantuan. Ketika repositori dihapus, Amazon ECR pensiunkan (retire) bantuan atas nama Anda.
- Ada biaya yang terkait dengan penggunaan AWS KMS kunci. Untuk informasi lebih lanjut, lihat [Harga AWS Key Management Service](#).

Izin IAM yang Diperlukan

Saat membuat atau menghapus repositori Amazon ECR dengan enkripsi server-side menggunakan AWS KMS, izin yang diperlukan tergantung pada kunci KMS tertentu yang Anda gunakan.

Izin IAM yang diperlukan saat menggunakan untuk Kunci yang dikelola AWS Amazon ECR

Secara default, ketika AWS KMS enkripsi diaktifkan untuk repositori Amazon ECR tetapi tidak ada kunci KMS yang ditentukan, ECR untuk Kunci yang dikelola AWS Amazon digunakan. Ketika kunci

KMS AWS-managed untuk Amazon ECR digunakan untuk mengenkripsi repositori, setiap prinsipal yang memiliki izin untuk membuat repositori juga dapat mengaktifkan enkripsi pada repositori. AWS KMS Namun, IAM principal yang menghapus repositori harus memiliki izin `kms:RetireGrant`. Hal ini memungkinkan pensiun dari hibah yang ditambahkan ke AWS KMS kunci ketika repositori dibuat.

Contoh kebijakan IAM berikut dapat ditambahkan sebagai kebijakan inline untuk pengguna guna memastikan bahwa mereka memiliki izin minimum yang diperlukan untuk menghapus repositori yang memiliki enkripsi yang diaktifkan. Kunci KMS yang digunakan untuk mengenkripsi repositori dapat ditentukan menggunakan parameter sumber daya.

```
{
  "Version": "2012-10-17",
  "Id": "ecr-kms-permissions",
  "Statement": [
    {
      "Sid": "AllowAccessToRetireTheGrantsAssociatedWithTheKey",
      "Effect": "Allow",
      "Action": [
        "kms:RetireGrant"
      ],
      "Resource": "arn:aws:kms:us-  
west-2:111122223333:key/b8d9ae76-080c-4043-92EXAMPLE"
    }
  ]
}
```

Izin IAM yang diperlukan saat menggunakan kunci terkelola pelanggan

Saat membuat repositori dengan AWS KMS enkripsi diaktifkan menggunakan kunci yang dikelola pelanggan, ada izin yang diperlukan untuk kebijakan kunci KMS dan kebijakan IAM untuk pengguna atau peran yang membuat repositori.

Saat membuat kunci KMS Anda sendiri, Anda dapat menggunakan kebijakan kunci default AWS KMS membuatnya atau Anda dapat menentukan sendiri. Untuk memastikan bahwa kunci yang dikelola pelanggan tetap dapat dikelola oleh pemilik akun, kebijakan kunci untuk kunci KMS harus memungkinkan semua AWS KMS tindakan untuk pengguna root akun. Izin cakupan tambahan dapat ditambahkan ke kebijakan kunci tetapi pengguna root paling tidak harus diberikan izin untuk mengelola kunci KMS. Untuk mengizinkan kunci KMS digunakan hanya untuk permintaan yang berasal dari Amazon ECR, Anda dapat menggunakan [kunci ViaService kondisi kms:](#) dengan nilainya. `ecr.<region>.amazonaws.com`

Contoh kebijakan kunci berikut memberikan AWS akun (pengguna root) yang memiliki kunci KMS akses penuh ke kunci KMS. Untuk informasi selengkapnya tentang kebijakan kunci contoh ini, lihat [Mengizinkan akses ke AWS akun dan mengaktifkan kebijakan IAM](#) di Panduan AWS Key Management Service Pengembang.

```
{
  "Version": "2012-10-17",
  "Id": "ecr-key-policy",
  "Statement": [
    {
      "Sid": "EnableIAMUserPermissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    }
  ]
}
```

Pengguna IAM, peran IAM, atau AWS akun yang membuat repositori Anda harus memiliki `kms:CreateGrant`, `kms:RetireGrant`, dan `kms:DescribeKey` izin selain izin Amazon ECR yang diperlukan.

Note

Izin `kms:RetireGrant` harus ditambahkan ke kebijakan IAM untuk pengguna atau peran yang membuat repositori. Izin `kms:CreateGrant` dan `kms:DescribeKey` dapat ditambahkan ke kebijakan kunci untuk kunci KMS atau kebijakan IAM untuk pengguna atau peran yang membuat repositori. Untuk informasi selengkapnya tentang cara kerja AWS KMS izin, lihat [Izin AWS KMS API: Referensi tindakan dan sumber daya](#) di Panduan AWS Key Management Service Pengembang.

Contoh kebijakan IAM berikut dapat ditambahkan sebagai kebijakan inline untuk pengguna guna memastikan bahwa mereka memiliki izin minimum yang diperlukan untuk membuat repositori dengan enkripsi yang diaktifkan dan menghapus repositori ketika mereka selesai. Yang AWS KMS key digunakan untuk mengenkripsi repositori dapat ditentukan menggunakan parameter sumber daya.

```
{
  "Version": "2012-10-17",
  "Id": "ecr-kms-permissions",
  "Statement": [
    {
      "Sid":
"AllowAccessToCreateAndRetireTheGrantsAssociatedWithTheKeyAsWellAsDescribeTheKey",
      "Effect": "Allow",
      "Action": [
        "kms:CreateGrant",
        "kms:RetireGrant",
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:us-
west-2:111122223333:key/b8d9ae76-080c-4043-92EXAMPLE"
    }
  ]
}
```

Izinkan pengguna mencantumkan kunci KMS di konsol saat membuat repositori

Bila menggunakan konsol Amazon ECR untuk membuat repositori, Anda dapat memberikan izin untuk mengizinkan pengguna memasukkan kunci KMS yang dikelola pelanggan di Wilayah saat mengaktifkan enkripsi untuk repositori. Contoh kebijakan IAM berikut menunjukkan izin yang diperlukan untuk memasukkan kunci KMS dan alias saat menggunakan konsol.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:ListKeys",
      "kms:ListAliases",
      "kms:DescribeKey"
    ],
    "Resource": "*"
  }
}
```


Pemantauan Interaksi Amazon ECR dengan AWS KMS

Anda dapat menggunakan AWS CloudTrail untuk melacak permintaan yang dikirimkan Amazon ECR atas nama Anda. AWS KMS Entri log di CloudTrail log berisi kunci konteks enkripsi untuk membuatnya lebih mudah diidentifikasi.

Konteks enkripsi Amazon ECR

Konteks enkripsi adalah seperangkat pasangan nilai kunci yang berisi data non-rahasia yang berubah-ubah. Ketika Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, secara AWS KMS kriptografis mengikat konteks enkripsi ke data terenkripsi. Untuk mendekripsi data, Anda harus meneruskan konteks enkripsi yang sama.

Dalam permintaannya [GenerateDataKey](#) dan [Dekripsi](#) ke, AWS KMS Amazon ECR menggunakan konteks enkripsi dengan dua pasangan nama-nilai yang mengidentifikasi repositori dan bucket Amazon S3 yang digunakan. Ini seperti yang ditunjukkan dalam contoh berikut. Nama-nama tidak bervariasi, tetapi nilai-nilai konteks enkripsi gabungan akan berbeda untuk setiap nilai.

```
"encryptionContext": {
  "aws:s3:arn": "arn:aws:s3::us-west-2-starport-manifest-bucket/EXAMPLE1-90ab-cdef-fedc-ba987BUCKET1/sha256:a7766145a775d39e53a713c75b6fd6d318740e70327aaa3ed5d09e0ef33fc3df",
  "aws:ecr:arn": "arn:aws:ecr:us-west-2:111122223333:repository/repository-name"
}
```

Anda dapat menggunakan konteks enkripsi untuk mengidentifikasi operasi kriptografi ini dalam catatan audit dan log, seperti [AWS CloudTrail](#) dan Amazon CloudWatch Logs, dan sebagai syarat untuk otorisasi dalam kebijakan dan hibah.

Enkripsi konteks Amazon ECR terdiri dari dua pasangan nama-nilai.

- `aws:s3:arn` – Pasangan nama-nilai pertama mengidentifikasi bucket. Kuncinya adalah `aws:s3:arn`. Nilai tersebut adalah Amazon Resource Name (ARN) dari bucket Amazon S3.

```
"aws:s3:arn": "ARN of an Amazon S3 bucket"
```

Sebagai contoh, jika ARN bucket adalah `arn:aws:s3::us-west-2-starport-manifest-bucket/EXAMPLE1-90ab-cdef-fedc-ba987BUCKET1/sha256:a7766145a775d39e53a713c75b6fd6d318740e70327aaa3ed5d09e0ef33fc3df`, maka konteks enkripsi akan mencakup pasangan berikut.

```
"arn:aws:s3:::us-west-2-starport-manifest-bucket/EXAMPLE1-90ab-cdef-fedc-ba987BUCKET1/sha256:a7766145a775d39e53a713c75b6fd6d318740e70327aaa3ed5d09e0ef33fc3df"
```

- `aws:ecr:arn` – Pasangan nama-nilai kedua mengidentifikasi Amazon Resource Name (ARN) dari repositori. Kuncinya adalah `aws:ecr:arn`. Nilai tersebut merupakan ARN dari repositori.

```
"aws:ecr:arn": "ARN of an Amazon ECR repository"
```

Misalnya, jika ARN repositori adalah `arn:aws:ecr:us-west-2:111122223333:repository/repository-name`, maka konteks enkripsi akan mencakup pasangan berikut.

```
"aws:ecr:arn": "arn:aws:ecr:us-west-2:111122223333:repository/repository-name"
```

Pemecahan Masalah

Ketika menghapus repositori Amazon ECR dengan konsol, jika repositori berhasil dihapus tetapi Amazon ECR tidak dapat mempersiapkan (retire) bantuan yang ditambahkan ke kunci KMS Anda untuk repositori Anda, maka Anda akan menerima pesan kesalahan berikut.

```
The repository [{repository-name}] has been deleted successfully but the grants created by the kmsKey [{kms_key}] failed to be retired
```

Ketika ini terjadi, Anda dapat mempersiapkan AWS KMS hibah untuk repositori sendiri.

Untuk pensiun AWS KMS hibah untuk repositori secara manual

1. Buat daftar hibah untuk AWS KMS kunci yang digunakan untuk repositori. Nilai `key-id` termasuk dalam pesan kesalahan yang Anda terima dari konsol. Anda juga dapat menggunakan `list-keys` perintah untuk mencantumkan kunci KMS Kunci yang dikelola AWS dan yang dikelola pelanggan di Wilayah tertentu di akun Anda.

```
aws kms list-grants \
  --key-id b8d9ae76-080c-4043-9237-c815bfc21dfc
  --region us-west-2
```

Outputnya termasuk `EncryptionContextSubset` dengan Amazon Resource Name (ARN) dari repositori Anda. Ini dapat digunakan untuk menentukan bantuan yang ditambahkan ke kunci mana yang ingin Anda pensiunkan (retire). Nilai `GrantId` akan digunakan saat memensiunkan (retire) bantuan pada langkah berikutnya.

2. Pensiun setiap hibah untuk AWS KMS kunci yang ditambahkan untuk repositori. Ganti nilai untuk `GrantId` dengan ID hibah dari output dari langkah sebelumnya.

```
aws kms retire-grant \  
  --key-id b8d9ae76-080c-4043-9237-c815bfc21dfc \  
  --grant-id GrantId \  
  --region us-west-2
```

Validasi kepatuhan Amazon Elastic Container Registry

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat [Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS](#) dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat [Program AWS Kepatuhan Program AWS](#) .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#) .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- [Arsitektur untuk Keamanan dan Kepatuhan HIPAA di Amazon Web Services](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

Note

Tidak semua memenuhi Layanan AWS syarat HIPAA. Untuk informasi selengkapnya, lihat [Referensi Layanan yang Memenuhi Syarat HIPAA](#).

- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Panduan Kepatuhan Pelanggan](#) - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk mengevaluasi sumber daya AWS Anda dan memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [Referensi kontrol Security Hub](#).
- [AWS Audit Manager](#) Ini Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Keamanan Infrastruktur di Amazon Elastic Container Registry

Sebagai layanan terkelola, Amazon Elastic Container Registry dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat [Keamanan AWS Cloud](#). Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik Pilar Keamanan](#).

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses Amazon ECR melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan pengguna utama IAM. Atau Anda bisa menggunakan [AWS Security Token Service](#) (AWS STS) untuk membuat kredensial keamanan sementara untuk menandatangani permintaan masuk.

Anda dapat menghubungi operasi API ini dari lokasi jaringan mana pun, tetapi Amazon ECR support kebijakan akses berbasis sumber daya, yang dapat mencakup pembatasan berdasarkan alamat IP sumber. Anda juga dapat menggunakan kebijakan Amazon ECR untuk mengontrol akses dari titik akhir Amazon Virtual Private Cloud (Amazon VPC) atau VPC tertentu. Secara efektif, ini mengisolasi akses jaringan ke sumber daya Amazon ECR tertentu hanya dari VPC tertentu dalam jaringan. AWS Untuk informasi selengkapnya, lihat [Titik akhir VPC antarmuka Amazon ECR \(AWS PrivateLink\)](#).

Titik akhir VPC antarmuka Amazon ECR (AWS PrivateLink)

Anda dapat meningkatkan postur keamanan VPC Anda dengan mengonfigurasi Amazon ECR untuk menggunakan antarmuka VPC endpoint. Titik akhir VPC didukung oleh AWS PrivateLink, teknologi yang memungkinkan Anda mengakses Amazon ECR API secara pribadi melalui alamat IP pribadi. AWS PrivateLink membatasi semua lalu lintas jaringan antara VPC Anda dan Amazon ECR ke jaringan Amazon. Anda tidak memerlukan sebuah gateway internet, perangkat NAT, atau gateway privat virtual.

Untuk informasi selengkapnya tentang AWS PrivateLink dan titik akhir VPC, lihat Titik [Akhir VPC di Panduan Pengguna Amazon VPC](#).

Pertimbangan untuk VPC endpoint Amazon ECR

Sebelum Anda mengkonfigurasi VPC endpoint untuk Amazon ECR, perhatikan pertimbangan berikut.

- Untuk mengizinkan tugas Amazon ECS yang dihosting di instans Amazon EC2 menarik gambar pribadi dari Amazon ECR, pastikan Anda juga membuat titik akhir VPC antarmuka untuk Amazon ECS. Untuk informasi selengkapnya, lihat [Titik Akhir VPC Antarmuka \(AWS PrivateLink\)](#) di Panduan Pengembang Layanan Amazon Elastic Container.

Important

Tugas Amazon ECS yang dihosting di Fargate tidak memerlukan titik akhir VPC antarmuka Amazon ECS.

- Tugas Amazon ECS yang dihosting di Fargate menggunakan **1.3.0** versi platform Linux atau sebelumnya hanya memerlukan `com.amazonaws.region.ecr.dkr` Titik akhir Amazon ECR VPC dan titik akhir gateway Amazon S3 untuk memanfaatkan fitur ini.
- Tugas Amazon ECS yang dihosting di Fargate menggunakan **1.4.0** versi platform Linux atau yang lebih baru memerlukan `com.amazonaws.wilayah.ecr.dkr` dan `com.amazonaws.region.ecr.api` Titik akhir Amazon ECR VPC serta titik akhir gateway Amazon S3 untuk memanfaatkan fitur ini.
- Tugas Amazon ECS yang dihosting di Fargate menggunakan **1.0.0** versi platform Windows atau yang lebih baru memerlukan `com.amazonaws.wilayah.ecr.dkr` dan `com.amazonaws.region.ecr.api` Titik akhir Amazon ECR VPC serta titik akhir gateway Amazon S3 untuk memanfaatkan fitur ini.
- Tugas Amazon ECS yang dihosting di Fargate yang menarik gambar kontainer dari Amazon ECR dapat membatasi akses ke VPC tertentu yang digunakan tugas mereka dan ke titik akhir VPC yang digunakan layanan dengan menambahkan kunci kondisi ke peran IAM eksekusi tugas untuk tugas tersebut. Untuk informasi selengkapnya, lihat [Izin IAM Opsional untuk Tugas Fargate dalam Menarik Citra Amazon ECR melalui Antarmuka Titik Akhir](#) dalam Panduan Developer Amazon Elastic Container Service.
- Tugas Amazon ECS yang dihosting di Fargate yang menarik gambar kontainer dari Amazon ECR yang juga menggunakan `awslogs` driver log untuk mengirim informasi log ke Log memerlukan titik akhir CloudWatch VPC CloudWatch Log. Untuk informasi selengkapnya, lihat [Buat titik akhir CloudWatch Log](#).
- Grup keamanan yang terkait dengan VPC endpoint harus mengizinkan koneksi masuk pada port 443 dari subnet privat VPC.
- VPC endpoint saat ini tidak support permintaan lintas wilayah. Pastikan bahwa Anda membuat VPC endpoint Anda di Wilayah yang sama tempat Anda berencana untuk mengeluarkan panggilan API ke Amazon ECR.
- Titik akhir VPC saat ini tidak mendukung repositori Publik Amazon ECR. Pertimbangkan untuk menggunakan aturan pull through cache untuk meng-host citra publik di repositori pribadi di Wilayah yang sama dengan titik akhir VPC. Untuk informasi selengkapnya, lihat [Menggunakan aturan pull through cache](#).
- Titik akhir VPC hanya mendukung DNS yang AWS disediakan melalui Amazon Route 53. Jika Anda ingin menggunakan DNS Anda sendiri, Anda dapat menggunakan penerusan DNS bersyarat. Untuk informasi selengkapnya, lihat [Pengaturan DHCP](#) dalam Panduan Pengguna Amazon VPC.
- Jika kontainer Anda memiliki koneksi yang tersedia ke Amazon S3, koneksi mereka mungkin akan terganggu sebentar ketika Anda menambahkan titik akhir gateway Amazon S3. Jika Anda ingin

menghindari gangguan ini, buatlah VPC baru yang menggunakan titik akhir gateway Amazon S3 dan kemudian migrasikan kluster Amazon ECS dan kontainer ke VPC baru.

- Saat gambar ditarik menggunakan aturan cache tarik untuk pertama kalinya, jika Anda telah mengonfigurasi Amazon ECR untuk AWS PrivateLink menggunakan titik akhir VPC antarmuka, maka Anda perlu membuat subnet publik di VPC yang sama, dengan gateway NAT, dan kemudian merutekan semua lalu lintas keluar ke internet dari subnet pribadinya ke gateway NAT agar tarikan berfungsi. Penarikan gambar berikutnya tidak memerlukan ini. Untuk informasi selengkapnya, lihat [Skenario: Mengakses internet dari subnet pribadi](#) di Panduan Pengguna Amazon Virtual Private Cloud.

Pertimbangan untuk citra Windows

Citra yang didasarkan pada sistem operasi Windows termasuk artefak yang distribusinya dibatasi oleh lisensi. Secara default, ketika Anda mendorong citra Windows ke repositori Amazon ECR, lapisan yang menyertakan artefak ini tidak didorong karena mereka dianggap sebagai lapisan asing. Ketika artefak disediakan oleh Microsoft, lapisan asing diambil dari infrastruktur Microsoft Azure. Dengan demikian, untuk mengaktifkan kontainer Anda untuk menarik lapisan asing ini dari Azure, langkah-langkah tambahan selain membuat VPC endpoint diperlukan.

Hal ini dimungkinkan untuk menimpa perilaku ini ketika mendorong citra Windows ke Amazon ECR dengan menggunakan tanda `--allow-nondistributable-artifacts` di Docker daemon. Bila diaktifkan, tanda ini akan mendorong lapisan berlisensi untuk Amazon ECR yang mengizinkan citra ini ditarik dari Amazon ECR melalui VPC endpoint tanpa memerlukan akses tambahan ke Azure.

Important

Menggunakan tanda `--allow-nondistributable-artifacts` tidak menghalangi kewajiban Anda untuk mematuhi persyaratan lisensi citra berbasis kontainer Windows; Anda tidak dapat mengirim konten Windows untuk redistribusi publik atau pihak ke tiga. Penggunaan dalam lingkungan Anda sendiri diperbolehkan.

Untuk mengaktifkan penggunaan tanda ini untuk instalasi Docker, Anda harus memodifikasi file konfigurasi Docker daemon yang, tergantung pada instalasi Docker Anda, biasanya dapat dikonfigurasi di pengaturan atau menu preferensi pada bagian Mesin Docker atau dengan mengedit file `C:\ProgramData\docker\config\daemon.json` secara langsung.

Berikut ini adalah contoh konfigurasi yang diperlukan. Ganti nilai dengan repositori URI yang Anda gunakan untuk mendorong citra.

```
{
  "allow-nondistributable-artifacts": [
    "111122223333.dkr.ecr.us-west-2.amazonaws.com"
  ]
}
```

Setelah memodifikasi file konfigurasi Docker daemon, Anda harus memulai ulang Docker daemon sebelum mencoba mendorong citra Anda. Konfirmasikan dorongan yang dikerjakan dengan memverifikasi bahwa lapisan dasar didorong ke repositori Anda.

Note

Lapisan dasar untuk citra Windows berukuran besar. Ukuran lapisan akan menyebabkan waktu mendorong menjadi lebih lama dan biaya penyimpanan tambahan di Amazon ECR untuk citra ini akan dibebankan. Dengan demikian, sebaiknya hanya menggunakan opsi ini bila diperlukan jika benar-benar diperlukan untuk mengurangi waktu pembuatan dan biaya penyimpanan yang sedang berlangsung. Misalnya, citra `mcr.microsoft.com/windows/servercore` berukuran sekitar 1,7 GiB ketika dikompresi di Amazon ECR.

Buat VPC endpoint untuk Amazon ECR

Untuk membuat VPC endpoint untuk layanan Amazon ECR, gunakan prosedur [Membuat Antarmuka Titik Akhir](#) dalam Panduan Pengguna Amazon VPC.

Tugas Amazon ECS yang dihosting di instans Amazon EC2 memerlukan titik akhir Amazon ECR dan titik akhir gateway Amazon S3.

Tugas Amazon ECS yang dihosting di Fargate menggunakan 1.4.0 versi platform atau yang lebih baru memerlukan titik akhir Amazon ECR VPC dan titik akhir gateway Amazon S3.

Tugas Amazon ECS yang dihosting di Fargate yang menggunakan 1.3.0 versi platform atau sebelumnya hanya memerlukan `com.amazonaws.region.ecr.dkr` Titik akhir Amazon ECR VPC dan titik akhir gateway Amazon S3.

Note

Urutan titik akhir boleh dibuat di dalam.

com.amazonaws. **wilayah .ecr.dkr**

titik akhir ini digunakan untuk Docker Registry API. Perintah klien Docker seperti push dan pull menggunakan titik akhir ini.

Saat Anda membuat titik akhir ini, Anda harus mengaktifkan nama host DNS pribadi. Untuk melakukan ini, pastikan opsi Aktifkan Nama DNS Pribadi dipilih di konsol VPC Amazon saat Anda membuat titik akhir VPC.

com.amazonaws. **wilayah .ecr.api**

Note

Wilayah yang ditentukan mewakili pengenalan Wilayah untuk AWS Wilayah yang didukung oleh Amazon ECR, seperti us-east-2 untuk Wilayah AS Timur (Ohio).

titik akhir ini digunakan untuk panggilan ke API Amazon ECR. Tindakan API seperti DescribeImages dan CreateRepository masuk ke titik akhir ini.

Ketika titik akhir ini dibuat, Anda memiliki opsi untuk mengaktifkan nama host DNS pribadi. Aktifkan pengaturan ini dengan memilih Aktifkan Nama DNS privat di konsol VPC saat Anda membuat VPC endpoint. Jika Anda mengaktifkan nama host DNS pribadi untuk titik akhir VPC, perbarui SDK Anda atau AWS CLI ke versi terbaru sehingga menentukan URL titik akhir saat menggunakan SDK atau tidak diperlukan. AWS CLI

Jika Anda mengaktifkan nama host DNS pribadi dan menggunakan SDK atau AWS CLI versi yang dirilis sebelum 24 Januari 2019, Anda harus menggunakan --endpoint-url parameter untuk menentukan titik akhir antarmuka. Contoh berikut menunjukkan format untuk titik akhir URL.

```
aws ecr create-repository --repository-name name --endpoint-url https://  
api.ecr.region.amazonaws.com
```

Jika Anda tidak mengaktifkan nama host DNS privat untuk VPC endpoint, Anda harus menggunakan parameter `--endpoint-url` yang menentukan ID VPC endpoint untuk titik akhir antarmuka. Contoh berikut menunjukkan format untuk titik akhir URL.

```
aws ecr create-repository --repository-name name --endpoint-url
https://VPC_endpoint_ID.api.ecr.region.vpce.amazonaws.com
```

Buat titik akhir gateway Amazon S3

Untuk tugas-tugas Amazon ECS Anda yang berfungsi untuk menarik citra privat dari Amazon ECR, Anda harus membuat titik akhir gateway untuk Amazon S3. titik akhir gateway diperlukan karena Amazon ECR menggunakan Amazon S3 untuk menyimpan lapisan citra Anda. Ketika kontainer Anda mengunduh citra dari Amazon ECR, mereka harus mengakses Amazon ECR untuk mendapatkan manifest citra dan kemudian Amazon S3 untuk mengunduh lapisan citra yang sebenarnya. Berikut adalah Amazon Resource Name (ARN) dari bucket Amazon S3 yang berisi lapisan untuk setiap citra Docker.

```
arn:aws:s3:::prod-region-starport-layer-bucket/*
```

Menggunakan prosedur [Membuat titik akhir gateway](#) dalam Panduan Pengguna Amazon VPC untuk membuat titik akhir gateway Amazon S3 berikut untuk Amazon ECR. Saat membuat titik akhir, pastikan untuk memilih tabel rute untuk VPC Anda.

com.amazonaws. *wilayah* .s3

titik akhir gateway Amazon S3 menggunakan dokumen kebijakan IAM untuk membatasi akses ke layanan. Kebijakan Akses penuh dapat digunakan karena pembatasan yang telah Anda masukkan ke dalam tugas IAM role Anda atau kebijakan pengguna IAM lainnya yang masih berlaku di atas kebijakan ini. Jika Anda ingin membatasi akses bucket Amazon S3 ke izin minimum yang diperlukan untuk menggunakan Amazon ECR, lihat [Izin Bucket Amazon S3 Minimum untuk Amazon ECR](#).

Izin Bucket Amazon S3 Minimum untuk Amazon ECR

titik akhir gateway Amazon S3 menggunakan dokumen kebijakan IAM untuk membatasi akses ke layanan. Untuk mengizinkan hanya izin minimum bucket Amazon S3 untuk Amazon ECR, maka batasi akses ke bucket Amazon S3 yang menggunakan Amazon ECR ketika Anda membuat dokumen kebijakan IAM untuk titik akhir.

Tabel berikut menjelaskan izin kebijakan bucket Amazon S3 yang dibutuhkan oleh Amazon ECR.

Izin	Deskripsi
<code>arn:aws:s3:::prod- <i>region</i>-starport-layer-bucket/*</code>	Menyediakan akses ke bucket Amazon S3 yang berisi lapisan untuk setiap citra Docker. Merepresentasikan identifier Wilayah untuk Wilayah AWS yang didukung oleh Amazon ECR, seperti <code>us-east-2</code> untuk (US East (Ohio)).

Contoh

Contoh berikut menggambarkan bagaimana memberikan akses ke bucket Amazon S3 yang diperlukan untuk operasi Amazon ECR.

```
{
  "Statement": [
    {
      "Sid": "Access-to-specific-bucket-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::prod-region-starport-layer-bucket/*"]
    }
  ]
}
```

Buat titik akhir CloudWatch Log

Tugas Amazon ECS menggunakan tipe peluncuran Fargate yang menggunakan VPC tanpa gateway internet yang juga menggunakan **awslogs** driver log untuk mengirim informasi log ke Log mengharuskan Anda membuat CloudWatch `com.amazonaws.region.logs` antarmuka VPC CloudWatch endpoint untuk Log. Untuk informasi selengkapnya, lihat [Menggunakan CloudWatch Log dengan titik akhir VPC antarmuka di Panduan Pengguna Amazon CloudWatch Logs](#).

Buat kebijakan titik akhir untuk VPC endpoint Amazon ECR

Kebijakan VPC endpoint adalah kebijakan sumber daya IAM yang Anda lampirkan ke titik akhir ketika membuat atau memodifikasi titik akhir. Jika Anda tidak melampirkan kebijakan saat membuat titik akhir, AWS lampirkan kebijakan default untuk Anda yang memungkinkan akses penuh ke layanan. Kebijakan titik akhir tidak membatalkan atau mengganti kebijakan pengguna IAM atau kebijakan khusus layanan. Ini adalah kebijakan terpisah untuk mengendalikan akses dari titik akhir ke layanan tertentu. Kebijakan titik akhir harus ditulis dalam format JSON. Untuk informasi selengkapnya, lihat [Pengontrolan Akses ke Layanan dengan VPC endpoint](#) dalam Panduan Pengguna Amazon VPC.

Kami merekomendasikan agar Anda membuat kebijakan sumber daya IAM tunggal dan melampirkannya ke kedua VPC endpoint Amazon ECR.

Berikut adalah contoh kebijakan titik akhir untuk Amazon ECR. Kebijakan ini mengizinkan IAM role tertentu untuk menarik citra dari Amazon ECR.

```
{
  "Statement": [{
    "Sid": "AllowPull",
    "Principal": {
      "AWS": "arn:aws:iam::1234567890:role/role_name"
    },
    "Action": [
      "ecr:BatchGetImage",
      "ecr:GetDownloadUrlForLayer",
      "ecr:GetAuthorizationToken"
    ],
    "Effect": "Allow",
    "Resource": "*"
  }]
}
```

Contoh kebijakan titik akhir berikut mencegah penghapusan repositori tertentu.

```
{
  "Statement": [{
    "Sid": "AllowAll",
    "Principal": "*",
    "Action": "*",
    "Effect": "Allow",
    "Resource": "*"
  }]
}
```

```

},
{
  "Sid": "PreventDelete",
  "Principal": "*",
  "Action": "ecr:DeleteRepository",
  "Effect": "Deny",
  "Resource": "arn:aws:ecr:region:1234567890:repository/repository_name"
}
]
}

```

Contoh kebijakan titik akhir berikut menggabungkan dua contoh sebelumnya ke kebijakan tunggal.

```

{
  "Statement": [{
    "Sid": "AllowAll",
    "Effect": "Allow",
    "Principal": "*",
    "Action": "*",
    "Resource": "*"
  },
  {
    "Sid": "PreventDelete",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "ecr:DeleteRepository",
    "Resource": "arn:aws:ecr:region:1234567890:repository/repository_name"
  },
  {
    "Sid": "AllowPull",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::1234567890:role/role_name"
    },
    "Action": [
      "ecr:BatchGetImage",
      "ecr:GetDownloadUrlForLayer",
      "ecr:GetAuthorizationToken"
    ],
    "Resource": "*"
  }
]
}

```

Untuk mengubah kebijakan VPC endpoint untuk Amazon ECR

1. Buka konsol Amazon VPC di <https://console.aws.amazon.com/vpc/>.
2. Di panel navigasi, pilih titik akhir.
3. Jika Anda belum membuat VPC endpoint untuk Amazon ECR, lihat [Buat VPC endpoint untuk Amazon ECR](#).
4. Pilih VPC endpoint Amazon ECR untuk menambahkan kebijakan, dan pilih tab Kebijakan di bagian bawah layar.
5. Pilih Edit Kebijakan dan buat perubahan pada kebijakan.
6. Pilih Terapkan untuk menyimpan kebijakan tersebut.

Subnet bersama

Anda tidak dapat membuat, mendeskripsikan, memodifikasi, atau menghapus titik akhir VPC di subnet yang dibagikan dengan Anda. Namun, Anda dapat menggunakan titik akhir VPC di subnet yang dibagikan dengan Anda.

Pemantauan Amazon ECR

Anda dapat memantau penggunaan Amazon ECR API dengan Amazon CloudWatch, yang mengumpulkan dan memproses data mentah dari Amazon ECR menjadi metrik yang dapat dibaca, mendekati waktu nyata. Statistik ini dicatat dalam jangka waktu dua minggu, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang penggunaan API Anda. Data metrik Amazon ECR dikirim secara otomatis CloudWatch dalam periode satu menit. Untuk informasi selengkapnya CloudWatch, lihat [Panduan CloudWatch Pengguna Amazon](#).

Amazon ECR menyediakan metrik berdasarkan penggunaan API Anda untuk tindakan otorisasi, dorongan citra, dan tarikan citra.

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan kinerja Amazon ECR dan solusi AWS Anda. Kami menyarankan Anda mengumpulkan data pemantauan dari sumber daya yang membentuk solusi AWS Anda sehingga Anda dapat dengan lebih mudah melakukan debug kegagalan multi-titik jika hal tersebut terjadi. Namun sebelum Anda mulai memantau Amazon ECR, Anda harus membuat rencana pemantauan yang mencakup jawaban atas pertanyaan berikut:

- Apa sasaran pemantauan Anda?
- Sumber daya apa yang akan Anda pantau?
- Seberapa sering Anda akan memantau sumber daya ini?
- Alat pemantauan apa yang akan Anda gunakan?
- Siapa yang akan melakukan tugas pemantauan?
- Siapa yang harus diberi tahu saat terjadi kesalahan?

Langkah berikutnya adalah menetapkan baseline untuk kinerja Amazon ECR normal di lingkungan Anda, dengan mengukur kinerja di berbagai waktu dan dengan kondisi beban yang berbeda. Saat Anda memantau Amazon ECR, simpan data pemantauan historis sehingga Anda dapat membandingkannya dengan data kinerja baru, mengidentifikasi pola kinerja normal dan anomali kinerja, dan merancang metode untuk mengatasi masalah.

Topik

- [Memvisualisasikan kuota layanan Anda dan mengatur alarm](#)
- [Metrik penggunaan Amazon ECR](#)
- [Laporan penggunaan Amazon ECR](#)

- [Metrik repositori Amazon ECR](#)
- [Acara Amazon ECR dan EventBridge](#)
- [Mencatat tindakan Amazon ECR dengan AWS CloudTrail](#)

Memvisualisasikan kuota layanan Anda dan mengatur alarm

Anda dapat menggunakan CloudWatch konsol untuk memvisualisasikan kuota layanan Anda dan melihat bagaimana penggunaan Anda saat ini dibandingkan dengan kuota layanan. Anda juga dapat mengatur alarm agar diberi notifikasi ketika mendekati kuota.

Untuk memvisualisasikan kuota layanan dan secara opsional mengatur alarm

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Di panel navigasi, pilih Metrik.
3. Pada tab Semua metrik, pilih Penggunaan, lalu pilih Oleh Sumber Daya AWS.

Daftar metrik penggunaan kuota layanan muncul.

4. Pilih kotak centang di samping salah satu metrik.

Grafik menampilkan penggunaan Anda saat ini atas sumber daya AWS tersebut.

5. Untuk menambahkan kuota layanan Anda ke grafik, lakukan hal berikut:
 - a. Pilih tab Metrik bergrafik.
 - b. Pilih Pernyataan matematika, Mulai dengan pernyataan kosong. Lalu, di baris baru, di bawah Rincian, masukkan **SERVICE_QUOTA(m1)**.

Baris baru ditambahkan ke grafik, menampilkan kuota layanan untuk sumber daya yang diwakili dalam metrik.
6. Untuk melihat penggunaan Anda saat ini sebagai persentase dari kuota, tambahkan pernyataan baru atau ubah pernyataan SERVICE_QUOTA saat ini. Untuk pernyataan baru, gunakan **m1/60/SERVICE_QUOTA(m1)*100**.
7. (Opsional) Untuk mengatur alarm yang memberi tahu Anda jika mendekati kuota layanan, lakukan hal berikut:
 - a. Di **m1/60/SERVICE_QUOTA(m1)*100** baris, di bawah Tindakan, pilih ikon alarm. Terlihat seperti lonceng.

Halaman pembuatan alarm muncul.

- b. Di bawah Ketentuan, pastikan bahwa Jenis ambang batas bersifat Statis dan Setiap kali Pernyataan¹ diatur menjadi Lebih Besar. Di bawah dari, masukkan **80**. Tindakan ini akan membuat alarm yang masuk ke status ALARM ketika penggunaan Anda melebihi 80 persen dari kuota.
- c. Pilih Selanjutnya.
- d. Di halaman berikutnya, pilih topik Amazon SNS atau buat topik baru. Topik ini akan diberi tahu ketika alarm masuk ke status ALARM. Lalu, pilih Selanjutnya.
- e. Di halaman berikutnya, masukkan nama dan penjelasan untuk alarm, lalu pilih Selanjutnya.
- f. Pilih Buat alarm.

Metrik penggunaan Amazon ECR

Anda dapat menggunakan metrik CloudWatch penggunaan untuk memberikan visibilitas ke dalam penggunaan sumber daya akun Anda. Gunakan metrik ini untuk memvisualisasikan penggunaan layanan Anda saat ini pada CloudWatch grafik dan dasbor.

Metrik penggunaan Amazon ECR sesuai dengan kuota layanan AWS. Anda dapat mengonfigurasi alarm yang memperingatkan ketika penggunaan Anda mendekati kuota layanan. Untuk informasi lebih lanjut tentang kuota layanan Amazon ECR, lihat [Kuota layanan Amazon ECR](#).

Amazon ECR menerbitkan metrik berikut di namespace AWS/Usage.

Metrik	Deskripsi
CallCount	<p>Jumlah tindakan panggilan API dari akun Anda. Sumber daya ditentukan oleh dimensi yang terkait dengan metrik.</p> <p>Statistik yang paling berguna untuk metrik ini adalah SUM, yang mewakili jumlah nilai dari semua kontributor selama periode yang ditentukan.</p>

Dimensi berikut digunakan untuk menyempurnakan metrik penggunaan yang diterbitkan oleh Amazon ECR.

Dimensi	Deskripsi
Service	Nama layanan AWS yang memuat sumber daya. Untuk metrik penggunaan Amazon ECR, nilai untuk dimensi ini adalah ECR.
Type	Jenis entitas yang dilaporkan. Saat ini, satu-satunya nilai yang valid untuk metrik penggunaan Amazon ECR adalah API.
Resource	Jenis sumber daya yang sedang berjalan. Saat ini, Amazon ECR mengembalikan informasi tentang penggunaan API Anda untuk tindakan API berikut. <ul style="list-style-type: none">• GetAuthorizationToken• BatchCheckLayerAvailability• InitiateLayerUpload• UploadLayerPart• CompleteLayerUpload• PutImage• BatchGetImage• GetDownloadUrlForLayer
Class	Kelas sumber daya yang ditelusuri. Saat ini, Amazon ECR tidak menggunakan dimensi kelas.

Laporan penggunaan Amazon ECR

AWS menyediakan alat pelaporan gratis bernama Cost Explorer yang memungkinkan Anda menganalisis biaya dan penggunaan sumber daya Amazon ECR Anda.

Gunakan Cost Explorer untuk melihat bagan penggunaan dan biaya Anda. Anda dapat melihat data hingga 13 bulan terakhir, dan memperkirakan seberapa besar kemungkinan pengeluaran Anda untuk tiga bulan ke depan. Anda dapat menggunakan Cost Explorer untuk melihat pola pengeluaran sumber daya AWS Anda dari waktu ke waktu, mengidentifikasi area-area yang memerlukan penyelidikan lebih lanjut, dan melihat tren yang dapat Anda gunakan untuk memahami biaya Anda. Anda juga dapat menentukan rentang waktu untuk data, dan melihat data waktu berdasarkan hari atau bulan.

Data pengukuran dalam Laporan Biaya dan Penggunaan menunjukkan penggunaan di semua repositori Amazon ECR Anda. Untuk informasi selengkapnya, lihat [Penandaan sumber daya Anda untuk penagihan](#).

Untuk informasi lebih lanjut tentang membuat AWS Laporan Biaya dan Penggunaan, lihat [Laporan Biaya dan Penggunaan AWS](#) di AWS Billing Panduan Pengguna.

Metrik repositori Amazon ECR

Amazon ECR mengirimkan metrik jumlah tarik repositori ke Amazon. CloudWatch Data metrik Amazon ECR secara otomatis dikirim ke CloudWatch dalam periode 1 menit. Untuk informasi selengkapnya CloudWatch, lihat [Panduan CloudWatch Pengguna Amazon](#).

Topik

- [Mengaktifkan metrik CloudWatch](#)
- [Metrik dan dimensi yang tersedia](#)
- [Melihat metrik Amazon ECR](#)

Mengaktifkan metrik CloudWatch

Amazon ECR mengirimkan metrik repositori secara otomatis untuk semua repositori. Tidak perlu mengambil langkah manual apa pun.

Metrik dan dimensi yang tersedia

Bagian berikut mencantumkan metrik dan dimensi yang dikirimkan Amazon ECR ke Amazon. CloudWatch

Metrik Amazon ECR

Amazon ECR menyediakan metrik bagi Anda untuk memantau repositori Anda. Anda dapat mengukur jumlah tarik.

Namespace AWS/ECR mencakup metrik berikut.

RepositoryPullCount

Jumlah total tarikan untuk gambar di repositori.

Dimensi yang valid: RepositoryName.

Statistik yang valid: Rata-rata, Minimum, Maksimum, Jumlah, Jumlah Sampel. Statistik yang paling berguna adalah Sum.

Satuan: Bilangan bulat.

Dimensi untuk metrik Amazon ECR

Metrik Amazon ECR menggunakan AWS/ECR namespace dan menyediakan metrik untuk dimensi berikut.

RepositoryName

Dimensi ini menyaring data yang Anda minta untuk semua gambar kontainer dalam repositori tertentu.

Melihat metrik Amazon ECR

Anda dapat melihat metrik repositori Amazon ECR di konsol. CloudWatch CloudWatch Konsol menyediakan tampilan sumber daya Anda yang berbutir halus dan dapat disesuaikan.

Melihat metrik Amazon ECR menggunakan konsol CloudWatch

Metrik repositori Amazon ECR dapat dilihat di konsol. CloudWatch Konsol menyediakan tampilan metrik Amazon ECR yang paling detail, dan Anda dapat menyesuaikan tampilan sesuai dengan kebutuhan Anda. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).

Untuk melihat metrik di konsol CloudWatch

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Di navigasi kiri, pilih Metrik, Semua metrik.
3. Pada tab Browse, di bawah AWSNamespaces, pilih ECR.
4. Pilih metrik untuk dilihat. Metrik repositori dicakup sebagai ECR > Metrik Repositori.

Acara Amazon ECR dan EventBridge

Amazon EventBridge memungkinkan Anda untuk mengotomatiskan AWS layanan Anda dan merespons secara otomatis peristiwa sistem seperti masalah ketersediaan aplikasi atau perubahan

sumber daya. Acara dari AWS layanan dikirimkan ke EventBridge dalam waktu dekat. Anda dapat menulis aturan sederhana untuk menunjukkan kejadian mana yang sesuai kepentingan Anda, dan memasukkan tindakan otomatis apa yang diambil ketika suatu kejadian sesuai dengan suatu aturan. Tindakan yang dapat dipicu secara otomatis meliputi hal berikut:

- Menambahkan peristiwa ke grup log di CloudWatch Log
- Mengambil fungsi AWS Lambda
- Meminta Perintah Amazon EC2 Run
- Mengirim peristiwa ke Amazon Kinesis Data Streams
- Mengaktifkan mesin keadaan AWS Step Functions
- Memberi tahu topik Amazon SNS atau antrian Amazon SQS

Untuk informasi selengkapnya, lihat [Memulai Amazon EventBridge](#) di Panduan EventBridge Pengguna Amazon.

Contoh kejadian dari Amazon ECR

Berikut adalah contoh kejadian dari Amazon ECR. Kejadian dipancarkan atas dasar upaya terbaik.

Acara untuk push gambar yang lengkap

Kejadian berikut dikirim ketika setiap dorongan citra selesai. Untuk informasi selengkapnya, lihat [Mendorong gambar Docker](#).

```
{
  "version": "0",
  "id": "13cde686-328b-6117-af20-0e5566167482",
  "detail-type": "ECR Image Action",
  "source": "aws.ecr",
  "account": "123456789012",
  "time": "2019-11-16T01:54:34Z",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "result": "SUCCESS",
    "repository-name": "my-repository-name",
    "image-digest":
      "sha256:7f5b2640fe6fb4f46592dfd3410c4a79dac4f89e4782432e0378abcd1234",
  }
}
```

```
    "action-type": "PUSH",
    "image-tag": "latest"
  }
}
```

Acara untuk aksi pull through cache

Peristiwa berikut dikirim ketika tindakan pull through cache dicoba. Untuk informasi selengkapnya, lihat [Menggunakan aturan pull through cache](#).

```
{
  "version": "0",
  "id": "85fc3613-e913-7fc4-a80c-a3753e4aa9ae",
  "detail-type": "ECR Pull Through Cache Action",
  "source": "aws.ecr",
  "account": "123456789012",
  "time": "2023-02-29T02:36:48Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:ecr:us-west-2:123456789012:repository/docker-hub/alpine"
  ],
  "detail": {
    "rule-version": "1",
    "sync-status": "SUCCESS",
    "ecr-repository-prefix": "docker-hub",
    "repository-name": "docker-hub/alpine",
    "upstream-registry-url": "public.ecr.aws",
    "image-tag": "3.17.2",
    "image-digest":
      "sha256:4aa08ef415aecc80814cb42fa41b658480779d80c77ab15EXAMPLE",
  }
}
```

Acara untuk pemindaian gambar selesai (pemindaian dasar)

Ketika pemindaian dasar diaktifkan untuk registri Anda, peristiwa berikut dikirim ketika setiap pemindaian gambar selesai. Parameter `finding-severity-counts` hanya akan mengembalikan nilai untuk suatu tingkat keparahan jika ada. Contohnya, jika citra tidak mengandung temuan di tingkat CRITICAL, maka tidak ada hitungan kritis yang dikembalikan. Untuk informasi selengkapnya, lihat [Pemindaian dasar](#).

Note

Untuk detail tentang peristiwa yang dipancarkan Amazon Inspector saat pemindaian yang disempurnakan diaktifkan, lihat [EventBridge acara](#)

```
{
  "version": "0",
  "id": "85fc3613-e913-7fc4-a80c-a3753e4aa9ae",
  "detail-type": "ECR Image Scan",
  "source": "aws.ecr",
  "account": "123456789012",
  "time": "2019-10-29T02:36:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ecr:us-east-1:123456789012:repository/my-repository-name"
  ],
  "detail": {
    "scan-status": "COMPLETE",
    "repository-name": "my-repository-name",
    "finding-severity-counts": {
      "CRITICAL": 10,
      "MEDIUM": 9
    },
    "image-digest":
      "sha256:7f5b2640fe6fb4f46592dfd3410c4a79dac4f89e4782432e0378abcd1234",
    "image-tags": []
  }
}
```

Acara untuk pemberitahuan perubahan pada sumber daya dengan pemindaian yang ditingkatkan diaktifkan (pemindaian yang disempurnakan)

Ketika pemindaian yang disempurnakan diaktifkan untuk registri Anda, peristiwa berikut dikirim oleh Amazon ECR ketika ada perubahan dengan sumber daya yang telah ditingkatkan pemindaian diaktifkan. Ini termasuk repositori baru yang sedang dibuat, frekuensi pemindaian untuk repositori yang diubah, atau ketika gambar dibuat atau dihapus di repositori dengan pemindaian yang ditingkatkan diaktifkan. Untuk informasi selengkapnya, lihat [Pemindaian citra](#).

```
{
  "version": "0",
```

```

{id": "0c18352a-a4d4-6853-ef53-0ab8638973bf",
"detail-type": "ECR Scan Resource Change",
"source": "aws.ecr",
"account": "123456789012",
"time": "2021-10-14T20:53:46Z",
"region": "us-east-1",
"resources": [],
"detail": {
  "action-type": "SCAN_FREQUENCY_CHANGE",
  "repositories": [{
    "repository-name": "repository-1",
    "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-1",
    "scan-frequency": "SCAN_ON_PUSH",
    "previous-scan-frequency": "MANUAL"
  },
  {
    "repository-name": "repository-2",
    "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-2",
    "scan-frequency": "CONTINUOUS_SCAN",
    "previous-scan-frequency": "SCAN_ON_PUSH"
  },
  {
    "repository-name": "repository-3",
    "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-3",
    "scan-frequency": "CONTINUOUS_SCAN",
    "previous-scan-frequency": "SCAN_ON_PUSH"
  }
  ],
  "resource-type": "REPOSITORY",
  "scan-type": "ENHANCED"
}
}

```

Acara untuk penghapusan gambar

Kejadian berikut dikirim ketika suatu citra dihapus. Untuk informasi selengkapnya, lihat [Menghapus citra](#).

```

{
  "version": "0",
  "id": "dd3b46cb-2c74-f49e-393b-28286b67279d",
  "detail-type": "ECR Image Action",
  "source": "aws.ecr",

```



```
"account": "123456789012",
"time": "2019-11-16T02:01:05Z",
"region": "us-west-2",
"resources": [],
"detail": {
  "result": "SUCCESS",
  "repository-name": "my-repository-name",
  "image-digest":
  "sha256:7f5b2640fe6fb4f46592dfd3410c4a79dac4f89e4782432e0378abcd1234",
  "action-type": "DELETE",
  "image-tag": "latest"
}
```

Mencatat tindakan Amazon ECR dengan AWS CloudTrail

Amazon ECR terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amazon ECR. CloudTrail menangkap tindakan Amazon ECR berikut sebagai peristiwa:

- Semua panggilan API, termasuk panggilan dari konsol Amazon ECR
- Semua tindakan yang diambil karena pengaturan enkripsi pada repositori Anda
- Semua tindakan yang diambil karena aturan kebijakan siklus hidup, termasuk tindakan yang berhasil dan tidak berhasil

Important

Karena keterbatasan ukuran CloudTrail peristiwa individual, untuk tindakan kebijakan siklus hidup di mana 10 gambar atau lebih kedaluwarsa Amazon ECR mengirimkan beberapa peristiwa ke CloudTrail. Selain itu, Amazon ECR menyertakan maksimum 100 tag per gambar.

Saat jejak dibuat, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk acara untuk Amazon ECR. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi ini, Anda dapat menentukan permintaan yang dibuat ke Amazon ECR, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail lainnya.

Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS CloudTrail](#).

Informasi Amazon ECR di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di Amazon ECR, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS Anda. Untuk informasi selengkapnya, lihat [Melihat Acara dengan Riwayat CloudTrail Acara](#).

Untuk catatan kejadian yang sedang berlangsung di akun AWS Anda, termasuk kejadian untuk Amazon ECR, buatlah jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Bila Anda membuat jejak di konsol, Anda dapat menerapkan jejak ke satu Wilayah atau semua Wilayah. Jejak tersebut mencatat kejadian dari semua Wilayah di partisi AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat:

- [Membuat jejak untuk AWS akun Anda](#)
- [AWS integrasi layanan dengan log CloudTrail](#)
- [Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima file CloudTrail log dari beberapa Wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#)

Semua tindakan Amazon ECR API dicatat oleh CloudTrail dan didokumentasikan dalam [Referensi API Amazon Elastic Container Registry](#). Saat Anda melakukan tugas umum, bagian dibuat dalam file CloudTrail log untuk setiap tindakan API yang merupakan bagian dari tugas tersebut. Misalnya, ketika Anda membuat repositori, `GetAuthorizationToken`, `CreateRepository` dan `SetRepositoryPolicy` bagian yang dihasilkan dalam file CloudTrail log. Saat Anda mendorong citra ke repositori, bagian `InitiateLayerUpload`, `UploadLayerPart`, `CompleteLayerUpload`, dan `PutImage` dihasilkan. Ketika Anda menarik suatu citra, bagian `GetDownloadUrlForLayer` dan `BatchGetImage` dihasilkan. Untuk contoh tugas umum ini, lihat [CloudTrail contoh entri log](#).

Setiap entri kejadian atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan dibuat dengan root atau kredensial pengguna

- Jika permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan
- Jika permintaan tersebut dibuat oleh layanan AWS lainnya

Untuk informasi selengkapnya, lihat [CloudTrail userIdentity Elemen](#).

Memahami entri file log Amazon ECR

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber mana pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan informasi lainnya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga file tersebut tidak muncul dalam urutan tertentu.

CloudTrail contoh entri log

Berikut ini adalah contoh entri CloudTrail log untuk beberapa tugas Amazon ECR umum.

Note

Contoh-contoh ini telah diformat untuk meningkatkan keterbacaan. Dalam file CloudTrail log, semua entri dan peristiwa digabungkan menjadi satu baris. Selain itu, contoh ini telah terbatas pada entri tunggal Amazon ECR. Dalam suatu berkas log CloudTrail nyata, Anda melihat entri dan peristiwa dari beberapa layanan AWS.

Topik

- [Contoh: Buat tindakan repositori](#)
- [Contoh: Tindakan AWS KMS CreateGrant API saat membuat repositori Amazon ECR](#)
- [Contoh: Tindakan dorongan citra](#)
- [Contoh: Tindakan tarikan citra](#)
- [Contoh: Tindakan kebijakan siklus hidup citra](#)

Contoh: Buat tindakan repositori

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateRepository tindakan.

```
{
  "eventVersion": "1.04",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:account_name",
    "arn": "arn:aws:sts::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-07-11T21:54:07Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
      }
    }
  },
  "eventTime": "2018-07-11T22:17:43Z",
  "eventSource": "ecr.amazonaws.com",
  "eventName": "CreateRepository",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "203.0.113.12",
  "userAgent": "console.amazonaws.com",
  "requestParameters": {
    "repositoryName": "testrepo"
  },
  "responseElements": {
    "repository": {
      "repositoryArn": "arn:aws:ecr:us-east-2:123456789012:repository/testrepo",
      "repositoryName": "testrepo",
      "repositoryUri": "123456789012.dkr.ecr.us-east-2.amazonaws.com/testrepo",
      "createdAt": "Jul 11, 2018 10:17:44 PM",
      "registryId": "123456789012"
    }
  },
  "requestID": "cb8c167e-EXAMPLE",
  "eventID": "e3c6f4ce-EXAMPLE",
  "resources": [
```

```

    {
      "ARN": "arn:aws:ecr:us-east-2:123456789012:repository/testrepo",
      "accountId": "123456789012"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}

```

Contoh: Tindakan AWS KMS CreateGrant API saat membuat repositori Amazon ECR

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan AWS KMS CreateGrant tindakan saat membuat repositori Amazon ECR dengan enkripsi KMS diaktifkan. Untuk setiap repositori yang dibuat dengan enkripsi KMS diaktifkan, Anda akan melihat dua entri CreateGrant log di CloudTrail

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAIEP6W46J43IG7LXAQ",
    "arn": "arn:aws:iam::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major",
    "sessionContext": {
      "sessionIssuer": {
        },
      "webIdFederationData": {
        },
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2020-06-10T19:22:10Z"
      }
    }
  },
  "invokedBy": "AWS Internal"
},
"eventTime": "2020-06-10T19:22:10Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-west-2",
"sourceIPAddress": "203.0.113.12",

```

```
"userAgent": "console.amazonaws.com",
"requestParameters": {
  "keyId": "4b55e5bf-39c8-41ad-b589-18464af7758a",
  "granteePrincipal": "ecr.us-west-2.amazonaws.com",
  "operations": [
    "GenerateDataKey",
    "Decrypt"
  ],
  "retiringPrincipal": "ecr.us-west-2.amazonaws.com",
  "constraints": {
    "encryptionContextSubset": {
      "aws:ecr:arn": "arn:aws:ecr:us-west-2:123456789012:repository/testrepo"
    }
  }
},
"responseElements": {
  "grantId": "3636af9adfee1accb67b83941087dcd45e7fadc4e74ff0103bb338422b5055f3"
},
"requestID": "047b7dea-b56b-4013-87e9-a089f0f6602b",
"eventID": "af4c9573-c56a-4886-baca-a77526544469",
"readOnly": false,
"resources": [
  {
    "accountId": "123456789012",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-west-2:123456789012:key/4b55e5bf-39c8-41ad-
b589-18464af7758a"
  }
],
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

Contoh: Tindakan dorongan citra

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan dorongan gambar yang menggunakan PutImage tindakan.

Note

Saat mendorong gambar, Anda juga akan melihat `InitiateLayerUpload`, `UploadLayerPart`, dan `CompleteLayerUpload` referensi di CloudTrail log.

```
{
  "eventVersion": "1.04",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:account_name",
    "arn": "arn:aws:sts::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-04-15T16:42:14Z"
      }
    }
  },
  "eventTime": "2019-04-15T16:45:00Z",
  "eventSource": "ecr.amazonaws.com",
  "eventName": "PutImage",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "203.0.113.12",
  "userAgent": "console.amazonaws.com",
  "requestParameters": {
    "repositoryName": "testrepo",
    "imageTag": "latest",
    "registryId": "123456789012",
    "imageManifest": "{\n  \"schemaVersion\": 2,\n  \"mediaType\": \"application/\n  vnd.docker.distribution.manifest.v2+json\",\n  \"config\": {\n    \"mediaType\":\n    \"application/vnd.docker.container.image.v1+json\",\n    \"size\": 5543,\n    \"digest\": \"sha256:000b9b805af1cdb60628898c9f411996301a1c13afd3dbef1d8a16ac6dbf503a\n  \"\n  },\n  \"layers\": [\n    {\n      \"mediaType\": \"application/\n  vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 43252507,\n      \"digest\": \"sha256:3b37166ec61459e76e33282dda08f2a9cd698ca7e3d6bc44e6a6e7580cdeff8e\n  \"\n    },\n    {\n      \"mediaType\": \"application/\n  vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 846,\n      \"digest
```

```

\": \"sha256:504facff238fde83f1ca8f9f54520b4219c5b8f80be9616ddc52d31448a044bd
\n      },\n      {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 615, \n        \"digest
\": \"sha256:ebbcacd28e101968415b0c812b2d2dc60f969e36b0b08c073bf796e12b1bb449\" \n
      },\n      {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 850, \n        \"digest
\": \"sha256:c7fb3351ecad291a88b92b600037e2435c84a347683d540042086fe72c902b8a
\n      },\n      {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 168, \n        \"digest\":
 \"sha256:2e3debadcbf7e542e2aefbce1b64a358b1931fb403b3e4aeca27cb4d809d56c2\" \n      },
\n      {\n        \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip
\", \n        \"size\": 37720774, \n        \"digest\":
 \"sha256:f8c9f51ad524d8ae9bf4db69cd3e720ba92373ec265f5c390ffb21bb0c277941\" \n
      },\n      {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 30432107, \n
        \"digest\": \"sha256:813a50b13f61cf1f8d25f19fa96ad3aa5b552896c83e86ce413b48b091d7f01b
\n      },\n      {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 197, \n        \"digest
\": \"sha256:7ab043301a6187ea3293d80b30ba06c7bf1a0c3cd4c43d10353b31bc0cecf7d
\n      },\n      {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 154, \n        \"digest
\": \"sha256:67012cca8f31dc3b8ee2305e7762fee20c250513effdedb38a1c37784a5a2e71\" \n
      },\n      {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 176, \n        \"digest
\": \"sha256:3bc892145603fffc9b1c97c94e2985b4cb19ca508750b15845a5d97becbd1a0e
\n      },\n      {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 183, \n        \"digest
\": \"sha256:6f1c79518f18251d35977e7e46bfa6c6b9cf50df2a79d4194941d95c54258d18\" \n
      },\n      {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 212, \n        \"digest
\": \"sha256:b7bcfbc2e2888afebede4dd1cd5eebf029bb6315feeaf0b56e425e11a50afe42\" \n
      },\n      {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 212, \n        \"digest\":
 \"sha256:2b220f8b0f32b7c2ed8eaafe1c802633bbd94849b9ab73926f0ba46cdae91629\" \n      } \n
    ] \n  } \n} \n
\"responseElements\": {
  \"image\": {
    \"repositoryName\": \"testrepo\",
    \"imageManifest\": \"{ \n    \"schemaVersion\": 2, \n    \"mediaType\": \"application/
vnd.docker.distribution.manifest.v2+json\", \n    \"config\": { \n    \"mediaType\":
 \"application/vnd.docker.container.image.v1+json\", \n    \"size\": 5543, \n
    \"digest\": \"sha256:000b9b805af1cdb60628898c9f411996301a1c13afd3dbef1d8a16ac6dbf503a
\n    }, \n    \"layers\": [ \n    { \n    \"mediaType\": \"application/

```



```

vnd.docker.image.rootfs.diff.tar.gzip",\n          \size": 43252507,\n
 \digest": \sha256:3b37166ec61459e76e33282dda08f2a9cd698ca7e3d6bc44e6a6e7580cdeff8e
 \n          },\n          {\n          \mediaType": \application/
vnd.docker.image.rootfs.diff.tar.gzip",\n          \size": 846,\n          \digest
 \": \sha256:504facff238fde83f1ca8f9f54520b4219c5b8f80be9616ddc52d31448a044bd
 \n          },\n          {\n          \mediaType": \application/
vnd.docker.image.rootfs.diff.tar.gzip",\n          \size": 615,\n          \digest
 \": \sha256:ebbcacd28e101968415b0c812b2d2dc60f969e36b0b08c073bf796e12b1bb449"\n
          },\n          {\n          \mediaType": \application/
vnd.docker.image.rootfs.diff.tar.gzip",\n          \size": 850,\n          \digest
 \": \sha256:c7fb3351ecad291a88b92b600037e2435c84a347683d540042086fe72c902b8a
 \n          },\n          {\n          \mediaType": \application/
vnd.docker.image.rootfs.diff.tar.gzip",\n          \size": 168,\n          \digest":
 \sha256:2e3debadcbf7e542e2aefbce1b64a358b1931fb403b3e4aeca27cb4d809d56c2"\n          },
\n          {\n          \mediaType": \application/vnd.docker.image.rootfs.diff.tar.gzip
 \",\n          \size": 37720774,\n          \digest":
 \sha256:f8c9f51ad524d8ae9bf4db69cd3e720ba92373ec265f5c390ffb21bb0c277941"\n
          },\n          {\n          \mediaType": \application/
vnd.docker.image.rootfs.diff.tar.gzip",\n          \size": 30432107,\n
 \digest": \sha256:813a50b13f61cf1f8d25f19fa96ad3aa5b552896c83e86ce413b48b091d7f01b
 \n          },\n          {\n          \mediaType": \application/
vnd.docker.image.rootfs.diff.tar.gzip",\n          \size": 197,\n          \digest
 \": \sha256:7ab043301a6187ea3293d80b30ba06c7bf1a0c3cd4c43d10353b31bc0cecf7d
 \n          },\n          {\n          \mediaType": \application/
vnd.docker.image.rootfs.diff.tar.gzip",\n          \size": 154,\n          \digest
 \": \sha256:67012cca8f31dc3b8ee2305e7762fee20c250513effdedb38a1c37784a5a2e71"\n
          },\n          {\n          \mediaType": \application/
vnd.docker.image.rootfs.diff.tar.gzip",\n          \size": 176,\n          \digest
 \": \sha256:3bc892145603fffc9b1c97c94e2985b4cb19ca508750b15845a5d97becbd1a0e
 \n          },\n          {\n          \mediaType": \application/
vnd.docker.image.rootfs.diff.tar.gzip",\n          \size": 183,\n          \digest
 \": \sha256:6f1c79518f18251d35977e7e46bfa6c6b9cf50df2a79d4194941d95c54258d18"\n
          },\n          {\n          \mediaType": \application/
vnd.docker.image.rootfs.diff.tar.gzip",\n          \size": 212,\n          \digest
 \": \sha256:b7bcfbc2e2888afebede4dd1cd5eebf029bb6315feeaf0b56e425e11a50afe42"\n
          },\n          {\n          \mediaType": \application/
vnd.docker.image.rootfs.diff.tar.gzip",\n          \size": 212,\n          \digest":
 \sha256:2b220f8b0f32b7c2ed8eaafe1c802633bbd94849b9ab73926f0ba46cdae91629"\n          }
\n          ]\n          },
  "registryId": "123456789012",
  "imageId": {
    "imageDigest":
      "sha256:98c8b060c21d9adbb6b8c41b916e95e6307102786973ab93a41e8b86d1fc6d3e",
    "imageTag": "latest"
  }
}

```

```

    }
  }
},
"requestID": "cf044b7d-5f9d-11e9-9b2a-95983139cc57",
"eventID": "2bfd4ee2-2178-4a82-a27d-b12939923f0f",
"resources": [{
  "ARN": "arn:aws:ecr:us-east-2:123456789012:repository/testrepo",
  "accountId": "123456789012"
}],
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}

```

Contoh: Tindakan tarikan citra

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan tarikan gambar yang menggunakan BatchGetImage tindakan.

Note

Saat menarik gambar, jika Anda belum memiliki gambar secara lokal, Anda juga akan melihat `GetDownloadUrlForLayer` referensi di CloudTrail log.

```

{
  "eventVersion": "1.04",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:account_name",
    "arn": "arn:aws:sts::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-04-15T16:42:14Z"
      }
    }
  },
  "eventTime": "2019-04-15T17:23:20Z",
  "eventSource": "ecr.amazonaws.com",

```

```
"eventName": "BatchGetImage",
"awsRegion": "us-east-2",
"sourceIPAddress": "203.0.113.12",
"userAgent": "console.amazonaws.com",
"requestParameters": {
  "imageIds": [{
    "imageTag": "latest"
  }],
  "acceptedMediaTypes": [
    "application/json",
    "application/vnd.oci.image.manifest.v1+json",
    "application/vnd.oci.image.index.v1+json",
    "application/vnd.docker.distribution.manifest.v2+json",
    "application/vnd.docker.distribution.manifest.list.v2+json",
    "application/vnd.docker.distribution.manifest.v1+prettyjws"
  ],
  "repositoryName": "testrepo",
  "registryId": "123456789012"
},
"responseElements": null,
"requestID": "2a1b97ee-5fa3-11e9-a8cd-cd2391aeda93",
"eventID": "c84f5880-c2f9-4585-9757-28fa5c1065df",
"resources": [{
  "ARN": "arn:aws:ecr:us-east-2:123456789012:repository/testrepo",
  "accountId": "123456789012"
}],
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

Contoh: Tindakan kebijakan siklus hidup citra

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan kapan gambar kedaluwarsa karena aturan kebijakan siklus hidup. Jenis kejadian ini dapat ditemukan dengan mem-filter `PolicyExecutionEvent` untuk bidang nama kejadian.

Important

Karena keterbatasan ukuran CloudTrail peristiwa individual, untuk tindakan kebijakan siklus hidup di mana 10 gambar atau lebih kedaluwarsa Amazon ECR mengirimkan beberapa peristiwa ke CloudTrail. Selain itu, Amazon ECR menyertakan maksimum 100 tag per gambar.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2020-03-12T20:22:12Z",
  "eventSource": "ecr.amazonaws.com",
  "eventName": "PolicyExecutionEvent",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "9354dd7f-9aac-4e9d-956d-12561a4923aa",
  "readOnly": true,
  "resources": [
    {
      "ARN": "arn:aws:ecr:us-west-2:123456789012:repository/testrepo",
      "accountId": "123456789012",
      "type": "AWS::ECR::Repository"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
    "repositoryName": "testrepo",
    "lifecycleEventPolicy": {
      "lifecycleEventRules": [
        {
          "rulePriority": 1,
          "description": "remove all images > 2",
          "lifecycleEventSelection": {
            "tagStatus": "Any",
            "tagPrefixList": [],
            "countType": "Image count more than",
            "countNumber": 2
          },
          "action": "expire"
        }
      ]
    },
    "lastEvaluatedAt": 0,
    "policyVersion": 1,
  }
}
```

```
    "policyId": "ceb86829-58e7-9498-920c-aa042e33037b"
  },
  "lifecycleEventImageActions": [
    {
      "lifecycleEventImage": {
        "digest":
"sha256:ddba4d27a7ffc3f86dd6c2f92041af252a1f23a8e742c90e6e1297bfa1bc0c45",
        "tagStatus": "Tagged",
        "tagList": [
          "alpine"
        ],
        "pushedAt": 1584042813000
      },
      "rulePriority": 1
    },
    {
      "lifecycleEventImage": {
        "digest":
"sha256:6ab380c5a5acf71c1b6660d645d2cd79cc8ce91b38e0352cbf9561e050427baf",
        "tagStatus": "Tagged",
        "tagList": [
          "centos"
        ],
        "pushedAt": 1584042842000
      },
      "rulePriority": 1
    }
  ]
}
```

Kuota layanan Amazon ECR

Tabel berikut menyediakan kuota layanan default untuk Amazon Elastic Container Registry (Amazon ECR).

Nama	Default	Dapat disetel	Deskripsi
Filter per aturan dalam konfigurasi replikasi	Setiap Wilayah yang didukung: 100	Tidak	Jumlah maksimum filter per aturan dalam konfigurasi replikasi.
Gambar per repositori	Setiap Wilayah yang didukung: 10.000	Ya	Jumlah citra maksimum per repositori.
Bagian lapisan	Setiap Wilayah yang didukung: 4.200	Tidak	Jumlah maksimum bagian lapisan. Ini hanya berlaku jika Anda menggunakan tindakan API Amazon ECR secara langsung untuk memulai unggahan multipart untuk operasi dorongan citra.
Panjang kebijakan siklus hidup	Setiap Wilayah yang didukung: 30.720	Tidak	Jumlah maksimum karakter dalam kebijakan siklus hidup.
Ukuran bagian lapisan maksimum	Setiap Wilayah yang didukung: 10	Tidak	Ukuran maksimum (MiB) dari sebuah bagian lapisan. Ini hanya berlaku jika Anda menggunakan tindakan API Amazon ECR secara langsung untuk memulai unggahan

Nama	Default	Dapat disesu an	Deskripsi
			multipart untuk operasi dorongan citra.
Ukuran lapisan maksimum	Setiap Wilayah yang didukung: 52.000	Tidak	Ukuran maksimum (MiB) lapisan.
Ukuran bagian lapisan minimum	Setiap Wilayah yang didukung: 5	Tidak	Ukuran minimum (MiB) dari bagian lapisan. Ini hanya berlaku jika Anda menggunakan tindakan API Amazon ECR secara langsung untuk memulai unggahan multipart untuk operasi dorongan citra.
Tarik aturan cache per registri	Setiap Wilayah yang didukung: 50	Tidak	Jumlah maksimum aturan cache pull-through.
Tingkat BatchCheckLayerAvailability permintaan	Setiap Wilayah yang didukung: 1.000 per detik	Ya	Jumlah maksimum BatchCheckLayerAvailability permintaan yang dapat Anda buat per detik di Wilayah saat ini. Ketika citra didorong ke repositori, setiap lapisan citra diperiksa untuk memverifikasi apakah itu telah diunggah sebelumnya. Jika sudah diunggah, maka lapisan citra dilewati.

Nama	Default	Dapat disesu an	Deskripsi
Tingkat BatchGetImage permintaan	Setiap Wilayah yang didukung: 2.000 per detik	Ya	Jumlah maksimum BatchGetImage permintaan yang dapat Anda buat per detik di Wilayah saat ini. Saat gambar ditarik, BatchGetImage API dipanggil sekali untuk mengambil manifes gambar. Jika Anda meminta peningkatan kuota untuk API ini, tinjau juga GetDownloadUrlForLayer penggunaan Anda.
Tingkat CompleteLayerUpload permintaan	Setiap Wilayah yang didukung: 100 per detik	Ya	Jumlah maksimum CompleteLayerUpload permintaan yang dapat Anda buat per detik di Wilayah saat ini. Saat gambar didorong, CompleteLayerUpload API dipanggil sekali per setiap layer gambar baru untuk memverifikasi bahwa unggahan telah selesai.
Tingkat GetAuthorizationToken permintaan	Setiap Wilayah yang didukung: 500 per detik	Ya	Jumlah maksimum GetAuthorizationToken permintaan yang dapat Anda buat per detik di Wilayah saat ini.

Nama	Default	Dapat disesu an	Deskripsi
Tingkat GetDownloadUrlForLayer permintaan	Setiap Wilayah yang didukung: 3.000 per detik	Ya	Jumlah maksimum GetDownloadUrlForLayer permintaan yang dapat Anda buat per detik di Wilayah saat ini. Saat gambar ditarik, GetDownloadUrlForLayer API dipanggil sekali per layer gambar yang belum di-cache. Jika Anda meminta peningkatan kuota untuk API ini, tinjau juga BatchGetImage penggunaan Anda.
Tingkat InitiateLayerUpload permintaan	Setiap Wilayah yang didukung: 100 per detik	Ya	Jumlah maksimum InitiateLayerUpload permintaan yang dapat Anda buat per detik di Wilayah saat ini. Saat gambar didorong, InitiateLayerUpload API dipanggil sekali per layer gambar yang belum diunggah. Apakah lapisan gambar telah diunggah atau tidak ditentukan oleh tindakan BatchCheckLayerAvailability API.

Nama	Default	Dapat disesu an	Deskripsi
Tingkat PutImage permintaan	Setiap Wilayah yang didukung: 10 per detik	Ya	Jumlah maksimum PutImage permintaan yang dapat Anda buat per detik di Wilayah saat ini. Ketika gambar didorong dan semua lapisan gambar baru telah diunggah, PutImage API dipanggil sekali untuk membuat atau memperbarui manifes gambar dan tag yang terkait dengan gambar.
Tingkat UploadLayerPart permintaan	Setiap Wilayah yang didukung: 500 per detik	Ya	Jumlah maksimum UploadLayerPart permintaan yang dapat Anda buat per detik di Wilayah saat ini. Ketika gambar didorong, setiap layer gambar baru diunggah dalam beberapa bagian dan UploadLayerPart API dipanggil sekali per setiap bagian layer gambar baru.
Tingkat pemindaian citra	Setiap Wilayah yang didukung: 1	Tidak	Jumlah maksimum pemindaian citra per citra, per 24 jam.

Nama	Default	Dapat disesuaikan	Deskripsi
Repositori terdaftar	Setiap Wilayah yang didukung: 10.000	Ya	Jumlah maksimum repositori yang dapat Anda buat di akun ini di Wilayah saat ini.
Aturan per kebijakan siklus hidup	Setiap Wilayah yang didukung: 50	Tidak	Jumlah maksimum aturan dalam kebijakan siklus hidup
Aturan per konfigurasi replikasi	Setiap Wilayah yang didukung: 10	Tidak	Jumlah maksimum aturan dalam konfigurasi replikasi.
Tag per citra	Setiap Wilayah yang didukung: 1.000	Tidak	Jumlah maksimum tanda per citra.
Tujuan unik di semua aturan dalam konfigurasi replikasi	Setiap Wilayah yang didukung: 25	Tidak	Jumlah maksimum tujuan unik di semua aturan dalam konfigurasi replikasi.

Mengelola kuota layanan Amazon ECR Anda di AWS Management Console

Amazon ECR telah berintegrasi dengan Service Quotas, sebuah layanan AWS yang memungkinkan Anda untuk melihat dan mengelola kuota Anda dari lokasi pusat. Untuk informasi selengkapnya, lihat [Apa Itu Service Quotas?](#) dalam Panduan Pengguna Service Quotas.

Service Quotas mempermudah Anda mencari nilai semua kuota layanan Amazon ECR.

Untuk melihat kuota layanan Amazon ECR (AWS Management Console)

1. Buka konsol Service Quotas di <https://console.aws.amazon.com/servicequotas/>.

2. Di panel navigasi, pilih Layanan AWS.
3. Dari daftar AWS layanan, cari dan pilih Amazon Elastic Container Registry (Amazon ECR).

Di daftar Kuota layanan, Anda dapat melihat nama kuota layanan, nilai yang diterapkan (jika tersedia), kuota default AWS, dan apakah nilai kuota dapat disesuaikan.

4. Untuk melihat informasi tambahan tentang kuota layanan, seperti deskripsi, pilih nama kuota.

Untuk meminta kenaikan kuota, lihat [Meminta kenaikan kuota](#) di Panduan Pengguna Service Quotas.

Membuat CloudWatch alarm untuk memantau metrik penggunaan API

Amazon ECR menyediakan metrik CloudWatch penggunaan yang sesuai dengan kuota AWS layanan untuk setiap API yang terlibat dengan autentikasi registri, dorongan gambar, dan tindakan tarik gambar. Di konsol Service Quotas, Anda dapat memvisualisasikan penggunaan Anda pada grafik dan mengonfigurasi alarm yang memperingatkan Anda ketika penggunaan mendekati kuota layanan. Untuk informasi selengkapnya, lihat [Metrik penggunaan Amazon ECR](#).

Gunakan langkah-langkah berikut untuk membuat CloudWatch alarm berdasarkan salah satu metrik penggunaan Amazon ECR API.

Untuk membuat alarm berdasarkan kuota penggunaan Amazon ECR Anda (AWS Management Console)

1. Buka konsol Service Quotas di <https://console.aws.amazon.com/servicequotas/>.
2. Di panel navigasi, pilih Layanan AWS.
3. Dari daftar AWS layanan, cari dan pilih Amazon Elastic Container Registry (Amazon ECR).
4. Di daftar Kuota layanan, pilih kuota penggunaan Amazon ECR yang ingin Anda buat alarm.
5. Di bagian alarm CloudWatch Acara Amazon, pilih Buat.
6. Untuk Ambang batas Alarm, pilih persentase nilai kuota yang ingin Anda tetapkan sebagai nilai alarm.
7. Untuk Nama alarm, masukkan nama untuk alarm, lalu pilih Buat.

Pemecahan masalah Amazon ECR

Bab ini membantu Anda menemukan informasi diagnostik untuk Amazon Elastic Container Registry (Amazon ECR), dan menyediakan langkah-langkah pemecahan masalah untuk masalah umum dan pesan kesalahan.

Topik

- [Mengaktifkan output debug Docker](#)
- [Mengaktifkan AWS CloudTrail](#)
- [Mengoptimalkan performa untuk Amazon ECR](#)
- [Memecahkan masalah kesalahan dengan perintah Docker saat menggunakan Amazon ECR](#)
- [Memecahkan masalah pesan kesalahan Amazon ECR](#)

Mengaktifkan output debug Docker

Untuk memulai debugging masalah terkait Docker, Anda harus mulai dengan mengaktifkan output debugging Docker pada daemon Docker yang berjalan pada instans host Anda. Untuk informasi lebih lanjut tentang mengaktifkan debugging Docker jika Anda menggunakan citra yang ditarik dari Amazon ECR pada instans kontainer Amazon ECS, lihat [Mengaktifkan Output Debug Docker](#) dalam Panduan Developer Amazon Elastic Container Service.

Mengaktifkan AWS CloudTrail

Informasi tambahan tentang kesalahan yang dikembalikan oleh Amazon ECR dapat ditemukan dengan mengaktifkan AWS CloudTrail, yang merupakan layanan yang merekam AWS panggilan untuk akun Anda AWS . CloudTrail mengirimkan file log ke bucket Amazon S3. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan apa yang berhasil dibuat untuk AWS layanan, siapa yang membuat permintaan, kapan dibuat, dan sebagainya. Untuk mempelajari selengkapnya CloudTrail, termasuk cara mengaktifkannya dan menemukan file log Anda, lihat [Panduan AWS CloudTrail Pengguna](#). Untuk informasi selengkapnya tentang penggunaan CloudTrail dengan Amazon ECR, lihat [Mencatat tindakan Amazon ECR dengan AWS CloudTrail](#).

Mengoptimalkan performa untuk Amazon ECR

Bagian berikut memberikan rekomendasi tentang pengaturan dan strategi yang dapat digunakan untuk mengoptimalkan performa ketika menggunakan Amazon ECR.

Gunakan Docker versi 1.10 ke atas untuk memanfaatkan unggahan lapisan secara simultan

Citra Docker terdiri dari lapisan, yang merupakan tahap pembangunan intermediate dari citra. Setiap baris dalam Dockerfile menghasilkan pembuatan lapisan baru. Ketika Anda menggunakan Docker versi 1.10 ke atas, Docker secara default mendorong sebanyak mungkin lapisan sebagai unggahan simultan ke Amazon ECR, sehingga waktu unggah lebih cepat.

Gunakan citra dasar yang lebih kecil

Citra default yang tersedia melalui Docker Hub mungkin berisi banyak dependensi yang tidak diperlukan aplikasi Anda. Pertimbangkan untuk menggunakan citra yang lebih kecil yang dibuat dan dikelola oleh orang lain di komunitas Docker, atau membangun citra dasar Anda sendiri menggunakan citra scratch minimal Docker. Untuk informasi selengkapnya, lihat [Membuat citra dasar](#) dalam dokumentasi Docker.

Tempatkan dependensi yang paling tidak berubah sebelumnya di Dockerfile

Docker men-cache lapisan, dan itu mempercepat waktu pembangunan. Jika tidak ada yang berubah pada lapisan sejak pembuatan terakhir, Docker menggunakan versi yang di-cache alih-alih membangun kembali lapisan. Namun, setiap lapisan tergantung pada lapisan yang ada sebelumnya. Jika sebuah lapisan berubah, Docker mengkompilasi ulang tidak hanya lapisan itu, tetapi juga setiap lapisan yang datang setelah lapisan tersebut.

Untuk meminimalkan waktu yang diperlukan untuk membangun kembali Dockerfile dan mengunggah ulang lapisan, pertimbangkan untuk menempatkan dependensi yang paling tidak sering berubah sebelumnya di Dockerfile Anda. Tempatkan dependensi yang berubah dengan cepat (seperti kode sumber aplikasi Anda) nanti di tumpukan.

Perintah rantai untuk menghindari penyimpanan file yang tidak perlu

File-file intermediate yang dibuat pada lapisan tetap menjadi bagian dari lapisan tersebut meskipun dihapus dalam lapisan berikutnya. Pertimbangkan contoh berikut:

```
WORKDIR /tmp
RUN wget http://example.com/software.tar.gz
RUN wget tar -xvf software.tar.gz
RUN mv software/binary /opt/bin/myapp
```

```
RUN rm software.tar.gz
```

Dalam contoh ini, lapisan yang dibuat oleh perintah RUN pertama dan kedua berisi file .tar.gz asli dan semua isinya yang di-unzip. Ini meskipun file .tar.gz dihapus oleh perintah RUN keempat. Perintah ini dapat dirantai bersama-sama menjadi pernyataan RUN tunggal untuk memastikan bahwa file-file yang tidak perlu ini bukan bagian dari citra Docker akhir:

```
WORKDIR /tmp
RUN wget http://example.com/software.tar.gz &&\
  wget tar -xvf software.tar.gz &&\
  mv software/binary /opt/bin/myapp &&\
  rm software.tar.gz
```

Gunakan titik akhir regional terdekat

Anda dapat mengurangi latensi dalam menarik citra dari Amazon ECR dengan memastikan bahwa Anda menggunakan titik akhir regional yang terdekat dengan tempat aplikasi Anda berjalan. Jika aplikasi Anda berjalan pada instans Amazon EC2, Anda dapat menggunakan kode shell berikut ini untuk mendapatkan wilayah dari Availability Zone instans:

```
REGION=$(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone
|\
  sed -n 's/\(\\d*\)[a-zA-Z]*$/\1/p')
```

Wilayah dapat diteruskan ke AWS CLI perintah menggunakan --region parameter, atau ditetapkan sebagai wilayah default untuk profil menggunakan aws configure perintah. Anda juga dapat mengatur wilayah saat melakukan panggilan menggunakan AWS SDK. Untuk informasi lebih lanjut, lihat dokumentasi SDK untuk bahasa pemrograman khusus Anda.

Memecahkan masalah kesalahan dengan perintah Docker saat menggunakan Amazon ECR

Dalam beberapa kasus, menjalankan perintah Docker terhadap Amazon ECR dapat mengakibatkan pesan kesalahan. Beberapa pesan kesalahan umum dan kemungkinan solusi dijelaskan di bawah ini.

Topik

- [Kesalahan: "Verifikasi Sistem Berkas Gagal" atau "404: Citra Tidak Ditemukan" saat menarik citra dari repositori Amazon ECR](#)

- [Kesalahan: "Verifikasi Lapisan Sistem Berkas Gagal" saat menarik citra dari Amazon ECR](#)
- [Kesalahan HTTP 403 atau kesalahan "no basic auth credentials" ketika mendorong ke repositori](#)

Kesalahan: "Verifikasi Sistem Berkas Gagal" atau "404: Citra Tidak Ditemukan" saat menarik citra dari repositori Amazon ECR

Anda mungkin menerima kesalahan `Filesystem verification failed` ketika menggunakan perintah `docker pull` untuk menarik citra dari repositori Amazon ECR dengan Docker versi 1.9 atau di atasnya. Anda mungkin menerima kesalahan `404: Image not found` saat Anda menggunakan Docker versi sebelum 1.9.

Beberapa kemungkinan alasan dan penjelasannya diberikan di bawah ini.

Disk lokal penuh

Jika disk lokal yang Anda jalankan `docker pull` penuh, maka hash SHA-1 yang dihitung pada file lokal mungkin berbeda dari yang dihitung oleh Amazon ECR. Periksa apakah disk lokal Anda memiliki cukup ruang kosong untuk menyimpan citra Docker yang Anda tarik. Anda juga dapat menghapus citra lama untuk memberi ruang bagi yang baru. Gunakan perintah `docker images` untuk melihat daftar semua citra Docker yang diunduh secara lokal, bersama dengan ukurannya.

Client tidak dapat terhubung ke repositori jarak jauh karena kesalahan jaringan

Panggilan ke repositori Amazon ECR memerlukan koneksi internet yang berfungsi. Verifikasi pengaturan jaringan Anda, dan verifikasi bahwa alat dan aplikasi lain dapat mengakses sumber daya di internet. Jika Anda menjalankan `docker pull` pada instans Amazon EC2 di subnet pribadi, verifikasi bahwa subnet memiliki rute ke internet. Gunakan server penerjemahan alamat jaringan (NAT) atau gateway NAT terkelola.

Saat ini, panggilan ke repositori Amazon ECR juga memerlukan akses jaringan melalui firewall perusahaan Anda ke Amazon Simple Storage Service (Amazon S3). Jika organisasi Anda menggunakan perangkat lunak firewall atau perangkat NAT yang memungkinkan titik akhir layanan, pastikan bahwa titik akhir layanan Amazon S3 untuk Wilayah Anda saat ini diperbolehkan.

Jika Anda menggunakan Docker di belakang proksi HTTP, Anda dapat mengonfigurasi Docker dengan pengaturan proksi yang sesuai. Untuk informasi selengkapnya, lihat [Proksi HTTP](#) dalam dokumentasi Docker.

Kesalahan: "Verifikasi Lapisan Sistem Berkas Gagal" saat menarik citra dari Amazon ECR

Anda mungkin menerima kesalahan `image image-name not found` saat menarik citra menggunakan perintah `docker pull`. Jika Anda memeriksa log Docker, Anda mungkin melihat kesalahan seperti berikut ini:

```
filesystem layer verification failed for digest sha256:2b96f...
```

Kesalahan ini menunjukkan bahwa satu atau beberapa lapisan untuk citra Anda gagal mengunduh. Beberapa kemungkinan alasan dan penjelasannya diberikan di bawah ini.

Anda menggunakan Docker versi lama

Kesalahan ini dapat terjadi dalam persentase kecil kasus saat menggunakan versi Docker kurang dari 1.10. Upgrade client Docker Anda menjadi 1.10 atau lebih tinggi.

Client Anda mengalami kesalahan jaringan atau disk

Disk penuh atau masalah jaringan dapat mencegah pengunduhan satu atau lebih lapisan, seperti yang dibahas sebelumnya tentang pesan `Filesystem verification failed`. Ikuti rekomendasi di atas untuk memastikan bahwa sistem file Anda tidak penuh, dan bahwa Anda telah mengaktifkan akses ke Amazon S3 dari dalam jaringan Anda.

Kesalahan HTTP 403 atau kesalahan "no basic auth credentials" ketika mendorong ke repositori

Ada kalanya Anda mungkin menerima kesalahan HTTP 403 (Forbidden), atau pesan kesalahan `no basic auth credentials` dari perintah `docker push` atau `docker pull`, bahkan jika Anda telah berhasil diautentikasi ke Docker menggunakan perintah `aws ecr get-login-password`. Berikut ini adalah beberapa penyebab yang diketahui dari masalah ini:

Anda telah mengautentikasi ke wilayah yang berbeda

Permintaan autentikasi terkait dengan wilayah tertentu, dan tidak dapat digunakan di seluruh wilayah. Misalnya, jika Anda mendapatkan token otorisasi dari US West (Oregon), Anda tidak dapat menggunakannya untuk mengautentikasi terhadap repositori Anda di US East (N. Virginia). Untuk mengatasi masalah ini, pastikan bahwa Anda telah mengambil token autentikasi dari

Wilayah yang sama dengan tempat repositori Anda berada. Untuk informasi selengkapnya, lihat [the section called “Otentikasi registri”](#).

Anda telah mengautentikasi untuk mendorong ke repositori yang tidak memiliki izin

Anda tidak memiliki izin yang diperlukan untuk mendorong ke repositori. Untuk informasi selengkapnya, lihat [Kebijakan repositori pribadi](#).

Token Anda kedaluwarsa

Masa kedaluwarsa token otorisasi default untuk token yang diperoleh dengan menggunakan operasi `GetAuthorizationToken` adalah 12 jam.

Bug di pengelola kredensial `wincred`

Beberapa versi Docker untuk Windows menggunakan pengelola kredensial yang disebut dengan `wincred`, yang tidak menangani perintah masuk Docker yang dihasilkan oleh `aws ecr get-login-password` dengan benar (Untuk informasi selengkapnya, lihat <https://github.com/docker/docker/issues/22910>). Anda dapat menjalankan perintah masuk Docker yang merupakan output, tetapi ketika Anda mencoba untuk mendorong atau menarik citra, perintah tersebut gagal. Anda dapat bekerja di sekitar bug ini dengan menghapus skema `https://` dari argumen registri dalam perintah masuk Docker yang merupakan output dari `aws ecr get-login-password`. Contoh perintah masuk Docker tanpa skema HTTPS ditampilkan di bawah ini.

```
docker login -u AWS -p <password> <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

Memecahkan masalah pesan kesalahan Amazon ECR

Dalam beberapa kasus, panggilan API yang telah Anda picu melalui konsol Amazon ECS atau AWS CLI keluar dengan pesan kesalahan. Beberapa pesan kesalahan umum dan kemungkinan solusi dijelaskan di bawah ini.

HTTP 429: Terlalu Banyak Permintaan atau `ThrottlingException`

Anda mungkin menerima 429: Too Many Requests kesalahan atau `ThrottlingException` kesalahan dari satu atau beberapa tindakan Amazon ECR atau panggilan API. Hal ini menunjukkan bahwa Anda memanggil titik akhir tunggal di Amazon ECR berulang kali dalam jangka waktu pendek, dan bahwa permintaan Anda semakin dibatasi (`throttled`). `Throttling` terjadi ketika panggilan ke titik akhir tunggal dari satu pengguna melebihi ambang batas tertentu selama periode waktu.

Setiap operasi API di Amazon ECR memiliki kecepatan yang terkait dengannya. Misalnya, pembatasan untuk tindakan [GetAuthorizationToken](#) adalah 20 transaksi per detik (TPS), dengan lonjakan hingga 200 TPS diperbolehkan. Di setiap wilayah, setiap akun menerima bucket yang dapat menyimpan hingga 200 kredit `GetAuthorizationToken`. Kredit ini diisi ulang dengan kecepatan 20 per detik. Jika bucket Anda memiliki 200 kredit, Anda bisa mencapai 200 transaksi API `GetAuthorizationToken` per detik untuk satu detik, dan kemudian mempertahankan 20 transaksi per detik tanpa batas waktu. Untuk informasi selengkapnya tentang batas tarif Amazon ECR API, lihat [Kuota layanan Amazon ECR](#).

Untuk menangani kesalahan throttling, terapkan fungsi percobaan ulang dengan backoff tambahan ke dalam kode Anda. Untuk informasi selengkapnya, lihat [Mencoba lagi perilaku](#) di AWS SDK dan Panduan Referensi Alat. Pilihan lainnya adalah meminta kenaikan batas tarif, yang dapat Anda lakukan menggunakan konsol Service Quotas. Untuk informasi selengkapnya, lihat [Mengelola kuota layanan Amazon ECR Anda di AWS Management Console](#).

HTTP 403: "Pengguna [arn] tidak memiliki otorisasi untuk melakukan [operasi]"

Anda mungkin menerima kesalahan berikut ketika mencoba untuk melakukan tindakan dengan Amazon ECR:

```
$ aws ecr get-login-password
A client error (AccessDeniedException) occurred when calling the GetAuthorizationToken
operation:
  User: arn:aws:iam::account-number:user/username is not authorized to perform:
  ecr:GetAuthorizationToken on resource: *
```

Hal ini menunjukkan bahwa pengguna Anda tidak memiliki izin yang diberikan untuk menggunakan Amazon ECR, atau bahwa izin tersebut tidak diatur dengan benar. Secara khusus, jika Anda melakukan tindakan terhadap repositori Amazon ECR, verifikasi bahwa pengguna telah diberikan izin untuk mengakses repositori tersebut. Untuk informasi selengkapnya tentang membuat dan memverifikasi izin untuk Amazon ECR, lihat [Identity and Access Management untuk Amazon Elastic Container Registry](#).

HTTP 404: kesalahan "Repositori Tidak Ada"

Jika Anda menentukan repositori Docker Hub yang saat ini tidak ada, Docker Hub membuatnya secara otomatis. Dengan Amazon ECR, repositori baru harus dibuat secara eksplisit sebelum

dapat digunakan. Hal ini mencegah repositori baru dibuat secara tidak sengaja (misalnya, karena kesalahan ketik), dan juga memastikan bahwa kebijakan akses keamanan yang sesuai secara eksplisit ditetapkan ke repositori baru. Untuk informasi selengkapnya tentang membuat repositori, lihat [Repositori pribadi Amazon ECR](#).

Kesalahan: Tidak dapat melakukan login interaktif dari perangkat non TTY

Jika Anda menerima kesalahan `Cannot perform an interactive login from a non TTY device`, langkah-langkah pemecahan masalah berikut akan membantu.

- Verifikasi bahwa Anda menggunakan AWS CLI versi 2 dan Anda tidak memiliki versi AWS CLI versi 1 yang bertentangan di sistem Anda. Untuk informasi selengkapnya, lihat [Menginstal atau memperbarui versi terbaru AWS CLI](#).
- Verifikasi bahwa Anda telah mengonfigurasi AWS CLI dengan kredensi yang valid. Untuk informasi selengkapnya, lihat [Menginstal atau memperbarui versi terbaru AWS CLI](#).
- Verifikasi bahwa sintaks AWS CLI perintah Anda benar.

Riwayat dokumen

Tabel berikut menjelaskan perubahan penting pada dokumentasi sejak rilis terakhir Amazon ECR. Kami juga rutin memperbarui dokumentasi untuk menjawab umpan balik yang Anda kirimkan kepada kami.

Perubahan	Deskripsi	Tanggal
Pembaruan kebijakan siklus hidup Amazon ECR untuk menambahkan dukungan untuk menggunakan wildcard	Amazon ECR menambahkan dukungan untuk wildcard dalam kebijakan siklus hidup melalui penggunaan <code>tagPatternList</code> parameter dalam aturan kebijakan siklus hidup. Untuk informasi selengkapnya, lihat Kebijakan siklus hidup .	18 Desember 2023
Templat pembuatan repositori Amazon ECR	Amazon ECR menambahkan dukungan untuk template pembuatan repositori. Untuk informasi selengkapnya, lihat Kelola templat pembuatan repositori Anda .	15 November 2023
Amazon ECR menarik cache ditambahkan didukung untuk registri upstream yang diautentikasi	Amazon ECR menambahkan dukungan untuk menggunakan pendaftar upstream yang memerlukan otentikasi untuk aturan cache pull through Anda. Untuk informasi selengkapnya, lihat Menggunakan aturan pull through cache .	15 November 2023
AWSECRPullThroughCache_ServiceRolePolicy – Perbaruan ke kebijakan yang sudah ada	Amazon ECR menambahkan izin baru ke kebijakan <code>AWSECRPullThroughCache_ServiceRolePolicy</code> . Izin ini memungkinkan Amazon ECR untuk mengambil konten terenkripsi dari rahasia Secrets Manager. Ini diperlukan saat menggunakan aturan pull through cache untuk menyimpan gambar dari registri hulu yang memerlukan otentikasi.	15 November 2023
Penandatanganan gambar Amazon ECR	Amazon ECR dan AWS Signer menambahkan dukungan untuk membuat dan mendorong tanda tangan gambar kontainer menggunakan klien Notaris.	6. Juni 2023

Perubahan	Deskripsi	Tanggal
	Untuk informasi selengkapnya, lihat Menandatangani gambar .	
Menambahkan registri kontainer Kubernetes untuk menarik aturan cache	Amazon ECR menambahkan dukungan untuk membuat aturan cache pull through untuk registri container Kubernetes. Untuk informasi selengkapnya, lihat Menggunakan aturan pull through cache .	1 Juni 2023
Amazon ECR meningkatkan dukungan durasi pemindaian	Amazon Inspector menambahkan dukungan untuk menyetel durasi penyimpanan Anda dipantau saat pemindaian yang disempurnakan diaktifkan. Untuk informasi selengkapnya, lihat Mengubah durasi pemindaian yang ditingkatkan .	28 Juni 2022
Amazon ECR mengirimkan metrik jumlah tarik repositori ke Amazon CloudWatch	Amazon ECR mengirimkan metrik jumlah tarik repositori ke Amazon CloudWatch. Untuk informasi selengkapnya, lihat Metrik repositori Amazon ECR .	6 Januari 2022
Dukungan replikasi yang diperluas	Amazon ECR menambahkan dukungan untuk memfilter repositori mana yang direplikasi. Untuk informasi selengkapnya, lihat Replikasi citra pribadi .	21 September 2021
Kebijakan terkelola AWS untuk Amazon ECR	Amazon ECR menambahkan dokumentasi kebijakan terkelola AWS. Untuk informasi selengkapnya, lihat AWS kebijakan terkelola untuk Amazon Elastic Container Registry .	24 Juni 2021
Replikasi Lintas-Wilayah dan Lintas-Akun	Amazon ECR menambahkan support untuk mengkonfigurasi pengaturan replikasi untuk registrasi privat Anda. Untuk informasi selengkapnya, lihat Pengaturan registri pribadi .	8 Desember 2020

Perubahan	Deskripsi	Tanggal
Support artefak OCI	<p>Amazon ECR menambahkan support untuk mendorong dan menarik artefak Open Container Initiative (OCI). Parameter <code>artifactMediaType</code> baru ditambahkan ke <code>Respon DescribeImages</code> API untuk menunjukkan jenis artefak.</p> <p>Untuk informasi selengkapnya, lihat Mendorong grafik Helm.</p>	24 Agustus 2020
Enkripsi saat tidak aktif	<p>Amazon ECR menambahkan support untuk mengkonfigurasi enkripsi untuk repositori Anda menggunakan enkripsi server-side dengan kunci terkelola pelanggan yang disimpan di AWS Key Management Service (AWS KMS).</p> <p>Untuk informasi selengkapnya, lihat Enkripsi diam.</p>	29 Juli 2020
citra multi-arsitektur	<p>Amazon ECR menambahkan support untuk membuat dan mendorong daftar Docker manifest yang digunakan untuk citra multi-arsitektur.</p> <p>Untuk informasi selengkapnya, lihat Mendorong citra multi-arsitektur.</p>	28 April 2020
Metrik Penggunaan Amazon ECR	<p>Amazon ECR menambahkan metrik CloudWatch penggunaan yang memberikan visibilitas ke dalam penggunaan sumber daya akun Anda. Anda juga memiliki kemampuan untuk membuat CloudWatch alarm dari konsol Service Quotas CloudWatch dan Service Quotas untuk mendapatkan peringatan ketika penggunaan Anda mendekati kuota layanan yang diterapkan.</p> <p>Untuk informasi selengkapnya, lihat Metrik penggunaan Amazon ECR.</p>	28 Feb 2020

Perubahan	Deskripsi	Tanggal
Service quotas Amazon ECR yang diperbarui	<p>Service quotas Amazon ECR yang diperbarui untuk menyertakan kuota per-API.</p> <p>Untuk informasi selengkapnya, lihat Kuota layanan Amazon ECR.</p>	19 Feb 2020
Perintah get-login -password yang ditambahkan	<p>Support yang ditambahkan untuk get-login-password , yang menyediakan metode sederhana dan aman untuk mengambil token otorisasi.</p> <p>Untuk informasi selengkapnya, lihat Menggunakan token otorisasi.</p>	4 Feb 2020
Pemindaian citra	<p>Support yang ditambahkan untuk pemindaian citra, yang membantu dalam mengidentifikasi kerentanan perangkat lunak dalam citra kontainer Anda. Amazon ECR menggunakan basis data Common Vulnerabilities and Exposures (CVE) dari sumber terbuka proyek CoreOS Clair dan memberi Anda daftar temuan pemindaian.</p> <p>Untuk informasi selengkapnya, lihat Pemindaian citra.</p>	24 Okt 2019
Kebijakan VPC Endpoint	<p>Support yang ditambahkan untuk menetapkan kebijakan IAM pada antarmuka VPC endpoint Amazon ECR.</p> <p>Untuk informasi selengkapnya, lihat Buat kebijakan titik akhir untuk VPC endpoint Amazon ECR.</p>	26 Sep 2019
Tanda Ketetapan Citra	<p>Support yang ditambahkan untuk mengkonfigurasi repositori agar tetap untuk mencegah ketimpaan tanda citra.</p> <p>Untuk informasi selengkapnya, lihat Tanda ketetapan citra.</p>	25 Juli 2019

Perubahan	Deskripsi	Tanggal
VPC endpoint Antarmuka (AWS PrivateLink)	<p>Menambahkan dukungan untuk mengonfigurasi titik akhir VPC antarmuka yang didukung oleh AWS PrivateLink. Hal ini memungkinkan Anda untuk membuat koneksi privat antara VPC Anda dan Amazon ECR tanpa memerlukan akses melalui internet, melalui instans NAT, koneksi VPN, atau AWS Direct Connect.</p> <p>Untuk informasi selengkapnya, lihat Titik akhir VPC antarmuka Amazon ECR (AWS PrivateLink).</p>	25 Jan 2019
Penandaan sumber daya	<p>Amazon ECR menambahkan support untuk menambahkan tanda metadata ke repositori Anda.</p> <p>Untuk informasi selengkapnya, lihat Menandai repositori pribadi.</p>	18 Des 2018
Peggantian Nama Amazon ECR	<p>Nama Amazon Elastic Container Registry telah diganti (sebelumnya Amazon EC2 Container Registry).</p>	21 Nov 2017
Kebijakan siklus hidup	<p>Kebijakan siklus hidup Amazon ECR memungkinkan Anda untuk menentukan manajemen siklus hidup citra dalam repositori.</p> <p>Untuk informasi selengkapnya, lihat Kebijakan siklus hidup.</p>	11 Okt 2017
Support Amazon ECR untuk Docker image manifest 2, skema 2	<p>Kini Amazon ECR dapat digunakan pada Docker Image Manifest V2 Skema 2 (digunakan dengan Docker versi 1.10 dan yang lebih baru).</p> <p>Untuk informasi selengkapnya, lihat Format manifest citra kontainer.</p>	27 Jan 2017
Ketersediaan Umum Amazon ECR	<p>Amazon Elastic Container Registry (Amazon ECR) adalah layanan registrasi Docker AWS terkelola yang aman, dapat diskalakan, dan andal.</p>	21 Des 2015

AWSGlosarium

Untuk AWS terminologi terbaru, lihat [AWSglosarium di Referensi](#). Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.