

Replicate filtered Amazon ECR container images across accounts or Regions

[PDF \(prescriptive-guidance.pdf#replicate-filtered-amazon-ecr-container-images-across-accounts-or-regions\)](#)

Created by Abdal Garuba (AWS)

Environ ment: Product ion	Technologies: Containers & microservices; DevOps	AWS services: AWS EC2 Container Registry; Amazon CloudWatch; AWS CodeBuild; AWS Identity and Access Management; AWS CLI
--	--	---

Summary

This pattern describes how to replicate container images that are stored in Amazon Elastic Container Registry (Amazon ECR) across Amazon Web Services (AWS) accounts and AWS Regions, based on image tag patterns. The pattern uses Amazon CloudWatch Events to listen for push events for images that have a predefined, custom tag. A push event starts an AWS CodeBuild project and passes the image details to it. The CodeBuild project copies the images from the source Amazon ECR registry to the destination registry based on the details provided.


Amazon ECR supports cross-Region and cross-account replication for images. Both options replicate all new images that are pushed to the source registry. (For more information, see [Private image replication \(https://docs.aws.amazon.com/AmazonECR/latest/userguide/replication.html\)](https://docs.aws.amazon.com/AmazonECR/latest/userguide/replication.html) in the Amazon ECR documentation.) However, there is no way to filter the images copied across AWS Regions or accounts based on any criteria.

This pattern copies images that have specific tags across accounts. For

example, you can use this pattern to copy only production-ready, secure images to the production AWS account. In the development account, after images are thoroughly tested, you can add a predefined tag to the secure images and use the steps in this pattern to copy the marked images to the production account.

Prerequisites and limitations

Prerequisites

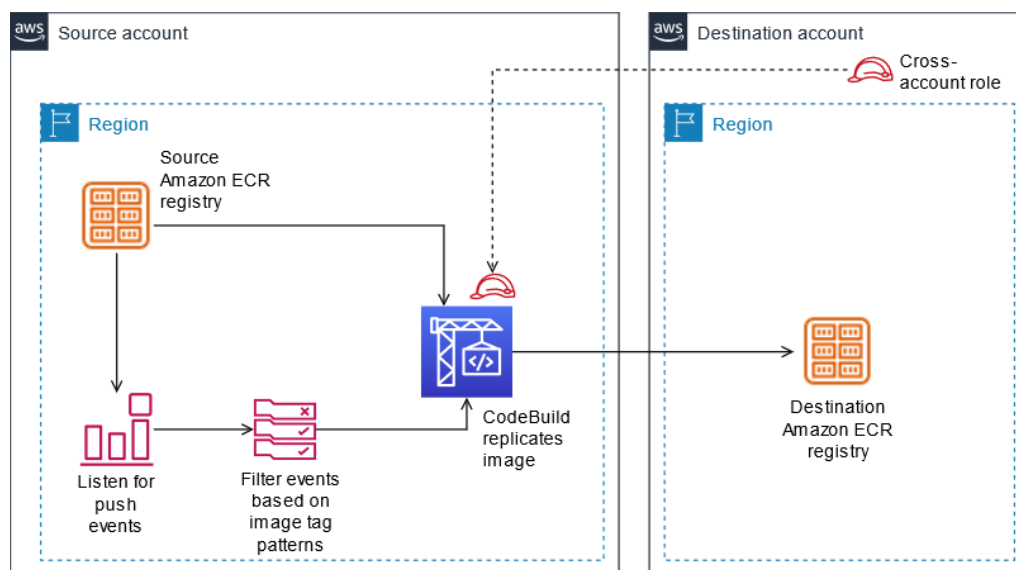
- An active AWS account for source and destination Amazon ECR registries
- Administrative permissions for the tools used in this pattern
- [Docker](https://docs.docker.com/get-docker/)  (<https://docs.docker.com/get-docker/>) installed on your local machine for testing
- [AWS Command Line Interface \(AWS CLI\)](https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html) (<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html>) , for authenticating into Amazon ECR

Limitations

- This pattern watches the push events of the source registry in only one AWS Region. You can deploy this pattern to other Regions to watch registries in those Regions.
 - In this pattern, one Amazon CloudWatch Events rule listens for a single image tag pattern. If you want to check for multiple patterns, you can add events to listen for additional image tag patterns.
-

Architecture

Target architecture



Automation and scale




This pattern can be automated with an infrastructure as code (IaC) script and deployed at scale. To use AWS CloudFormation templates to deploy this pattern, download the attachment and follow the instructions in the [Additional information \(#replicate-filtered-amazon-ecr-container-images-across-accounts-or-regions-additional\)](#) section.

You can point multiple Amazon CloudWatch Events events (with different custom event patterns) to the same AWS CodeBuild project to replicate multiple image tag patterns, but you will need to update the secondary validation in the `buildspec.yaml` file (which is included in the attachment and in the [Tools \(#replicate-filtered-amazon-ecr-container-images-across-accounts-or-regions-tools\)](#) section) as follows to support multiple patterns.


```
...
if [[ ${IMAGE_TAG} != release-* ]]; then
...
```

Tools

Amazon services

- [IAM](https://aws.amazon.com/iam/)  (https://aws.amazon.com/iam/) – AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. In this pattern, you would need to create the cross-account IAM role that AWS CodeBuild will assume when pushing container images to the destination registry.
- [Amazon ECR](https://aws.amazon.com/ecr/)  (https://aws.amazon.com/ecr/) – Amazon Elastic Container Registry (Amazon ECR) is a fully managed container registry that makes it easy to store, manage, share, and deploy your container images and artifacts anywhere. Image push actions to the source registry send system event details to the event bus that is picked up by Amazon CloudWatch Events.
- [AWS CodeBuild](https://aws.amazon.com/codebuild/)  (https://aws.amazon.com/codebuild/) – AWS CodeBuild is a fully managed continuous integration service that provides compute power to perform jobs such as compiling source code, running tests, and producing artifacts that are ready to be deployed. This pattern uses AWS CodeBuild to perform the copy action from the source Amazon ECR registry to the destination registry.
- [CloudWatch Events](https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/WhatIsCloudWatchEvents.html) (https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/WhatIsCloudWatchEvents.html) – Amazon CloudWatch Events delivers a stream of system events that describe changes in AWS resources. This pattern uses rules to match Amazon ECR push actions with a specific image tag pattern.

Tools

- [Docker CLI](https://www.docker.com/)  (https://www.docker.com/) – Docker is a tool that makes it easier to create and manage containers. Containers pack an application and all its dependencies into one unit or package that can easily be deployed on any platform that supports the container runtime.

Code

You can implement this pattern in two ways:

- Automated setup: Deploy the two AWS CloudFormation templates provided in the attachment. For instructions, see the [Additional information \(#replicate-filtered-amazon-ecr-container-images-across-accounts-or-regions-additional\)](#) section.

- Manual setup: Follow the steps in the [Epics \(#replicate-filtered-amazon-ecr-container-images-across-accounts-or-regions-epics\)](#) section.

Sample buildspec.yaml

If you're using the CloudFormation templates that are provided with this pattern, the `buildspec.yaml` file is included in the CodeBuild resources.

```
version: 0.2
env:
  shell: bash
phases:
  install:
    commands:
      - export CURRENT_ACCOUNT=$(echo
${CODEBUILD_BUILD_ARN} | cut -d':' -f5)
      - export
CURRENT_ECR_REGISTRY=${CURRENT_ACCOUNT}.dkr.ecr.$
{AWS_REGION}.amazonaws.com
      - export
DESTINATION_ECR_REGISTRY=${DESTINATION_ACCOUNT}.d
kr.ecr.${DESTINATION_REGION}.amazonaws.com
  pre_build:
    on-failure: ABORT
    commands:
      - echo "Validating Image Tag
${IMAGE_TAG}"
      - |
        if [[ ${IMAGE_TAG} != release-* ]];
then
          aws codebuild stop-build --id
${CODEBUILD_BUILD_ID}
          sleep 60
          exit 1
        fi
      - aws ecr get-login-password --region
${AWS_REGION} | docker login -u AWS --password-
stdin ${CURRENT_ECR_REGISTRY}
      - docker pull
${CURRENT_ECR_REGISTRY}/${REPO_NAME}:${IMAGE_TAG}
```

```
    build:
      commands:
        - echo "Assume cross-account role"
        - CREDENTIALS=$(aws sts assume-role
--role-arn ${CROSS_ACCOUNT_ROLE_ARN} --role-
session-name Rolesession)
        - export
AWS_DEFAULT_REGION=${DESTINATION_REGION}
        - export AWS_ACCESS_KEY_ID=$(echo
${CREDENTIALS} | jq -r
'.Credentials.AccessKeyId')
        - export AWS_SECRET_ACCESS_KEY=$(echo
${CREDENTIALS} | jq -r
'.Credentials.SecretAccessKey')
        - export AWS_SESSION_TOKEN=$(echo
${CREDENTIALS} | jq -r
'.Credentials.SessionToken')
        - echo "Logging into cross-account
registry"
        - aws ecr get-login-password --region
${DESTINATION_REGION} | docker login -u AWS
--password-stdin ${DESTINATION_ECR_REGISTRY}
        - echo "Check if Destination Repository
exists, else create"
        - |
            aws ecr describe-repositories
--repository-names ${REPO_NAME} --region
${DESTINATION_REGION} \
            || aws ecr create-repository
--repository-name ${REPO_NAME} --region
${DESTINATION_REGION}
        - echo "retag image and push to
destination"
        - docker tag
${CURRENT_ECR_REGISTRY}/${REPO_NAME}:${IMAGE_TAG}
${DESTINATION_ECR_REGISTRY}/${REPO_NAME}:${IMAGE_
TAG}
        - docker push
${DESTINATION_ECR_REGISTRY}/${REPO_NAME}:${IMAGE_
TAG}
```

Epics

▼ Create IAM roles

Task	Description	Skills required
Create a CloudWatch Events role.	<p>In the source AWS account, create an IAM role for Amazon CloudWatch Events to assume. The role should have permissions to start a AWS CodeBuild project.</p> <p>To create the role by using the AWS CLI, follow the instructions in the IAM documentation (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-service.html#roles-creatingrole-service-cli).</p> <p>Example trust policy (trustpolicy.json):</p> <pre>{ "Version": "2012-10-17", "Statement": { "Effect": "Allow", "Principal": {"Service": "ec2.amazonaws.com"} },</pre>	AWS administrator, AWS DevOps, AWS systems administrator, Cloud administrator, Cloud architect, DevOps engineer

Task	Description	Skills required
	<pre>"Action": "sts:AssumeRole" } }</pre> <p>Example permission policy (permissionpolicy.json):</p> <pre>{ "Version": "2012-10-17", "Statement": { "Effect": "Allow", "Action": "codebuild:StartBuild", "Resource": "<CodeBuild Project ARN>" } }</pre>	
Create a CodeBuild role.	Create an IAM role for AWS CodeBuild to assume, by following the instructions in the IAM documentation (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-service.html#roles-creatingrole-service-cli). The role should have the following permissions: <ul style="list-style-type: none">• Permission to assume the	AWS administrator, AWS DevOps, AWS systems administrator, Cloud administrator, Cloud architect, DevOps

Task	Description	Skills required
	<p>destination cross-account role</p> <ul style="list-style-type: none">• Permission to create log groups and log streams, and to put log events• Read-only permissions to all Amazon ECR repositories, by adding the AmazonEC2ContainerRegistryReadOnly (https://docs.aws.amazon.com/AmazonECR/latest/userguide/security-iam-awsmanpol.html#security-iam-awsmanpol-AmazonEC2ContainerRegistryReadOnly) managed policy to the role• Permission to stop CodeBuild <p>Example trust policy (trustpolicy.json):</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Service": "codebuild.amazonaws.com" }, "Action":</pre>	engineer

Task	Description	Skills required
	<pre>"sts:AssumeRole" }] }</pre> <p>Example permission policy (permissionpolicy.json):</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["codebuild:StartBuild", "codebuild:StopBuild", "codebuild:Get*", "codebuild:List*", "codebuild:BatchGet*"], "Resource": "*", "Effect": "Allow" }, {</pre>	

Task	Description	Skills required
	<pre>"Action": ["logs:CreateLogGroup" , "logs:CreateLogStrea m" , "logs:PutLogEvents"], "Resource": "*", "Effect": "Allow" }, { "Action": "sts:AssumeRole", "Resource": "<ARN of destination role>", "Effect": "Allow", "Sid": "AssumeCrossAccountA rn" }] }</pre> <p>Attach the managed policy AmazonEC2ContainerRegistry ReadOnly to the CLI command as follows:</p>	

Task	Description	Skills required
	<pre>~\$ aws iam attach- role-policy --policy-arn arn:aws:iam::aws:pol icy/AmazonEC2Contain erRegistryReadOnly --role-name <name of CodeBuild Role></pre>	
Create a cross-account role.	<p>In the destination AWS account, create an IAM role for the AWS CodeBuild role for the source account to assume. The cross-account role should allow container images to create a new repository and upload container images to Amazon ECR.</p> <p>To create the IAM role by using the AWS CLI, follow the instructions in the IAM documentation (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-service.html#roles-creatingrole-service-cli).</p> <p>To allow the AWS CodeBuild project from the previous step, use the following trust policy:</p>	AWS administrator, AWS DevOps, Cloud administrator, Cloud architect, DevOps engineer, AWS systems administrator
	<pre>{ "Version": "2012-10-17",</pre>	

Task	Description	Skills required
	<pre>"Statement": { "Effect": "Allow", "Principal": { "AWS": "<ARN of source codebuild role>" }, "Action": "sts:AssumeRole" } }</pre> <p>To allow the AWS CodeBuild project from the previous step to save images in the destination registry, use the following permission policy:</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["ecr:GetDownloadUrlForLayer", "ecr:BatchCheckLayerAvailability", "ecr:PutImage",</pre>	

Task	Description	Skills required
	"ecr:InitiateLayerUp load".	

▼ **Create the CodeBuild project**

Tas k	Description	Skills required
Cre ate a Cod eBu ild pro ject .	<p>Create a AWS CodeBuild project in the source account by following the instructions in the AWS CodeBuild documentation (https://docs.aws.amazon.com/codebuild/latest/userguide/create-project-console.html) . The project should be in the same Region as the source registry.</p> <p>Configure the project as follows:</p> <ul style="list-style-type: none"> • Environment type: LINUX CONTAINER • Service role: CodeBuild Role • Privileged mode: true • Environment image: aws/codebuild /standard:x.x (use the latest image available) • Environment variables: <ul style="list-style-type: none"> ◦ CROSS_ACCOUNT_ROLE_A 	<p>AWS administrator , AWS DevOps, AWS systems administrator , Cloud administrator , Cloud architect, DevOps engineer</p>

Task	Description	Skills required
	<p>RN: The Amazon Resource Name (ARN) of the cross-account role</p> <ul style="list-style-type: none">◦ DESTINATION_REGION: The name of the cross-account Region◦ DESTINATION_ACCOUNT: The number of the destination account• Build specifications: Use the <code>buildspec.yaml</code> file listed in	

▼ Create the event

Task	Description	Skills required
Create a CloudWatch Events rule.	<p>Because the pattern uses the content filtering feature, you need to create the event by using Amazon EventBridge. Create the event and target by following the instructions in the EventBridge documentation (https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-create-rule.html), with a few modifications:</p> <ul style="list-style-type: none">• For Define pattern, choose Event Pattern, and then choose Custom pattern.• Copy the following custom events	<p>AWS administrator, AWS DevOps, AWS systems administrator, Cloud administrator, Cloud architect, DevOps</p>

Task	Description	Skills required
	<p>pattern sample code into the text box provided:</p> <pre data-bbox="548 384 1049 1211"> { "source": ["aws.ecr"], "detail-type": ["ECR Image Action"], "detail": { "action-type": ["PUSH"], "result": ["SUCCESS"], "image-tag": [{ "prefix": "release-"}] } } </pre> <ul style="list-style-type: none"> For Select targets, choose the AWS CodeBuild project, and paste the ARN for the AWS CodeBuild project that you created in the previous epic. For Configure Input, choose Input Transformer. <ul style="list-style-type: none"> In the Input Path text box, paste: <pre data-bbox="605 1736 1049 1997"> {"IMAGE_TAG": "\$. detail.image- tag", "REPO_NAME" </pre> 	engineer

Task	Description	Skills required
	<pre>:"\$.detail.repository-name"}</pre> <ul style="list-style-type: none"> In the Input Template text box, paste: <pre>{ "environmentVariablesOverride": [{ "name": "IMAGE_TAG", "value": "<IMAGE_TAG>" }, { "name": "REPO_NAME", "value": " </pre>	

▼ Validate

Task	Description	Skills required
Authenticate with Amazon ECR.	Authenticate to both source and destination registries by following the steps in the Amazon ECR documentation (https://docs.aws.amazon.com/AmazonECR/latest/userguide/registry_auth.html) .	AWS administrator , AWS DevOps, AWS systems administrator , Cloud administrator , DevOps engineer,

Task	Description	Skills required
		Cloud architect
Test image replication.	<p>In your source account, push a container image to a new or existing Amazon ECR source repository with an image tag prefixed with <code>release-</code>. To push the image, follow the steps in the Amazon ECR documentation (https://docs.aws.amazon.com/AmazonECR/latest/userguide/getting-started-cli.html#cli-push-image).</p> <p>You can monitor the progress of the CodeBuild project in the CodeBuild console (https://console.aws.amazon.com/codesuite/codebuild/home).</p> <p>After the CodeBuild project has completed successfully, sign in to the destination AWS account, open the Amazon ECR console and confirm that the image exists in the destination ECR registry.</p>	AWS administrator, AWS DevOps, AWS systems administrator, Cloud administrator, Cloud architect, DevOps engineer
Test image exclusion.	<p>In your source account, push a container image to a new or existing Amazon ECR source repository with an image tag that doesn't have the custom prefix.</p> <p>Confirm that the CodeBuild project isn't started, and that no container images appear in the</p>	AWS administrator, AWS DevOps, AWS systems administrator, Cloud administrator

Related resources

- [Getting started with CodeBuild \(https://docs.aws.amazon.com/codebuild/latest/userguide/getting-started-overview.html\)](https://docs.aws.amazon.com/codebuild/latest/userguide/getting-started-overview.html)
 - [Getting started with Amazon EventBridge \(https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-get-started.html\)](https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-get-started.html)
 - [Content-based filtering in Amazon EventBridge event patterns \(https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-event-patterns-content-based-filtering.html#filtering-prefix-matching\)](https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-event-patterns-content-based-filtering.html#filtering-prefix-matching)
 - [Delegate access across AWS accounts using IAM roles \(https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html\)](https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html)
-

Additional information

To automatically deploy the resources for this pattern, follow these steps:

1. Download the attachment and extract the two CloudFormation templates: `part-1-copy-tagged-images.yaml` and `part-2-destination-account-role.yaml`.
2. Log in to the [AWS CloudFormation console](https://console.aws.amazon.com/cloudformation/) <https://console.aws.amazon.com/cloudformation/>, and deploy `part-1-copy-tagged-images.yaml` in the same AWS account and Region as the source Amazon ECR registries. Update the parameters as needed. The template deploys the following resources:
 - Amazon CloudWatch Events IAM role
 - AWS CodeBuild project IAM role
 - AWS CodeBuild project
 - AWS CloudWatch Events rule
3. Take note of the value of `SourceRoleName` in the **Outputs** tab. You will need this value in the next step.
4. Deploy the second CloudFormation template, `part-2-destination-account-role.yaml`, in the AWS account that you want to copy the Amazon ECR container images to. Update the parameters as needed. For the `SourceRoleName` parameter, specify

the value from step 3. This template deploys the cross-account IAM role.

5. Validate image replication and exclusion, as described in the last step of the [Epics \(#replicate-filtered-amazon-ecr-container-images-across-accounts-or-regions-epics\)](#) section.

Attachments