



nextwork.org

Secure Packages with CodeArtifact



Sanjana Tripathy

The screenshot shows a web-based interface for managing CodeArtifact repositories. At the top, there's a navigation bar with links for 'Developer Tools', 'CodeArtifact', 'Repositories', and the current repository 'nextwork-devops-cicd'. Below the navigation, there's a brief description of the repository: 'This repository stores packages related to a Java web app created as a part of NextWork's CI/CD Pipeline series.' On the right side of the header, there are buttons for 'Delete repository', 'Apply repository policy', and 'Edit'. The main content area is titled 'Packages' and contains a table with one row of data:

Package name	Namespace	Format	Latest version	Latest publish date	Publish	Upstream
nextwork-web-project	com.nextwork.app	maven	1.0-SNAPSHOT	15 hours ago	Allow	Block



Introducing Today's Project!

In this project, I will demonstrate deploying a web app by launching an EC2 instance, setting up the app, connecting it to GitHub, creating a CodeArtifact repository, configuring EC2 permissions, and storing app packages securely.

Key tools and concepts

The services I used were EC2, CodeArtifact, and GitHub because they enabled me to host, manage, and version-control my web app. Key concepts I learnt included IAM roles, artifact repositories, Maven integration, and secure dependency management.

Project reflection

This project took me approximately a few hours to complete. The most challenging part was configuring the setup to be able to see the packages in my CodeArtifact repository. It was most rewarding when the setup finally worked.

This project is part three of my DevOps series where I'm building a CI/CD pipeline! I'll be working on the next project today to keep the momentum and continue strengthening my cloud and DevOps skills.

Sanjana Tripathy
NextWork Student

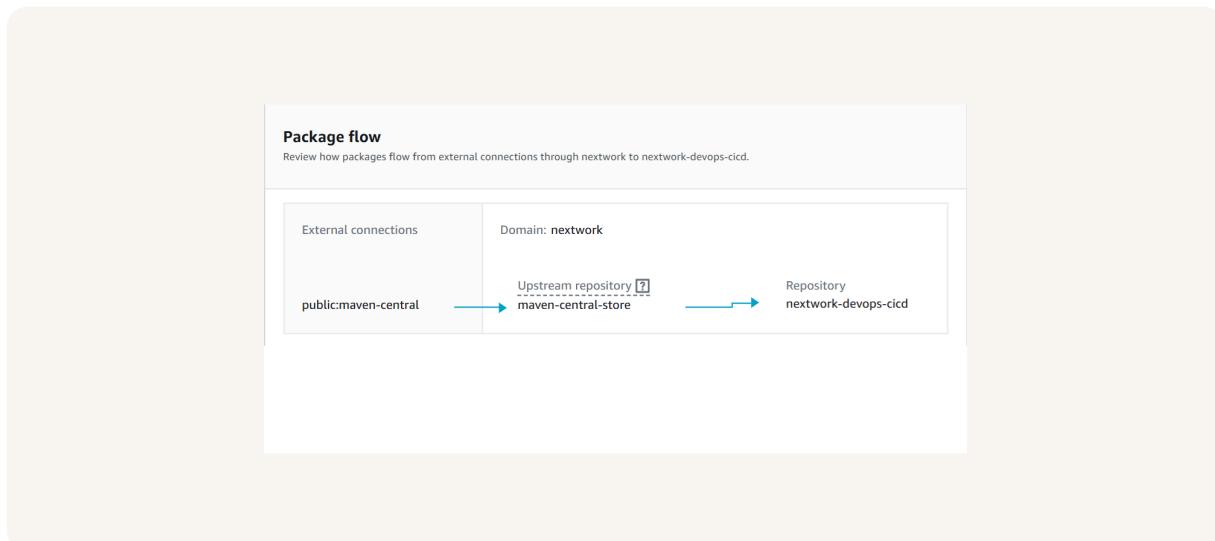
nextwork.org

CodeArtifact Repository

CodeArtifact is a fully managed artifact repository service. Engineering teams will use artifact repositories because they provide a secure, centralized place to store, manage, and share project dependencies efficiently.

A domain is a logical grouping of CodeArtifact repositories that allows centralized management of permissions and access. My domain will be nextwork, which will organize and control access to our project repositories efficiently.

A CodeArtifact repository can have an upstream repository, which means it can pull packages from another repository automatically. My repository's upstream repository will be set to the public Maven repository to access standard Java packages.



A circular profile picture of a young woman with dark hair, wearing a pink top, sitting on a blue couch in an office setting.

Sanjana Tripathy
NextWork Student

nextwork.org

CodeArtifact Security

Issue

To access CodeArtifact, we will need proper IAM permissions. I will address an error when retrieving a token because, by default, the EC2 instance won't have permission, following AWS's principle of least privilege for secure access.

Resolution

To resolve the error with my security token, I will attach the IAM role with the correct policy to my EC2 instance. This will resolve the error because the instance will then have the required permissions to access CodeArtifact securely.

Using IAM roles is considered a security best practice because they will allow EC2 instances and other resources to access AWS services without embedding long-term credentials, ensuring secure, least-privilege access.

The JSON policy attached to my role

The JSON policy I will set up grants my EC2 instance permissions to access CodeArtifact. It will allow retrieving authorization tokens, repository endpoints, and reading packages, while securely restricting STS actions to CodeArtifact only.

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```
1▼ {
2    "Version": "2012-10-17",
3    "Statement": [
4        {
5            "Effect": "Allow",
6            "Action": [
7                "codeartifact:GetAuthorizationToken",
8                "codeartifact:GetRepositoryEndpoint",
9                "codeartifact:ReadFromRepository"
10           ],
11           "Resource": "*"
12       },
13       {
14           "Effect": "Allow",
15           "Action": "sts:GetServiceBearerToken",
16           "Resource": "*",
17           "Condition": {
18               "StringEquals": {
19                   "sts:AWSServiceName": "codeartifact.amazonaws.com"
20               }
21           }
22       }
23   ]
24 }
```

Maven and CodeArtifact

To test the connection between Maven and CodeArtifact, I compiled my web app using settings.xml

The settings.xml file will configure Maven to authenticate with CodeArtifact because it stores the repository URL, authorization token, and credentials, ensuring Maven can securely download and manage our project dependencies.

Compiling means transforming human-readable Java source code into bytecode because the JVM can only execute bytecode. This process ensures our application runs smoothly on any system with a Java Virtual Machine.

```
index.jsp settings.xml U
src > main > webapp > settings.xml
  1  <settings>
  2  <servers>
  3  <profiles>
  4  <mirrors>
  5    <mirror>
  6      <id>nextwork-nextwork-devops-cicd</id>
  7      <name>nextwork-nextwork-devops-cicd</name>
  8      <url>https://nextwork-794038222820.d.codeartifact.ap-south-1.amazonaws.com/maven/nextwork-devops-cicd/</url>
  9      <mirrorOf>*</mirrorOf>
 10    </mirror>
 11  </mirrors>
 12  <profile>
 13    <id>nextwork-nextwork-devops-cicd</id>
 14    <activation>
 15      <activeByDefault>true</activeByDefault>
 16    </activation>
 17    <repositories>
 18      <repository>
 19        <id>nextwork-nextwork-devops-cicd</id>
 20        <url>https://nextwork-794038222820.d.codeartifact.ap-south-1.amazonaws.com/maven/nextwork-devops-cicd/</url>
 21      </repository>
 22    </repositories>
 23  </profile>
 24 </profiles>
 25 <server>
 26   <id>nextwork-nextwork-devops-cicd</id>
 27   <username>aws</username>
 28   <password>${env.CODEARTIFACT_AUTH_TOKEN}</password>
 29 </server>
 30 </servers>
 31 </settings>
```

Sanjana Tripathy
NextWork Student

nextwork.org

Verify Connection

After compiling, I checked my CodeArtifact repository because it stores project packages. I noticed that the web app's dependencies had been uploaded successfully, confirming that the setup and connection worked as expected.

The screenshot shows the AWS CodeArtifact interface. At the top, there are navigation links: Developer Tools > CodeArtifact > Repositories > nextwork-devops-cicd. Below this, there is a 'Details' section with a sub-section 'Packages'. A table lists one package:

Package name	Namespace	Format	Latest version	Latest publish date	Publish	Upstream
nextwork-web-project	com.nextwork.app	maven	1.0-SNAPSHOT	15 hours ago	Allow	Block



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

