



Secure Packages with CodeArtifact



shilpa kale

Successfully created nextwork-devops-cicd in nextwork. Configure your package manager to install packages from nextwork-devops-cicd. [View connection instructions](#)

Developer Tools > CodeArtifact > Repositories > nextwork-devops-cicd

nextwork-devops-cicd [Info](#) [Delete repository](#) [Apply repository policy](#) [Edit](#)

Repository This repository stores packages related to a Java web app created as a part of NextWork's CI/CD Pipeline series.

Details Domain, policy, tags, ARN, and upstream repositories.

Packages [Info](#) [C](#) [Delete package](#) [View connection instructions](#)

Filter by package name prefix, format, namespace prefix, and origin controls

Package name	Namespace	Format	Latest version	Latest publish date	Publish	Upstream
backport-util-concurrent	backport-util-concurrent	maven	3.1	13 minutes ago	Block	Allow
classworlds	classworlds	maven	1.1	14 minutes ago	Block	Allow
google	com.google	maven	1	13 minutes ago	Block	Allow

Introducing Today's Project!

In this project, we will demonstrate how to setup CodeArtifact as our CI/CD pipeline's artifact repository. We're doing this project to learn the importance and value of having an artifact repository. It's a huge time saver.

Key tools and concepts

Services we used were CodeArtifact, EC2, GitHub, VS Code, IAM. Key concepts we learnt include using artifact repositories, connecting Maven with CodeArtifact, setting up IAM permissions to give our EC2 instance.

Project reflection

This project took us approximately 2 hrs including documentation time. The most challenging part was connecting Maven with CodeArtifact. It was most rewarding to see CodeArtifact fill up with heaps of packages when the connection was all setup.

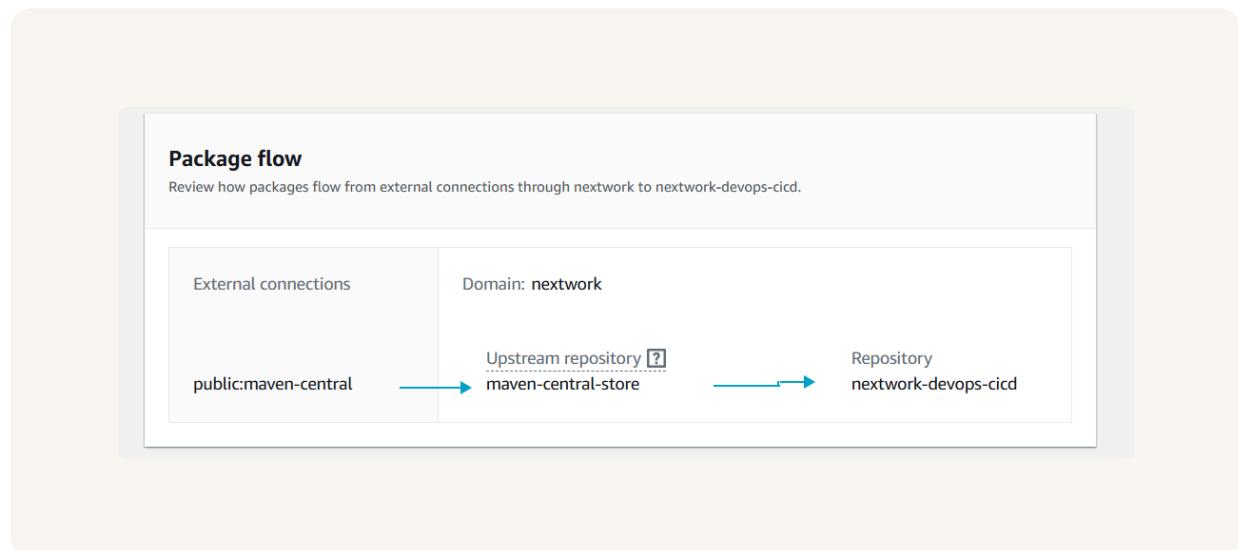
This project is part three of a series of DevOps projects where I'm building a CI/CD pipeline! We'll be working on the next project tomorrow.

CodeArtifact Repository

CodeArtifact is an artifact repository service, which we can use to create repositories to store our web apps packages and dependencies. Engineering teams use artifact repositories because they provide security, reliability and control.

A domain is like a folder that holds together multiple repositories under the same project/organization. They are helpful for setting up permissions for multiple CodeArtifact repository in one go. Our domain is called `vnextwork`.

A CodeArtifact repository can have an upstream repository, which means a public source of packages, if they can't find it in our local CodeArtifact repository. Our repository's upstream repository is Maven Central Store, largest java repository.



CodeArtifact Security

Issue

To access CodeArtifact, we need an authorization token that allows our EC2 instance to have access to CodeArtifact for next 12 hours. We ran into an error when retrieving a token because it does not have permission to access AWS resources by default.

Resolution

To resolve the error with our security token, we setup an IAM policy that grants access to CodeArtifact and then an IAM role can be attached to our EC2 instance. This resolved the error because our EC2 instance now has access to request token.

It's security best practice to use IAM roles because IAM roles are more secure and scalable. Hardcoded credentials are much more vulnerable to security attacks resulting in errors, losses, delays. Here we have IAM roles which are applied via AWS.

The JSON policy attached to my role

The JSON policy we set up grants access to CodeArtifact, specifically retrieving authorization token, finding the repository endpoint and viewing the packages inside the repository.

Policy editor Vis

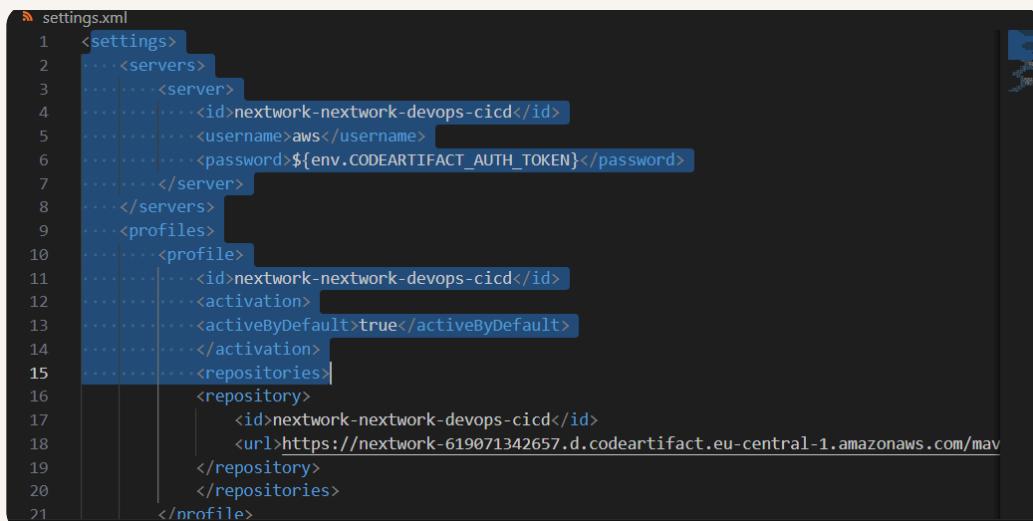
```
1▼ {
2    "Version": "2012-10-17",
3▼   "Statement": [
4▼     {
5       "Effect": "Allow",
6▼         "Action": [
7           "codeartifact:GetAuthorizationToken",
8           "codeartifact:GetRepositoryEndpoint",
9           "codeartifact:ReadFromRepository"
10      ],
11      "Resource": "*"
12    },
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 }
25 |
```

Maven and CodeArtifact

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

The settings.xml file configures Maven to use our CodeArtifact repository. It supplies Maven with authorization token to get access to CodeArtifact repository and also sets up a profile section in case we have multiple repositories.

Compiling means translating our source code into machine language. Maven is our complier. While compiling, maven will need packages in order to run our webapp code. To retrieve these packages maven visits CodeArtifact who then request the upstream.



```
settings.xml
1 <settings>
2   <servers>
3     <server>
4       <id>nextwork-nextwork-devops-cicd</id>
5       <username>aws</username>
6       <password>${env.CODEARTIFACT_AUTH_TOKEN}</password>
7     </server>
8   </servers>
9   <profiles>
10    <profile>
11      <id>nextwork-nextwork-devops-cicd</id>
12      <activation>
13        <activeByDefault>true</activeByDefault>
14      </activation>
15      <repositories>
16        <repository>
17          <id>nextwork-nextwork-devops-cicd</id>
18          <url>https://nextwork-619071342657.d.codeartifact.eu-central-1.amazonaws.com/mav
19        </repository>
20      </repositories>
21    </profile>
```

Verify Connection

After compiling, we checked our CodeArtifact repository and we noticed 4 pages of packages inside. This tells us that we successfully stored our webapps dependencies in a artifact repository.

The screenshot shows the CodeArtifact repository interface. At the top, a green banner displays a success message: "Successfully created nextwork-devops-cicd in nextwork. Configure your package manager to install packages from nextwork-devops-cicd." with a "View connection instructions" button. Below the banner, the repository name "nextwork-devops-cicd" is shown with an "info" link, and three buttons: "Delete repository", "Apply repository policy", and "Edit". A note below the name states: "Repository This repository stores packages related to a Java web app created as a part of NextWork's CI/CD Pipeline series." Under the "Details" section, there is a table of packages:

Package name	Namespace	Format	Latest version	Latest publish date	Publish	Upstream
backport-util-concurrent	backport-util-concurrent	maven	3.1	13 minutes ago	Block	Allow
classworlds	classworlds	maven	1.1	14 minutes ago	Block	Allow
google	com.google	maven	1	13 minutes ago	Block	Allow



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

