



[nextwork.org](http://nextwork.org)

# Secure Packages with CodeArtifact

A

Amirul Zaol-kefli

Packages <small>Info</small>						
<input type="text"/> 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	1 minute ago	Block	Allow
classworlds	classworlds	maven	1.1	1 minute ago	Block	Allow
google	com.google	maven	1	1 minute ago	Block	Allow
jnr305	com.google.code.findbug s	maven	2.0.1	1 minute ago	Block	Allow
google-collections	com.google.collections	maven	1.0	1 minute ago	Block	Allow
commons-cli	commons-cli	maven	1.0	1 minute ago	Block	Allow
commons-logging-api	commons-logging	maven	1.1	1 minute ago	Block	Allow
junit	junit	maven	3.8.2	1 minute ago	Block	Allow
log4j	log4j	maven	1.2.12	1 minute ago	Block	Allow
apache	org.apache	maven	13	1 minute ago	Block	Allow
maven	org.apache.maven	maven	2.2.1	1 minute ago	Block	Allow
maven-artifact	org.apache.maven	maven	2.2.1	1 minute ago	Block	Allow
maven-artifact-manager	org.apache.maven	maven	2.2.1	1 minute ago	Block	Allow
maven-core	org.apache.maven	maven	2.2.1	1 minute ago	Block	Allow

# Introducing Today's Project!

In this project, I will demonstrate how to setup an AWS CodeArtifact Repo. I'm doing this project to learn how to store and get safe and secure packages from AWS CodeArtifact.

## Key tools and concepts

Services I used were CodeArtifact and IAM Roles. Key concepts I learnt include giving EC2 instance IAM Role and permission to access CodeArtifact.

## Project reflection

This project took me approximately 1.5 hours. The most challenging part was to access the CodeArtifact because I couldn't access it with free tier account. It was most rewarding to see all the packages stored in the CodeArtifact.

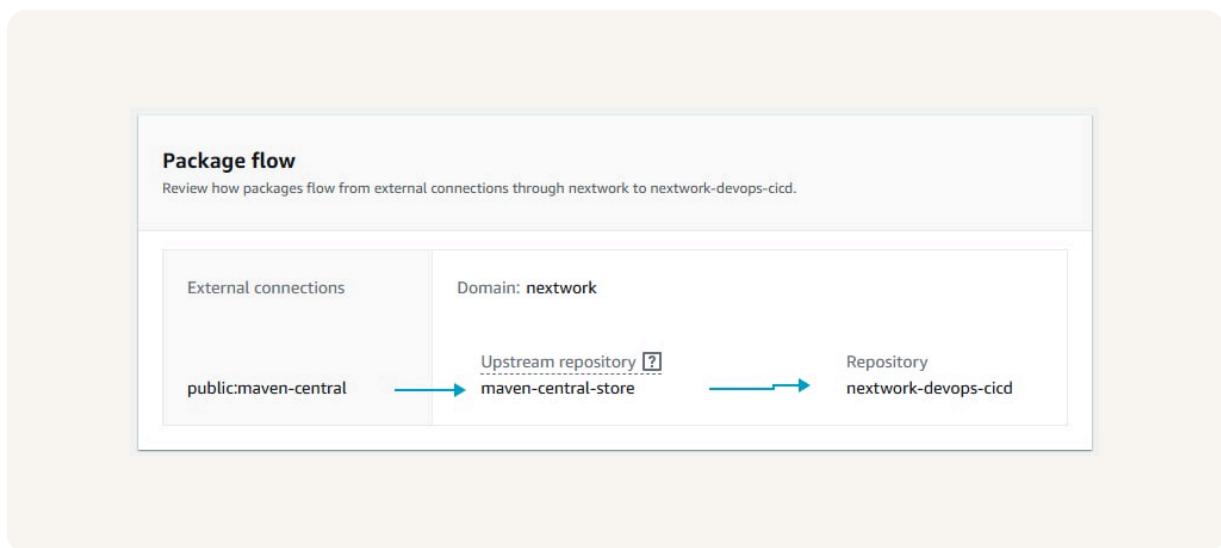
This project is part three of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project next Thursday.

# CodeArtifact Repository

CodeArtifact is secure, central place to store all your software packages. Engineering teams use artifact repositories because it is secure, reliable, and easy to control.

A domain is a central management construct that aggregates and organises multiple CodeArtifact repositories. My domain is nextwork.

A CodeArtifact repository can have an upstream repository, which means my primary repository can access backup libraries when it doesn't have what you need. My repository's upstream repository is Maven Central, the App Store of the Java world.



# CodeArtifact Security

## Issue

To access CodeArtifact, we need to get a authorization token for the repo. I ran into an error when retrieving a token because I do not have access to the repo yet.

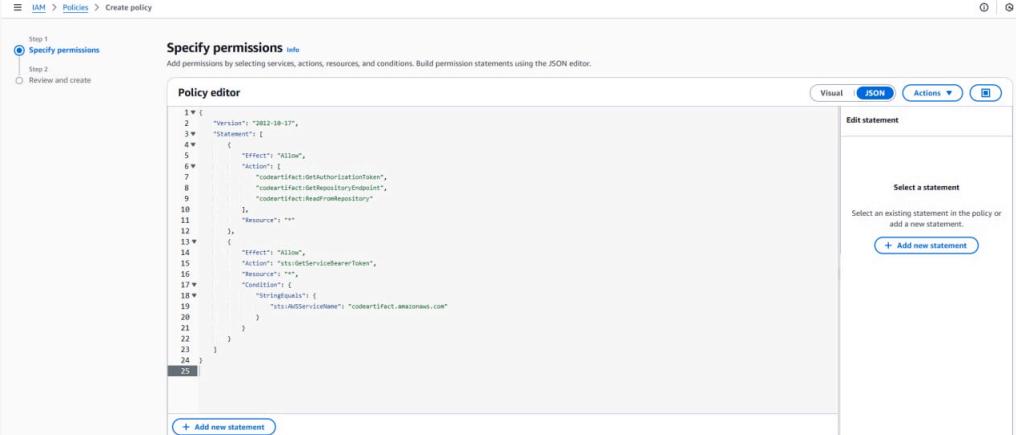
## Resolution

To resolve the error with my security token, I gave the EC2 instance an IAM Role. This resolved the error because the IAM role is the one that gives the EC2 instance permission to access the CodeArtifact Repo.

It's security best practice to use IAM roles because attaching the IAM role to our EC2 instance is what actually grants the instance the permissions defined in the policy, enabling secure access to CodeArtifact.

# The JSON policy attached to my role

The JSON policy I set up grants my IAM access to the CodeArtifact Repo, and it is necessary in order for my EC2 instance to use the other AWS features.



The screenshot shows the AWS IAM Policy Editor interface. The top navigation bar includes 'IAM', 'Policies', and 'Create policy'. Below this, a progress bar indicates 'Step 1 Specify permissions' is selected, with 'Step 2 Review and create' available. The main area is titled 'Specify permissions' with a 'JSON' tab selected. A code editor displays the following JSON policy:

```
1 ┌ {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "codeartifact:GetAuthorizationToken",
8         "codeartifact:DescribeRepositoryEndpoint",
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 |
```

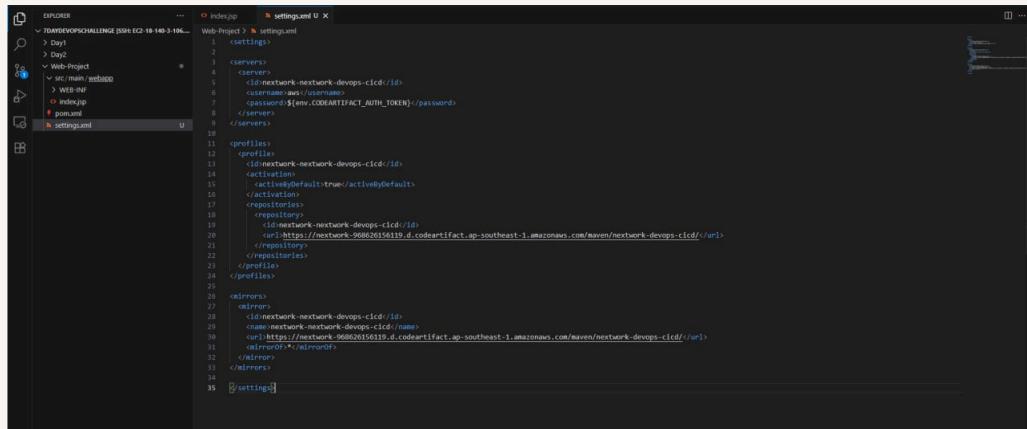
At the bottom of the code editor is a button labeled '+ Add new statement'. To the right of the code editor is a sidebar titled 'Edit statement' with a sub-section 'Select a statement' and a button '+ Add new statement'.

# 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 look into the CodeArtifact repo first before downloading any dependency and how to authenticate itself.

Compiling means getting all the dependencies from the CodeArtifact instead of from a public repo.



The screenshot shows the Eclipse IDE interface with the 'Web-Project' selected in the 'EXPLORER' view. The 'settings.xml' file is open in the 'EDITOR' view, displaying the XML configuration for Maven. The code includes details about servers, profiles, and mirrors, specifically pointing to a CodeArtifact repository named 'nextwork-devops-clid' located at 'https://nextwork-90002015619.d.codeartifact.ap-southeast-1.amazonaws.com/maven/nextwork-devops-clid/'. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings>
    <servers>
        <server>
            <id>nextwork-nextwork-devops-clid</id>
            <username>aws</username>
            <password><![CDATA[${env.CODEARTIFACT_AUTH_TOKEN}]]></password>
        </server>
    </servers>
    <profiles>
        <profile>
            <id>nextwork-nextwork-devops-clid</id>
            <activation>
                <activeByDefault>true</activeByDefault>
            </activation>
            <repositories>
                <repository>
                    <id>nextwork-nextwork-devops-clid</id>
                    <url>https://nextwork-90002015619.d.codeartifact.ap-southeast-1.amazonaws.com/maven/nextwork-devops-clid/</url>
                </repository>
            </repositories>
        </profile>
    </profiles>
    <mirrors>
        <mirror>
            <id>nextwork-nextwork-devops-clid</id>
            <name>nextwork-nextwork-devops-clid</name>
            <url>https://nextwork-90002015619.d.codeartifact.ap-southeast-1.amazonaws.com/maven/nextwork-devops-clid/</url>
        </mirror>
    </mirrors>
</settings>
```

# Verify Connection

After compiling, I checked the nextwork-devops-cicd repo. I noticed that there are multiple different packages, such as google, etc. They are all in maven format.

Packages		Info					
		Filter by package name prefix, format, namespace prefix, and origin controls				Delete package	View connection instructions
	Package name	Namespace	Format	Latest version	Latest publish date	Publish	Upstream
○	backport-util-concurrent	backport-util-concurrent	maven	3.1	1 minute ago	Block	Allow
○	classworlds	classworlds	maven	1.1	1 minute ago	Block	Allow
○	google	com.google	maven	1	1 minute ago	Block	Allow
○	jnr305	com.google.code.findbug.s	maven	2.0.1	1 minute ago	Block	Allow
○	google-collections	com.google.collections	maven	1.0	1 minute ago	Block	Allow
○	commons-cli	commons-cli	maven	1.0	1 minute ago	Block	Allow
○	commons-logging-api	commons-logging	maven	1.1	1 minute ago	Block	Allow
○	junit	junit	maven	3.8.2	1 minute ago	Block	Allow
○	log4j	log4j	maven	1.2.12	1 minute ago	Block	Allow
○	apache	org.apache	maven	13	1 minute ago	Block	Allow
○	maven	org.apache.maven	maven	2.2.1	1 minute ago	Block	Allow
○	maven-artifact	org.apache.maven	maven	2.2.1	1 minute ago	Block	Allow
○	maven-artifact-manager	org.apache.maven	maven	2.2.1	1 minute ago	Block	Allow
○	maven-core	org.apache.maven	maven	2.2.1	1 minute ago	Block	Allow



[nextwork.org](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

