



NextWork.org

# Dependencies and CodeArtifact



sru anu

nextwork-packages [Info](#)

[Delete repository](#) [Apply repository policy](#) [Edit](#)

[Repository](#) nextwork-packages

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

**Packages** [Info](#)

[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	P
<a href="#">...</a>	backport-util-concurrent	backport-util-concurrent	maven	3.1	2 minutes ago	
<a href="#">...</a>	classworlds	classworlds	maven	1.1	2 minutes ago	
<a href="#">...</a>	google	com.google	maven	1	2 minutes ago	
<a href="#">...</a>	jsr305	com.google.code.findbugs	maven	2.0.1	2 minutes ago	
<a href="#">...</a>	google-collections	com.google.collectio ns	maven	1.0	2 minutes ago	



# Introducing today's project!

## What is AWS CodeArtifact?

AWS CodeArtifact is a managed artifact repository that simplifies storing, publishing, and sharing software packages. It supports various formats, enhancing dependency management and team collaboration.

## How I used CodeArtifact in this project

In today's project, I used AWS CodeArtifact to create a secure repository for my web app's dependencies, enabling easy management and access for my team while ensuring efficient collaboration and version control.

## One thing I didn't expect in this project was...

One unexpected aspect of this project was the complexity of configuring AWS CodeArtifact, which required careful attention to IAM permissions and repository settings to ensure proper access and functionality.

## This project took me...

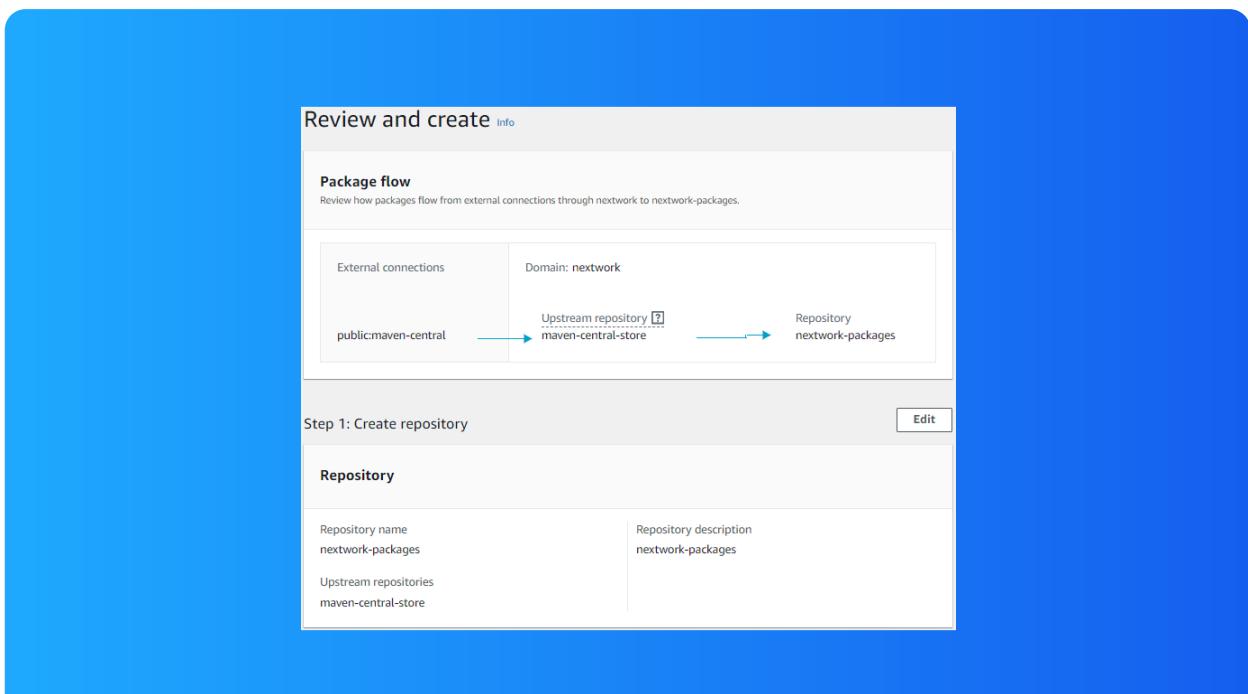
This project took me approximately two hours, primarily due to the time spent configuring AWS services and ensuring everything was set up correctly for smooth operation.

## My project has three artifact repositories

The local repository is a version-controlled folder on your local machine where Git tracks all changes to your files and allows you to commit and manage your project before pushing it to a remote repository like GitHub or AWS CodeCommit.

The upstream repository is maven-central-store, which acts as the main repository where updates and changes are fetched from. It is used to synchronize and pull the latest changes from the original repository into your local project.

The public repository is nextwork-packages, and it stores and shares the project packages publicly. It allows users to access and use the project's resources, making them available for collaboration or general use.



# Connecting my project with CodeArtifact

I connected my web app project (via my Cloud9 IDE) to CodeArtifact to manage dependencies easily, securely store artifacts, and ensure seamless integration with other AWS services for better package versioning and sharing.

## I created a new file, `settings.xml`, in my web app

`settings.xml` is a Maven config file that stores repository URLs, authentication tokens, and rules for accessing dependencies. It defines servers, profiles, and mirrors to control how Maven connects to remote repositories like CodeArtifact.

The snippets in `settings.xml` configure Maven for AWS CodeArtifact. The servers section contains auth details, profiles specify the primary repository, and mirrors set backup locations, ensuring seamless access to project dependencies.

```
PS C:\Users\SRUJANI\SAI> ssh -l "project5 (1).pm" ec2-user@ec2-65-0-250-59.ap-south-1.compute.amazonaws.com
[sudo] nemo 5.8
[sudo] password: settings.xml

<servers>
  <server>
    <id>nextwork-nextwork-packages</id>
    <username>nemo</username>
    <password>${env.CODEARTIFACT_AUTH_TOKEN}</password>
  </server>
</servers>

<profiles>
  <profile>
    <id>nextwork-nextwork-packages</id>
    <activation>
      <activeByDefault>true</activeByDefault>
    </activation>
    <repositories>
      <repository>
        <id>nextwork-nextwork-packages</id>
        <url>https://nextwork-782256755529.d.codeartifact.ap-south-1.amazonaws.com/maven/nextwork-packages/</url>
      </repository>
    </repositories>
  </profile>
</profiles>

<mirrors>
</mirrors>
```

# Testing the connection

## To test the connection between Cloud9 and CodeArtifact, I compiled my web app

Compiling means translating your project's code into a format that computers can understand and execute. It ensures that the code is correctly structured and prepares it to function as a working application.

## Success!

After compiling, I checked the **\*\*nextwork-packages\*\*** repository, and I saw that all the packages were successfully added, which were collected from the Maven upstream repository.

The screenshot shows the AWS CodeArtifact console. At the top, there is a navigation bar with 'nextwork-packages' and 'Info' buttons, and three buttons: 'Delete repository', 'Apply repository policy', and 'Edit'. Below this is a 'Details' section with a 'Repository' tab selected, showing the name 'nextwork-packages'. Underneath is a 'Packages' section with a 'Info' button, a search bar, and a 'View connection instructions' button. A table lists several packages:

Package name	Namespace	Format	Latest version	Latest publish date	P
backport-util-concurrent	backport-util-concurrent	maven	3.1	2 minutes ago	BL
classworlds	classworlds	maven	1.1	2 minutes ago	BL
google	com.google	maven	1	2 minutes ago	BL
jsr305	com.google.code.findbugs	maven	2.0.1	2 minutes ago	BL
google-collections	com.google.collections	maven	1.0	2 minutes ago	BL

# Create IAM policies

## The importance of IAM policies

I also created an IAM policy because I wanted to grant access to other DevOps services, like AWS CodeBuild and AWS CodePipeline, as well as my development environments (like Cloud9), to utilize the packages stored in CodeArtifact for future projects.

## I defined my IAM policy using JSON

This policy will allow actions such as retrieving, publishing, and managing packages stored in CodeArtifact. It ensures that services like AWS CodeBuild, AWS CodePipeline, and my Cloud9 environment can seamlessly access and utilize the dependency bac

Policy editor

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



NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

