



NextWork.org

Dependencies and CodeArtifact



Shivam Kakade

The screenshot shows the AWS CodeArtifact interface in a web browser. The left sidebar has a tree view with 'Source + CodeCommit', 'Artifacts + CodeArtifact' (selected), 'Build + CodeBuild', 'Deploy + CodeDeploy', 'Pipeline + CodePipeline', and 'Settings'. Under 'Artifacts + CodeArtifact', 'Repositories' is expanded, and 'nextwork-packages' is selected. The main content area shows the 'nextwork-packages' repository details, including its ARN and policy. Below this 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	1 minute ago	Block	Allow
classworlds	classworlds	maven	1.1	1 minute ago	Block	Allow
google	com.google	maven	1	Just now	Block	Allow



Introducing today's project!

What is AWS CodeArtifact?

AWS CodeArtifact is a fully managed artifact repository service that securely stores, publishes, and shares software packages. It supports multiple formats, helping teams manage dependencies, ensure version control, and streamline collaboration.

How I used CodeArtifact in this project

In today's project, I used AWS CodeArtifact to securely store and manage dependencies. I connected my web app to CodeArtifact for easy access to both public and private packages, ensuring smooth dependency management during development.

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

One thing I didn't expect in this project was the complexity of managing dependencies through AWS CodeArtifact, especially configuring IAM policies and repository access, which required careful setup for seamless integration with my web app.

This project took me...

This project took me approximately 30 minutes to complete, including the setup of AWS CodeArtifact, configuring dependencies, and integrating it with my web app for smooth development and deployment.

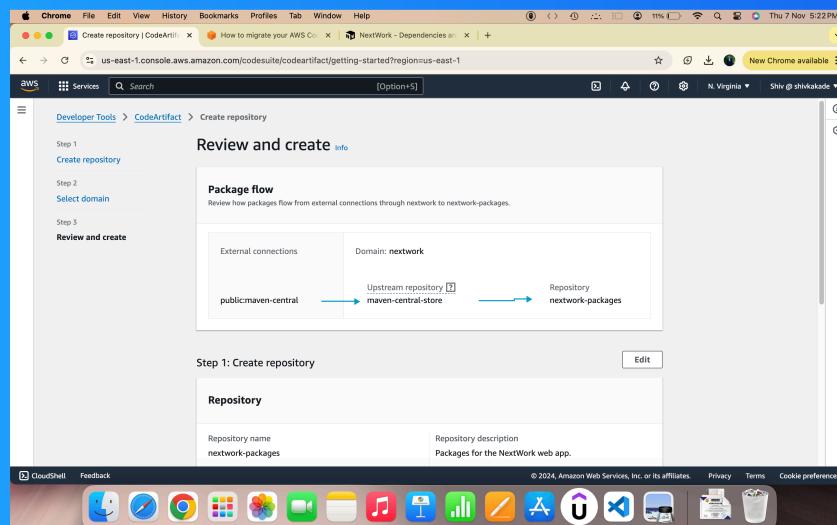


My project has three artifact repositories

A local repository is a private storage for software packages within your AWS account. It enables you to securely manage dependencies and artifacts, supports multiple formats (e.g., npm, Maven), and can connect to public repositories.

The upstream repository is an external source that a local repository connects to for fetching packages not stored locally. It ensures broader access to dependencies, allowing seamless package retrieval and caching for project needs.

The public repository is a repository that hosts open-source packages and is accessible to everyone. It allows developers to share and download packages without restrictions, enabling easy access to widely used libraries and tools.





Shivam Kakade

NextWork Student

NextWork.org

Connecting my project with CodeArtifact

I connected my web app project (via my Cloud9 IDE) to CodeArtifact to securely manage and store project dependencies, ensuring seamless access to both public and private packages, and facilitating better version control and artifact sharing.

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

`settings.xml` is a configuration file used by Maven to define repository information, proxy settings, build profiles, and credentials. It customizes how Maven interacts with repositories, ensuring correct artifact retrieval and build execution.

The snippets of code in `settings.xml` configure Maven's behavior, such as defining repository locations, proxy settings, and credentials. They ensure MVN can access required repositories, authenticate properly, & download dependencies for the project.



Testing the connection

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

Compiling means converting source code written in a programming language (e.g., Java, C++) into machine-readable code, usually in the form of bytecode or executable files, so the program can be run on a computer or server.

Success!

After compiling, I checked the CodeArtifact repository. I saw the compiled artifacts, including dependencies and package versions, stored and ready for use in the project, ensuring they could be accessed and shared across environments.

The screenshot shows a browser window with multiple tabs open. The main content is the AWS CodeArtifact interface. On the left, there's a sidebar with 'Developer Tools' and 'CodeArtifact' selected. Under 'Artifacts > CodeArtifact', 'Repositories' is expanded, showing 'nextwork-packages'. The 'Details' section shows the repository is for the 'nextwork' web app. Below it, the 'Packages' section lists three packages:

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	Just now	Block	Allow



Shivam Kakade
NextWork Student

NextWork.org

Create IAM policies

The importance of IAM policies

I also created an IAM policy because it allows me to define specific permissions for users or roles, ensuring controlled access to AWS resources like CodeArtifact, while following the principle of least privilege for enhanced security.

I defined my IAM policy using JSON

This policy will allow the actions: retrieving an authorization token for CodeArtifact, accessing repository endpoints, reading from repositories, & getting a service bearer token for CodeArtifact. It ensures authorized interactions with CodeArtifact

The screenshot shows the AWS IAM 'Create policy' interface. The title bar says 'Specify permissions'. The main area is titled 'Policy editor' with tabs for 'Visual', 'JSON' (which is selected), and 'Actions'. The JSON code is as follows:

```
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  }
```

Below the JSON editor is a sidebar with the heading 'Edit statement' and a button '+ Add new statement'.