



NextWork.org

Dependencies and CodeArtifact



Nikhil Bhan

nextwork-packages info

Repository Packages for the NextWork web app.

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

Packages info

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	Just now	Block	Allow
classworlds	classworlds	maven	1.1	1 minute ago	Block	Allow
google	com.google	maven	1	Just now	Block	Allow
jxr305	com.google.code.findbugs	maven	2.0.1	Just now	Block	Allow
google-collections	com.google.collections	maven	1.0	Just now	Block	Allow
commons-cli	commons-cli	maven	1.0	Just now	Block	Allow
commons-logging-api	commons-logging	maven	1.1	Just now	Block	Allow
junit	junit	maven	3.8.2	Just now	Block	Allow
log4j	log4j	maven	1.2.12	Just now	Block	Allow
apache	org.apache	maven	5	Just now	Block	Allow
maven	org.apache.maven	maven	2.2.1	Just now	Block	Allow



Introducing today's project!

What is AWS CodeArtifact?

AWS CodeArtifact is a highly scalable and secure managed repository service. Developers can use this service to store, organize and share their software packages for developing applications.

How I used CodeArtifact in this project

In this project I used AWS CodeArtifact to create repositories to download and store packages and dependencies that would be ready to be leveraged by my web application.

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

I wasn't expecting to get an issue with downloading packages to my upstream repository. I had to verify that my AWS services were set up correctly.

This project took me...

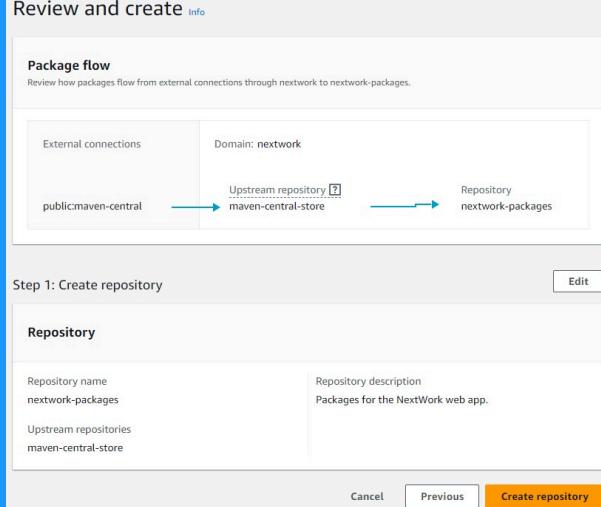
This project took me almost 2 hours to complete, with an additional 15 minutes to write my documentation. It took longer since I needed to troubleshoot my issue and find a solution.

My project has three artifact repositories

The local repository is a toolbox that's local on the computer and that is where all the software packages are installed.

The upstream repository is a location where your local repository will check for packages it needs if they're not on your local computer. The advantage of having an upstream repository is for convenience and fast access.

The public repository is a location that's available on the Internet and stores a vast amount of tools and packages. It's not convenient to access these tools and packages directly; it's used by your upstream repository if it needs to access it.



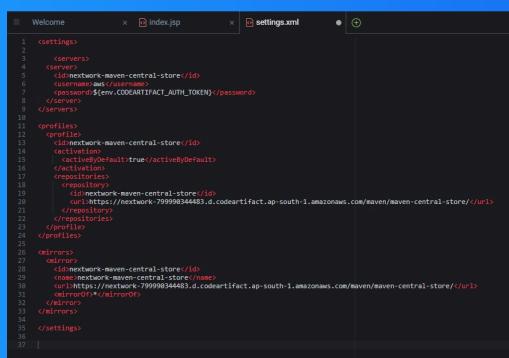
Connecting my project with CodeArtifact

I connected my web app project (via my Cloud9 IDE) to CodeArtifact so that it can locate and leverage the required packages for my local repository.

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

settings.xml is a file I created. It works as a guide and it shows the web app in Cloud9 where to find the dependencies and how to connect to the correct repositories.

The snippets of code have settings. The "server" setting provides the web app details to connect to the repositories. The "profile" setting determines when the repository is to be used. The "mirror" setting shows backup locations for packages.



```
<settings>
  <servers>
    <server id="nextwork-maven-central-store">
      <username>aws</username>
      <password>${env.CODEARTIFACT_AUTH_TOKEN}</password>
    </server>
  </servers>
  <profiles>
    <profile id="nextwork-maven-central-store">
      <activation><activeByDefault>true</activeByDefault></activation>
      <repositories>
        <repository id="maven-central-store">
          <url>https://nextwork-299998934483.d.codeartifact.ap-south-1.amazonaws.com/maven/maven-central-store</url>
        </repository>
      </repositories>
    </profile>
  </profiles>
  <mirrors>
    <mirror id="nextwork-maven-central-store">
      <name>nextwork-maven-central-store</name>
      <url>https://nextwork-299998934483.d.codeartifact.ap-south-1.amazonaws.com/maven/maven-central-store</url>
      <mirrorOf>*</mirrorOf>
    </mirror>
  </mirrors>
</settings>
```



Testing the connection

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

Compiling means converting your project's code into a language that computers can recognize, understand and run for you.

Success!

After compiling, I checked the nextwork-packages repository in CodeArtifact. I observed that this upstream repository now contains many packages that were fetched from the maven-central-store repository.

The screenshot shows the 'Details' tab of the 'nextwork-packages' repository in CodeArtifact. It displays a list of packages with their details such as namespace, format, latest version, publish date, and upstream status. The packages listed are: backport-util-concurrent, classworlds, google, jet305, google-collections, commons-cli, commons-logging-api, junit, log4j, apache, and maven. All packages are in Maven format, with the latest versions ranging from 1.0 to 3.1. The publish date is 'Just now' for most, except for log4j which was published 1 minute ago. The upstream status is 'Allow' for all packages.

Package name	Namespace	Format	Latest version	Latest publish date	Publish	Upstream
backport-util-concurrent	backport-util-concurrent	maven	3.1	Just now	Block	Allow
classworlds	classworlds	maven	1.1	1 minute ago	Block	Allow
google	com.google	maven	1	Just now	Block	Allow
jet305	com.google.code.findbugs	maven	2.0.1	Just now	Block	Allow
google-collections	com.google.collections	maven	1.0	Just now	Block	Allow
commons-cli	commons-cli	maven	1.0	Just now	Block	Allow
commons-logging-api	commons-logging	maven	1.1	Just now	Block	Allow
junit	junit	maven	3.8.2	Just now	Block	Allow
log4j	log4j	maven	1.2.12	Just now	Block	Allow
apache	org.apache	maven	5	Just now	Block	Allow
maven	org.apache.maven	maven	2.2.1	Just now	Block	Allow

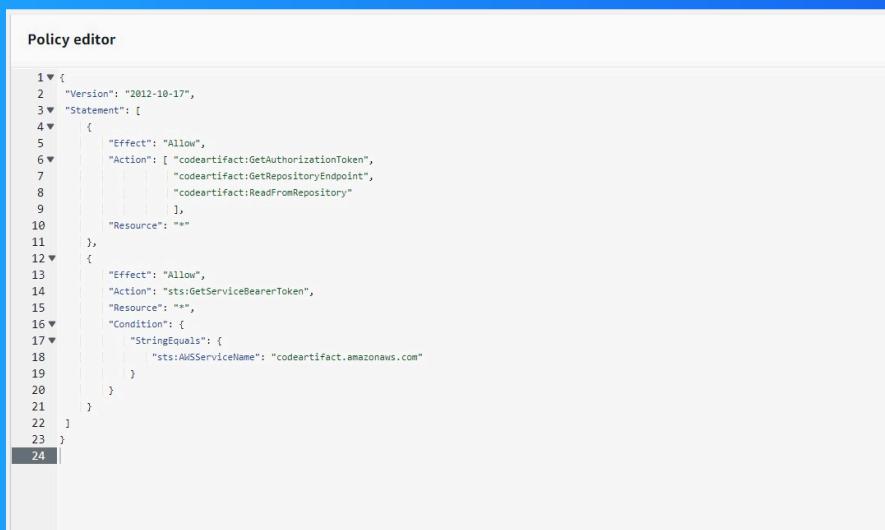
Create IAM policies

The importance of IAM policies

I also created an IAM policy because I'll need to use other AWS DevOps services that will need to access the resources in the CodeArtifact repository.

I defined my IAM policy using JSON

This policy will allow other AWS services to get temporary security credentials to interact with CodeArtifact, which will then allow them to gain authorization to see the repository and leverage its resources.



The screenshot shows the AWS Policy Editor interface with a JSON-based policy document. The policy defines two statements. The first statement allows actions on the CodeArtifact service (GetAuthorizationToken, GetRepositoryEndpoint, ReadFromRepository) on all resources with an effect of 'Allow'. The second statement allows the sts:GetServiceBearerToken action on all resources with an effect of 'Allow', subject to a condition where the sts:AWSServiceName must be 'codeartifact.amazonaws.com'.

```
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 }
```



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

