



NextWork.org

Dependencies and CodeArtifact



mallangiharinathreddy727@gmail.com

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	3 minutes ago	Block	Allow	
classworlds	classworlds	maven	1.1	3 minutes ago	Block	Allow	
google	com.google	maven	1	2 minutes ago	Block	Allow	
jor305	com.google.code.findbugs	maven	2.0.1	2 minutes ago	Block	Allow	
google-collections	com.google.collections	maven	1.0	2 minutes ago	Block	Allow	
commons-dli	commons-dli	maven	1.0	3 minutes ago	Block	Allow	
commons-logging-api	commons-logging	maven	1.1	2 minutes ago	Block	Allow	
junit	junit	maven	3.8.2	2 minutes ago	Block	Allow	
log4j	log4j	maven	1.2.12	2 minutes ago	Block	Allow	
apache	org.apache	maven	5	3 minutes ago	Block	Allow	
maven	org.apache.maven	maven	2.2.1	3 minutes ago	Block	Allow	
maven-artifact	org.apache.maven	maven	2.2.1	3 minutes ago	Block	Allow	
maven-artifact-manager	org.apache.maven	maven	2.2.1	3 minutes ago	Block	Allow	
maven-core	org.apache.maven	maven	2.2.1	3 minutes ago	Block	Allow	
maven-error-diagnostics	org.apache.maven	maven	2.2.1	3 minutes ago	Block	Allow	

Introducing today's project!

What is AWS CodeArtifact?

This is where AWS CodeArtifact steps in. Think of AWS CodeArtifact as a private storage locker for you to keep a backup copy of all your web app's dependencies. CodeArtifact adds a layer of reliability and security, protecting your development process.

How I used CodeArtifact in this project

In today's project, I used AWS CodeArtifact to create Domain and Repositories and also grabbed the packages from the Maven central repository.

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

one thing I didn't expect in this project was the installation of packages from the Maven.

This project took me...

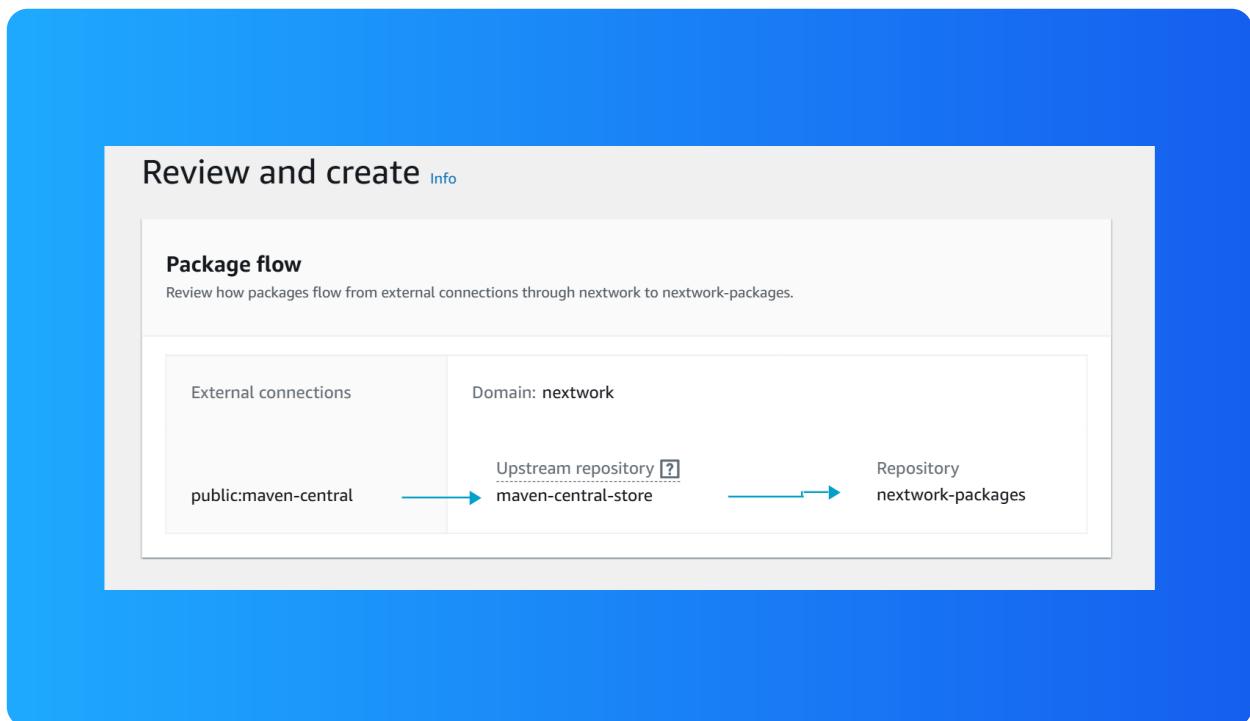
This project took me 2 Hours include documentation.

My project has three artifact repositories

The local repository, which Maven checks first for the packages/dependencies of my web app.

The upstream repository, which Maven will check next if a package is not in local repository.

The public repository with the greatest collection of packages for java applications. However, Maven will only visit this repository if the first two do not have the packages/dependencies it is looking for.



Connecting my project with CodeArtifact

I connected my web app project (via my VS CODE IDE) to CodeArtifact so that CodeArtifact knows which project it is going to storing dependencies for!

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

I created a new file, settings.xml, in my web app. settings.xml is the file that will give Maven instructions on WHERE to find the dependencies Maven will need to fetch, and HOW Maven will get access to these repositories.

The snippets of code stores authentication tokens to CodeArtifact and defines when Maven will visit which repository, plus where Maven should visit to find backup local repositories(optional).

```
settings.xml
1 <settings>
2   <servers>
3     <server>
4       <id>nextwork-nextwork-packages</id>
5       <username>aws</username>
6       <password>${env.CODEARTIFACT_AUTH_TOKEN}</password>
7     </server>
8   </servers>
9 
10  <profiles>
11    <profile>
12      <id>nextwork-nextwork-packages</id>
13      <activation>
14        | <activeByDefault>true</activeByDefault>
15      </activation>
16      <repositories>
17        <repository>
18          <id>nextwork-nextwork-packages</id>
19          <url>https://nextwork-908027415038.d.codeartifact.ap-south-1.amazonaws.com/maven/nextwork-packages/</url>
20        </repository>
21      </repositories>
22    </profile>
23  </profiles>
24
```

Testing the connection

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

Compiling is like translating your project's code into a language that computers can understand and run. When you compile your project, you're making sure everything is correctly set up and ready to turn into a working app.

Success!

After compiling, I checked my local repository and saw that my local repository now has pages and pages of packages inside! This means Maven has now grabbed packages from the Maven central repository and installed/kept a copy locally.

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	3 minutes ago	Block	Allow
classworlds	classworlds	maven	1.1	3 minutes ago	Block	Allow
google	com.google	maven	1	2 minutes ago	Block	Allow
jxr305	com.google.code.findbugs	maven	2.0.1	2 minutes ago	Block	Allow
google-collections	com.google.collections	maven	1.0	2 minutes ago	Block	Allow
commons-cli	commons-cli	maven	1.0	3 minutes ago	Block	Allow
commons-logging-api	commons-logging	maven	1.1	2 minutes ago	Block	Allow
junit	junit	maven	3.8.2	2 minutes ago	Block	Allow
log4j	log4j	maven	1.2.12	2 minutes ago	Block	Allow
apache	org.apache	maven	5	3 minutes ago	Block	Allow
maven	org.apache.maven	maven	2.2.1	3 minutes ago	Block	Allow
maven-artifact	org.apache.maven	maven	2.2.1	3 minutes ago	Block	Allow
maven-artifact-manager	org.apache.maven	maven	2.2.1	3 minutes ago	Block	Allow
maven-core	org.apache.maven	maven	2.2.1	3 minutes ago	Block	Allow
maven-error-diagnostics	org.apache.maven	maven	2.2.1	3 minutes ago	Block	Allow

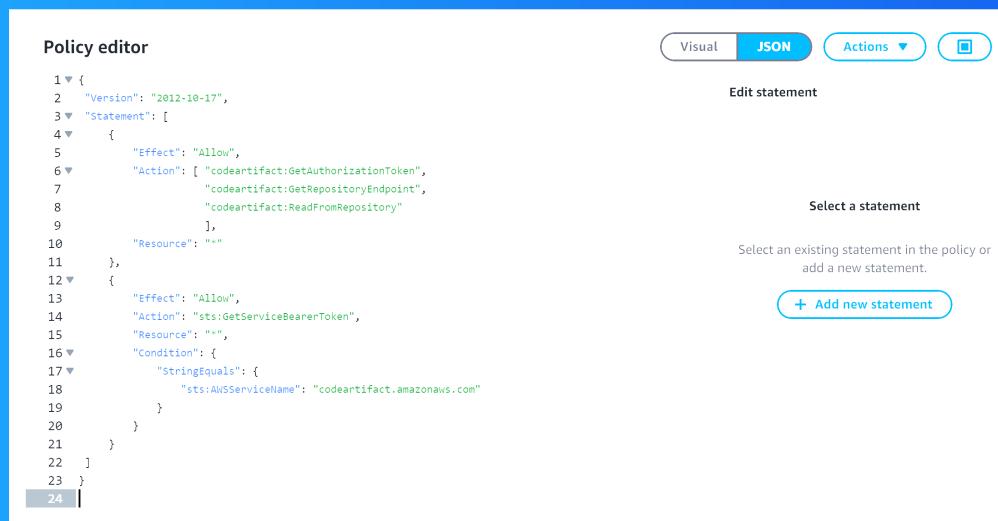
Create IAM policies

The importance of IAM policies

I also created an IAM policy because other services in my CI/CD pipeline e.g. CodeBuild, CodePipeline will be needing access to the packages/dependencies stored in CodeArtifact.

I defined my IAM policy using JSON

This policy will enable the policy holder to get authorization tokens (i.e. access to CodeArtifact), fetch the packages stored in CodeArtifact's repositories.



The screenshot shows the AWS IAM Policy editor interface. At the top, there are tabs for "Visual", "JSON" (which is selected), and "Actions". Below the tabs, there are buttons for "Edit statement", "Select a statement", and a button to "+ Add new statement". The main area contains the JSON code for the policy:

```
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

