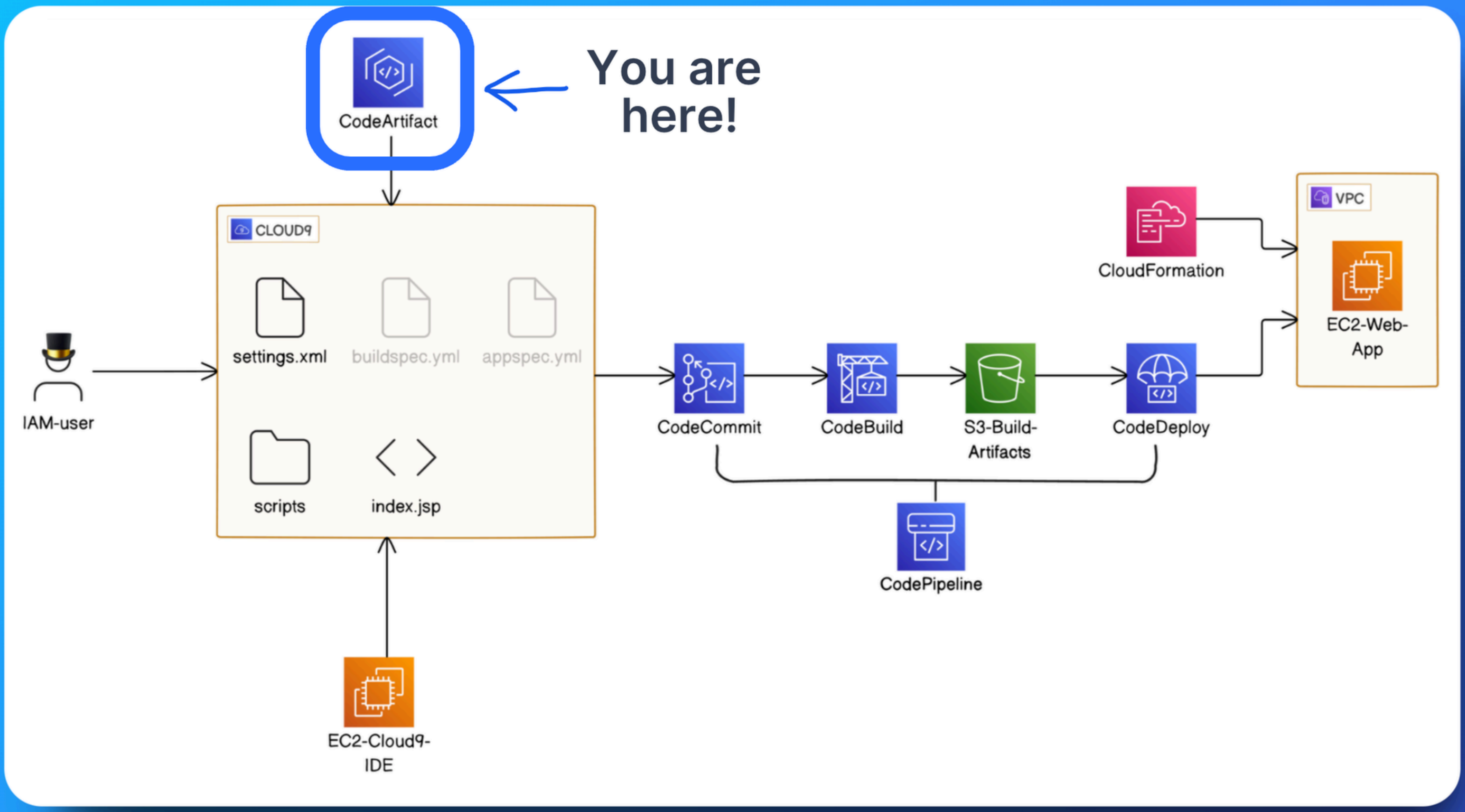
NextWork.org

# Secure dependencies with CodeArtifact

Rasha M
@Badry2022
Rasha M.



Rasha M
@Badry2022
Rasha M.

# Introducing AWS CodeArtifact!

## What it does & how it's useful

AWS Cloud9 is a cloud-based IDE that allows developers to write, run, and debug code with just a browser. It includes a code editor, terminal, and essential tools for working with programming languages and frameworks.
Artifact repositories are often used to share software packages for use in builds and deployments. Java developers using Apache Maven use artifact repositories to share and reuse Maven packages.

## How I'm using it in today's project

I'm using AWS CodeArtifact in this project to store and manage my web app's dependencies securely. It helps to ensure that my project's required packages are always available and protected from potential outages or disruptions in public repositories.

## This project took me...

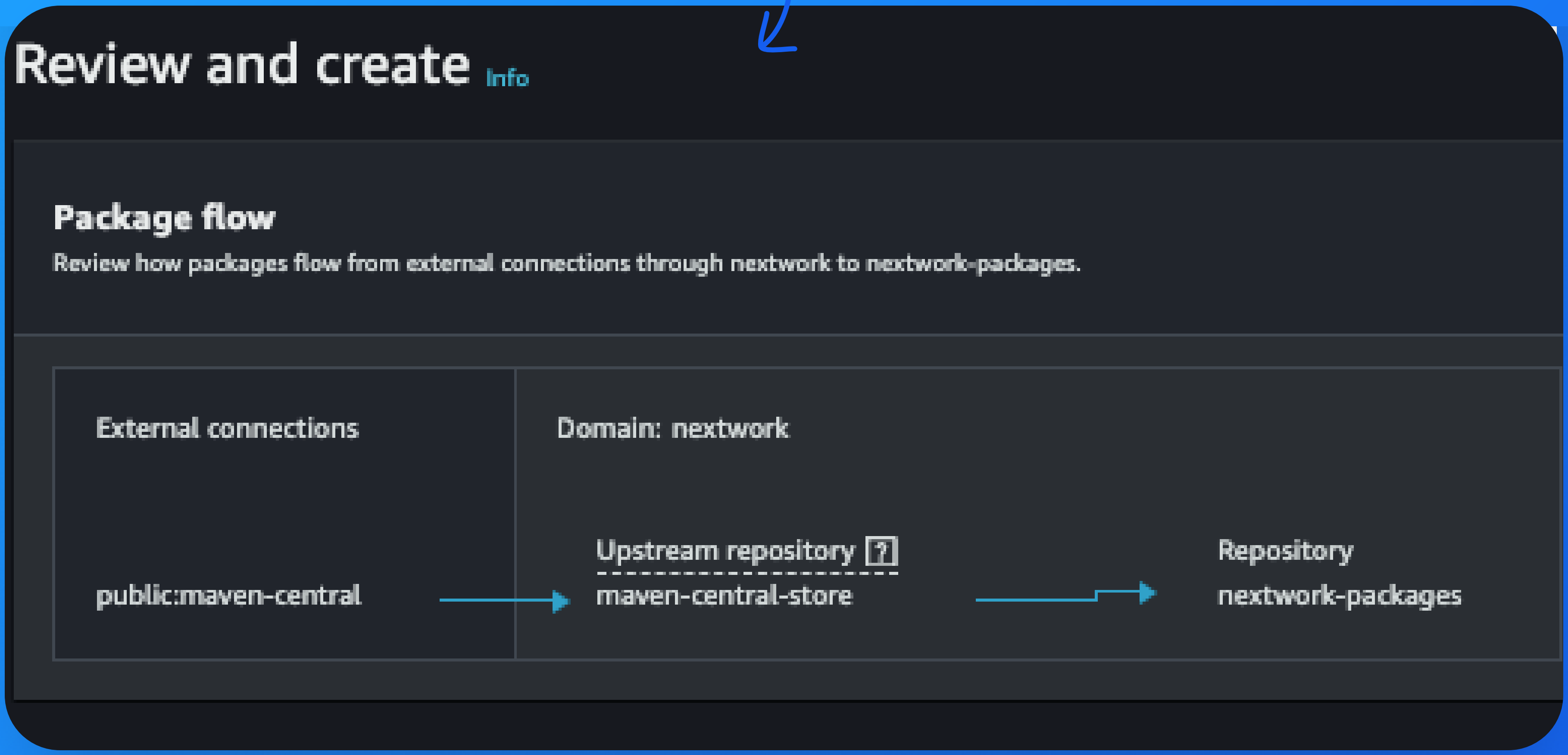
Documentation took me one hour.

Rasha M
@Badry2022
Rasha M.

## Create a repository

- CodeArtifact is a managed artifact repository service that makes it easy for developers to store and retrieve software packages and dependencies. It supports multiple package formats, including Maven, npm, and PyPI, and integrates seamlessly with other AWS services.
- I'm using CodeArtifact for my web app to ensure that my project's dependencies are reliably stored and available even if the public repositories come from experience issues. It helps to manage and secure my project's build and deployment processes.
- Instead of a single repository, there are actually three connected repositories that Maven uses to fetch packages.
  - Local Repository: This is where Maven first looks for packages that are already available locally. It helps in quick access and avoids redundant downloads.
  - Public Upstream Repository (maven-central-store): If the required packages are not found in the local repository, Maven checks the public upstream repository. This repository mirrors a vast collection of packages from the Maven Central Repository, providing a reliable source of packages.
  - Maven Central Repository: This is the central repository that Maven uses as a last resort if packages are not available in the local or upstream repositories. It is a comprehensive, public repository with a large number of packages available to all developers.

Package flow illustrating the connections between the three repositories.


Review and create Info

**Package flow**

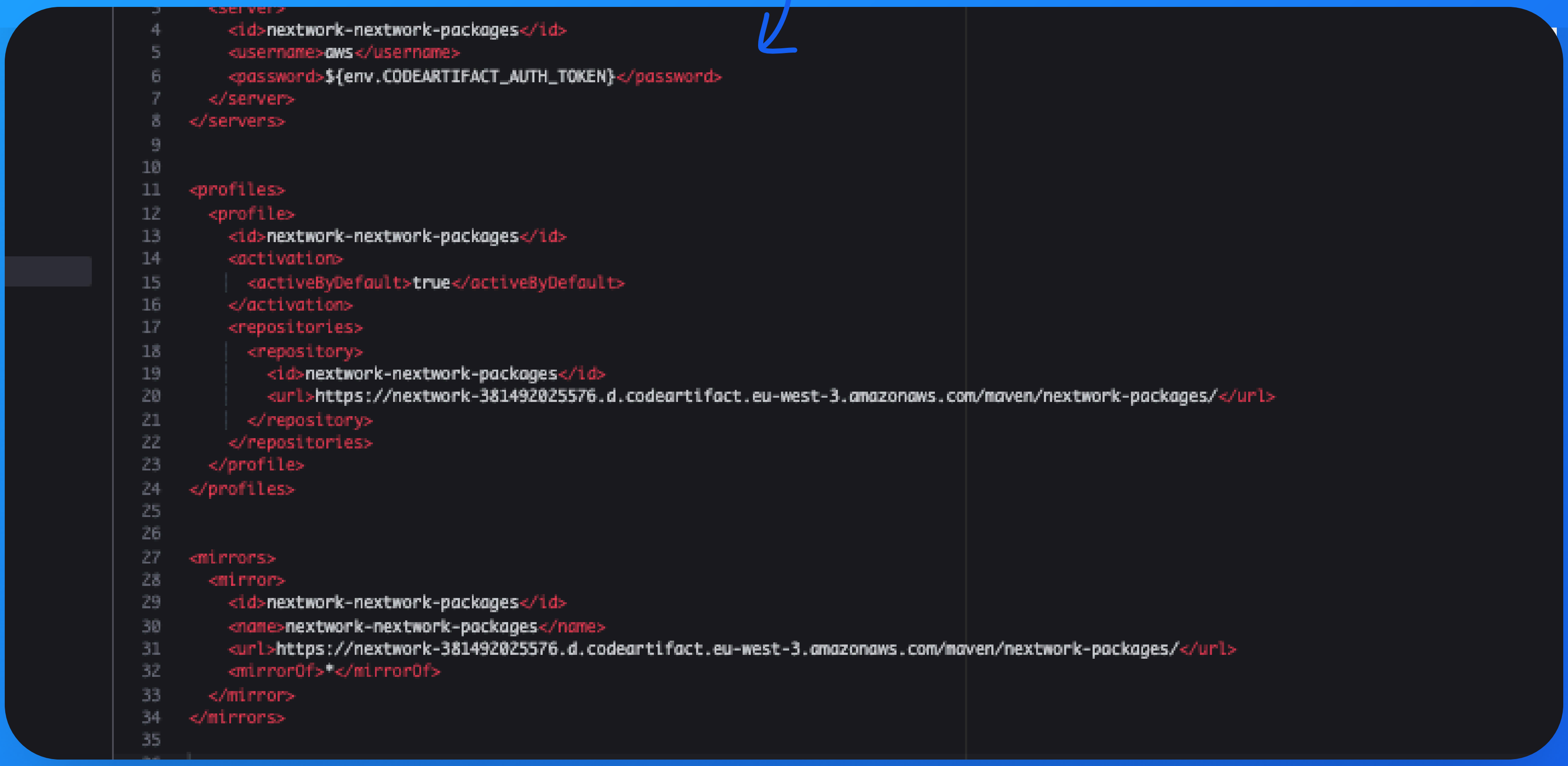
Review how packages flow from external connections through nextwork to nextwork-packages.

Rasha M
@Badry2022
Rasha M.

## Connecting my project to CodeArtifact

- Next, I connected my Cloud9 IDE to CodeArtifact so that my development environment could access and manage dependencies stored in the CodeArtifact repository. This setup ensures that my build process can retrieve necessary packages from CodeArtifact seamlessly.
- I created a new file, settings.xml, in my web app to configure Maven to use the CodeArtifact repository for fetching and storing dependencies.
- The code I pasted into settings.xml were provided by CodeArtifact, so I did not have to write from scratch. The snippets of code provided by CodeArtifact include repository URLs, authentication tokens, and configuration settings that allow Maven to properly connect to and interact with the CodeArtifact repositories. This setup ensures that Maven can securely and effectively fetch dependencies as needed.

My settings.xml file.



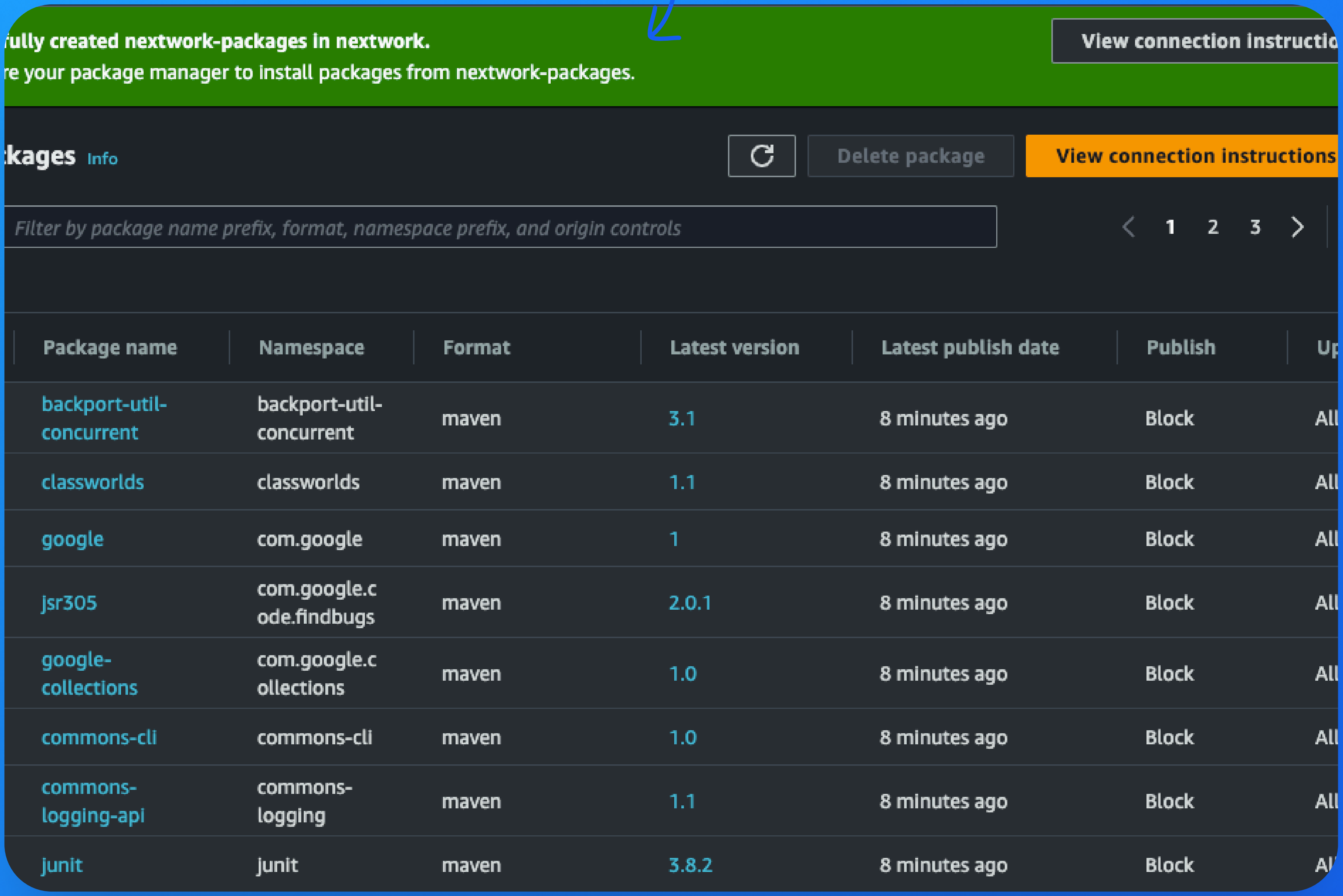
```xml
      <id>nextwork-nextwork-packages</id>
      <username>aws</username>
      <password>${env.CODEARTIFACT_AUTH_TOKEN}</password>
    </server>
  </servers>


  <profiles>
    <profile>
      <id>nextwork-nextwork-packages</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <repositories>
        <repository>
          <id>nextwork-nextwork-packages</id>
          <url>https://nextwork-381492025576.d.codeartifact.eu-west-3.amazonaws.com/maven/nextwork-packages/</url>
        </repository>
      </repositories>
    </profile>
  </profiles>


  <mirrors>
    <mirror>
      <id>nextwork-nextwork-packages</id>
      <name>nextwork-nextwork-packages</name>
      <url>https://nextwork-381492025576.d.codeartifact.eu-west-3.amazonaws.com/maven/nextwork-packages/</url>
      <mirrorOf>*</mirrorOf>
    </mirror>
  </mirrors>
```

Rasha M
@Badry2022
Rasha M.

## Test the connection

- To test the connection between Cloud9 and CodeArtifact, I compiled my web app using Maven. Compiling verifies that the setup is correct and that Maven can fetch dependencies from CodeArtifact as expected.
- After compiling, I checked the CodeArtifact repository to see the packages that were downloaded and stored. This confirmed that Maven successfully retrieved and saved the required packages in the local repository.

My web app's packages popping up in my local repository.



fully created nextwork-packages in nextwork.
re your package manager to install packages from nextwork-packages.

kages Info

View connection instructio
Delete package
View connection instructions
Filter by package name prefix, format, namespace prefix, and origin controls
1 2 3

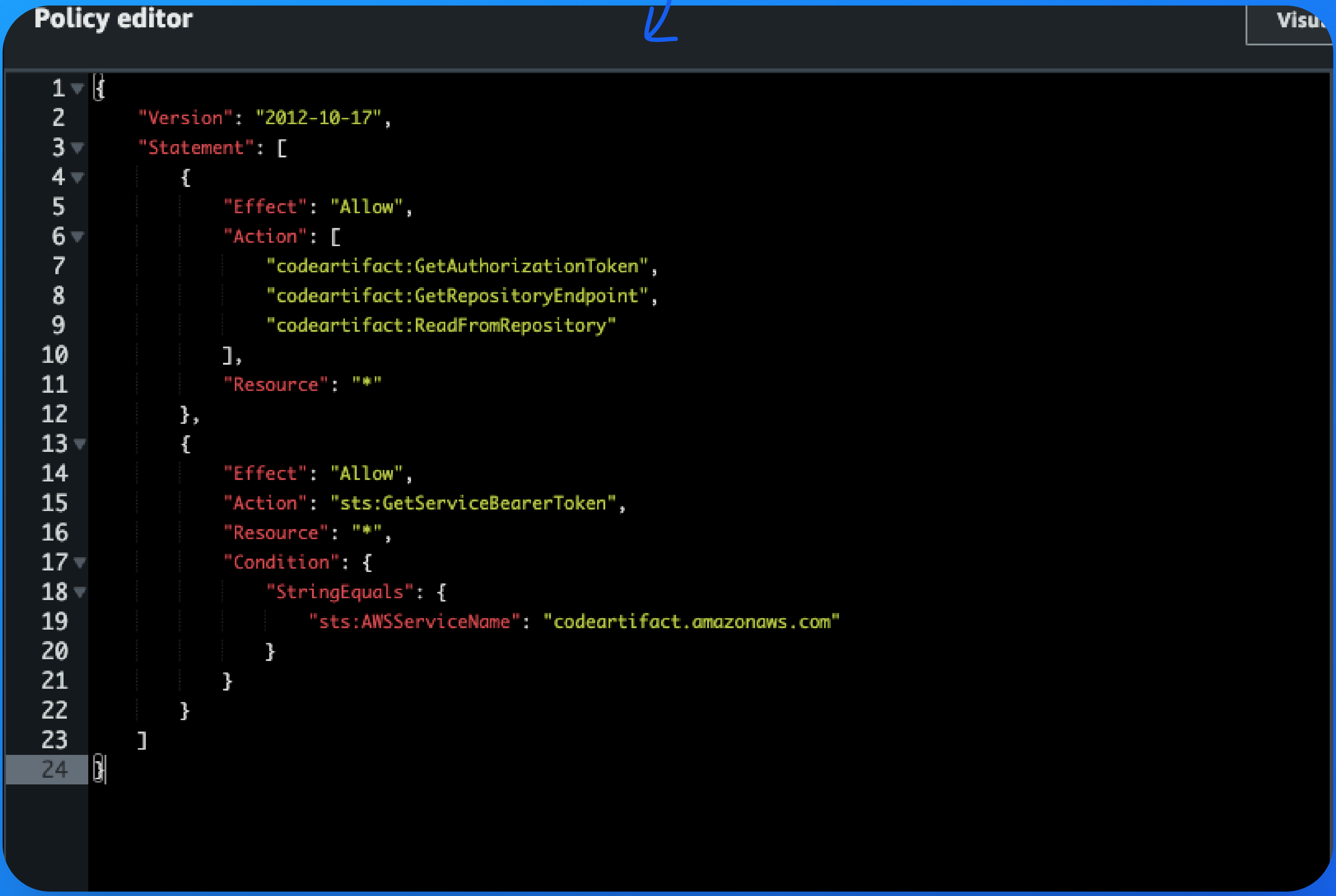
| Package name | Namespace | Format | Latest version | Latest publish date | Publish | Up |
|---|---|---|---|---|---|---|
| backport-util-concurrent | backport-util-concurrent | maven | 3.1 | 8 minutes ago | Block | All |
| classworlds | classworlds | maven | 1.1 | 8 minutes ago | Block | All |
| google | com.google | maven | 1 | 8 minutes ago | Block | All |
| jsr305 | com.google.code.findbugs | maven | 2.0.1 | 8 minutes ago | Block | All |
| google-collections | com.google.collections | maven | 1.0 | 8 minutes ago | Block | All |
| commons-cli | commons-cli | maven | 1.0 | 8 minutes ago | Block | All |
| commons-logging-api | commons-logging | maven | 1.1 | 8 minutes ago | Block | All |
| junit | junit | maven | 3.8.2 | 8 minutes ago | Block | Al |

Rasha M
@Badry2022
Rasha M.

## Create IAM policies

- I also created an IAM policy because I needed to grant permissions for other AWS services, such as CodeBuild and CodePipeline, to access and use the dependencies stored in CodeArtifact. This ensures that services could integrate with CodeArtifact for a complete CI/CD pipeline.
- I defined my IAM policy using JSON. This policy will allow services to access the CodeArtifact repository by enabling actions such as getting authorization tokens, finding repository endpoints, and reading from the repository. It also provides temporary security credentials for accessing CodeArtifact.

A peek at the policy that will provide access to CodeArtifact



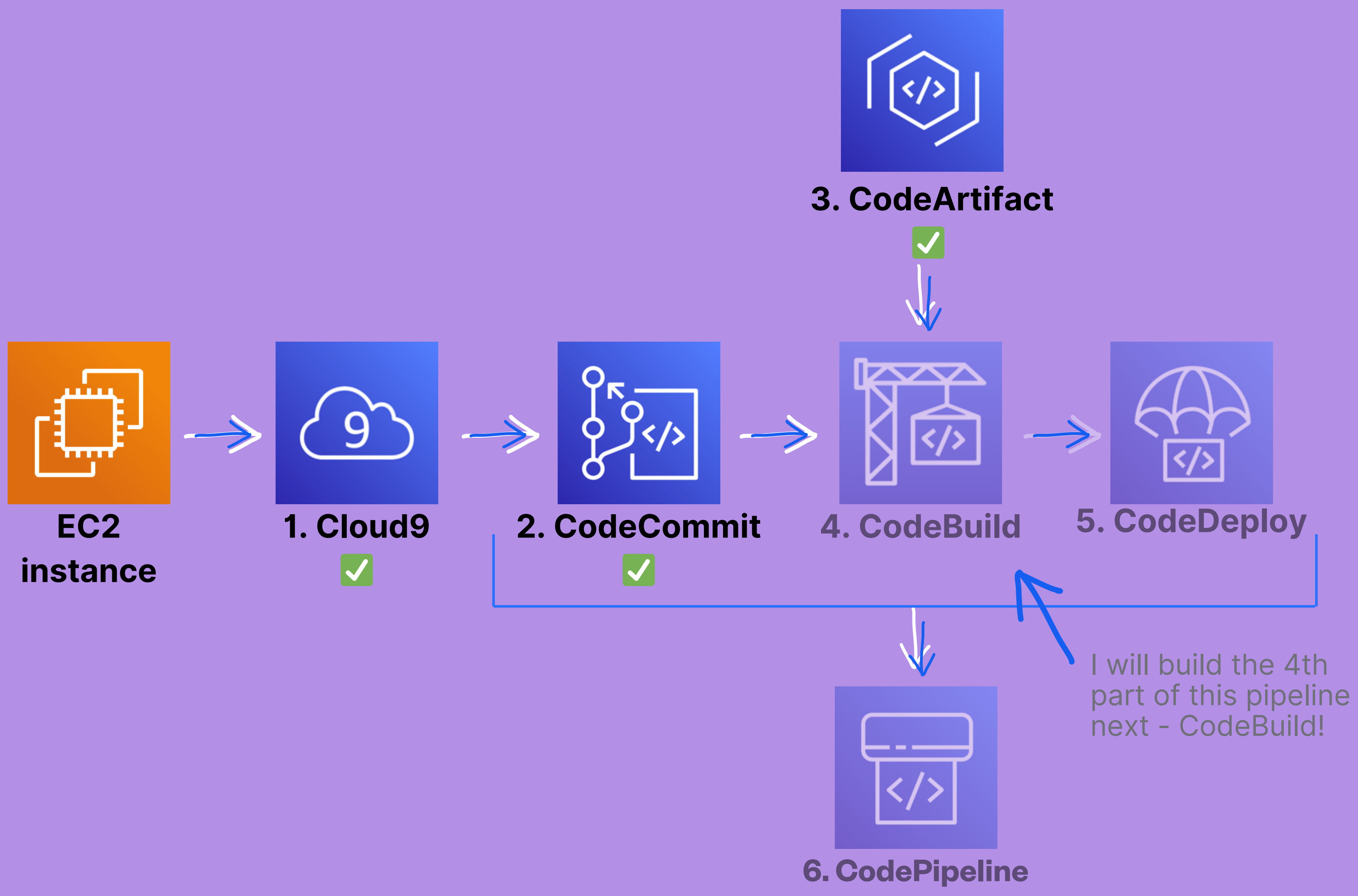
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codeartifact:GetAuthorizationToken",
                "codeartifact:GetRepositoryEndpoint",
                "codeartifact:ReadFromRepository"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "sts:GetServiceBearerToken",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "sts:AWSServiceName": "codeartifact.amazonaws.com"
                }
            }
        }
    ]
}
```

Rasha M
@Badry2022
Rasha M.

NextWork.org

# My CI/CD pipeline so far...

1. AWS Cloud9 is responsible for providing a development environment where code is written and compiled.
2. AWS CodeCommit is responsible for version control, storing the codebase, and managing commits.
3. AWS CodeArtifact is responsible for managing and securing the project's dependencies.



Rasha M
@Badry2022
Rasha M.

## My key learnings

1. A public upstream repository is a repository that provides access to a wide range of packages available publicly on the internet. In this case, it mirrors packages from the Maven Central Repository to ensure a reliable source for dependencies.

2. settings.xml is a file I set up to configure Maven to connect to the CodeArtifact repository. It includes necessary information for Maven to authenticate and fetch packages from the CodeArtifact repository.

3. To test the connection between Cloud9 and CodeArtifact, I compiled my web app. This ensures that Maven can successfully retrieve and use the dependencies stored in CodeArtifact.

4. One thing I didn’t expect was how quickly the CodeArtifact repository populated with packages after the initial compilation. It was impressive to see all the dependencies being managed so efficiently.