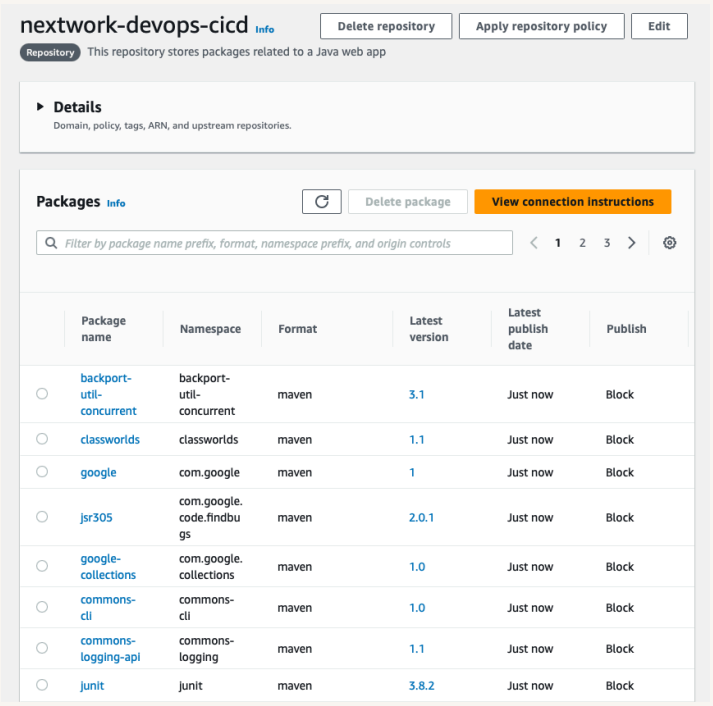
nextwork.org

Secure Packages with CodeArtifact

Tawanda Nigel Chitapi



Tawanda Nigel Chitapi
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to setup CodeArtifect as the CI/CD pipeline's artifect repository. I'm doing this project to learn the importance and value of having an artifect repository. it is a huge time saver.

Key tools and concepts

Services I used were EC2, IAM, CodeArtifact and Cloudshell.

Project reflection

This project took me approximately 1.5hrs

This project is part three of a series of DevOps projects where I'm building a CI/CD pipeline!

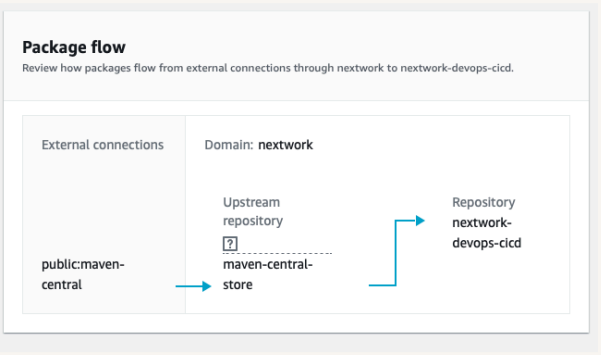
Tawanda Nigel Chitapi
NextWork Student

CodeArtifact Repository

CodeArtifact is a secure, central artifact repository to store all relevant web app (project) software packages. Engineering teams use Code Artifact for reliabily and security of dependencies.

A domain is like a folder that holds multiple repositories under the same project or organization. They are helpful for setting up permission settings.

A CodeArtifact repository can have an upstream repository, which means a public source of packages can be queried for packages if they do not exist in the local code artifact repository. My repository's upstream repository is Maven central store which is the largest Java repository and is extremely helpful when building a java web app.



Tawanda Nigel Chitapi
NextWork Student

CodeArtifact Security

Issue

To access CodeArtifact, we need EC2 to have the permission to access CodeArtifact. I ran into an error when retrieving a token because the EC2 instance does not have the permisson to access the CodeArtifact by default

Resolution

To resolve the error with my security token, I created an IAM policy that grants access to CodeArtifact and then applied that policy to an IAM rtole that I can attach to the EC2 instance. This resolved the error because the EC2 instance now has access to request an authorization token from the repository.

It's security best practice to use IAM roles because IAM roles are more secure and scalablable. Hard coded credentials are much more valnurable to security attacks.

Tawanda Nigel Chitapi
NextWork Student

The JSON policy attached to my role

The JSON policy I set up grants access to CodeArtifact. Retrieving the an authorization token, finding the repository endpoint and viewing the packages inside the repository

Tawanda Nigel Chitapi
NextWork Student

Maven and CodeArtifact

To test the connection between Maven and CodeArtifact, I compiled my web app using settings.xml

The settings.xml file configures Maven to use the CodeArtifact repository. Its supplies Maven with the name and authetication token to get access to CodeArtifact repository as well as sets up a profile section in case there are multiple repositories, maven will know which one to use.

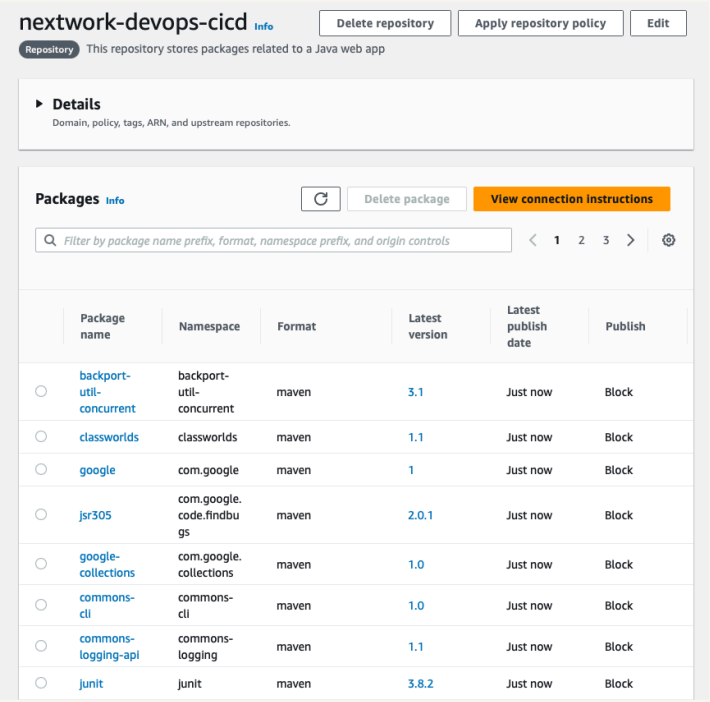
Compiling means the process of translating the webapp source code into something machinescan run. Maven is the compiler. While compiling Maven will need to put together all the package that the webapp needs. To Retrieve these packages, Maven visits CodeArtifact who will then say i do not have any packages, Maven will then go upstream to reqquest the exact same package from the upstream repository with is Maven Central and stores these package in the CodeArtifact repository.

Tawanda Nigel Chitapi
NextWork Student

Tawanda Nigel Chitapi
NextWork Student

Verify Connection

After compiling, I checked the CodeArtifact repository and I noticed 4 pages of packages which was a sign that all necessary packages were retrieved and stored.



Tawanda Nigel Chitapi
NextWork Student

Uploading My Own Packages

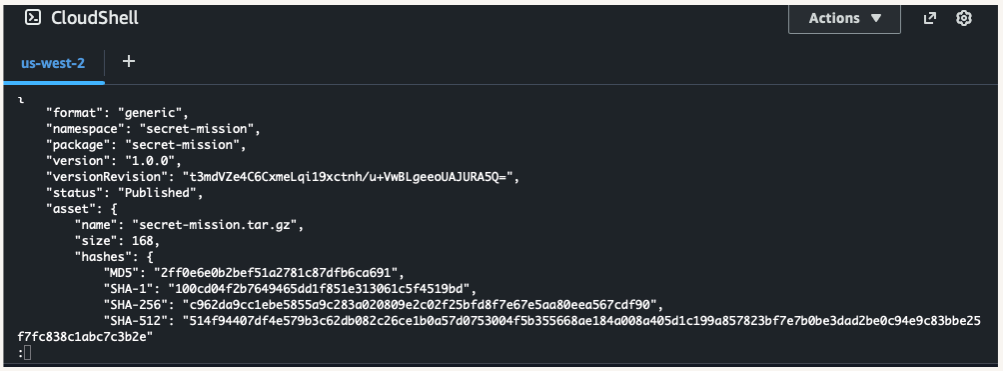
In a project extension, I also decided to become a package publisher which means publish my own packages into the CodeArtifact Repository. This is useful in situations where there may be internal team members developing packages and wanting fellow team members to access them without giving the entire world access.

To create my own package, I set up a txt file and used tar to package that txt file. I also generated a security hash because that will give CodeArtifact a way to figure out if the package has been tempered with in transit if the hashes do not match up.

To publish the package, I ran a CLI command that uploads the package to the repository. When I look at the package details in CodeArtifact, I can see the version number, publish date and even the origin, in this case the CodeArtifact itself is the repository.

To validate my packages, I then tried to download the package into the cloud terminal. The package was successfully installed from Code Artifact and unzipped it to read its contents.

Tawanda Nigel Chitapi
NextWork Student



The place to learn & showcase your skills

Check out nextwork.org for more projects