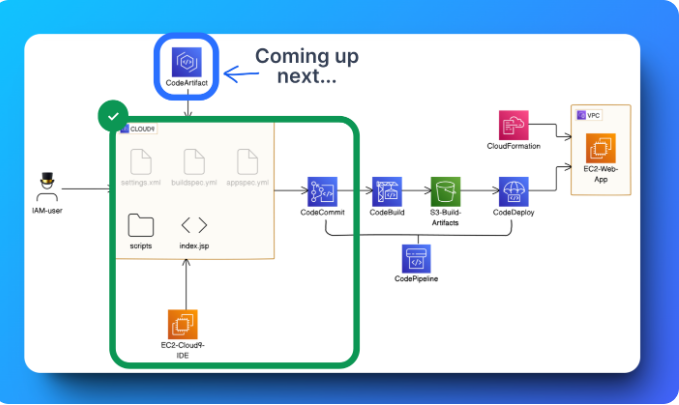
# Project 3: [Cloud DevOps - Secure Project Dependencies with AWS CodeArtifact](https://community.nextwork.org/c/all-aws-projects/project-7-cloud-devops-secure-project-dependencies-with-aws-codeartifact)

Services used: AWS Cloud9, codecommit, codeArtifact, IAM and EC2.

AWS Artifact is a service that provides access to compliance-related documents for AWS. Think of it as a library where you can find reports and certifications that show AWS meets certain security and compliance standards. This is useful if your organization needs to prove that the cloud services you're using are secure and compliant with regulations.

AWS Artifact is a service that provides access to AWS’s security and compliance documents, including reports and agreements. It helps organizations ensure that their use of AWS meets regulatory requirements.



Think of AWS CodeArtifact like a secure closet where you can keep all the building blocks (we call these "packages") your web app needs to work. Just like you might store extra batteries or light bulbs at home so you're ready if something goes out, CodeArtifact helps you keep copies of these packages safe. This way, even if something goes wrong with the public repositories on the internet where these packages usually come from, you still have everything you need to keep building your app.

### STEP 1: **Set Up Your Web App and Git Repo (Projects 1 and 2 of this series)**

**How to use AWS CodeArtifact to secure our project's dependencies**

**What are project dependencies?**

If your application depends on a package to function correctly (i.e. your app won't work as it should/won't run at all if the package becomes unavailable), then that package is considered a dependency for your application.

**What are project dependencies?**

If your application depends on a package to function correctly (i.e. your app won't work as it should/won't run at all if the package becomes unavailable), then that package is considered a dependency for your application.

STEP 2: **Create Your Domain and Repository**

AWS CodeArtifact is a service that acts as a private repository for your application's dependencies. It ensures that even if public resources become unavailable, you can still access your backup copies from CodeArtifact. This adds reliability and security to your CI/CD pipeline, preventing disruptions in your development process

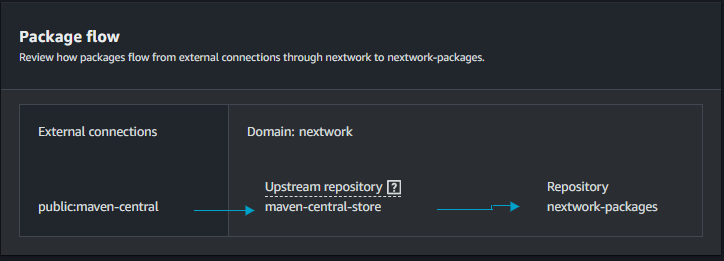
**Why are we creating a domain?**

A domain in AWS CodeArtifact serves as a central organizational structure where you can manage repositories containing your project materials. It provides a unique URL that applications use to access and store dependencies. Developers typically create a single domain to manage all their artifacts, while larger organizations might use multiple domains to isolate resources for different projects or security needs.

* Create a new domain
* Create a repository

**What is the public upstream repository?**

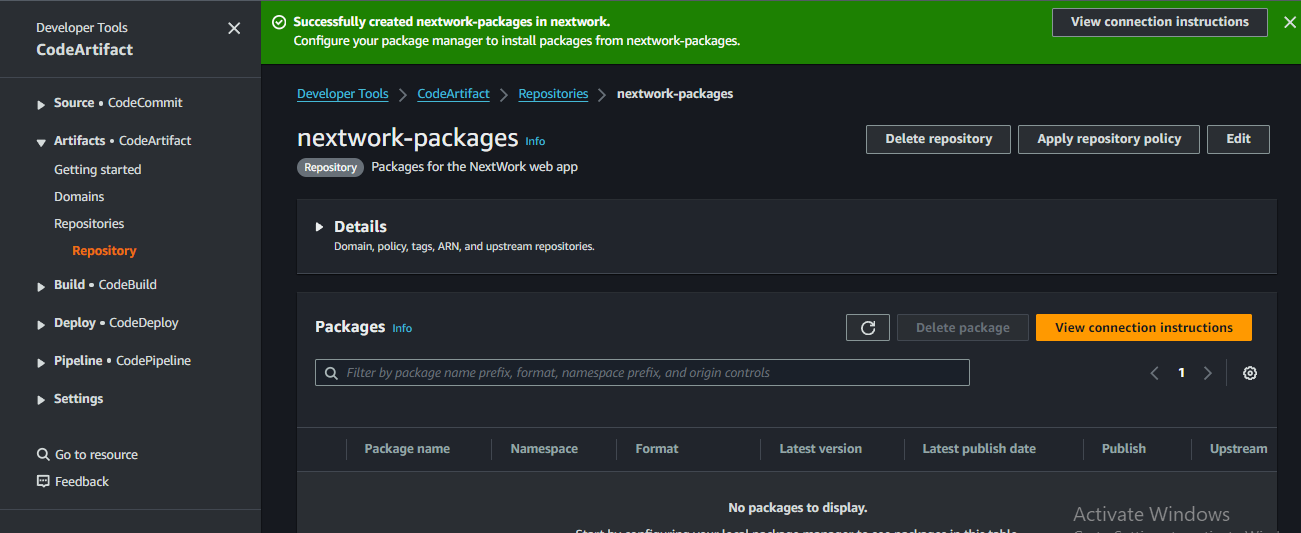
The public upstream repository in AWS CodeArtifact, like "maven-central-store," serves as an intermediary between your local repository and the Maven Central Repository. When Maven builds your project, it first checks your local repository (e.g., "nextwork-packages") for required packages. If they are not found locally, Maven checks the upstream repository, which acts like a nearby store with frequently used packages. This upstream repository is linked to the Maven Central Repository, a vast global library of Java packages. By selecting "maven-central-store" as the upstream repository, you ensure that your CodeArtifact domain can quickly access and backup necessary packages from this reliable source.



**What is in a package flow?**

The package flow in AWS CodeArtifact involves three key repositories:

1. **Public Repository (maven-central):** This is the vast global library on the internet, storing a wide variety of Java tools and materials. It's the primary source that supplies packages to other repositories.
2. **Upstream Repository (maven-central-store):** This acts as your contact point with the public repository, helping you retrieve and store new tools as needed for your projects.
3. **Local Repository (nextwork-packages):** This is your personal toolbox, where you keep all the specific tools and materials for your current project.



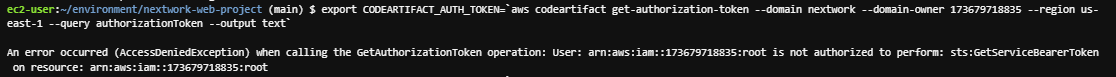
STEP 3: **Connect the CodeArtifact repository**

* Connect Cloud9 to to CodeArtifact
* Copy the command to connect Cloud9 to CodeArtifact.

We view connection instructions to link your Cloud9 IDE with CodeArtifact repositories, enabling your development environment to fetch and store dependencies. The instructions are based on the EC2 instance running Amazon Linux 2 and Maven as the package manager, which manages your Java web app's dependencies and building process.

The command exports a CodeArtifact authorization token, which acts as a key for your Cloud9 IDE to access your repositories in CodeArtifact.

Error encountered:



Today you've learnt how to:

1. 🌱 **Set up a CodeArtifact domain and repository:** You created a new domain and repository in AWS CodeArtifact to securely store and manage the dependencies for your Java web app.
2. ⛓️ **Connect Cloud9 to CodeArtifact**: You configured your Cloud9 IDE to connect seamlessly with CodeArtifact.
3. 🥪 **Compile your project with CodeArtifact:** You compiled your Java project using the settings from CodeArtifact, verifying that all dependencies were correctly fetched and utilized.
4. 💂‍♀️ **Implement IAM policies for CodeArtifact:** You defined and applied IAM policies to grant secure access to the CodeArtifact repository, ensuring that only authorized services can interact with your stored dependencies.