NextWork.org

Secure dependencies with CodeArtifact

Sadeesha Perera


Sadeesha Perera
linkedin.com/sadeesha-perera

NextWork.org

Introducing AWS CodeArtifact!

What it does & how it’s useful

AWS CodeArtifact is a secure, highly scalable, managed artifact repository service. Developers and teams use AWS CodeArtifact because to help organize, store and share software packages for application development.

How I’m using it in today’s project

I’m using AWS CodeArtifact in this project to Store backups of my packages and dependencies for my web application.

This project took me...

This project toke me about 1 hour to complete. Documentation took me 30 minutes to complete, in total I spent 1 hour and 30 minutes on this project.


Sadeesha Perera
linkedin.com/sadeesha-perera

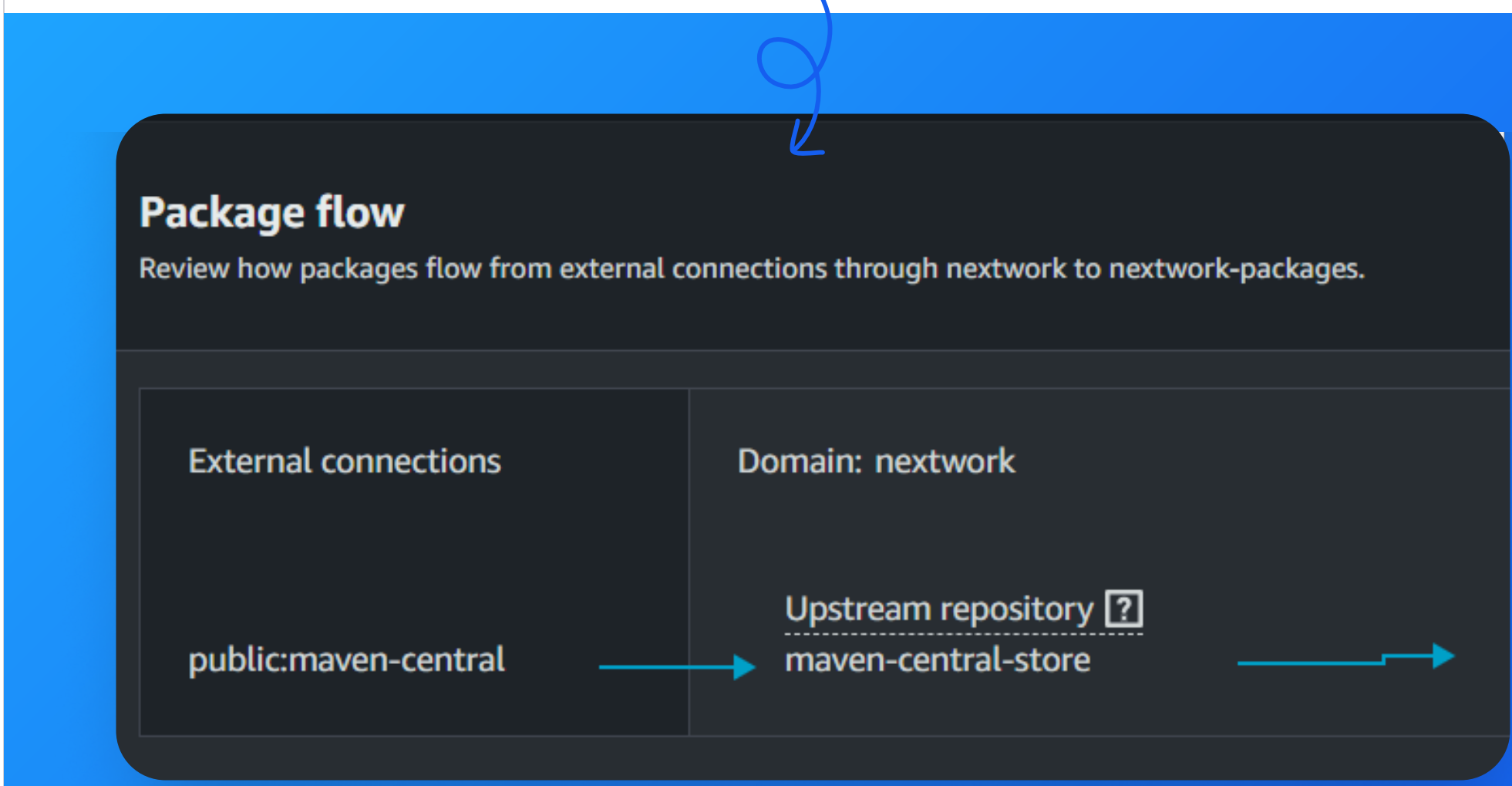
Create a repository

- CodeArtifact is a service that sits in the CI/Cd pipeline. I am using today to store backup copies of packages (and more importantly, dependencies) relevant to my Java web app.

- The reason why I'm using CodeArtifact for my web app is for security/risk management, and continuity. Now, even if the public packages/dependencies of my project are no longer available, there is a copy in my AWS CodeArtifact repositories to make sure i can keep developing my web app.

- The third is the Maven Central Repository, which is a public repository with the greatest collection of packages of Java applications. However, Maven will only visit this repository if the first two do not have the packages/dependencies it is looking for (due to high traffic going into Maven Repository

Package flow illustrating the connections between the three repositories.




Sadeesha Perera
linkedin.com/sadeesha-perera

Connecting my project to CodeArtifact

- Next, I connected my web app project via (Cloud9 IDE) to CodeArtifact so that CodeArtifact knows which project it is going to store the dependencies for.

- 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 the repositories that are storing these 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 stores authentication tokens to CodeArtifact and defines when Maven will visit which repository, and where Maven should visit to find backup local repositories (optional).

My settings.xml file.

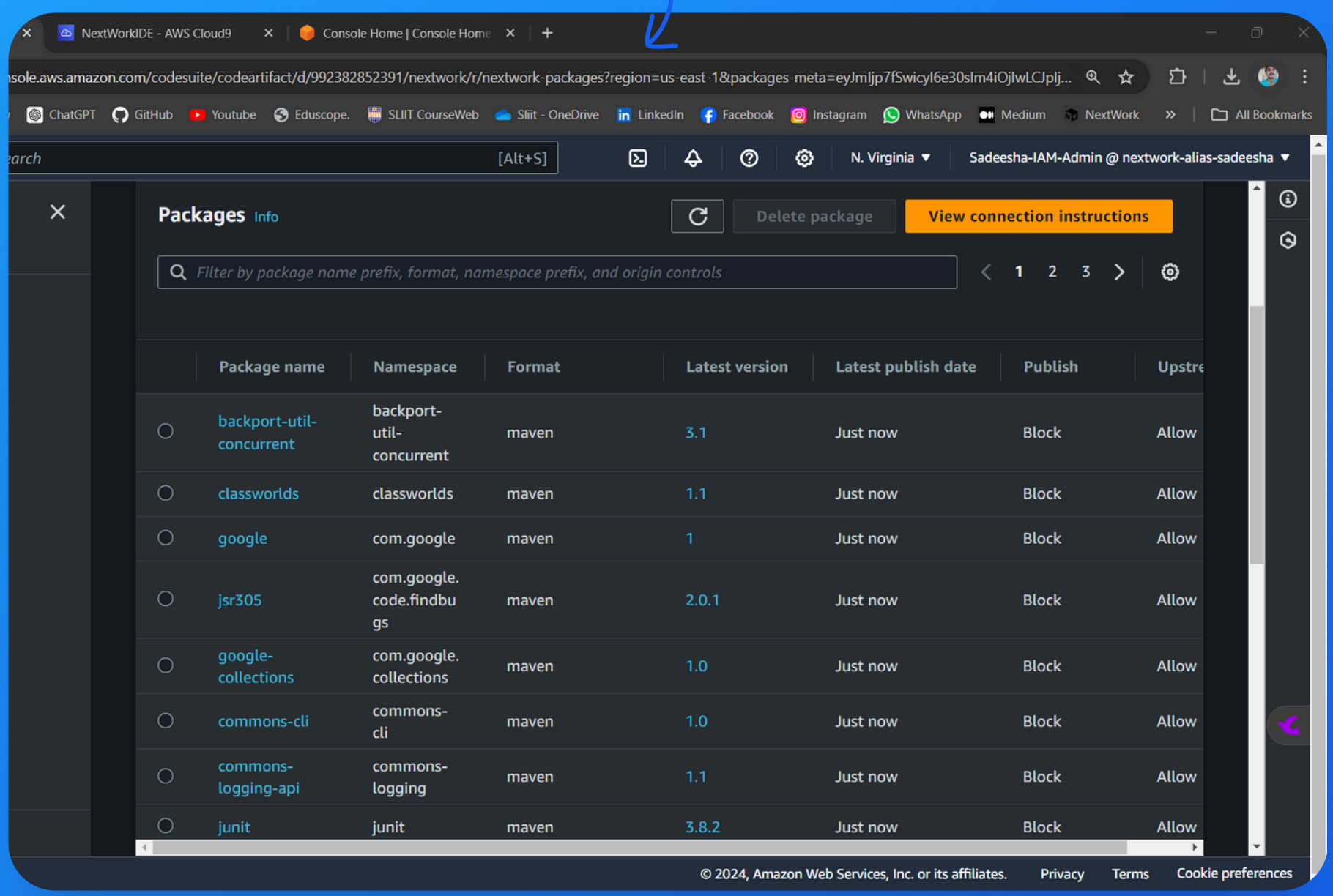
Welcome | index.jsp | settings.xml

```xml
<settings>
    <servers>
  <server>
    <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-992382852391.d.codeartifact.us-east-1.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-992382852391.d.codeartifact.us-east-1.amazonaws.com/maven/nextwork-packages/</url>
    <mirrorOf>*</mirrorOf>
  </mirror>
</mirrors>
```

Test the connection

- To test the connection between Cloud9 and CodeArtifact, I compiled my web app. Compiling means translating my web application's code into machine code that servers can actually understand and run.
- After compiling, I checked my local repository and saw that my local repository now has a lot of pages of packages inside. This means that Maven has now grabbed packages from the upstream repository/Maven Central Repository and installed /kept a copy locally.

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



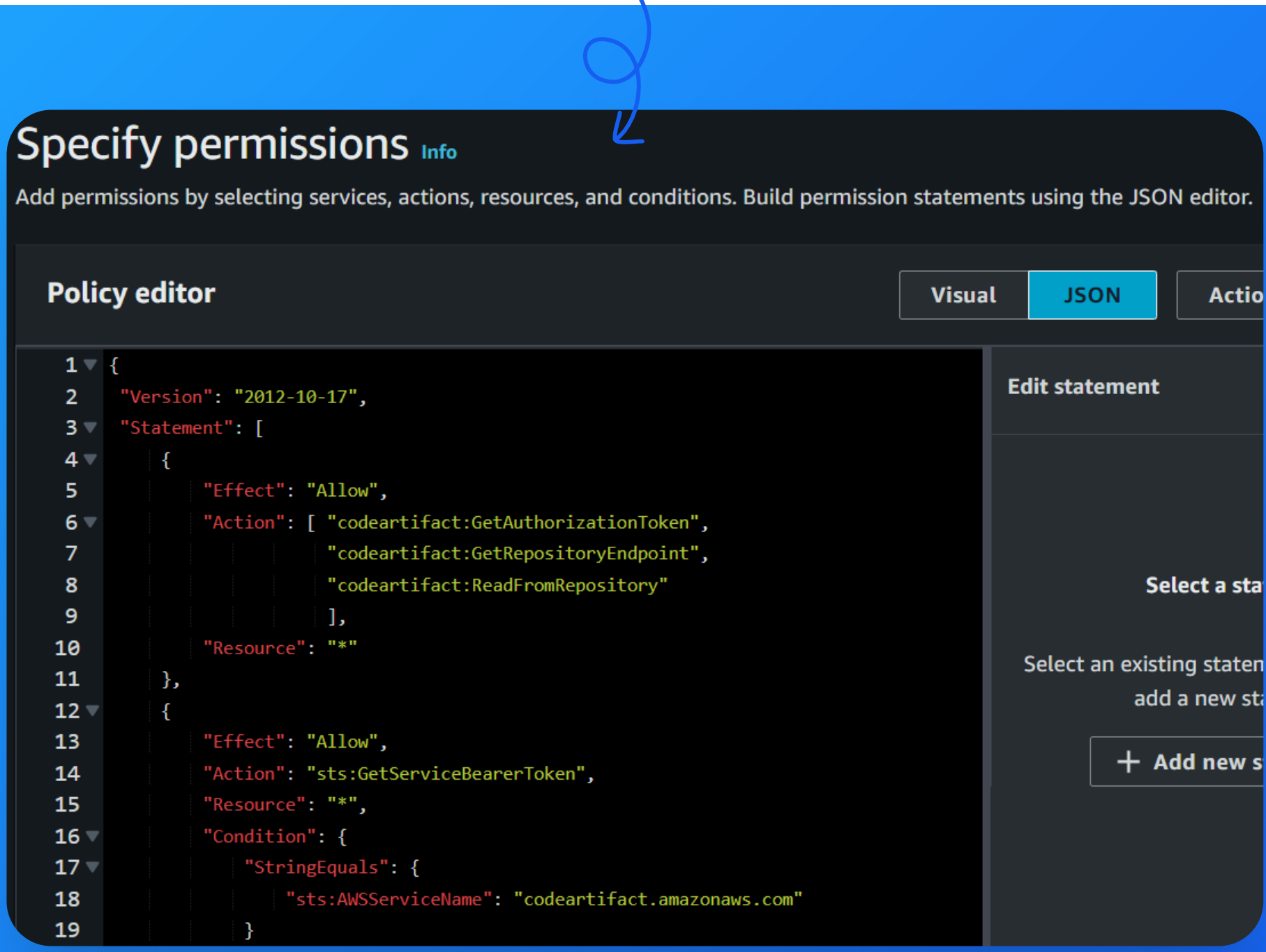
Sadeesha Perera
linkedin.com/sadeesha-perera

Create IAM policies

- I also created an IAM policy because other services in my CI/CD Pipeline e.g. CodeBuild, and CodePipeline will be needing access to the packages/dependencies stored on CodeArtifact. By default, these services do not have access - so they need to be granted access through an IAM policy.

- 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 repositories.

A peek at the policy that will provide access to CodeArtifact



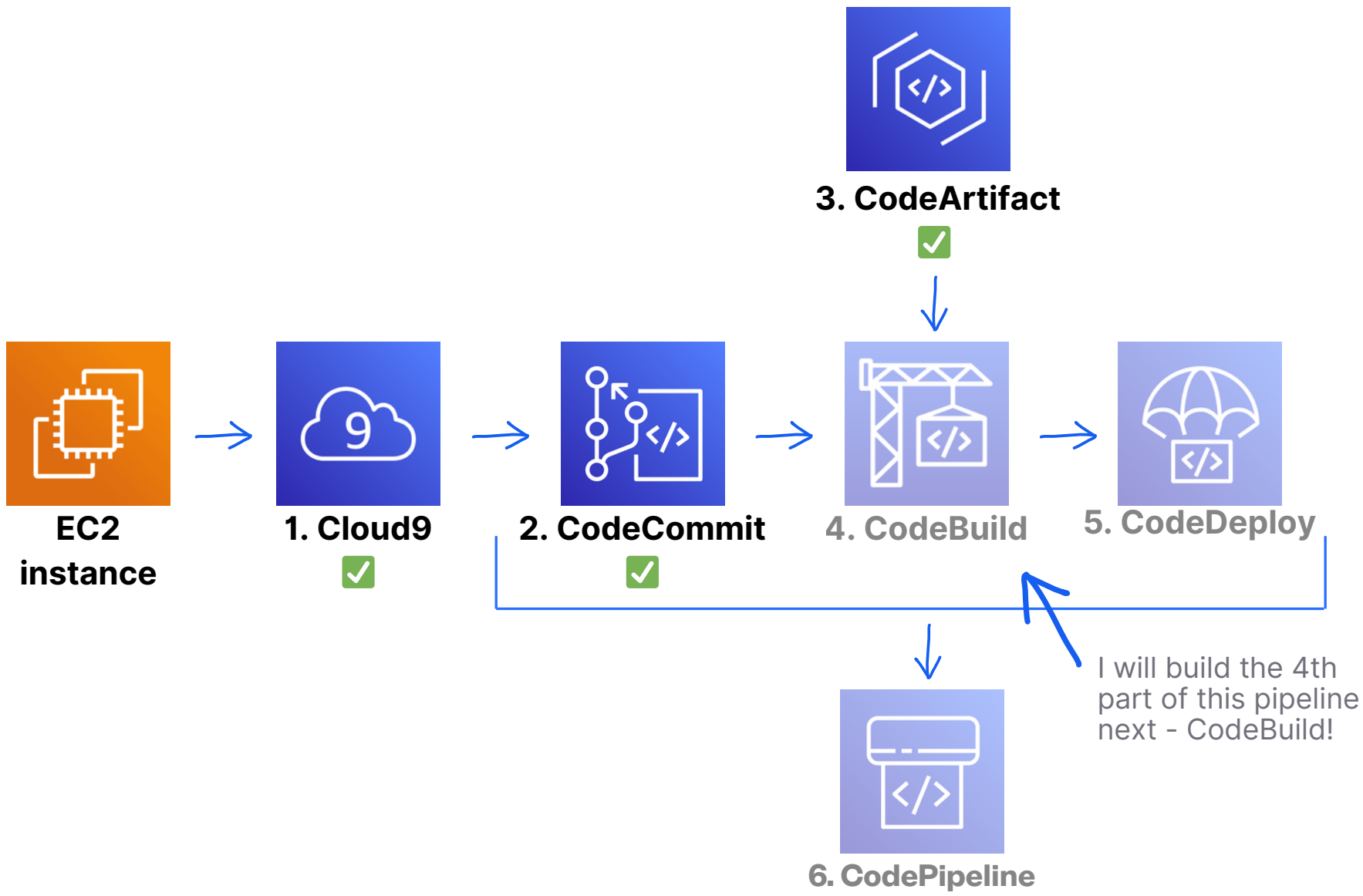
Sadeesha Perera
linkedin.com/sadeesha-perera

My CI/CD pipeline so far...

1. **AWS Cloud9** is responsible for running the IDE where Java and Maven are both installed on, this is where i will be building and editing the code for my web app.

2. **AWS CodeCommit** is responsible for storing my Git repository from my Cloud9 environment that I will be using for my web application project.

3. **AWS CodeArtifact** is responsible for storing backup copies of my web app dependencies, thus providing security and reliabilty from external disruption for my web application.




Sadeesha Perera
linkedin.com/sadeesha-perera

My key learnings

1. A public upstream repository i.e., maven-central-store is, basically where your local repository gets the tools it needs when said tools are not available on the local repository.

2. settings.xml is a file I set up to tell Maven where to find the dependencies and how to connect to the right repositories (e.g. the ones in CodeArtifact) to get access.

3. To test the connection between Cloud9 and CodeArtifact, I compiled my web app. Compiling means translating my web application’s code into machine code that servers can actually understand and run.

4. One thing I didn’t expect was how easy this is.

NEXTWORK

Everyone should be in a job they love. yes!

Check out community.nextwork.org for more free projects