MANAGING A CI/CD PIPELINE WITH AWS CODE FAMILY

PROJECT 3 / 6

SECURE PROJECT DEPENDENCIES WITH AWS CODEARTIFACT

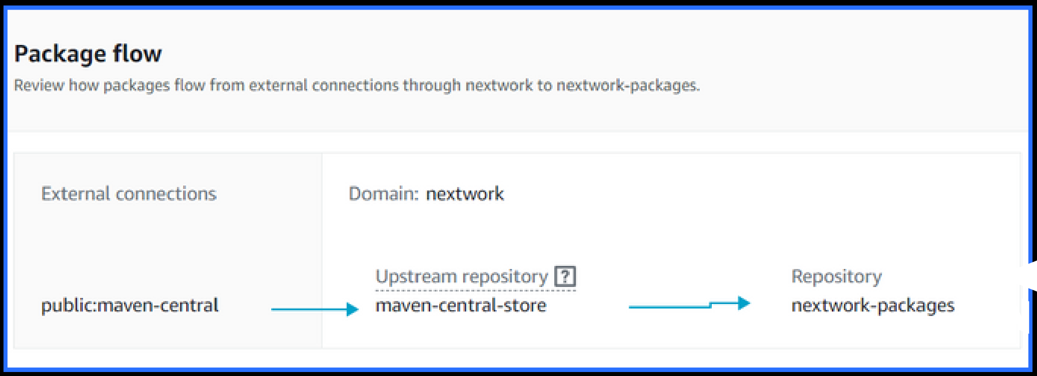
Kanika Mathur

@github.com/KanikaGenesis

01 CREATE A REPOSITORY

- CodeArtifact is a service that sits in the CI/CD pipeline. I am using it today to store backup copies of packages (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. Even if public packages or 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.
- Instead of a single repository, there are actually three connected repositories that Maven uses to fetch packages.
    - The first is my local repository, which Maven checks for packages/dependencies for my web app.
    - The second is my public upstream repository which Maven will check next if the package is not in the local repository.
    - The third is the Maven Central Repository, which is a public repository with the greatest collection of packages for Java application. 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 Central Repository).

Package flow illustrating the connections between the three repositories.



Package flow

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

External connections: public:maven-central → Domain: nextwork: Upstream repository: maven-central-store → Repository: nextwork-packages

Kanika Mathur

@github.com/KanikaGenesis

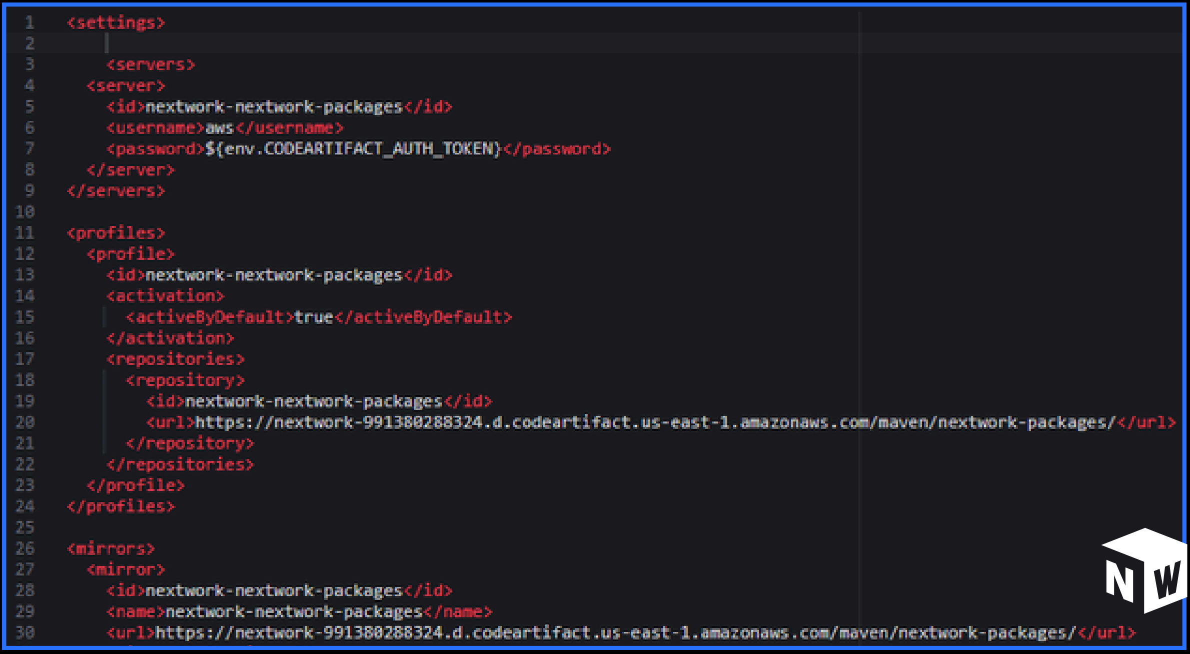
02 CONNECT IDE TO CODEARTIFACT

- Next, I connected my web app project (via my Cloud9 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. 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 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, plus where Maven should visit to find backup local repositories (optional).

My settings.xml file



```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-991380288324.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-991380288324.d.codeartifact.us-east-1.amazonaws.com/maven/nextwork-packages/</url>
```

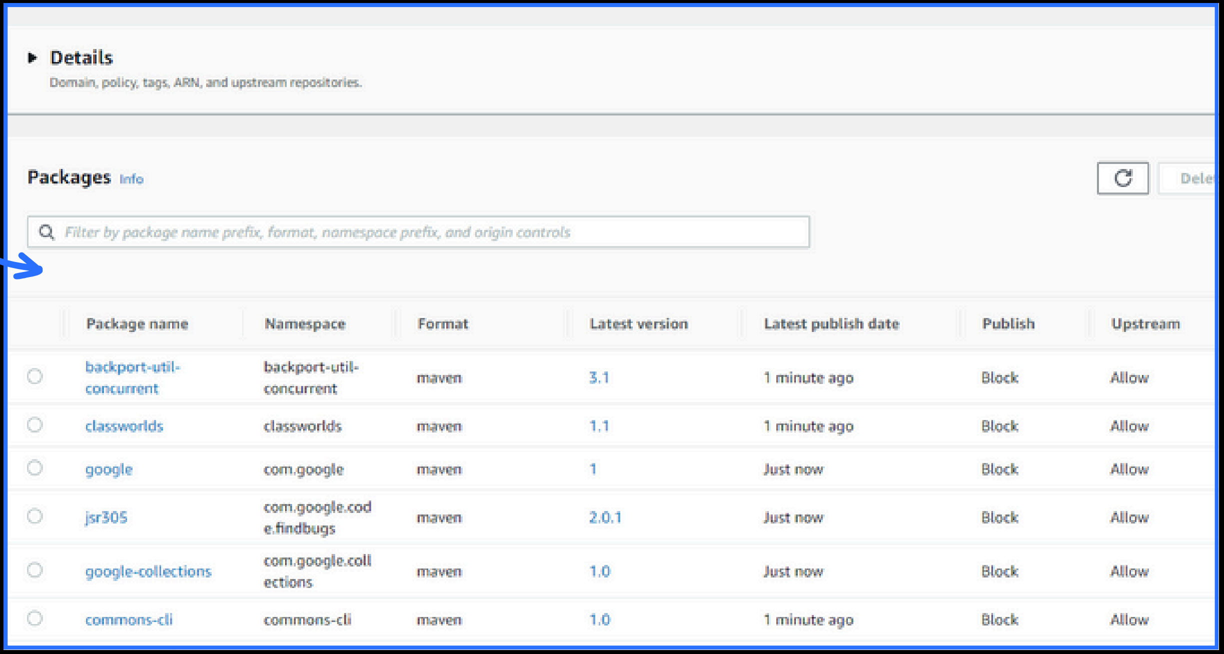
Kanika Mathur

@github.com/KanikaGenesis

03

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 pages and pages of packages inside! This means 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

▸ Details
Domain, policy, tags, ARN, and upstream repositories.
Packages Info
Dele
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	1 minute ago	Block	Allow
○	classworlds	classworlds	maven	1.1	1 minute ago	Block	Allow
○	google	com.google	maven	1	Just now	Block	Allow
○	jsr305	com.google.code.findbugs	maven	2.0.1	Just now	Block	Allow
○	google-collections	com.google.collections	maven	1.0	Just now	Block	Allow
○	commons-cli	commons-cli	maven	1.0	1 minute ago	Block	Allow

Kanika Mathur
@github.com/KanikaGenesis

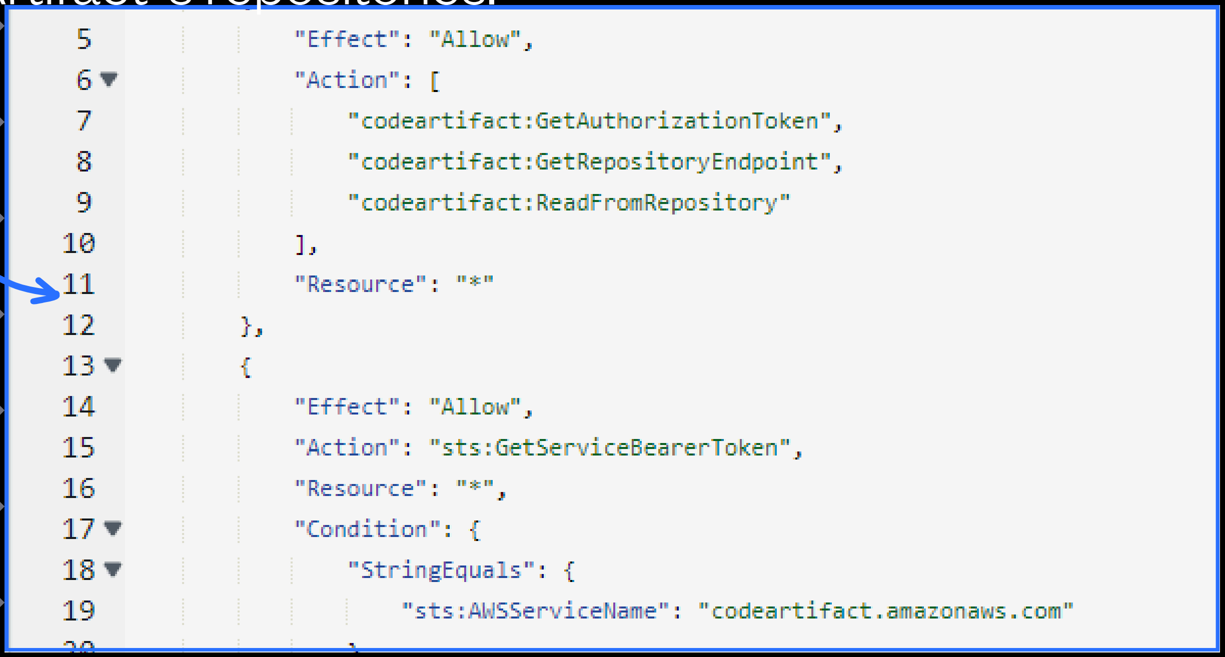
CREATE IAM POLICIES

- 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. 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 token (i.e. access to CodeArtifact), fetch packages stored in CodeArtifact's repositories.

04

A peek at the JSON Policy added to provide access to my CodeArtifact repositories



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

Kanika Mathur

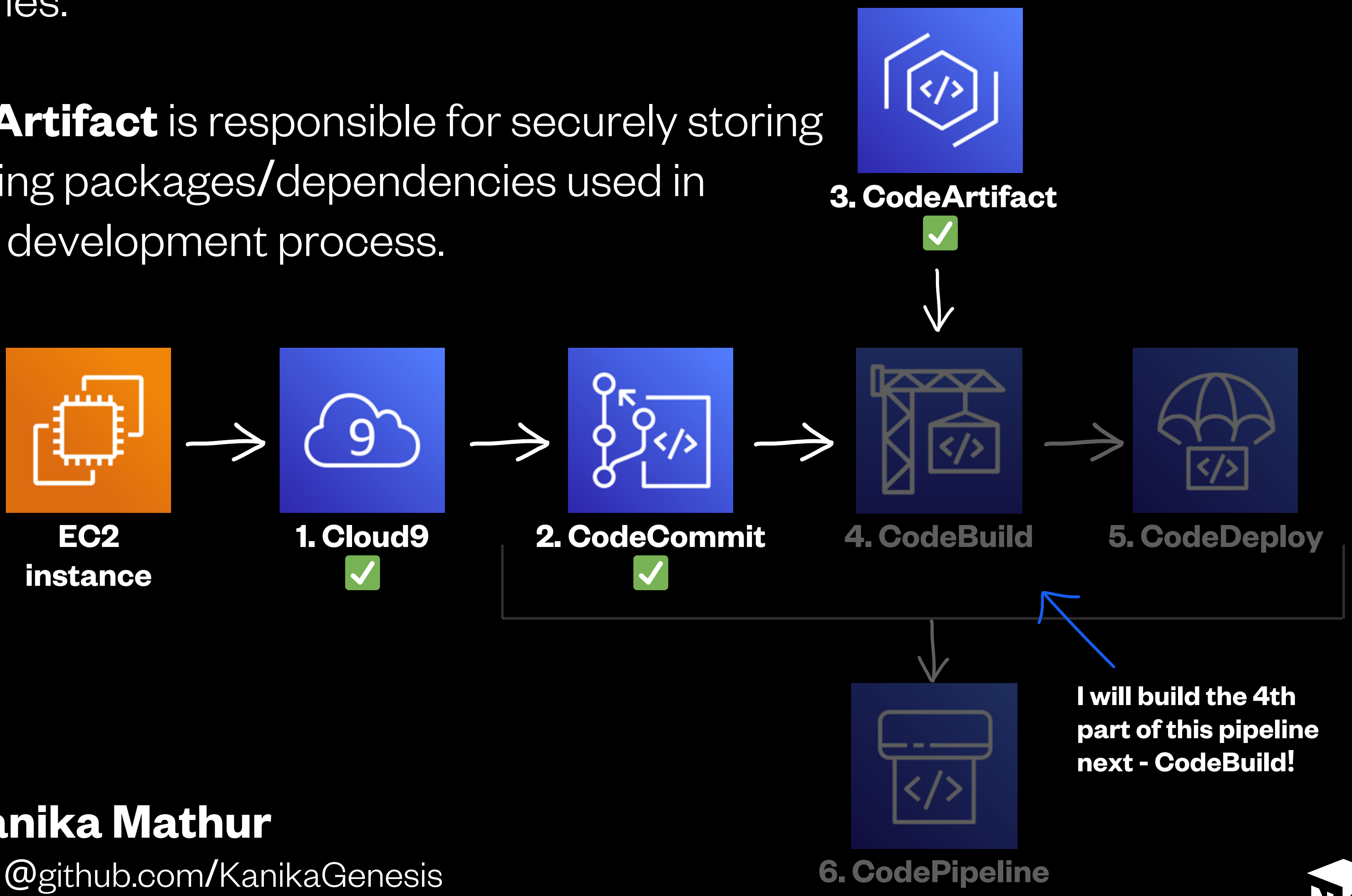
@github.com/KanikaGenesis

MY CI/CD PIPELINE PROGRESS...

1. Cloud9 is responsible for IDE that allows you to write ,run and debug your code.

2. CodeCommit is responsible for hosting git repositories.

3. CodeArtifact is responsible for securely storing and sharing packages/dependencies used in software development process.



I will build the 4th part of this pipeline next - CodeBuild!

Kanika Mathur

@github.com/KanikaGenesis

MY KEY LEARNINGS

01 CodeArtifact is an AWS service that provides a secure, scalable repository for storing and managing software packages, making it easy to share packages within your organization and integrate them into your development workflow.

02 A public upstream repository is a central code repository that is accessible to everyone, where changes can be pulled from. It serves as a primary source for contributors to fetch updates and integrate them into their own local repositories

03 settings.xml is a file I set up to configure Maven settings, such as repository locations, proxies, and authentication details for accessing private repositories.

04 To test the connection between Cloud9 and CodeArtifact, I configured the settings.xml file with the necessary repository information and authentication details, then ran Maven commands to compile the project and ensure it could fetch dependencies from the CodeArtifact repository successfully.

05 I also learned how to manage and configure domain-specific settings for Maven, improve my understanding of secure package management with CodeArtifact, and enhance my skills in integrating various AWS services for a seamless development workflow.

Kanika Mathur
@github.com/KanikaGenesis

FINAL THOUGHTS...

- This project took me 40 minutes to complete.
- Delete **EVERYTHING** at the end! Let's keep this project free :)
- One thing I didn't expect was how seamless and efficient the integration between Cloud9 and CodeArtifact would be, making the process of compiling and managing dependencies much smoother than anticipated.
- In the next part of this 6-project series, I will use **AWS CodeBuild** to build our WAR (Web Application Resource) file for the project.

FIND THIS HELPFUL?

- Like this post
- Leave a comment
- Save for later
- Let's connect!

Kanika Mathur

@github.com/KanikaGenesis

Thanks NextWork for the free project guide!

NEXTWORK