**A Note on AWS CodeArtifact**

The AWS CodeArtifact is a fully managed artifact repository service that makes it easy for organizations of any size to securely store, publish, and share software packages used in their software development process – <https://aws.amazon.com/codeartifact/>. A CodeArtifact repository is an all-in-one repository that that can serve as a NUGET repository, a NPM registry, a PIP repository, a Maven repository, etc. In this note, we will provision a CodeArtifact repository. We will demonstrate how to push and restore NUGET packages to the repository. In particular, we will try to answer the following questions.

* What is the minimal permissions required for a developer to push and restore Packages?
* If we want to give access to the repository to a user in another AWS account other than the account where the repository is provisioned, how can we grant the permissions to the AWS account?
* How can we prepare the developer’s client environment so he can push and restore packages?
* How can we access the repository from a CodeBuild pipeline – <https://aws.amazon.com/codebuild/>?

But first let us provision a CodeArtifact repository.

**The CodeArtifact Domain & Repository**

A CodeArtifact repository can be either provisioned in the AWS console or through a CloudFormation stack. The following is the AWS-CDK stack used in this Note. The stack will deploy all that is needed for the CodeArtifact repository including all the permissions.

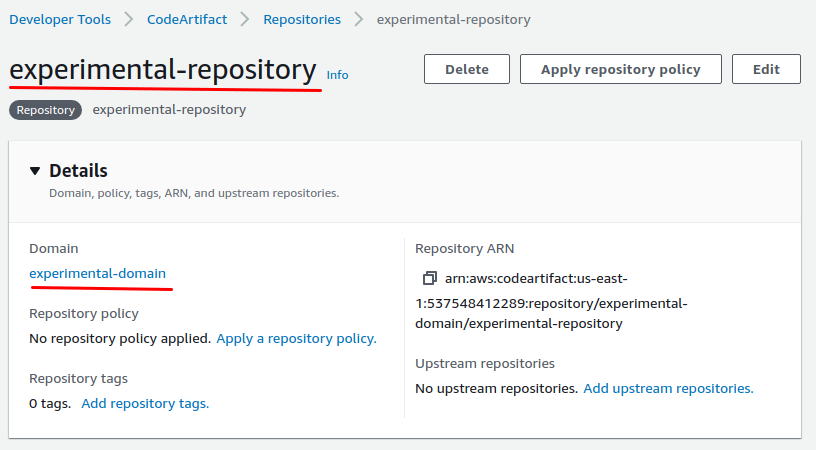




The CDK stack defined three AWS resources:

1. A CodeArtifact domain named “experimental-domain”;
2. A CodeArtifact repository named “experimental-repository”. The repository is created in the “experimental-domain”;
3. A user group named “experiment-code-artifact-user-group” that defines the minimal permission required to push and restore NUGET packages from the repository.

The following is the repository as shown in the AWS console.



**Minimal Permissions Required to Push and Restore**

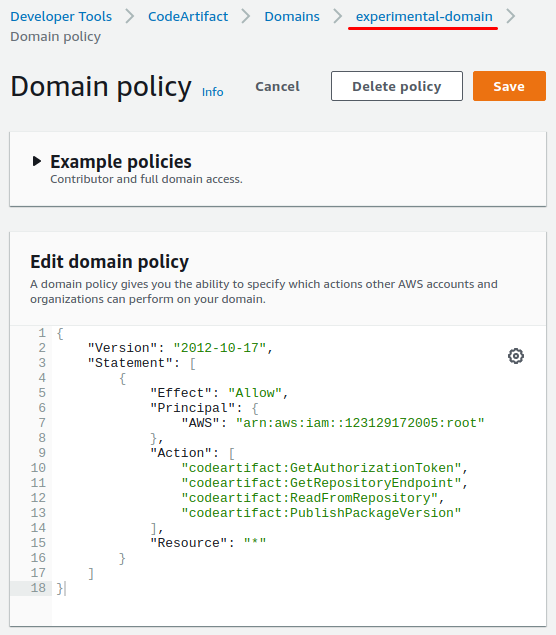
The permissions to access the CodeArtifact repository is managed by AWS IAM. In the user group “experiment-code-artifact-user-group”, we have the minimal permissions required to push and restore a NUGET package defined.



A user may have been granted the permission. For example, an admin user may have all the permissions to the AWS resources. But if the user does not have the permission, we can assign the user to the “experiment-code-artifact-user-group”, so he can push and restore NUGET packages.

**Permissions to Another AWS Account**

If we want to grant access permissions to another AWS account, we need to add a policy explicitly. This is very useful when we want to access the repository from a CodeBuild pipeline. The following is the domain policy to explicitly grant permissions to the CodePipeline account.

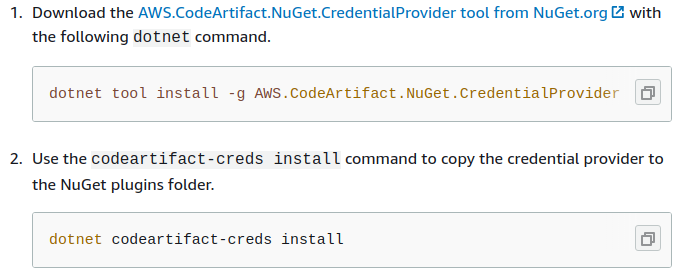


A

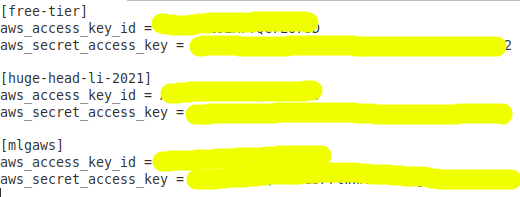
**Access the Repository from a Developer Computer**

To use the repository as a NugetT server and access it from a developer computer, we need to have dotnet SDK installed – https://dotnet.microsoft.com/download. In my case, I used the .NET 5.0. We can follow the instructions on this link to set up the development environment – <https://docs.aws.amazon.com/codeartifact/latest/ug/nuget-cli.html>.

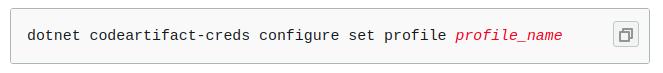
We can run the following command to install the credential provider for NUGET.



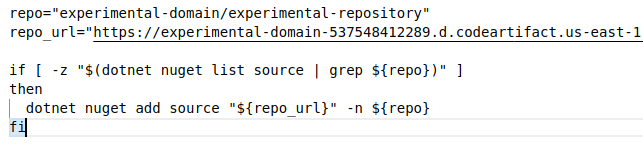
The credential provider will look in the “.aws/credentials” file for the actual user access keys to access the repository.



If you do not have a default profile or if you want to use a named profile explicitly, you need to run the following command to set the profile.



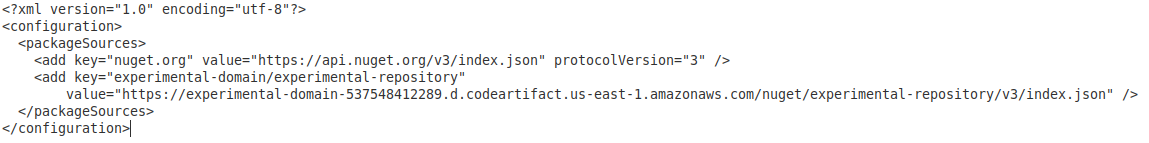
You will also need to run the following command to add the repository to your NUGET repository list.



The full repository URL is the following and you can easily see the pattern and its relation to the domain and the repository names.

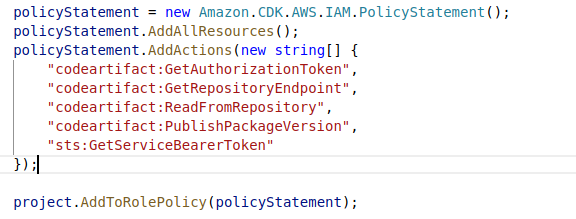
https://**experimental-domain**-537548412289.d.codeartifact.**us-east-1**.amazonaws.com/nuget/**experimental-repository**/v3/index.json

If the repository is successfully added to the NUGET repository list, you “NuGet.Config” file should look like something like this.



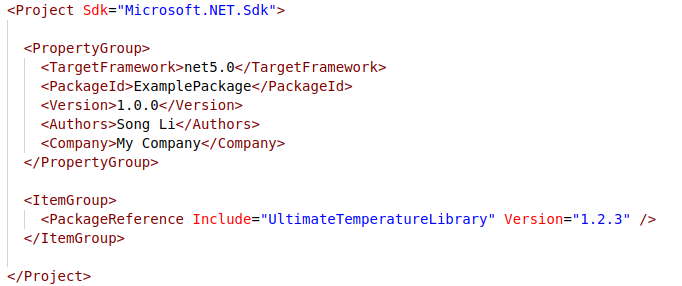
**Access the Repository from a CodeBuild Project**

In order to access the repository from the CodePipeline, we need to add the following permission to the CodeBuild project.

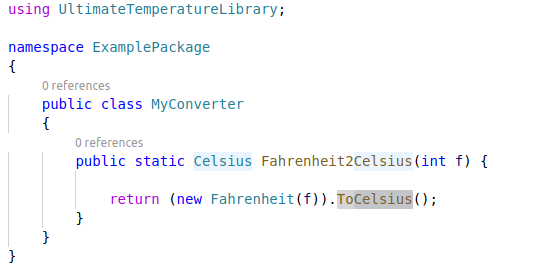


**Examples**

In order to test the CodeArtifact repository, I created a .NET package and a package client. The following is the package.



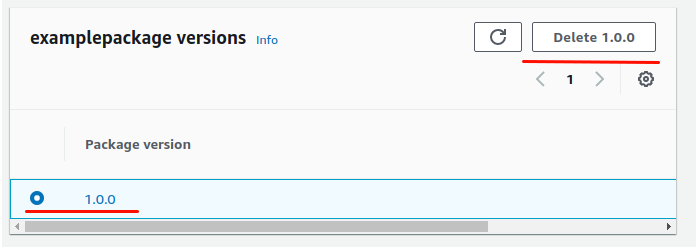
It created a simple method to convert a temperature from “Fahrenheit” to “Celsius”.



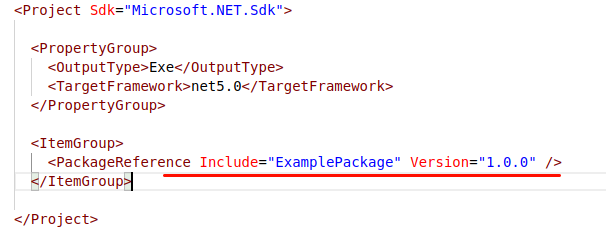
We can refer this link on how to create a NUGET package – <https://docs.microsoft.com/en-us/nuget/create-packages/creating-a-package-dotnet-cli>. And push the package to the repository by the follow command.

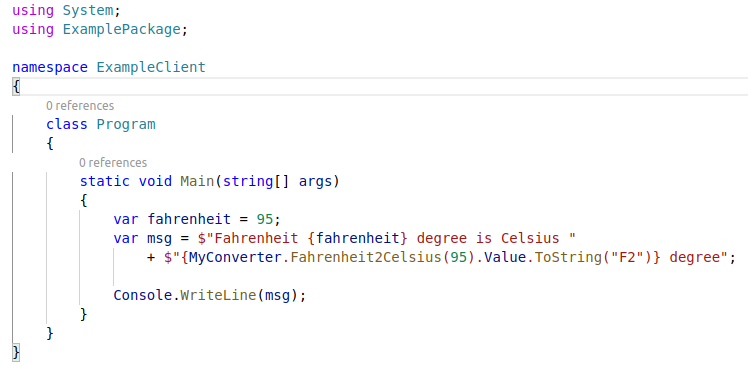


The CodeArtifact repository does not allow to push the same version of a package multiple times. If you want to replace a version, you need to delete the existing version first.



The package client is created as the following.





If everything does well, we can simply run the client by the “donet run” command. The NUGET package will be downloaded from the repository. The program will print out the output as the following.



**Conclusions**

In conclusion, the AWS CodeArtifact is easy to setup and easy to use. It works nicely in both a developer environment and AWS CodePipelines. It is an all-in-one repository for NUGET, NPM, PIP, Maven and other servers. Comparing with a stand-alone NUGET server, it has the following advantages.

* It operates in server-less fashion. Pay as it is used;
* It has virtually unlimited scalability;
* It has the same 11/9 availability and reliability as a AWS S3 bucket;
* It used IAM to manage the permission and security as other AWS services;
* It integrates nicely with other AWS services;
* etc.

**References**

* https://docs.microsoft.com/en-us/nuget/create-packages/creating-a-package-dotnet-cli
* https://docs.aws.amazon.com/codeartifact/latest/ug/nuget-cli.html
* https://docs.aws.amazon.com/codeartifact/latest/ug/codebuild.html
* https://docs.aws.amazon.com/codeartifact/latest/ug/using-nuget-packages-in-codebuild.html
* https://docs.aws.amazon.com/cli/latest/topic/config-vars.html
* [https://docs.aws.amazon.com/codeartifact/latest/ug/domain-policies.html#enabling-cross-acount-access-to-a-domain](https://docs.aws.amazon.com/codeartifact/latest/ug/domain-policies.html" \l "enabling-cross-acount-access-to-a-domain)
* https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-envvars.html