CREATE THE CONTROL

What are we going to do?

We are going to perform the following steps:

- define a product with a version and a portfolio in a hub account
- add the source code for the product
- provision that product into a spoke account

The hub AWS Account is the source of truth for our AWS Service Catalog products. Spoke AWS accounts are consumers of these products, you can think of them as accounts that need governance controls applied. For this workshop, we are using the same account as both the hub and spoke for simplicity; in a multi-account setup, these could be separate AWS Accounts and Regions.

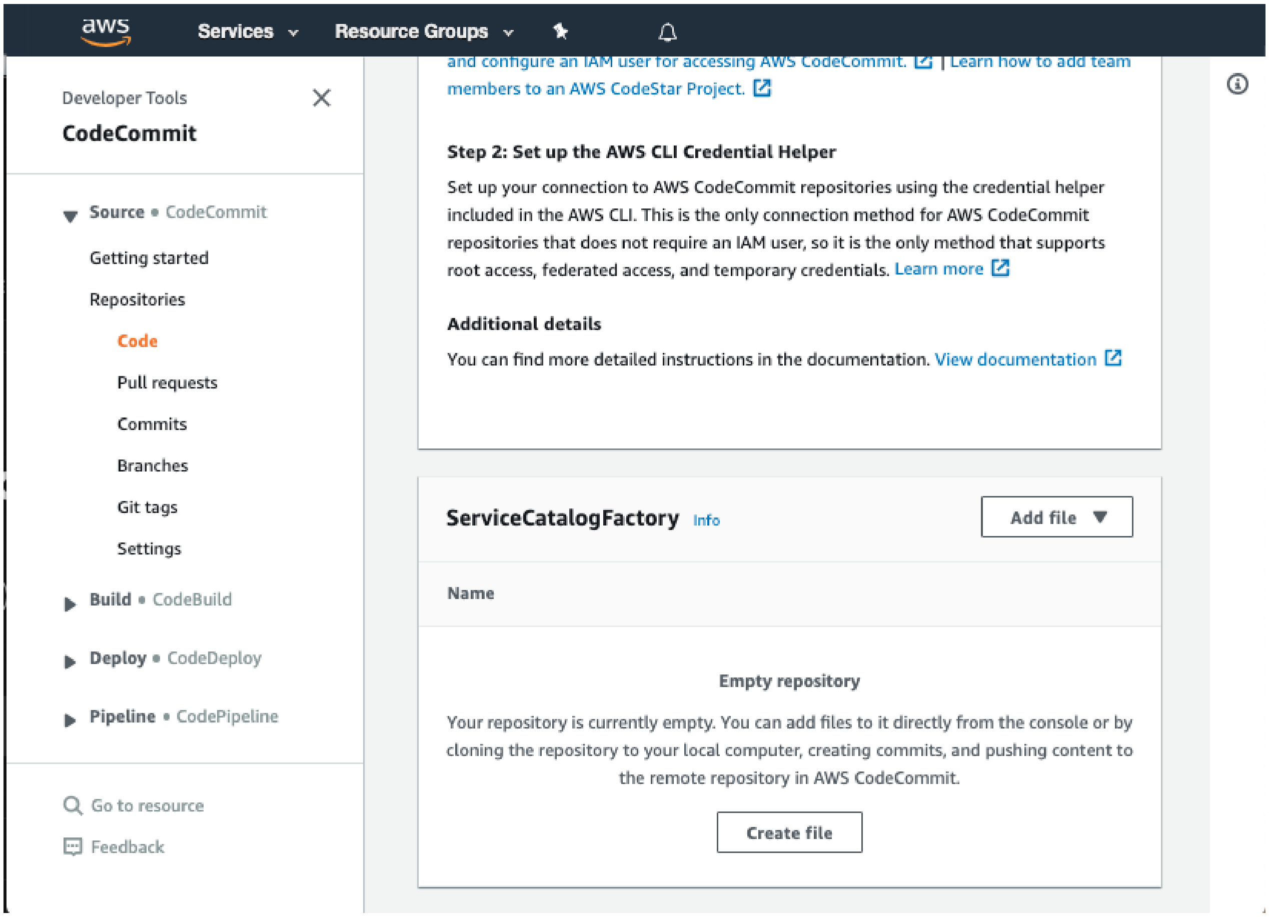
Step by step guide

Here are the steps you need to follow to "Create the control"

Define a product with a version and a portfolio

- Navigate to the ServiceCatalogFactory CodeCommit repository

- Scroll down to the bottom of the page and hit the *Create file* button



and configure an IAM user for accessing AWS CodeCommit. | Learn how to add team members to an AWS CodeStar Project.

Step 2: Set up the AWS CLI Credential Helper

Set up your connection to AWS CodeCommit repositories using the credential helper included in the AWS CLI. This is the only connection method for AWS CodeCommit repositories that does not require an IAM user, so it is the only method that supports root access, federated access, and temporary credentials. Learn more

Additional details

You can find more detailed instructions in the documentation. View documentation

ServiceCatalogFactory Info

Name

Empty repository

Your repository is currently empty. You can add files to it directly from the console or by cloning the repository to your local computer, creating commits, and pushing content to the remote repository in AWS CodeCommit.

- Copy the following snippet into the main input field:

```
Schema: factory-2019-04-01
Products:
  - Name: "aws-config-desired-instance-types"
    Owner: "budget-and-cost-governance@example.com"
    Description: "Enables AWS Config rule - desired-instance-type with our RIs"
    Distributor: "cloud-engineering"
    SupportDescription: "Speak to budget-and-cost-governance@example.com about exceptions and speak to cloud-engineering@example.com about implementation issues"
    SupportEmail: "cloud-engineering@example.com"
    SupportUrl: "https://wiki.example.com/cloud-engineering/budget-and-cost-governance/aws-config-desired-instance-types"
    Tags:
      - Key: "type"
        Value: "governance"
      - Key: "creator"
        Value: "cloud-engineering"
      - Key: "cost-center"
        Value: "governance"
    Versions:
      - Name: "v1"
```

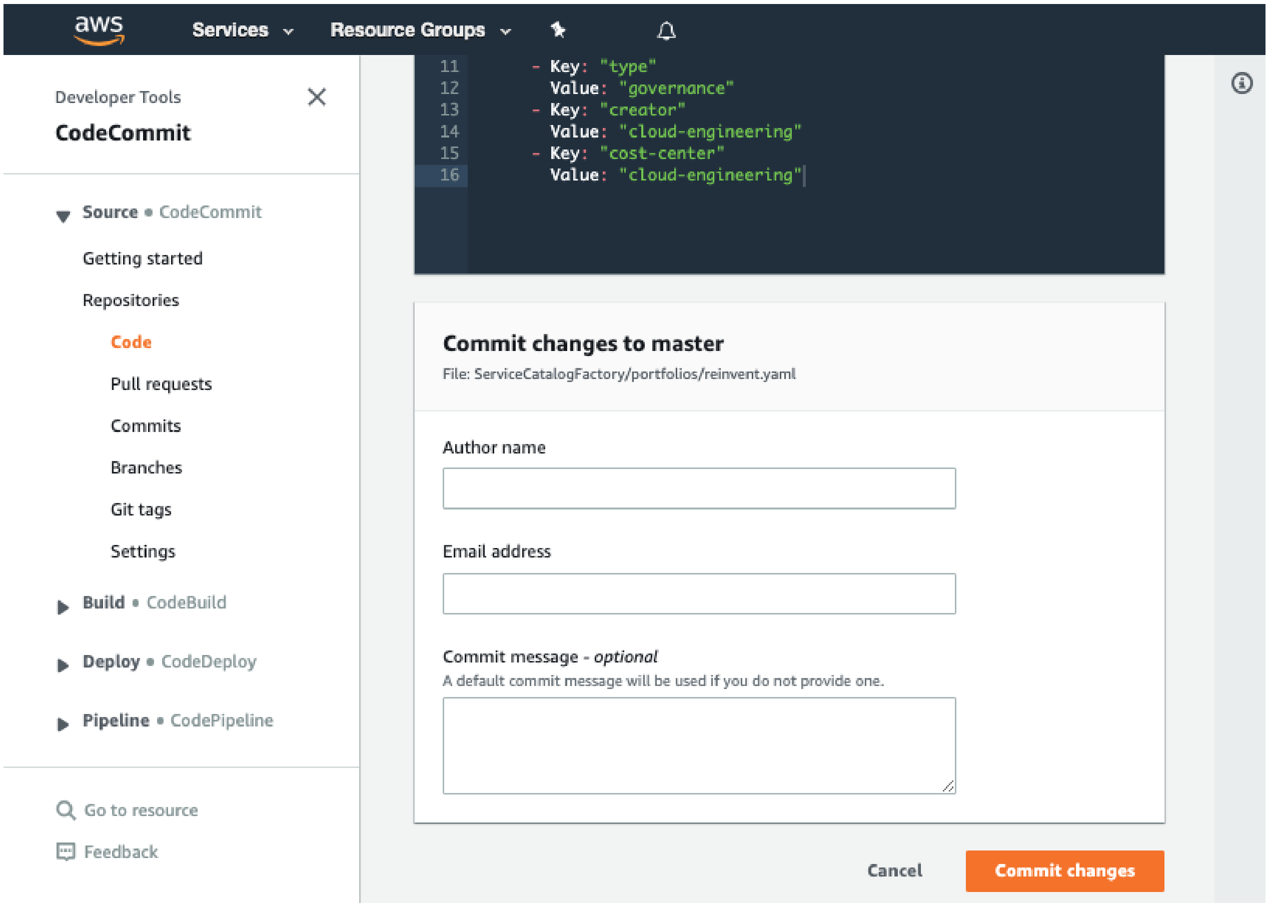
```
        Description: "v1 of aws-config-desired-instance-types"
        Active: True
        Source:
          Provider: "CodeCommit"
          Configuration:
            RepositoryName: "aws-config-desired-instance-types"
            BranchName: "master"
    Portfolios:
      - "cloud-engineering-governance"
Portfolios:
  - DisplayName: "cloud-engineering-governance"
    Description: "Portfolio containing the products needed to
govern AWS accounts"
    ProviderName: "cloud-engineering"
    Associations:
      - "arn:aws:iam::${AWS::AccountId}:role/TeamRole"
    Tags:
      - Key: "type"
        Value: "governance"
      - Key: "creator"
        Value: "cloud-engineering"
      - Key: "cost-center"
        Value: "governance"
```

- Set the *File name* to `portfolios/reinvent.yaml`
- Set your *Author name*
- Set your *Email address*
- Set your *Commit message*

Tip

Using a good / unique commit message will help you understand what is going on later.

- Click the *Commit changes* button:



```
        - Key: "type"
          Value: "governance"
        - Key: "creator"
          Value: "cloud-engineering"
        - Key: "cost-center"
          Value: "cloud-engineering"
```

Commit changes to master

File: ServiceCatalogFactory/portfolios/reinvent.yaml

Author name

Email address

Commit message - *optional*

A default commit message will be used if you do not provide one.

What did we just do?

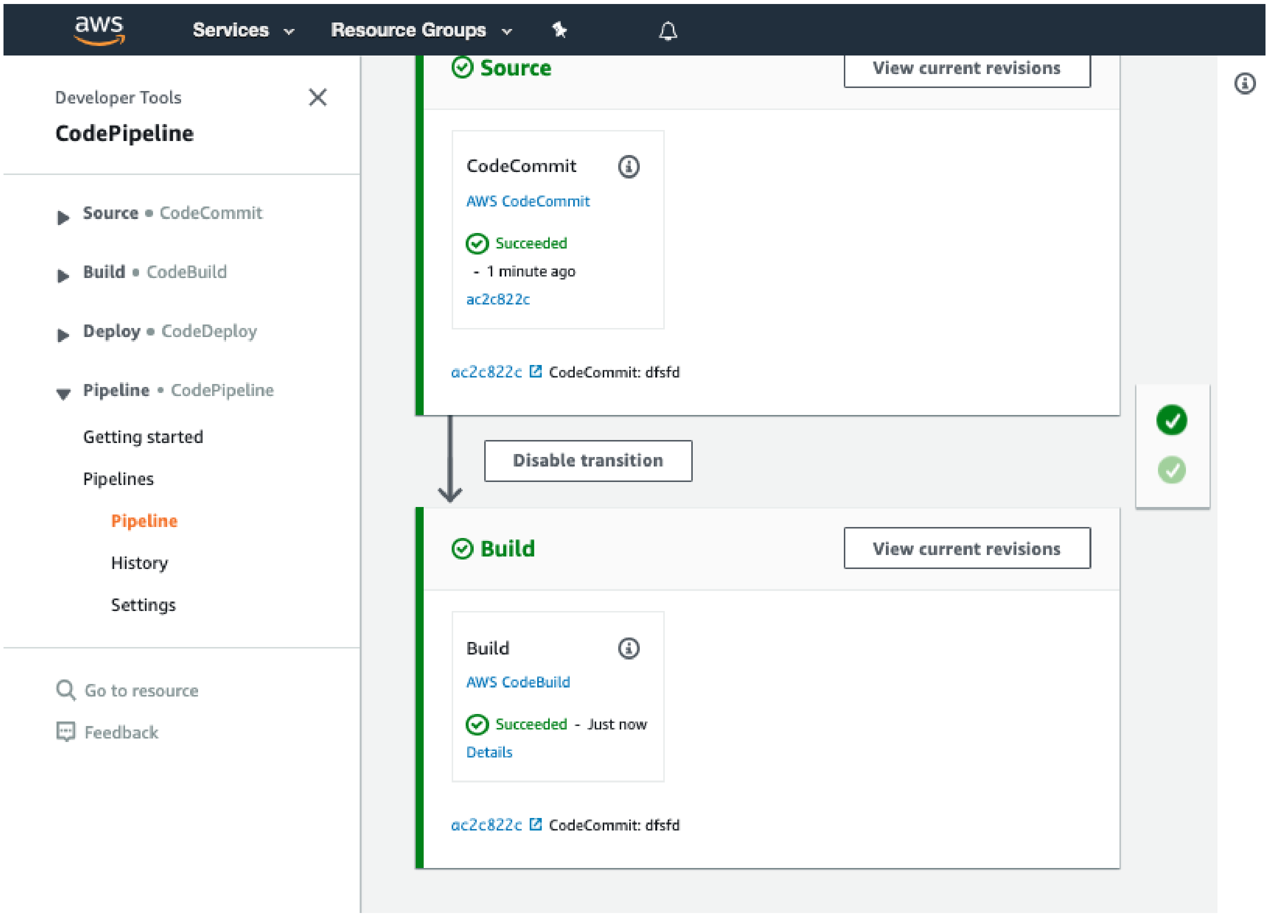
The YAML file we created in the CodeCommit repository told the framework to perform several actions:

- create a product named *aws-config-desired-instance-types*
- add a *v1* of our product
- create a portfolio named *cloud-engineering-governance*
- add the product: *aws-config-desired-instance-types* to the portfolio: *cloud-engineering-governance*

Verify the change worked

Once you have made your changes the ServiceCatalogFactory Pipeline should have run. If you were very quick in making the change, the pipeline may still be running. If it has not yet started feel free to the hit the *Release change* button.

Once it has completed it should show the *Source* and *Build* stages in green to indicate they have completed successfully:



Note

If this is failing please raise your hand for some assistance

Add the source code for our product

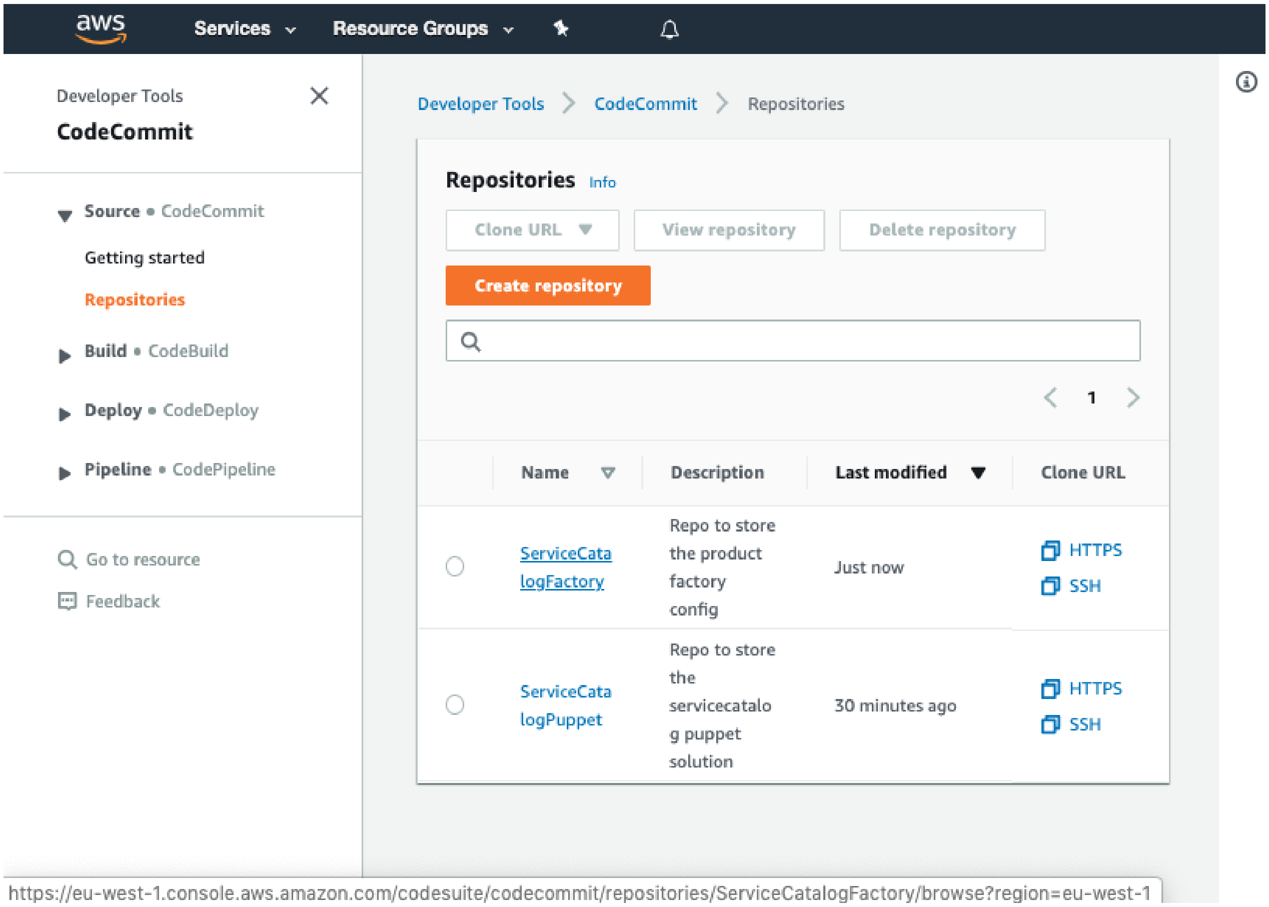
When you configured your product version, you specified the following version:

```
Versions:
  - Name: "v1"
    Description: "v1 of aws-config-desired-instance-types"
    Active: True
    Source:
      Provider: "CodeCommit"
      Configuration:
        RepositoryName: "aws-config-desired-instance-types"
        BranchName: "master"
```

This tells the framework the source code for the product comes from the *master* branch of a *CodeCommit* repository of the name *aws-config-desired-instance-types*.

We now need to create the CodeCommit repository and add the AWS CloudFormation template we are going to use for our product.

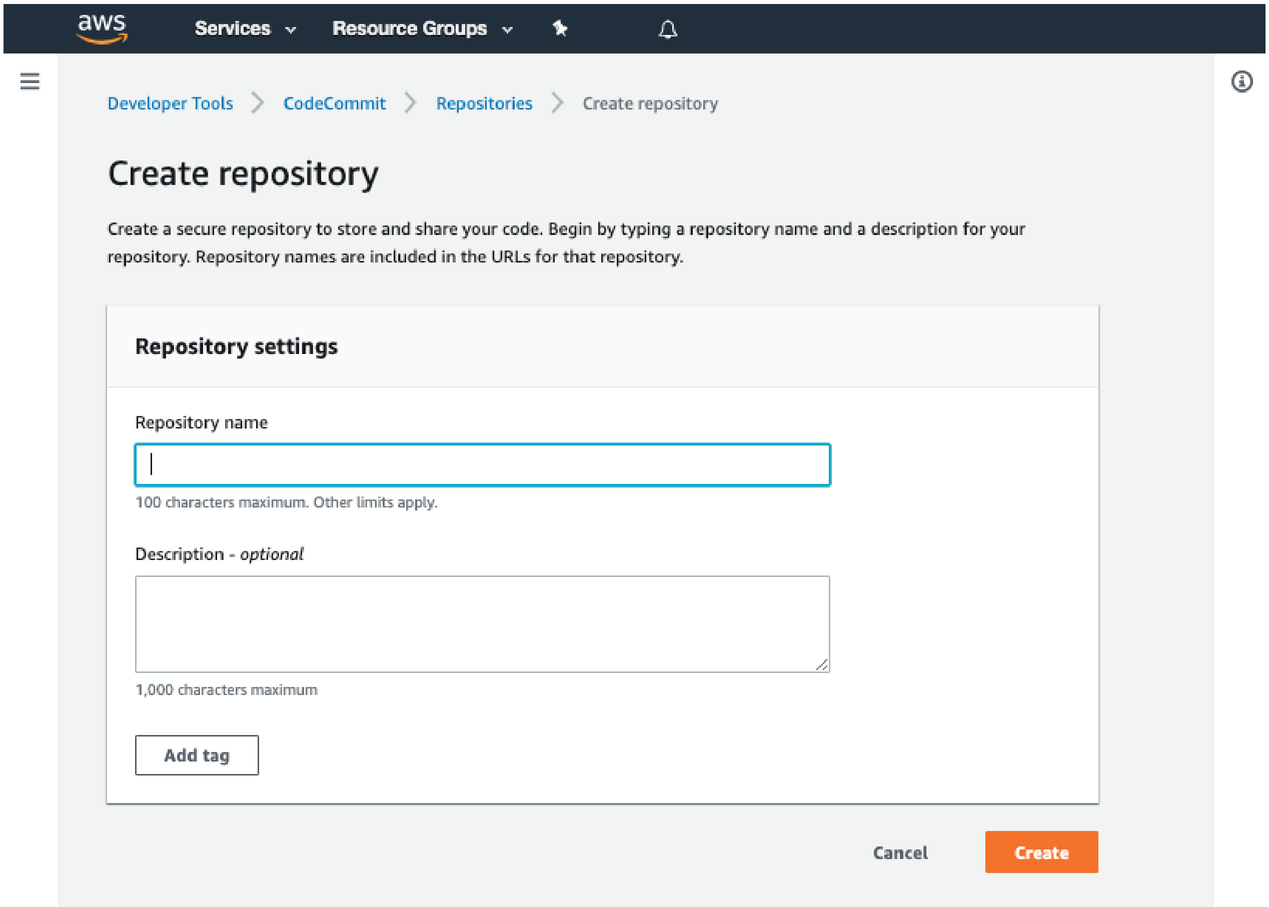
- Navigate to AWS CodeCommit
- Click *Create repository*



	Name	Description	Last modified	Clone URL
○	ServiceCatalogFactory	Repo to store the product factory config	Just now	HTTPS SSH
○	ServiceCatalogPuppet	Repo to store the servicecatalog puppet solution	30 minutes ago	HTTPS SSH

https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/ServiceCatalogFactory/browse?region=eu-west-1

- Input the name `aws-config-desired-instance-types`



Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Repository settings

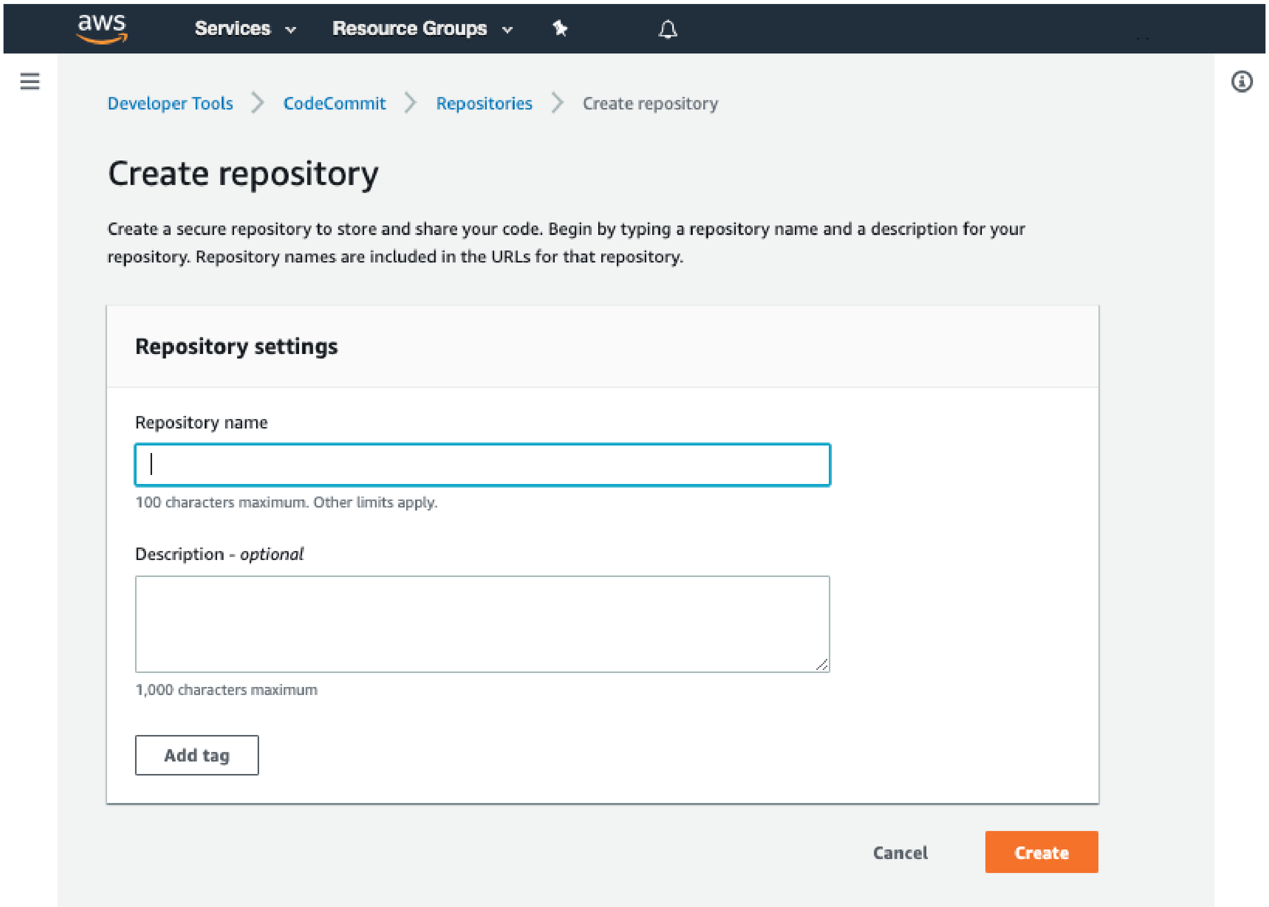
Repository name

100 characters maximum. Other limits apply.

Description - *optional*

1,000 characters maximum

- Click *Create*



aws
Services
Resource Groups
Developer Tools > CodeCommit > Repositories > Create repository

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Repository settings

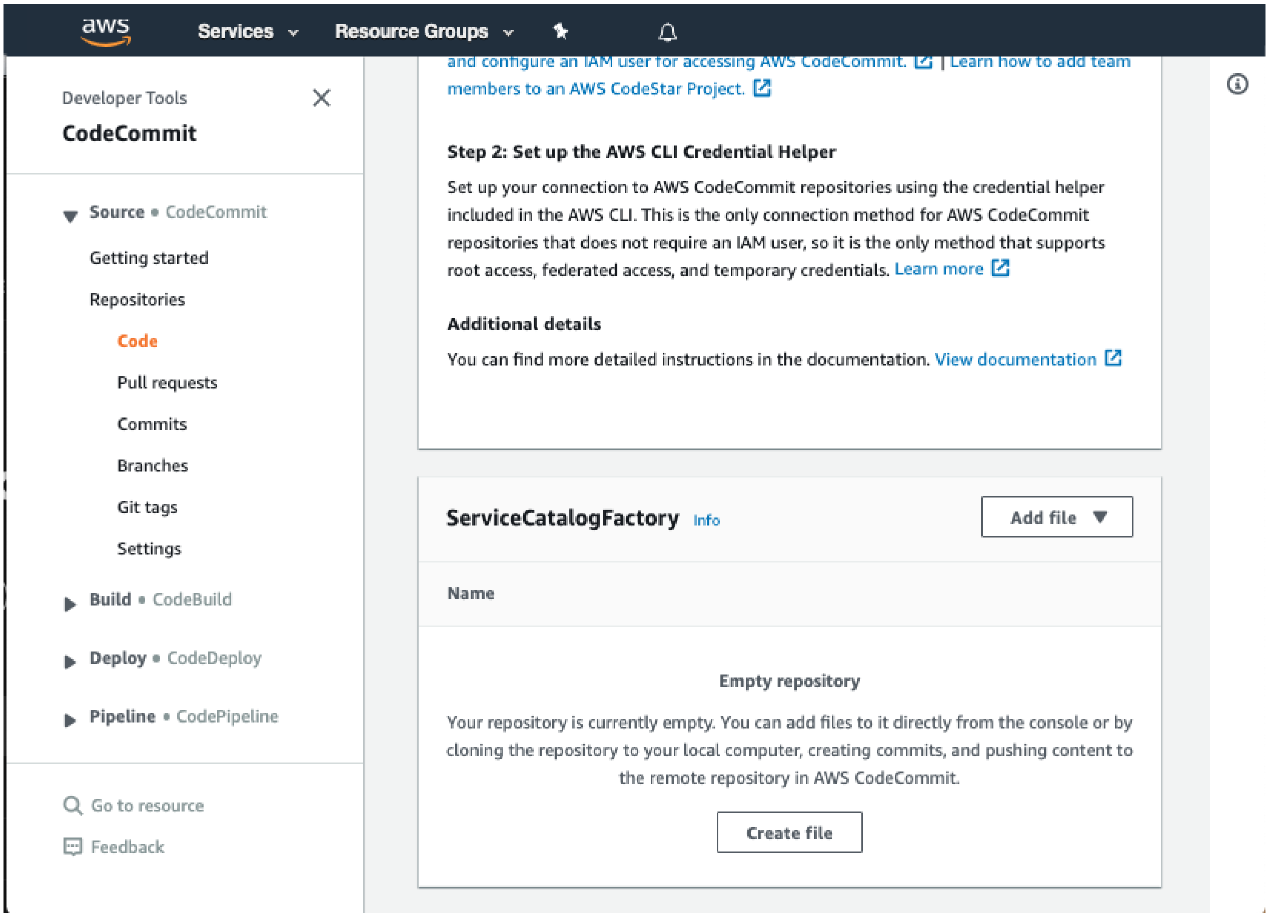
Repository name

100 characters maximum. Other limits apply.

Description - *optional*

1,000 characters maximum

- Scroll down to the bottom of the page and hit the *Create file* button



and configure an IAM user for accessing AWS CodeCommit. | Learn how to add team members to an AWS CodeStar Project.

Step 2: Set up the AWS CLI Credential Helper

Set up your connection to AWS CodeCommit repositories using the credential helper included in the AWS CLI. This is the only connection method for AWS CodeCommit repositories that does not require an IAM user, so it is the only method that supports root access, federated access, and temporary credentials. Learn more

Additional details

You can find more detailed instructions in the documentation. View documentation

ServiceCatalogFactory Info

Add file ▼

Name

Empty repository

Your repository is currently empty. You can add files to it directly from the console or by cloning the repository to your local computer, creating commits, and pushing content to the remote repository in AWS CodeCommit.

Create file

- Copy the following snippet into the main input field:

```
AWSTemplateFormatVersion: "2010-09-09"
Description: "Create an AWS Config rule ensuring the given
instance types are the only instance types used"

Parameters:
  InstanceType:
    Type: String
    Description: "Comma separated list of EC2 instance types (for
example, 't2.small, m4.large')."
    Default: "t2.micro, t2.small"

Resources:
  AWSConfigRule:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: "desired-instance-type"
      Description: "Checks whether your EC2 instances are of the
specified instance types."
      InputParameters:
        instanceType: !Ref InstanceType
      Scope:
        ComplianceResourceTypes:
          - "AWS::EC2::Instance"
```

```
            Source:
              Owner: AWS
              SourceIdentifier: DESIRED_INSTANCE_TYPE
```

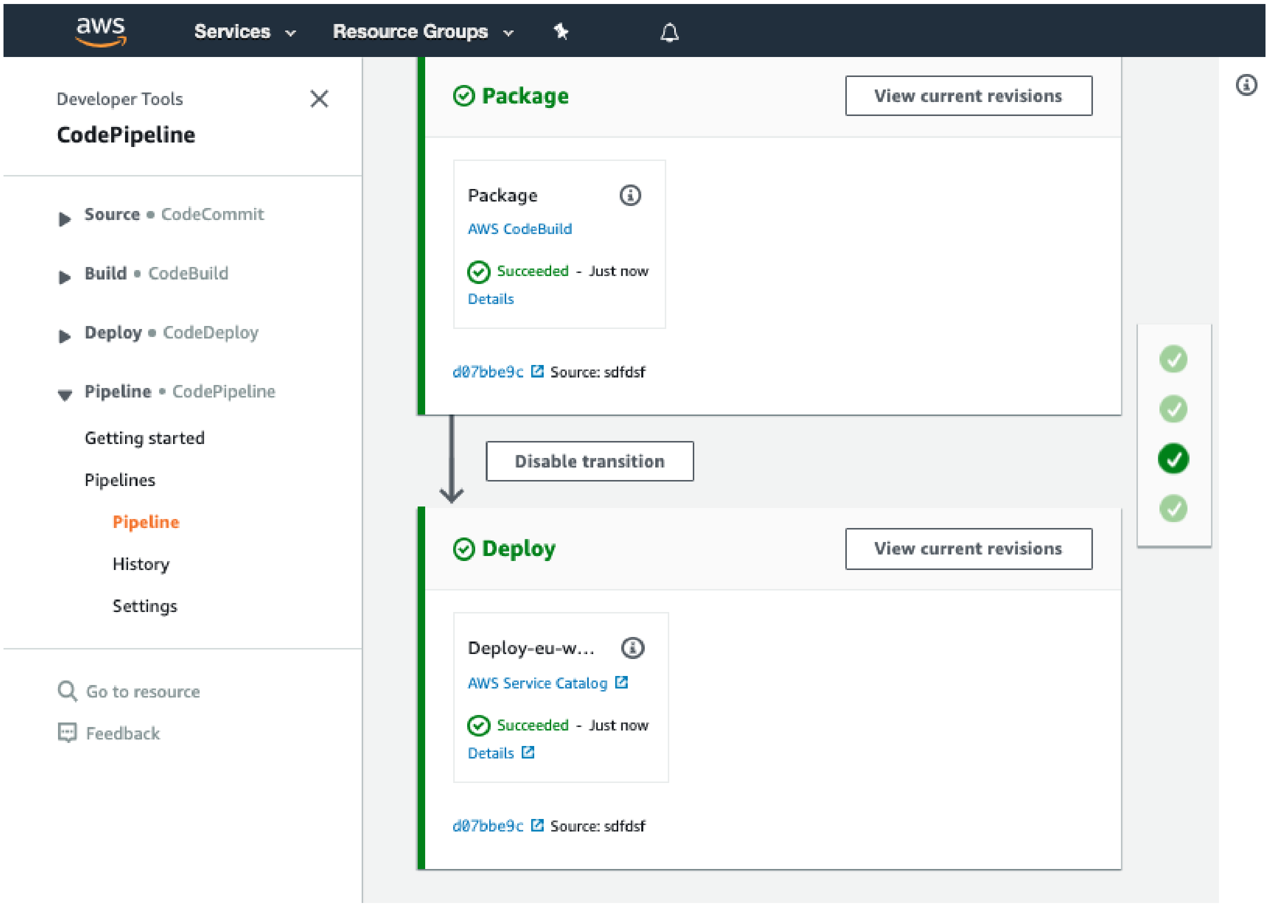
- Set the *File name* to `product.template.yaml`
- Set your *Author name*
- Set your *Email address*
- Set your *Commit message*

Tip

Using a good / unique commit message will help you understand what is going on later.

Creating that file should trigger your aws-config-desired-instance-types-v1-pipeline.

Once the pipeline has completed it should show the *Source*, *Tests*, *Package* and *Deploy* stages in green to indicate they have completed successfully:



Tip

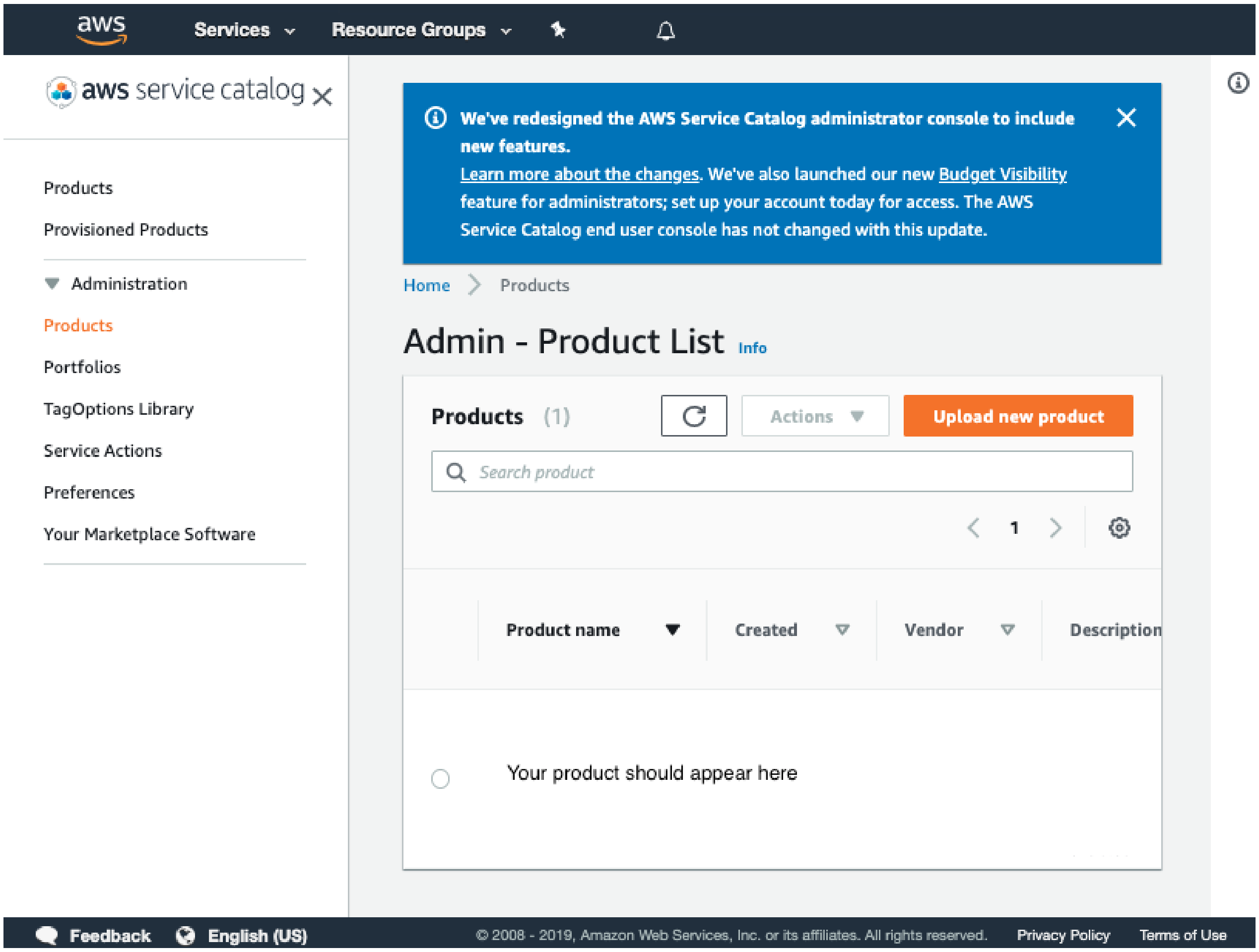
You should see your commit message on this screen, it will help you know which version of ServiceCatalogFactory repository the pipeline is processing.

Note

If this is failing please raise your hand for some assistance

Once you have verified the pipeline has run you can go to Service Catalog products to view your newly created version.

You should see the product you created listed:



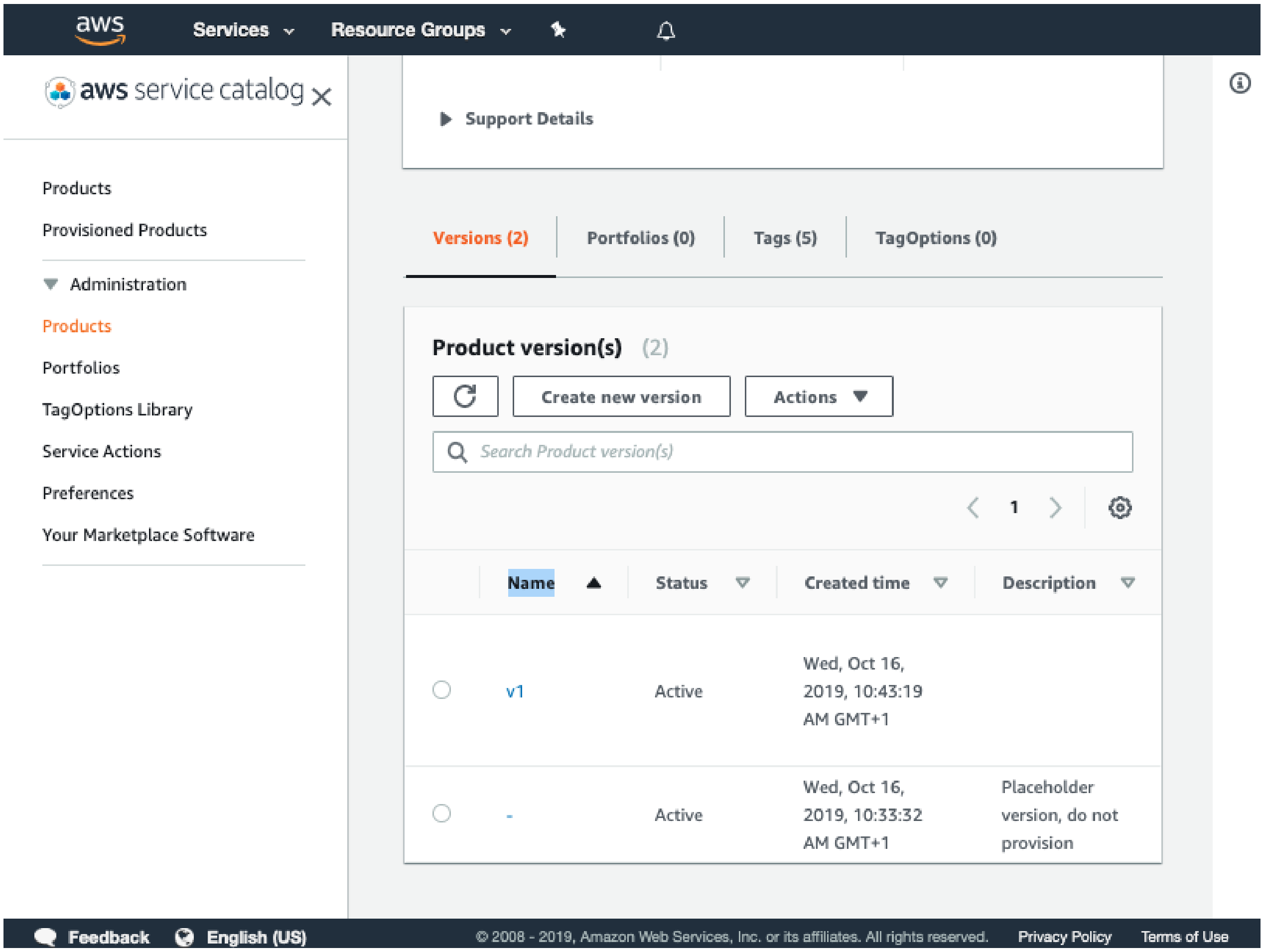
We've redesigned the AWS Service Catalog administrator console to include new features. Learn more about the changes. We've also launched our new Budget Visibility feature for administrators; set up your account today for access. The AWS Service Catalog end user console has not changed with this update.

Admin - Product List

○ Your product should appear here

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Click on the product and verify *v1* is there



Product version(s) (2)

	Name	Status	Created time	Description
○	v1	Active	Wed, Oct 16, 2019, 10:43:19 AM GMT+1	
○	-	Active	Wed, Oct 16, 2019, 10:33:32 AM GMT+1	Placeholder version, do not provision

Note

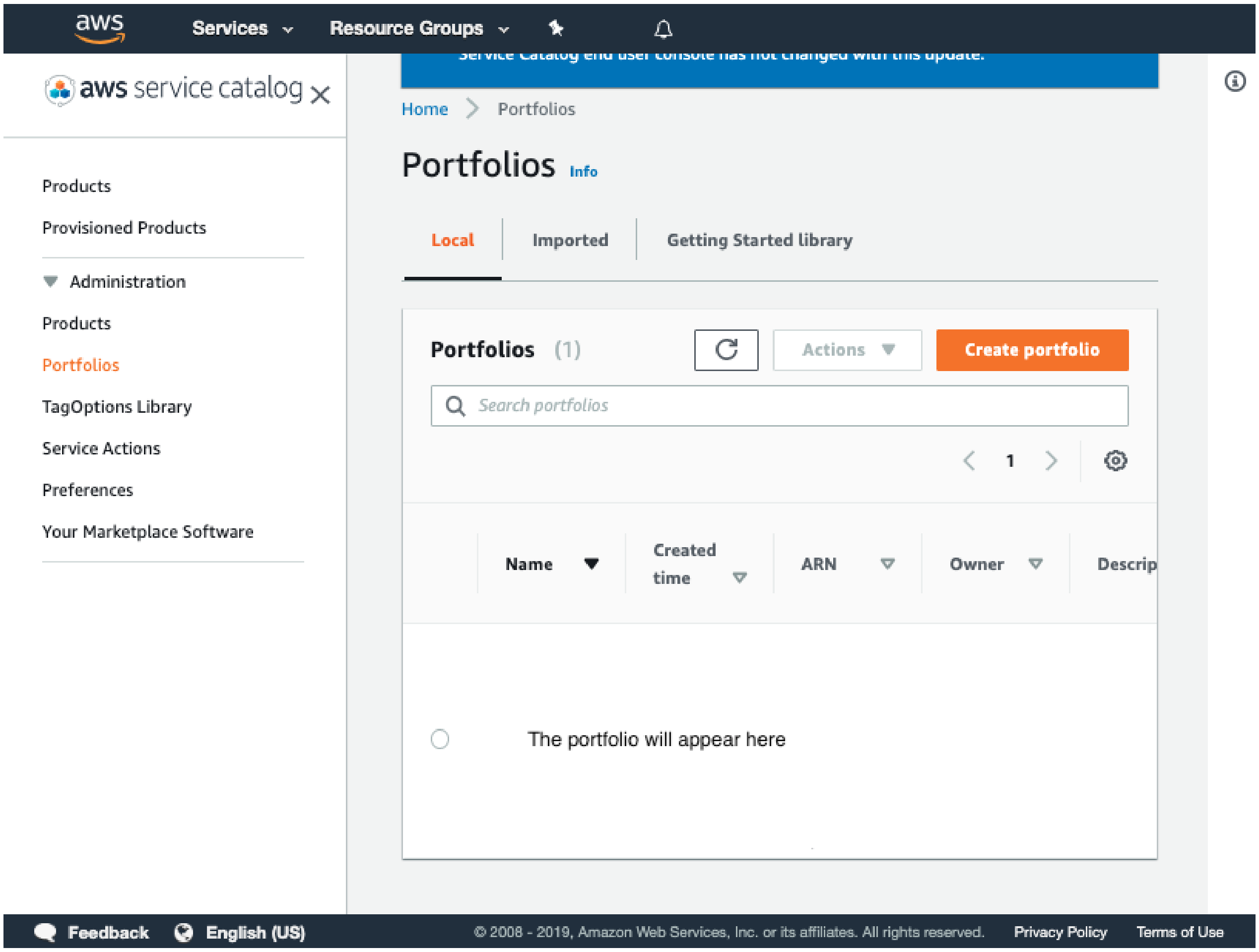
If you cannot see your version please raise your hand for some assistance

You have now successfully created a version for your product!

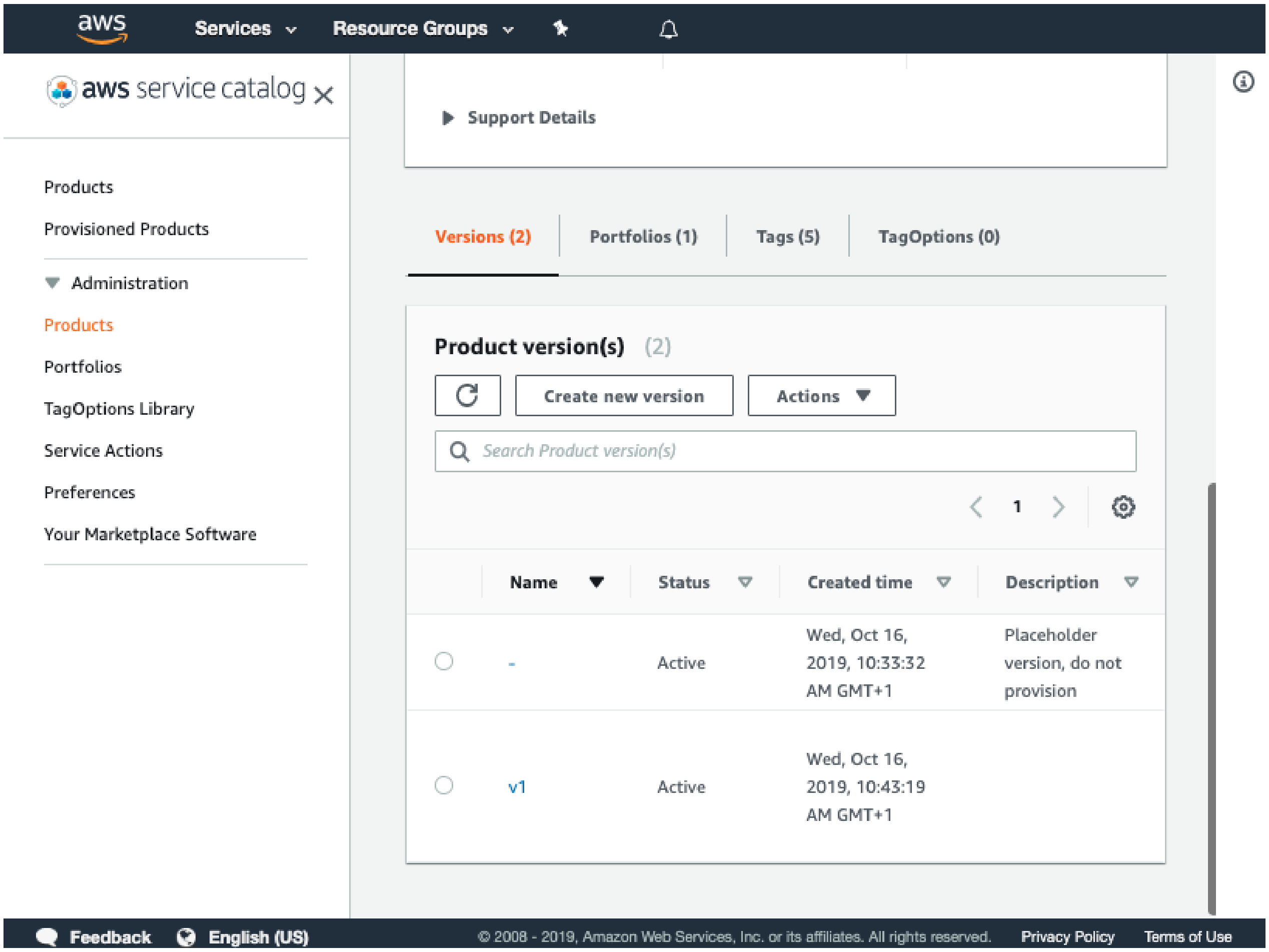
Verify the product was added to the portfolio

Now that you have verified the pipeline has run you can go to Service Catalog portfolios to view your portfolio.

- Click on *reinvent-cloud-engineering-governance*



- Click on the product *aws-config-desired-instance-types*
- Click on the version *v1*



Product version(s) (2)

	Name	Status	Created time	Description
○	-	Active	Wed, Oct 16, 2019, 10:33:32 AM GMT+1	Placeholder version, do not provision
○	v1	Active	Wed, Oct 16, 2019, 10:43:19 AM GMT+1	