**Blog Owners:**

* **prajjgup@ Prajjaval Gupta**
* **sanbrar@ San Brar**

# Using AWS Service Catalog and the AWS Service Management Connector for ServiceNow to deliver infrastructure in AMS-governed environments

[Amazon Managed Services](https://aws.amazon.com/managed-services/) (AMS) operates AWS on your behalf, providing a secure and compliant AWS landing zone, a proven enterprise operating model, ongoing cost optimization, and day-to-day infrastructure management.

AMS provides a secure and efficient means to make controlled changes to your infrastructure to ensure compliance. Changes are approved and automated through its approval engine. You can schedule changes through the console.

[AWS Service Catalog](https://aws.amazon.com/servicecatalog/) allows organizations to create and manage catalogs of IT services that are approved for use on AWS. AMS recently added AWS Service Catalog to their offerings, as an alternative to the AMS change management system.

In this blog post, we describe how you can use AWS Service Catalog to deliver infrastructure in AMS-governed environments through CI/CD pipelines. The ServiceNow workflow triggers the pipelines, which are deployed through the integration of ServiceNow and AWS Service Catalog.

## Solution overview

This solution shows how to perform automated infrastructure deployment to an AMS managed account from a non-AMS account (the source AWS account). The development tool (in our solution, [AWS CodePipeline](https://aws.amazon.com/codepipeline/)) is running in the source AWS account. The account is used to register the product package in AWS Service Catalog and ServiceNow.

The CI/CD pipeline assumes a cross account role to register and provision the product in AWS Service Catalog in the AMS account. After the product has been successfully registered in AWS Service Catalog, the product is deployed in the AMS account.

The solution covers the following:

* Automated deployment of infrastructure using CI/CD pipelines to AMS.
* Use of AWS Service Catalog instead of the AMS console.
* ServiceNow integration with CI/CD pipelines to orchestrate deployments and rollbacks.

After the AWS Service Catalog product has been packaged and uploaded to an [Amazon Simple Storage Service](https://aws.amazon.com/s3/) (Amazon S3) bucket, the CI/CD pipeline is triggered. The pipeline registers the product in AWS Service Catalog in the source AWS account. Each version of the deployed package is registered and made available in ServiceNow for future deployments and rollbacks.

Figure 1 shows the six steps of the solution process.

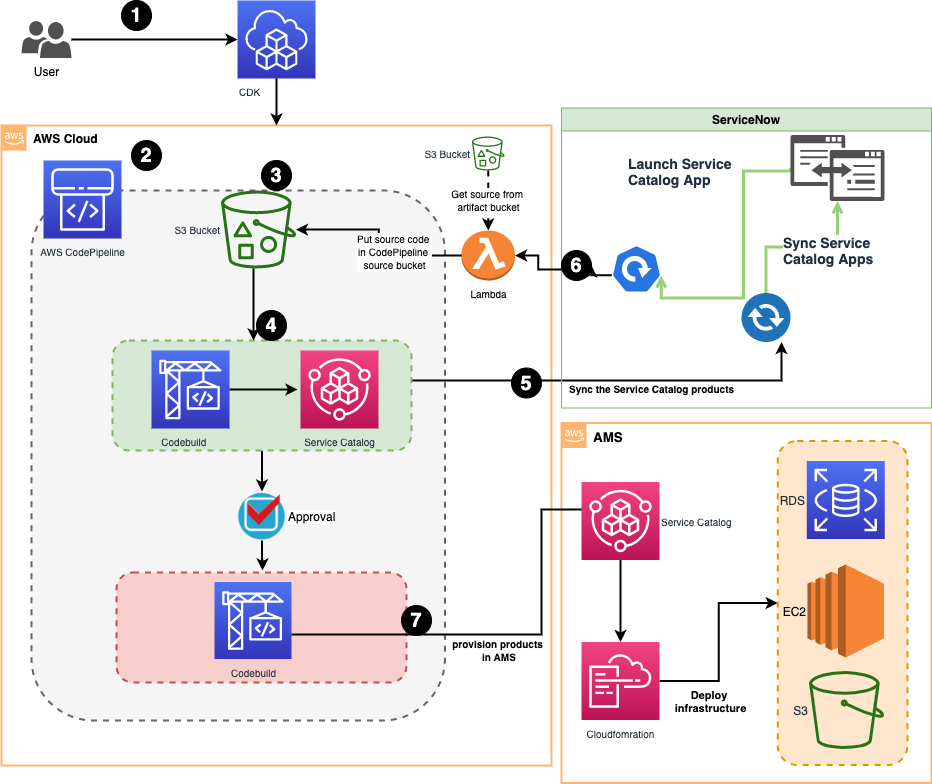


Figure 1: Solution architecture

1. The user uses the [AWS Cloud Development Kit](https://aws.amazon.com/cdk/) to deploy this solution in their source AWS account.
2. AWS CodePipeline orchestrates deployments to the AMS account and updates the AWS Service Catalog portfolio.
3. The user uploads the deployment package to an S3 bucket. An S3 event is triggered, which starts the pipeline.
4. A project in [AWS CodeBuild](https://aws.amazon.com/codebuild/) registers the product in the AWS Service Catalog portfolio.
5. After the product is registered in AWS Service Catalog, the [AWS Service Management Connector for ServiceNow](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/integrations-servicenow.html) syncs the portfolio with ServiceNow. This synchronization makes it possible to deploy any registered version of the product from ServiceNow. The connector uses a IAM user, it is recommended that access keys for the users are rotated on an regular basis. For details, see [Rotating Access Keys (AWS CLI, Tools for Windows PowerShell, and AWS API)](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html#Using_RotateAccessKey) in the *IAM User Guide* and [How to Rotate Access Keys for IAM Users](https://aws.amazon.com/blogs/security/how-to-rotate-access-keys-for-iam-users/) on the AWS Security Blog.
6. When the product is also deployed from ServiceNow, the AWS Service Management Connector for ServiceNow triggers a AWS CloudFormation template, triggering a [AWS Lambda](https://aws.amazon.com/lambda/) to copy the source from an artifact bucket to the source code CodePipeline bucket.
7. The AWS CodeBuild project uses AWS Service Catalog in the AMS account to register and provision the product.

## Solution prerequisites

Before you begin, you need the following:

* An AWS account. This is the non-AMS account that is used to deploy the CI/CD pipeline.
* Git client.
* An [AMS account](https://aws.amazon.com/managed-services/).
* [AMS CloudFormation deployment requirements.](https://aws.amazon.com/managed-services/)
* The AWS CLI v2, see [Installing, updating, and uninstalling the AWS CLI](https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html)
* Node.Js, see [Setting up your Node.js development environment](https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/nodejs-devenv.html)
* The AWS CDK (TypeScript), see [Install AWS CDK](https://docs.aws.amazon.com/cdk/latest/guide/getting_started.html#getting_started_install). For more information on AWS CDK see [Getting started with the AWS CDK](https://docs.aws.amazon.com/cdk/latest/guide/getting_started.html).
* [AWS Service Management Connector for ServiceNow](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/integrations-servicenow.html)

## Create the solution

1. In the AMS account, create an RFC to request the cross-account role. For the IAM role name, enter ci\_cd\_ams\_cross\_account\_role. For trusted entity, enter arn:aws:iam::<SourceAWSAccountId>:role/CI-CD-Pipeline. For the managed policy to attach, enter arn:aws:iam::<AMSAccountId>:policy/AMSCustomerServiceCatalogAdminPolicy. For see [Change Management, RFCs, change types, and examples](https://docs.aws.amazon.com/managedservices/latest/ams-intro/change-mgmt.html) to learn about RFCs.
2. In the AMS account, create an RFC to create the S3 bucket. The bucket name prefix must be sc-.
3. Clone the [Git repository for this solution](https://github.com/aws-samples/ci-cd-servicecatalog-ams).

Git clone https://github.com/aws-samples/ci-cd-servicecatalog-ams.git

1. Using the source AWS account CLI credentials, build and deploy the CDK application in the source AWS account. Replace the “source\_account\_profile” with the name of the source AWS account CLI profile. See [Named profiles](https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-profiles.html) to learn more about CLI profiles.

cd to application root directory

npm install

npm run build

cdk bootstrap --profile source\_account\_profile

cdk deploy ams-pipeline \

--parameters crossAccountRole=" arn:aws:iam::<AMSAccountId>:role/ci\_cd\_ams\_cross\_account\_role" \

--parameters LaunchRole="arn:aws:iam::<AMSAccountId>:role/AWSManagedServicesService CatalogLaunchRole" \

--parameters bucketname="<Name of the bucket created in step 2>"

1. Build the IAM users and role to be used by ServiceNow to sync the AWS Service Catalog products using the AWS Service Management Connector for ServiceNow.

cdk deploy SnowRoles

1. Create the access keys for two users (SCSync and SCEndUser) that were created by the SnowRoles stack in the previous step. See, [create an access key for an IAM user](https://docs.aws.amazon.com/cli/latest/userguide/cli-services-iam-create-creds.html) on how to create access keys.
2. Follow the [configuration instructions](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/integrations-servicenow.html) in the AWS Service Catalog Administration Guide to integrate ServiceNow with AWS Service Catalog in the source AWS account. Use the access keys created in the previous step.

## Test the solution

1. Upload the test product included in the solution to the sc-bucket-<SourceAWSAccountId> bucket by running the following commands.

cd src

zip package.zip serviceCatalog.py version.json cloudformation.json buildspec.yml ( Windows user please use manual steps to zip the files )

aws s3 cp package.zip sc-bucket-<SourceAWSAccountId>

The version.json file contains the app name and build version number.

{

“AppName”: “CreateS3BucketProduct”,

“Version”: “4.0”

}

**Note**: AMS only supports JSON [AWS CloudFormation](https://aws.amazon.com/cloudformation/) templates.

Add the required AMS object to your template, as shown here:

"Transform":{"Name":"AmsStackTransform","Parameters":{"StackId":{"Ref":"AWS::StackId"}}}

1. The pipeline execution starts as soon as you upload an artifact to the S3 bucket. You can monitor its progress in the AWS Code Pipeline console.

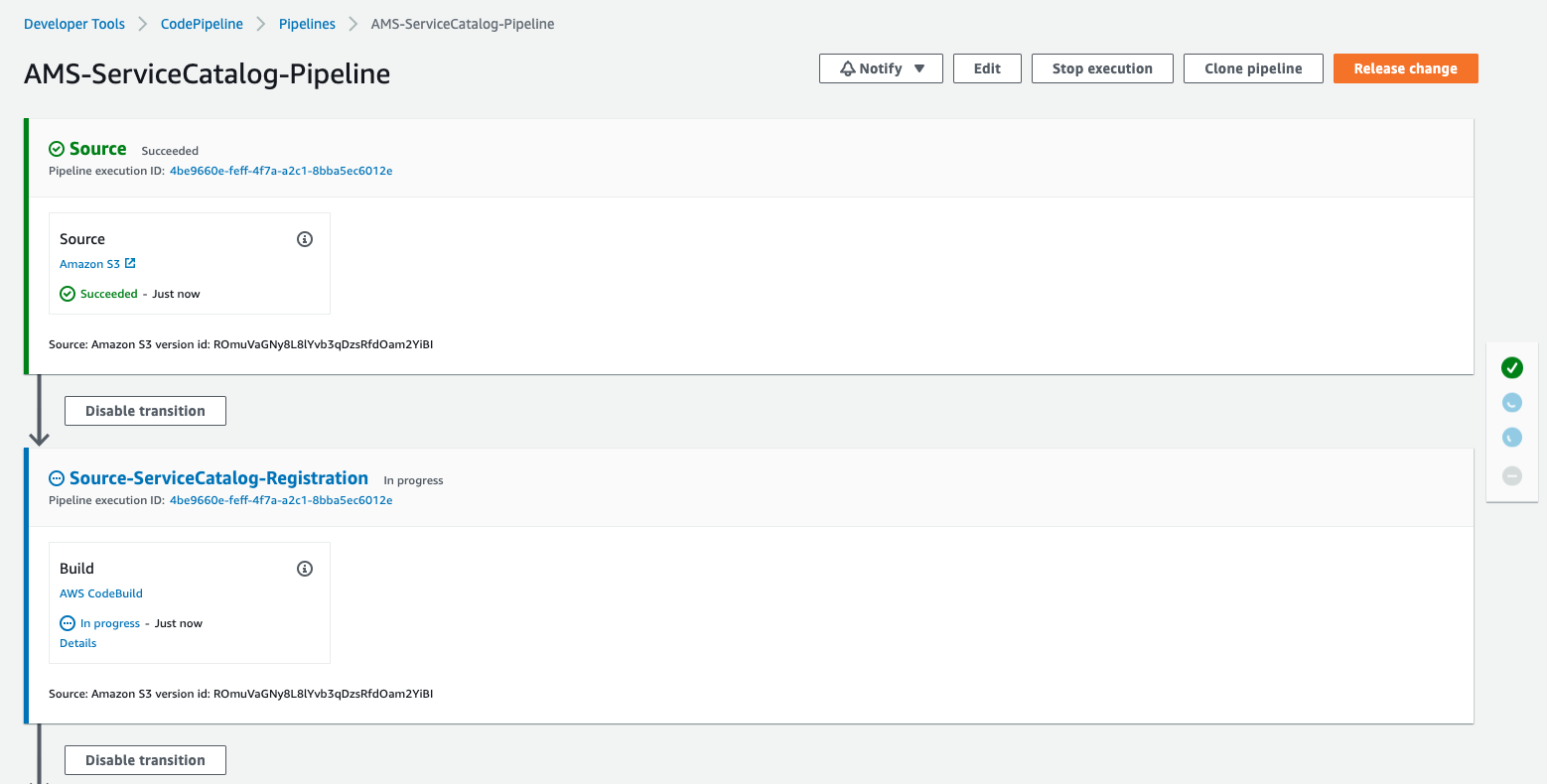


Figure 2: CodePipeline execution

1. The pipeline creates and registers the product in the AWS Service Catalog portfolio of the source account. The AWS Service Management Connector for ServiceNow syncs the portfolio and the products into ServiceNow. Verify that the product is available in ServiceNow.

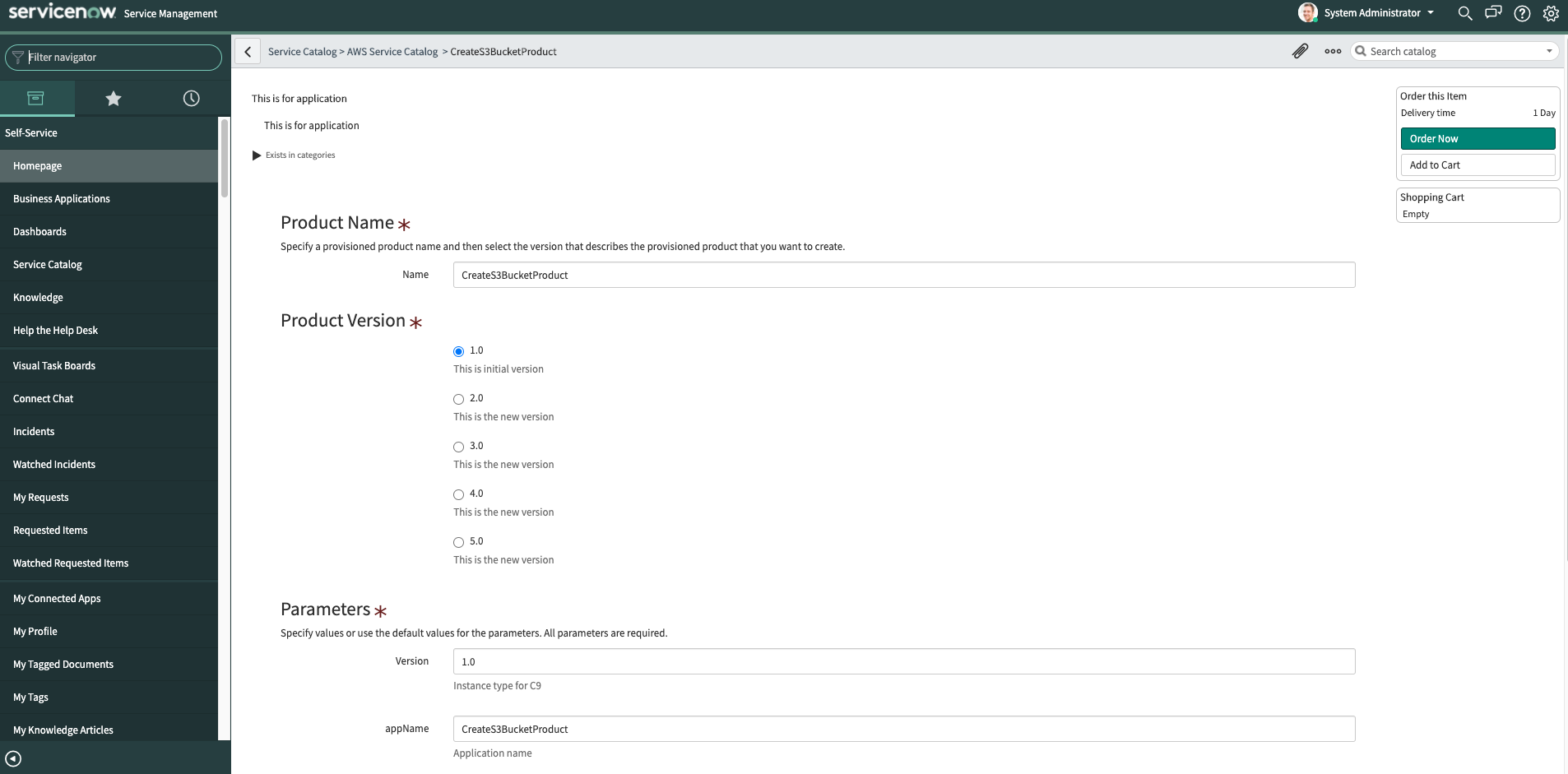


Figure 3: CreateS3BucketProduct displayed in ServiceNow

After the portfolio is synchronized, ServiceNow can trigger a product deployment or rollback to a specific build version through a pipeline.

1. In last stage of the pipeline, AWS CodeBuild assumes a cross account role and creates or updates the product in the AWS Service Catalog portfolio in the AMS account. After the product is registered, the product provisioning begins.

Figure 4 shows the successful completion of the pipeline.

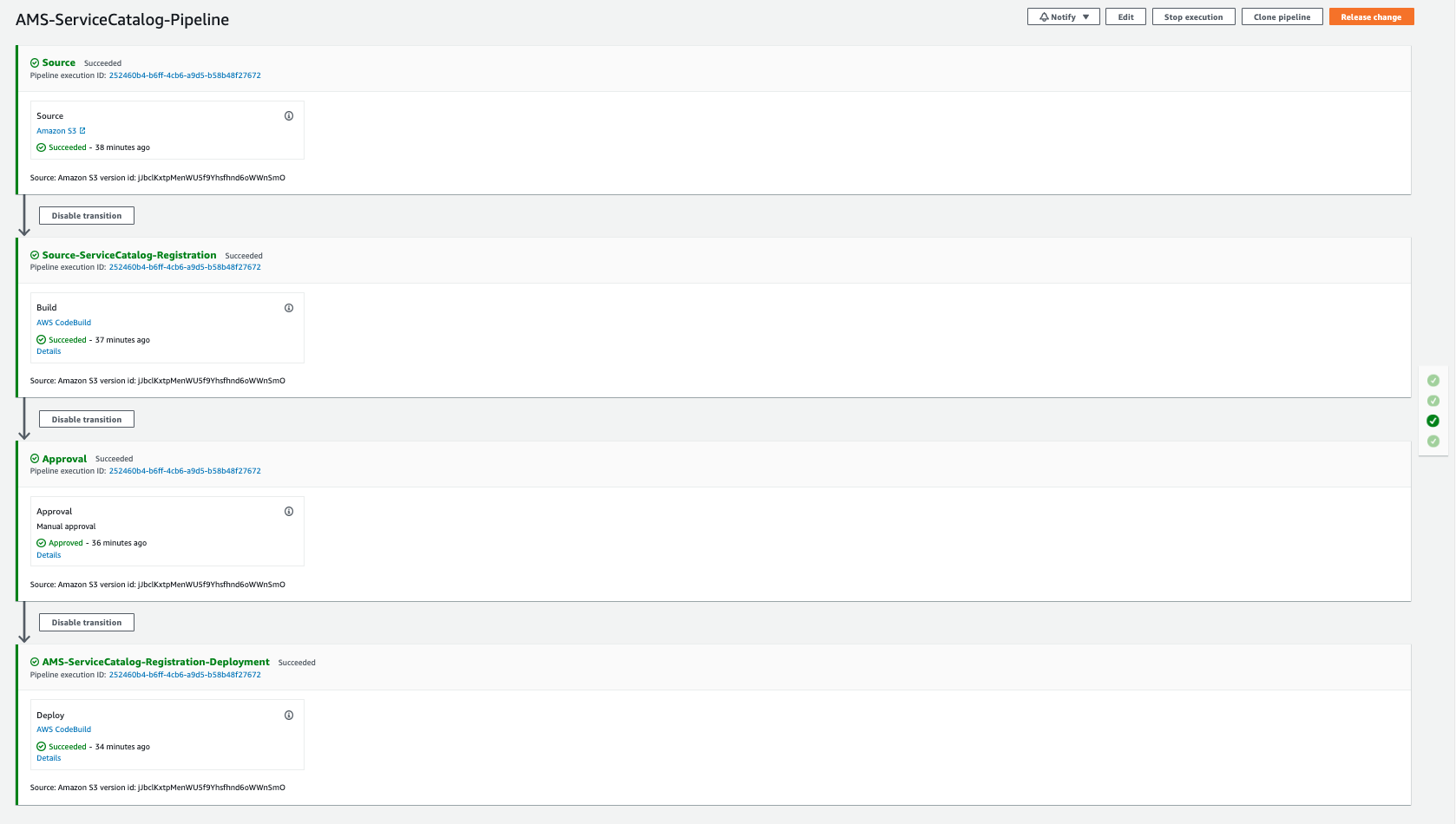


Figure 4: Successful completion of the pipeline

## Cleanup

To avoid ongoing charges to your account, delete the resources created by this solution.

cdk destroy ams-pipeline

cdk destroy snowRoles

## Conclusion

Many of our customers are using AMS to manage AWS infrastructure because it allows them to focus on applications and faster product delivery. In this blog post, we showed how you can perform a fully automated deployment of infrastructure to an AMS-managed account. Through the integration of AWS Service Catalog and ServiceNow you get a unified view of your products and versions built and deployed in your AWS environment. The integration provided through the AWS Service Management Connector for ServiceNow makes it possible for you to trigger deployments directly from ServiceNow.

## About the authors

Prajjaval Gupta is a DevOps consultant and has over 5 years of experience working in the DevOps field. Prajjaval has been assisting Enterprise customers to adopt DevOps culture during their migration to the AWS Cloud.

San Brar is a Data Architect with AWS, where he helps customers architect data analytics solutions, which gives them an accelerated path towards modern and sustainable data platforms to drive business growth. Outside of work, San enjoys hiking and building small robots and rockets with his son.