CI/CD Pipeline for AWS CloudFormation Templates on the AWS Cloud

Using AWS TaskCat

Quick Start Reference Deployment

Shivansh Singh and Tony Vattathil
Solutions Architects, AWS Quick Start team

October 2018

This guide is also available in HTML format at https://docs.aws.amazon.com/quickstart/latest/cicd-taskcat/.

Contents

About This Guide

This Quick Start deployment guide describes how to deploy a continuous integration and continuous delivery (CI/CD) pipeline with AWS TaskCat on the Amazon Web Services (AWS) Cloud, to automatically test and deploy AWS CloudFormation templates from a GitHub repository. AWS CloudFormation templates automate the deployment of the CI/CD environment.

The guide is for IT infrastructure architects, administrators, and DevOps professionals who are planning to implement or extend their AWS CloudFormation-based workloads on the AWS Cloud.

Quick Links

The links in this section are for your convenience. Before you launch the Quick Start, please review the architecture, configuration, and other considerations discussed in this guide.

- If you have an AWS account and you're already familiar with AWS services and TaskCat, you can launch the Quick Start to build an architecture for the workflow shown in Figure 1. The deployment takes approximately 15 minutes. If you're new to AWS or to TaskCat, please review the details and follow the step-by-step instructions provided later in this guide.

Launch Quick Start

- If you want to take a look under the covers, you can view the AWS CloudFormation template that automates the deployment.

View template

About Quick Starts

Quick Starts are automated reference deployments for key workloads on the AWS Cloud. Each Quick Start launches, configures, and runs the AWS compute, network, storage, and other services required to deploy a specific workload on AWS, using AWS best practices for security and availability.

Overview

TaskCat is an open-source tool that tests AWS CloudFormation templates. TaskCat tests your templates by creating a stack in multiple AWS Regions simultaneously, and generates a report with a pass/fail grade for each region. You can specify the regions, indicate the number of Availability Zones you want to include in the test, and pass in the AWS CloudFormation parameter values you want to test. TaskCat is implemented in Python and is available in two formats: as a **pip** module and in a Docker container.

> **Note** TaskCat was developed by the AWS Quick Start team as an internal tool, to automatically test the AWS CloudFormation templates for Quick Starts. It is now available as an open-source tool in GitHub, and you can use this Quick Start to easily deploy TaskCat into your AWS account.

The AWS Cloud makes it easy for developers to code, build, and deploy software. AWS has a continuous integration (CI) orchestration service called AWS CodePipeline and multiple deployment options to help developers quickly deploy and manage their software. Using

AWS CodePipeline with TaskCat makes it easy to continuously build and test your AWS CloudFormation templates whenever you change them.

You can use this Quick Start to test any AWS CloudFormation templates, including nested templates. Your templates must be available in a GitHub repository, which you set up by following the instructions in the Prerequisites section.

This guide provides infrastructure and configuration information for planning and deploying a continuous CI/CD pipeline for AWS CloudFormation templates on the AWS Cloud. It doesn't cover the TaskCat tool in detail. For general guidance and best practices for TaskCat, see the TaskCat documentation.

Cost and Licenses

You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using the Quick Start.

The AWS CloudFormation template for this Quick Start includes configuration parameters that you can customize. For cost estimates, see the pricing pages for each AWS service you will be using. Prices are subject to change.

> **Tip** After you deploy the Quick Start, we recommend that you enable the AWS Cost and Usage Report to track costs associated with the Quick Start. This report delivers billing metrics to an S3 bucket in your account. It provides cost estimates based on usage throughout each month, and finalizes the data at the end of the month. For more information about the report, see the AWS documentation.

This Quick Start deploys AWS TaskCat, which is open source and free to use under the Apache 2.0 license.

AWS Services

The core AWS components used by this Quick Start includes the following services. (If you are new to AWS, see Getting Started with AWS.)

For information about how the Quick Start uses these services, see the Pipeline Workflow section later in this guide.

- Amazon S3 – Amazon Simple Storage Service (Amazon S3) provides developers and IT teams with secure, durable, highly-scalable cloud storage. Amazon S3 offers easy-to-use object storage, and includes a simple web interface so you can store and retrieve any amount of data from anywhere on the web. With Amazon S3, you pay only for the storage you actually use. There is no minimum fee and no setup cost.

- AWS CodePipeline – AWS CodePipeline is a continuous delivery service for fast and reliable application updates. CodePipeline builds, tests, and deploys your code every time there is a code change, based on the release process models you define. You can rapidly and reliably deliver features and updates. You can easily build an end-to-end solution by using our pre-built plugins for popular third-party services like GitHub, or by integrating your own custom plugins into any stage of your release process. With CodePipeline, you pay only for what you use. There are no upfront fees or long-term commitments.
- AWS CodeBuild – AWS CodeBuild is a fully managed build service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers. CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue. You can get started quickly by using prepackaged build environments, or you can create custom build environments that use your own build tools. With CodeBuild, you are charged by the minute for the compute resources you use.
- AWS CodeDeploy – AWS CodeDeploy automates code deployments to any instance, including Amazon Elastic Compute Cloud (Amazon EC2) instances and instances running on premises. CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications. You can use CodeDeploy to automate your software deployments, and to eliminate the need for error-prone manual operations. The service scales with your infrastructure so you can easily deploy to one instance or to thousands of instances.
- AWS Lambda – AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume—there is no charge when your code isn't running. With Lambda, you can run code for virtually any type of application or backend service—all with zero administration. Upload your code and Lambda takes care of requirements for running and scaling your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.
- AWS CloudFormation – AWS CloudFormation gives you an easy way to create and manage a collection of related AWS resources, and provision and update them in an orderly and predictable way. You use a template to describe all the AWS resources (e.g.,

EC2 instances) that you want. You don't have to individually create and configure the resources or figure out dependencies—AWS CloudFormation handles all of that.

Architecture

Deploying this Quick Start with **default parameters** builds the following CI/CD pipeline environment in the AWS Cloud.

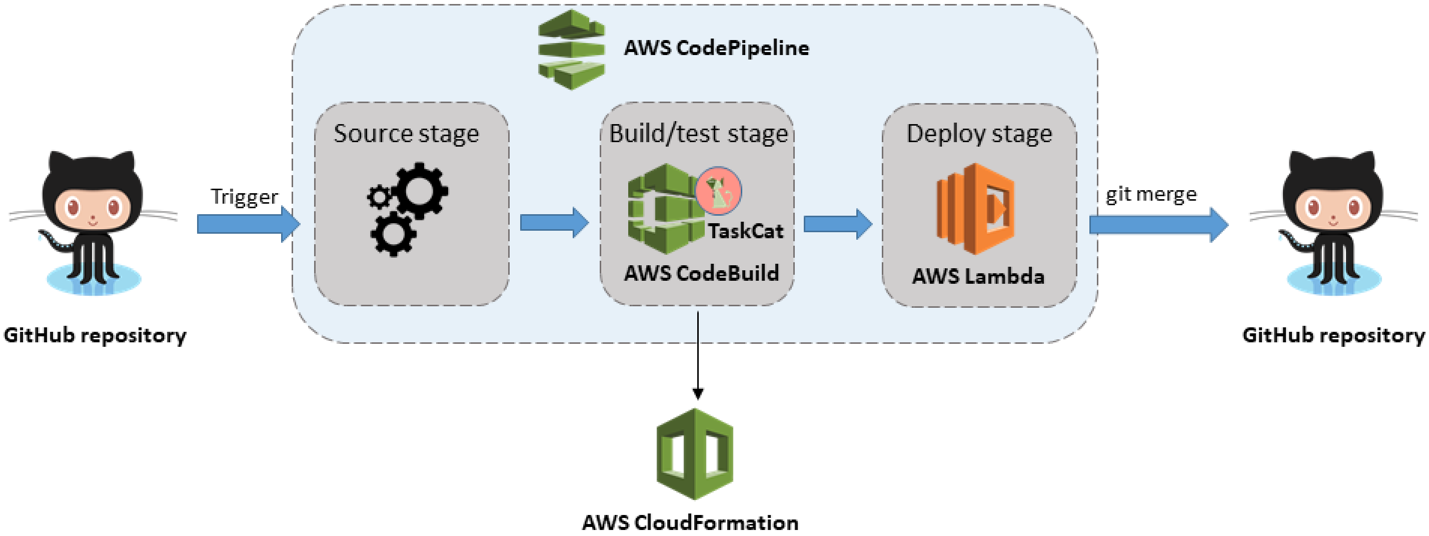


Figure 1: CI/CD pipeline for CloudFormation templates on AWS

The Quick Start sets up the following:

- A pipeline created by CodePipeline, which is triggered when a commit is made to the referenced branch of the Github repository used in the source stage.
- A build project in CodeBuild to run TaskCat and launch AWS CloudFormation templates for testing.
- A Lambda function that merges the source branch of the Github repository with the release branch.
- AWS Identity and Access Management (IAM) roles for the Lambda function and the build project.
- An S3 bucket to stash the build artifacts temporarily and to store the TaskCat report.

Pipeline Workflow

As implemented by this Quick Start, the deployment of the CI/CD pipeline for AWS CloudFormation templates requires two branches in the GitHub repository: a source branch and a release branch. (See the Prerequisites section later in this guide for information about setting up your repository.) The **source branch** is actively used for development and needs to be tested for any code changes. The **release branch** contains the stable code that has been tested successfully and is ready to deploy.

The CI/CD pipeline consists of three stages:

- **Source stage**. When a commit is made into the source branch, it triggers the CI/CD pipeline. In the source stage of the pipeline, the entire contents of the GitHub repository are pulled, zipped, and stored in an S3 bucket. The successful completion of the source stage triggers the build/test stage.
- **Build/test stage**. CodeBuild creates a Linux container, installs TaskCat and its dependencies in the container, downloads the zipped file that contains the source code from the S3 bucket, unpacks it, and runs tests using TaskCat. When the tests are completed, the report generated by TaskCat is uploaded to the S3 bucket. If the tests are successful, the deploy stage is triggered. Otherwise, the build is marked as failed.
- **Deploy stage**. CodePipeline runs a Lambda function that merges the source branch of the GitHub repository into the release branch. The code is now ready to deploy from the GitHub repository. You can also use AWS CodeDeploy and add your own scripts to deploy your software.

Deployment Steps

Prerequisites

This Quick Start requires a GitHub repository that contains the AWS CloudFormation templates you want to test as part of the CI/CD pipeline. Your GitHub repository must have a specific folder structure:

- A **templates** folder, which includes your AWS CloudFormation templates. Templates can be in either JSON or YAML format.
- A **ci** folder, which includes a TaskCat configuration file named **taskcat.yml** and an input **parameters file**. The configuration file should specifiy the template name that needs to be tested, the parameters file, and the tests that TaskCat should run.

For detailed information about these input files, see the TaskCat documentation.

If you want to give TaskCat a trial run, you can download and use any of the AWS CloudFormation templates and configuration files in the Quick Start GitHub organization at https://github.com/aws-quickstart.

Step 1. Prepare Your AWS Account

1. If you don't already have an AWS account, create one at https://aws.amazon.com by following the on-screen instructions.
2. Use the region selector in the navigation bar to choose the AWS Region where you want to deploy CI/CD pipeline for AWS CloudFormation templates on AWS.

Step 2. Set Up Your GitHub Token and Collect Your Information

1. Log in to your GitHub account.
2. Follow the steps in the GitHub documentation to create a new (OAuth 2) token with the following scopes (permissions): `admin:repo_hook` and `repo`. If you already have a token with these permissions, you can use that. You can find a list of all your personal access tokens in https://github.com/settings/tokens.
3. Make a note of the following information:
   - GitHub token name.
   - GitHub repository name – This repository should have the folder structure and files described earlier in the Prerequisites section.
   - Source branch name – This is the branch that CodePipeline should monitor for any changes.
   - Release branch name – This is the branch that the source branch will be merged into after a successful test.

   You will be prompted for this information when you launch the Quick Start.

Step 3. Launch the Quick Start

> **Note** You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using this Quick Start. For full details, see the pricing pages for each AWS service you will be using in this Quick Start. Prices are subject to change.

1. Launch the AWS CloudFormation template into your AWS account.

   Deploy the CI/CD pipeline for AWS CloudFormation templates on AWS

   Launch

   The deployment takes about 15 minutes to complete.

2. Check the region that's displayed in the upper-right corner of the navigation bar, and change it if necessary. This is where the CI/CD pipeline for AWS CloudFormation templates will be built. The template is launched in the US West (Oregon) Region by default.

3. On the **Select Template** page, keep the default setting for the template URL, and then choose **Next**.

4. On the **Specify Details** page, change the stack name if needed. Review the parameters for the template. Provide values for the parameters that require input. For all other parameters, review the default settings and customize them as necessary. When you finish reviewing and customizing the parameters, choose **Next**.

   In the following tables, parameters are listed and described by category.

   View template

   GitHub Configuration:

Parameter label (name)	Default	Description
Repository owner (GitHubUser)	*Requires input*	The GitHub user name of the repository owner.
Repository name (GitHubRepoName)	*Requires input*	The name of the GitHub repository that CodePipeline should monitor for AWS CloudFormation template changes. This GitHub repository must be set up as described in the Prerequisites section.
Source branch (SourceRepoBranch)	*Requires input*	The name of the source branch to be monitored in the GitHub repository. This branch has the AWS CloudFormation templates that are actively being developed. CodePipeline monitors this branch and initiates testing whenever it detects any changes to the templates.

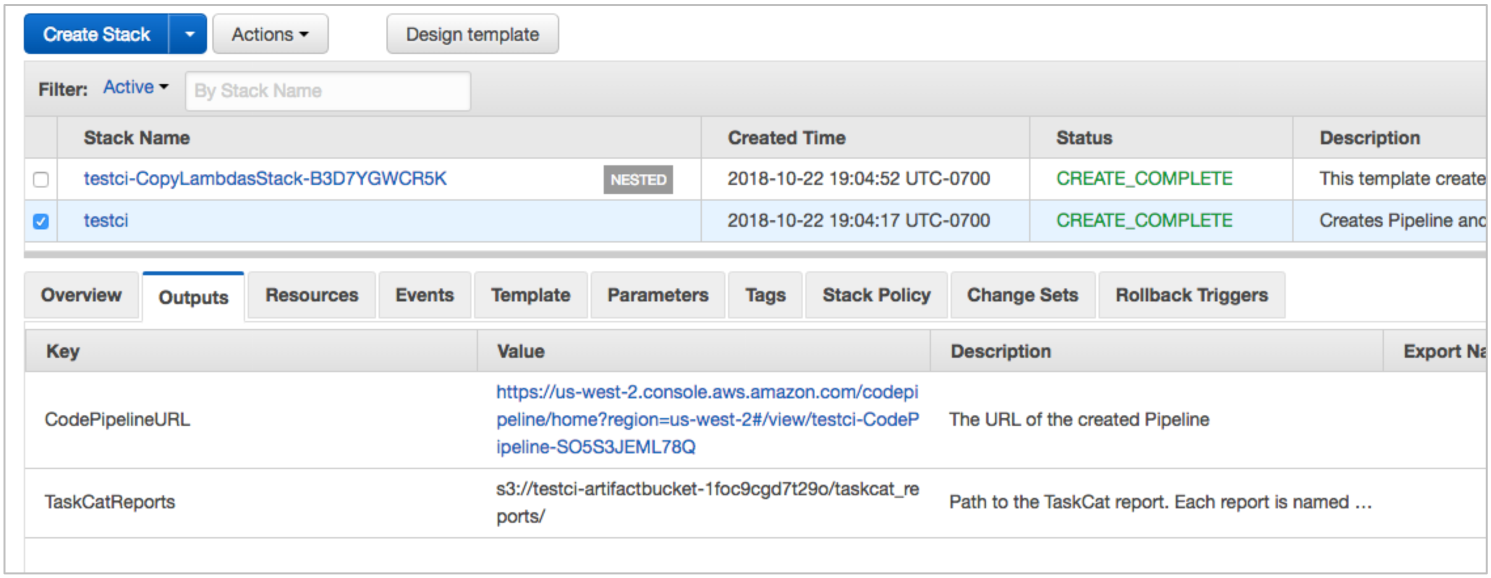
Parameter label (name)	Default	Description
Release branch (ReleaseBranch)	*Requires input*	The name of the release branch in the GitHub repository. This is the branch that contains the stable code that has already been tested. After a successful build, the source branch is merged into the release branch.
OAuth2 token (GitHubOAuthToken)	*Requires input*	The name of the GitHub token with `repo` and `admin:repo_hook` permissions, from step 2.

AWS Quick Start Configuration:

Parameter label (name)	Default	Description
Quick Start S3 Bucket Name (QSS3BucketName)	quickstart-reference	The S3 bucket you have created for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. The bucket name can include numbers, lowercase letters, uppercase letters, and hyphens, but should not start or end with a hyphen.
Quick Start S3 Key Prefix (QSS3KeyPrefix)	quickstart-taskcat-ci/	The S3 key name prefix used to simulate a folder for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. This prefix can include numbers, lowercase letters, uppercase letters, hyphens, and forward slashes.

5. On the **Options** page, you can specify tags (key-value pairs) for resources in your stack and set advanced options. When you're done, choose **Next**.
6. On the **Review** page, review and confirm the template settings. Under **Capabilities**, select the check box to acknowledge that the template will create IAM resources.
7. Choose **Create** to deploy the stack.
8. Monitor the status of the stack. When the status is **CREATE_COMPLETE**, the CI/CD pipeline for AWS CloudFormation templates is ready.

   The **Outputs** tab for the stack will provide information about the resources that were created, as shown in Figure 2.



	Stack Name	Created Time	Status	Description
☐	testci-CopyLambdasStack-B3D7YGWCR5K NESTED	2018-10-22 19:04:52 UTC-0700	CREATE_COMPLETE	This template create
☑	testci	2018-10-22 19:04:17 UTC-0700	CREATE_COMPLETE	Creates Pipeline anc

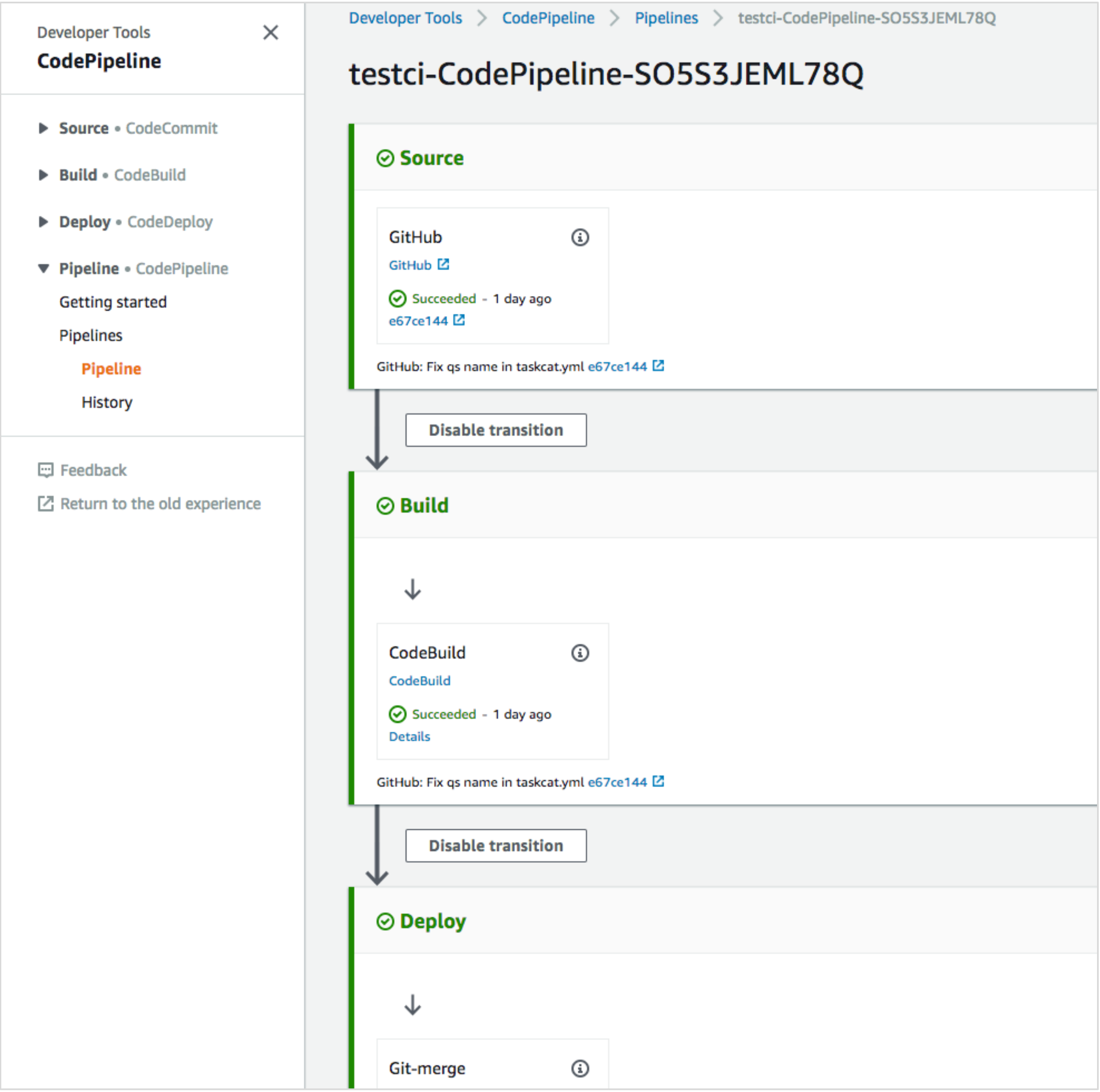
Key	Value	Description	Export Na
CodePipelineURL	https://us-west-2.console.aws.amazon.com/codepipeline/home?region=us-west-2#/view/testci-CodePipeline-SO5S3JEML78Q	The URL of the created Pipeline	
TaskCatReports	s3://testci-artifactbucket-1foc9cgd7t29o/taskcat_reports/	Path to the TaskCat report. Each report is named …	

Figure 2: Quick Start stack outputs

Step 4. Verify the Deployment

In the AWS CloudFormation console, in the **Outputs** tab of your stack, choose the URL for **CodePipelineURL**. This will redirect you to the AWS CodePipeline console.

You should see the CI/CD pipeline for your AWS CloudFormation templates being triggered and the *Source* stage in progress, as shown in Figure 3.



testci-CodePipeline-SO5S3JEML78Q

Source

GitHub
GitHub
Succeeded - 1 day ago
e67ce144

GitHub: Fix qs name in taskcat.yml e67ce144

Build

CodeBuild
CodeBuild
Succeeded - 1 day ago
Details

GitHub: Fix qs name in taskcat.yml e67ce144

Deploy

Git-merge

Figure 3: CI/CD pipeline for the templates in your GitHub repository

FAQ

Q. I encountered a CREATE_FAILED error when I launched the Quick Start.

A. If AWS CloudFormation fails to create the stack, we recommend that you relaunch the template with **Rollback on failure** set to **No**. (This setting is under **Advanced** in the AWS CloudFormation console, **Options** page.) With this setting, the stack's state will be retained, so you can troubleshoot the issue.

> **Important** When you set **Rollback on failure** to **No**, you will continue to incur AWS charges for this stack. Please make sure to delete the stack when you finish troubleshooting.

For additional information, see Troubleshooting AWS CloudFormation on the AWS website.

Q. I encountered a size limitation error when I deployed the AWS Cloudformation template.

A. We recommend that you launch the Quick Start template from the links in this guide or from another S3 bucket. If you deploy the template from a local copy on your computer or from a non-S3 location, you might encounter template size limitations when you create the stack. For more information about AWS CloudFormation limits, see the AWS documentation.

Q. Where can I find the TaskCat report?

A. The report for each TaskCat run is saved in the S3 bucket. The path to the S3 bucket is listed as the value for the **TaskCatReports** key in the **Outputs** section of your CloudFormation stack, as illustrated in Figure 2.

GitHub Repository

You can visit our GitHub repository to download the templates and scripts for this Quick Start, to post your comments, and to share your customizations with others.

Additional Resources

AWS services

- AWS CloudFormation
  https://aws.amazon.com/documentation/cloudformation/
- AWS CodePipeline
  https://aws.amazon.com/documentation/codepipeline/
- AWS CodeBuild
  https://aws.amazon.com/documentation/codebuild/
- AWS Lambda
  https://aws.amazon.com/documentation/lambda
- Amazon S3
  https://aws.amazon.com/documentation/s3/

TaskCat documentation

- TaskCat user documentation
  https://aws-quickstart.github.io/auto-testing.html
- TaskCat API reference
  https://s3-us-west-2.amazonaws.com/taskcat-docs/stacker.m.html

TaskCat source code

- GitHub repository
  https://github.com/aws-quickstart/taskcat
- Pip module
  https://pypi.org/project/taskcat/
- Docker hub
  https://hub.docker.com/r/taskcat/taskcat

Build status

- Travis CI (open-source tool used to validate TaskCat builds and releases)
  https://travis-ci.org/aws-quickstart/taskcat

Quick Start reference deployments

- AWS Quick Start home page
  https://aws.amazon.com/quickstart/

Document Revisions

Date	Change	In sections
October 2018	Initial publication	—

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.