aws re/start

Troubleshooting AWS CloudFormation

This module discusses how to troubleshoot common types of AWS CloudFormation errors.

What you will learn

At the core of the lesson

You will learn how to:

- Identify common AWS CloudFormation errors.
- Remedy common AWS CloudFormation errors.

aws re/start

In this module, you will learn how to:

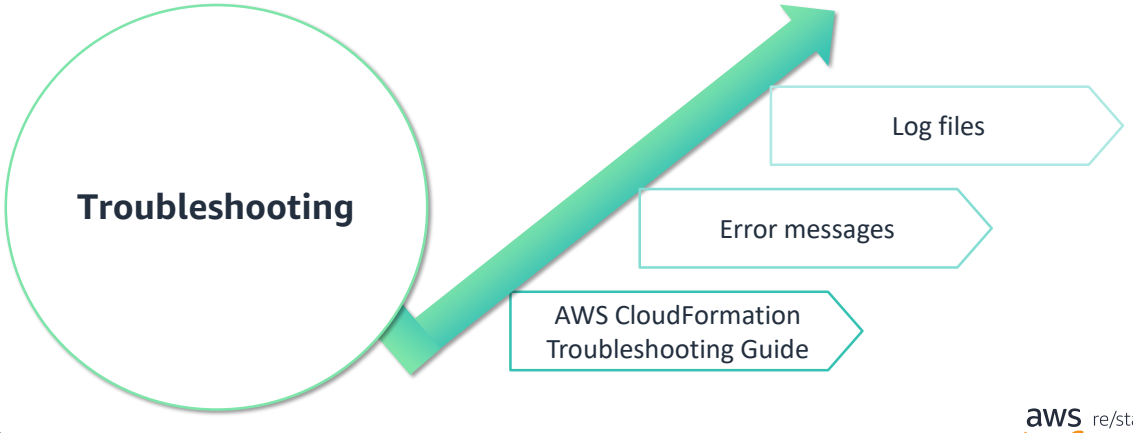
- Identify common AWS CloudFormation errors.
- Remedy common AWS CloudFormation errors.

Troubleshooting AWS CloudFormation

Troubleshooting AWS CloudFormation.

Troubleshooting AWS CloudFormation

What do you do if AWS CloudFormation encounters issues when it creates, updates, or deletes stacks?



aws re/start

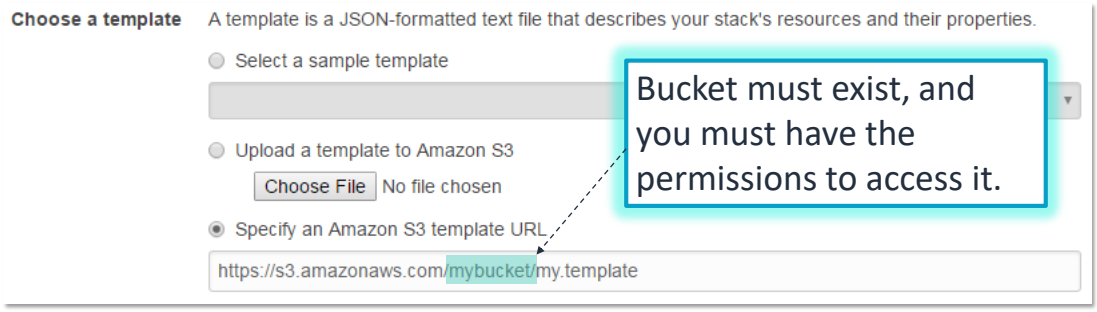
If AWS CloudFormation fails to create, update, or delete your stack, you can view error messages or logs to help you learn more about the issue.

The AWS CloudFormation Troubleshooting Guide is a good place to start when you troubleshoot errors. It provides guidance that ranges from general to specific.

One of the first actions that the Troubleshooting Guide recommends is to look for error messages in the AWS CloudFormation console. If the errors occurred on an Amazon Elastic Compute Cloud (Amazon EC2) instance, you can log in to that instance and examine the cloud-init and cfn log files.

Troubleshooting templates

TemplateURL must reference a valid Amazon S3 bucket



aws re/start

It is a common practice to store AWS CloudFormation templates on Amazon Simple Storage Service (Amazon S3). You then specify the Amazon S3 location when you choose a template so that you can run a stack. If you receive an error message that the TemplateURL must reference a valid S3 bucket, confirm that the bucket exists and that you have the permissions to access the bucket.

Troubleshooting resources

Resources fail to create

- Does the user have the permissions to create the corresponding AWS resource?
- Does the resource type have all the required parameters?
  - Compare your code against working templates.
  - Check the documented syntax for the resource in the AWS CloudFormation User Guide.

aws re/start

If you attempt to create an AWS CloudFormation stack from a template and you see an error message that one of the resources failed to create, you can investigate the following areas.

First, does the user have the permissions to create that type of resource? Check the AWS Identity and Access Management (IAM) policy permissions. Check the user's level of access for the AWS service that the resource is a part of.

Second, verify that the specified resource type has all the required parameters. The easiest way to do this is to review the JavaScript Object Notation (JSON) or YAML Ain't Markup Language (YAML) code section in the template where you specify the resource. Then, compare that code against other code that creates the same type of resource, and is also known to work.

You can also refer to the AWS CloudFormation User Guide section that describes the syntax for how to create that resource type. For example, the AWS::EC2::Instance page describes the syntax for creating an EC2 instance resource.

Troubleshooting a WaitCondition

WaitCondition failure

If WaitCondition times out or returns an error, your `CloudFormation::Init` code has an error.

- Analyze the **cfn-init.log** and **cfn-wire.log** files for details.
- Collect logs through **Amazon CloudWatch Logs**.
- Set `--on-failure DO_NOTHING` to keep the instance from rolling back so that you can log in and examine the logs.
- Most common error:
  - URLs for referenced resources—such as scripts, MSI files (Microsoft installer files), and others—return HTTP 403 or 404 errors.

Recall that WaitConditions are often used when an EC2 instance resource is created. If a WaitCondition times out or returns an error when you attempt to create your AWS CloudFormation stack, it means your CloudFormation::Init code has an error.

If the EC2 instance is still accessible after you notice the error, analyze the cfn-init.log and the cfn-wire.log files for details. You can configure Amazon CloudWatch Logs to collect the logs from the instance.

You could also try running the `create stack`-command again. However, this time, specify `--on-failure DO_NOTHING` to keep instance from rolling back so that you can log in to the instance and examine the logs.

The most common error occurs when files that the CloudFormation::Init code references for resources—such as scripts, MSI files (Microsoft Installer files), and others—return HTTP 403 or 404 errors. These errors means that the referenced resources cannot be accessed from the instance.

AWS CloudFormation errors

Troubleshooting resources

- Template and resource-creation errors are returned in the **AWS CloudFormation console** and on **stdout**.
- User data and `CloudFormation::Init` errors are written to the error logs **cloud-init.log**, **cfn-init.log**, and **cfn-wire.log** located at:
  - Linux: /var/log/
  - Microsoft Windows: C:\cfn\
- Options for retrieving
  - Log in to instance
  - Use CloudWatch Logs to automatically upload logs

aws re/start

If AWS CloudFormation fails to create, update, or delete your stack, you can view error messages or logs to help you learn more about the issue.

This slide describes *general methods* for troubleshooting an AWS CloudFormation issue.

Template and resource-creation errors are returned in the AWS CloudFormation console and on stdout. For example, in the AWS CloudFormation console, you can view the status of your stack. You can view a list of *stack event*s while your stack is being created, updated, or deleted.

If an error occurs when the stack runs, you can find the stack event that failed in this view. You can then expand the details to learn information about what caused the error for that event. The status reason might contain an error message from AWS CloudFormation or from a particular service. These error messages can help you troubleshoot your problem.

For Amazon EC2 issues, connect to the instance—if you can—and review the cloud-init and cfn logs. These logs are published on the EC2 instance. In Linux instances, they are in the /var/log/ directory. In Microsoft Windows instances, they are in the C:\cfn directory. These logs capture processes and command outputs while AWS CloudFormation sets up your instance. For Windows, you can also access the EC2Configure service and cfn logs in the %ProgramFiles%\Amazon\EC2ConfigService

and C:\cfn\log directories. You can also configure your AWS CloudFormation template so that the logs are published to Amazon CloudWatch. CloudWatch displays logs in the AWS Management Console, so you do not need to connect to your EC2 instance.

For more information, refer to Troubleshooting AWS CloudFormation.

Key takeaways

- If AWS CloudFormation fails to create, update, or delete your stack, you can view error messages or logs to help you learn more about the issue.
- It is common practice to store AWS CloudFormation templates on Amazon S3. You then specify the Amazon S3 location when you choose a template so that you can run a stack.
- If a WaitCondition times out or returns an error when you attempt to create your AWS CloudFormation stack, your `CloudFormation::Init` code has an error.

aws re/start

Some key takeaways from this module:

- If AWS CloudFormation fails to create, update, or delete your stack, you can view error messages or logs to help you learn more about the issue.
- It is a common practice to store AWS CloudFormation templates on Amazon S3. You then specify the Amazon S3 location when you choose a template so that you can run a stack.
- If a WaitCondition times out or returns an error when you attempt to create your AWS CloudFormation stack, your `CloudFormation::Init` code has an error.