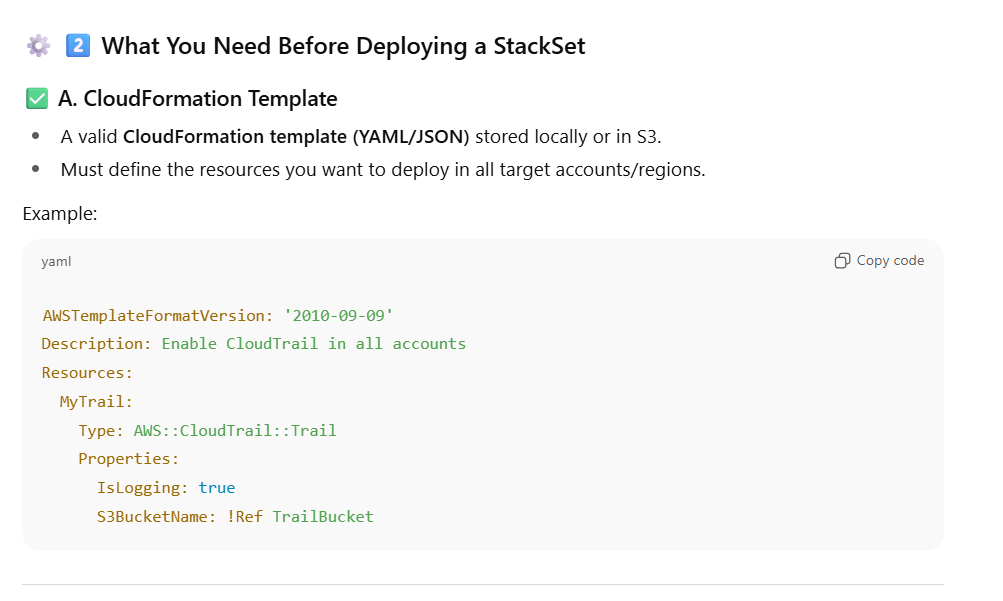
**CLOUDFORMATION**

1. What is the difference between “Stack” and “StackSet” in CloudFormation?

 **Stack:** A single CloudFormation deployment within one region and one account. It manages resources defined in a single template.

 **StackSet:** Allows deployment of the same CloudFormation stack across **multiple AWS accounts and multiple regions** from a single operation.

1. What is needed to deploy cloudformation stackset?

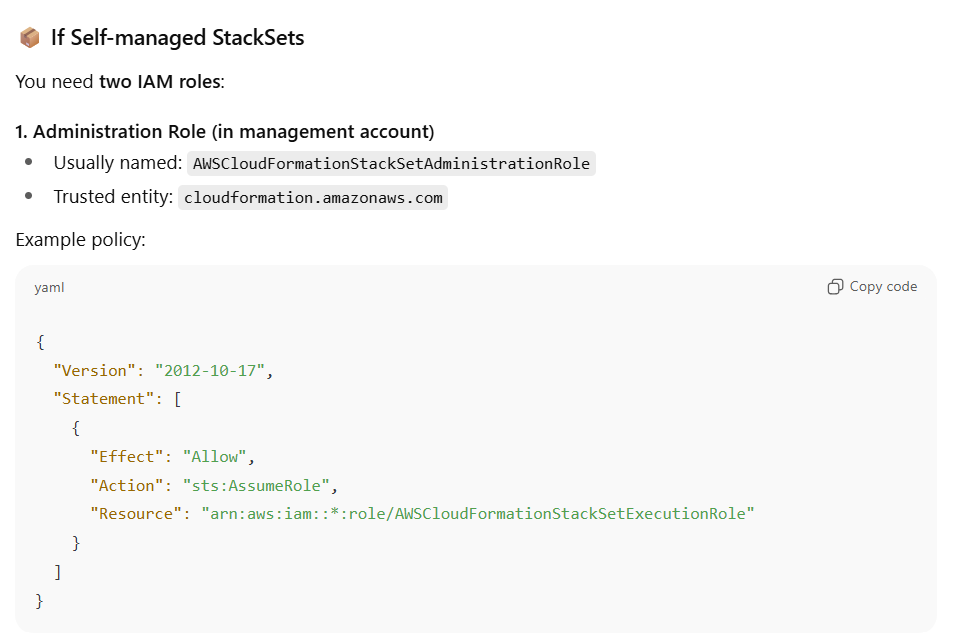


A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



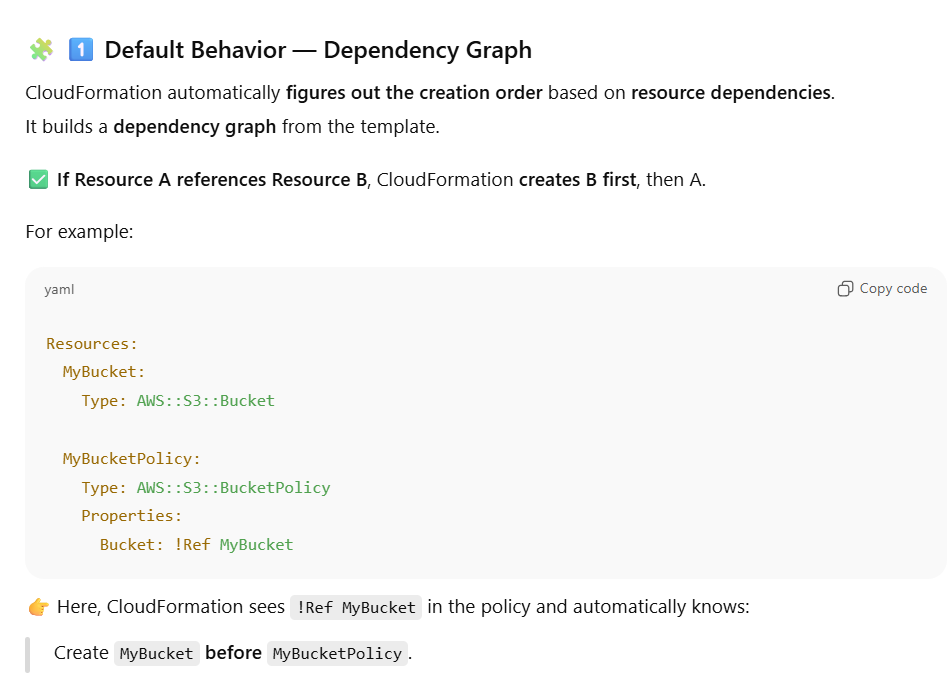
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

1. How do you handle resource creation order in CloudFormation?

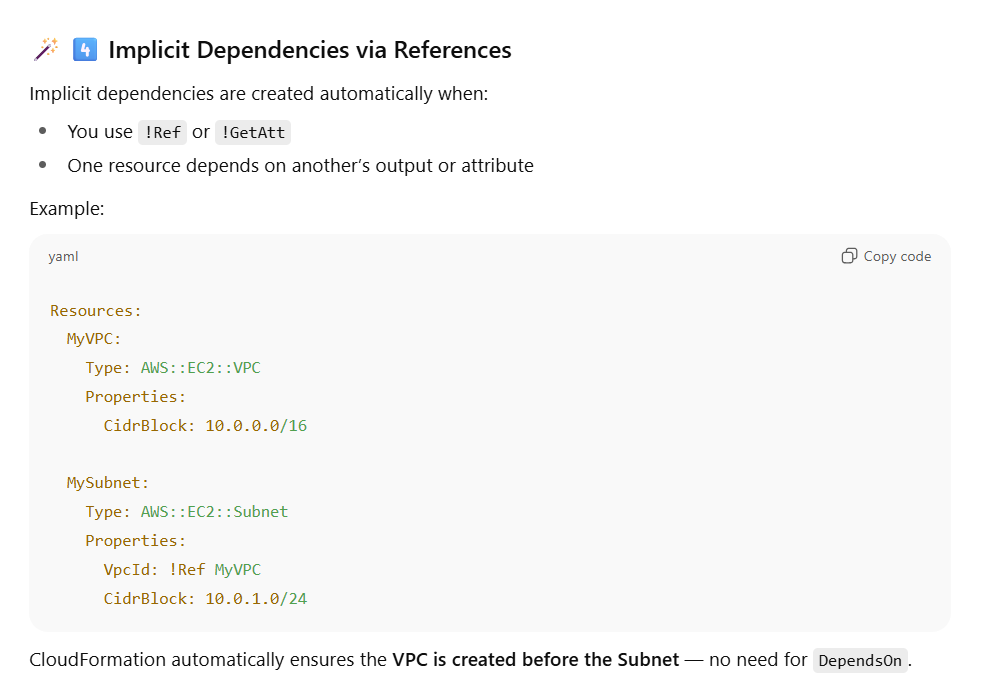


**⚙️ 2️⃣ How CloudFormation Determines Order**

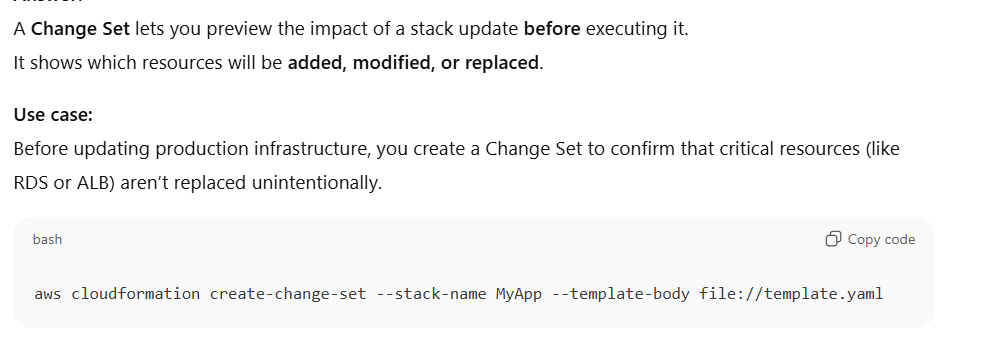
CloudFormation analyzes references like:

* !Ref (references another resource)
* !GetAtt (gets attributes of another resource)
* !Sub with ${LogicalResourceName}
* DependsOn (explicit)
* Nested stacks or outputs from other stacks

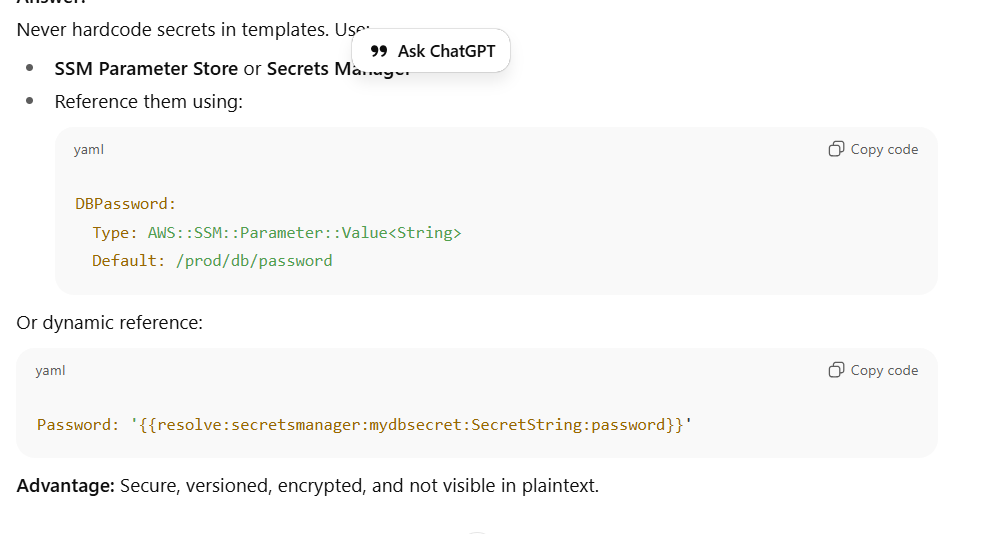
When it detects these, it builds an internal dependency tree and executes in the right order automatically.



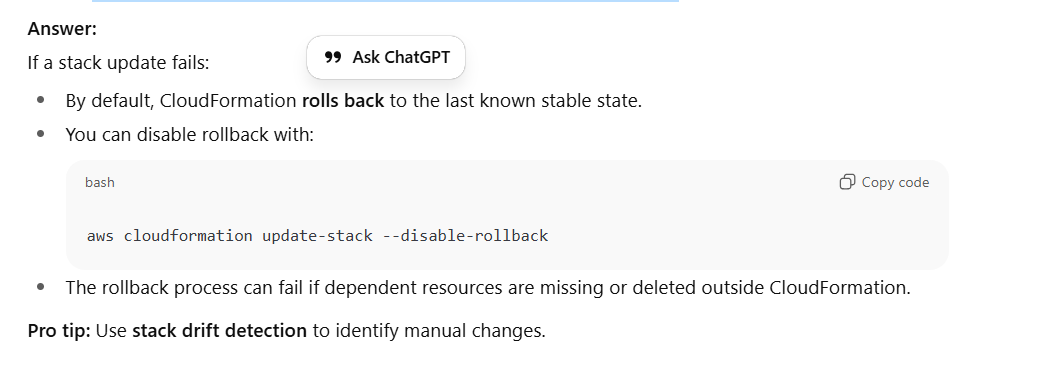
1. What is a “Change Set” and when would you use it?



1. How do you securely reference sensitive data (like passwords) in CloudFormation?



1. What happens if a CloudFormation stack update fails?



1. How do you use Conditions in CloudFormation?



1. How do you reuse values between stacks?



1. Explain the difference between Ref, GetAtt, and Sub.

**🧩 1️⃣ Ref — Reference a Resource or Parameter**

**📘 Purpose:**

Ref returns the **value** of a resource or parameter.

* For **parameters**, it returns the **parameter value** provided at stack creation.
* For **resources**, it returns the resource’s **physical ID** (like ARN, name, or ID) — depending on the resource type.

A screenshot of a computer

AI-generated content may be incorrect.

**🧠 3️⃣ Sub — String Substitution (Interpolation)**

**📘 Purpose:**

Sub (short for **Substitute**) allows you to **embed variables inside strings**, substituting values dynamically.

It can reference:

* Parameters
* Resource logical names (via their Ref value)
* Attributes (via ${Name.Attribute} syntax)
* Explicit variable mappings



A screenshot of a computer

AI-generated content may be incorrect.

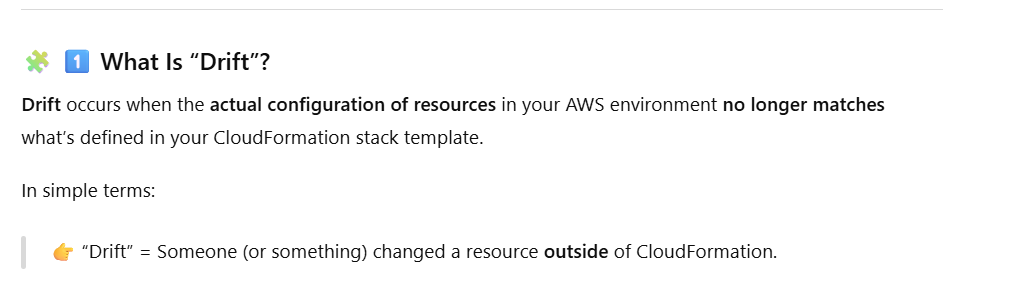
A white background with black text

AI-generated content may be incorrect.

1. How can you handle cross-region references in CloudFormation?

CloudFormation **does not natively support cross-region references**.  
Workarounds:

1. Use **StackSets** to deploy in multiple regions.
2. Store outputs (like ARNs) in **SSM Parameter Store** or **AWS Secrets Manager**.
3. Retrieve them via AWS::SSM::Parameter::Value<String> in another region.
4. What are “Drift” and “Drift Detection”?



A white background with black text

AI-generated content may be incorrect.

1. How do you deploy nested CloudFormation stacks?

Use AWS::CloudFormation::Stack resource type to include nested templates.

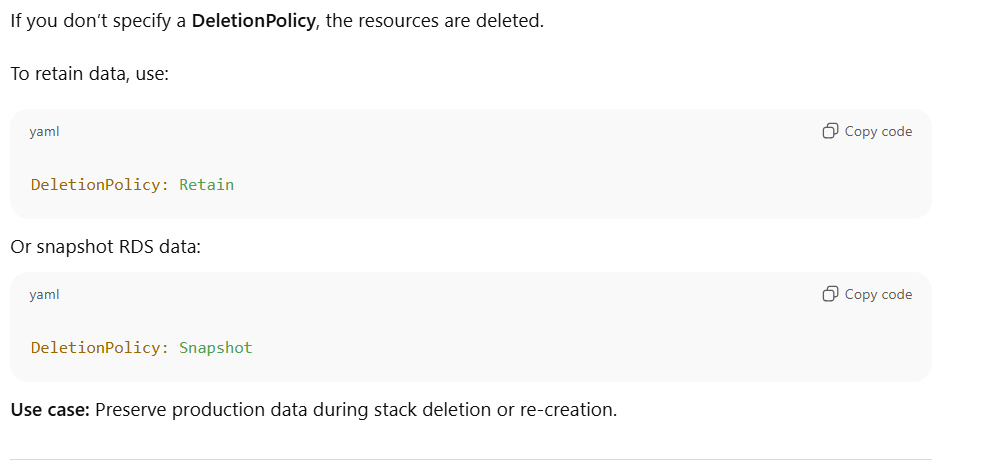


1. How do you dynamically name resources based on environment?

A screenshot of a computer

AI-generated content may be incorrect.

1. What happens when you delete a stack that contains S3 buckets or RDS instances?



1. What is SAM and how it is connected to Cloudformation?

**🧩 What Is AWS SAM?**

**AWS SAM (Serverless Application Model)** is an **extension of AWS CloudFormation** — designed specifically for **building, deploying, and managing serverless applications**.

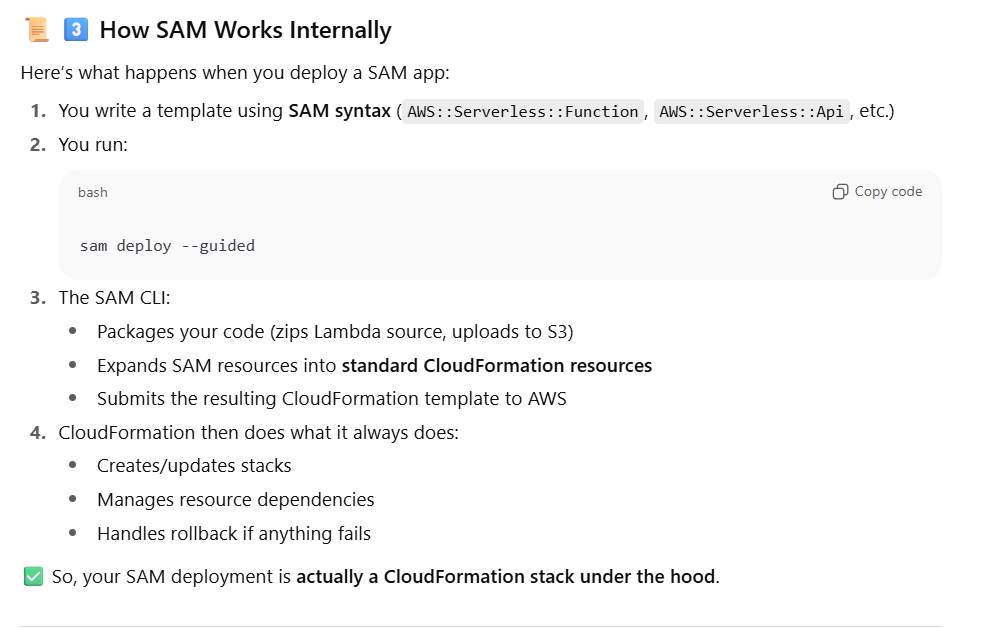
It gives you a **simplified syntax** for defining **Lambda functions**, **API Gateway APIs**, **Step Functions**, **DynamoDB tables**, and other serverless resources — without needing to write full, verbose CloudFormation templates.

**⚙️ Relationship Between SAM and CloudFormation**

AWS SAM **is built on top of CloudFormation**.  
That means every SAM template is essentially a **CloudFormation template**, but with **shortcuts** (called *SAM resources* or *macros*) that expand into standard CloudFormation resources when you deploy.

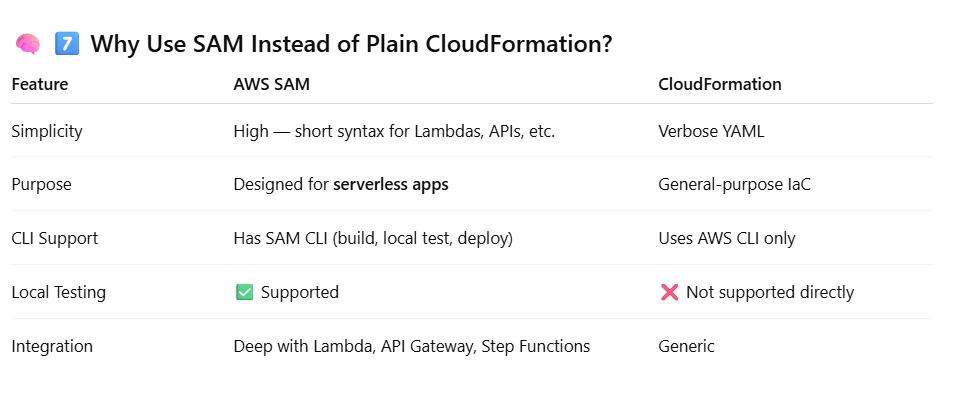
**🧠 Think of it like:**

**SAM = CloudFormation + Serverless shortcuts**



A screenshot of a computer

AI-generated content may be incorrect.



1. What Are Mappings in CloudFormation?

A **Mapping** is a **static lookup table** in your CloudFormation template — it lets you define **key-value pairs** that you can reference later with the **Fn::FindInMap** function.

Help you manage **region-, environment-, or configuration-specific values** directly inside your template

