AWS CloudFormation

If you only need to deploy a small number of services or applications on AWS, you will quickly find that setting up each one manually is tedious and time-consuming.

Not to mention, configuring each AWS resource by hand puts you at a much higher risk of making errors or introducing inconsistencies.

Enter AWS CloudFormation

CloudFormation is an infrastructure automation platform for AWS that makes deploying AWS resources at a much faster, more efficient, and more secure scale.

This article provides an overview of AWS CloudFormation, including how it works, its benefits, and how to create and deploy CloudFormation templates using the console, CloudFormation Designer, and the AWS command line.

What is AWS CloudFormation?

AWS CloudFormation is an AWS service that uses template files to automate the setup of AWS resources.

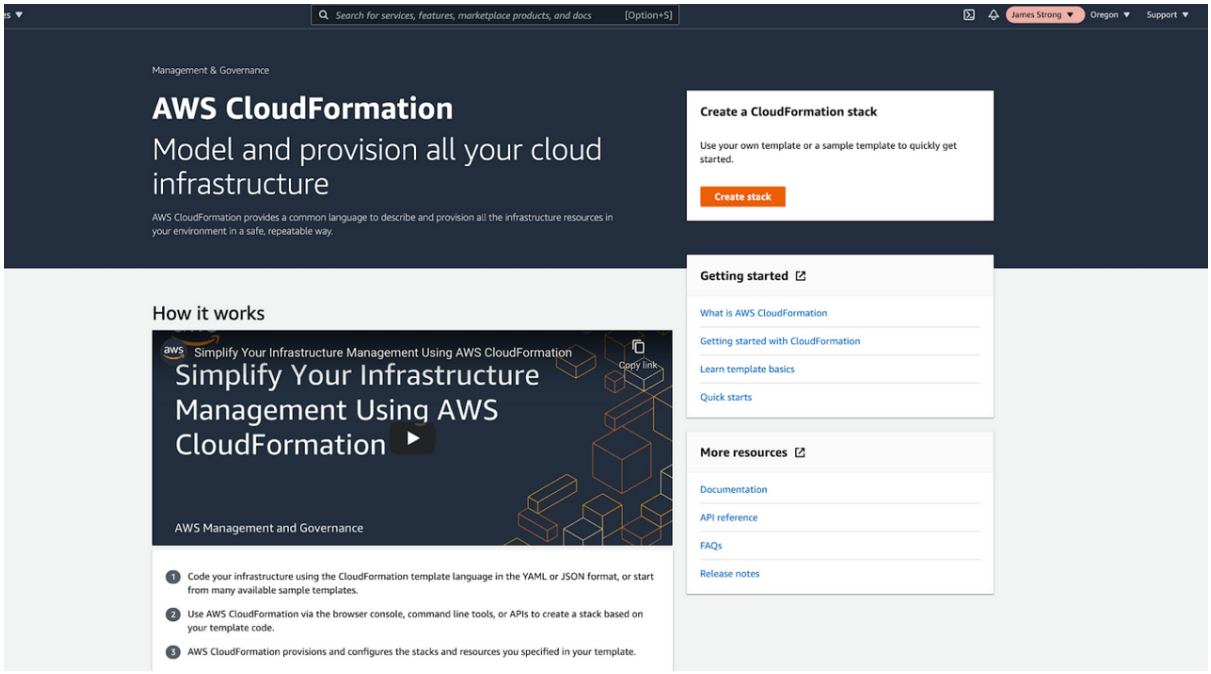
It can also be described as an infrastructure automation or infrastructure-as-code (IAC) tool and cloud automation solution as it automates the setup and deployment of various infrastructure-as-a-service (IaaS) offerings on AWS CloudFormation Support. You can use CloudFormation to automate the configuration of workloads running on the most popular AWS services, such as the EC2 Compute Service, the S3 Storage Service, and the IAM Service to configure access control.

You can also apply CloudFormation templates to AWS services that meet specific use cases, such as ground stations, AWS satellite management solutions.

In general, if a service runs on AWS, it's a safe bet that you can use CloudFormation to automate its configuration and deployment.

It is worth noting that CloudFormation is not the only way to configure and deploy services on AWS. You *can* handle these processes manually using the AWS command-line interface, API, or Web console.

Manual provisioning is the approach that teams typically take when getting started with AWS and learning how to deploy services. However, as they scale their environments up in size, many teams quickly realize they need a solution like CloudFormation to make the deployment process faster and more consistent.



Benefits of CloudFormation

CloudFormation and other AWS-compatible IaC tools offer a range of benefits that make cloud service deployment and management faster and more efficient.

Deployment speed

When you create CloudFormation templates to manage how AWS resources are configured and deployed, you can almost instantly deploy multiple instances of the same resources using just one template. If you manually set up each deployment by running commands on the CLI or by pressing a button in the AWS console, this approach leads to a much faster deployment than you can get.

The caveat, of course, is that you'll need to spend time setting up your CloudFormation templates. However, let's say you repeat the same type of deployment multiple times. In that case, it would overall be much faster to create a CloudFormation template that you can reuse for each deployment, rather than having to manually configure each one.

Scaling up

Even if you don't initially expect to deploy multiple instances of the same AWS resources, CloudFormation templates are useful because they ensure you can quickly scale up your environment when the time comes. With a CloudFormation template on hand, you'll find that you can add more virtual machine instances or storage space, for example, at a moment's notice, if your application has increased traffic and you need to upgrade your environment. is required.

Alternatively, you can take some of your deployments offline when demand dips and you want to do less to save money, while maintaining the ability to quickly redeploy them using CloudFormation when demand increases.

Service integration

A single CloudFormation template can manage individual services or resources, and the deployment of multiple resources. This manageability means you can use CloudFormation to integrate various AWS cloud services. For example, you can write a template that sets up an EC2 virtual machine within an AWS Virtual Private Cloud (VPC) or deploys an S3 storage bucket and configures access control using an IAM service.

Managing multiple services through a single template AWS integrates services as you build a complete cloud environment.

Consistency

When you use CloudFormation templates to define and deploy AWS resources, you can apply the exact same configuration over and over again. In this way, CloudFormation ensures that your applications and services will be consistent and uniform, no matter how many instances you create.

The alternative approach, which is to set up each resource by hand, introduces the risk that the working engineer may apply different settings to different instances, resulting in incompatibility. In turn, managing your environment will be more challenging because some resources will appear different than others, even though they perform the same primary function. You can have different EC2 instances hosting replicas of the same application, for example, or different IAM access controls for the same service. This discrepancy will make it challenging to manage resources equitably.

Security

Along similar lines, although CloudFormation isn't a security tool per se, it can improve the overall security of your AWS environment, reducing the risk of oversight or human errors that could turn into breaches. As long as you design your CloudFormation templates to be secure, you don't have to worry that an engineer deploying resources will forget to turn on critical access controls, for example, or leave data unrestricted. This will expose you to public access.

Easy update

In addition to deploying new resources, you can apply changes to existing resources with CloudFormation templates.

This capability simplifies the process of, for example, adding more storage to a fleet of ec2 instances or changing access control rules.

Auditing and Change Management

When you use Cloud Formation to manage your infrastructure, you can track changes to which templates you have applied and how they change over time. Change tracking in Cloud Formation means you'll be able to determine how your AWS services and resources have changed over time without having to re-create a timeline of updates by looking at logs.

CloudFormation Alternatives

CloudFormation is Amazon's own IaC solution, but it isn't the only cloud automation or IaC tool you can use in the AWS cloud to get the benefits described above. Your team can also use third-party IaC platforms compatible with AWS, such as HashiCorp Terraform or Ansible.

The main advantage of choosing CloudFormation is that most third-party IaC tools can configure resources to run on various public clouds, not just AWS.

If you use multiple clouds at once-in other words, if you run some workloads on AWS and others on Microsoft Azure-a third-party IaC tools platform is convenient because it gives you access to the same configuration management tools for everyone. provides. allows to use.

On the other hand, as AWS's native IaC platform, CloudFormation provides the deepest level of integration with AWS Cloud, including features like Designer, which lets you create and modify CloudFormation templates directly on the AWS website.

CloudFormation also provides a high level of assurance that your templates will always be compatible with AWS services, even when Amazon makes changes to its services. If Amazon were to update one of its services, it's likely that third-party IAC tools would stop working with it, at least until the tools' developers had time to catch up on the changes.

This isn't a frequent problem because Amazon doesn't make significant changes to its cloud services often. It may take time for new services to become available in

CloudFormation, but third-party updates will still take longer. It is not a factor if you consider using CloudFormation or alternative infrastructure automation and cloud automation solutions.

CloudFormation Template Terms and Concepts

It is helpful to understand the basic concepts around which CloudFormation templates structure resources, variables, and functions.

Template

A CloudFormation template is simply a text file formatted in a specific way (see below for details on formatting) that defines how AWS services or resources should be configured and deployed.

Stack

Stack is a term AWS uses to refer to a collection of multiple AWS resources - such as EC2 virtual machines, S3 storage and IAM access control - that you can manage together using a single template.

CloudFormation supports templates that are formatted using JSON or YAML, and these are the most widely used file formats for structuring text files. Most other IaC tools use the same formatting languages as do platforms like Kubernetes.

Parameters

If you need to apply unique settings to each deployment, you can do so using parameters. Parameters let you define custom values for each deployment that CloudFormation will apply at runtime.

Conditions

You can also fine-tune deployments by setting conditions, which let you define conditional rules to precisely control how each deployment proceeds.

Changeset

If you want to update a deployment using CloudFormation, you can update the template that you used to create the deployment. You can then create a change set, which summarizes the changes from the updated template before making the change.

Tasks

There are several ways to get data into a CloudFormation template, with parameters being the primary. But those parameters may not be known at the time of deployment. CloudFormation functions allow CloudFormation designers to retrieve data from resources deployed in the current CloudFormation or from external sources in an AWS account. Refs are widely used to refer to other resources inside a template, as in the example below. It creates an EIP, for instance, created earlier in the template.

```
"MyEIP" : {
   "Type" : "AWS::EC2::EIP",
   "Properties" : {
      "InstanceId" : { "Ref" : "MyEC2Instance" }
   }
}
```

How to Create a CloudFormation Template

There are two ways to create a CloudFormation template:

- Using a pre-existing template as the foundation or
- Writing an entirely new template from scratch

The former approach will be faster and more comfortable in most cases, especially if you are new to CloudFormation and have a simple configuration for deployment. If you have an incredibly complex deployment or need to work with specific AWS resources that are not well represented in AWS's existing template library, it usually only makes sense to create new templates.

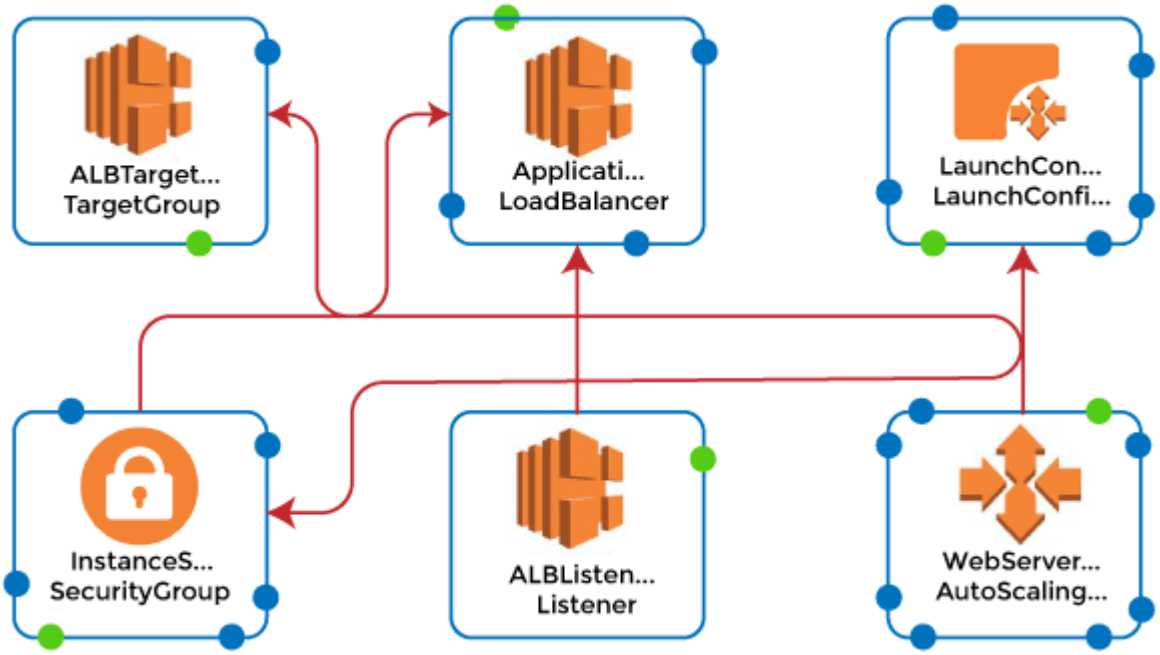
1: Pre-existing templates

To use a pre-existing template as the basis for your deployment, browse the AWS Template Collection to find a template that best suits your needs. For example, if you want to deploy an EC2 instance and configure storage for it, a template called "Amazon EC2 Instance with an Ephemeral Drive" would be an excellent place to start.

Once you've selected a template, you can either download it to your computer, edit it in a local text editor, or open it in AWS CloudFormation Designer, an online tool that works with AWS CloudFormation offers to create and modify templates. The designer provides visualization and drag-and-drop functionality that make it easy to create deployments without a lot of coding (though you should expect to write some code if you need to implement more than just basic functionality).

This template will deploy an elastic load balancing load balancer and an auto scaling group that receives traffic only from the load balancer.

Here is the view from the CloudFormation Template Designer



2: New templates

If you choose to create a new template, you can do so using any text editor. You can quickly create new deployment rules using the visualization features of CloudFormation Designer as it supports new and previously created templates.

How to Deploy CloudFormation Templates

Once your template is ready, you can start the deployment process to create the resources defined in the template in your actual AWS environment.

There are several ways to deploy a CloudFormation template. The approach you take will depend on how you created your template and in which part you prefer the AWS interface:

AWS Console: If your template is a text file stored on your local computer, the easiest way to deploy it is to log into the AWS CloudFormation console https://console.aws.amazon.com/cloudformation and click New Stack Create. The console will guide you through naming your template and uploading the template file to your computer. Once you have completed the steps and reviewed your configuration, click the Create button to deploy your template.

From CloudFormation Designer: If you create your template in CloudFormation Designer, you can create it directly by clicking the Create Stack button, following the on-screen instructions, and pressing Create when you're ready to apply the template. Can deploy directly.

AWS CLI: You can use the AWS CloudFormation deploy command to deploy the template using the AWS CLI tool. Use command-line arguments to determine where your template is stored (typically, you'll upload it to S3 first and point the CLI at that file) and other options you want to configure. The AWS Documentation provides complete details on CLI deployment.

Updating the CloudFormation Stack

There are two basic ways to make changes to the stack you deploy using the CloudFormation template.

The first is to update the relevant template, then immediately deploy the updated template using one of the deployment methods described above. AWS calls this method Direct Update. This is the quickest and most obvious way to update your deployment, and it doesn't provide an opportunity to see the effect of the changes you want before they are implemented.

The second approach, which provides more control and the ability to preview changes before they take effect, creates a change set and uses that to update. You can create a change set in the CloudFormation console by selecting Stack and then Stack Actions. You can then select the template you want to update, and the console will then walk you through the steps to modify your template, review the changes, and apply them. You can also create a changeset on the AWS CLI using the create-change-set command.

Advanced CloudFormation

There are many advanced functionalities that can be used in CloudFormation. These functionalities are auxiliary scripts in CloudFormation, and they extend CloudFormation so that designers can execute scripts outside of CloudFormation.

- **The if-init**: cfn-init helper script reads the metadata from the AWS::CloudFormation::Init key and acts according to the received and parsing metadata from CloudFormation. It can also install packages, write files to disk, and enable or disable services on EC2 instances.
- **cfn-signal:** Designers use cfn-signal with a CreationPolicy or WaitCondition as a signal to synchronize resources in the stack when the prerequisites are ready. This

allows updates or tasks to be performed on resources and allows CloudFormation to continue creating resources in the stack.

- **Can retrieve metadata**: Used to retrieve metadata from the CloudFormation stack.
- **cfn-hup**: Used to check for updates to metadata and execute custom hooks when changes are detected, such as updates to scripts and files.
- Services like AWS SAM and eksctl use CloudFormation on the back end to facilitate serverless and Kubernetes cluster deployments.

Conclusion

Manually setting up and deploying AWS resources is an unproductive use of your team's time. This configuration also increases the risk of oversight and inconsistencies, leading to management problems and security risks. In addition, it makes it difficult to quickly update or scale resources.

By taking advantage of IaC tools like CloudFormation, your team can streamline the AWS deployment process. You can define your resource configurations once and then deploy them as many times as you want. You can also update resources quickly and predictably through changes, and you can use your CloudFormation templates to keep track of how you configure your resources and how they work over time.