
AWS CloudFormation - инфраструктура как код

SOFTPROM
softprom.com • info@softprom.com

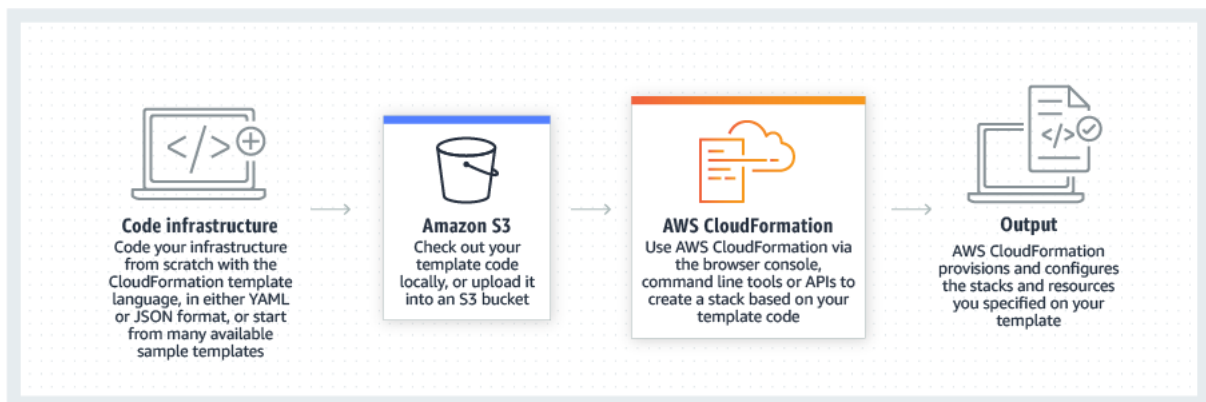


AWS CloudFormation - инфраструктура как код

AWS CloudFormation предлагает простой способ моделирования необходимых ресурсов AWS, их быстрого и единообразного выделения, а также управления ими на протяжении всего жизненного цикла за счет работы с инфраструктурой как с кодом.

Сервис CloudFormation вводит четыре понятия:

- **Шаблон (Template)** – это декларативный файл кода JSON или YAML, который описывает надлежащее состояние ресурсов, необходимых для развертывания приложения.
- **Стек (Stack)** реализует группу ресурсов, перечисленных в шаблоне, и управляет ей, а также позволяет управлять состоянием и зависимостями этих ресурсов совместно.
- **Набор изменений (Change Set)** – это ознакомительная версия изменений, которые будут выполнены операциями со стеком для создания, обновления или удаления ресурсов.
- **Набор стеков (Stack Set)** – это группа стеков с совместным управлением, которые можно реплицировать как группу.



Преимущества шаблонов AWS CloudFormation

Знакомый формат: шаблон CloudFormation – это текстовый файл в формате JSON (JavaScript Object Notation) или YAML, который описывает элементы инфраструктуры AWS, необходимой для запуска приложения или сервиса, а также любые взаимосвязи между ними.

Управление связями: шаблоны подробно описывают связи между ресурсами, например какие инстансы EC2 должны быть связаны с балансировщиком нагрузки

Elastic Load Balancing или тот факт, что том EBS должен находиться в одной зоне доступности EC2 с экземпляром, за которым он закреплен.

Возможность многократного использования: параметры шаблонов позволяют использовать один шаблон для развертывания множества инфраструктур с различными значениями конфигурации, например количеством экземпляров, которые должны быть развернуты для приложения.

Ценная обратная связь: шаблоны также показывают характеристики на выходе, которые позволяют пользователю узнать результаты развертывания или информацию о конфигурации. Например, по завершении работы шаблон может вернуть URL сервера Elastic Load Balancing, необходимый клиенту для подключения к только что запущенному приложению.

Отсутствие конфликтов: все ресурсы AWS в шаблоне определяются с помощью логических имен, благодаря чему можно создать несколько стеков из шаблона, не опасаясь возникновения конфликтов между именами ресурсов AWS.

Удобный запуск: используйте любой из методов запуска стека без предварительной регистрации шаблона в AWS CloudFormation.

Визуализация стека: CloudFormation Designer позволяет визуализировать шаблоны в виде диаграммы. Можно легко просматривать ресурсы AWS и связи между ними, а также удобным образом менять их взаимное расположение на диаграмме. Можно редактировать шаблоны путем перетаскивания в графическом интерфейсе и с помощью встроенного редактора JSON. При изменении диаграммы автоматически изменяется и файл JSON данного шаблона.

Поиск ресурсов: AWS CloudFormation сохраняет копию шаблона стека, что позволяет использовать Консоль управления AWS, инструменты командной строки и API для поиска конкретных конфигураций ресурсов, использованных при создании стека.

Автоматизация: процесс создания шаблона можно автоматизировать с помощью любого языка программирования или удобного инструмента. Для автоматизации создания стека из шаблона также можно использовать API сервиса CloudFormation, AWS SDK или интерфейс командной строки AWS.

Перед началом работы

Прежде всего необходимо ознакомиться с [документацией](#) и [передовыми практиками](#) использования AWS CloudFormation.

Настоятельно рекомендуем пройти по [контрольному списку](#) для DevOps-инженеров. В упомянутом списке содержится масса полезных ссылок на готовые шаблоны, инструменты, упрощающие разработку в разных средах, сниппеты. Также в контрольном списке приведены рекомендации по разработке, которые перекликаются с передовыми практиками представленными в документации AWS.

Пакеты AWS QuickStart

Библиотека готовых шаблонов [Quick Start](#) созданы архитекторами решений и партнерами AWS, чтобы помочь клиентам развертывать на AWS распространенные технологии с учетом рекомендаций AWS по обеспечению безопасности и высокой доступности. Такие пакеты позволяют заменить сотни операций, выполняемых вручную, на несколько простых шагов, чтобы моментально создавать рабочие среды и сразу же начинать их использование.

В каждый пакет Quick Start включены шаблоны AWS CloudFormation, позволяющие автоматизировать развертывание, а также руководство, в котором рассматривается архитектура и приводятся пошаговые инструкции по развертыванию.

Передовые практики использования AWS CloudFormation

Планирование и организация

- Сегментируйте стеки по принципу многоуровневой инфраструктуры или сервис ориентированной архитектуры стеков
- Используйте перекрестные ссылки для экспорта общих ресурсов
- Используйте AWS IAM для управления доступом
- Используйте шаблоны для репликации стеков в средах разработки, тестирования и продакшн.
- Проверьте квоты для всех типов ресурсов в аккаунте AWS, выбранном регионе AWS
- Используйте вложенные стеки для повторного использования готовых шаблонов

Создание шаблонов

- Не вносите учетные данные в шаблоны
- Используйте характерные для AWS типы параметров
- Используйте ограничения для величин параметров
- Используйте AWS :: CloudFormation :: Init для развертывания приложений на инстансах Amazon EC2
- Используйте последние helper скрипты
- Тестируйте шаблоны перед использованием

Управление стеками

- Управляйте всеми ресурсами стека с помощью AWS CloudFormation
- Создавайте наборы изменений перед обновлением ваших стеков
- Используйте стековые политики
- Используйте AWS CloudTrail для регистрации обращений к AWS CloudFormation
- Используйте анализаторы кода и средства контроля версий при разработке шаблонов
- Регулярно обновляйте программное обеспечение на инстансах Amazon EC2

Структура шаблона

В JSON формате

```
{
  "AWSTemplateFormatVersion" : "version date",

  "Description" : "JSON string",

  "Metadata" : {
    template metadata
  },

  "Parameters" : {
    set of parameters
  },

  "Mappings" : {
    set of mappings
  },

  "Conditions" : {
    set of conditions
  },

  "Transform" : {
    set of transforms
  },

  "Resources" : {
    set of resources
  },

  "Outputs" : {
    set of outputs
  }
}
```

В формате YAML

```
---
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
```

template metadata

Parameters:

set of parameters

Mappings:

set of mappings

Conditions:

set of conditions

Transform:

set of transforms

Resources:

set of resources

Outputs:

set of outputs

Более подробно со структурой шаблона AWS CloudFormation можно ознакомиться [в документации](#).