

# Развертывание WCS при помощи CloudFormation

## Описание

AWS CloudFormation позволяет развертывать наборы облачных серверов по заданному шаблону. Таким образом, например, можно развернуть простейшую CDN. Обновление WCS до последней сборки и настройка конкретного экземпляра при этом проводятся при помощи `UserData` скриптов.

## Пример шаблона CloudFormation для развертывания CDN

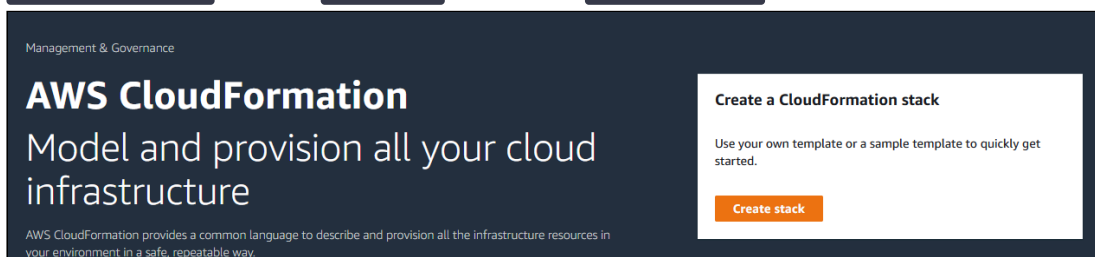
Ниже приводится пример шаблона CloudFormation для развертывания простейшей CDN из двух WCS серверов: Origin и Edge. При этом:

- в качестве базового образа может быть выбран либо [образ WCS из AWS Marketplace](#), либо образ Amazon Linux 2, Ubuntu 18.04 и [других поддерживаемых ОС](#)
- при необходимости, устанавливается Java 14
- при необходимости, устанавливается либо обновляется WCS до последней сборки

 [CloudFormation example template to deploy WCS CDN of one Origin and one Edge](#) >

## Пример развертывания WCS CDN по шаблону в веб консоли CloudFormation

1. Войдите в свой аккаунт AWS, перейдите в желаемый регион и откройте `CloudFormation` в меню `Services`. Нажмите `Create Stack`



2. Выберите `Upload a template file`, нажмите кнопку `Choose file` и загрузите шаблон

Step 1  
**Specify template**

Step 2  
Specify stack details

Step 3  
Configure stack options

Step 4  
Review

### Create stack

**Prerequisite - Prepare template**

Prepare template  
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready  Use a sample template  Create template in Designer

**Specify template**  
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source  
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL  Upload a template file

Upload a template file  
 No file chosen  
JSON or YAML formatted file

S3 URL: Will be generated when template file is uploaded

3. После загрузки шаблона, нажмите **Next**

Upload a template file

wcs-ec2-template-origin-edge.yml  
JSON or YAML formatted file

S3 URL: https://s3.eu-north-1.amazonaws.com/cf-templates-1c0pbvffxqz0-eu-north-1/20211695zL-wcs-ec2-template-origin-edge.yml

4. Введите имя набора серверов

Step 1  
Specify template

Step 2  
**Specify stack details**

Step 3  
Configure stack options

Step 4  
Review

### Specify stack details

**Stack name**

Stack name  
wcs-test-stack  
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

5. Введите AMI ID образа Amazon Linux 2 для выбранного региона, или оставьте WCSAMI (в этом случае будет использован образ WCS из AWS Marketplace с почасовой тарификацией)

ImageId  
Basic instance ami (WebCallServer 5.2.944 AMI by default, mapped by region)

ami-08b4d798afd78c423

6. Введите базовую часть имени сервера (к имени будет добавлено **-origin** и **-edge** соответственно), выберите тип сервера, введите размер Java heap и выберите

## ключ SSH доступа к серверу

**InstanceName**  
Name of EC2 instance

**InstanceType**  
Basic EC2 instance type

**JavaHeapSize**  
Maximum Java heap size in megabytes (2048m) or gigabytes (2g), 1024m by default

**KeyName**  
Name of an existing EC2 KeyPair to enable SSH access to the instance

## 7. Укажите Id подсети

**SubnetId**  
SubnetId of an existing subnet in your Virtual Private Cloud (VPC)

## 8. Введите `true`, чтобы автоматически обновить WCS до последней сборки

**UpdateWCS**  
Update WCS to the latest build

## 9. Укажите VPC Id

**VpcId**  
VpcId of your existing Virtual Private Cloud (VPC)

## 10. Если не используется образ WCS из AWS Marketplace, укажите лицензионный ключ для активации и нажмите `Next`

**WCSLicense**  
WCS License key to activate (optional, if you do not use Marketplace AMI)

## 11. При необходимости, добавьте тэги и права на запуск серверов

Step 1  
Specify template

Step 2  
Specify stack details

Step 3  
**Configure stack options**

Step 4  
Review

### Configure stack options

**Tags**  
You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more](#)

**Permissions**  
Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#)

**IAM role - optional**  
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

## 12. При необходимости, установите дополнительные параметры набора серверов и нажмите `Next`

### Advanced options

You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

- ▶ **Stack policy**  
Defines the resources that you want to protect from unintentional updates during a stack update.
- ▶ **Rollback configuration**  
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#)
- ▶ **Notification options**
- ▶ **Stack creation options**

Cancel Previous **Next**

### 13. Проверьте все параметры набора серверов

### Review wcs-test-stack

Step 1: Specify template Edit

**Template**

Template URL  
https://s3.eu-north-1.amazonaws.com/cf-templates-1c0pwbvffxqz0-eu-north-1/20211695zL-wcs-ec2-template-origin-edge.yml

Stack description  
Create WCS CDN stack

[Estimate cost](#)

Step 2: Specify stack details Edit

**Parameters (18)**

Search parameters

### 14. Нажмите **Create stack**

### Stack creation options

Rollback on failure  
Enabled

Timeout  
-

Termination protection  
Disabled

▶ Quick-create link

Cancel Previous **Create change set** **Create stack**

## 15. Дождитесь окончания создания серверов

Timestamp	Logical ID	Status	Status reason
2021-06-18 14:39:15 UTC+0700	wcs-test-stack	CREATE_COMPLETE	-
2021-06-18 14:39:13 UTC+0700	WCSEdgeInstance	CREATE_COMPLETE	-
2021-06-18 14:39:06 UTC+0700	WCSEdgeInstance	CREATE_IN_PROGRESS	Resource creation initiated
2021-06-18 14:39:04 UTC+0700	WCSEdgeInstance	CREATE_IN_PROGRESS	-
2021-06-18 14:39:01 UTC+0700	WCSOriginInstance	CREATE_COMPLETE	-
2021-06-18 14:38:54 UTC+0700	WCSOriginInstance	CREATE_IN_PROGRESS	Resource creation initiated
2021-06-18 14:38:52 UTC+0700	WCSOriginInstance	CREATE_IN_PROGRESS	-
2021-06-18 14:38:50 UTC+0700	WCSOriginInstance	CREATE_COMPLETE	-
2021-06-18 14:38:47 UTC+0700	WCSOriginInstance	CREATE_IN_PROGRESS	Resource creation initiated
2021-06-18 14:38:42 UTC+0700	WCSOriginInstance	CREATE_IN_PROGRESS	-
2021-06-18 14:38:37 UTC+0700	wcs-test-stack	CREATE_IN_PROGRESS	User initiated

## 16. Перейдите на вкладку **Outputs**

Key	Value	Description	Export name
EdgeInstancecld	i-0a15ee908afcs25a4	Instance Id of newly created WCS Edge instance	-
EdgePrivateIp	172.31.26.94	Private IP address of the Edge instance	-
EdgePublicIp	13.51.156.51	Public IP address of the Edge instance	-
EdgeWebsiteURL	https://ec2-13-51-156-51.eu-north-1.compute.amazonaws.com:8444/admin/	URL for newly created WCS Edge instance web interface. Use instance id as admin password	-
OriginInstancecld	i-06bb1e29aa40acaee	Instance Id of newly created WCS Origin instance	-
OriginPrivateIp	172.31.18.18	Private IP address of the Origin instance	-
OriginPublicIp	13.51.156.202	Public IP address of the Origin instance	-
OriginWebsiteURL	https://ec2-13-51-156-202.eu-north-1.compute.amazonaws.com:8444/admin/	URL for newly created WCS Origin instance web interface. Use instance id as admin password	-

## 17. Откройте веб интерфейсы Origin и Edge, опубликуйте на Origin в примере Two Way Streaming поток test и проигрывайте этот поток на Edge

The first screenshot shows the 'Streaming' interface. On the left, there are several options: Streamer, Player, Two-Way Streaming, Firewall Stream, WebRTC as RTN, Stream Record, Several Stream Recording, Stream Server Snapshot, Stream Local Snapshot, Stream Diagno, Screen Sharing, and Embed Player. The 'Two-Way Streaming' section is active, showing a video player with a rabbit character. Below it, there is a 'test' stream being published. The 'Publish' button is visible, and the status is 'PUBLISHING'. There is also a 'Send payload as object' button.

The second screenshot shows the 'Player' interface. It has the same options as the first screenshot. The 'Two-Way Streaming' section is active, showing the same video player. Below it, there is a 'test' stream being played. The 'test' button is visible, and the status is 'PLAYING'. There is also a 'Send payload as object' button.

