



CDK and infrastructure as code

Курсон Сергей (AWS)

О чём мы поговорим

История работы с управлением инфраструктурой

The AWS Cloud Development Kit (CDK)

Демо: вступление и строим приложение

Плюсы, вещи которые стоит знать, примеры использования

Еще столько всего нужно узнать

Вступление и концепты



Ручной способ

Screenshot of the AWS Lambda console showing a list of recent commands. One command is selected, and its details are shown below.

Command ID	Instance ID	Document name	Status	Requested date	Comment
65555b90-ee60-45...	i-8fd6aa30	AWS-RunPowerSh...	Success	October 21, 2015 at...	Listing services
65555b90-ee60-45...	i-d583f76a	AWS-RunPowerSh...	Success	October 21, 2015 at...	Listing services
65555b90-ee60-45...	i-8ed6aa31	AWS-RunPowerSh...	Success	October 21, 2015 at...	Listing services
ca4b10c6-cee1-437...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	getting list of pro...
561e5f4a-27d2-419...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	ipconfig on the b...

Command ID: 65555b90-ee60-4520-9dc3-e42e94445469 Instance ID: i-8fd6aa30

Description Output

Command ID: 65555b90-ee60-4520-9dc3-e42e94445469
Document name: AWS-RunPowerShellScript
Date requested: October 21, 2015 at 3:56:59 PM UTC-7
Output S3 bucket: run-command-test

Instance ID: i-8fd6aa30
Status: Success
Comment: Listing services
Document parameters: AWS-RunPowerShellScript



👍 Легко начать

🤔 Сложно воспроизвести

🤔 Подвержен ошибкам

🤔 Занимает кучу времени

Высокоур
овневый

Ручной

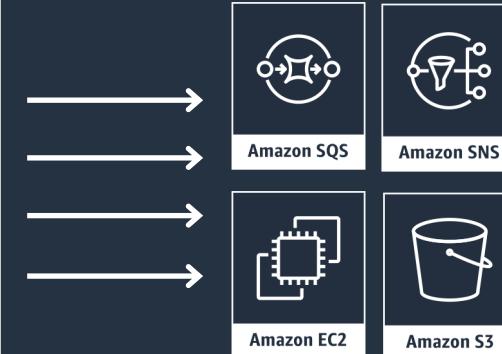
Низкоур
овневый

Скрипты

```
require 'aws-sdk-ec2'

ec2 = Aws::EC2::Resource.new(region: 'us-west-2')

instance = ec2.create_instances({
  image_id: 'IMAGE_ID',
  min_count: 1,
  max_count: 1,
  key_name: 'MyGroovyKeyPair',
  security_group_ids: ['SECURITY_GROUP_ID'],
  instance_type: 't2.micro',
  placement: {
    availability_zone: 'us-west-2a'
  },
  subnet_id: 'SUBNET_ID',
  iam_instance_profile: {
    arn: 'arn:aws:iam::' + 'ACCOUNT_ID' + ':instance-profile/aws-opsworks-ec2-role'
  }
})
```



🤔 Что делать если API вызов не отработает?

🤔 Как накатывать обновления?

🤔 Как узнать что ресурс готов?

🤔 Как откатиться?

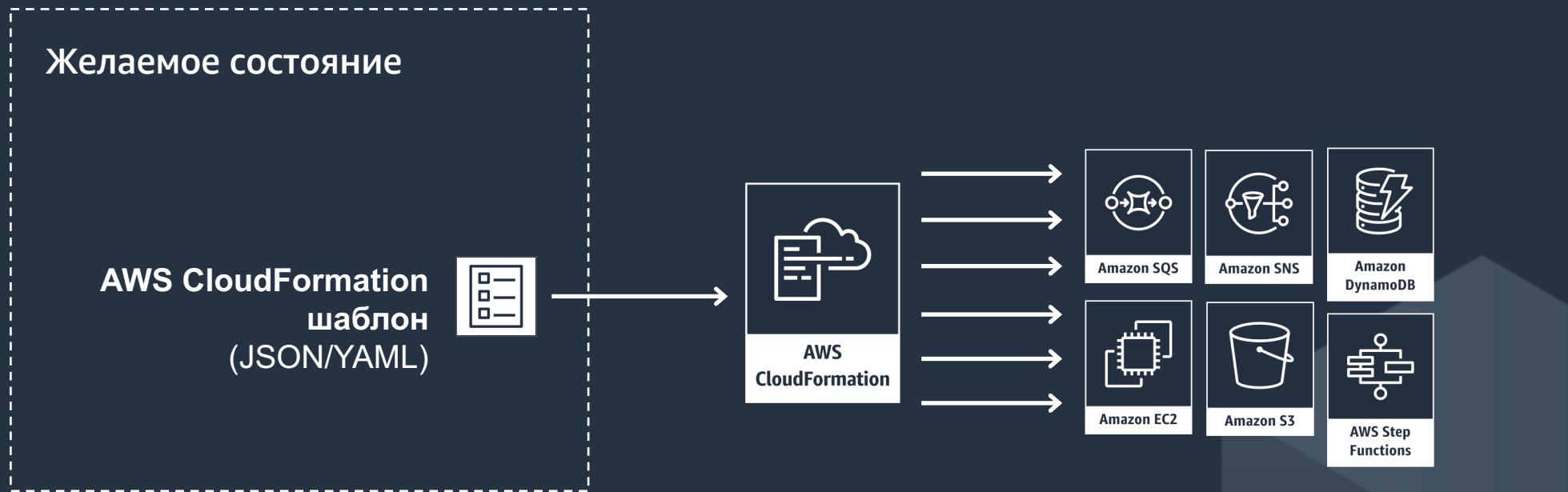
Высокоур
овневый

Закриптованный

Ручной

Низкоур
овневый

Механизмы разворачивания



- 👍 Легко автоматизировать
- 👍 Воспроизводимый
- 👍 «Что» вместо «как»
- 👎 Синтаксис конфигурации
- 👎 Нет абстракций, куча деталей

Высокоуровневый

Декларативный

Закриптованный

Ручной

Низкоуровневый

Проблемы

Сложное выражение бизнес-логики и рефакторинг

Много Copy – paste и неудобные инструменты разработки

Сложное тестирование

Как делиться лучшими практиками?

Высокий порог вхождения

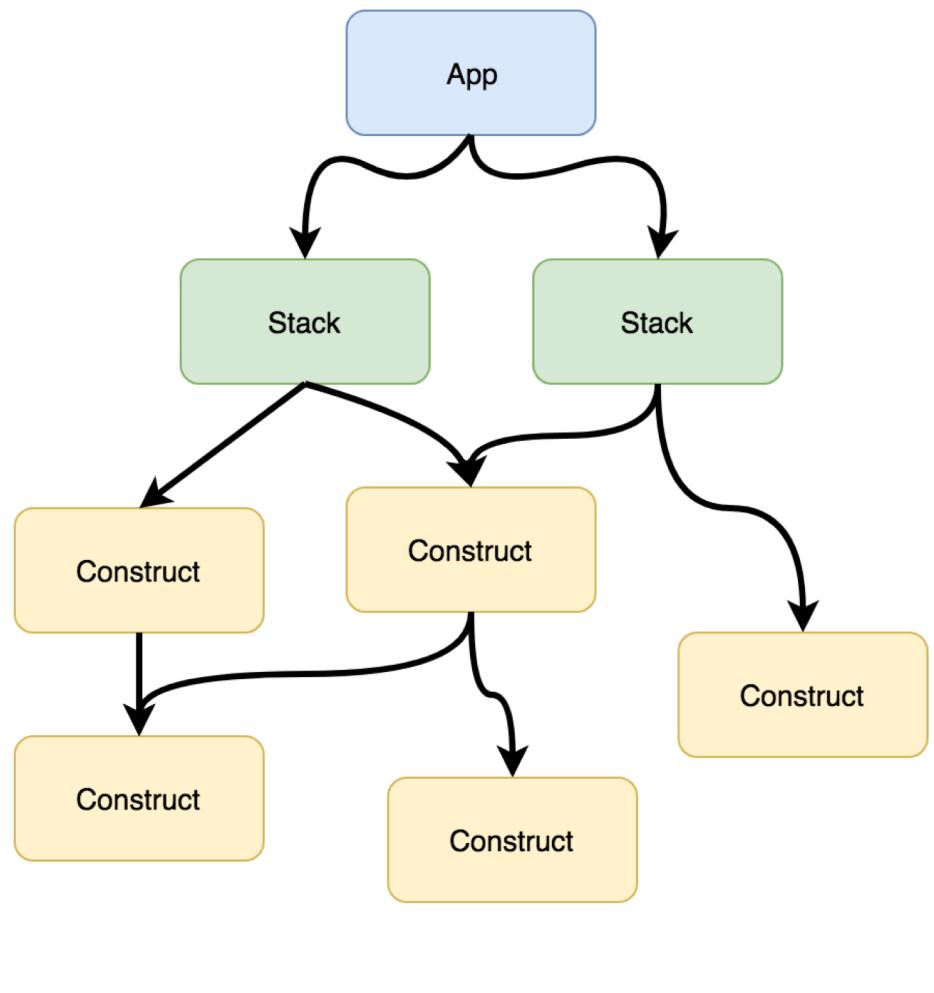
AWS Cloud Development Kit (AWS CDK)



Cloud Development Kit

Apps / Приложения

- Исполняемый код
- Используется для рендера и разворачивания cfn шаблонов



Stacks / СтЭК

- Разворачиваемая единица
- Знает о регионе и аккаунте

Constructs / Конструкт

- Представление ресурсов AWS
- Формирует «дерево конструктов»

Высокоуровневый

Компонентный

Декларативный

Закриптованный

Ручной

Низкоуровневый

AWS Cloud Development Kit design goals

Improve the developer experience

Все конструкты созданы равными

Сократить количество кода и сложность создания инфраструктуры

Оптимизация разработки в стиле CDK-way

Понизить порог вхождения

Сохранить время наших пользователей

3 УРОВНЯ КОНСТРУКТОВ

Уровень 1: Ресурсы CloudFormation (Cfn-based)

- Один-к-одному маппинг на существующие ресурсы
- Автоматически генерируется на основании „AWS Resource Types Reference“

Ex. CfnVpc, CfnCluster

Ресурсы

Фреймворк

3 Уровня конструктов

Уровень 2: Библиотека конструктов AWS

- Написаны на уровне сервисов AWS
- Вручную написаны с применением well-architected практик
- Встроенные принципы least privilege

Ex. cdk-ec2, cdk-ecs

Абстракции

Ресурсы

Фреймворк

3 Уровня конструктов

Уровень 3+: Высокоуровневые конструкты (opinionated)

- Production-ready сервисы
 - Типовые оповещения (alarms)
 - Бэкап и восстановление
- Воспроизводимые решения

Ex. cdk-dynamo-scanner, cdk-ecs-patterns

Целевые

Абстракции

Ресурсы

Фреймворк

Демо 1

Основы

The screenshot shows the Visual Studio Code (VS Code) interface with the title bar "Welcome — ContainersDemoCDK". The left sidebar contains icons for Explorer, Search, Container, Recent, Outline, Problems, Terminal, and Settings. The "OPEN EDITORS" section lists "Welcome" and "CONTAINERDEMOCDK". The "OUTLINE" section indicates "The active editor cannot provide outline information." The main "Welcome" tab has sections for "Start" (New file, Open folder..., Add workspace folder...), "Customize" (Tools and languages, Install support for JavaScript, TypeScript, P...), and "Recent" (cdkContainersDemo, meetupdemosample). The bottom navigation bar includes PROBLEMS, TERMINAL, and a three-dot menu. The terminal shows two sessions: "f018982d5b20:ContainersDemoCDK kuronsk" and "f018982d5b20:ContainersDemoCDK kuronsk". The status bar at the bottom shows "0 0" and a smiley face icon.

EXPLORER

OPEN EDITORS

Welcome

CONTAINERDEMOCDK

OUTLINE

The active editor
cannot provide outline
information.

PROBLEMS

TERMINAL

...

1: bash, bash

f018982d5b20:ContainersDemoCDK kuronsk

\$

f018982d5b20:ContainersDemoCDK kuronsk

\$

Welcome

Start

New file

Open folder...

Add workspace folder...

Customize

Tools and languages

Install support for [JavaScript](#), [TypeScript](#), P...

Recent

cdkContainersDemo ~/Documents/AWS/Sam...

meetupdemosample ~/Documents/AWS/Sam...

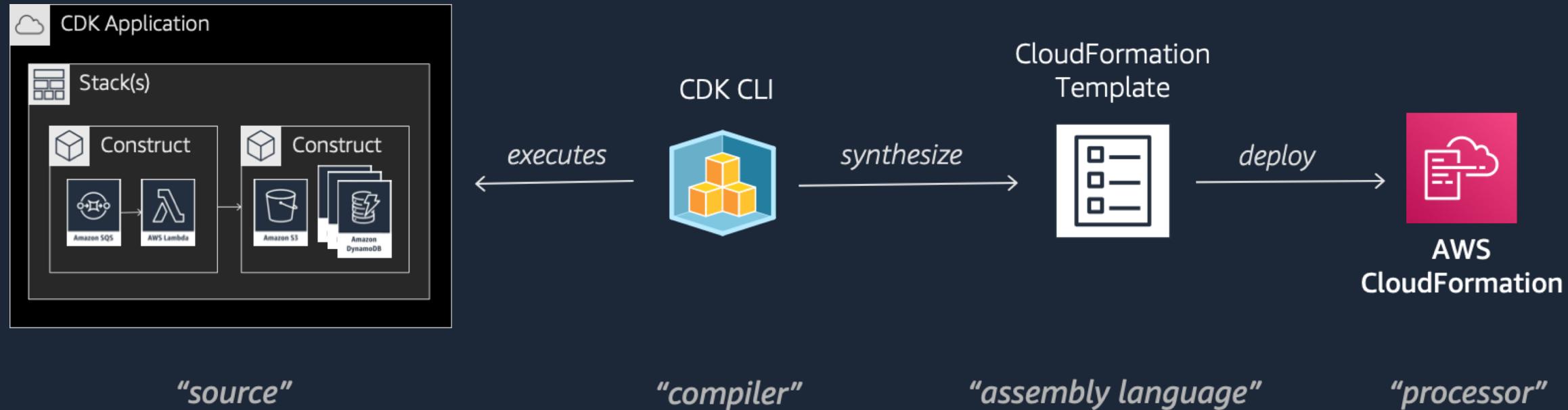
Color theme

Settings and keybindings

Install the settings and keyboard shortcuts ...

AWS Cloud Development Kit (AWS CDK)

Широкий взгляд—от приложения AWS CDK к готовой инфраструктуре



CDK + code = Love

- Строгая типизация
 - Намного проще рефакторинг / поддержка intellisense
 - Secret store / Vault
 - Пример: использовать код + SDK чтобы проверить существование контейнера перед разворачиванием в ECS

```
1 reference
public ContainersD
{
    var vpc = new
    {
        MaxAzs = 2
    });
    var ec2 = new Instance_()
}
```

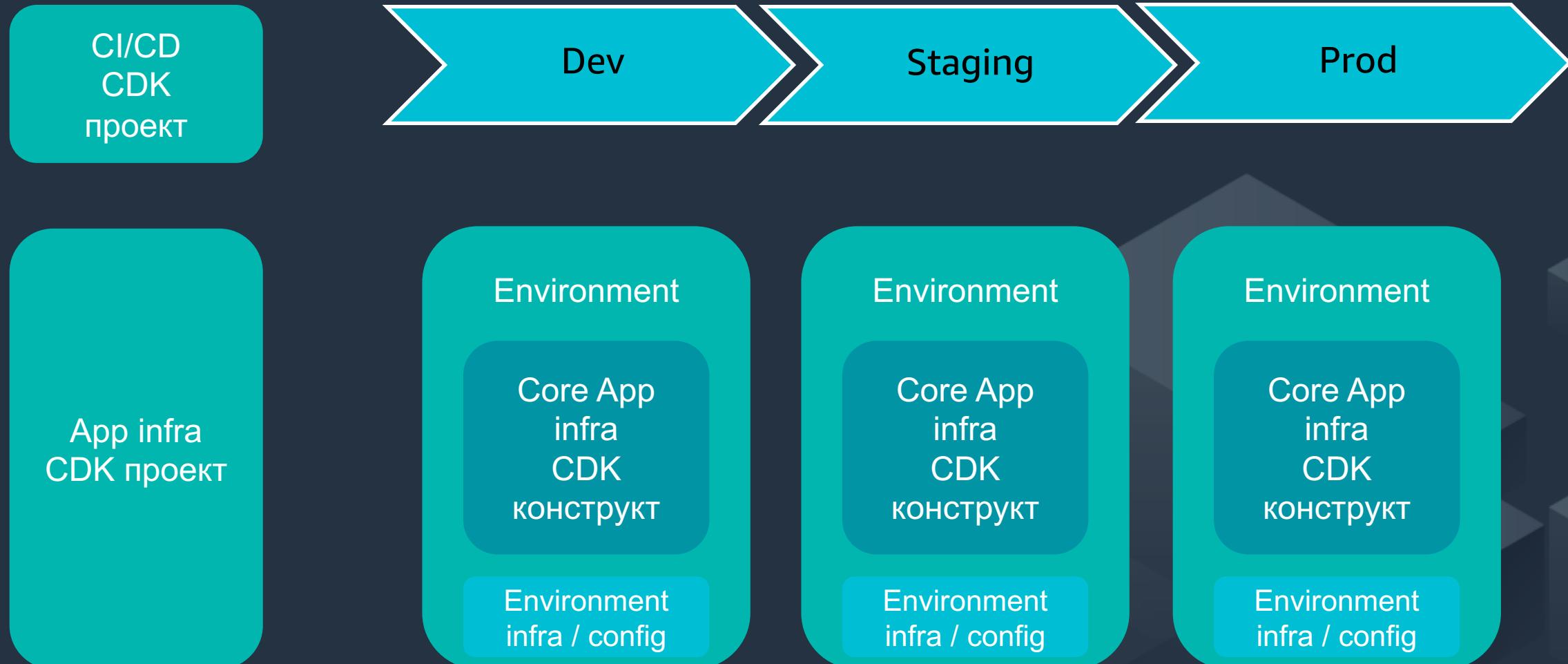
```
var ecs = new Cluster(this, "DemoDotnetCluster"
    Vpc = vpc
});

var fargateService = new NetworkLoadBalancedF
    Cluster = ecs,
    TaskImageOptions = new NetworkLoadBalancedF
        Image = ContainerImage.FromAsset("src")
    }
}
```

OUTPUT DEBUG CONSOLE TERMINAL

```
oDotnetFargateService/LoadBalancerDNS DemoDotnetFargateServiceLB8A26B06C", "DNSName"] }}  
ContainersDemoCDKSharp kursonsk$ 
```

Пайплайны



Тестирование CDK

- «Snapshot» тесты (используется командой CDK чтобы тестировать изменения)
- Integration тесты (используется командой CDK чтобы тестировать развернутую инфраструктуру в целом)
- Fine-grained предположения о шаблоне

```
expect(stack).toHaveResource('AWS::SQS::Queue',  
    { MessageRetentionPeriod: 1209600 }  
);
```
- Validation тесты

```
expect(() => {  
    new dlq.DeadLetterQueue(stack, 'DLQ', {  
        retentionDays: 15  
    });  
}).toThrowError(/retentionDays may not exceed 14 days/);
```



Рекомендуемые практики тестирования CDK

- **Код теста – код приложения.**
Не надо копипастить общие assertions.
Общий код должен быть абстрагирован.
- **Не нужно слишком много assert в одном тесте.**
Тест покрывает только одно поведение
Одна поломка – один упавший тест

О чём стоит знать

- Конструкты 3 уровня – черный ящик. Обязательны проверки конструктов из публичных источников
- Cfn-классы часто необходимы
- Необходимы тесты инфраструктуры CloudFormation перед запуском чтобы не было разницы с реальной инфраструктурой

Как сегодня используется CDK

- Чистый CDK
- Гибридный подход
 - Гибрид для миграции в существующих CI/CD
 - Внутренние ресурсы вроде пайплайнов разработки
 - Гибрид с подключением готовых cfn шаблонов
 - Гибрид для сервисов вроде service catalog
- Обучение, демо, тестовые проекты

CDK команды

`cdk bootstrap` - Устанавливает bootstrap CDK kit на регион

`cdk init` - Локально создает новый CDK проект

`Cdk diff` - Показывает разницу между текущим стэком-инфраструктурой и стэком CDK.

добавляем необходимые библиотеки

```
npm install @aws-cdk/aws-lambda @aws-cdk/aws-apigateway
```

Компилируем:

e.g. `npm run build`

`cdk synth` - Синтезирует и выдает CloudFormation шаблон для стэка

`cdk deploy` - Разворачивает стэк на аккаунт

`cdk destroy` - Уничтожает стэк

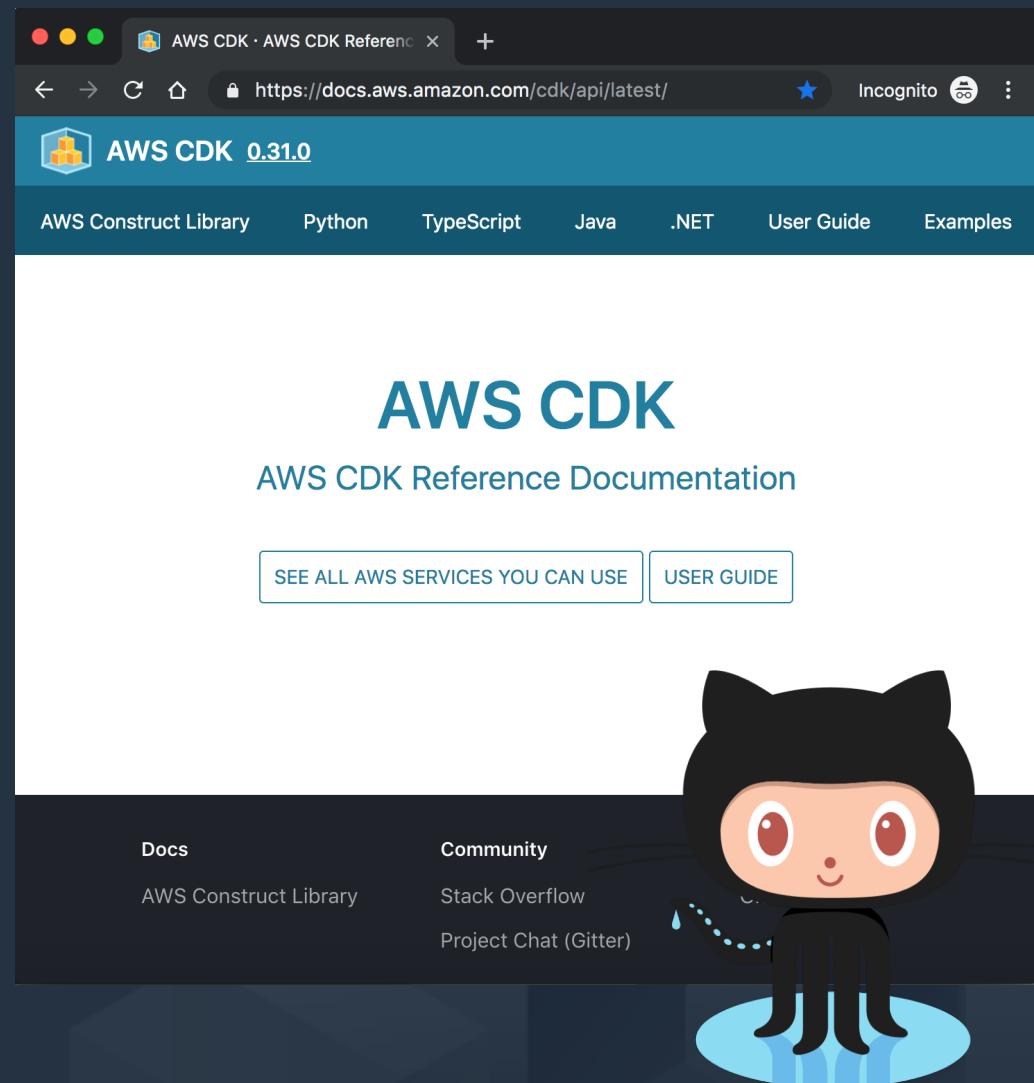
Следующие шаги

Пробуем CDK

- <https://cdkworkshop.com>
- <https://docs.aws.amazon.com/cdk/api/latest/>
- <https://github.com/eladb/awesome-cdk>

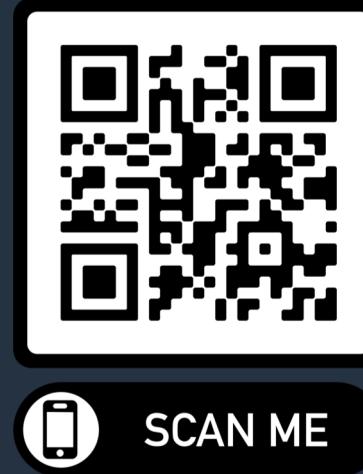
Сообщества

- <https://github.com/awslabs/aws-cdk>
- <https://gitter.im/awslabs/aws-cdk>
- <https://github.com/eladb/awesome-cdk>



AWS Content hub

Подпишитесь на контент-хаб, чтобы получать приглашения на мероприятия в регионе



<https://pages.awscloud.com/EMEA-In-T1-selfregistrationpage-2019-interest.html>

Спасибо!

Sergey Kurson

Email: kuronsk@amazon.de