

Доброе утро!

**Wi-Fi: Deworkacy
Pwd: 333555777**

**Пожалуйста, выполните
подготовительные шаги:**

<https://mikhail.io/miacp>



Инфраструктура Azure как код

DevOps Pro | Москва | 19 ноября 2019

Михаил Шилков

Михаил Шилков

- Россия / Нидерланды
- Разработчик ПО / фрилансер
- Cloud / Serverless
- Functional programming
- Microsoft Azure MVP

<https://mikhail.io>

© Mikhail Shilkov



Расписание

10:00 - 11:30 Инфраструктура как код, Azure Resource Manager templates

11:30 - 11:40 Перерыв

11:40 - 13:00 Terraform

13:00 - 14:00 Обед

14:00 - 15:20 Pulumi

15:20 - 15:30 Перерыв

15:30 - 17:00 Hacking time

4 сессии по ~1.5 часа

~10 минут – вводная лекция

~1 час – практическое задание

~10 минут – обзор и обсуждение

Инфраструктура как код



Проблемы с ручным управлением

Деградация

Различия в конфигурации «одинаковых» серверов

Уникальность

Никто не знает точно, как можно создать такой же сервер заново

Неожиданные ошибки

Сложности в тестировании изменений

Инфраструктура как код (ИкК)

Создание и выполнение
кода, который **определяет**,
устанавливает и обновляет
инфраструктуру

Управление всем через
написание кода, включая
сервера, базы данных, сети,
логи, конфигурацию,
документацию, тесты,
развертывание

Преимущества ИкК

- Повторяемость
- Контроль версий
- Повторное использование
- Документация
- Скорость и безопасность
- Счастье 😊

Виды ИкК

Скрипты (bash, PowerShell)

Configuration management
(Puppet, Chef – управление сервером)

Server templating
(Docker, Vagrant)

Orchestration (Kubernetes)

Provisioning
(создание инфраструктуры)

Виды ИкК

Скрипты (bash, PowerShell)

Configuration management
(Puppet, Chef – управление сервером)

Server templating
(Docker, Vagrant)

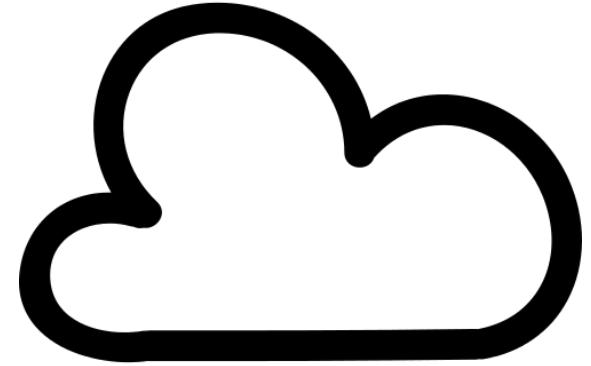
Orchestration (Kubernetes)

Provisioning
(создание инфраструктуры)

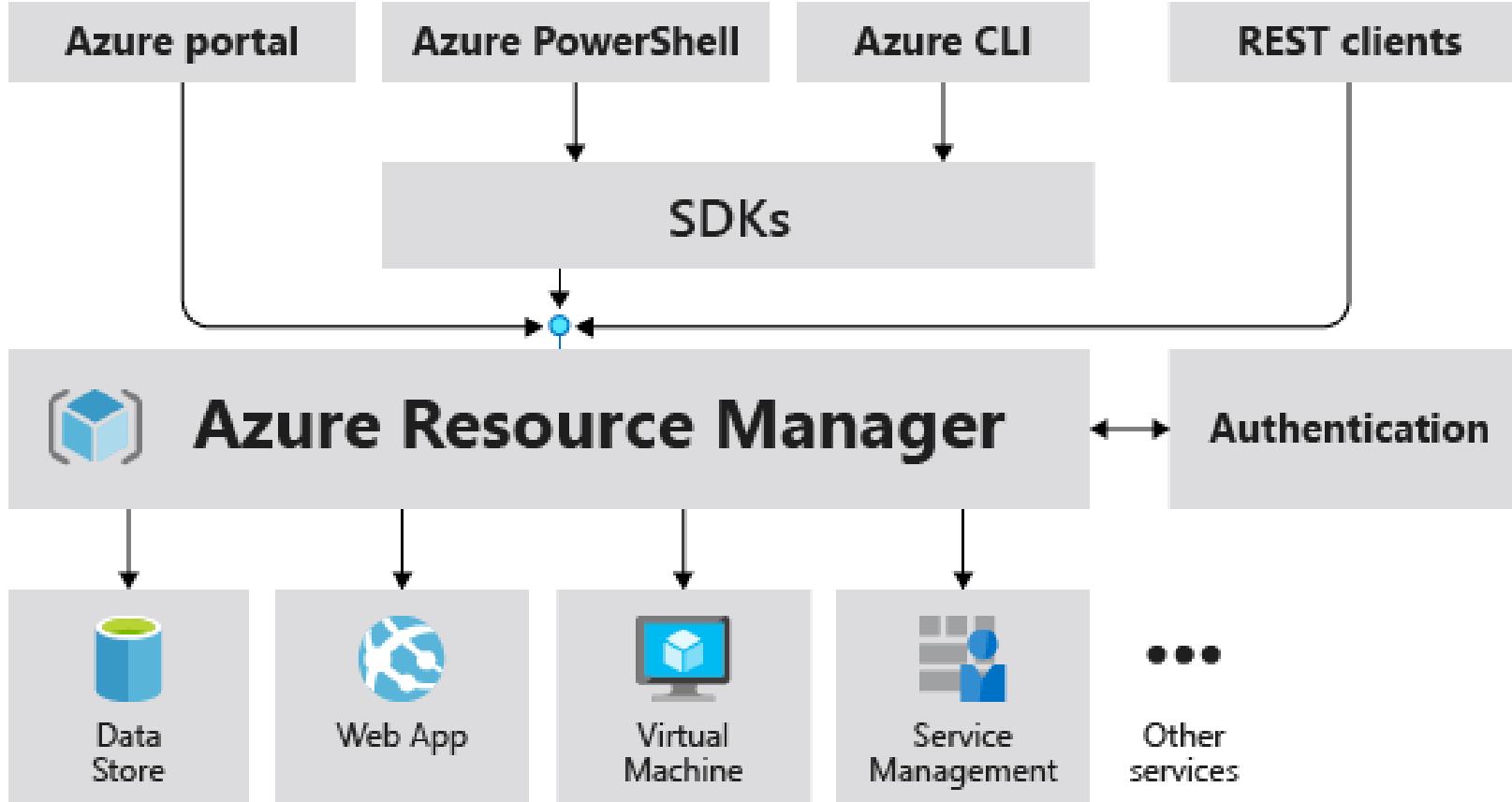
Компромиссы в ИкК

- Процедурный /
Декларативный
- Источник управления
(агент/клиент/сервер)
- Формат и язык
- Один/много
провайдеров
- Покрытие
- Зрелость, экосистема

Автоматизация Cloud



Единый API для управления



Сервисы HTTP/REST

```
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Amz-Date: 20130813T150211Z
Host: ec2.amazonaws.com
Authorization: AWS4-HMAC-SHA256 Credential=AKIDEXAMPLE/20130813/us-east-
1/ec2/aws4_request, SignedHeaders=content-type;host;x-amz-
date, Signature=ced6826de92d2bdeed8f846f0bf508e8559e98e4b0194b84example54174deb456c
```

```
http://ec2.amazonaws.com/?Action=RunInstances
ImageId=ami-2bb65342
&MaxCount=3
&MinCount=1
&Monitoring.Enabled=true
&Placement.AvailabilityZone=us-east-1a
&Version=2016-11-15
```

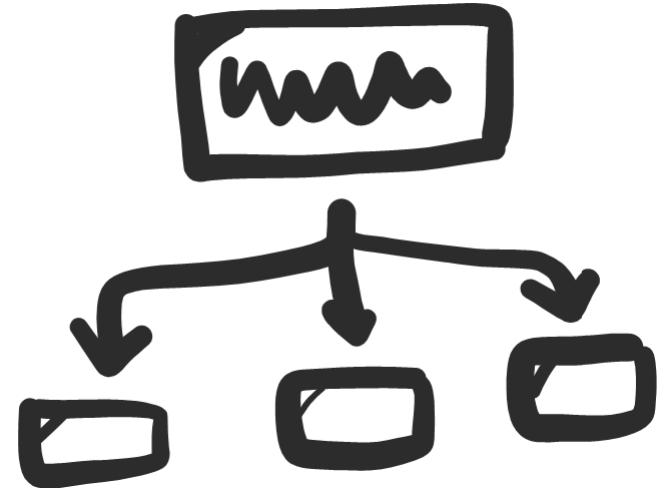
SDK для вызова REST сервисов

```
var storageAccount = az.StorageAccounts.Define(storageAccountName)
    .WithRegion(Region.US_EAST)
    .WithNewResourceGroup(resourceGroupName)
    .Create();
```

Командная строка

```
az storage account create \
    --location westus \
    --name samplesa \
    --resource-group myrg \
    --sku LRS
```

Конфигурация желаемого состояния



Конфигурация желаемого состояния



Управление графами ресурсов



Важные аспекты

Выразительность

Способность
описать целевое
состояние: полно,
кратко, верно

Зависимости

Определить
порядок создания и
уничтожения

Предсказуемость

Предварительный
просмотр
изменений до
запуска

Инструменты от вендоров



AWS
CloudFormation



Azure
Resource
Manager



Google Cloud
Deployment
Manager

Сервисы Microsoft Azure

Что мы будем создавать сегодня



Resource Group

© Mikhail Shilkov

**Каждый ресурс принадлежит
одной группе**

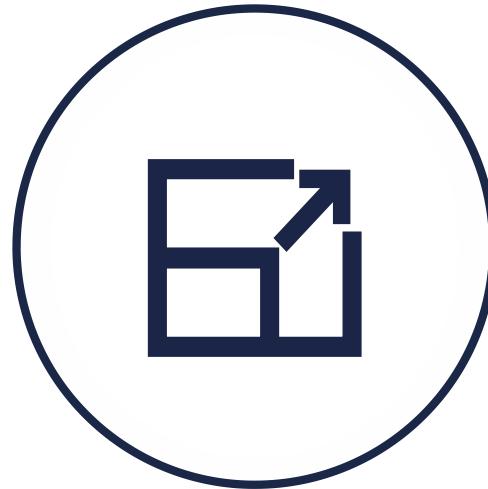
**Ресурсы в одной группе
изменяются вместе**

**Разрешены ресурсы в одной
группе, но в разных регионах**

Azure Functions



Делегированное
управление



Масштабируемость и
высокая доступность
из коробки

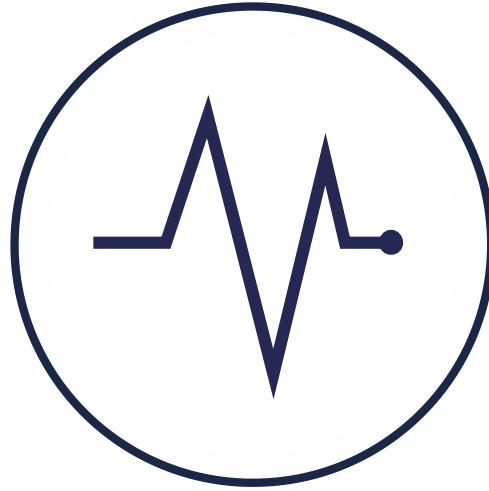


Оплата за
фактическое
использование

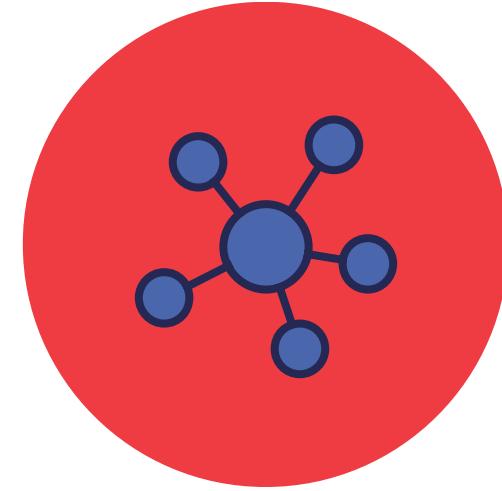
Azure Functions



Запуск по
событию



Короткое
время
выполнения



Состояние не
сохраняется

Azure Functions: примеры использования

Таймер

Выполнить скрипт
PowerShell один
раз в день

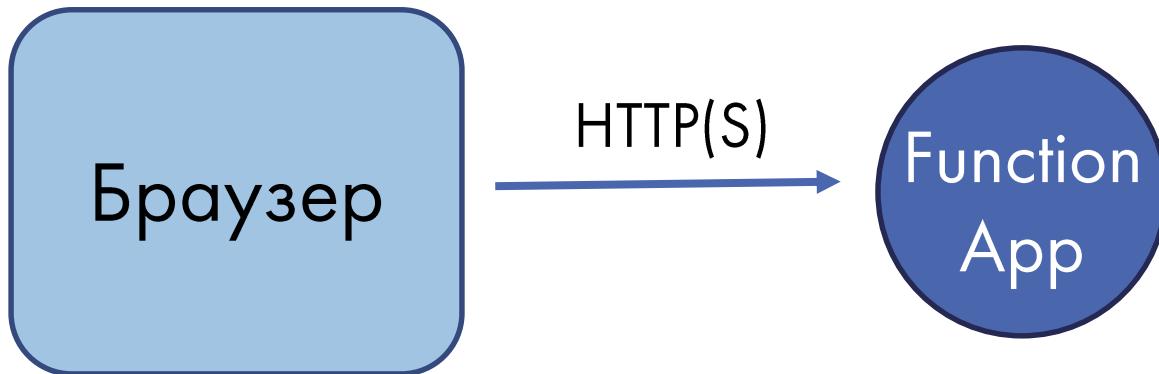
HTTP

Высоконагруженные
веб-сервисы с
сотнями запросов в
секунду

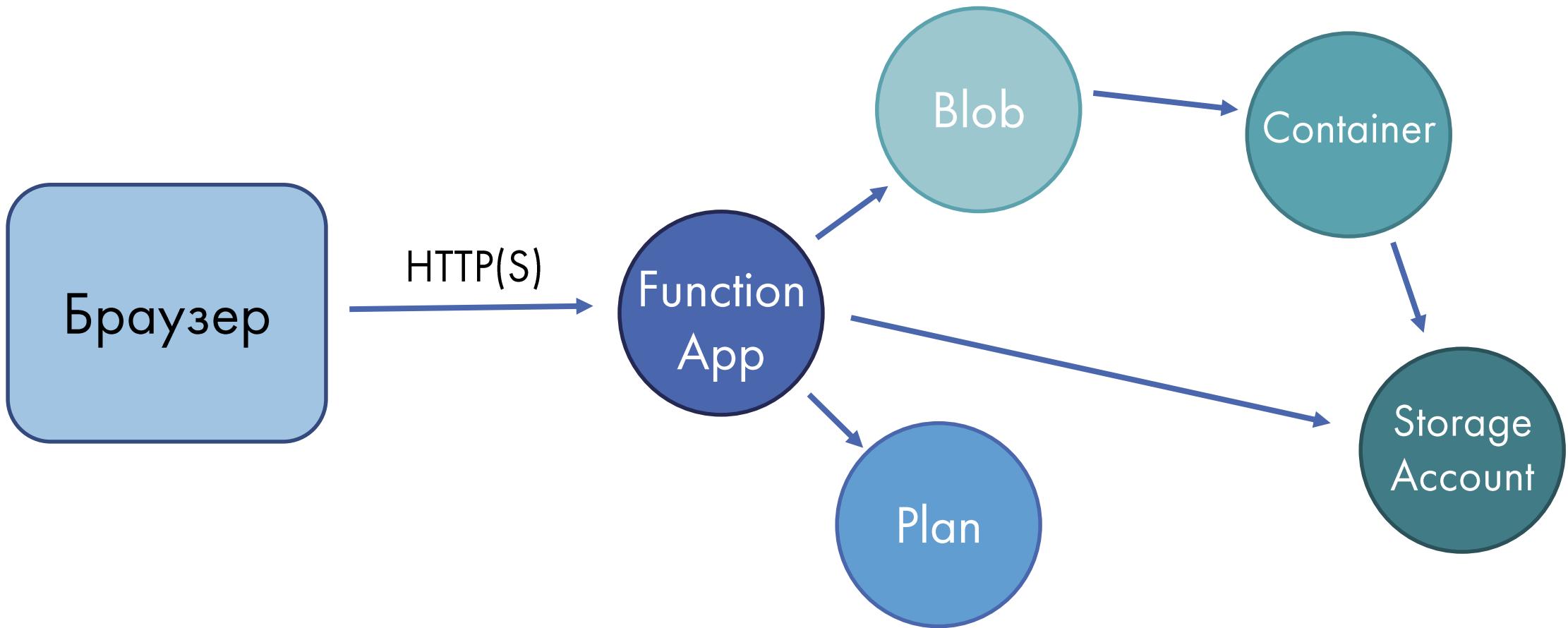
Очереди

Всплески входящих
сообщений, разное
время обработки

Ресурсы, нужные для Azure Function

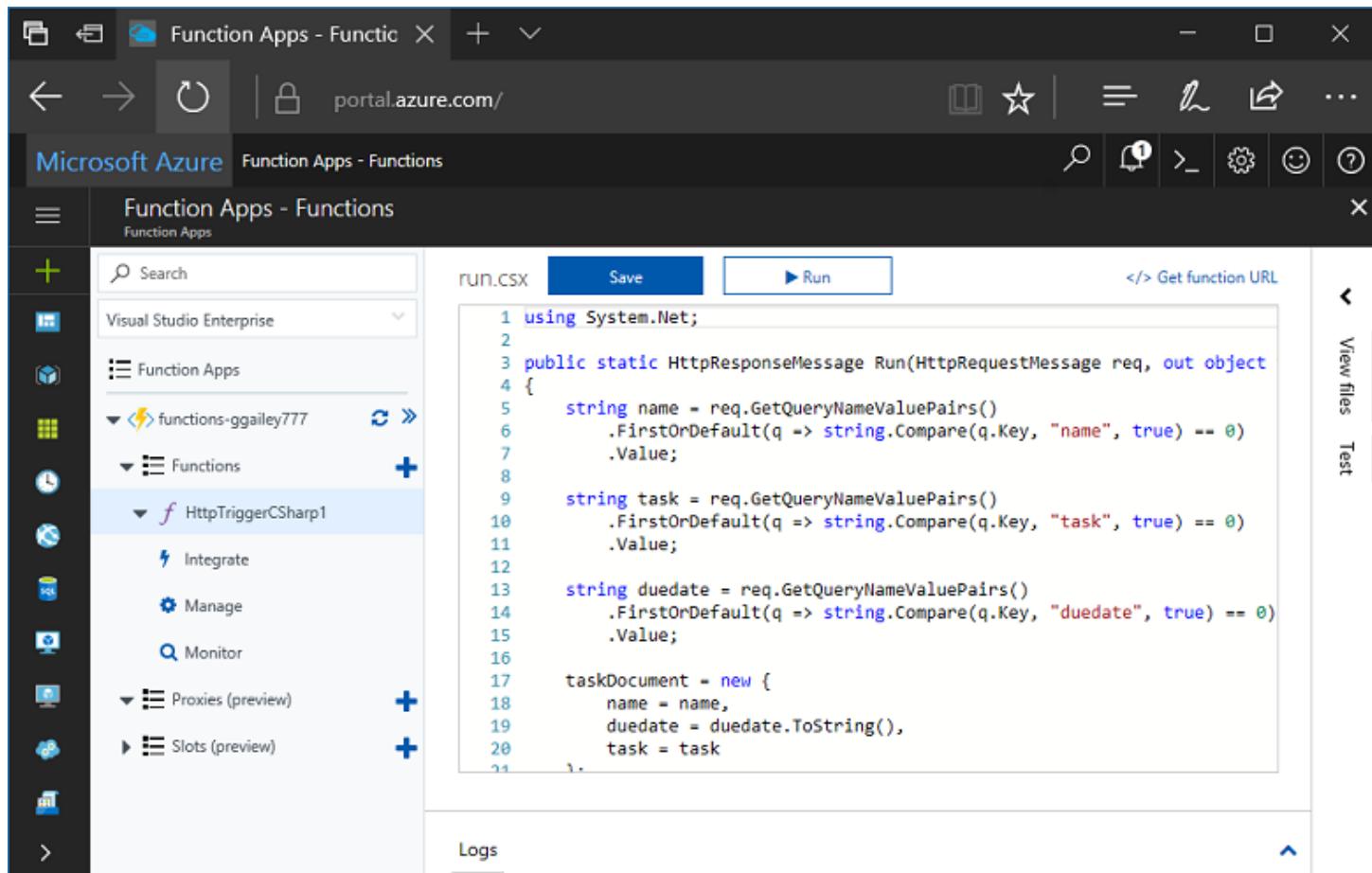


Ресурсы, нужные для Azure Function

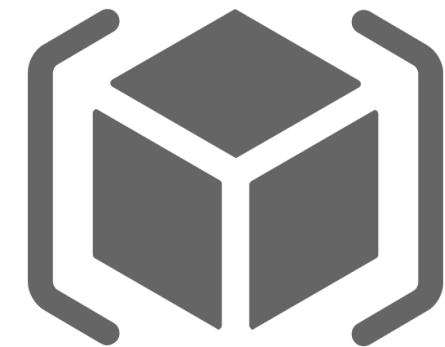


Azure Portal

Легко начать,
но не стоит
использовать
в продакшне



ARM Templates



Формат: JSON (почти)

```
"resources": [
  {
    "apiVersion": "2016-01-01", // You can do comments
    "type": "Microsoft.Storage/storageAccounts",
    "name": "mystorageaccount",
    "location": "westus",
    "sku": {
      "name": "Standard_LRS"
    },
    "kind": "Storage",
    "properties": {
    }
  }
]
```

Преобразуется в REST вызовы

```
PUT https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/  
{resourceGroupName}/providers/Microsoft.Storage/storageAccounts/mystorageaccount?api-  
version=2016-01-01  
{  
  "location": "westus",  
  "properties": {  
  },  
  "sku": {  
    "name": "Standard_LRS"  
  },  
  "kind": "Storage"  
}
```

Pecypc Deployment

The screenshot shows the 'Overview' page for a deployment named 'Microsoft.VirtualNetwork-20190822124104'. The left sidebar includes 'Search (Ctrl+/' and navigation links for 'Overview', 'Inputs', 'Outputs', and 'Template'. The main area displays a success message: 'Your deployment is complete'. It provides deployment details: name (Microsoft.VirtualNetwork-20190822...), start time (8/22/2019, 12:41:04 PM), subscription (Documentation Testing 1), resource group (newgroup), and correlation ID (a2a03520-91d3-4e82-...). It also features sections for 'Deployment details (Download)' and 'Next steps', and a 'Go to resource' button.

Microsoft.VirtualNetwork-20190822124104 - Overview

Deployment

Search (Ctrl+ /) «

✓ Your deployment is complete

Deployment name: Microsoft.VirtualNetwork-20190822... Start time: 8/22/2019, 12:41:04 PM
Subscription: Documentation Testing 1 Correlation ID: a2a03520-91d3-4e82-...
Resource group: newgroup

▼ Deployment details ([Download](#))

^ Next steps

[Go to resource](#)

Overview

Inputs

Outputs

Template

Delete Cancel Redeploy Refresh

Как выполнять задания

**Не обязательно выполнить
все шаги: не надо спешить!**

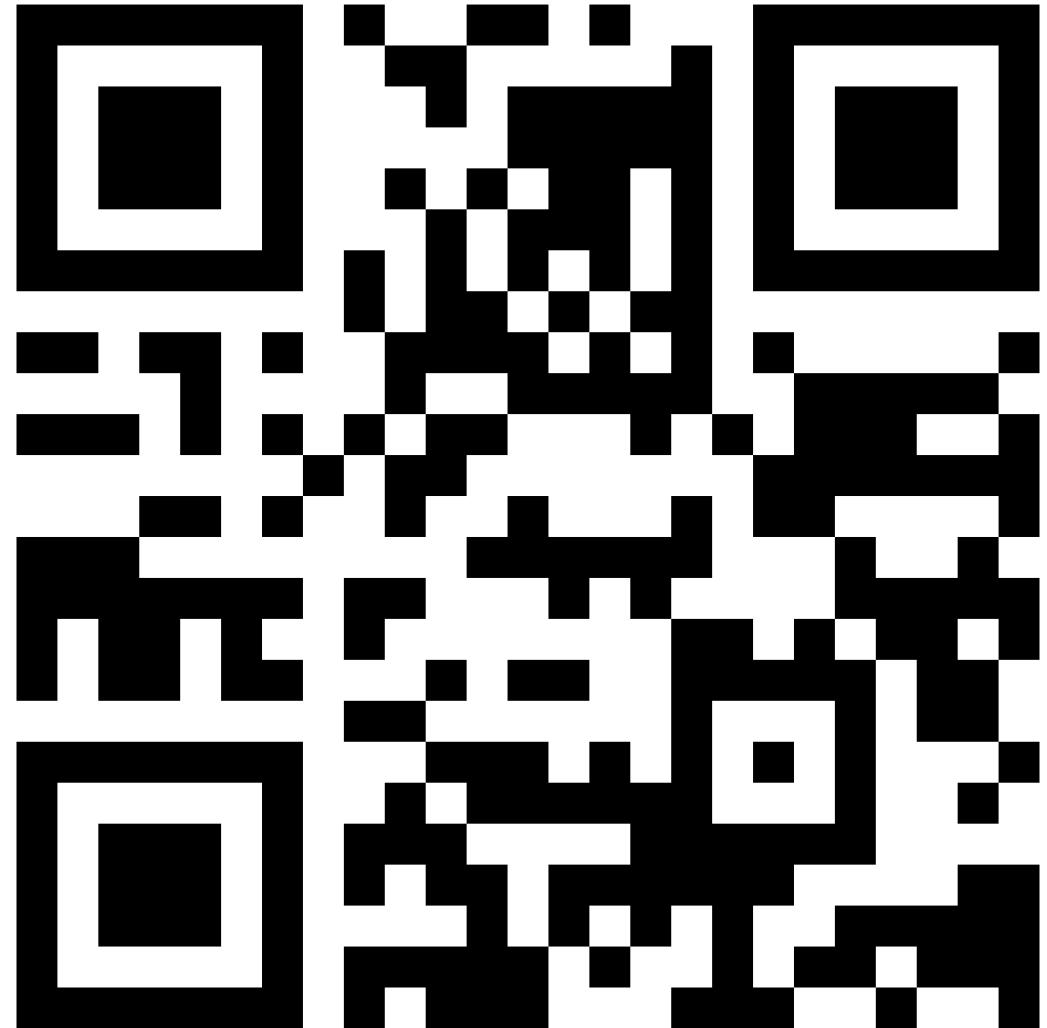
**Самостоятельно
или в парах**

Задавайте вопросы

Получайте удовольствие 😊

Практическое задание 1:

[https://mikhail.io/
lab/arm/](https://mikhail.io/lab/arm/)



ARM Templates: обзор

- Процедурный /
Декларативный
- Источник управления
(агент/клиент/сервер)
- Формат: JSON
- Один/много
провайдер
- Покрытие: все
возможности ARM
- Создано Microsoft,
закрытый код



Перерыв
увидимся в 11:40

Сложности с ARM Templates

Один вендор

Инструменты и навыки применимы только к Azure

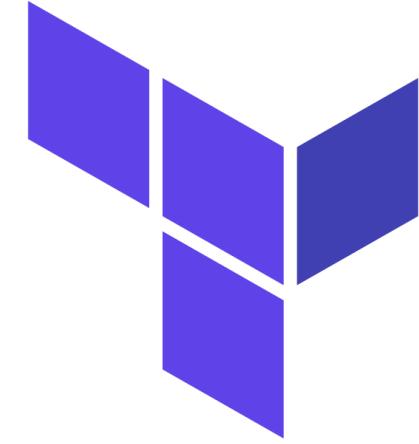
Примитивный UX

Сложность работы с JSON, нет предпросмотра, сложность создания с нуля

Закрытый

Много «legacy», невозможно повлиять извне

Terraform



Terraform

Создано HashiCorp
в 2014 году

Также: Vagrant, Packer,
Vault, Consul

Написан на Go,
открытый код,
лицензия MPL 2.0

HashiCorp Configuration Language (HCL)

```
resource "azurerm_storage_account" "sa" {
    name          = "mysa"
    location      = "${azurerm_resource_group.rg.location}"
    resource_group_name = "${azurerm_resource_group.rg.name}"
    account_tier   = "Standard"
    account_replication_type = "LRS"
}
```

HCL 2.0 в Terraform 0.12

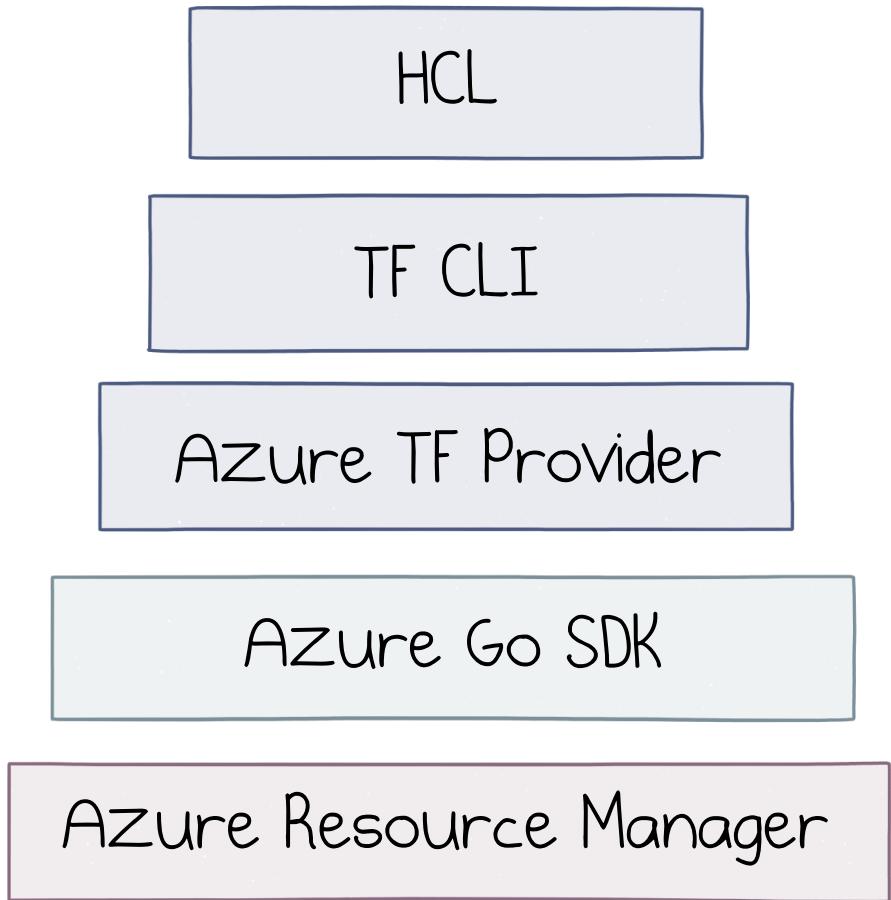
```
resource "azurerm_storage_account" "sa" {
    name          = "mysa"
    location      = azurerm_resource_group.rg.location
    resource_group_name = azurerm_resource_group.rg.name
    account_tier   = "Standard"
    account_replication_type = "LRS"
}
```

Провайдеры

- AWS
- Azure
- GCP
- Digital Ocean
- Cloudflare
- Docker
- Kubernetes
- Vault
- PostgreSQL
- New Relic

... и 100+ других

Как работает Terraform



Workflow: Plan, Apply, Destroy

```
terraform init
```

```
# Загрузить провайдеры
```

```
terraform plan
```

```
# Создать и показать черновой план изменений
```

```
terraform apply
```

```
# Произвести изменения в инфраструктуре
```

```
terraform destroy
```

```
# Удалить все ресурсы
```

State File

Текстовый файл (JSON)

Текущее состояние
инфраструктуры

Хранится на локальном диске или
на внешнем хранилище
(например Azure Storage)

Контроль одновременного
выполнения

“Freemium”

**Terraform CLI используется
бесплатно**

**Платный Terraform Cloud для
команд**

Практическое задание 2:

[https://mikhail.io/
lab/terraform/](https://mikhail.io/lab/terraform/)



Terraform: обзор

- Процедурный / Декларативный
- Источник управления (агент/клиент/сервер)
- Формат: HCL
- Один/много провайдеров
- Покрытие может быть ограничено
- Открытый код, существует 5 лет



Языки разметки

Инструменты

Создаются с нуля,
медленная
обратная связь

Структура

Какие свойства
нужно заполнить и
каковы возможные
значения

Размер

Тысячи строк
разметки: сложно
создавать, читать,
менять,
поддерживать

Ссылки на ресурсы

```
{  
  "type": "Microsoft.Web/sites",  
  "apiVersion": "2016-08-01",  
  "name": "[parameters('appName')]",  
  "location": "[resourceGroup().location]",  
  "kind": "functionapp",  
  "properties": {  
    "serverFarmId": "[resourceId('Microsoft.Web/serverfarms', variables('planName'))]"  
  }  
}
```

УСЛОВИЯ

```
{  
  "condition": "[equals(parameters('production'), 'Yes')]",  
  "type": "Microsoft.Compute/availabilitySets",  
  "apiVersion": "2017-03-30",  
  "name": "[variables('availabilitySetName')]",  
  "location": "[resourceGroup().location]",  
  "properties": {  
    "platformFaultDomainCount": 2,  
    "platformUpdateDomainCount": 3  
  }  
}
```

ЦИКЛЫ

```
locals {  
    files = ["index.php","main.php"]  
}  
  
resource "aws_s3_bucket_object" "files" {  
    count  = "${length(local.files)}"  
    key    = "${local.files[count.index]}"  
    bucket = "${aws_s3_bucket.examplebucket.id}"  
    source = "./src/${local.files[count.index]}"  
}
```

Повторное использование

Конфигурация

Похожая, но
разная
инфраструктура
для production,
staging, testing

Абстракция

Создание
высокоуровневых
компонентов из
низкоуровневых

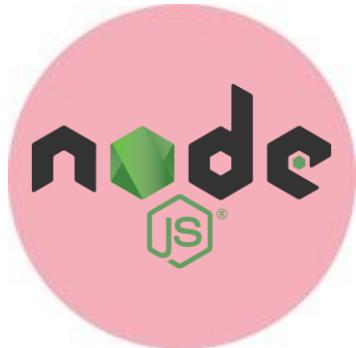
Пакеты

Распространение
библиотек и
компонентов
внутри компании
или сообщества

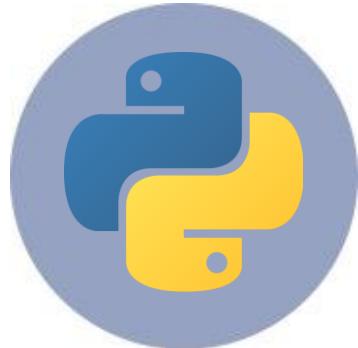
Pulumi



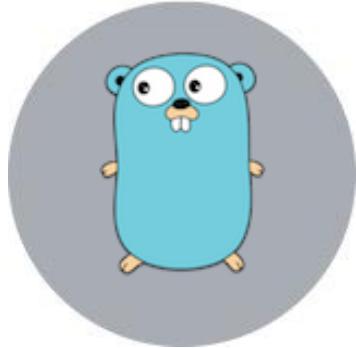
Языки программирования



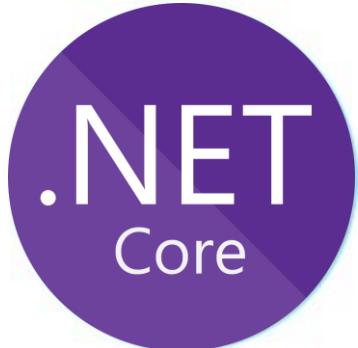
Node.js



Python



Go



.NET

Pulumi

Создано Pulumi Corp.
в 2018 году

Написан на Go,
открытый код,
Apache License 2.0

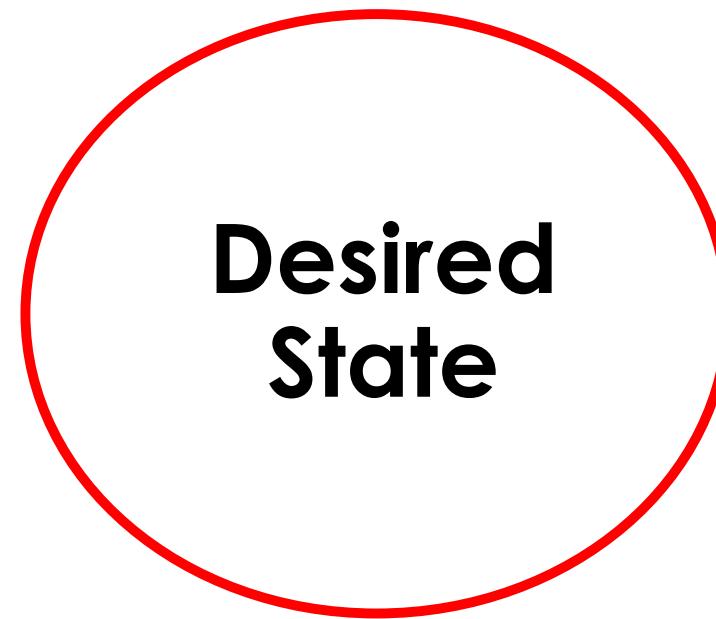
Пример: TypeScript

```
const resourceGroup = new azure.core.ResourceGroup("rg");

const storageAccount = new azure.storage.Account("storage", {
    resourceGroupName: resourceGroup.name,
    accountReplicationType: "LRS",
    accountTier: "Standard",
});
```

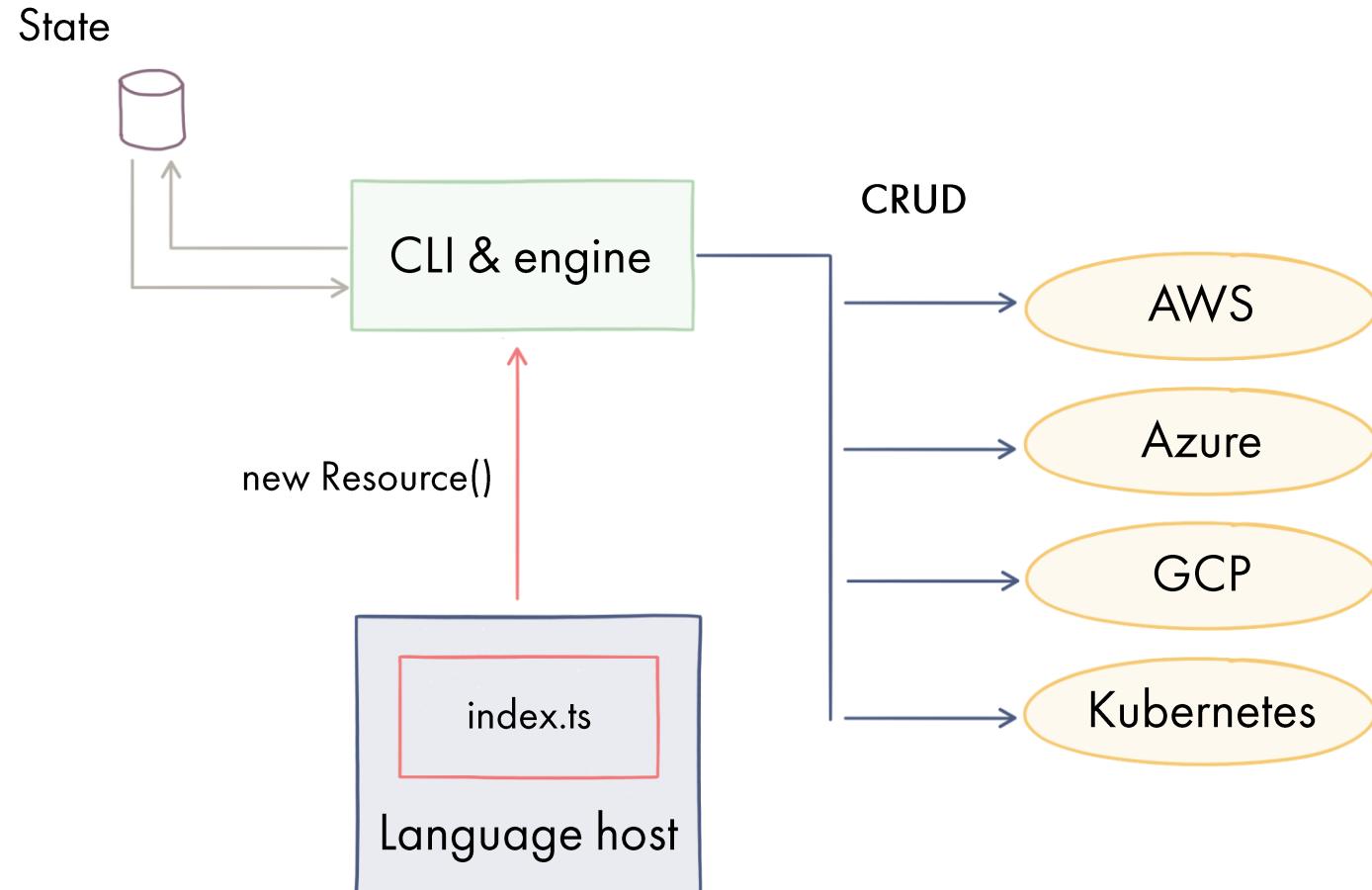
```
const resourceGroup = new azure.core.ResourceGroup("rg");

const storageAccount = new azure.storage.Account("storage", {
    resourceGroupName: resourceGroup.name,
    accountReplicationType: "LRS",
    accountTier: "Standard",
});
```



Desired
State

Как работает Pulumi



Провайдеры

- AWS
- Azure
- GCP
- Digital Ocean
- Cloudflare
- Docker
- Kubernetes
- OpenStack
- PostgreSQL
- New Relic

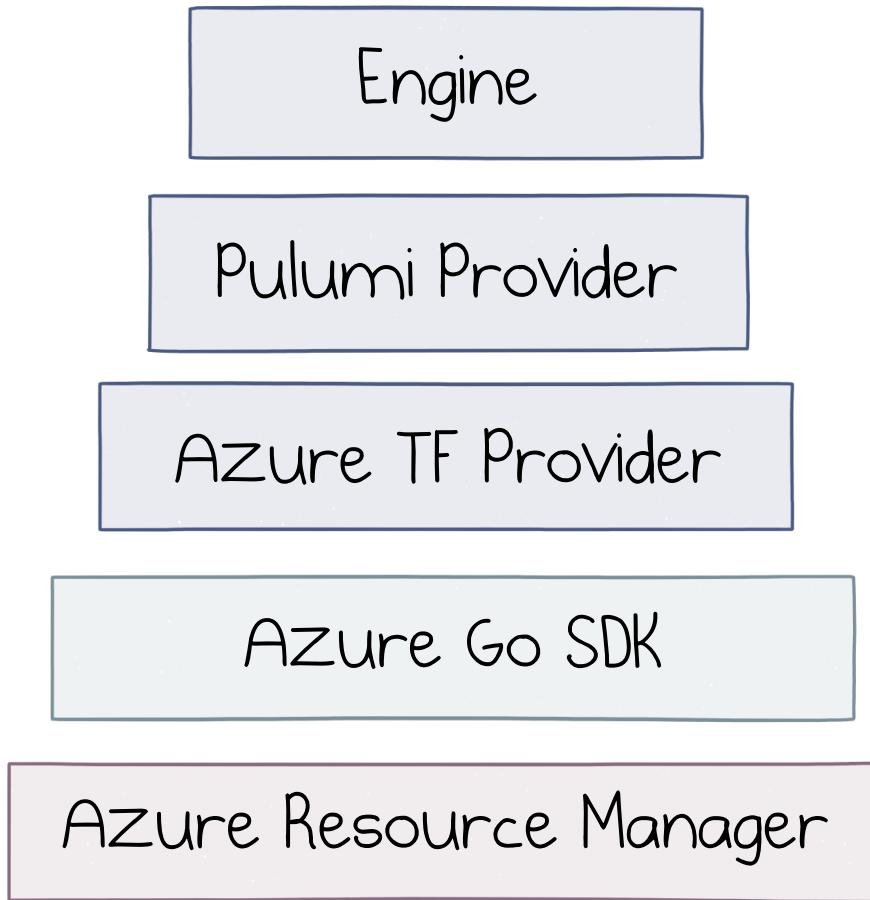
... и другие

Pulumi- Terraform Bridge

**Многие провайдеры –
обёртка вокруг Terraform
провайдеров**

Примеры: AWS, Azure

Контрпример: Kubernetes



Как работает провайдер **Pulumi Azure**

Workflow: Preview, Up, Destroy

```
pulumi preview
```

```
# Отобразить план изменений
```

```
pulumi up
```

```
# Обновить инфраструктуру
```

```
pulumi destroy
```

```
# Удалить все ресурсы из текущего стека
```

State File

© Mikhail Shilkov

Текстовый файл (JSON)

Текущее состояние
инфраструктуры

В сервисе Pulumi, на диске, или
в Azure Storage

Контроль одновременного
выполнения

Stack и State

```
pulumi refresh
```

```
# Определить «drift» и обновить текущее состояние
```

```
pulumi stack
```

```
# Выбрать одну из сред выполнения
```

“Freemium”

Бесплатная Pulumi CLI

**Платный Pulumi Cloud
для команд**

Практическое задание 3:

[https://mikhail.io/
lab/pulumi/](https://mikhail.io/lab/pulumi/)



Pulumi: обзор

- Процедурный / Декларативный
- Источник управления (агент/клиент/сервер)
- Формат:
TypeScript/Python/C#/...
- Один/много провайдеров
- Покрытие может быть ограничено
- Открытый код,
достаточно новый



Перерыв
увидимся в 15:30

Что мы не
успели
обсудить?



Azure Management SDKs

```
var storageAccount = az.StorageAccounts.Define(storageAccountName)
    .WithRegion(Region.US_EAST)
    .WithNewResourceGroup(resourceGroupName)
    .Create();
```

Azure SDKs (.NET/Node/Go/...)

- Процедурный /
Декларативный
- Источник управления
(агент/клиент/сервер)
- Формат:
TypeScript/Python/C#/...
- Один/много
провайдер
- Покрытие: хорошее
- «Сам себе хозяин»

Azure CLI

```
az storage account create \
    --location westus \
    --name samplesa \
    --resource-group myrg \
    --sku LRS
```

Сложности со скриптами

Процедурные

Процесс
описывается шаг
за шагом

Сложные

Миграция
зависит от
текущего
состояния

Обработка ошибок

Что делать, если
скрипт упал?

Azure CLI

- Процедурный /
Декларативный
- Источник управления
(агент/клиент/сервер)
- Формат:
Bash/PowerShell
- Один/много
провайдер
- Покрытие: хорошее
- «Сам себе хозяин»

Генераторы ARM (ex: Farmer)

```
// This is F#
let myStorage = storageAccount {
    name "mystorage"                      // set account name
    sku Storage.Sku.PremiumLRS             // use Premium LRS
}
```

Генераторы ARM

- Процедурный /
Декларативный
- Источник управления
(агент/клиент/сервер)
- Формат:
F#, C#, Python, ...
- Один/много
провайдер
- Покрытие:
ограниченное
- Создано
энтузиастами

Другие темы

© Mikhail Shilkov

Обработка ошибок

Обновления без downtime

Декомпозиция

Создание компонентов

Взаимодействие в команде

Безопасность и аудит

Множественный среды

Тестирование

Другие темы (2)

Хакатон

Придумай задачу!



Идеи

- Можно доделать предыдущее задание, если не успели
- Попробуйте другие виды ресурсов
(VM, App Service, AKS, ACI, KeyVault, Cosmos DB,)
- Попробуйте AWS / GCP (Terraform/Pulumi)
- Попробуйте Python / C# (Pulumi)
- Да что угодно придёт в голову!
- Некоторые идеи – по ссылке на следующем слайде

Хакатон:

[https://mikhail.io/
miaci/](https://mikhail.io/miaci/)

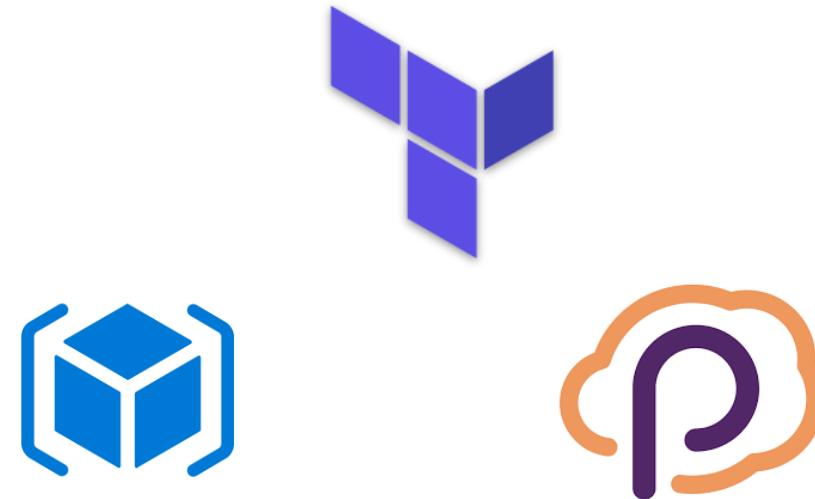


Выводы



Создаёте
приложения
для cloud?

Используйте
Инфраструктуру как
код



Меньшие
компоненты

Лучшая
автоматизация

Слайды:

[https://mikhail.io/
miac/slides.pdf](https://mikhail.io/miac/slides.pdf)





Thank you!

@MikhailShilkov
<https://mikhail.io>