



DevCon School
Технологии будущего

Гибридные приложения и практики DevOps на базе Azure Stack

Александр Шаповал, Андрей Выставкин, Сергей Груздов

Александр Шаповал | @ashapoval

Эксперт по стратегическим технологиям, Microsoft

ashapo@microsoft.com

<https://habrahabr.ru/company/microsoft/>



Более 20 лет в ИТ-индустрии

Работает в Microsoft с 2003 года

Занимается взаимодействием с ISV-партнерами, специализируется на вопросах использования Azure IaaS и гибридных облаков

Отвечал за содержание конференций Платформа 2010, Платформа 2011, TechEd Russia 2011

Автор многочисленных докладов и курсов на порталах Channel9, TechDays и MVA

Андрей Выставкин

Менеджер по маркетингу гибридных
инфраструктур, Microsoft

andvis@microsoft.com

Работает в Microsoft 8,5 лет

Последние три года занимается развитием бизнеса вокруг Microsoft Azure в
России. С февраля 2017 года очень много времени уделяет Azure Stack и OMS



Сергей Груздов

Ведущий инженер, Dataline

egel@egel.su

В настоящий момент работает ведущим инженером в компании Dataline. Занимается внедрением, поддержкой и развитием инфраструктур, в частности публичного облака Cloud-V, на базе продуктов Microsoft: Windows Server, System Center.



Интенсив проводится при технологической поддержке:

NVision Group



Softline



Что такое ARM?

Что представляет собой ARM API

Какова структура шаблонов ARM

Как использовать ARM-шаблоны для развертывания инфраструктуры

#msdevcon

Что такое Azure Stack?

Какие сервисы поддерживает MAS

Какова архитектура MAS

Как осуществляется управление ресурсами MAS

Как выглядят гибридные приложения?

Каковы основные паттерны гибридных приложений

Как использовать ARM для гибридных решений

Как использовать практики DevOps вместе с MAS



Программа

11:30 - 12:30	Модель управления ресурсами Azure Resource Manager (ARM). Структура ARM-шаблонов
12:30 - 13:30	Практическая работа. Развёртывание инфраструктуры на основе ARM-шаблонов
13:30 - 14:30	Обед
14:30 - 15:30	Архитектура и возможности Azure Stack
15:30 - 16:30	Практическая работа. Управление ресурсами Azure Stack для запуска приложений
16:30 - 17:00	Перерыв
17:00 - 18:00	Гибридные приложения в Microsoft Azure и Azure Stack
18:00 - 19:00	Практическая работа. Развёртывание гибридного приложения

Модель управления ресурсами Azure Resource Manager (ARM). Структура ARM-шаблонов

#msdevcon

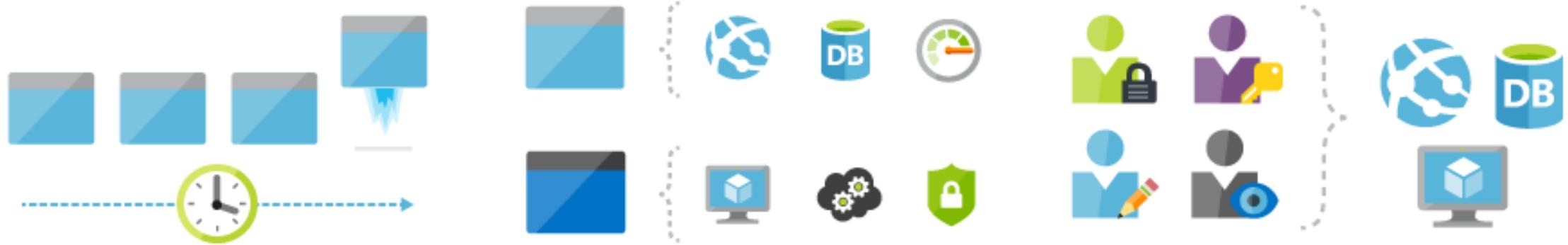
Azure Service Management (ASM)

- Azure Service Management – первая версия API, которая предоставляет программный доступ к возможностям Azure
 - Предоставляется в виде REST API
- ASM основан на XML, для создания и запуска скриптов могут использоваться PowerShell, CLI
- ASM может использоваться для настройки таких ресурсов как: Cloud Services, Storage accounts, Virtual Networks
- Пакеты облачных приложений требуют наличия сертификатов, которые отделены от кода и загружаются с помощью портала управления

ASM: недостатки и ограничения

- Сложно управлять большим количеством ресурсов в рамках организации
- Ограничены средства контроля доступа
 - Доп. администратор подписки для привилегированных действий
- Ограниченный средства аудита, доступные на портале
 - **List Subscription Operations** возвращает список операций **create**, **update** и **delete**, выполненных в рамках подписки за определенный интервал времени

Направления усовершенствований



Развертывание

Структурирование

Контроль

Azure Resource Manager (ARM)

уровень
управления

Инструменты



Microsoft Azure



Command line

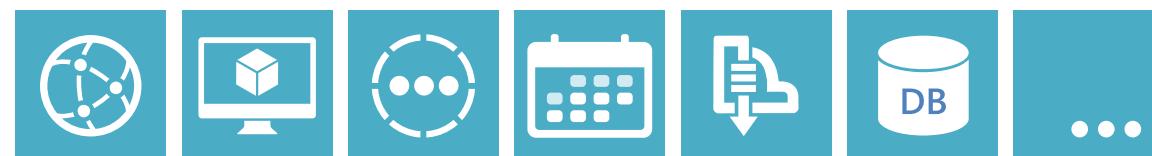


Visual Studio

Расширения

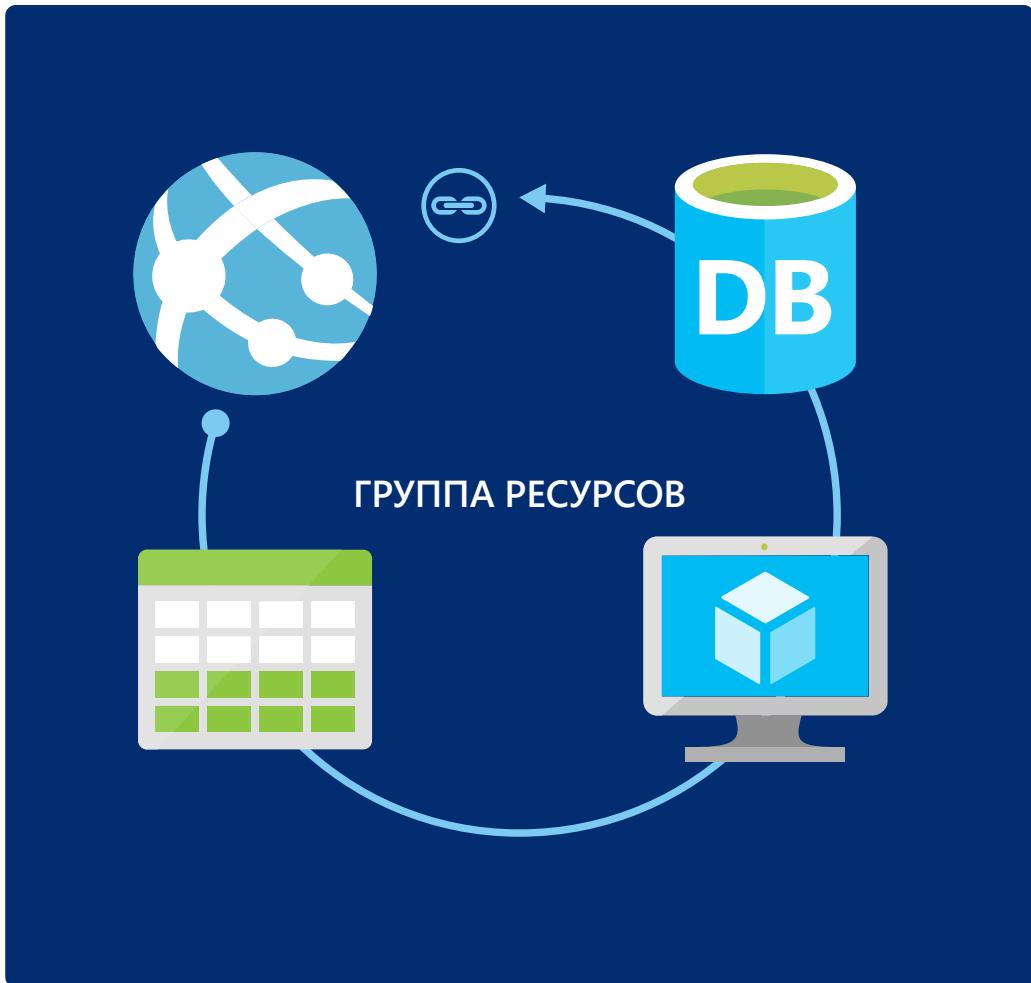


Провайдеры



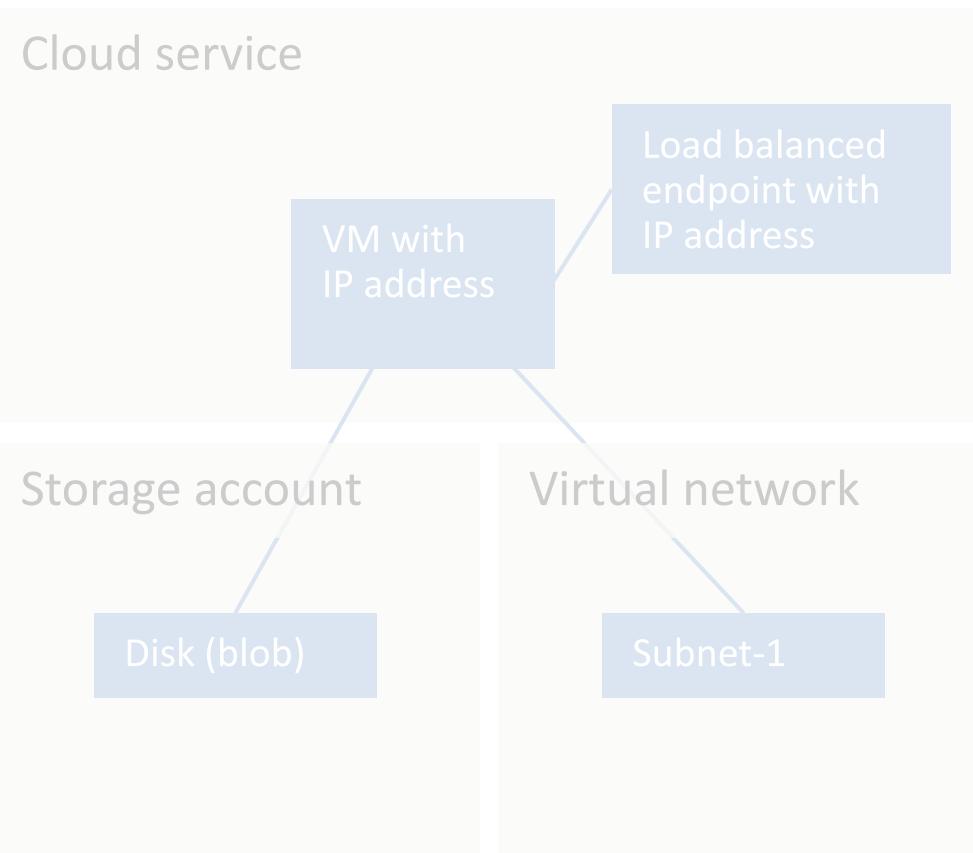
Группы ресурсов

- Контейнеры с множеством экземпляров ресурсов
- Каждый экземпляр относится к определенному типу ресурса
- Типы ресурсов определяются провайдерами ресурсов
- Каждый ресурс должен принадлежать одной и только одной группе ресурсов

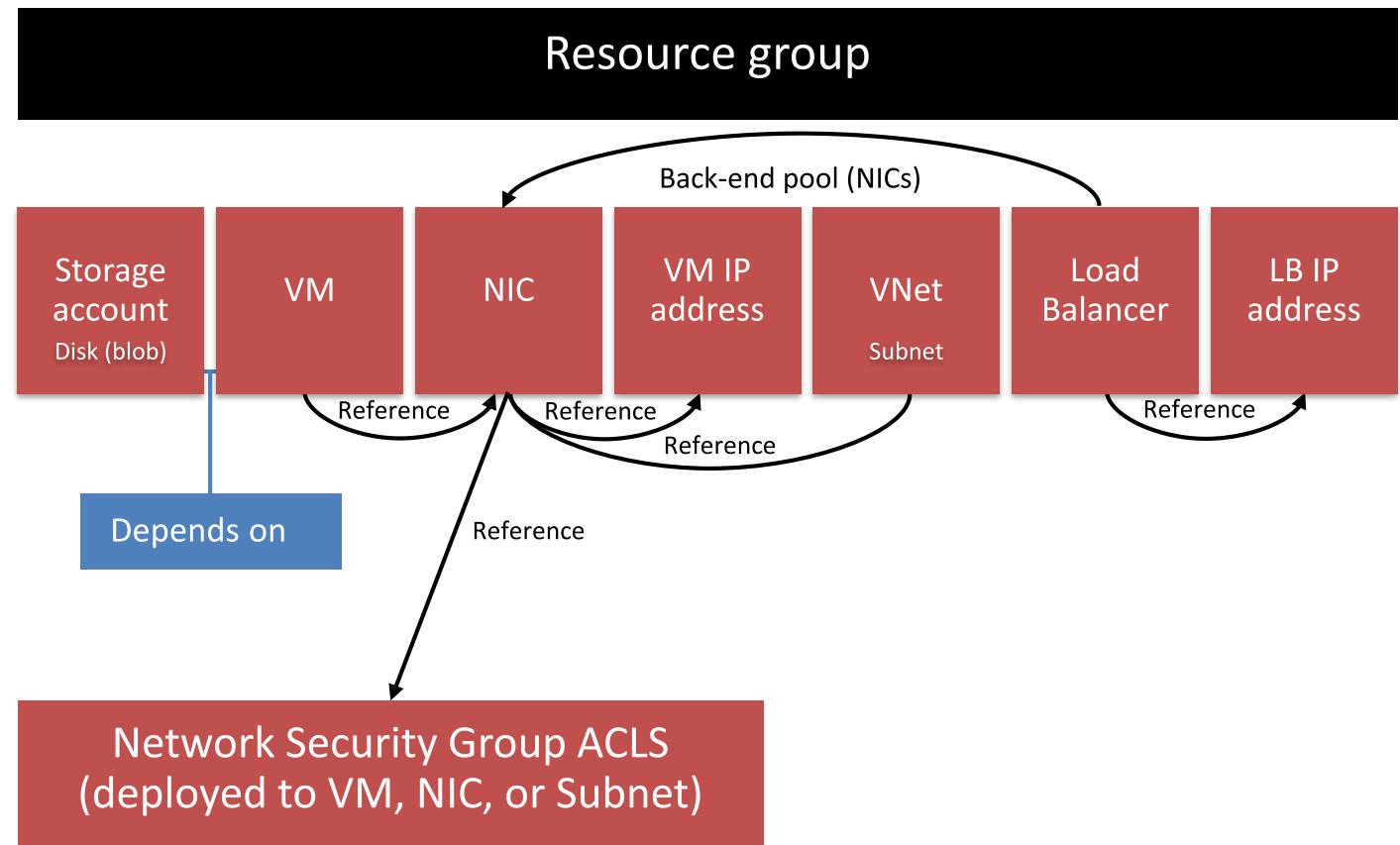


Пример использования Resource Manager

Классическая модель (v1)



Resource Manager (v2)



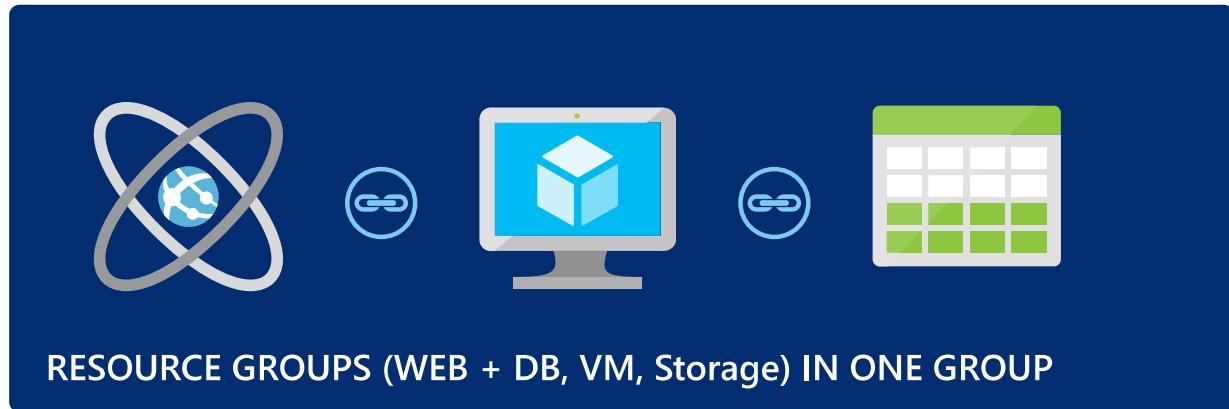
Жизненный цикл группы ресурсов

Вопрос:

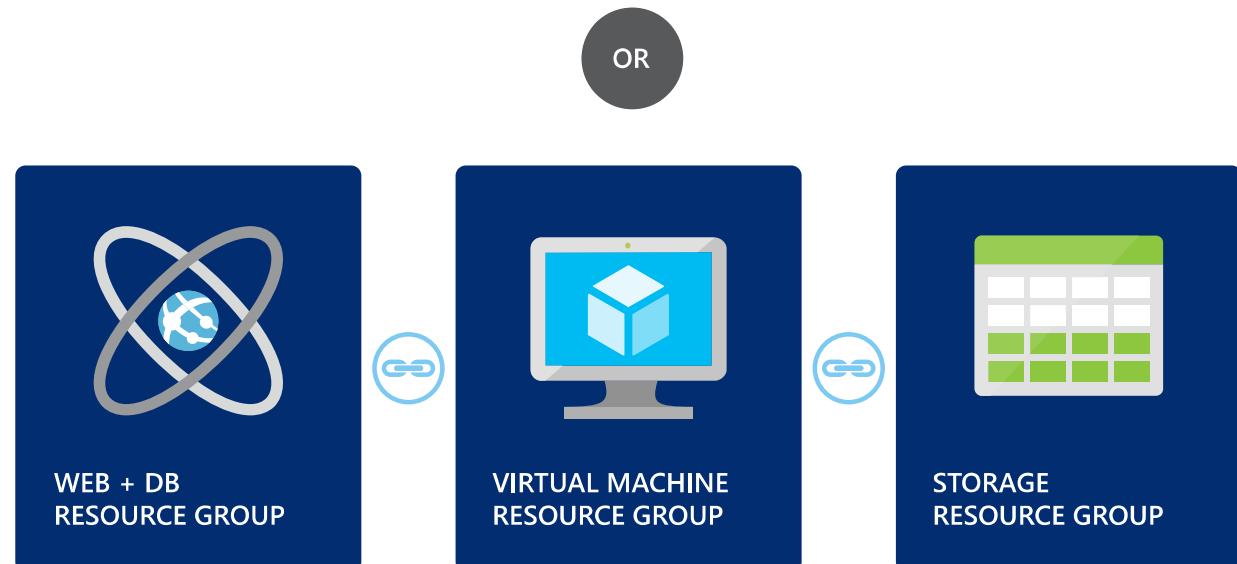
Должны некоторые ресурсы принадлежать одной группе или разным?

Ответ:

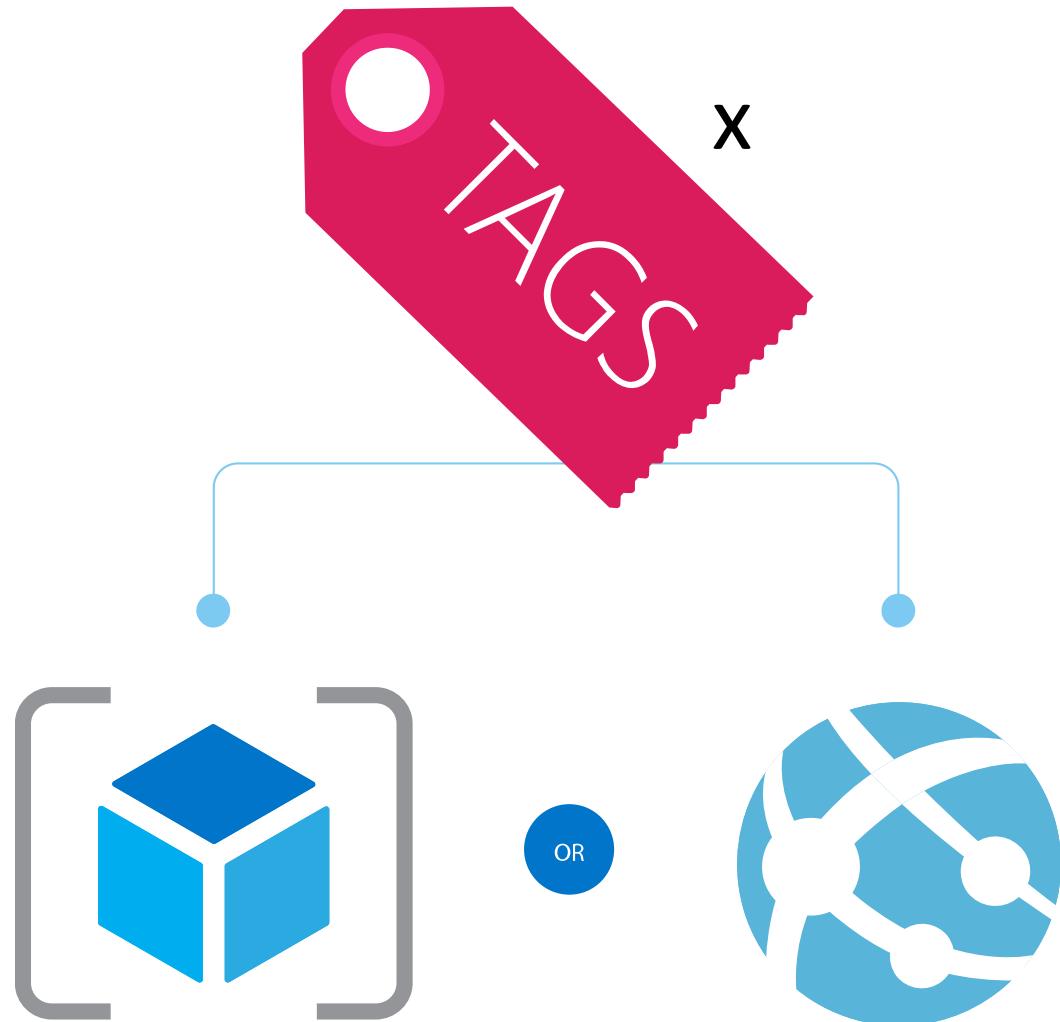
Определяется тем, имеют ли они общий жизненный цикл и общее управление



OR



Теги ресурсов



- Пара «имя-значение», присвоенная ресурсам или группам ресурсов
- Могут применяться в рамках подписки
- Каждый ресурс может иметь до 15 тегов
- Различные принципы тегирования, такие как: принадлежность к роли, отделу, окружению и пр.

❖ Демонстрация

Создание виртуальной машины на основе ARM с помощью портала

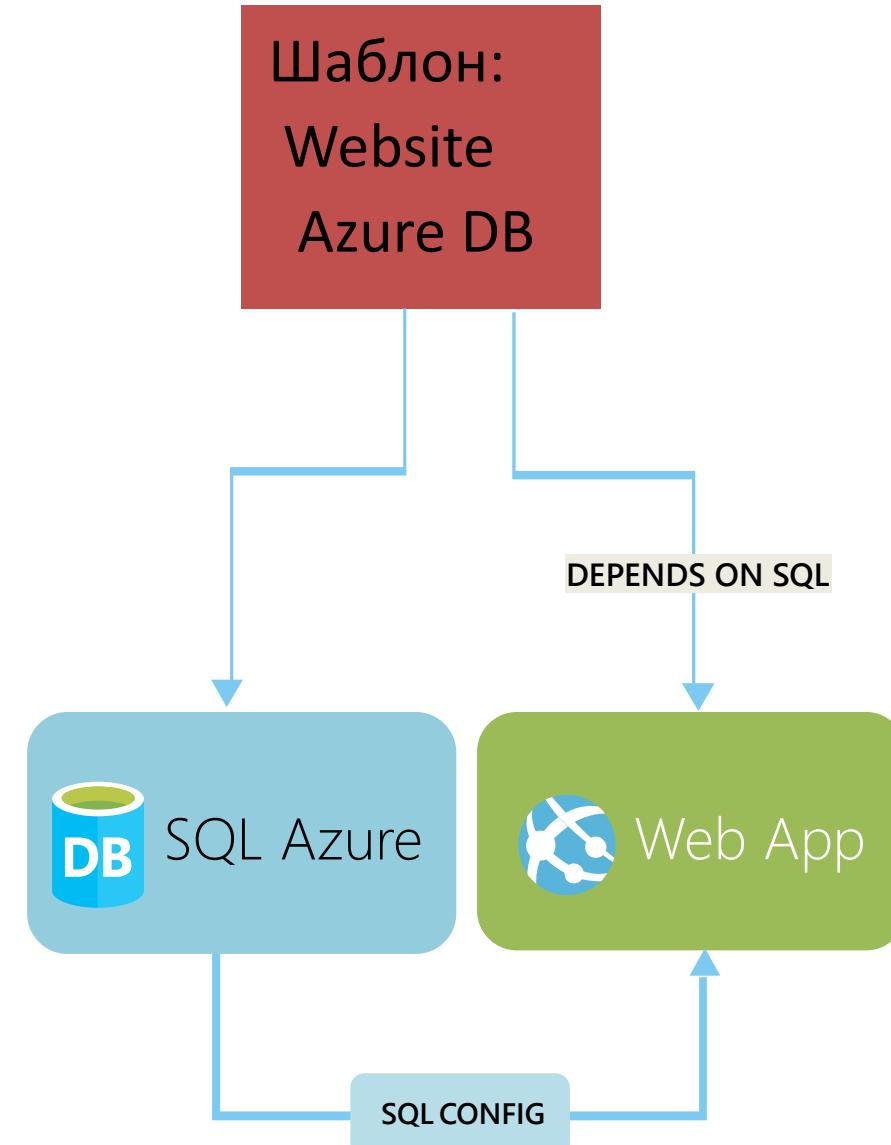
#msdevcon

Шаблоны Azure Resource Manager

#msdevcon

Шаблоны ресурсов

- Основанная на модели декларативная спецификация ресурсов, их конфигурации, кода, расширений
- Многократная применимость
- Согласованное развертывание
- Использование в системах контроля версий
- Параметризация ввода/вывода



Структура шаблона

```
{  
  "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",  
  "contentVersion": "",  
  "parameters": { },  
  "variables": { },  
  "resources": [ ],  
  "outputs": { }  
}
```

ELEMENT NAME	REQUIRED	DESCRIPTION
\$schema	Yes	Location of the JSON schema file that describes the version of the template language.
contentVersion	Yes	Version of the template (such as 1.0.0.0). When deploying resources using the template, this value can be used to make sure that the right template is being used.
parameters	No	Values that are provided when deployment is executed to customize resource deployment.
variables	No	Values that are used as JSON fragments in the template to simplify template language expressions.
resources	Yes	Types of services that are deployed or updated in a resource group.
outputs	No	Values that are returned after deployment.

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-authoring-templates>

Параметры

```
"parameters": {  
    "<parameterName>" : {  
        "type" : "<type-of-parameter-value>",  
        "defaultValue": "<optional-default-value-of-parameter>",  
        "allowedValues": [ "<optional-array-of-allowed-values>" ]  
    }  
}
```

ELEMENT NAME	REQUIRED	DESCRIPTION
parameterName	Yes	Name of the parameter. Must be a valid JavaScript identifier.
type	Yes	Type of the parameter value. See the list below of allowed types.
defaultValue	No	Default value for the parameter, if no value is provided for the parameter.
allowedValues	No	Array of allowed values for the parameter to make sure that the right value is provided.

The allowed types and values are:

- string or secureString - any valid JSON string
- int - any valid JSON integer
- bool - any valid JSON boolean
- object or secureObject - any valid JSON object
- array - any valid JSON array

Пример параметров

```
"parameters": {  
    "siteName": {  
        "type": "string"  
    },  
    "siteLocation": {  
        "type": "string"  
    },  
    "hostingPlanName": {  
        "type": "string"  
    },  
    "hostingPlanSku": {  
        "type": "string",  
        "allowedValues": [  
            "Free",  
            "Shared",  
            "Basic",  
            "Standard",  
            "Premium"  
        ],  
        "defaultValue": "Free"  
    }  
}
```

Переменные

```
"parameters": {
    "username": {
        "type": "string"
    },
    "password": {
        "type": "secureString"
    }
},
"variables": {
    "connectionString": "[concat('Name=', parameters('username'), ';Password=', parameters('password'))]"
}
```

```
"parameters": {
    "environmentName": {
        "type": "string",
        "allowedValues": [
            "test",
            "prod"
        ]
    }
},
"variables": {
    "environmentSettings": {
        "test": {
            "instancesSize": "Small",
            "instancesCount": 1
        },
        "prod": {
            "instancesSize": "Large",
            "instancesCount": 4
        }
    },
    "currentEnvironmentSettings": "[variables('environmentSettings')[parameters('environmentName')]]",
    "instancesSize": "[variables('currentEnvironmentSettings').instancesSize]",
    "instancesCount": "[variables('currentEnvironmentSettings').instancesCount]"
}
```

Ресурсы

```
"resources": [
  {
    "apiVersion": "<api-version-of-resource>",
    "type": "<resource-provider-namespace/resource-type-name>",
    "name": "<name-of-the-resource>",
    "location": "<location-of-resource>",
    "tags": "<name-value-pairs-for-resource-tagging>",
    "dependsOn": [
      "<array-of-related-resource-names>"
    ],
    "properties": "<settings-for-the-resource>",
    "resources": [
      "<array-of-dependent-resources>"
    ]
  }
]
```

ELEMENT NAME	REQUIRED	DESCRIPTION
apiVersion	Yes	Version of the API that supports the resource. For the available versions and schemas for resources, see Azure Resource Manager Schemas .
type	Yes	Type of the resource. This value is a combination of the namespace of the resource provider and the resource type that the resource provider supports.
name	Yes	Name of the resource. The name must follow URI component restrictions defined in RFC3986.
location	No	Supported geo-locations of the provided resource.
tags	No	Tags that are associated with the resource.
dependsOn	No	Resources that the resource being defined depends on. The dependencies between resources are evaluated and resources are deployed in their dependent order. When resources are not dependent on each other, they are attempted to be deployed in parallel. The value can be a comma separated list of a resource names or resource unique identifiers.
properties	No	Resource specific configuration settings.
resources	No	Child resources that depend on the resource being defined.

Пример ресурса

```
"resources": [
  {
    "apiVersion": "2014-06-01",
    "type": "Microsoft.Web/serverfarms",
    "name": "[parameters('hostingPlanName')]",
    "location": "[resourceGroup().location]",
    "properties": {
      "name": "[parameters('hostingPlanName')]",
      "sku": "[parameters('hostingPlanSku')]",
      "workerSize": "0",
      "numberOfWorkers": 1
    }
  },
  {
    "apiVersion": "2014-06-01",
    "type": "Microsoft.Web/sites",
    "name": "[parameters('siteName')]",
    "location": "[resourceGroup().location]",
    "tags": {
      "environment": "test",
      "team": "ARM"
    },
    "dependsOn": [
      "[resourceId('Microsoft.Web/serverfarms', parameters('hostingPlanName'))]"
    ],
    "properties": {
      "name": "[parameters('siteName')]",
      "serverFarm": "[parameters('hostingPlanName')]"
    },
    "resources": [
      {
        "apiVersion": "2014-06-01",
        "type": "Extensions",
        "name": "MSDeploy",
        "properties": {
          "packageUri": "https://auxmktpliceprod.blob.core.windows.net/packages/StarterSite-modified.zip",
          "dbType": "None",
          "connectionString": "",
          "setParameters": {
            "Application Path": "[parameters('siteName')]"
          }
        }
      }
    ]
  }
]
```

Выходные данные (Outputs)

```
"outputs": {  
    "<outputName>" : {  
        "type" : "<type-of-output-value>",  
        "value": "<output-value-expression>",  
    }  
}
```

ELEMENT NAME	REQUIRED	DESCRIPTION
outputName	Yes	Name of the output value. Must be a valid JavaScript identifier.
type	Yes	Type of the output value. Output values support the same types as template input parameters.
value	Yes	Template language expression which will be evaluated and returned as output value.

```
"outputs": {  
    "siteUri" : {  
        "type" : "string",  
        "value": "[concat('http://',reference(resourceId('Microsoft.Web/sites', parameters('siteName'))).hostNames[0])]"  
    }  
}
```

ФУНКЦИИ И ВЫРАЖЕНИЯ ЯЗЫКА ШАБЛОНА

`copyIndex(offset)` – возвращает текущий индекс итерации

`length(array)` – возвращает число элементов в массиве

`listKeys(storageAccountResourceId, apiVersion)` – возвращает ключи учетной записи хранения

`parameters('parameterName')` – возвращает значение параметра

`providers(namespace, resourceType)` – возвращает информацию о провайдере ресурса

`resourceGroup()` – возвращает структурированный объект, который представляет собой текущую группу ресурсов

`resourceId('namespace/resourceType', 'resourceName')` – возвращает уникальный идентификатор ресурса при ссылке на объект на пределами текущей группы ресурсов

`subscription()` – возвращает информацию о подписке

`variables('variables')` – возвращает значение переменной

Несколько экземпляров ресурсов

```
"parameters": {
  "count": {
    "type": "int",
    "defaultValue": 3
  }
},
"resources": [
  {
    "name": "[concat('examplecopy-', copyIndex())]",
    "type": "Microsoft.Web/sites",
    "location": "East US",
    "apiVersion": "2014-06-01",
    "copy": {
      "name": "websitescopy",
      "count": "[parameters('count')]"
    },
    "properties": {}
  }
]
```

- examplecopy-0
- examplecopy-1
- examplecopy-2.

Определение зависимостей

```
{  
  "name": "<name-of-the-resource>",  
  "type": "<resource-provider-namespace/resource-type-name>",  
  "apiVersion": "<supported-api-version-of-resource>",  
  "location": "<location-of-resource>",  
  "tags": { <name-value-pairs-for-resource-tagging> },  
  "dependsOn": [ <array-of-related-resource-names> ],  
  "properties": { <settings-for-the-resource> },  
  "resources": { <dependent-resources> }  
}
```

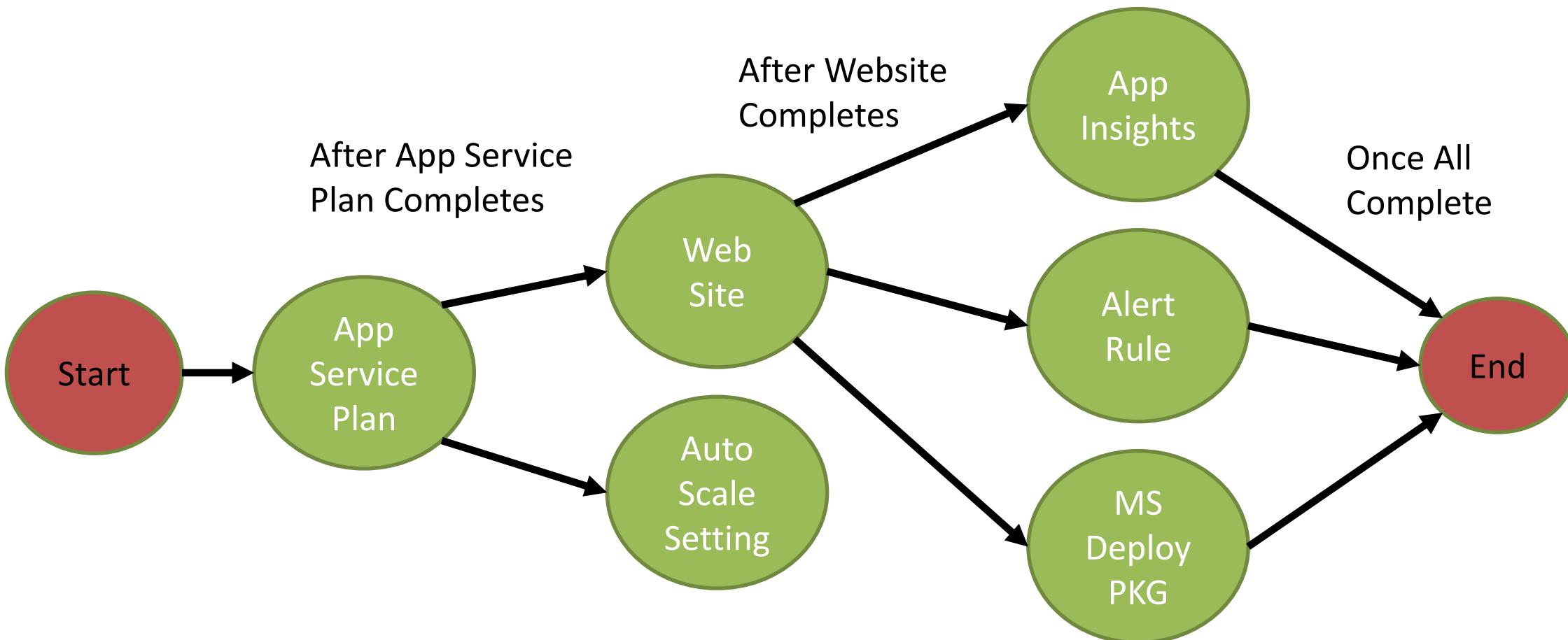
dependsOn: определяет зависимости от других ресурсов, которые должны быть доступны перед созданием данного ресурса. Влияет на скорость развертывания

resources: указывает на дочерние ресурсы, которые не влияют на процесс оптимизации развертывания. Можно определить до 5 уровней дочерних ресурсов в глубину, при этом функциональность dependsOn не обеспечивается

reference: функция определяет неявную зависимость и влияет на порядок развертывания ресурсов. Не используйте одновременно dependsOn и reference для определения зависимостей конкретного ресурса, достаточно какого-то одного варианта. Использование reference предпочтительнее, так как позволяет минимизировать количество последовательных шагов развертывания

Реализация шаблона

- Модуль выполнения строит машину состояния
- dependsOn() и reference() определяют зависимости



❖ Демонстрация

Создание виртуальной машины на основе ARM с помощью шаблона

#msdevcon

❖ Демонстрация

Разработка ARM-шаблонов

#msdevcon

Контроль доступа к ресурсам ARM

#msdevcon

Контроль доступа в Azure Resource Manager

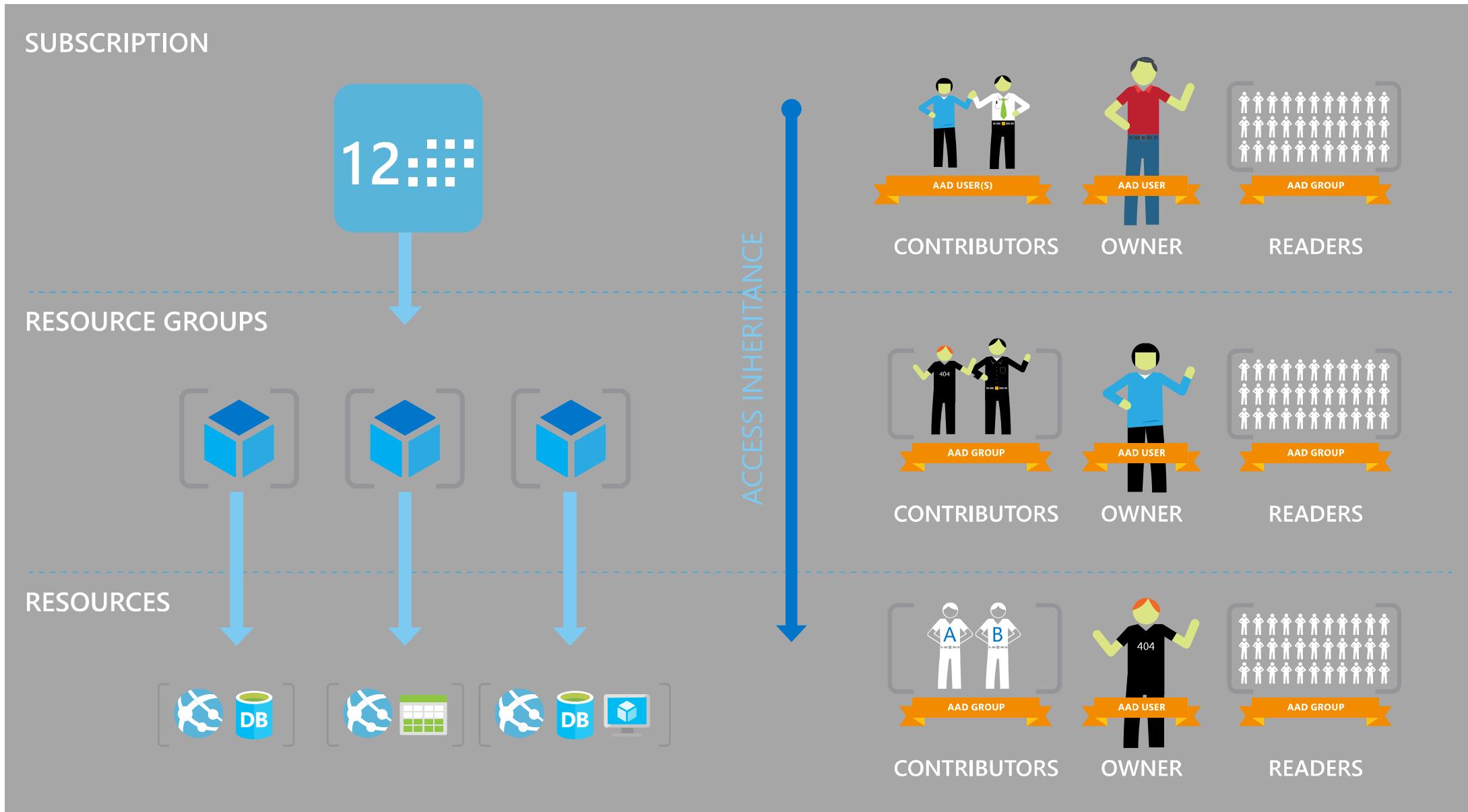
- Механизм ролей (Role Based Access Control, RBAC)
- Журналы аудита
- Блокировки ресурсов

Role Based Access Control

- Обеспечивает безопасный доступ с гранулярным разграничением полномочий
- Настраивается для пользователей, групп или сервисов
- Встроенные роли упрощают использование

Замечание – RBAC применяется только к ресурсам ARM. ПО внутри ВМ может иметь собственные механизмы защиты, которые следует учитывать при планировании

Role Based Access Control



Уровни гранулярности

/subscriptions/{id}/resourceGroups/{name}/providers.../sites/{site}

Уровень подписки –
предоставляет права на все
ресурсы подписки

Уровень ресурсной группы –
предоставляет права на все
ресурсы группы

Уровень ресурса –
предоставляет права на
конкретный ресурс



Журналы аудита

- Регистрируются все операции записи/удаления/выполнения
- Централизованное расположение
- Единый формат

Блокировки ресурсов

- Блокировки ресурсов позволяют предотвратить аварийные ситуации
- Блокировки ресурсов позволяют администратору создавать политики, которые запрещают операции записи или удаления

❖ Демонстрация

Разделение полномочий с помощью RBAC

#msdevcon

❖ Практика

Развертывание инфраструктуры на основе ARM-шаблонов

#msdevcon

Практические работы

Репозиторий с файлами

<https://github.com/ashapoms/DevCon201706>

Ваш двузначный номер участника XX

Сеть WiFi

SSID: **azurestack**

Password: **Pass@word2017!**

Архитектура и возможности Azure Stack

#msdevcon



Microsoft Azure

Глобальное • Надежное • Гибкое

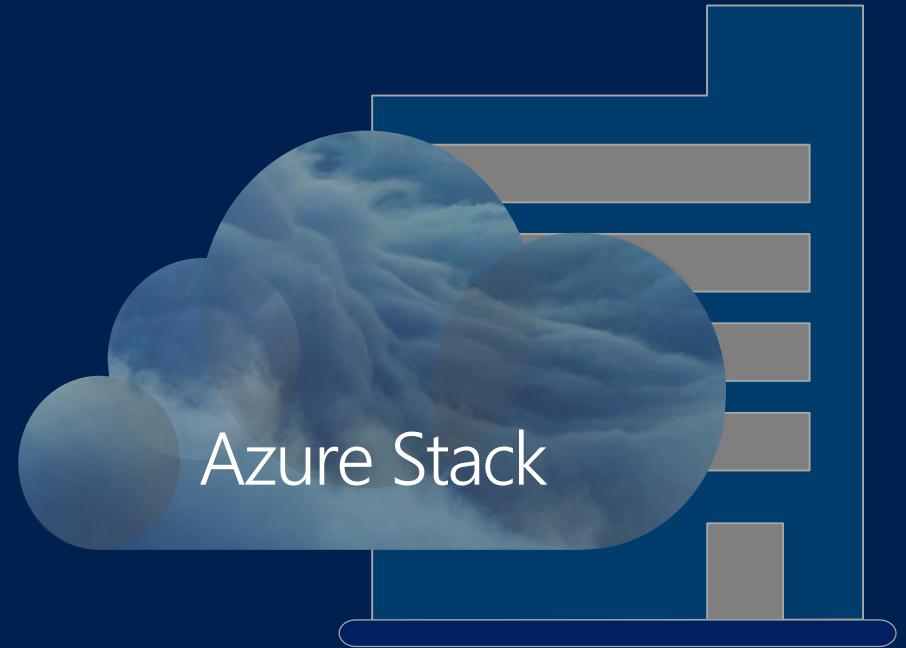


Azure Stack

Azure Stack — это расширение Azure, которое обеспечивает адаптивность и эффективность облачных вычислений в локальных центрах, а кроме того, позволяет создавать современные приложения в гибридных облачных средах с необходимыми уровнями гибкости, контроля и защиты.

Azure и Azure Stack

По-настоящему целостная гибридная облачная платформа



Создание гибридных приложений с Azure и Azure Stack

По-настоящему целостная гибридная облачная платформа



Microsoft Azure
Публичное облако

Microsoft Azure Stack
Локальная среда

* - После выхода общедоступной версии

Возможности Azure Stack



Единство разработки
приложений



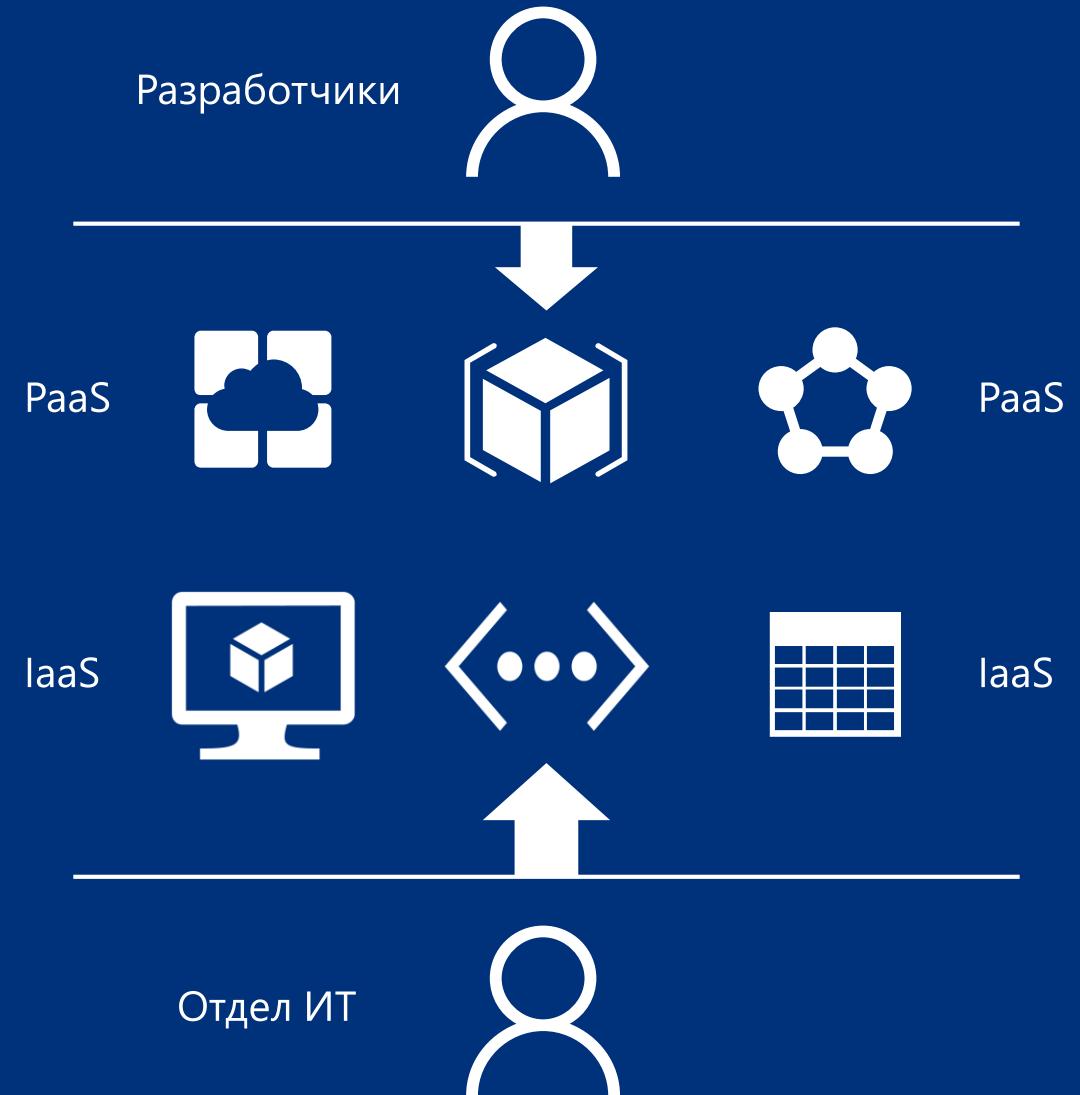
Локально доступные
службы Azure



Интегрированные
системы

Службы Azure доступны локально

Внедряйте гибридное облако с учетом ваших условий. Выполняйте запросы бизнеса и технические требования, выбирая нужное сочетание облачной и локальной моделей развертывания



Azure IaaS доступна локально: больше возможностей, чем у традиционной виртуализации

			
Виртуальные машины (ВМ) Масштабируемые наборы ВМ	Контейнеры Docker	Сеть	Хранилище
Быстрое развертывание, автоматическое масштабирование	Контейнеры Linux и Windows Server	Виртуальная сеть, VPN-шлюз, балансировщик нагрузки	Блобы, таблицы, очереди

Все перечисленные службы планируется представить в общедоступной версии.
По мере обновления Azure Stack будут предлагаться дополнительные службы Azure

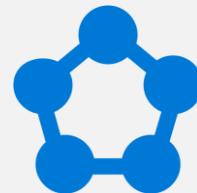
Azure PaaS доступна локально: управляемые платформы для высокопродуктивной разработки



Azure App Service



Azure Functions



Azure Service Fabric



Azure Container Service



Cloud Foundry

Веб-, мобильные,
API-приложения

Бессерверные
вычисления

Непрерывные
масштабируемые
распределенные
приложения

Надежное
управление
контейнерами

Платформа с
открытым кодом

Azure Service Fabric и Azure Container Service появятся после выхода общедоступной версии. Остальные службы будут доступны с выходом общедоступной версии. По мере обновления Azure Stack будут предлагаться дополнительные службы Azure

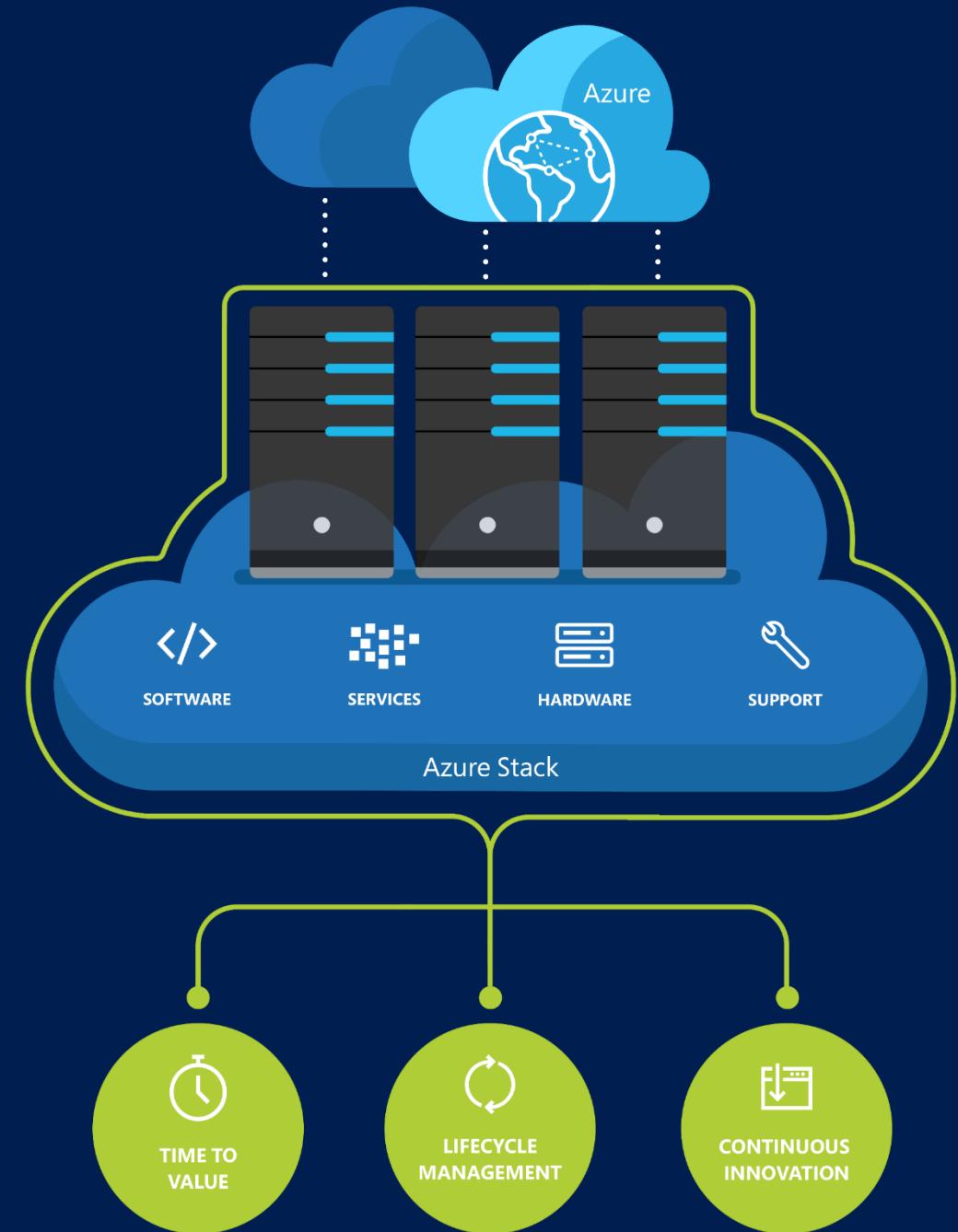
❖ Демонстрация

Настройка подписок, развертывание сервисов

#msdevcon

Системы для высочайшей продуктивности

Оптимизируйте бизнес-приложения и службы с помощью интегрированных систем, которые обеспечивают непрерывный доступ к новшествам Azure



Интегрированные системы Azure Stack

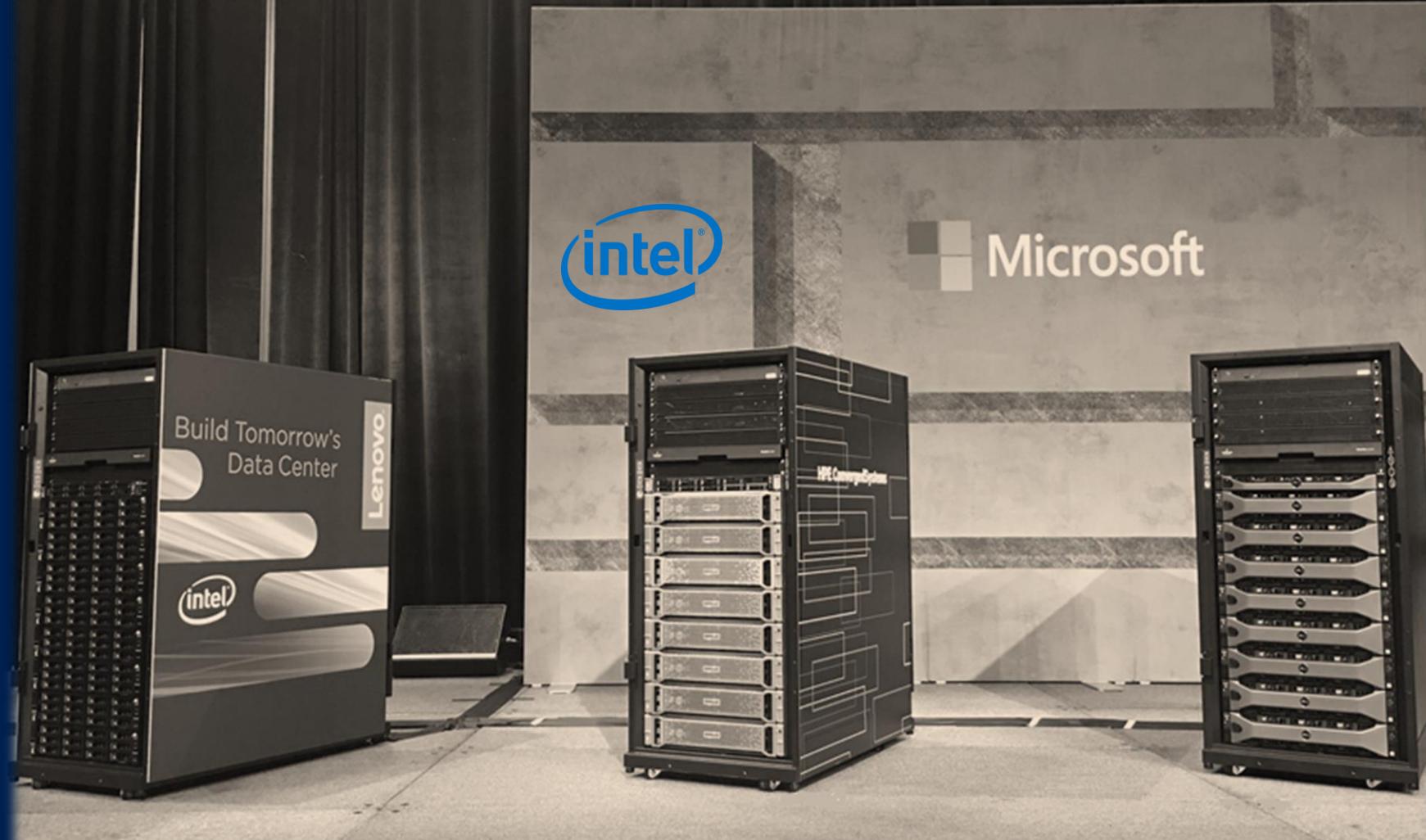
DELL EMC



**Hewlett Packard
Enterprise**

Lenovo


CISCO™



Мы будем привлекать новых партнеров и добавлять аппаратные конфигурации

Жизненный цикл интегрированных систем Azure Stack



Непрерывная
синхронизация
обновлений Azure
и Azure Stack

Интегрированные услуги поддержки

- Неизменное качество услуг поддержки независимо от выбора поставщика
- Скоординированные процессы эскалации и разрешения проблем
- Поддержка облачных служб специалистами Microsoft
- Поддержка систем партнерами – поставщиками оборудования



Принципы масштабирования Azure Stack

1 Одна «точка» управления

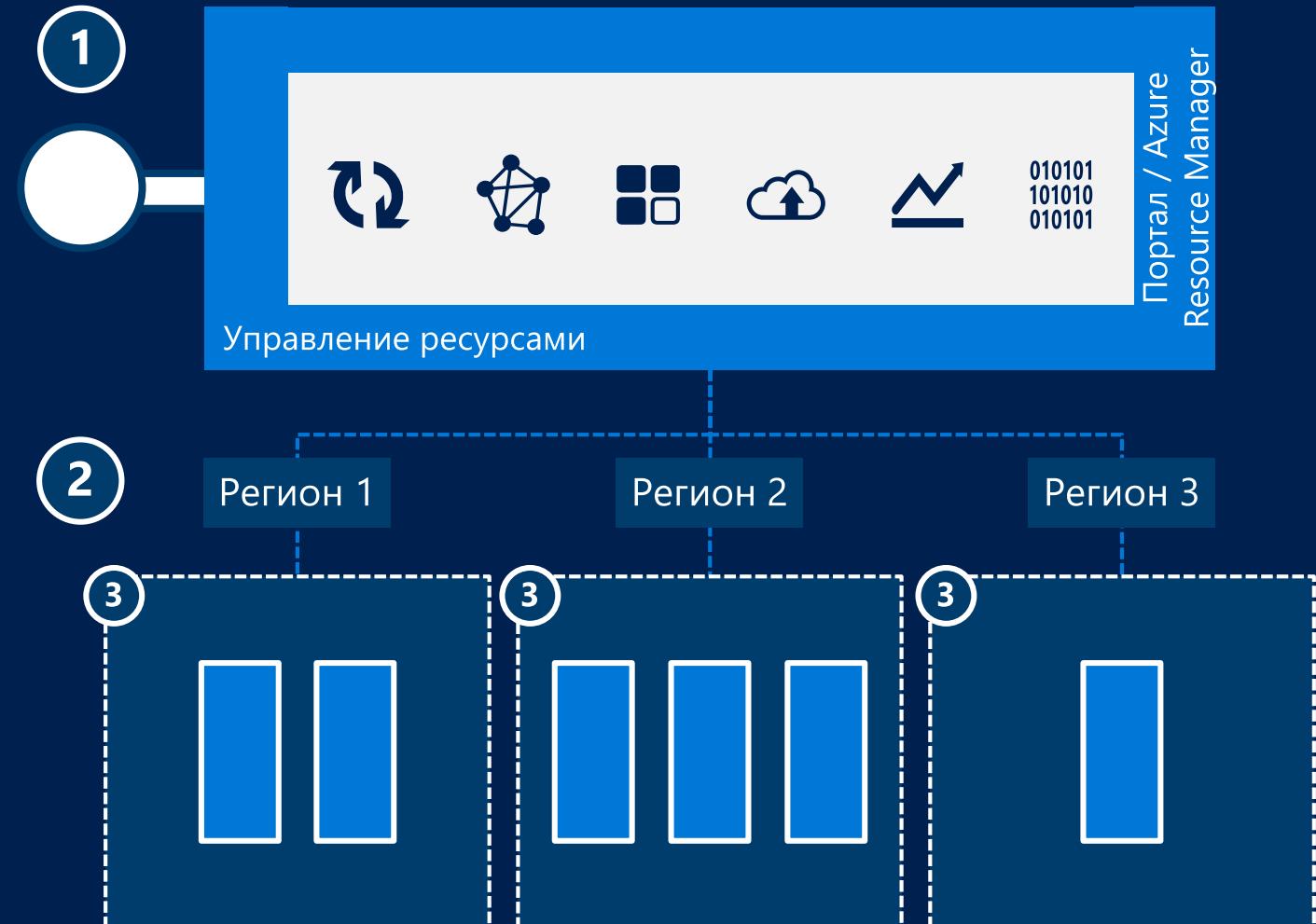
2 Несколько регионов

3 Несколько единиц масштабирования
(Scale Units, SU) в регионе

- В различных SU может использоваться оборудование разного поколения
- SU = домен отказа = высокодоступный кластер

- Минимальная SU = 4 сервера
- Масштаб = число серверов в SU * число SU в регионе * число регионов
- Масштаб общедоступной версии: 1 регион с 1 SU с числом серверов от 4 до 12
- Можно добавлять по одному серверу

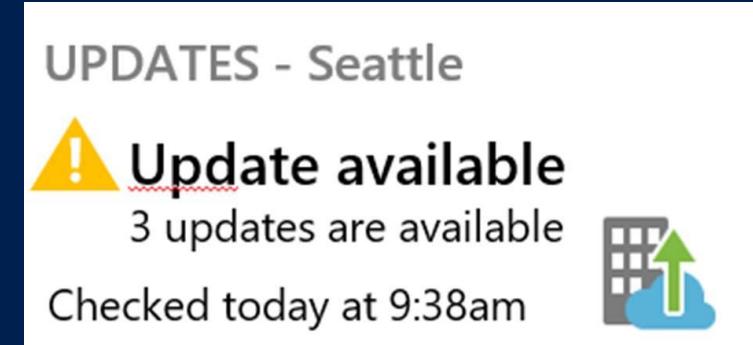
Пример: крупная компания или сервис-провайдер



После выхода общедоступной версии: 1) поддержка нескольких регионов для расширения ёмкости и обеспечения географической избыточности; 2) добавление единиц масштабирования в регионе; 3) улучшенное управление инфраструктурой

Azure Stack: применение обновлений

- Заранее проверенные обновления ПО (функциональные и средства безопасности), микропрограмм и драйверов от Microsoft и партнеров
- Регулярные обновления по определенному графику; решение о времени их установки принимает клиент; максимальный срок ожидания задается политикой
- Автоматическое обновление в рамках инфраструктуры для минимизации рисков нарушения работы клиентских приложений

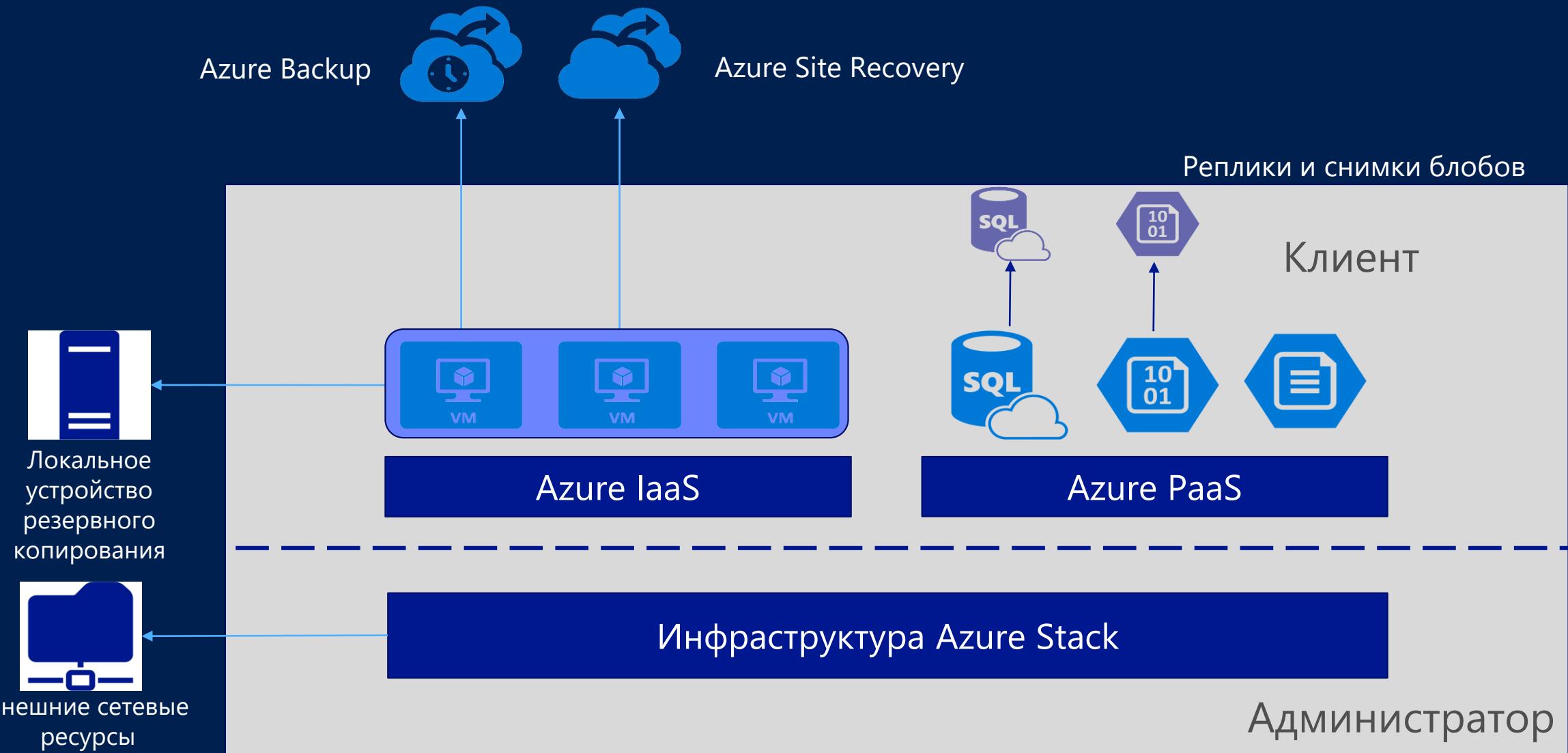
A screenshot of the 'Update Details' window. At the top, it says 'Update Details' and 'Update rollup for Jan 2016'. Below that are three buttons: 'Update Now', 'View Log', and 'View KB Article'. The main area displays the following details:

VERSION	1601
DATE AVAILABLE	2016-01-02
DATE STARTED	2016-01-11
STARTED BY	MAS_ADMIN
DURATION (HH:MM)	01:03 (in progress)
PACKAGE SIZE	5.2 MB

DETAILS For more info see <HTTP://support.MAS.com/KB/3014412>

NAME	PROGRESS	STATUS	DURATION
NC	58% completed	In progress	00:43
SLB	12% completed	In progress	00:12
Gateway	5% completed	In progress	00:08
Console	Not started	-	-
WOSS	Not started	-	-
WSUS	Not started	-	-

Резервное копирование и аварийное восстановление



После выхода общедоступной версии: улучшенные средства обеспечения устойчивости и доступности, в том числе надежные механизмы резервного копирования и восстановления

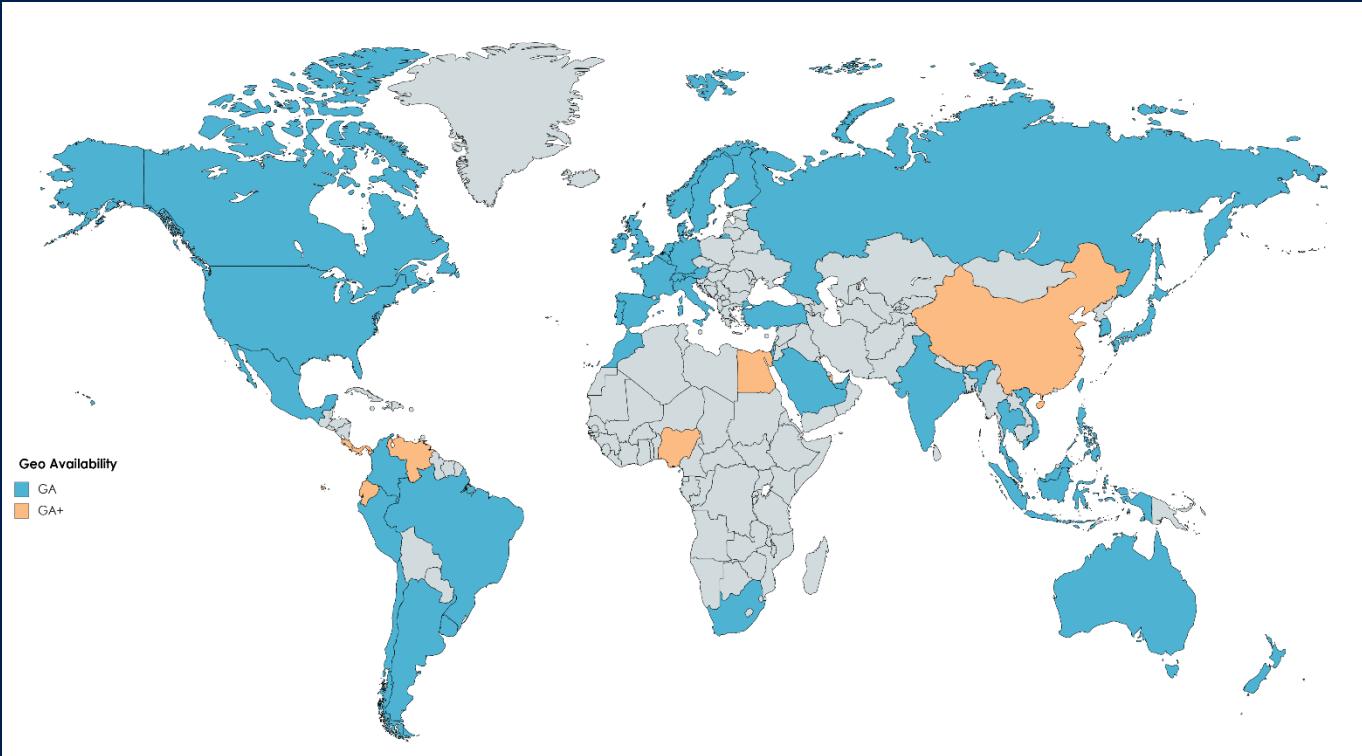
❖ Демонстрация

Мониторинг, реализация Azure Stack TP3 Refresh

#msdevcon

Географическая доступность Azure Stack

С момента запуска (середина 2017 года) Azure Stack будет доступен в 46 странах



Южная и Северная Америка: Аргентина, Бразилия, Канада, Чили, Колумбия, Мексика, Перу, США, Уругвай

Европа, Ближний Восток и Африка: Австрия, Бельгия, Дания, Финляндия, Франция, Германия, Ирландия, Израиль, Италия, Люксембург, Монако, Голландия, Норвегия, Португалия, Россия, Саудовская Аравия, ЮАР, Испания, Швеция, Швейцария, ОАЭ, Великобритания

Азиатско-Тихоокеанский регион: Австралия, Гонконг, Индия, Индонезия, Япония, Корея, Малайзия, Новая Зеландия, Филиппины, Сингапур, Тайвань, Таиланд, Вьетнам

Примечание: доступность оборудования различных производителей варьируется в зависимости от страны

Доступность в других регионах будет обеспечиваться в соответствии с клиентским спросом и особенностями логистики

Azure Stack Technical Preview 3

- Azure Stack TP3 POC (Proof of Concept) разворачивается на одном физическом сервере
 - 12 физических ядер
 - 96 ГБ оперативной памяти
 - 1 диск для ОС, минимум 200 ГБ свободного пространства
 - 4 неразмеченных диска (HDD или SSD) объемом не менее 140 ГБ
 - 1 сетевой адаптер (остальные необходимо отключить)
 - Поддержка процессором виртуализации и SLAT
- Azure AD или Active Directory Domain Services
 - При использовании Azure AD необходимо подключение к Интернет, напрямую или через transparent proxy
- Инсталляционный пакет содержит vhdx-файл, с которого настраивается загрузка сервера
- Провайдеры MySql, SQL и App Service устанавливаются отдельно и требуют дополнительных ресурсов

❖ Практика

Управление ресурсами Azure Stack для запуска приложений

#msdevcon

Гибридные приложения в Microsoft Azure и Azure Stack

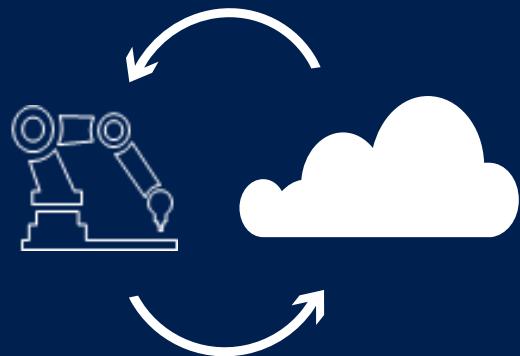
#msdevcon

Гибридные приложения

- Идентичные приложения, развернутые в разных облаках
- Одно приложение, компоненты которого развернуты в разных облаках
- Azure Stack может выступать в качестве облака для этих вариантов

Варианты использования

Обработка и аналитика
на периферии



Облачная модель
приложения, доступная
локально



Приложение
развертывается в любой
среде без изменений





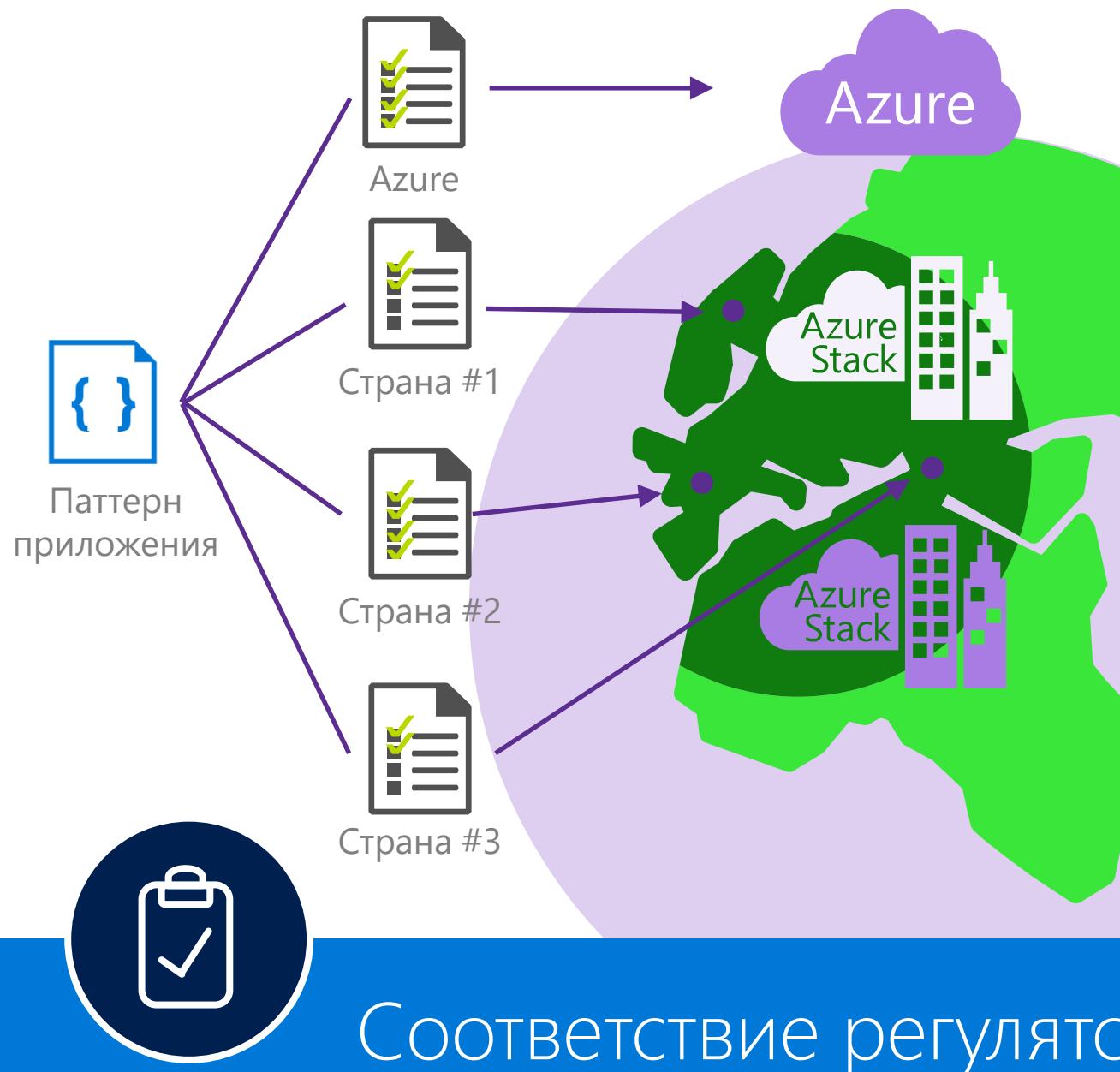
Обработка и аналитика на периферии

Используйте Azure Stack для:

- Минимизации задержек (real-time)
- Систем с ограниченной связью
- Локальной обработки данных

Используйте Azure для аналистики агрегированных данных и построения моделей

Единая логика приложения

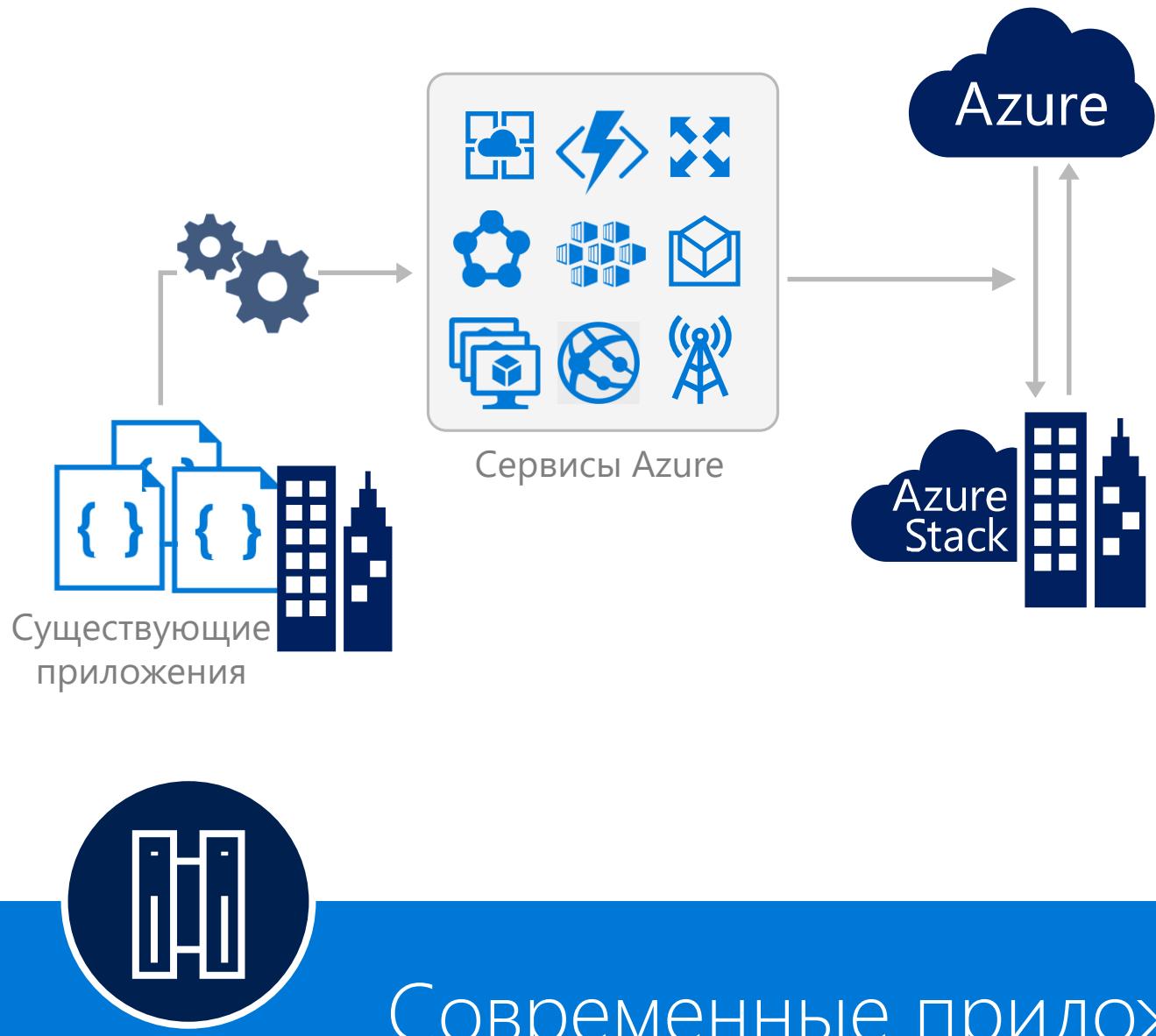


Разрабатывайте и внедряйте
глобальные приложения в Azure

Используйте Azure Stack
для удовлетворения требований
регуляторов:

- Правительство
- Индустрия
- Регион

Целостная модель приложения для
всех сценариев



Применяйте современные архитектуры в приложениях, не готовых к облаку

- PaaS
- Безсерверные вычисления
- Микросервисы и контейнеры

Мигрируйте в Azure без изменений кода

Единая программная модель и процессы, использование накопленного опыта

Современные приложение в любой среде

Единая экосистема Azure

Пользуйтесь нужными вам
инструментами и технологиями
в Azure и Azure Stack

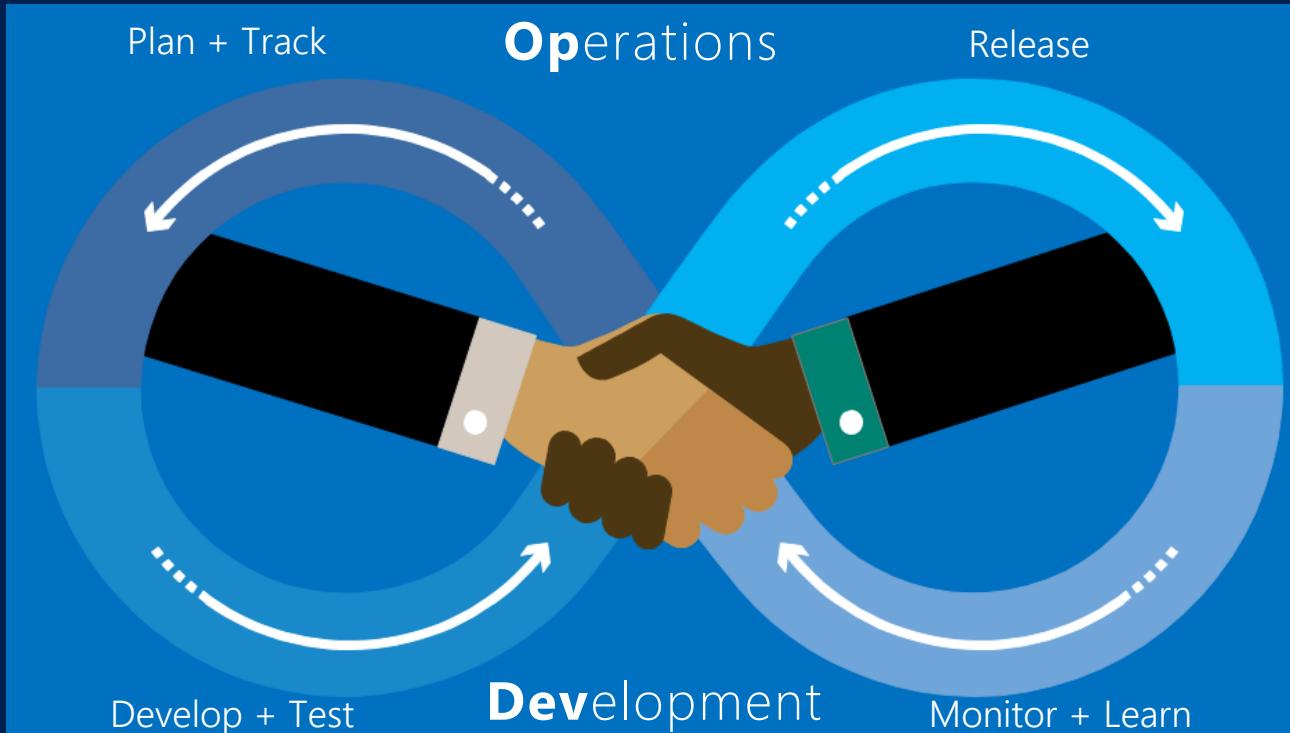


CLOUD FOUNDRY



Большинство решений Azure Marketplace без дополнительных модификаций работают в Azure Stack

Единый жизненный цикл DevOps в Azure и Azure Stack



Люди | Процессы | Инструменты

Однаковые процессы развертывания с помощью Visual Studio

Интеграция процессов CI/CD с инструментами на основе открытого кода (например, Jenkins) и Visual Studio Team System* (VSTS)

Единый инструментарий автоматизации рабочих процессов с поддержкой Chef и DSC-расширений Azure PowerShell

* - Интеграция с VSTS появится после выхода общедоступной версии

Рекомендации при разработке гибридных приложений

#msdevcon

Чего следует избегать

- Прописывания жестких (hardcoded) конечных точек
- Предположений, что контент существует
- Использования недоступных возможностей

Hardcoded Endpoints

```
},
"diagnosticsProfile": {
  "bootDiagnostics": {
    "enabled": "true",
    "storageUri": "[concat('http://',variables('storageAccountName'),'.blob.core.windows.net')]"
  }
}
}

},
"diagnosticsProfile": {
  "bootDiagnostics": {
    "enabled": "true",
    "storageUri": "[reference(resourceId('Microsoft.Storage/storageaccounts', variables('storageAccountName'))).primaryEndpoints.blob]"
  }
}
}
```

Версии API

Resource Provider	apiVersion
Compute	'2015-06-15'
Network	'2015-06-15' , '2015-05-01-preview'
Storage	'2016-01-01' , '2015-06-15' , '2015-05-01-preview'
KeyVault	'2015-06-01'
App Service	'2015-08-01'
MySQL	'2015-09-01'
SQL	'2014-04-01-preview'

<https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-develop-templates>

Контент

- Добавьте необходимые образы в Azure Stack
- Настройте интеграцию с Azure Marketplace
 - Удобно
 - Доступно для многих образов
 - Количество совместимых с Azure Stack решений будет расти

ARM-шаблоны

- Используйте динамические конечные точки
- Параметризуйте расположение (location)
- Обеспчивайте гибкость в выборе размеров ВМ и их возможностей

Профили API (API Profiles)

- Позволяют контролировать разницу в версиях API между Azure и Azure Stack
- Представляют собой наборы модулей AzureRM PowerShell с определенной версией API
- Реализуются с помощью AzureRM.Bootstrapper, который доступен в галерее PowerShell Gallery

Профили API (API Profiles)

- # Install the AzureRM.Bootstrapper module.
- Install-Module -Name AzureRm.BootStrapper
- # Installs and imports the API Version Profile required by Azure Stack into the current PowerShell session.
- Use-AzureRmProfile -Profile 2017-03-09-profile
- # Install the Azure Stack-specific PowerShell modules such as AzureStackAdmin, and AzureStackStorage.
- Install-Module -Name AzureStack -RequiredVersion 1.2.9

❖ Демонстрация

Интеграция с Azure Marketplace, использование ARM-шаблонов

#msdevcon

❖ Практика

Развертывание гибридного приложения

#msdevcon



Дополнительные материалы

Документация по ARM

<https://azure.microsoft.com/en-us/documentation/articles/resource-group-overview/>

Структура ARM-шаблонов

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-authoring-templates>

Azure Stack Technical Preview

<https://aka.ms/azurestack/>

Шаблоны для Azure Stack

<https://aka.ms/azurestackgithub>

#msdevcon



Используйте ARM

Начните использовать ARM
для реализации
инфраструктуры в своих
проектах

Попробуйте Azure Stack

Разверните Azure Stack TP3 с
дополнительными
провайдерами SQL, MySQL,
PaaS

Настройте интеграцию с
Azure Marketplace

Рассмотрите гибридные сценарии

Оцените применимость
гибридных сценариев к
вашим решениям
Используйте паттерны
гибридных приложений



Q&A

Гибридные приложения и практики DevOps на базе Azure Stack

Александр Шаповал, ashapo@microsoft.com

Андрей Выставкин, andvis@microsoft.com

Сергей Груздов, egel@egel.su

#msdevcon



Помогите нам стать лучше!

На вашу почту отправлена индивидуальная ссылка на электронную анкету. 3 июня в 23:30 незаполненная анкета превратится в тыкву.

Заполните анкету и подходите к стойке регистрации за приятным сюрпризом!

#msdevcon

Оставляйте отзывы в социальных сетях. Мы все читаем. Спасибо вам! 😊



© 2016 Microsoft Corporation. All rights reserved.