

## Лекция. Общий обзор и тенденции развития облачных технологий

### 1. Google (<https://developers.google.com>).

**Google Drive** – облачное хранилище данных, принадлежащее компании *Google*, позволяющее пользователям хранить свои данные на серверах в облаке и делиться ими с другими пользователями в Интернете. *Google Drive* отличается лаконичным интерфейсом и предлагает установить удобные программные клиенты для смартфонов и планшетов на базе операционной системы *Android*, ПК и ноутбуков под управлением операционной системы *Windows* или *MacOS*, мобильных устройств *iPhone* и *iPad*. В будущем ожидается более тесная интеграция хранилища с операционной системой *Chrome OS* и поддержка *Linux*. Каждый пользователь *Google Drive* получает до 15 Гбайт свободного пространства на все сервисы *Google* (в том числе *Gmail* и *Photos*). При этом он сам может решить, сколько места выделить под почту и какой объем оставить под важные файлы. Работать с файлами в *Google Drive* можно прямо в браузере. *Google Drive* можно превратить в отдельную папку в документах смартфона, планшета или ПК, и ее содержимое будет синхронизироваться автоматически.

**Google Docs** – бесплатный онлайн-офис, включающий в себя текстовый, табличный процессоры и сервис для создания презентаций, а также интернет-сервис облачного хранения файлов с функциями файлообмена. Позволяет создавать и редактировать стандартные документы, таблицы и презентации, а также поддерживает функции совместной работы над ними.

**Google App Engine** – сервис хостинга сайтов и *web*-приложений на серверах *Google*. Бесплатно предоставляется до 1 Гб дискового пространства, 10 Гб входящего трафика в день, 10 Гб исходящего трафика в день, 200 миллионов гигабайт CPU в день и 2 000 операций отправления электронной почты в день. Приложения, разворачиваемые на базе *App Engine*, должны быть написаны на *Python*, *Java* либо *Go*. Предлагается набор *API* для сервисов хранилища *datastore API (BigTable)* аккаунтов *Google*, набор *API* для загрузки данных по URL, электронной почты и т.д.

Платформа *Google* конкурирует с аналогичными сервисами от *Amazon*, которые предоставляют возможность размещать файлы и *web* -приложения, используя свою инфраструктуру. В отличие от многих обычных размещений приложений на виртуальных машинах, таких как *Amazon EC2*, платформа *App Engine* тесно интегрирована с приложениями и накладывает на разработчиков некоторые ограничения.

**Google Cloud Storage** – сервис хостинга файлов. Все файлы, которые записываются или перезаписываются на серверы, автоматически шифруются по алгоритму AES-128. Является конкурентом продукта *Amazon S3*.

### 2. Amazon (<http://aws.amazon.com>).

**Amazon Simple Storage Service (Amazon S3)** – онлайн-сервис *web* -службы, предлагаемая *Amazon Web Services*, предоставляющая возможность для хранения и получения любого объема данных, в любое время из любой точки сети, так называемый файловый хостинг. В марте 2012 года компания *Nasuni* провела опыт, в течение которого поочередно передавала массивный объем данных (12 Тб) из одного облачного сервиса в другой. В эксперименте участвовали наиболее рей-

тинговые облака: *Amazon S3*, *Windows Azure* и *Rackspace*. К удивлению исследователей, скорость передачи данных сильно отличалась в зависимости от того, какое облако принимало данные. Самый лучший показатель скорости записи данных оказался у *Amazon S3*, передача данных из двух других сервисов занимала всего 4–5 часов, в то время как передача данных в *Rackspace* заняла чуть меньше недели, а в *Windows Azure* – 40 часов.

**Amazon Elastic Compute Cloud** (*Amazon EC2*) – *web*-сервис, предоставляющий вычислительные мощности в облаке. Он дает пользователям полный контроль над вычислительными ресурсами, а также доступную среду для работы. *Amazon EC2* позволяет пользователям создать *Amazon Machine Image* (AMI), который будет содержать их приложения, библиотеки, данные и связанные с ними конфигурационные параметры, или использовать заранее настроенные шаблоны образов для работы *Amazon S3*. *Amazon EC2* предоставляет инструменты для хранения AMI. *Amazon S3* предоставляет безопасное, надежное и быстрое хранилище для хранения образов.

### 3. **Microsoft** (<http://www.windowsazure.com>).

**Microsoft SkyDrive** – интернет-сервис хранения файлов с функциями файлообмена, созданный и управляемый компанией *Microsoft*. Сервис *SkyDrive* позволяет хранить до 7 ГБ информации (или 25 ГБ для пользователей, имеющих право на бесплатное обновление) в виде стандартных папок. Пользователи могут просматривать, загружать, создавать, редактировать и обмениваться документами *Microsoft Office* (*Word*, *Excel*, *PowerPoint* и *OneNote*) непосредственно в *web*-браузере. Присутствует удаленный доступ к компьютеру, работающему под управлением *Windows*.

**Windows Azure** – платформа облачных сервисов, разработанная *Microsoft*. Платформа предоставляет возможность разработки и выполнения приложений и хранения данных на серверах, расположенных в распределенных центрах данных.

**Windows Azure Compute** – компонент, реализующий вычисления на платформе *Windows Azure*, предоставляет среду выполнения на основе ролевой модели.

**Windows Azure Storage** – компонент хранилища, предоставляющий масштабируемое хранилище. Не имеет возможности использовать реляционную модель и является альтернативой (либо дополняющим решением) *SQL Databases* (*SQL Azure*) – масштабируемой «облачной» версией *SQL Server*.

**Windows Azure Fabric** – по своему назначению является контролером и ядром платформы, выполняя функции мониторинга в реальном времени, обеспечения отказоустойчивости, выделения мощностей, развертывания серверов, виртуальных машин и приложений, балансировки нагрузки и управления оборудованием.

Платформа *Windows Azure* имеет API, построенное на REST, HTTP и XML, что позволяет разработчикам использовать облачные сервисы с любой операционной системой, устройствами и платформами.

Рассмотрим несколько типовых задач и сравним их возможные решения на каждой из платформ – **Amazon EC2**, **Google App Engine**, **Windows Azure**.

Задача 1. *Локально созданное приложение, отправляемое в облако на обработку и не меняющее своего содержания в процессе исполнения*

*Описание:* Это могут быть самые разнообразные прикладные задачи, построенные на базе предыдущих технологий – *Java* или *.NET*.

**Amazon:** Следует рассматривать локальный компьютер как еще один сервер, входящий в состав общего ЦОД, и внести необходимые изменения в конфигурацию.

**Google:** Для разработок на *Java* требуется значительная переработка существующих прикладных программ и логики работы с данными.

**Microsoft:** Если нужно запустить прикладную программу, написанную для *ASP.NET*, то потребуется переработать только то, что касается использования данных. В остальных случаях придется приложить достаточно много усилий для полной переработки всей программы.

*Задача 2. Масштабируемое Web-приложение.*

*Описание:* *Web-приложение*, использующее фоновое обновление данных и средства балансировки нагрузки.

**Amazon:** Следует рассматривать локальный компьютер как сервер, входящий в состав общего ЦОД, и внести необходимые изменения в конфигурацию. Требования по масштабированию и гибкой настройке применяемых вычислительных ресурсов выполняются вручную путем конфигурирования.

**Google:** Для построения эффективно работающих приложений следует использовать функции динамического масштабирования *AppEngine* и скриптовые программы.

**Microsoft:** Построение масштабируемых *Web-приложений* с использованием типовых приемов технологии *.NET*.

*Задача 3. Масштабирование для наращивания или уменьшения мощности управляется исключительно настройкой параметров конфигурации. Вычислительная задача с использованием методов параллельной обработки.*

*Описание:* Прикладная задача, выполняемая в автоматическом режиме без применения интерфейса с пользователем.

**Amazon:** Необходимо сконфигурировать несколько *ВМ* в зависимости от требуемого уровня масштабирования. Затем следует управлять средой исполнения прикладной задачи.

**Google:** На этой платформе имеется минимальный встроенный набор функций для выполнения прикладных задач, ориентированных на интенсивные вычисления.

**Microsoft:** Используя роли *Worker* и специальные типы представления данных, такие как *Queues* (потoki данных) и *BLOBs* (наборы крупноразмерных массивов двоичных данных), очень просто создать приложение, предназначенное для выполнения интенсивных вычислений. Управление и контроль за масштабируемостью и гибкой настройкой потребляемых ресурсов также не вызывает сложностей.

*Задача 4. Запускаемое в облаке приложение, которое обменивается данными с клиентским ПК.*

*Описание:* Приложения, работающие в облаке, взаимодействуют с запущенными на стороне клиента прикладными службами, которые управляют транзакциями данных между ПК и облаком.

**Amazon:** Прикладные программы, запускаемые в серверном облаке *Amazon Elastic Compute Cloud (Amazon EC2)*, можно легко сконфигурировать, чтобы обеспечить возможность их взаимодействия с приложениями, запускаемыми локально.

**Google:** Этот сценарий не имеет специальной поддержки. Можно использовать промежуточное хранилище для обмена данными.

**Microsoft:** Используя функционал компонентов платформы *Service Bus to Sync*, можно осуществлять необходимое взаимодействие между двумя средами исполнения приложений.

Задача 5. *Запускаемое в облаке приложение, которое взаимодействует с локальной программой.*

*Описание:* Запуск локально исполняемых прикладных задач.

**Amazon:** Прикладные программы, запускаемые в серверном облаке *Amazon EC2*, могут быть легко сконфигурированы для взаимодействия с приложениями, работающими локально на стороне клиента.

**Google:** Этот сценарий не имеет специальной поддержки на этой платформе. Можно использовать промежуточное хранилище, через которое осуществлять обмен данными.

**Microsoft:** Используя функционал компонентов платформы *Service Bus to Sync*, можно осуществлять необходимое взаимодействие между двумя средами исполнения приложений.