ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет прикладной математики и механики

Кафедра вычислительной математики и механики

Реферат

Тема: “Программирование с использованием облачных платформ

Google Apps/Amazon/Azure”

Группа: ИТСИ-17-1м

Студент: Ермакова О.А.

Преподаватель: Истомин Д.А.

Пермь 2018

Содержание

[Docker 3](#_Toc533607718)

[Kubernetes 3](#_Toc533607719)

[Виртуализация 5](#_Toc533607720)

[AWS Compute 6](#_Toc533607721)

[AWS Storage & Database 7](#_Toc533607722)

[AWS Database 8](#_Toc533607723)

[AWS Machine learning 9](#_Toc533607724)

[AWS Internet of things 10](#_Toc533607725)

[Google Cloud Platform: Identity & Security 11](#_Toc533607726)

[Google Cloud Platform: Maps platform 12](#_Toc533607727)

[Microsoft Azure: Media 12](#_Toc533607728)

[Microsoft Azure: Mobile 13](#_Toc533607729)

[Microsoft Azure: Analytics 13](#_Toc533607730)

[Google Cloud Platform: Networking 14](#_Toc533607731)

# Docker

Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в среде виртуализации на уровне операционной системы. Позволяет “упаковать” приложение со всем его окружением и зависимостями в контейнер, который может быть перенесён на любую Linux-систему с поддержкой cgroups в ядре, а также предоставляет среду по управлению контейнерами. Изначально использовал возможности LXC, с 2015 года применял собственную библиотеку, абстрагирующую виртуализационные возможности ядра Linux — libcontainer[2]. Разрабатывается и поддерживается одноимённой компанией-стартапом, распространяется в двух редакциях — общественной (Community Edition) по лицензии Apache 2.0 и для организаций (Enterprise Edition) по проприетарной лицензии. Написан на языке Go.

Компоненты

ПО:

– Демон Docker ( dockerd ) — процесс, управляющий контейнерами и дру-

гими объектами Docker. Он принимает запросы, посылаемые посредством

Docker Engine API.

– Клиент Docker ( docker ) предоставляет интерфейс командной строки для

управления демонами Docker.

∙

Объекты: Объекты Docker — различные сущности, используемые для сборки

приложения в Docker, например, образы, контейнеры и сервисы.

– Контейнер — стандартизованное изолированное окружение для запуска

приложений.

– Образ — неизменяемый шаблон, используемый для сборки приложений.

– Служба позволяет масштабировать приложение путем распределения кон-

тейнеров между несколькими демонами Docker. Результат этого — рой

(swarm), множество работающих вместе демонов Docker, общающихся меж-

ду собой посредством Docker API.

– Реестры — репозитории образов. Клиенты Docker используют реестры

для получения или сохранения образов. Реестры могут быть публичными

и частными. Главные публичные реестры — Docker Hub и Docker Cloud.

# Kubernetes

Kubernetes, k8s — открытое программное обеспечение для автоматизации раз-

вёртывания, масштабирования и управления контейнеризированными приложения-

ми.

Оригинальная версия была разработана компанией Google для внутренних нужд,

впоследствии система передана под управление Cloud Native Computing Foundation.

Концепции

Узел (node) — это отдельная физическая или виртуальная машина, на которой

развёрнуты и выполняются контейнеры приложений. Каждый узел в кластере

содержит сервисы для запуска приложений в контейнерах (например, Docker),

а также компоненты, предназначенные для централизованного управления уз-

лом.

Под (pod) — базовая единица для управления и запуска приложений, один или

несколько контейнеров, которым гарантирован запуск на одном узле, обеспечи-

вается разделение ресурсов и предоставляется уникальный в пределах кластера

IP-адрес. Последнее позволяет приложениям, развёрнутым на поде, использо-

вать фиксированные и предопределённые номера портов без риска конфлик-

та. Поды могут напрямую управляться с использованием API Kubernetes или

управление ими может быть передано контроллеру.

∙

Том (volume) — общий ресурс хранения для совместного использования из кон-

тейнеров, развёрнутых в пределах одного пода.

∙

Метка (label), селектор метки (label selector) — все объекты управления (уз-

лы, поды, контейнеры) в Kubernetes помечаются метками, селекторы меток —

это запросы, которые позволяют получить ссылку на объекты, соответствую-

щие какой-то из меток; метки и селекторы — это главный механизм Kubernetes,

который позволяет выбрать, какой из объектов следует использовать для за-

прашиваемой операции.

∙

Сервис (service) — совокупность логически связанных наборов подов и поли-

тик доступа к ним. Например, сервис может соответствовать одному из уров-

ней программного обеспечения, разработанного в соответствии с принципами

многоуровневой архитектуры программного обеспечения. Набор подов, соот-

ветствующий сервису, получается в результате выполнения селектора соответ-

ствующей метки. Kubernetes обеспечивает функции обнаружения сервисов и

маршрутизации по запросу, в частности, система умеет переназначать необхо-

димые для обращения к сервису IP-адрес и доменное имя сервиса различным

подам, входящим в его состав. При этом обеспечивается балансировка нагруз-

ки между подами, чьи метки соответствуют сервису в стиле Round robin DNS,

а также корректная работа в том случае, если один из узлов кластера вышел

из строя и размещённые на нём поды автоматически переместились на другой.

По умолчанию сервис доступен внутри управляемого Kubernetes кластера, на-

пример, поды бэкенда группируются для обеспечения балансировки нагрузки

и в таком виде предоставляются фронтенду, но он может быть настроен и для

того, чтобы предоставлять доступ к входящим в его состав подам извне, как к

единому фронтенду.

∙

Контроллер (controller) — это процесс, который управляет состоянием класте-

ра, пытаясь привести его от фактического к желаемому; он делает это, оперируя

набором подов, который определяется с помощью селекторов меток, являющих-

ся частью определения контроллера. Выполнение контроллеров обеспечивается

компонентом Kubernetes Controller Manager. Один из типов контроллеров, са-

мый известный — это контроллер репликации (Replication Controller), который

обеспечивает масштабирование, запустив указанное количество копий пода в

кластере. Он также обеспечивает запуск новых экземпляров пода, в том случае

5если узел, на который работает управляемый этим контроллером под, выходит

из строя. Другие контроллеры, входящие в основную систему Kubernetes, вклю-

чают в себя DaemonSet Controller, который обеспечивает запуск пода на каждой

машине (или подмножеством машин) и Job Controller для запуска подов, кото-

рые выполняются до завершения, например, как часть пакетного задания.

∙

Операторы (operators) — специализированный вид программного обеспечения

Kubernetes, предназначенный для включения в кластер сервисов, сохраняющих

своё состояние между выполнениями (stateful), таких, как СУБД, системы мо-

ниторинга или кэширования. Назначение операторов — предоставить возмож-

ность управления stateful-приложениями в кластере Kubernetes прозрачным

для способом и скрыть подробности их настроек от основного процесса управ-

ления кластером Kubernetes.

Архитектура

Система реализует архитектуру “ведущий — ведомый”: выделяется подсистема

управления кластером, а часть компонентов управляют индивидуальными ведомыми

узлами (называемых собственно узлами Kubernetes).

∙

Подсистема управления обеспечивает управление распределением нагрузки и

коммуникациями внутри кластера и состоит из следующих приложений:

– Etcd — компонент подсистемы управления, отвечающий за согласован-

ное хранение конфигурационных данных кластера, в некотором смысле

— распределённый эквивалент каталога /etc Unix-систем. Реализован как

легковесная распределённая NoSQL-СУБД класса “ключ — значение”.

– Сервер API — ключевой компонент подсистемы управления, предостав-

ляющий API в стиле REST (с использованием коммуникации в формате

JSON поверх HTTP-транспорта), и используемый для организации внеш-

него и внутреннего доступа к функциям Kubernetes. Сервер API обновляет

состояние объектов, хранящееся в etcd, позволяя своим клиентам управ-

лять распределением контейнеров и нагрузки между нодами управляемой

системы.

– Планировщик (scheduler) — компонент подсистемы управления, который

выбирает, на каком узле должен выполняться конкретный под, опираясь

на критерии доступности ресурсов. Планировщик отслеживает использо-

вание ресурсов на каждом из узлов, обеспечивая распределение нагрузки

так, чтобы она не превышала доступный объём ресурсов.

– Менеджер контроллеров (controller manager) — процесс, выполняю-

щий основные контроллеры Kubernetes, такие как DaemonSet Controller

и Replication Controller run in. Контроллеры взаимодействуют с сервером

API Kubernetes, создавая, обновляя и удаляя управляемые ими ресурсы

(поды, точки входа в сервисы, и другие).

– Kubectl — интерфейс командной строки, наряду с API обеспечивающий

управление ресурсами, подконтрольными Kubernetes.

∙

Компоненты узлов

– Kubelet отвечает за статус выполнения подов на узле — отслеживает,

корректно ли выполняется каждый из контейнеров, находясь в рабочем со-

стоянии. Kubelet обеспечивает запуск, останов и управление контейнерами

6приложений, организованными в поды. Если обнаруживается, что какой-

то из подов находится в неверном состоянии, компонент пытается осуще-

ствить его повторное развёртывание и перезапуск на узле. Статус самого

узла отправляется на подсистеме управления каждые несколько секунд в

форме диагностических сообщений (heartbeat message). Если мастер-узел,

исходя из содержания этих сообщений или их отсутствия, обнаруживает,

что конкретный узел не работает должным образом, процесс подсистемы

управления Replication Controller пытается перезапустить необходимые по-

ды на другом узле, находящемся в рабочем состоянии.

– Kube-proxy — компонент, являющийся комбинацией сетевого прокси-сервера

и балансировщика нагрузки. Реализованные в нём операции сетевого уров-

ня используют абстракцию сервиса. Он отвечает за маршрутизацию вхо-

дящего трафика на конкретные контейнеры, работающие в пределах по-

да, расположенного на узле. Маршрутизация обеспечивается на основе IP-

адреса и порта входящего запроса.

– cAdvisor — агент системы внутреннего мониторинга Kubernetes, собира-

ющий метрики производительности и информацию об использовании кон-

тейнерами, работающими в пределах ноды, таких ресурсов как время рабо-

ты центрального процессора, оперативной памяти, нагрузку на файловую

и сетевую системы.

∙

Инструменты

– Minikube — специализированная конфигурация Kubernetes, предназна-

ченная для развёртывания на локальной машине, например, компьютере

разработчика, применяется для изучения и локальных экспериментов над

Kubernetes;

– Helm — официальный менеджер пакетов Kubernetes, функциональный

эквивалент apt-get и yum;

– Monocular — веб-интерфейс для управления пакетами, упакованными в

соответствии со стандартами Helm.

# 2.3 Виртуализация

Виртуализация — предоставление набора вычислительных ресурсов или их ло-

гического объединения, абстрагированное от аппаратной реализации, и обеспечи-

вающее при этом логическую изоляцию друг от друга вычислительных процессов,

выполняемых на одном физическом ресурсе.

Примером использования виртуализации является возможность запуска несколь-

ких операционных систем на одном компьютере: при том каждый из экземпляров та-

ких гостевых операционных систем работает со своим набором логических ресурсов

(процессорных, оперативной памяти, устройств хранения), предоставлением которых

из общего пула, доступного на уровне оборудования, управляет хостовая операцион-

ная система — гипервизор. Также могут быть подвергнуты виртуализации сети пе-

редачи данных, сети хранения данных, платформенное и прикладное программное

обеспечение[4].

Существует множество видов виртуализации, здесь будут рассмотрены два из

них:

аппаратная виртуализация и виртуализация на уровне операционной

системы .

Аппаратная виртуализация

Аппаратная виртуализация — виртуализация с поддержкой специальной процес-

сорной архитектуры. В отличие от программной виртуализации, с помощью данной

техники возможно использование изолированных гостевых систем, управляемых ги-

первизором напрямую. Гостевая система не зависит от архитектуры хостовой плат-

формы и реализации платформы виртуализации.

Аппаратная виртуализация обеспечивает производительность, сравнимую с про-

изводительностью невиртуализованной машины, что дает виртуализации возмож-

ность практического использования и влечет её широкое распространение. Наиболее

распространены технологии виртуализации Intel-VT и AMD-V.

Платформы, использующие аппаратную виртуализацию:

∙ IBM LPAR

∙ VMware

∙ Hyper-V

∙ Xen

∙ KVM

∙ Bhyve

Виртуализация на уровне операционной системы

Виртуализация на уровне операционной системы позволяет запускать изолиро-

ванные и безопасные виртуальные машины на одном физическом узле, но не позво-

ляет запускать операционные системы с ядрами, отличными от типа ядра базовой

операционной системы. При виртуализации на уровне операционной системы не су-

ществует отдельного слоя гипервизора. Вместо этого сама хостовая операционная

система отвечает за разделение аппаратных ресурсов между несколькими виртуаль-

ными машинами и поддержку их независимости друг от друга.

Среди реализаций:

∙ Solaris Containers/Zones

∙ FreeBSD Jail

∙ Linux-VServer

∙ LXC (Linux Containers)

∙ FreeVPS

∙ OpenVZ

∙ Virtuozzo

# AWS Compute

AWS Compute предлагает полный набор вычислительных сервисов, позволяю-

щих разрабатывать, развертывать, запускать и масштабировать приложения и ра-

бочие нагрузки[5].

Сервисы:

∙

AWS Elastic Cloud Computing (EC2) : позволяет запускать любые прило-

жения, контролировать и управлять функциями уровня сервера или класте-

ра, такими, как масштабирование и развертывание. Единица развертывания —

виртуальные машины. Клиент управляет сервером, ОС и программным обес-

печением для развертывания (при необходимости).

∙

AWS Elastic Container Service (ECS) : обеспечивает запуск приложений,

упакованных в контейнеры Docker, без сохранения состояния или с фиксацией

состояния. Единица развертывания — контейнеры. Клиент выделяет и масшта-

бирует ресурсы сервера и управляет его загрузкой и доступностью.

∙

AWS Lambda : позволяет запускать инициируемые событиями приложения без

сохранения состояния, требующие быстрого времени отклика. Единица развер-

тывания — код.

∙

AWS Lightsail : позволяет запускать простые приложения и веб-сайты на од-

ном или нескольких серверах по предсказуемой низкой цене. Единица разверты-

вания — виртуальный частный сервер (инстанс). Клиент управляет сервером,

ОС и другим программным обеспечением через интуитивно понятную консоль

Lightsail. При этом он получает меньше возможностей контроля и вариантов

настройки, чем при использовании EC2.

# 2.5 AWS Storage & Database

Платформа AWS предлагает полный спектр сервисов облачных хранилищ, поз-

воляющих обеспечить соответствие нормативным требованиям как к приложениям,

так и к архивированию[6].

Сервисы:

∙

Amazon Elastic Block Store (EBS) : высокодоступное и надежное блочное

хранилище с низкой задержкой для Amazon EC2. С его помощью можно добить-

ся необходимых для конкретного приложения объема хранилища, эффективно-

сти и стоимости. Сервис EBS предназначен для рабочих нагрузок, требующих

наличия постоянного хранилища, доступного отдельным инстансам EC2. Тра-

диционные примеры использования включают реляционные БД NoSQL (такие

как Microsoft SQL Server, MySQL, Cassandra и MongoDB), ядра систем ана-

литики больших данных (такие как система Hadoop/HDFS и Amazon EMR),

приложения для обработки журналов и потоковых данных (такие как Kafka

и Splunk), а также приложения для хранения данных (такие как Vertica и

Teradata).

∙

Amazon Elastic File System (EFS) : простое масштабируемое файловое хра-

нилище, предназначенное для использования с инстансами Amazon EC2 в об-

лаке AWS. Оно предоставляет интерфейс файловой системы со стандартной

9семантикой доступа к файлам для инстансов Amazon EC2 Ресурсы EFS расши-

ряются и сокращаются автоматически, при этом обеспечивается высокая про-

пускная способность и стабильно низкий уровень задержек. EFS обеспечивает

высокую доступность, надежность и производительность для широкого спек-

тра рабочих нагрузок и приложений, включая большие данные и аналитику,

рабочие процессы обработки мультимедийных данных, управление контентом,

распространение файлов через Интернет, контейнерные хранилища и домашние

каталоги.

∙

Amazon Simple Storage Service (S3) В платформе от Amazon нет  
привычной файловой системы — она распределенная. Здесь нет такого понятия как  
файл или каталог: на практике это означает, что ты не сможешь взять FTP- или  
SSH-клиент и обратиться к ним.

Amazon S3 концептуально — это очень простое key-based хранилище. Это означит,  
что когда ты размещаешь данные, им присваивается уникальный ключ объекта,  
который дальше может быть использован, чтобы эти файлы получить. Каждый объект  
находится внутри так называемого bucket’а — это фундаментальное понятие S3,  
означающее контейнер для хранения любого количества объектов. Но поскольку ключ  
может быть строкой, то он может имитировать иерархию каталогов. Последнее  
означает, что если у тебя есть файл scanner.zip в каталоге xtoolz, то ты можешь  
залить его в хранилище S3 и присвоить ему ключ xtoolz/scanner.zip.

Любые объекты находятся в облаке не сами по себе, а внутри так называемого  
bucket’а. Bucket или ведро — это концептуальное понятие контейнера внутри Amazon  
S3, в котором может находиться нелимитированное количество объектов. Название  
bucket должно быть уникально по всей системе Amazon S3, потому как является  
идентификатором для доступа к объектам bucket’а извне. Например, если объект  
называется xtoolz/scanner.zip и находится в bucket’е xakep, то к нему можно  
обратиться, используя URL http://xakep.s3.amazonaws.com/xtoolz/scanner.zip.  
Обращаться к bucket’ам можешь только ты сам, а можешь разрешить это делать  
другим. С помощью атрибутов объекта ты можешь разрешить или запретить доступ  
другим людям, которые хотят скачивать или заливать файлы в твои bucket’ы. Так,  
чтобы любой человек мог скачать объект из твоего хранилища по адресу http://xakep.s3.amazonaws.com/xtoolz/scanner.zip,  
ему должны быть присвоены права для чтения "All".

Для работы с объектами используются технологии REST и SOAP, а все операции  
выполняются с помощью запросов (именно поэтому за них надо платить). Пять  
основных операций: создать bucket, записать объект (указав уникальный ключ и  
bucket), считать объект (можно скачать файл по HTTP или BitTorrent протоколу),  
удалить объект, вывести листинг ключей (получить список всех ключей, хранящихся  
внутри указанного "ведра").

∙

Amazon Glacier : очень экономичный сервис объектного хранилища с высокой

степенью надежности для длительного хранения и архивации данных любого

типа. Glacier — это решение для клиентов, которым требуется недорогое храни-

лище для редко запрашиваемых данных. Оно может стать заменой ленточным

хранилищам для мультимедиа и развлекательных приложений и соответствует

всем требованиям для организаций, работающих в строго регулируемых от-

раслях, таких как здравоохранение, медико-биологические исследования или

финансовые услуги.

∙

AWS Storage Gateway : программное устройство, обеспечивающее эффектив-

ное подключение локальной среды к облачному хранилищу Amazon. Это реше-

ние предлагает локальное хранилище с высокооптимизированным подключени-

ем к облачному хранилищу AWS и может быть особенно полезным в системах

миграции, расширения в облако и многоуровневого хранения данных. Заме-

ните ленточные хранилища с сохранением существующих рабочих процессов,

подключайте к локальным рабочим нагрузкам дополнительные ресурсы по тре-

бованию и расширяйте существующие локальные хранилища за счет уровня

облака.

∙

Сервисы передачи данных : Amazon предлагает набор сервисов переноса

данных, позволяющих осуществить миграцию данных в облако AWS или из

него. При помощи этих сервисов можно быстро и безопасно переносить ар-

хивы размером в несколько петабайтов, ускорять передачу данных по сети в

существующей инфраструктуре и записывать потоковые данные из множества

источников.

# AWS Database

В число сервисов баз данных Amazon входят реляционные базы данных для тран-

закционных приложений, нереляционные базы данных для приложений в масштабе

всего Интернета, хранилище данных для аналитики, хранилище данных в памяти

для кэширования и рабочих нагрузок в режиме реального времени, графовая база

данных для создания приложений со сложносвязанными данными, база данных вре-

менных рядов для оценки изменений с течением времени, а также база данных для

10реестров, обеспечивающая полную и проверяемую регистрацию транзакций. AWS

Database Migration Service помогает экономично и просто перенести существующие

базы данных в AWS.

Сервисы:

∙

Реляционные БД —

Amazon Aurora (MySQL, PostgreSQL), Amazon RDS

(MySQL, PostgreSQL, MariaDB, Oracle, SQL Server),

Amazon Redshift . При-

менение: традиционные, ERP- и CRM-приложения, а также приложения для

интернет-коммерции.

∙

БД на основе пар “ключ-значение” —

Amazon DynamoDB . Применение: при-

ложения, работающие в масштабе всего Интернета, торги в режиме реального

времени, корзины покупок и предпочтения клиентов.

∙

Документные БД —

Amazon DynamoDB . Применение: управление контен-

том, индивидуальная настройка и мобильные приложения.

∙

БД в памяти —

Amazon ElastiCache для Redis , Amazon ElastiCache для

Memcached . Применение: кэширование, игровые таблицы лидеров и аналити-

ка в режиме реального времени.

∙

Графовые БД —

Amazon Neptune . Применение: выявление мошенничества,

социальные сети и сервисы рекомендаций.

∙

БД временных рядов —

Amazon Timestream . Применение: приложения IoT,

DevOps и промышленная телеметрия.

# 2.7 AWS Machine learning

В AWS имеется самый широкий и разнообразный набор сервисов машинного обу-

чения и AI[8].

Сервисы ML

Amazon SageMaker : позволяет разработчикам и специалистам по работе с дан-

ными быстро и просто создавать, обучать и развертывать модели машинного обу-

чения в любом масштабе. Этот сервис упрощает внедрение машинного обучения в

различных сценариях использования и отраслях: от выполнения моделей для обна-

ружения мошенничества в режиме реального времени и эффективного анализа био-

логического воздействия разрабатываемых лекарств до прогнозирования событий в

спортивных матчах.

Сервисы AI

Предварительно обученные сервисы AI AWS предоставляют готовые интеллек-

туальные функции, которые можно использовать в своих приложениях и рабочих

процессах. Сервисы AI можно без труда интегрировать со своими приложениями

для использования в стандартных сценариях использования, например для созда-

ния персонализированных рекомендаций, модернизации контакт-центра, повышения

безопасности и защищенности и увеличения активности клиентов. Так как мы ис-

пользуем те же технологии глубокого обучения, которые применяются для сервиса

11Amazon.com и наших сервисов ML, вы получаете качество и точность, обеспечива-

емые непрерывно обучаемыми API. И самое лучшее заключается в том, что для

использования сервисов AI на AWS не требуется опыт в сфере машинного обучения.

Области применения:

∙

Рекомендации (Amazon Personalize): Персонализируйте информацию, предо-

ставляемую вашим клиентам, с помощью той же технологии создания рекомен-

даций, которая применяется в сервисе Amazon.com.

∙

Прогнозирование (Amazon Forecast): Создавайте модели для точного прогно-

зирования на основе той же технологии машинного обучения для прогнозиро-

вания, которая применяется в сервисе Amazon.com.

∙

Анализ изображений и видео (Amazon Rekognition): Добавьте в свои прило-

жения функции анализа изображений и видео, чтобы каталогизировать ресур-

сы, автоматизировать рабочие процессы обработки мультимедиа и извлекать

смысловую информацию.

∙

Расширенная текстовая аналитика (Amazon Comprehend): Используйте

функции обработки естественных языков, чтобы извлекать необходимые све-

дения и информацию о связях из неструктурированного текста.

∙

Анализ документов (Amazon Textract): Автоматически извлекайте текст и

данные из миллионов документов за считанные часы, сокращая объемы руч-

ного труда.

∙

Голос (Amazon Polly): Включите функцию преобразования текста в речь с

естественным звучанием для своих приложений.

∙

Диалоговые агенты (Amazon Lex): Легко создавайте диалоговые агенты для

повышения качества обслуживания клиентов и увеличения эффективности ра-

боты контакт-центров.

∙

Перевод (Amazon Translate): Расширьте свою аудиторию, включив в нее кли-

ентов, разговаривающих на разных языках, с помощью эффективных и эконо-

мически эффективных функций перевода.

∙

Транскрибирование (Amazon Transcribe): Легко добавляйте высококачествен-

ные функции преобразования речи в текст для приложений и рабочих процес-

сов.

Платформы ML

Доступны

TensorFlow , PyTorch , Apache MXNet и другие популярные платформы,

на которых вы можете проводить эксперименты и настраивать алгоритмы машинно-

го обучения. Вы можете использовать выбранную вами платформу в качестве управ-

ляемого интерфейса в сервисе Amazon SageMaker или применять образы AWS Deep

Learning AMI (Amazon Machine Image), которые полностью настроены с помощью

последних версий самых популярных платформ и средств глубокого обучения.

# 122.8 AWS Internet of things

Сегодня в домах, на заводах, объектах нефтедобычи, в больницах, автомобилях и

других местах используются миллиарды различных устройств. Стремительный рост

числа устройств вызывает потребность в решениях, позволяющих подключать эти

устройства к сети, собирать, хранить и анализировать их данные. AWS IoT предла-

гает широкие возможности, охватывающие множество компонентов — от периферий-

ных устройств до облачных систем. Это позволяет создавать IoT-решения практиче-

ски для любых устройств и примеров использования. Интеграция AWS IoT с сервиса-

ми искусственного интеллекта расширяет интеллектуальные возможности устройств

даже при отсутствии подключения к Интернету. AWS IoT работает на основе облака

AWS, используется миллионами клиентов в 190 странах и без труда масштабиру-

ется по мере роста парка устройств и потребностей бизнеса. Кроме того, AWS IoT

обладает полным комплексом возможностей безопасности, что позволяет создавать

превентивные политики безопасности и немедленно реагировать на потенциальные

угрозы[9].

Промышленные решения

Промышленный IoT (IIoT) позволяет сократить разрыв между традиционной

промышленной инфраструктурой и новыми технологиями, такими как машинное

обучение и облачные сервисы. Клиенты используют приложения IIoT для техни-

ческого обслуживания, удаленного мониторинга и прогнозирования проблем с ка-

чеством. Сервисы AWS IoT позволяют промышленным компаниям из добывающей,

энергетической, коммунальной, производственной, сельскохозяйственной, нефтегазо-

вой и других отраслей делать логические умозаключения на основе операционных

данных с помощью аналитики и машинного обучения.

“Умный дом”

Технологии “умного дома” повышают качество жизни потребителя, расширяя воз-

можности привычных устройств и предметов — кофеварок, штор, вентиляторов,

камер видеонаблюдения. AWS IoT предоставляет технологии “умного дома”, наде-

ляя интеллектуальные устройства новыми возможностями, такими как подключение

устройств к облаку, профилактическое техобслуживание и анализ. Все эти возможно-

сти находят особое применение в ключевых примерах использования “умного дома”:

автоматизация, обеспечение безопасности, мониторинг и создание домашней сети.

∙

Периферийное ПО

Amazon FreeRTOS — это операционная система для микроконтроллеров, ко-

торая упрощает программирование, развертывание, обеспечение безопасности,

подключение небольших маломощных периферийных устройств и управление

ими.

∙

AWS IoT Greengrass — программное решение для безопасного выполнения

таких задач, как локальные вычисления, передача сообщений, синхронизация,

кэширование данных, а также формирование выводов с использованием ма-

шинного обучения на подключенных устройствах.

132.8.4

∙

Облачные сервисы

AWS IoT Core позволяет сетевым устройствам просто и безопасно взаимо-

действовать с облачными приложениями и другими устройствами.

∙

AWS IoT Device Management упрощает безопасное подключение, организа-

цию, мониторинг подключенных устройств IoT и удаленное управление ими в

любом масштабе.

∙

AWS IoT Device Defender выполняет непрерывный мониторинг и аудит кон-

фигураций IoT, чтобы не допускать отклонений от рекомендаций по безопас-

ности.

∙

AWS IoT Things Graph позволяет без труда подключать различные устрой-

ства к облачным сервисам для создания приложений IoT.

∙

AWS IoT Analytics позволяет без труда выполнять сложный анализ огром-

ных объемов данных IoT.

∙

AWS IoT SiteWise позволяет без труда собирать, структурировать и искать

данные IoT в базах данных промышленных объектов и использовать их для

анализа эффективности оборудования и процессов.

∙

AWS IoT Events позволяет без труда обнаруживать события множества IoT-

датчиков и приложений и реагировать на них.

# 2.9 Google Cloud Platform: Identity & Security

Cloud Identity and Access Management (Cloud IAM) позволяет управлять

разрешениями и правами доступа для ресурсов Google Cloud Platform. Cloud IAM

унифицирует контроль доступа к сервисам GCP, предоставляя согласованный набор

операций[10].

Управление правами доступа осуществляется путем определения

имеет какой

тип доступа

(role) к

чему

кто

(identity)

(resource).

Основные понятия

Понятия, относящиеся к identity:

∙ Google account: любой пользователь GCP

∙ Service account: аккаунт, связанный с определенным приложением, а не поль-

зователем.

∙ Google group: именованная коллекция двух вышеописанных.

∙ G Suite domain: группа аккаунтов, связанных с организацией G Suite.

∙ Cloud Identity domain: то, же, что и предыдущее, но для организаций, не свя-

занных с G Suite.

Понятия, относящиеся к управлению доступом:

∙

Ресурс : любой ресурс GCP, такой, как проект, инстанс Compute Engine или

Cloud Storage bucket.

14∙ Разрешение : определяет действия, разрешенные для ресурса.

∙ Роль : именованный набор разрешений. Разрешения нельзя назначить напря-

мую пользователям.

∙

Политика : набор назначений ролей аккаунтам. Политики образуют иерархию.

# 2.10 Google Cloud Platform: Maps platform

Maps platform — это набор сервисов Google Cloud Platform, связанных с геолока-

цией и картами[11]. Включает в себя такие продукты, как:

∙

Google Maps — предоставление доступа к всемирному, масштабируемому и

настраиваемому сервису карт.

∙

Google Routes — построение надежных маршрутов с учетом большого коли-

чества релевантной информации по всему миру.

∙

Google Places — поиск интересующих пользователя локаций и предоставление

подробной полезной информации о них.

# Microsoft Azure: Media

Службы мультимедиа Azure упрощают использование служб кодирования и по-

токовой передачи видео с высоким разрешением, а также позволяют предоставлять

эти возможности на современных устройствах. Кроме того, эти службы позволяют

извлекать ценные метаданные, используя службу анализа видео с помощью искус-

ственного интеллекта, а также улучшают доступность, распределение и масштаби-

руемость, обеспечивая при этом защиту содержимого[12].

Продукты:

∙

Сервис кодирования — обеспечение студийного качества кодирования в мас-

штабе облака.

∙

Индексатор видео — служба извлечения мультимедийных метаданных на

основе ИИ в Службах мультимедиа Azure.

∙

Потоковая передача — потоковая передача или передача содержимого по

запросу на любое устройство.

∙

Защита контента — более защищенное предоставление содержимого при по-

мощи технологий AES или мульти-DRM.

∙

Сеть кэширующих серверов (CDN) — безопасная и надежная доставка

содержимого благодаря глобальному охвату аудитории и широким функцио-

нальным возможностям.

∙

Проигрыватель мультимедиа Azure — использование единого проигрыва-

теля для всех необходимых возможностей воспроизведения.

# Microsoft Azure: Mobile

Служба Mobile Microsoft Azure предназначена для облегчения создания, развер-

тывания и эксплуатации мобильных приложений для различных платформ[13].

Возможности:

∙ Быстрое создание мобильных приложений

∙ Быстрое добавление функции корпоративного единого входа

∙ Автономная синхронизация данных для построения быстро реагирующих при-

ложений

∙ Подключение приложений к локальным данным

∙ Рассылка индивидуальных push-уведомлений

∙ Автоматическое масштабирование в соответствии с бизнес-требованиями

# Microsoft Azure: Analytics

Azure Stream Analytics — это служба аналитики в реальном времени по требова-

нию, расширяющая возможности для интеллектуальных действий[14].

Возможности:

∙

Мгновенный анализ данных со всех устройств и шлюзов Интернета вещей.

Azure Stream Analytics легко интегрируется с Центром Интернета вещей Azure

и Azure IoT Suite, обеспечивая мощную аналитику в реальном времени для

данных, поступающих от устройств и приложений Интернета вещей. Кроме то-

го, служба Azure Stream Analytics доступна в Azure IoT Edge. Azure Stream

Analytics в IoT Edge помогает разработчикам развертывать решения аналити-

ки ближе к устройствам Интернета вещей практически в реальном времени,

что позволяет максимально эффективно использовать данные, полученные от

устройств.

∙

Простая разработка конвейеров массовой параллельной обработки сложных со-

бытий (CEP). Используйте мощную аналитику в режиме реального времени

с помощью простого декларативного SQL-подобного языка со встроенной под-

держкой темпоральной логики. Обширный набор готовых соединителей, расши-

ренные возможности отладки и мониторинга заданий помогают снизить затра-

ты, так как для разработки не требуются особые навыки. Поддержка настраи-

ваемого кода с помощью пользовательских функций JavaScript дополнительно

расширяет логику потоковой передачи, написанную на языке SQL. Выноски

службы машинного обучения Azure помогают выполнить прогнозную оценку

потоковых данных.

∙

Быстрое начало работы, мгновенное масштабирование и оплата за задание. Вы

можете приступить к работе буквально за несколько секунд: здесь нет инфра-

структуры, о которой нужно беспокоиться, а также серверов, виртуальных ма-

шин или кластеров, которыми нужно управлять. Вы можете мгновенно увели-

чить вычислительную мощность от одной до сотен единиц потоковой передачи

для выполнения любого задания. Вы платите только за вычислительные ресур-

сы, использованные заданием.

16∙

Построение панелей мониторинга, работающих в реальном времени. Быстрое

построение панелей мониторинга, работающих в реальном времени, с помощью

Power BI для динамического просмотра команд и элементов управления. Пане-

ли мониторинга, работающие в реальном времени, позволяют преобразовывать

динамические данные в практичные и информативные графические представ-

ления, помогая вам сосредоточиться на наиболее важных сведениях.

∙

Доступность, аудит и поддержка корпоративного уровня. Гарантированная до-

ставка событий и соглашение об уровне обслуживания корпоративного уровня,

обеспечивающее доступность в течение 99,9% времени, позволяют использовать

Azure Stream Analytics в критически важных рабочих нагрузках. Автоматиче-

ские контрольные точки обеспечивают отказоустойчивую работу с быстрым

перезапуском без потери данных. Модуль Stream Analytics отвечает всем ос-

новным нормам соответствия и доступен в 19 регионах Azure по всему миру,

включая Китай и Германию.

# Google Cloud Platform: Networking

Google Cloud Platform: Networking — это компонент GCP, отвечающий за органи-

зацию сетевого взаимодействия в пределах платформы[15].

3.1 Сетевые компоненты (службы)

∙ VPC (Virtual Private Cloud) — SDN (Software Defined Network)

∙ Cloud Load Balancing — балансировка трафика

∙ Cloud Armor — Web Application Firewall (WAF) для защиты веб-приложений

от DDoS и прочих типов атак

∙ Cloud CDN — сеть доставки и дистрибуции контента

∙ Cloud Interconnect — подключение имеющегося датацентра к GCP

∙ Cloud DNS — служба доменных имен для использования в VPC

∙ Network Service Tiers — уровни обслуживания для VPC (Standard, Premium,

...)

Структура GCP

Глобальность GCP

GCP делится на 11

регионов , 33 зоны и более 100 точек присутствия по всему

земному шару

Регионы, зоны и ресурсы

Регион — это географическая область, разделенная на несколько зон .

Глобальные ресурсы:

∙ Сети

∙ Маршруты

∙ Настройки файрволла

Региональные ресурсы:

∙ Статические внешние IP-адреса

∙ Подсети

Зональные ресурсы:

∙ Инстансы (виртуальные машины)

∙ Хранилища (диски)

3.5 Virtual Private Cloud (VPC)

VPC — это глобальная виртуальная частная сеть, позволяющая управлять рас-

положенными в ней ресурсами GCP.

Все инстансы в рамках одного VPC имеют внутренние IP-адреса и могут взаимо-

действовать друг с другом.

Проект

Проект — это организационное понятие, в основном связанное с биллингом. Про-

ект может содержать одно или несколько VPC и изолирован от других проектов.

Другие ресурсы

∙ Подсети — группы инстансов (региональный уровень)

∙ Сети — группы подсетей (глобальный уровень)

∙ Маршруты — правила маршрутизации между между (под)сетями

∙ Настройки файрволла — правила, регулирующие прохождение трафика как

на уровне сети в целом, так и между отдельными инстансами

∙

IP адреса — каждый инстанс имеет

и обычно также

внешний

внутренний

(используемый внутри сети)

(для коммуникации между сетями и с Интернет) IP

адрес

Как начать использовать

Free Service Tier:

https://cloud.google.com/free/

∙ 12 месяцев

∙ Начальная сумма $300

∙ Доступны все продукты GCP (с некоторыми ограничениями)

∙ Только один раз, только для новых пользователей GCP