Слайд 1.

Добрый день, уважаемые **члены государственной экзаменационной комиссии** и присутствующие.
Представляется **выпускная квалификационная работа магистра** на тему:
"**Система автоматизации разработки и обеспечения доступности проектов**".

Выполнил студент группы ИУ5-41М Кучеренко Михаил.
Научный руководитель к.т.н., доцент кафедры ИУ5 Галкин Валерий Александрович.

> 14 секунд

Слайд 2.

Цель работы – спроектировать и внедрить систему **автоматизированной инфраструктуры** в для **автоматизации основных процессов разработки и обеспечения доступности**. То есть разработатать систему, способную стать **основой цифровой экосистемы кафедральных проектов**.

Актуальность работы объясняется 4 аспектами, подробно рассмотренными в моем выступлении на студенчекой весне:
- **социальным** - сближение участников учебного процесса для работы над общей системой, как над продуктом деятельности всех коллективов.
- **преемственным** - разработать техническое решение, чтобы содержать все наработки в одном месте в одной форме и предоставлять к ним доступ.
- **технологическим** - возможность опробовать на практике самые современные технологии и пользоваться готовыми модулями единой инфраструктуры.
- **экономическим** - объединение проектов в коммунальную инфраструктуру, что позволит значительно сэкономить ресурсы.

> 50 секунд

Слайд 3.

Для достижения цели требовалось решить следующий перечень задач. Были проведены следующие работы:

- Анализ предметной области - современных IT-инфраструктур
- Из множества вариантов было отобрано ПО, позволяющее обеспечить работу такой инфраструктуры
- Подготовлена аппаратная платформа в виде кластера микрокомпьютеров
- Подготовлены все подсистемы и установлены на кластер
- Проведен анализ нагрузки на подсистему контроля

> 25 секунд

Слайд 4.

Рассмотрим **предметную область проекта на правом рисунке**. Согласно философии DevOps и типичном процессе разработки и обеспечения доступности согласно философии DevOps есть следующий набор **акторов**:

- **Пользователь** - рядовой клиент некоторых сервисов внутри экосистемы. Только пользуется системой снаружи как единым продуктом.
- **Разработчик** – программист или DevOps-инженер команды разработки, занимается подготовкой приложения и передачей этих наработок в экосистему.
- **SR-инженер** – член команды обеспечения доступности экосистемы. Проводит рецензирование кода приложений перед раскладыванием приложения на продукционную систему. Дополнительно занимается отладкой системы, настройкой и реагирует на инциденты.

Объекты:

- Шлюз
- Хранилище
- Сервис контроля
- Мониторинг
- Сервисы для вывода графики
- Сервисы отправки оповещений
- И собственно само приложение

Рассмотрим данную систему **с точки зрения разработчика**. Развертывание проектов внутри цифровой экосистемы **изображено в качестве примера на рисунке слева** - есть два сервиса различной архитектуры, оба из которых зависят от СУБД, все развернутые внутри экосистемы.

> Около 55-59 секунд

Слайд 5.

Проведен анализ аналогов и прототипов. Существует множество систем, обеспечивающих уровень Infrastructure-as-a-Service (OpenStack, OpenShift, Kubernetes), однако все современные Platform-as-a-Service решения предоставляются крупными компаниями исключительно в **коммерческих целях и проприетрно**. Таким образом, подходящих прямых аналогов данной системы найти не удалось, однако, по части функционала есть совпадение с другими системами, представляющим IaaS-решения – OpenStack и Kubernetes. Разумеется, **как более узкопрофильное решение для решения требуемых задач прототип оказался выгоднее**.

> 35 секунд

Слайд 6.

Подразумевается максимальная кроссплатформенность готового решения и возможность затем перенести его на любые аппаратные мощности. Но для создания прототипа разрабатываемой платформы требовалось выбрать аппаратное обеспечение, **на которое будет размещено ПО и компоненты в процессе разработки, отладки и демонстрации**. Для этого проведен анализ трех возможных вариантов размещения - аренда серверов, покупка собственных серверов и облачные решения. Для оценки эффективности каждого из вариантов на основе предварительных требований к составу ПО **было составлено эмпирическое соотношение 1** (в красной рамке) удельной стоимости за условные ресурсы (1 ядро 1 гигагерц, 1 гибибайт оперативной памяти).

Наиболее эффективными оказались варианты:

- Покупки собсвенных серверов Raspberry Pi - для них соотношения и график зависимости от числа серверов (N) приводятся вверху слайда.
- Облачные технологии, т.е. аренда виртуальных машин - для них соотношения приводятся внизу.

Из соотношений следует, что **функция зависимости от числа серверов (N) для покупки собственных серверов** стабилизируется на значении асимптоты, это видно на верхнем графике. А для аренды виртуальных машин монотонно **возрастает для любых комбинаций**, что показано на графиках зависимости от ядер и оперативной памяти. Таким образом наиболее выгодным решением по выбранной метрике является закупка собственных серверов, **что и было произведено**.

> 1 минута 15 секунд

Слайд 7.

Для упрощения восприятия, разработки и поддержки системы используется принцип структурной декомпозиции. Каждая подсистема выступает в роли черного ящика с определенным интерфейсом (API) для других подсистем. Так мы максимизируем подход “as-a-Service” предоставления компонентов как сервис, что **позволяет строить максимально динамическую экосистему**, способную подстраиваться под любые функциональные требования - каждый модуль использует остальные как готовые сервисы.

При разработке и внедрении применялось разделение на этапы с приоритезацией. Пока не запущены все сервисы из младшего этапа система не должна переходить на новый этап развертывания. Далее мы рассмотрим основные подсистемы подробнее.

> 35 секунд

Слайд 8.

Начнем с обзора аппаратной части. Все сервера Raspberry Pi приналежат к определенной группе, группы могут быть вложенными. Каждая группа имеет свое логическое назначение. Глобально все разделяется на 3 основных PROD (оснвоная среда), DEV (среда отладки) и Bastion (сервера для защищенного доступа). Группы показаны на рисунке слева.

Для обеспечения надежной работы серверов применяются системы источников бесперебойного питания. Схема подключения устройств показана на рисунке справа.

> 25 секунд

Слайд 9.

Подсистема передачи информации. Сервера подключаются между собой через несколько коммутаторов и роутер по витой паре или беспрводному подключению. Каждый сервер получает набор статических IP-адресов и виртуальных IP-адресов для сервисов. При этом виртуальные адреса в случае отказа узла захватываются менее приоритетным узлом по протоколу VRRP.

> 19 секунд

Слайд 10.

Для обеспечения работы **сложной заранее неизвестной структуры проектов** в цифровой экосистеме потребовалось разработать специальную схему работы **подсистемы доменных имен**. Ее структурное представление приводится на слайде. Управление записями производится по API и веб-интерфейс, есть динамическое заведение и удаление записей в Consul и перенаправление запросов с кешированием наружу.

> 19 секунд

Слайд 11.

Для **балансировки нагрузки** используется сразу несколько подходов.

- Отказоустойчивость на уровне сети с помощью Keepalived
- Балансировка нагрузки на уровне DNS - цикличная выдача адресов
- Отказоустойчивость на уровне DNS с помощью Consul
- Балансировка нагрузки и отказоустойчивость на уровнях точки входа запросов пользователей и запросов сервисов с помощью Nginx и HAProxy соответственно

> 22 секунды

Слайд 12.

Подсистема мониторинга. Мониторинг осуществляется с помощью Prometheus-стека, где в качестве центрального элемента выступает VictoriaMetrics. Метрики временных рядов собираются в единую базу, откуда могут быть обработаны для создания оповещений о сбоях и отрисовки графиков для команды SR-инженеров. Также собираются и ротируются журналы с каждого из серверов.

> 20 секунд

Слайд 13.

Подсистема безопасного доступа. Для обеспечения максимального уровня информационной безопасности используется принцип эшелонированной защиты. Все взаимодействия с серверами и внутренними сервисами проводятся строго при подключении VPN с двухфакторной аутентификацией. Каждый сервер защищен индивидуальным сетевым фильтром.

> 18 секунд

Слайд 14.

Для хранения данных используется сразу несколько систем, так как различным проектам в составе экосистемы могут потребоваться различные интерфейсы доступа к ним.

- Ceph - S3, RBD и распределенная ФС
- MariaDB - упрощенный SQL
- PostgreSQL - полноценный SQL
- Redis, Memcached и Consul - NoSQL, ключ/значение

> 20 секунд

Слайд 15.

Подсистема сервисов очередей. Она необходима для надежной асинхронной доставки сообщений между сервисами и устройствами интернета вещей. Ее структурная схема приводится на слайде.

> 11 секунд

Слайд 16.

Подсистема выдачи сертификатов спроектирована для предоставления всем сервисам возможности работать по зашифрованным протоколам взаимодействия, таким как HTTPS. На структурной схеме отображено автоматическое получение и распространение сертификатов по серверам.

> 18 секунд

Слайд 17.

Также в состав проекта входят подсистемы, на которых мы не будем подробно сейчас останавливаться. Их список приводится на слайде. Часть подсистем, таких как миварная активная энциклопедия Романа Байбарина и сервисы умного дома Тагира Ханмурзина, будут подробно рассмотрены в отдельный докладах.

> 13 секунд

Слайд 18.

Для синхронизации и централизованного управления сетевыми доступами, слежения за доступами пользователей и надсистемного контроля и мониторинга разработана специальная **подсистема контроля.** Это полностью самостоятельная разработка. Ее структурная схема приводится на слайде.

Из состава подсистмы стоит выделить 2 основных компонента - Overmind-Controller (управляющий узел) и Overmind-Nerv (узел-агент). Хранилище управляющего узла может быть как его составной частью, так и отдельным сервисом.

> 27 секунд

Слайд 19.

На слайде представлена **диагамма запросов между этими компонентами**. Узел-агент регулярно обращается к управдяющему узлу для передачи своего состояния и обновления списков правил и пользователей. В случае нахождения несоответствия между текущим состоянием и переданными списками производится принудительная синхронизация и оповещение управляющего узла.

> 18 секунд

Слайд 20.

Поскольку разрабатываемая система **должна иметь возможность свободно масштабироваться** на сотни серверов требуется оценить потенциальную нагрузку на управляющий узел. Для этого можно представить подсистему контроля в виде модели Марковской цепи. Граф переходов, модель и основные соотношения представлены на слайде. Записав соответвующую СЛАУ, совершаем переход к рекурретной форме соотношению, которую уже можно оценить численными методами.

> 25 секунд

Слайд 21.

Была проведена **оценка реальных показателей времени приема передачи** (RTT) и времени решения (с помощью ApacheBench) на выбранном аппаратном обеспечении и подстановка этих значений в модель, для различных значений количества управляющих узлов (K). По графикам зависимостей средней длины очереди и среднего времени реакции системы можно оценить уровень загрузки системы и сделать вывод о том, когда стоит переходить к большему числу управляющих узлов. Например, при числе сероверов N от 20 до 40 необходимо использовать не менее 5 управляющих узлов, а при N большем 40 но меньшем 60 - не менее 7.

> 26 секунд

Слайд 22.

Развертывание всех подсистем производится в автоматизированном режиме на основе Ansible-сценариев. Также существует инструментарий для автоматизированной преднастройки серверов. За счет этого удалось без доработок внедрить часть компонентов инфраструктуры на кафедральный сервер ИУ5 для осуществления мониторинга, контроля и резервного копирования. Для кафедры было проработано отдельное потенциальное решение на основе виртуальных машин в Баумнском ЦоД-е, схема на слайде. **Для подготовки специалистов, способных работать** с системой на кафедре внедряется элективный курс по GNU/Linux DevOps.

> 30 секунд

Слайд 23.

По итогам работы все поставленные задачи были выполнены, цель считаю достигнутой. Разработанная система удовлетворяет всем функциональным требованиям и имеет богатый потенциал для усовершенствований через систему модулей и отдельных сервисов.

> 15 секунд

Слайд 24.

Список использованных публикаций за время обучения в магистратуре представлен на слайде.

> 5 секунд

Слайд 25.

Спасибо за внимание.