

ПОСТРОЕНИЕ И СОПРОВОЖДЕНИЕ ПРОЕКТОВ СО СЛОЖНОЙ ОБЛАЧНОЙ ИНФРАСТРУКТУРОЙ

Таборовец В.В.¹, Былина А.А.²

Email: Taborovets17122@scientifictext.ru

¹Таборовец Вячеслав Васильевич – кандидат технических наук, доцент,
кафедра программного обеспечения информационных технологий,
факультет компьютерных сетей и систем,
Белорусский государственный университет информатики и радиоэлектроники;

²Былина Александр Александрович – магистрант,
кафедра информационных технологий, инженерно-экономический факультет,
Минский инновационный университет,
г. Минск, Республика Беларусь

Аннотация: статья посвящена рассмотрению проблем, возникающих при проектировании и эксплуатации сложных информационных систем, и поиску и описанию возможных путей их решения с использованием DevOps методологий и технологии контейнеризации. Показано использование облачных технологий при обеспечении доступа к обобщенному фонду конфигурируемых вычислительных ресурсов. Описана необходимость разработки СИС с помощью гибких методологий разработки (Agile). Назван ряд проблем и предложены пути их решения за счет построения инфраструктуры, управляемой кодом, перебрасывания статических защищенных виртуальных туннелей между регионами, изолирования окружения, технологии контейнеризации и др. Предложены пути решения данных проблем за счет построения инфраструктуры управляемой кодом, перебрасывания статических защищенных виртуальных туннелей между регионами, изолирования окружения, технологии контейнеризации, автоматизации сценариев сбоя и др.

Ключевые слова: DevOps методология, инфраструктура, управляемая кодом, мониторинг, гибкие методологии разработки, облачные технологии, инфраструктура, контейнеризация, сложные информационные системы, мониторинг, датацентры.

CONSTRUCTION AND SUPPORT OF PROJECTS WITH COMPLEX CLOUD INFRASTRUCTURE

Taborovets V.V.¹, Bylina A.A.²

¹Taborovets Vjacheslav Vasilyevich – PhD in Techniques, Associate Professor,
DEPARTMENT SOFTWARE FOR INFORMATION TECHNOLOGIES,
FACULTY OF COMPUTER SYSTEMS AND NETWORKS,
BELARUSIAN STATE UNIVERSITY OF INFORMATICS AND RADIOELECTRONICS;

²Bylina Alexander Alexandrovich – Master's degree,
DEPARTMENT INFORMATION TECHNOLOGY,
FACULTY OF ENGINEERING AND ECONOMICS,
MINSK INNOVATION UNIVERSITY,
MINSK, REPUBLIC OF BELARUS

Abstract: the article is devoted to the consideration of problems arising in the design and operation of complex information systems, and the search and description of possible solutions to them using DevOps methodologies and containerization technology. It shows the use of cloud technologies while providing access to a generalized pool of configurable computing resources. The need to develop SIS using flexible development methodologies (Agile) is described. Named a number of problems). A number of problems have been identified and ways of their solution have been proposed by building a managed code infrastructure, moving static protected virtual tunnels between regions, isolating the environment, containerization technology, etc. We propose ways to solve these problems by building an infrastructure with managed code, moving static protected virtual tunnels between regions, isolating the environment, containerization technology, automating fault scenarios, and others.

Keywords: DevOps methodology, code-driven infrastructure, monitoring, flexible development methodologies, cloud technologies, infrastructure, containerization, complex information systems, monitoring, data centers.

УДК 002.6:0044.43

Рассмотрим возможные пути решения проблем, которые возникают на всех этапах построения и сопровождения сложных информационных систем (СИС). Под термином сложные информационные системы в контексте данной статьи, понимается системы, которые имеют множество окружений для разработки, тестирования, размещения, мониторинга и т.д. Такие системы имеют большой набор технологий и языков программирования. Также необходимо иметь в виду, что такие системы строят, используя модель обеспечения удобного сетевого доступа по требованию к некоторому общему фонду конфигурируемых вычислительных ресурсов, т.е. используют облачные вычисления.

В современном быстрорастущем информационном обществе трудно представить продукт, на разработку и создание которого может уйти более двух лет. Для бизнеса это большой риск, так как конечный продукт

может не соответствовать современным тенденциям и принести большие убытки. Как правило, сейчас данные проблемы решаются с помощью различных гибких методологий разработки (Agile). Гибкие методологии разработки – это набор подходов, ориентированных на использование итеративной модели (планирование, разработка, тестирование, улучшение). Иными словами современные программные продукты создаются с использованием вышеупомянутых методологий. Это позволяет быстро выводить на рынок и быть конкурентоспособным продуктом, который может не иметь полного функционала, но уже приносить доход. Такой продукт можно разрабатывать годами, постоянно внося правки и масштабируя его или сложную систему из множества продуктов, ориентируясь на рынок (отзывы, тенденции, прибыльность и т.д.).

Отсюда возникает первая проблема – содержание инфраструктуры. Отметим, что под термином «инфраструктура» имеются в виду не физические сервера, сетевая инфраструктура и другие аппаратные компоненты, а здесь и далее, – это наборы программного обеспечения, библиотек и компонентов операционной системы, которые необходимы для полноценного функционирования приложений или системы. Развитые технологии виртуализации позволяют абстрагироваться от особенностей построения физической инфраструктуры, используя лишь удобные программные интерфейсы. За счет развития облачных технологий и средств автоматизации в датацентрах поднять десятки, сотни и даже тысячи виртуальных серверов стало возможным лишь запуском единственной команды. С появлением таких возможностей, изменились и подходы к системному администрированию [1]. Например, такие СИС как рекламные платформы для видео могут насчитывать более 2000 виртуальных машин, иметь более 100 различных частей системы как frontend так и backend. И это только в облачной инфраструктуре сети. Из чего следует, что таким количеством ресурсов нужно управлять. Так как инфраструктура может быть большой, то никто из инженеров, сопровождающих данную систему, не сможет рассказать о ней все. В случае же, когда система будет выведена из строя, команда инженеров или системных администраторов не сможет в короткие сроки восстановить такую систему, так как информация о системе частично хранится в головах инженеров и не является полной. И даже имея на руках полный набор документации о системе, быстро восстановить систему невозможно.

В такой ситуации могут помочь DevOps методологии, которые пришли в мир разработки программного обеспечения сравнительно недавно. DevOps (Development & Operations) – это набор методологий разработки программного обеспечения, нацеленных на активное взаимодействие и интеграцию специалистов по разработке и специалистов по информационно-технологическому обслуживанию. Основной методологией, которую рекомендуется применять для предотвращения вышеописанного случая, является «Infrastructure as a Code» (инфраструктура, управляемая кодом). Данная методология позволяет описать всю инфраструктуру системы в виде кода и впоследствии повторно использовать с ожидаемым результатом для создания или масштабирования уже настроенной и рабочей версии инфраструктуры [2]. Описанную инфраструктуру можно создавать либо целиком, либо по частям. Кроме того данная методология позволяет отслеживать изменения при условии, что код хранится в распределённой системе управления версиями (Git) или аналогах.

Для построения инфраструктуры, управляемой кодом, предлагается разделить ее на два уровня. Нижний уровень отвечает за описание таких частей инфраструктуры как сеть, пользователи, группы безопасности, конфигурации хранилищ, конфигурация сервисов, которые предоставляет облачный провайдер. Верхний уровень отвечает за оркестрацию, то есть описание конфигурации виртуальных серверов, количество запускаемых серверов, настройку окружений на серверах и их содержимое. Так же на этом уровне возможно описание ролей, а значит, используя модульный подход в описании ресурсов, имеется возможность за считанные минуты запускать и настраивать нужный пользователю стек технологий и таким же образом за считанные минуты поднятый стек масштабировать либо удалять.

Еще одной проблемой, возникающей при построении и сопровождении СИС, является то, что такие системы, как правило, межрегиональные, что в свою очередь затрудняет быстрое, безопасное и надежное распределение трафика между регионами. Следует отметить, что скорость ответа, безопасность и надёжность канала связи являются для рекламной платформы важными составляющими. Сбой одной из двух составляющих может повлечь за собой сбой всей системы и последующие за этим сбоем убытки. Для решения данной проблемы, как правило, используются специальные сервисы-площадки в интернете, которые после некоторой конфигурации позволяют безопасно, быстро и надёжно передавать трафик из региона в регион. Такой способ является простым в реализации, но в тоже время дорогостоящим. Есть дешёвая альтернатива, хотя уровень имплементации выше и для реализации нужен хороший специалист в области сетевых технологий. Данный метод заключается в перебрасывании статических, защищённых виртуальных тоннелей между регионами, а также последующее их сопровождение и масштабирование при увеличении нагрузки. Данный метод можно автоматизировать для сокращения затрат на специалиста.

Мониторинг – еще одна проблема, с которой сталкиваются команды разработчиков и сопровождения СИС.

В качестве примера приведем следующую ситуацию с рекламной платформой, которая находится в различных подсетях. Система мониторинга находится в одной подсети, а инцидент произошел в другой подсети, более того в поврежденной части подсети отказал сервер, который отвечал за соединение двух

подсетей. Следует учитывать, что большинство систем мониторинга – как и в этом случае – работают по принципу клиент-сервер. В такой ситуации сервер не знает, что произошел сбой, так как не получает сигналов от клиентов.

Для решения вышеописанной проблемы предлагается использовать распределенный подход к проектированию системы мониторинга. Такой подход, как правило, предусмотрен в популярных системах мониторинга. Этот подход заключается в размещении в подсетях по одному подчиненному мониторинг-серверу и размещению одного мастер мониторинг-сервера в основной подсети. Данный подход позволит сигнализировать как о проблемах соединения между подсетями, так и о проблемах в самой подсети.

Изолирование окружения является еще одной проблемой, которая в большей степени относится к созданию окружений для тестирования и разработки. Изолирование окружения необходимо для предотвращения конфликта версий пакетов, установленных в системе, к тому же в изолированной среде проще смоделировать различные окружения и ситуации. Ранее для этой цели использовались отдельные сервера, сейчас используются виртуальные машины. Для ускорения и упрощения процесса создания изолированного окружения можно использовать технологию контейнеризации. Технология контейнеризации имеет ряд преимуществ перед технологиями виртуализации, так как основана на механизме, который, ограничивает и изолирует вычислительные ресурсы (процессорные, сетевые, ресурсы памяти, ресурсы ввода-вывода) для групп процессов. Механизм позволяет образовывать иерархические группы процессов с заданными ресурсными свойствами и обеспечивает программное управление ими. Технология контейнеризации позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть перенесён на любую Linux-систему с поддержкой cgroups в ядре и использовать как самостоятельную единицу, либо как базовую составляющую для других контейнеров, так как технология контейнеризации позволяет использовать принцип наследования [3]. Опираясь на это, в СИС рекомендуется создание и использование единого базового контейнера. Базовый контейнер будут сопровождать системные инженеры или DevOps инженеры. На основе базового контейнера уже разработчики и QA-инженеры (Quality Assurance engineer) смогут создавать свои контейнеры с необходимым им инструментарием и окружением.

При сопровождении СИС возникают трудности с ликвидацией сбоев. Для улучшения процессов связанных с восстановлением работоспособности СИС, необходима создавать и поддерживать в актуальном состоянии базу знаний, в которой, будет храниться вся техническая документация о проекте. Необходимо описывать и сохранять в базе знаний информацию о сбоях и информацию о способах их ликвидации в виде небольших сценариев, в которых будет описана последовательность действий. Впоследствии, описанные сбой проще будет ликвидировать либо автоматизировать сценарии для быстрого самовосстановления системы и оповещения о сбое при помощи инструментов мониторинга.

Перечисленные проблемы и предложенные подходы к их решению позволят уменьшить затраты на сопровождение, масштабировать имеющиеся СИС или построить с нуля новую сложную информационную систему, а также сократить затраты на разработку и повысить ее надежность.

Список литературы / References

1. *Овчинников М.А., Овчинникова Е.В.* Инфраструктура управляемая кодом: методология и инструменты // Вестник науки и образования Северо-Запада России, 2015. № 3. С. 63-67.
2. *Davis J., Daniels R.* Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale // O'Reilly Media, 2016. 410 p.
3. Official Documentation Dockers. [Электронный ресурс], 2015. Режим доступа: <https://docs.docker.com/> (дата обращения: 04.11.2017).