

Д.Ж. Алиппаева, Г.А. Бабулова  
**Автоматизация управления предприятием и облачные  
технологии**



**Костанай, 2024**

Министерство науки и высшего образования  
НАО «Костанайский региональный университет имени А. Байтұрсынұлы»  
Факультет машиностроения энергетики и информационных технологий  
Кафедра информационных систем

**Д.Ж. Алиппаева, Г.А. Бабулова**

**Автоматизация управления предприятием и облачные технологии**

Учебное пособие

Костанай, 2024

**УДК 004.6(075.8)**  
**ББК 32.973.202я73**  
**А 50**

**Составители:**

Алиппаева Динара Жанкабыловна, магистр, старший преподаватель кафедры информационных систем КРУ имени А. Байтұрсынұлы, Костанай, Казахстан

Бабулова Гульмира Айтжановна, магистр, старший преподаватель кафедры информационных систем КРУ имени А. Байтұрсынұлы, Костанай, Казахстан.

**Рецензенты:**

Классен Ю.В. – кандидат технических наук, ассоциированный профессор кафедры «информационных технологий и автоматике», КИЭУ имени М. Дулатова, Костанай

Жуаспаев Т.А. – кандидат физико-математических наук, старший преподаватель кафедры «информационных технологий и автоматике», КИЭУ имени М. Дулатова, Костанай

Ысмагул Р.С. – кандидат физико-математических наук, и.о. ассоциированного профессора кафедры «математики и физики» КРУ имени А. Байтұрсынұлы

Алиппаева Д.Ж.

**А 50** Автоматизация управления предприятием и облачные технологии: Учебное пособие / Д.Ж. Алиппаева, Г.А. Бабулова.- Костанай: КРУ имени А. Байтұрсынұлы, 2024. – 66 с.

**ISBN 978-601-356-392-3**

В учебном пособии представлены рекомендации, теоретический и практический материалы, необходимые для знаний и компетенций при изучении дисциплины Автоматизация управления предприятием и облачные технологии. Даны методические сведения, изложены краткие рекомендации по самостоятельной работе. Пособие предназначено для студентов направлений подготовки: «Информационно-коммуникационные технологии», однако может быть использовано и для других областей образования в высших учебных заведениях.

**ББК 32.973.202я73**  
**УДК 004.6(075.8)**

**ISBN 978-601-356-392-3**

Утверждено и рекомендовано к изданию Учебно-методическим советом Костанайского регионального университета имени А. Байтұрсынұлы, 25.04. 2024г., протокол №2

© Костанайский региональный  
университет им. А.  
Байтұрсынұлы  
© Алиппаева Д.Ж., 2024

## Содержание

<b>Введение</b> .....	5
<b>Тема 1 Anthos: Современная платформа для управления приложениями в современном гибридном и мультиоблачном мире</b> .....	6
1.1 Единое управление с помощью Anthos.....	6
1.2 Интеграция контейнеров и управление ими с помощью Anthos GKE.....	8
1.3 Возможности корпоративной безопасности предприятия.....	8
1.4 Сетевая интеграция платформы.....	11
1.5 Anthos GKE для управления корпоративными рабочими нагрузками.....	13
<b>Тема 2 Настройка политики с помощью Anthos Config Management</b> .....	14
2.1 Архитектура управления конфигурацией Anthos .....	14
2.2 Мониторинг и управление услугами с помощью Anthos Service Mesh.....	18
2.3 Архитектура сервисной сетки Anthos Service Mesh .....	18
<b>Тема 3 Бессерверная работа с облачным управлением для Anthos</b> .....	23
3.1 Сложность работы с микросервисами в Kubernetes.....	23
3.2 Бессерверный сервис в Kubernetes: открытый и расширяемый.....	24
3.3 Операции Cloud Run для Anthos.....	25
<b>Тема 4 Разработка приложений для Anthos</b> .....	28
4.1 Кодирование приложений для Kubernetes.....	28
4.2 Создание артефактов сборки.....	28
4.3 Защита программного обеспечения.....	29
4.4 Масштабируемая работа (масштабный запуск).....	30
<b>Тема 5 Создание безопасной программной платформы с помощью Anthos и GitOps</b> .....	31
5.1 Интегрированные услуги Anthos.....	31
5.2 Интеграция с облачным портфолио Google Cloud.....	32
5.3.Партнерская экосистема Anthos.....	34
5.4 Возможности Anthos в гибридной и мультиоблачной среде.....	36
<b>Примеры использования Anthos</b> .....	39
<b>Заключение</b> .....	64
<b>Список использованных источников</b> .....	65

## Введение

В современном мире, пронизанном быстрыми темпами развития информационных технологий, автоматизация управления предприятием и использование облачных технологий становятся неотъемлемой частью успешной бизнес-стратегии. Возможности, которые предоставляют современные технологии, переворачивают представление о том, как эффективно управлять бизнесом и обеспечивать его устойчивость и конкурентоспособность.

Это пособие предназначено для тех, кто стремится разобраться в том, каким образом автоматизация и облачные технологии могут быть внедрены в управленческие практики предприятия. Рассматриваются не только основные принципы и преимущества этих технологий, но и конкретные шаги по их внедрению и оптимизации в рамках организации.

Сфера применения автоматизации и облачных технологий огромна и постоянно расширяется. От сокращения операционных издержек до повышения производительности персонала, от улучшения качества обслуживания клиентов до увеличения масштабируемости бизнеса – возможности этих технологий впечатляют.

В этом пособии рассматривается современная платформа Anthos для управления приложениями в современном гибридном и мультиоблачном мире, возможности её использования организациями для достижения своих стратегических целей и обеспечения стабильного роста в динамичной и конкурентной среде современного бизнеса. Anthos является общедоступной платформой и организации различных отраслей используют преимущества облака, контейнеров и сервисной сети в своих приложениях.

## **Тема 1 Anthos: Современная платформа для управления приложениями в современном гибридном и мультиоблачном мире**

Anthos - это первая современная платформа приложений, обеспечивающая согласованную разработку и эксплуатацию в различных средах — в нескольких облаках, локально и на периферии. На сегодняшний момент данную платформу используют в различных отраслях для преобразования своих портфелей критически важных приложений.

Anthos помогает развертывать, управлять и оптимизировать приложения, как устаревшие, так и облачные. Anthos работает на абстрагированном уровне инфраструктуры, и приложения, работающие на нем, имеют доступ к ценным сервисам, которые позволяют им работать эффективно и безопасно, не опасаясь блокировки или ненужной сложности. Anthos оптимизирует затраты на инфраструктуру и управление, где бы ни находились приложения. Более того, Anthos предоставляет возможность поэтапной модернизации существующих приложений без их перезаписи, что позволяет добиться немедленной экономии эксплуатационных расходов.

Большинство организаций хотят использовать мультиоблачный подход, чтобы избежать зависимости от поставщика или воспользоваться преимуществами лучших в своем классе решений. Для решения этих задач Anthos предлагает унифицированную модель управления вычислениями, сетями и даже сервисами в облаках и локально, управляемую с помощью централизованной системы управления. Anthos снижает бизнес-риски, упрощая управление несколькими облаками, и делает это реальностью для корпоративных клиентов.

Платформа Anthos состоит из нескольких ключевых сервисов:

- управление инфраструктурой,
- управление контейнерами и оркестровка,
- управление сервисами и применение политик.

Разработчики также могут использовать интегрированные сервисы для разработки и развертывания приложений в среде Anthos, а операторы могут управлять Anthos с помощью тех же инструментов, которые они используют для управления приложениями в других частях Google Cloud.

### **1.1 Единое управление с помощью Anthos**

В данном пособии рассматривается каждый уровень платформы Anthos, ее использование и возможности. Эта платформа использует открытые технологии, такие как Kubernetes, Istio и Knative, чтобы улучшить процесс разработки приложений и повысить его скорость.

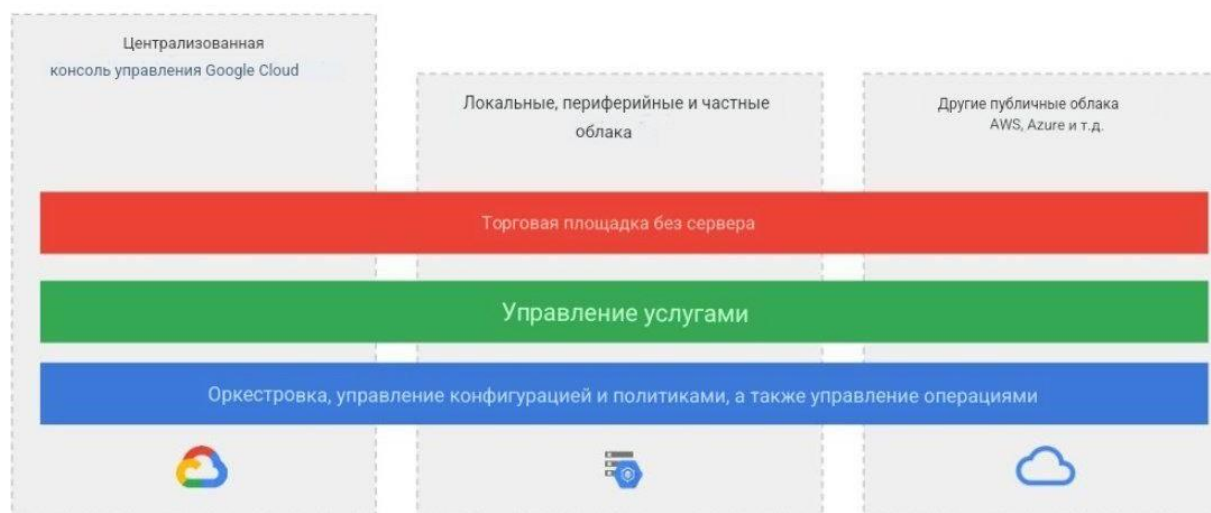


Рисунок 1 – Компоненты Anthos

На следующей диаграмме показаны компоненты Anthos и их взаимодействие в типичной корпоративной среде. (Рисунок 1)

Anthos предоставляет обширный набор возможностей для работы в мультиоблачной среде, которые можно описать следующим образом:

- Единую консоль для управления кластерами, подключенными томами и узлами;
- Функционал на основе API;
- Система настройки и управления политиками безопасности, правил RBAC и объектов Kubernetes;
- Метрики на основе журнала и пользовательские правила хранения;
- Использование проверенного кода при передаче данных;
- Идентификация с помощью Active Directory и AWS IAM;
- Использование API Kubernetes для равномерного распределения нагрузки на серверы;
- Веб-консоль для мониторинга всех нагрузок на серверах;
- Использование открытого исходного кода;
- Использование одной конечной точки API Kubernetes для подключения к кластерам;
- Автоматический сбор пользовательских метрик.

При использовании Anthos доступно легкое подключение к уже имеющимся кластерам без необходимости повторной настройки. Anthos совместим с такими сервисами, как Amazon EKS, Microsoft AKS и Red Hat OpenShift. Подключив уже существующие кластеры к консоли Anthos, получите возможность управлять ими с одной удобной панели без лишних усилий, для этого есть расширенный функционал управления кластерами:

- Управление политиками;
- Оповещения и мониторинг журнала;

- Единая точка API Kubernetes.

## **1.2 Интеграция контейнеров и управление ими с помощью Anthos GKE**

От Gmail до YouTube и поиска - все в Google работает в контейнерах. В течение последнего десятилетия использовались контейнерные рабочие нагрузки в процессе производства. Известный исходный код “k8s”, Kubernetes автоматизирует управление контейнерами, повышая надежность и сокращая время и ресурсы.

Управляемый дистрибутив Kubernetes от Google Cloud – это Google движок Kubernetes Engine (GKE), и именно этот дистрибутив является краеугольным камнем платформы Anthos. Дистрибутив — это комплект (набор файлов), приспособленный для распространения ПО.

Anthos GKE, входящая в состав Anthos, позволяет использовать преимущества Kubernetes и облачных технологий в центре обработки данных и в облаке.

По сути, Anthos GKE - это тот же GKE, который работает в Google Облако. Вы можете оставить свою среду в синхронизации с тем же Kubernetes версия OS, развернутый в локальной среде и в облаке. Также можно отслеживать, управлять и применять политикой во всех ваших GKE кластеры, как в облаке, так и на месте, из Google Cloud Приставка.

В дополнение к контейнерной оркестровке Anthos GKE предлагает дифференцированные возможности обеспечения безопасности и сетевого взаимодействия.

## **1.3 Возможности корпоративной безопасности предприятия**

Независимо от того, развертывается ли Anthos GKE в облаке или локально, он создан с учетом требований безопасности предприятия. Anthos GKE основан на Kubernetes и расширяет его возможности, предоставляя ряд важнейших функций безопасности, необходимых для запуска критически важных приложений.

### **Anthos GKE в Google Cloud**

**Безопасность уровня управления:** В Anthos GKE основные компоненты Kubernetes управляются и поддерживаются Google. Можно защитить сервер API Kubernetes, используя авторизованные сети master и частные кластеры, которые позволяют назначить главному серверу частный IP-адрес и отключить доступ по общедоступному IP-адресу.

### **Безопасность узлов**

Anthos GKE развертывает рабочие нагрузки на вычислительных серверах. Экземпляры Engine в облачном проекте Google. Каждый экземпляр использует



оптимизированную для контейнеров ОС Google в качестве операционной системы для запуска Kubernetes и его компонентов. По умолчанию используется оптимизированная для контейнеров ОС Google. Операционная система реализует заблокированный брандмауэр, файловую систему только для чтения и ограниченные учетные записи пользователей (с отключенным root). Для усиления изоляции в сценариях многопользовательского развертывания необходимо включить изолированную среду GKE в своем кластере, чтобы изолировать ненадежные рабочие нагрузки в изолированных средах на узле.

**Сетевая безопасность:** Anthos GKE использует мощную программно-определяемую сеть, которая обеспечивает простую связь между модулями в пределах Кластер Kubernetes и внутри VPC кластера. Использование сети политики можно заблокировать входные и выходные подключения, созданные к модулям в пространстве имен и из них. В рамках создания кластеров и/или пространств имен необходимо по умолчанию запретить трафик как для входа, так и для выхода из каждого модуля, чтобы гарантировать, что все новые рабочие нагрузки, добавляемые в кластер, явно разрешают трафик, который им требуется. Наконец, можно применить фильтрацию к входящему трафику с балансировкой нагрузки для служб, которым требуется внешний доступ, указав диапазоны IP-адресов CIDR из белого списка.

### **Безопасность рабочей нагрузки**

При выполнении рабочих нагрузок с использованием Anthos GKE отдельные модули и контейнеры могут быть настроены с ограниченными привилегиями с помощью контекста безопасности Kubernetes, а администраторы могут устанавливать ограничения привилегий для всего кластера с помощью политики безопасности PodSecurity. Кроме того, Anthos GKE применяет политики безопасности Docker AppArmor по умолчанию ко всем модулям Kubernetes. Наконец, идентификатор рабочей нагрузки для Anthos GKE сопоставляет учетные записи служб Kubernetes с разрешениями учетной записи облачной службы Google для управления доступом к облачным ресурсам Google из отдельных модулей. Учетные записи облачной службы Google централизованно управляются с помощью облачной идентификации и управления доступом (IAM)

### **Ведение журнала аудита**

Ведение журнала аудита позволяет администраторам сохранять, запрашивать, обрабатывать и оповещать о событиях, происходящих в Anthos среды GKE. Администраторы могут использовать регистрируемую информацию для проведения криминалистического анализа, оповещения в режиме реального времени или для каталогизации того, как работает парк оборудования. Какие кластеры Anthos GKE используются и кем. По умолчанию Anthos GKE регистрирует логику действий администратора. У вас также есть

возможность регистрировать данные. Доступ к событиям в зависимости от типов операций, которые необходимо проверить.

*Безопасность узлов:* Anthos GKE, развернутый на предварительном этапе, запускает ваши рабочие нагрузки в экземпляры VMware, которые присоединены к вашим кластерам в качестве узлов. Google использует оптимизированную версию Ubuntu Linux для запуска control plane и узлов в Anthos GKE. Этот дистрибутив был оптимизирован для облака, чтобы использовать современные стандарты, такие как автоматическое обновление ядра для системы безопасности, расширенные пакеты и ограниченный доступ пользователей.

*Ведение журнала:* аудит Kubernetes позволяет администраторам сохранять, запрашивать, обрабатывать и оповещать о событиях, происходящих в ваших Anthos среды GKE. Администраторы могут использовать регистрируемую информацию для проведения криминалистического анализа, оповещения в режиме реального времени или для каталогизации того, как работает парк оборудования. GKE clusters используется и кем.

По умолчанию Anthos GKE регистрирует действия администратора. Также можно регистрировать события доступа к данным, в зависимости от типов операций, которые вы хотите проверить. Агент Connect взаимодействует только с локальным сервером API, работающим локально, и каждый кластер должен иметь свой собственный набор журналов аудита. Все действия, которые пользователи выполняют из пользовательского интерфейса через Connect, регистрируются этим кластером.

*Шифрование:* Google Cloud Key Management Service (Cloud KMS) - это облачная служба управления ключами, которая позволяет управлять криптографическими ключами имеющихся сервисов. Можно генерировать, использовать, заменять и уничтожать криптографические ключи AES256, RSA 2048, RSA 3072, RSA 4096, EC P256 и EC P384. Cloud KMS интегрирован с Cloud IAM и журналами облачного аудита, что позволяет управлять разрешениями для отдельных ключей и отслеживать их использование. Можно использовать Cloud KMS для защиты секретов и других конфиденциальных данных, которые вам необходимо сохранить.

*Подключение:* Если кластеры Anthos GKE, развернутые на месте, и рабочие нагрузки надежно подключаются к облачным сервисам Google Cloud services через облако VPN, можно использовать Cloud KMS для управления ключами. В противном случае необходимо использовать альтернативные решения, такие как Kubernetes Secrets, HashiCorp Vault или аппаратный модуль безопасности.

## 1.4 Сетевая интеграция платформы

Anthos GKE использует мощную программно-определяемую сеть, которая обеспечивает простую связь между модулями в кластере Kubernetes и в VPC кластера

Независимо от того, развернут ли Anthos GKE в облаке или локально, он настраивается "из коробки" с рядом улучшений в сети, которые расширяют возможности Сетевые примитивы Kubernetes с интеграцией в конкретную среду.

### Anthos GKE в Google Cloud

Развертывания Anthos GKE, работающие в Google Cloud, позволяют использовать расширенную программно-определяемую сеть Google (SDN), которая обеспечивает маршрутизацию и пересылку пакетов для модулей, служб и узлов в разных зонах одного регионального кластера. Anthos GKE также способен динамически настраивать правила IP-фильтрации, таблицы маршрутизации и правила брандмауэра на каждом узле Kubernetes

Каждому узлу Anthos GKE присваивается IP-адрес из сети кластера Виртуальное частное облако (VPC). Этот IP-адрес узла обеспечивает подключение компонентов управления, таких как kube-proxy и kubelet, к серверу API Kubernetes. Этот IP-адрес является подключением узла к остальной части кластера. У каждого узла есть пул IP-адресов, который назначается модулям, работающим на этом узле. Каждому модулям присваивается один IP - адрес из диапазона IP-адресов его узла. Этот IP-адрес используется совместно всеми контейнерами, работающими в модуле, и соединяет их с другими модулями, работающими в кластере. Каждой службе присваивается IP-адрес, называемый ClusterIP, из сети VPC кластера. Можно дополнительно настроить сеть VPC при создании кластера.

Антос ГКЭ обеспечивает три различных вида нагрузки: облако балансировки, балансировщики нагрузки для контроля и управления доступом и распространения входящего трафика. Можно настроить один Сервис и использовать несколько видов балансировки нагрузки одновременно.

*Anthos GKE предоставляет три различных типа балансировщиков нагрузки для управления доступом и максимально равномерного распределения входящего трафика по существующему кластеру. (Рисунок 2)*

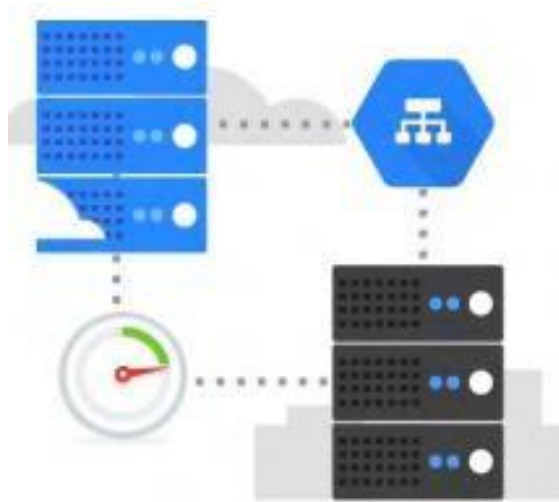


Рисунок 2 – Три различных типа балансировщиков нагрузки

- Внешние подсистемы балансировки нагрузки управляют трафиком, поступающим из - за пределов кластера и вашего виртуального компьютера Google Cloud. Они используют правила переадресации, связанные с облачной сетью Google, для маршрутизации трафика на узел Kubernetes.
- Внутренние подсистемы балансировки нагрузки управляют трафиком, поступающим из той же сети VPC. Как и внешние подсистемы балансировки нагрузки, они используют правила переадресации, связанные с облачной сетью Google, для маршрутизации трафика на узел Kubernetes.
- Протокол HTTP(S) балансировщиков нагрузки, специализированных внешней нагрузки балансировщиков для протокола HTTP(S). Они используют попадания ресурс скорее, чем правило пересылки для маршрутизации трафика к узлу Kubernetes.

Когда трафик достигает узла Anthos GKE, он обрабатывается одинаково, независимо от типа подсистемы балансировки нагрузки. Подсистема балансировки нагрузки не знает, на каких узлах в кластере запущены модули для его обслуживания. Вместо этого он балансирует трафик на всех узлах кластера, даже на тех, на которых не запущен соответствующий модуль. В региональном кластере нагрузка распределяется по всем узлам во всех зонах региона кластера. Когда трафик направляется к узлу, узел направляет трафик в модуль, который может быть запущен на том же узле или на другом узле

### **Автономный Anthos GKE**

Anthos GKE, развернутый локально, использует конфигурацию изолированного режима, в которой модули могут напрямую взаимодействовать друг с другом в кластере, но не могут быть доступны извне кластера. Эта конфигурация формирует "остров" в сети, который не подключен к внешней сети. Кластеры образуют полную сетку от узла к узлу между узлами кластера, позволяя модулю напрямую связываться с другими модулями внутри кластера.

Весь исходящий трафик от модуля к целевым объектам за пределами кластера поступает из модуля с использованием IP-адреса главного узла. Предварительно развернутый Anthos GKE включает в себя балансировщик нагрузки L7 с входным контроллером на базе Envoy, который обрабатывает правила входных объектов для служб ClusterIP, развернутых в кластере. Сам входной контроллер доступен как служба NodePort в кластере.

*Anthos GKE развернутый локально, использует конфигурацию островного режима, в которой модули могут напрямую взаимодействовать друг с другом внутри кластера, но не могут быть доступны извне кластера.*

Предварительно развернутый Anthos GKE включает встроенный балансировщик нагрузки L4, а также обеспечивает поддержку внешних балансировщиков нагрузки L3/L4 для сетей F5. В процессе установки в подсистеме балансировки нагрузки настраивается виртуальный IP-адрес (VIP) (с портами 80 и 443). VIP указывает на порты в службе NodePort для входного контроллера. Таким образом, внешние клиенты могут получать доступ к службам в кластере. При использовании внешнего балансировщика нагрузки F5 кластеры пользователей, в которых запущены службы с типом LoadBalancer, должны настроить поле IP LoadBalancer для использования настроенного выше VIP.

В качестве альтернативы использованию встроенных средств балансировки нагрузки или F5, можно включить режим ручной балансировки нагрузки. Если необходимо использовать ручную балансировку нагрузки, то невозможно запускать службы типа LoadBalancer. Вместо этого можно создать службы типа Node Port и вручную настроить свой балансировщик нагрузки для использования их в качестве серверных компонентов. Кроме того, можно предоставлять доступ к службам внешним клиентам с помощью объекта Ingress.

## **1.5 Anthos GKE для управления корпоративными рабочими нагрузками**

Anthos GKE включает в себя ряд улучшений в области сетевого взаимодействия и безопасности для поддержки критически важных корпоративных рабочих нагрузок. Anthos GKE основан на примитивах Kubernetes и интегрирует их с облачными системами Google, такими как Cloud IAM. Независимо от того, развернут ли Anthos GKE в облаке или на месте, он использует опыт Google в управлении крупномасштабными распределенными системами. Добавив к этому первое в отрасли четырехстороннее автоматическое масштабирование, соглашение об уровне обслуживания при финансовой поддержке и гибкие каналы выпуска, и Anthos GK станет основой, необходимой для создания высоконадежных сервисов.

## **Тема 2 Настройка политики с помощью Anthos Config Management**

Поскольку число кластеров Anthos GKE начинает расти, внесение индивидуальных изменений в конфигурацию для каждого развертывания приводит к накладным расходам и проблемам с управлением. Управление конфигурацией Anthos является ключевым компонентом стека Anthos и предоставляет операторам платформы, сервиса и безопасности единый, унифицированный подход к управлению несколькими кластерами, который охватывает как локальные, так и облачные среды. В частности, Anthos Config Management позволяет операторам создавать и применять общие конфигурации и политики в масштабе всей вашей системы Anthos GKE кластеры. Давайте подробнее рассмотрим, как мы спроектировали систему управления конфигурацией Anthos, и какие преимущества она дает

### **2.1 Архитектура управления конфигурацией Anthos**

Управление конфигурацией Anthos основано на современных программных инструментах и практиках, используя центральный репозиторий Git для управления политиками контроля доступа, такими как управление доступом на основе ролей (RBAC), квотами ресурсов и пространствами имен в разных средах. Поскольку изменения конфигурации и политики передаются в репозиторий Git, Anthos Config Management оценивает эти изменения и распространяет их на кластеры Anthos GKE в ваших развертываниях. Anthos Config следует лучшим практикам Kubernetes, отдавая предпочтение декларативным подходам, а не императивным операциям, и постоянно отслеживает состояние кластера и применяет желаемое состояние, как определено в Git. (Рисунок 3)

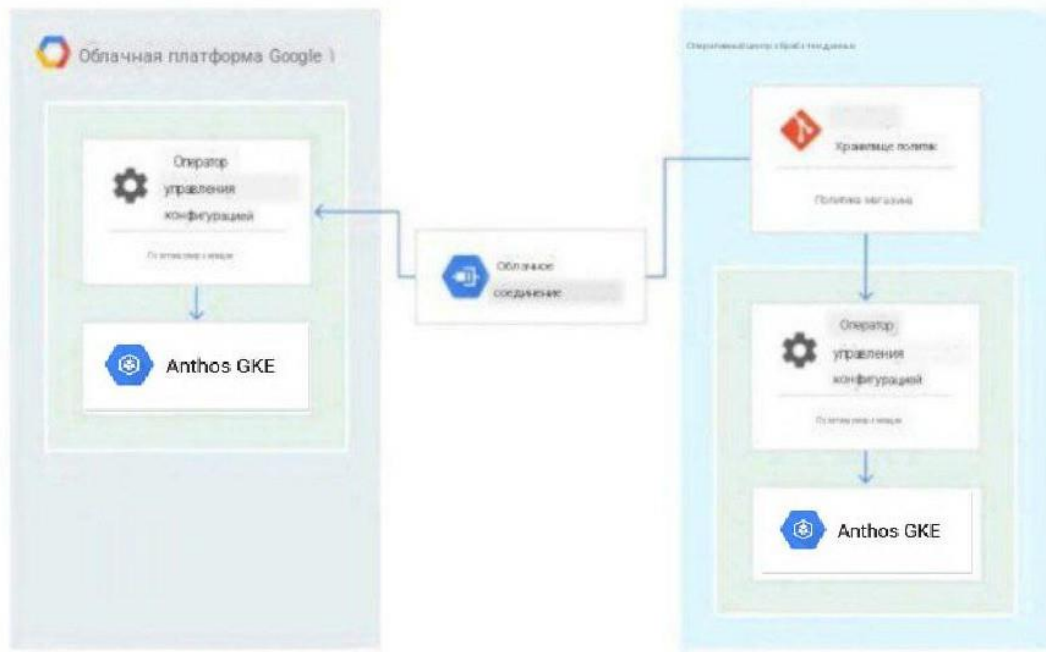


Рисунок 3 – Архитектура управления конфигурацией Anthos

Как показано на приведенной выше диаграмме, Anthos Config Management развертывается как пользовательский контроллер в каждом из ваших кластеров Anthos GKE и включает в себя три ключевых компонента:

1. Импортёр, который считывает данные из центрального репозитория Git;
2. Компонент для синхронизации и переноса сохраненных данных конфигурации в объекты Kubernetes;
3. Компонент, который отслеживает различия между сохраненными и активными конфигурациями кластера и согласовывает их по мере необходимости.

Центральное хранилище конфигурации и политик Git может быть размещено локально, в Google Cloud или с помощью любого размещенного поставщика Git (например, GitLab или Github) - единственное требование заключается в том, чтобы компонент импортера имел сетевое подключение к хранилищу Git.

### **Мастерство в управлении конфигурацией с Anthos**

Anthos Config Management позволяет администратору кластера следовать современным методам разработки программного обеспечения, делая изменения конфигурации кластера и политики доступными для аудита, возврата и управления версиями. Это также модернизирует методы управления конфигурацией, включив несколько ключевых функций и возможностей, присущих Kubernetes. Используя примитивы Kubernetes, Anthos Config

Управление может гибко применять различные конфигурации к группам кластеров или пространствам имен — например, применять различные уровни квот для промежуточных и производственных ресурсов.

### **Обеспечение эффективного управления ИТ**

Помимо настройки, Anthos Config Management также включает в себя поддержку определения и применения пользовательских правил, не предусмотренных собственными объектами Kubernetes. Механизм контроллера политик управления конфигурацией Anthos позволяет создавать ограждения, соответствующие уникальным требованиям вашей организации к безопасности, соответствию требованиям и управлению. Эти ограждения позволяют просматривать обновления на компьютере. Обновляйте инфраструктуру и отклоняйте изменения, которые не соответствуют вашим требованиям.

*Конфигурация Anthos механизм контроллера политики управления позволяет создавать ограждения, соответствующие уникальным требованиям вашей организации к безопасности, соблюдению нормативных требований и управлению.*

Уникальные политики. Например, в организации может потребоваться, чтобы приложения использовали определенные сетевые конфигурации или механизмы хранения. С помощью контроллера политик управления конфигурацией Anthos может помочь новым командам быстро приступить к работе, зная, что их приложения соответствуют рекомендациям по соблюдению требований.

Контроллер политики управления конфигурацией Anthos разворачивается в качестве дополнительного компонента в ваших кластерах Anthos GKE и основан на проекте Open Policy Agent Gatekeeper. Конфигурация Anthos Контроллер политики управления работает как веб-интерфейс контроллера доступа Kubernetes, что позволяет ему прозрачно интегрироваться с Kubernetes API и оценивать объекты Kubernetes по мере их допуска в кластер. Он усиливает защитные барьеры, отказывая в допуске или проверяя объекты, нарушающие предопределенные ограничения, предоставляя операторам безопасности централизованный способ контроля за соблюдением требований. Ограждения создаются с использованием двух объектов: правила ограничения определены в шаблоне с кластерной областью действия, а применение этого ограничения ограничено определенным пространством имен Kubernetes и/или определенными типами объектов Kubernetes. Например, можно создать ограждение, которое обеспечивает соблюдение определенных правил маркировки для всех контейнеров в пространстве имен “по умолчанию

#### *Унифицированное управление ресурсами*

Многие облачные команды разработчиков работают с различными системами настройки, API-интерфейсами и инструментами для управления своей инфраструктурой. Это сочетание часто бывает сложным для понимания,



что приводит к снижению скорости работы и дорогостоящим ошибкам. В дополнение к обеспечению управления ИТ с помощью контроллера политик, Anthos Config Management поддерживает Config Connector, который предоставляет метод настройки многих облачных сервисов и ресурсов Google, таких как виртуальные машины Compute Engine или облако для обмена публичными сообщениями с использованием инструментов Kubernetes и API-интерфейсов.

Config Connector предоставляет набор пользовательских настроек Kubernetes Определения ресурсов (CRD) и связанных с ними контроллеров. Конфигурация Connector создает облачные ресурсы Google, когда настраиваете и применяете пользовательские объекты к своему кластеру. (Рисунок 4)

Config Connector - это система, основанная на Kubernetes, что означает, что она может использовать общие лучшие практики Kubernetes, такие как хранение и использование конфиденциальных данных с использованием секретов, управление конфигурацией среды выполнения с использованием объектов ConfigMap и соблюдение стандартных подходов к управлению доступом на основе ролей. Кроме того, в сочетании с Anthos Config контроллер политики управления, правила управления ИТ могут применяться для создания и текущего управления ресурсами, управляемыми Config Connector.

Являясь ключевым компонентом стека Anthos, Anthos Config Management обеспечивает современный подход к управлению конфигурацией и политиками и позволяет операторам платформ, служб и систем безопасности использовать единый подход к управлению несколькими кластерами в гибридных развертываниях. Anthos Config Management позволяет управлять конфигурацией и политиками для Anthos GKE, а также для сервиса Anthos Mesh и Cloud Run для Anthos.



Рисунок 4 – Схема управления ресурсами

## 2.2 Мониторинг и управление услугами с помощью Anthos Service Mesh

Все более популярным подходом к модернизации приложений становится разбиение больших “монолитных” приложений, написанных в виде единого логически исполняемого файла, на независимые микросервисы, где функциональность разбивается на более мелкие независимые сервисы, взаимодействующие через API.

Но по мере того, как число развертываний таких микросервисов растет, часто резко возрастают усилия, необходимые для их эксплуатации и масштабирования. Anthos Service Mesh предоставляет набор инструментов, которые помогают отслеживать и управлять сервисами всех форм и размеров, независимо от того, работают ли они в облачных, гибридных или мультиоблачных средах.

### 2.3 Архитектура сервисной сетки Anthos Service Mesh

Anthos Service Mesh использует API-интерфейсы и основные компоненты из Istio, легко настраиваемой платформы service mesh с открытым исходным кодом, и опирается на них с помощью полностью управляемых механизмов обеспечения работоспособности служб, оперативной гибкости и безопасности.

Как и другие платформы service mesh, Anthos Service Mesh основана на двух основных компонентах: плоскости данных и плоскости управления. В Anthos Service Mesh при сетевом развертывании уровень данных развертывается как набор распределенных прокси-серверов, которые обеспечивают передачу всего входящего и исходящего сетевого трафика между отдельными службами. Сами прокси-серверы настраиваются с использованием уровня централизованного управления и открытого API. Такой подход позволяет более широко автоматизировать общие сетевые задачи, такие как разделение трафика или управление им между службами, а также поддержка аутентификации и шифрования "от службы к службе".



Рисунок 5 – Платформа управления Anthos Service Mesh

При использовании Anthos Service Mesh уровень управления работает как полностью управляемое решение за пределами кластеров Anthos GKE, что упрощает управление и обеспечивает максимально возможную доступность.

Управляемая платформа управления Anthos Service Mesh, между тем, состоит из трех компонентов. (Рисунок 5)

- Первый компонент, Traffic Director, полностью управляемая платформа управления трафиком Google Cloud service mesh, отвечает за перевод Istio Объекты API преобразуют информацию о конфигурации распределенных прокси-серверов, а также направляют входящий и исходящий трафик сервисной сети.

- Второй, управляемый центр сертификации, является централизованным центром сертификации, ответственным за предоставление SSL - сертификатов каждому из распределенных прокси-серверов, аутентификационную информацию и распространение секретов.

- Последним компонентом является инструментарий Google Cloud operations tooling (ранее Stackdriver), который обеспечивает управляемую точку доступа для наблюдения и телеметрии, в частности, для мониторинга, отслеживания и регистрации данных, генерируемых каждым из прокси-серверов. Кроме того, этот инструментарий поддерживает панель мониторинга наблюдаемости Anthos Service Mesh, которая позволяет операторам сервиса визуально проверять свои службы и зависимости от служб, а также внедрять лучшие практики SRE для мониторинга SLI и установления SLO.

До внедрения технологий service mesh от разработчиков приложений требовалось обеспечить поддержку общих функций управления, создания сетей или обеспечения безопасности, таких как средства проверки подлинности/авторизации между сервисами, шифрование служебных сообщений или детальное управление трафиком. С помощью сервисных сетей, и в частности Istio, эта работа переносится с приложений на распределенные прокси-серверы, которые формируют плоскость данных.

Anthos Service Mesh использует API-интерфейсы Istio для предоставления следующих функций для служб, развернутых в Anthos GKE, или для смешанных развертываний с сервисами на базе контейнеров и виртуальных машин (облачных или локальных).

В совокупности инструменты и технологии, входящие в состав Anthos Service Mesh, обеспечивают значительные эксплуатационные преимущества для сред Anthos с минимальными дополнительными накладными расходами.

#### *Единообразная наблюдаемость*

Как упоминалось ранее, распределенный прокси-сервер data plane в Anthos Service Mesh отвечает за посредничество во всех входящих и исходящих сообщениях от каждой развернутой службы. В то время как прокси-сервер передает данные о взаимодействии между сервисами обратно на уровень управления, чтобы можно было создать график зависимости от сервиса.

Прокси-сервер также проверяет трафик и вставляет заголовки для облегчения распределенной трассировки. Наконец, он собирает и отображает журналы обслуживания и показатели уровня обслуживания (например, задержки, ошибки, доступность). Все это достигается без необходимости интеграции развернутыми приложениями пользовательских подходов к метрикам, трассировке или сбору данных журнала.

Для приложений в средах с пользовательской интеграцией средств наблюдения данные, полученные с помощью прокси-серверов, также могут быть переданы в сторонние системы наблюдения.

### *Оперативная гибкость*

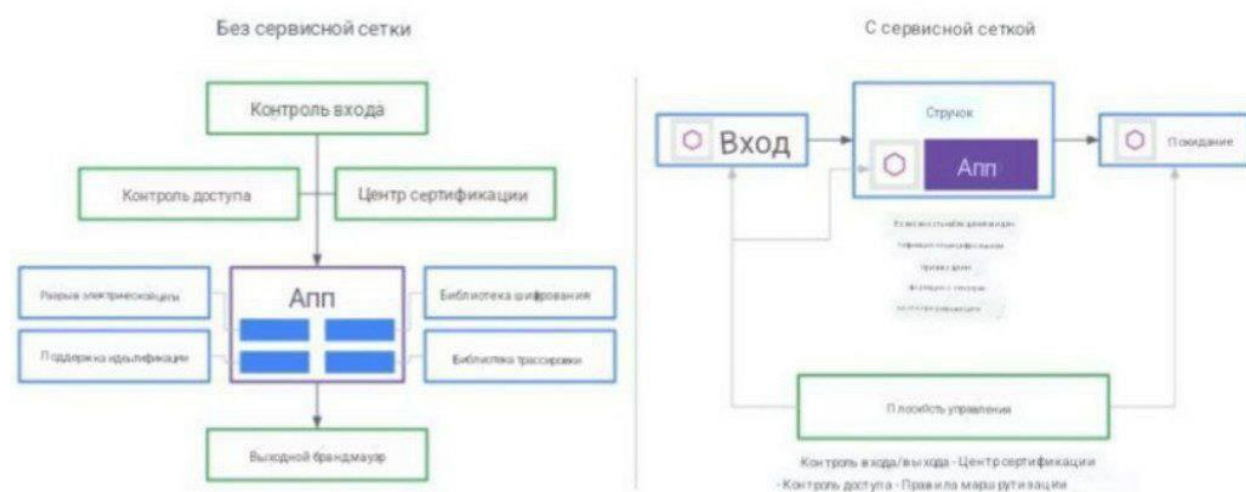


Рисунок 6 – Архитектура сервисной сетки Anthos Service Mesh

Сервисная сетка Anthos обеспечивает детальное управление потоками трафика между сетями (север-юг) и внутри сети (восток-запад). (Рисунок 6) Средства управления трафиком также могут быть интегрированы с механизмами входного и выходного трафика, предоставляя разработчикам и операторам сервисов полный контроль над тем, как трафик поступает в их сервисную сеть, выходит из нее и проходит через нее. Эти средства управления позволяют разработчикам и операторам внедрять такие средства управления трафиком, как:

- Разделение трафика между различными версиями служб для использования в canary или A/B-тестировании
- Управление трафиком на основе HTTP-заголовков между отдельными службами или версиями
- Разрыв цепи для предотвращения каскадных сбоев
- Внедрение ошибок для создания устойчивых и отказоустойчивых развертываний

Управление трафиком осуществляется с помощью объектов API Istio, а распределенные прокси-серверы настраиваются с помощью Traffic Director. По умолчанию каждый прокси-сервер хранит информацию обо всех возможных

входящих и исходящих маршрутах, а также о маршрутизируемых вышестоящих узлах и службах в сервисной сети. Traffic Director периодически предоставляет данные конфигурации каждому из прокси-серверов, чтобы убедиться, что они находятся в актуальном состоянии и осведомлены о других сервисах в сети.

*Безопасность, основанная на политике*

Цель Anthos Service Mesh - предоставить сервисам средства контроля безопасности "под ключ", уменьшив необходимость в интеграции с конкретными сервисами. Anthos Service Mesh обеспечивает базовый защищенный канал связи и управляет аутентификацией, авторизацией и шифрованием служебных сообщений. С помощью Anthos Service Mesh служебные коммуникации защищены по умолчанию, что позволяет обеспечить политики, согласованные с различными протоколами и средами выполнения. Как и в случае с трафиком и функциями наблюдения, распределенные прокси-серверы отвечают за реализацию каждой из функций безопасности Anthos Service Mesh

В Anthos Service Mesh идентификация является фундаментальным компонентом — службы обмениваются идентификационными данными для взаимной аутентификации. На стороне клиента идентификация сервера сверяется с информацией о безопасном именовании, чтобы определить, авторизован ли он для запуска службы. На стороне сервера сервер определяет, к какой информации клиент может получить доступ на основе политик авторизации, и проверяет использование сервисов. Эта модель идентификации обеспечивает значительную гибкость и детализацию для Anthos Service Mesh для понимания идентификационных данных конечного пользователя, отдельных сервисов или нескольких сервисов

В сочетании с сервисной идентификацией, Anthos Service Mesh managed CA предоставляет доверенный корневой центр сертификации, который распределяет ключи и сертификаты, необходимые для работы инфраструктуры открытых ключей Anthos Service Mesh, по каждому прокси-серверу. Управляемый центр сертификации также отвечает за смену ключей и сертификатов для каждого прокси-сервера до истечения срока их действия. Используя эту инфраструктуру, службы в Anthos Service Mesh могут использовать транспортную или исходную аутентификацию.

Аутентификация источника позволяет службам на основе mesh проверять личность конечного пользователя (или устройства) и может быть включена на уровне индивидуального запроса. Транспортная аутентификация предназначена для обмена данными между службами, которая осуществляется по туннелю через взаимно аутентифицированный сервер. Подключение по протоколу TLS между прокси-серверами позволяет создать зашифрованный канал между службами. Для развертываний, в которых некоторые службы могут не входить

в сеть служб, аутентификация по протоколу mTLS может вводиться поэтапно, чтобы обеспечить постепенный процесс адаптации.

Anthos Service Mesh также предоставляет механизмы контроля доступа, которые управляются с помощью детализированных политик авторизации. Эти политики поддерживают управление доступом на уровне сети, пространства имен и рабочей нагрузки для служб в рамках сети. Каждый прокси-сервер применяет меры контроля доступа до того, как трафик попадет в службу назначения. Когда создаются политики авторизации, они содержат подробную информацию о том, какие службы затронуты, и правила, которые определяют, кому (список источников) разрешено выполнять то, что (список операций), при каких условиях. Источники основаны на основе идентификатора службы или конечного пользователя. Список разрешенных операций содержит HTTP-методы и URL-адреса. Доступные условия, которые могут быть включены в политики авторизации, позволяют проверять метаданные отдельных запросов или сетевые атрибуты источника/получателя (например, IP-адрес, порты).

Следовательно, сервисная сеть Anthos является важнейшим компонентом общей Платформа Anthos обеспечивает функциональность, дополняющую базовый уровень оркестрации контейнеров Kubernetes, и обеспечивает гибкость, наблюдаемость и безопасность, необходимые для создания современных приложений на основе распределенных контейнерных микросервисов.

## Тема 3 Бессерверная работа с облачным управлением для Anthos

Используя бессерверные системы, часто имеют в виду функции как услугу или рабочие нагрузки в стиле AWS Lambda. В Google облаке рассматривается бессерверная среда более широко: рабочие нагрузки, выполняемые в управляемой инфраструктуре, где разработчики самостоятельно удовлетворяют свои потребности. Например, Google App Engine уже более десяти лет предлагает разработчикам по всему миру бессерверную платформу в качестве сервиса.

С новой платформой Cloud Run предлагаем контейнерам возможность работать без сервера. Любое HTTP-приложение, работающее в виде контейнера, может быть запущено в полностью управляемой инфраструктуре Google. Google занимается подключением к сети, автомасштабированием, доменными именами, протоколами TLS и многими другими аспектами, поэтому не нужно беспокоиться о настройке и управлении такими вещами, как виртуальные машины, кластеры или балансировщики нагрузки.

Бессерверные решения, такие как Cloud Run, позволяют командам разработчиков гибко удовлетворять свои собственные потребности. Например, с помощью модели самообслуживания команда специалистов по обработке данных может развертывать свои модели прогнозирования машинного обучения и автоматически масштабировать их, не обременяя при этом свои операционные подразделения или команды, работающие с платформами.

В этом разделе рассмотрим, как Cloud Run для Anthos позволяет работать с разработчиками на высоком уровне и изолировать ваши команды разработчиков от базовой инфраструктуры, одновременно упрощая операции по запуску служб для ваших операционных групп. Но сначала давайте рассмотрим некоторые проблемы, связанные с запуском современных приложений в Kubernetes.

*Облачный сервис для Anthos позволяет работать с разработчиками на высоком уровне и изолировать свои команды разработчиков от базовой инфраструктуры, одновременно упрощая операции по запуску сервисов для ваших операционных групп.*

### 3.1 Сложность работы с микросервисами в Kubernetes

Kubernetes отлично справляется с управлением набором рабочих нагрузок на множестве машин с помощью декларативной модели, управляемой состоянием.

Например, что касается Kubernetes, то микросервисное приложение - это просто контейнер с несколькими TCP-портами. Следовательно, Kubernetes не делает все возможное для упрощения работы с современными приложениями, основанными на двенадцатифакторной методологии разработки приложений.

Например, Kubernetes изначально не поддерживает “версии приложения” и, следовательно, не предлагает таких высокоуровневых функций, как быстрое развертывание или чистый откат.

Аналогичным образом, Kubernetes для создания сетей и балансировки нагрузки между службами не использует сетевые протоколы прикладного уровня (уровень 7) , такие как HTTP или gRPC, поэтому он не может распределять трафик между версиями, применять политики трафика или автоматически масштабировать приложения на основе показателей запросов

Автоматическое масштабирование приложений на основе микросервисов также является нетривиальной задачей в Kubernetes. Скачкообразные шаблоны трафика могут привести к падению контейнеров и отбрасыванию запросов, поскольку его горизонтальный модуль автоматического масштабирования (HPA) работает только с текущими средними показателями процессора и памяти, которые являются результатом шаблона трафика, и, таким образом, не могут надежно предотвратить сбой контейнера в случае скачка трафика. Также Kubernetes не предлагает способ буферизации запросов до тех пор, пока они не будут обработаны доступным контейнером.

Именно в этом преимущество бессерверных платформ: предоставьте им свой модуль развертывания (функцию, приложение или образ контейнера), и инфраструктура запустит и масштабирует приложение за вас.

### **3.2 Бессерверный сервер в Kubernetes: открытый и расширяемый**

Cloud Run for Anthos предоставляет кластерам Anthos возможности использования бессерверных контейнеров. Cloud Run for Anthos предлагает высокоуровневую платформу на базе кластеров и строительных блоков Kubernetes, позволяя командам разработчиков создавать собственные платформы на Kubernetes.

Cloud Run для Anthos создан с помощью Knative, оператора с открытым исходным кодом для Kubernetes, который предоставляет кластеру возможности для обслуживания приложений без сервера и проведения мероприятий. Изначально Knative был создан Google при участии более 50 различных компаний.

Knative добавляет недостающие элементы высокого уровня в ваши кластеры Kubernetes, обеспечивая при этом совместимость с другими компонентами Anthos и инструменты Kubernetes. В конце концов, рабочие нагрузки Knative по-прежнему остаются рабочими нагрузками Kubernetes.

### **3.3 Операции Cloud Run для Anthos**

На площадке команд, которые хотят застройки предлагают дополнительные инструменты для тестирования, развертывания и запуска приложений, Knative предоставляет простой способ предоставить



дополнительные возможности для разработчиков поверх существующих Kubernetes-кластеров.

Рассмотрим преимущества, которые Cloud Run for Anthos предоставляет вашим средам Kubernetes.

### **Простая миграция из развертываний Kubernetes**

Чтобы запустить микросервис в Kubernetes, необходимо настроить объекты Deployment, Service и HorizontalPodAutoscaler для балансировки нагрузки и автоматического масштабирования. Кроме того, невозможно легко изменить или откатить эти конфигурации, если приложение уже обслуживает трафик.

С Cloud Run для Anthos не нужно заранее настраивать эти функции, а откат выполняется легко. А переход на Cloud Run для Anthos прост: просто возьмите свое развертывание Kubernetes, измените несколько строк кода, удалите объекты HPA и Kubernetes Service, и будет манифест службы Knative, описывающий приложение для микросервисов, которое автоматически масштабируется и балансирует нагрузку.

#### *Автоматическое масштабирование*

Как описывалось ранее, Kubernetes HPA работает на основе показателей процессора и памяти, которые часто слишком медленны и запаздывают, чтобы реагировать на потребности микросервисов spiky. Внезапный скачок трафика может привести к сбою работы ваших контейнеров приложений в Kubernetes (и отбрасыванию запросов) поскольку они будут перегружены при попытке обслуживать большой объем трафика.

Cloud Run for Anthos предлагает три высокоуровневых готовых примитива автоматического масштабирования для ваших приложений, которых изначально нет в Kubernetes:

- Быстрое автоматическое масштабирование на основе запросов: По умолчанию все приложения Knative оснащены функцией автоматического масштабирования, которая отслеживает показатели запросов. Это позволяет Cloud Run for Anthos легко справляться со скачкообразным трафиком.

- Управление параллелизмом: Knative позволяет вам устанавливать ограничения на параллелизм (т.е. максимальное количество запросов в полете на контейнер), что гарантирует, что контейнер не будет перегружен и не выйдет из строя. Запросы буферизуются до тех пор, пока не будут добавлены дополнительные контейнеры для обработки скачкообразного трафика.

- Масштабирование до нуля: если приложение какое-то время не получает запросов или полностью неактивно, Cloud Run масштабирует его до нуля, чтобы уменьшить нагрузку на ваш кластер. Затем первый запрос, поступающий в приложение, ожидает, пока не будет создан контейнер для обработки запроса. Кроме того, можно отключить масштабирование до нуля, чтобы предотвратить холодный запуск.

### *Сеть*

По умолчанию в Kubernetes отсутствуют сетевые протоколы прикладного уровня, такие как HTTP или gRPC. Это часто приводит к неравномерному распределению нагрузки между репликами в процессе развертывания, поскольку Kubernetes поддерживает только балансировку нагрузки по протоколу TCP.

Cloud Run for Anthos обладает встроенными возможностями балансировки нагрузки и политиками для разделения трафика между несколькими версиями приложения. А поскольку он обладает глубоким пониманием запросов, он может буферизовать каждый запрос во время автоматического масштабирования и собирать показатели на уровне запросов из приложений "из коробки".

Cloud Run для Anthos совместим с Anthos Service Mesh, и любое приложение, развернутое на Knative, будет использовать возможности service mesh, такие как политика трафика, взаимный протокол TLS и телеметрия.

#### *Выпуски и развертывания*

Облачный запуск для Anthos также поддерживает концепцию API Knative Изменения, которые описывают новые версии или различные конфигурации вашего приложения. Например, изменение образа контейнера или увеличение объема памяти приводит к созданию новых ревизий.

Изменения являются неизменяемыми, следовательно, они позволяют вам полностью вернуться к предыдущей конфигурации. В Knative также есть концепция привязки к ссылкам контейнера <image:tag>, так что при повторном вставлении нового изображения в существующий тег ваше приложение Knative не получит новое изображение. Это гарантирует, что первоначальные развертывания воспроизводимы и могут быть легко откатаны.

Кроме того, можно выполнять развертывание canary, разделяя трафик для приложения, используя высокоуровневую конфигурацию трафика, например, отправляя 90% трафика на Rev1 и 10% на Rev2. Отправляя лишь небольшой процент трафика на новую версию вашего приложения, вы можете постепенно тестировать влияние новых функций или изменений на ваших пользователей. (Рисунок 7)



Рисунок 7 - Выпуски и развертывания

### *Мониторинг*

Cloud Run для Anthos отслеживает все запросы, поступающие к приложениям. Таким образом, он может отслеживать и записывать показатели golden star, такие как задержка, частота ошибок и количество запросов в секунду.

Эти показатели автоматически собираются и отправляются в Google Cloud monitoring and operations tools без какой-либо настройки, и они могут помочь устранить неполадки и выявить несоответствия между различными версиями вашего приложения. Например, отслеживать, измерять и наблюдать за внутренней целью уровня обслуживания (SLO), такой как задержка запроса для службы составляет 99,5%, и видеть, как она ведет себя в разных версиях вашего приложения.

### *Облачный запуск для вариантов использования Anthos*

Cloud Run for Anthos подходит для запуска приложений без сохранения состояния, поскольку не поддерживает приложения с отслеживанием состояния или зависающие сеансы. Некоторые примеры приложений, которые отлично работают в Cloud Run для Anthos, включают:

- Микросервисы, веб-интерфейсы, шлюзы API, промежуточное программное обеспечение API.
- Обработчики событий, ETL: когда приложение считывает передаваемую ему полезную информацию о событии и обрабатывает ее.
- Прогнозирование модели машинного обучения: например, TensorFlow, обслуживающий контейнеры.

*В целом, Cloud Run для Anthos упрощает операции и управление приложениями в сервис-ориентированной архитектуре (SOA) или архитектуре микросервисов*

Облачный запуск для Anthos имеет встроенные возможности балансировки нагрузки и политики для разделения трафика между несколькими версиями приложения.

## Тема 4 Разработка приложений для Anthos

Разработка приложений для Kubernetes часто означает необходимость интеграции различных инструментальных цепочек и создания систем на начальном этапе, а также механизмов обеспечения безопасности и развертывания. Интеграция этих систем может быть сложной. Кроме того, результирующие системы могут быть сложными и хрупкими, подверженными взлому. Для создания приложений на Anthos Google Cloud предлагает хорошо интегрированный набор инструментов, помогающих ускорить процесс разработки.

### 4.1 Кодирование приложений для Kubernetes

Разработка приложений начинается с кода, который проходит постоянный цикл написания, запуска и отладки. Чтобы облегчить разработку современных приложений на базе Kubernetes, Google создала Cloud Code - набор инструментов, которые помогут вам в написании, запуске и отладке. Облачный код доступен в виде расширений для популярных интегрированных сред разработки (IDE), таких как Visual Studio Code и IntelliJ, обеспечивая встроенную поддержку быстрой итерации, отладки и запуска приложений в средах разработки и производства Kubernetes

На самом деле, облако код поддерживает полный цикл разработки по Kubernetes-приложений, начиная от создания кластера для разработки и тестирования до запуска готового приложения - все из основного разработчика интегрированной среды разработки. В него также входят готовые к запуску образцы, готовые фрагменты конфигурации и индивидуальный подход к отладке. Для локальной разработки в Cloud Code используются такие инструменты, как Skaffold, Jib и kubectl, которые автоматизируют задачи разработки и обеспечивают постоянную обратную связь. А для производственных развертываний - облако Code предоставляет готовые профили Skaffold и использует Kustomize для управления ресурсами, зависящими от среды. Наконец, при отладке приложений Cloud Code позволяет разработчикам использовать встроенные в IDE средства отладки и просматривать журналы приложений, независимо от того, выполняются ли приложения локально или удаленно.

*Cloud Code поддерживает полный цикл разработки Приложений Kubernetes - от создания кластера для разработки и тестирования до запуска готовых приложений.*

### 4.2 Создание артефактов сборки

Помимо помощи разработчикам в создании приложений на базе Kubernetes с облачным кодом, Anthos обеспечивает прямую интеграцию для создания приложений и упаковки их в образы контейнеров с помощью Cloud

Build for Anthos - системы, которая выполняет сборку программного обеспечения в средах Anthos. Cloud Build позволяет импортировать исходный код из Google Cloud Storage, облачных хранилищ исходных текстов, GitHub или GitLab, выполнять сборку в соответствии с вашими требованиями и создавать артефакты, такие как контейнеры Docker или архивы Java.

Облачная сборка выполняет сборку в виде последовательности шагов, каждый из которых выполняется в контейнере Docker. Шаг сборки может выполнять все, что можно сделать из контейнера, независимо от среды. Для выполнения рабочих процессов сборки можно либо использовать поддерживаемые шаги сборки, предоставляемые Cloud Build, либо написать собственные шаги сборки. Изображения контейнеров, созданные с помощью Cloud Build, хранятся в Google Реестр контейнеров, в то время как другие артефакты, такие как двоичные файлы, могут храниться в корзинах облачных хранилищ Google или в любом стороннем репозитории. Запросы на облачную сборку могут выполняться вручную или запускаться автоматически в зависимости от изменений в источнике

#### **4.3 Защита программного обеспечения**

Чтобы безопасно разрабатывать и запускать приложения, Google Cloud предоставляет инструменты для внедрения передовых методов обеспечения безопасности на каждом этапе процесса разработки. Перед отправкой запросов на сборку можно настроить следующие механизмы доступа и управления, предоставляемые Cloud Build:

- Облачное управление идентификацией и доступом (IAM) управляет правами управления для создания, просмотра и отмены запросов на сборку;
- Журналы облачного аудита для последующего анализа запросов на сборку;
- Облачная служба управления ключами, позволяющая использовать зашифрованные ресурсы в запросах на сборку.

Во время сборки также можно использовать облачную сборку с двоичной авторизацией — средство контроля безопасности, которое гарантирует, что в кластерах Anthos GKE будут развернуты только надежные образы контейнеров. Бинарная авторизация работает за счет интеграции подтверждения в процесс создания образа контейнера, где изображения подписываются цифровой подписью на основе их уникального дайджеста. Бинарная авторизация также может быть расширена для поддержки других вариантов использования, таких как проверка сборки или сканирование уязвимостей. В сценариях проверки сборки бинарная авторизация может подтвердить, что образ контейнера был создан определенной системой сборки. В сценариях сканирования уязвимостей бинарная авторизация также может быть интегрирована с анализом контейнеров, чтобы гарантировать, что все выявленные уязвимости будут устранены до подписания образов контейнеров. Во время развертывания

средство принудительной двоичной авторизации использует средство проверки подлинности цифровой подписи. Таким образом, к развертыванию допускаются только образы контейнеров с подтвержденными сертификатами.

#### **4.4 Масштабируемая работа**

Когда речь заходит о повышении производительности разработчиков, критически важна возможность быстрого развертывания и запуска приложений в больших масштабах. Хотя существует множество инструментов, помогающих упаковывать и развертывать приложения для Kubernetes (таких как Helm), они требуют пользовательской интеграции со сторонними инструментами. Как обсуждалось выше, Cloud Code позволяет легко перейти от локального запуска к удаленному тестированию и, наконец, к развертыванию в производственных средах Anthos. Используя профили для каждой среды, Cloud Code предоставляет встроенные инструменты для организации безопасного развертывания приложений с помощью рабочих процессов облачной сборки. Это также позволяет создавать профили, поддерживающие создание образов контейнеров с помощью Cloud Build, которые затем развертываются в Anthos GKE с помощью Cloud Run. В целом, это сочетание облачного кода, облачной сборки и облачного запуска представляет собой интегрированный набор инструментов для ускорения процесса разработки современных приложений для вашей среды Anthos.

*Anthos обеспечивает прямую интеграцию для создания приложений и их упаковки в образы контейнеров с использованием облака Build for Anthos.*

## **Тема 5 Создание защищенной программной платформы с использованием Anthos и GitOps**

### **5.1 Интегрированные услуги Anthos**

Современные разработчики платформ преследуют общую цель: создать современную и безопасную платформу для запуска приложений в своей организации. Чтобы эта платформа была успешной, она должна обеспечивать соответствующие уровни абстракции для двух групп пользователей: разработчиков и операторов. Обеим группам требуется возможность быстро и эффективно отправлять обновления приложений или конфигурации, соблюдая при этом требования к управлению и эксплуатации своей организации. Именно разработчик платформы должен предоставить этим командам инструменты, необходимые для выполнения их основных задач, гарантируя при этом достижение приемлемого уровня безопасности и уровня обслуживания (SLO).

Отличным качеством является, то что Anthos уже в готовом виде предоставляет строительные блоки, необходимые архитектору платформы для создания именно такой защищенной программной платформы. Anthos также хорошо подходит для GitOps, который применяет контроль версий к инфраструктуре и правилам управления, используя стандартные рекомендации Git для управления изменениями конфигурации и объединения их.

Для разработчиков платформ первое, что нужно сделать, - это использовать Anthos. Управление конфигурацией как механизм для определения и предоставления инфраструктуры и управления в виде кода. Используя модель GitOps, операторы могут использовать Anthos Config Management для применения контроля версий к инфраструктуре и правилам управления. При производственных развертываниях разработчики платформы могут интегрировать Anthos Config Управление с Cloud Build, чтобы помочь операторам обеспечить соответствие изменений конфигурации существующим политикам и избежать внесения изменений, нарушающих инфраструктуру или управление.

Разработчикам, создающим и реализующим приложения, также необходима интеграция модели разработки приложений GitOps с их средой Anthos. При развертывании корпоративного программного обеспечения архитекторы платформы могут интегрировать репозитории Git с Cloud Build для управления развертываниями в промежуточных средах и их продвижения в производственных средах. Cloud Build также интегрируется с Anthos Config Management для проверки соответствия изменений в приложении/развертывании существующим политикам, гарантируя, что любые нарушения управления будут обнаружены во время сборки. Благодаря такой интеграции разработчики могут развертывать программное обеспечение, не нарушая существующих ограничений, и работать в режиме самообслуживания. (Рисунок 8)

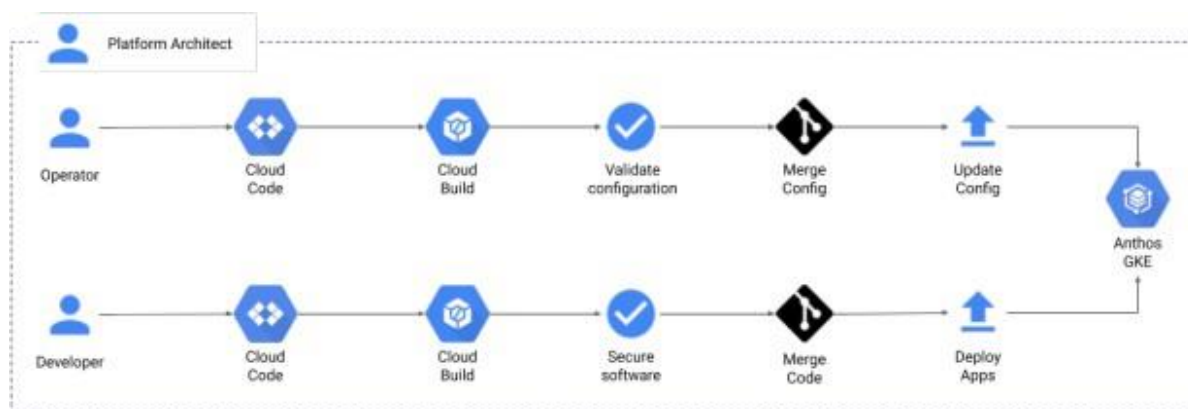


Рисунок 8 - Архитектура платформы

Как для операторов, так и для разработчиков этот комплексный подход к непрерывной интеграции (CI) перекладывает бремя соблюдения требований безопасности и передовых методов управления на “левую сторону” — любые критические изменения выявляются задолго до развертывания, снижая риски для производственных сред.

Anthos также предоставляет системы безопасности, которые обеспечивают соблюдение политик безопасности во время развертывания для образов контейнеров. С двоичной Авторизация, архитекторы платформы могут разработать систему, которая может автоматически и в цифровом виде проверять каждый компонент цепочки поставок программного обеспечения, гарантируя качество и целостность программного обеспечения перед развертыванием приложения в производственных средах (см. раздел "Защита программного обеспечения" в теме 4).

В резюме, можно создать современную и безопасную платформу поставка программного обеспечения с помощью Anthos и за счет использования GitOps лучшие практики для операторов и разработчиков, интегрируя ограждение безопасности и проверки управления с CI. инструменты, и с помощью таких инструментов как бинарные авторизации, чтобы обеспечить безопасный контейнер изображений развернутых на производстве.

В Anthos есть гораздо больше возможностей, чем в перечисленных выше основных компонентах.

## 5.2 Интеграция с облачным портфолио Google Cloud

Со временем цель Google Cloud состоит в том, чтобы обеспечить бесперебойную работу Anthos со всем набором продуктов. Рассмотрим примеры других текущих интеграций, не упомянутых ранее.

### *Операционные инструменты*

Google Cloud предлагает полный набор инструментов, предназначенных для мониторинга, устранения неполадок и улучшения производительности



облачной инфраструктуры, программного обеспечения и приложений. Ранее известные как Stackdriver, эти инструменты позволяют эффективно создавать и запускать рабочие нагрузки, обеспечивая высокую производительность и доступность приложений. В частности, эти инструменты позволяют:

- Собирать сигналы из внутренних и внешних приложений, платформ и сервисов Google Cloud
- Анализировать и отслеживать оперативную телеметрию;
- Настраивать соответствующие показатели производительности и доступности;
- Использовать встроенную наблюдаемость для устранения неполадок и улучшения работы приложений;
- Автоматизируйте операции, используя как готовые инструменты, так и инструменты, настраиваемые с помощью программных интерфейсов.

*Google Cloud предлагает полный операционный пакет, предназначенный для мониторинга, устранения неполадок и улучшения производительности облачной инфраструктуры, программного обеспечения и приложений*

### **Google Cloud Marketplace для Anthos**

Приложения Kubernetes - это корпоративные контейнерные решения с готовыми шаблонами развертывания, отличающиеся переносимостью, упрощенным лицензированием и консолидированным выставлением счетов. Их можно запускать на Anthos, в облаке, локально или в кластерах Kubernetes, размещенных в других средах. Это не просто образы контейнеров, а коммерческие приложения с открытым исходным кодом, созданные Google и повышающие производительность разработчиков, которые теперь доступны в Google Cloud Marketplace.

### **Миграция для Anthos**

Используйте Migrate for Anthos для перемещения и преобразования рабочих нагрузок непосредственно в контейнеры в Google Kubernetes Engine (GKE). Целевые рабочие нагрузки могут включать физические серверы и виртуальные машины, работающие локально, в Computer Engine или в других облаках, что дает возможность легко трансформировать существующую инфраструктуру.

### **Полная поддержка API для Kubernetes**

Anthos GKE поддерживает стандартный API Kubernetes, а также поставляется с дополнительными CRD, такими как Cluster API. Пользователи смогут использовать тот же интерфейс Kubernetes с открытым исходным кодом и несколькими дополнительными API для управления и масштабирования на уровне предприятия.

### **5.3 Партнерская экосистема Anthos**

Разработчики инвестируют в программное обеспечение и инфраструктуру, но есть еще возможность инвестировать в свое облачное будущее с помощью Anthos. Anthos тесно сотрудничает с экосистемой партнеров, предлагая инновационные решения, использующие передовые в отрасли технологии Google Cloud с открытым исходным кодом. Существуют партнеры по аппаратной, программной и системной интеграции, готовые помочь клиентам использовать Anthos с самого начала.

#### **Партнеры по консалтингу, MSP и системной интеграции**

Консалтинговые компании, поставщики управляемых услуг и партнеры по системной интеграции могут настраивать, устанавливать, управлять и эксплуатировать Anthos. Системные интеграторы, включая Accenture, Arctiq, Atos, Cognizant, Deloitte., HCL, IGW, SADA, SoftServe, Wipro, WWT и другие также помогают модернизировать и расширить приложения с помощью сервисов и решений, которые помогут внедрить Anthos в среду.

Google Cloud и партнеры стремятся встретить клиентов там, где они находятся, и предоставить им возможность запускать ключевые рабочие нагрузки и приложения в среде, наиболее подходящей для их бизнеса.

#### **Услуги канала**

Клиенты могут приобретать и устанавливать Anthos через предпочитаемый ими канал и партнеров-реселлеров, если они хотят сохранить существующие единые варианты выставления счетов и обслуживания.

#### **Встроенные и интегрированные решения**

Google тесно сотрудничает с избранными партнерами, внедряя и интегрируя ключевые технологии, позволяющие клиентам работать без подзарядки. Anthos поставляется с операционной системой Canonical Ubuntu и предварительно сконфигурирован для работы с балансировщиками нагрузки F5 BIG-IP для локальной установки. Также доступно руководство по установке для клиентов, которые хотят настроить предпочитаемые ими партнерские решения, такие как балансировщики нагрузки Citrix.

#### **Партнеры по аппаратной инфраструктуре и платформам**

Anthos GKE on-prem разработан для существующих у заказчиков сред vSphere. Заказчики, желающие обновить свои локальные аппаратные решения и инфраструктуру центров обработки данных, могут обратиться к партнерам, которые внедрили Anthos в свой стек решений. Такие партнеры, как Cisco, Dell EMC, Hewlett Packard Enterprise (HPE), Intel и Lenovo взяли на себя обязательства по внедрению Anthos в свою инфраструктуру. Проверять Anthos в

своих стеках решений, наши общие клиенты могут выбирать оборудование в зависимости от своих потребностей в хранении, памяти и производительности.

Партнеры по аппаратным платформам опубликовали эталонные архитектуры и могут поддерживать клиентов, использующих Anthos в своих решениях, включая:

- Atos (BullSequana)
- Cisco (HyperFlex)
- Dell EMC (семейство VxFlex)
- HPE (SimpliVity, Nimble Storage dHCI, Synergy)
- Intel (отдельные решения)
- Lenovo (ThinkAgile VX)
- NetApp HCI
- Nutanix AOS (совместно с vSphere)

Кроме того, HPE является авторизованным партнером Google по перепродаже облачных решений и предоставлению услуг, который может предложить клиентам гибридное облачное решение Anthos в рамках модели потребления с HPE GreenLake.

### **Google Cloud Marketplace для программного обеспечения и SaaS – решений.**

Клиенты могут использовать предпочитаемое ими программное обеспечение, готовое к использованию в Kubernetes и Istio, или выбрать из постоянно растущего списка программного обеспечения с открытым исходным кодом и коммерчески поддерживаемых SaaS-решений, доступных в Google Cloud Marketplace.

Google Cloud Marketplace - это канонический ресурс для поиска, развертывания и использования готовых к использованию приложений Kubernetes для обеспечения безопасности и идентификации, обработки данных, аналитики, инструментов разработчика, ИТ-операций и многого другого. Приложения Kubernetes - это готовые для предприятия контейнерные решения с готовыми шаблонами развертывания, обладающие такими возможностями, как переносимость, упрощенное лицензирование и консолидированное выставление счетов. Они могут быть развернуты в Anthos, в облаке и локально, а в будущем - в кластерах Kubernetes, размещенных в других средах. Это не просто образы контейнеров, а коммерческие решения, созданные партнерами.

### **Готовые партнерские решения Anthos**

Anthos Ready - это партнерские решения, соответствующие квалификационным требованиям Google Cloud и прошедшие проверку на совместимость с Anthos для удовлетворения потребностей корпоративных клиентов в разработке инфраструктуры и приложений. Партнерским решениям, которые соответствуют применимым квалификационным требованиям,

присваивается значок “Работает с Anthos” для идентификации совместимой инфраструктуры. Программа Anthos Ready включает в себя хранилище, сеть, платформу, безопасность и идентификацию, данные и аналитику, инструменты для разработчиков и ИТ-решения от партнеров. В будущем программа будет расширена и включит в себя другие категории решений.

Партнеры Anthos Ready отвечают множеству критериев, таких как:

- Демонстрация основных функциональных возможностей Kubernetes, включая динамическую подготовку программного обеспечения и сервисов с помощью открытых и переносимых API-интерфейсов, созданных на основе Kubernetes;

- Проверенная возможность автоматического управления службами в кластерах, включая масштабирование;

- Упрощенное развертывание в соответствии с практикой Kubernetes.

Например, Anthos Ready Storage распознает партнерские решения, которые отвечают основному набору требований для оптимальной работы с Anthos, работающими локально, и помогает организациям выбирать решения для хранения данных, которые могут быть развернуты с Anthos. В настоящее время специальные решения для хранения данных от Dell, HPE, NetApp, Portworx, Pure Storage и Robin.io получили квалификацию Anthos Ready.

## **5.4 Возможности Anthos в гибридной и мультиоблачной среде**

### **1 Вариант развертывания с использованием Google Cloud**

предлагает эффективный способ повышения производительности приложений путем перемещения вычислений ближе к данным хранилища. При работе с Google Cloud рекомендуется использовать Anthos для разработки, развертывания и оптимизации контейнерных задач прямо на этой платформе. Сервисы Google Cloud, основанные на искусственном интеллекте, машинном обучении и анализе данных, предоставляют ценную информацию, позволяя принимать более эффективные решения и оперативно внедрять инновации.

### **2 Вариант развертывания с использованием VMware vSphere**

Если ваша организация работает на платформе VMware vSphere, то имеется возможность запускать кластеры Anthos непосредственно в VMware. Это позволяет создавать и обновлять кластеры Kubernetes, а также управлять ими в контексте текущей инфраструктуры. Этот вариант особенно подходит, если организация полностью интегрирована с vSphere, а также если имеется общее оборудование с управлением жизненным циклом ОС, которое используется несколькими командами или кластерами.

При работе с кластерами Anthos в VMware можно выполнять все текущие задачи в локальной среде без необходимости в значительных изменениях инфраструктуры. Инструмент Migrate for Anthos позволяет обновить устаревшие приложения, переместив их из виртуальных машин в контейнеры.

Затем можно выбрать, где хранить обновленные контейнерные приложения: на местном оборудовании или в облаке. В любом случае, с Anthos управление и обновление приложений становится удобным и эффективным.

### **3 Вариант развертывания с использованием AWS**

При условии что в компании действуют несколько команд, вероятно, они применяют различные технологии или даже обращаются к разным облачным платформам. Anthos решает эту проблему, обеспечивая единую платформу для разработки.

В контексте данного варианта развертывания можно создавать кластеры, используя Google Kubernetes Engine, и получать все преимущества платформы Anthos и Google Cloud: инструменты Kubernetes для удобного развертывания, возможность использования Anthos Config Management для внедрения правил и настроек, а также функциональность Anthos Service Mesh для управления различными сервисами. С помощью Google Cloud Console можно управлять всеми приложениями с единого интерфейса, независимо от того, где они развернуты.

### **4 Вариант развертывания с использованием физических серверов**

Хотя виртуальные машины имеют свои преимущества, всё больше организаций предпочитают использовать Kubernetes на физических серверах. Это более просто и дешевле. При этом нет необходимости в гипервизоре: Anthos может запускаться на физических серверах с вашей операционной системой. Пользователям доступны функции организации сетей, управления жизненным циклом, диагностики, мониторинга и ведения журналов.

Критически важные приложения обычно требуют максимальной производительности и минимальной задержки для вычислений, хранения данных и работы с сетевыми стеками. Без использования гипервизора задержка сокращается, что позволяет запускать на физическом сервере ресурсоёмкие приложения для машинного обучения, обработки видео с использованием графических процессоров и т.д.

Выбор Anthos на сервере без ОС позволит сократить расходы на оборудование, операционную систему и сетевую инфраструктуру. Ещё один важный момент: Anthos имеет минимальные системные требования, что означает, что можно воспользоваться всеми преимуществами платформы (централизованное управление, повышенная гибкость и быстрота разработки) даже при работе с самыми ресурсоёмкими приложениями.

### **5 Вариант развертывания с использованием Microsoft Azure**

Непрерывно расширяется функциональность Anthos и внедряется поддержка новых сред и платформ. В скором времени Anthos будет доступен для использования на платформе Azure.

## **6 Вариант развертывания с использованием прикрепленных кластеров**

При использовании Anthos, возможно, возникнет вопрос о том, что делать с существующими кластерами Kubernetes. Предлагаемый вариант позволяет сохранить эти кластеры и в то же время воспользоваться основными функциями Anthos. Будь то Amazon EKS, Microsoft AKS или Red Hat OpenShift, кластеры могут быть интегрированы с Anthos. Это позволит централизованно управлять развёртыванием через Google Cloud Console, применять правила и настройки с помощью Anthos Config Management, а также отслеживать журналы на одной платформе.

Естественно, возможности Anthos не являются безграничными – обслуживать и обновлять кластеры необходимо вручную. Однако данный вариант обеспечивает максимально удобный переход на Anthos, независимо от используемых облачных сервисов.

Были рассмотрены шесть вариантов развертывания в гибридной и мультиоблачной среде с использованием Anthos. В зависимости от места размещения данных и инфраструктуры можно выбрать один или даже несколько вариантов для модернизации ваших предложений. Anthos представляет собой современную платформу разработки, которая эффективно работает как в локальной среде, так и в облачной, поддерживает существующую инфраструктуру ЦОД, экономит ресурсы и соответствует самым современным требованиям безопасности.

## Примеры использования Anthos

### Anthos Service Mesh на примере: Авторизация

Рассмотрим такую авторизацию и как включить ее с помощью Anthos Service Mesh в примере приложения, а также как включить политики авторизации для микросервисов.

Создаем Authorization Policy доступ DENY к микросервису, а затем создадим Authorization Policy конкретный ALLOW доступ к микросервису.

Аутентификация подтверждает личность — является ли этот сервис тем, кем он себя называет? Авторизация проверяет разрешение — разрешено ли этому сервису это делать? Идентичность является фундаментальной для этой идеи. С помощью Anthos Service Mesh Authorization Policies можно контролировать взаимодействие между рабочими нагрузками в сети для повышения безопасности и доступа.

В микросервисной архитектуре, где вызовы осуществляются через границы сети, традиционные правила брандмауэра на основе IP часто недостаточны для защиты доступа между рабочими нагрузками. С помощью Anthos Service Mesh можно установить правила авторизации для:

- Управления доступом к рабочим нагрузкам внутри вашей сетки, либо между рабочими нагрузками, либо между конечными пользователями.
- Определения широко или детально политики в зависимости от потребностей.

### 1. Развертывание входящего шлюза

1. Установите текущий контекст для kubectl кластера:

```
gcloud container clusters get-credentials
CLUSTER_NAME \
--project=PROJECT_ID \
--zone=CLUSTER_LOCATION
```

2. Создайте пространство имен для входного шлюза:

```
kubectl create namespace asm-ingress
```

3. Включите пространство имен для внедрения. Действия зависят от типа вашей Anthos Service Mesh (управляемой или внутрикластерной).

Примените asm-managed метку редакции к пространству имен:

```
kubectl label namespace asm-ingress \
istio-injection- istio.io/rev=asm-managed -
-overwrite
```

4. Разверните пример шлюза в anthos-service-mesh-samples репозитории:

```
kubectl apply -n asm-ingress \
-f docs/shared/asm-ingress-gateway
```

Ожидаемый результат:

```
serviceaccount/asm-ingressgateway configured
service/asm-ingressgateway configured
deployment.apps/asm-ingressgateway configured
gateway.networking.istio.io/asm-ingressgateway configured
```

## 2. Развертывание примера приложения интернет-бутика

1. Установите текущий контекст для kubectl кластера:

```
gcloud container clusters get-credentials
CLUSTER_NAME \
--project=PROJECT_ID \
--zone=CLUSTER_LOCATION
```

2. Создайте пространство имен для примера приложения:

```
kubectl create namespace onlineboutique
```

3. Пометьте onlineboutique пространство имен для автоматического внедрения прокси Envoy.

4. Разверните пример приложения, учетные записи VirtualService для внешнего интерфейса и сервисные учетные записи для рабочих нагрузок.

```
kubectl apply \
-n onlineboutique \
-f docs/shared/online-boutique/virtual-
service.yaml
kubectl apply \
-n onlineboutique \
-f docs/shared/online-boutique/service-accounts
```

### Посмотреть ваши услуги

1. Просмотрите модули в onlineboutiqueпространстве имен:

```
kubectl get pods -n onlineboutique
```

Ожидаемый результат:

NAME	READY	STATUS	RESTARTS	AGE
adservice-85598d856b-m84m6	2/2	Running	0	2m7s
cartservice-c77f6b866-m67vd	2/2	Running	0	2m8s
checkoutservice-654c47f4b6-hqtqr	2/2	Running	0	2m10s
currencyservice-59bc889674-jhk8z	2/2	Running	0	2m8s
emailservice-5b9ffff7cb8-8nq wz	2/2	Running	0	2m10s



frontend-77b88cc7cb-mr4rp	2/2	Running	0	2m9s
loadgenerator-6958f5bc8b-55q7w	2/2	Running	0	2m8s
paymentservice-68dd9755bb-2jmb7	2/2	Running	0	2m9s
productcatalogservice-84f95c95ff-c5kl6	2/2	Running	0	114s
recommendationservice-64dc9dfbc8-xfs2t	2/2	Running	0	2m9s
redis-cart-5b569cd47-cc2qd	2/2	Running	0	2m7s
shippingservice-5488d5b6cb-1fhht	2/2	Running	0	2m7s

Все модули вашего приложения должны быть запущены и работать, 2/2 в READY столбце должен быть отмечен значок. Это указывает на то, что в модули успешно внедрен прокси-сервер Envoy.

2. Получите внешний IP-адрес и установите его в переменную:

```
kubectl get services -n asm-ingress
export FRONTEND_IP=$(kubectl --namespace asm-ingress \
get service --output
jsonpath='{.items[0].status.loadBalancer.ingress[0].ip}'
\
)
```

3. Посетите EXTERNAL-IPадрес в своем веб-браузере. Вы должны ожидать, что увидите интернет-бутик в своем браузере.(Рисунок 9)

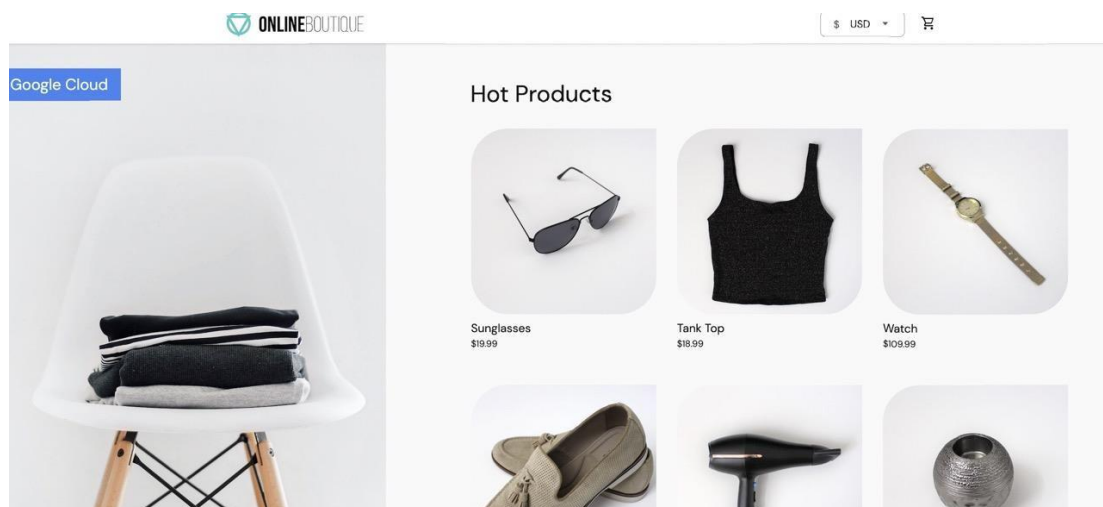


Рисунок 9 – Интернет-бутик в браузере

### DenyAll Авторизация для рабочей нагрузки

В этом разделе добавлен параметр AuthorizationPolicy для запрета всего входящего трафика к валютному сервису. AuthorizationPolicies работает путем преобразования AuthorizationPolicies в читаемые Envoy конфигурации и применения конфигураций к дополнительным прокси-серверам. Это позволяет прокси-серверу Envoy разрешать или отклонять входящие запросы к службе.

1. Примените `AuthorizationPolicy` к `currency-service`. Обратите внимание на совпадение метки `currency-service` в файле `YAML`.

```
kubectl apply -f docs/authorization/currency-deny-all.yaml -n onlineboutique
```

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: currency-policy
spec:
  selector:
    matchLabels:
      app: currency-service
```

2. Попробуйте получить доступ к своему шлюзу `EXTERNAL-IP` для просмотра онлайн-бутика в веб-браузере. Вы должны увидеть ошибку авторизации (500 Internal Service Error) от `currency-service`.

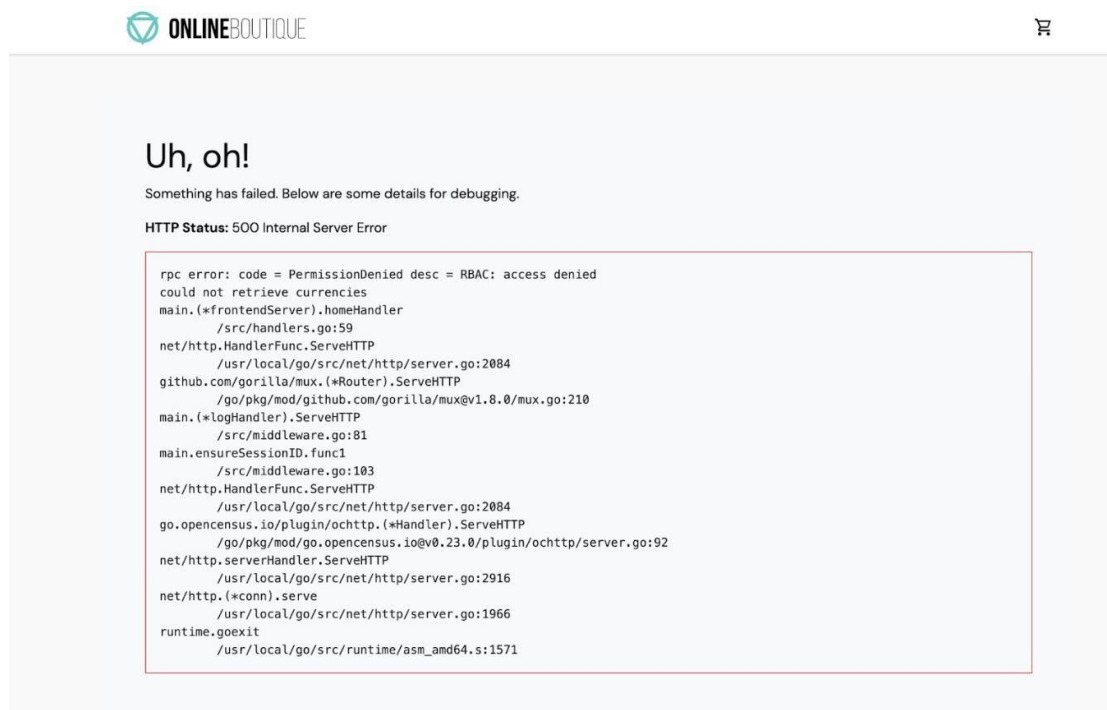


Рисунок 10 – Код ошибки авторизации

### Наблюдайте за журналами прокси-сервера

Чтобы узнать, что происходит в прокси-сервере, можно просмотреть журналы модуля.

1. Получите имя вашего `currency-service` модуля:

```
CURRENCY_POD=$(kubectl get pod -n onlineboutique |grep
currency|awk '{print $1}')
```

2. Настройте прокси-сервер Envoy, чтобы разрешить журналы уровня трассировки. По умолчанию заблокированные авторизационные вызовы не протоколируются:

```
kubectl exec -it $CURRENCY_POD -n onlineboutique -c
istio-proxy -- curl -X POST
"http://localhost:15000/logging?level=trace"
```

Ожидаемый результат: active loggers: admin: trace  
alternate\_protocols\_cache: trace ... tracing: trace  
upstream: trace udp: trace wasm: trace

3. Используйте curl для отправки трафика на ваш сайт EXTERNAL\_IP для создания журналов:

```
for i in {0..10}; do
  curl -s -I $FRONTEND_IP ; done
```

4. Просмотрите журналы, связанные с контролем доступа на основе ролей (RBAC), в вашем istio-прокси:

```
kubectl logs -n onlineboutique $CURRENCY_POD -c istio-
proxy | grep -m5 rbac
```

Ожидаемый результат:

```
2022-07-08T14:19:20.442920Z      debug    envoy rbac
checking request: requestedServerName:
outbound_.7000_..currencyservice.onlineboutique.svc.clus
ter.local, sourceIP: 10.8.8.5:34080, directRemoteIP:
10.8.8.5:34080, remoteIP: 10.8.8.5:34080, localAddress:
10.8.0.6:7000, ssl: uriSanPeerCertificate:
spiffe://christineskim-tf-
asm.svc.id.goog/ns/onlineboutique/sa/default,
dnsSanPeerCertificate: , subjectPeerCertificate:
OU=istio_v1_cloud_workload,O=Google LLC,L=Mountain
View,ST=California,C=US, headers: ':method', 'POST'
2022-07-08T14:19:20.442944Z      debug    envoy rbac
enforced denied, matched policy none
2022-07-08T14:19:20.442965Z      debug    envoy http
[C73987][S13078781800499437460] Sending local reply with
details rbac_access_denied_matched_policy[none]
...

```

В журналах должны увидеть сообщение о том, что currencyservice настроена блокировка входящих запросов.

## Разрешить ограниченный доступ

Вместо DENYALL политики можно разрешить доступ для определенных рабочих нагрузок. Это будет актуально в микросервисной архитектуре, где вы хотите гарантировать, что только авторизованные сервисы могут взаимодействовать друг с другом.

В этом разделе вы включите сервис frontend checkout возможность общения с currency сервисом.

1. В приведенном ниже файле увидите, что определенный source.principal (клиент) имеет доступ к разрешенному списку currencyservice:

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: currency-policy
spec:
  selector:
    matchLabels:
      app: currencyservice
  rules:
  - from:
    - source:
        principals:
        ["cluster.local/ns/onlineboutique/sa/frontend"]
    - from:
    - source:
        principals:
        ["cluster.local/ns/onlineboutique/sa/checkoutservice"]
```

Примените политику:

```
kubectl apply -f docs/authorization/currency-allow-frontend-checkout.yaml -n onlineboutique
```

1. Откройте EXTERNAL-IP в своем веб-браузере, теперь вы сможете получить доступ к онлайн-бутику.

## Очистить

Чтобы избежать списания средств с вашей учетной записи Google Cloud за ресурсы, используемые в этом руководстве, либо удалите проект, содержащий ресурсы, либо сохраните проект и удалите отдельные ресурсы.

Чтобы избежать постоянного списания средств с вашей учетной записи Google Cloud за ресурсы, используемые в этом руководстве, можете либо удалить проект, либо удалить отдельные ресурсы.

### Удалить проект

**Внимание:** Удаление проекта имеет следующие последствия:

- **Все в проекте удалено.** Если для этого руководства вы использовали существующий проект, то при его удалении вы также удаляете всю другую работу, выполненную в этом проекте.
- **Пользовательские идентификаторы проектов теряются.** Создавая этот проект, вы, возможно, создали собственный идентификатор проекта, который хотите использовать в будущем. Чтобы сохранить URL-адреса, использующие идентификатор проекта, например URL-адрес appspot.com, удалите выбранные ресурсы внутри проекта, а не удаляйте весь проект.

В Cloud Shell удалите проект:

```
gcloud projects delete PROJECT_ID
```

### Удалить ресурсы

- Если вы хотите сохранить свой кластер и удалить образец интернет-бутика:

1. Удалите пространства имен приложения:

```
kubectl delete namespace onlineboutique
```

Ожидаемый результат:

```
namespace "onlineboutique" deleted
```

2. Удалите пространство имен Ingress Gateway:

```
kubectl delete namespace asm-ingress
```

Ожидаемый результат:

```
amespace "asm-ingress" deleted
```

- Если вы хотите предотвратить дополнительные расходы, удалите кластер:

```
gcloud container clusters
delete CLUSTER_NAME \
--project=PROJECT_ID \
--zone=CLUSTER_LOCATION
```

### 3. Гибридное управление Anthos и несколькими облаками

Гибридное облако позволяет организациям использовать преимущества общедоступного облака, сохраняя при этом некоторую часть их прикладной ландшафт и данные будут размещаться в их текущих центрах обработки данных. Многие организации внедрение гибридных облаков по целому ряду

причин – конфиденциальность и безопасность данных, соглашения об уровне обслуживания

(SLA), соображения стоимости, проприетарные приложения, потребности пользователей и так далее. Благодаря гибриднему облачному подходу, компании могут выборочно переносить рабочие нагрузки и данные по запросу к поставщикам общедоступного облака, таким как

Облачная платформа Google, AWS, Azure и т.д. Это также известно как массовое предоставление трафика по запросу

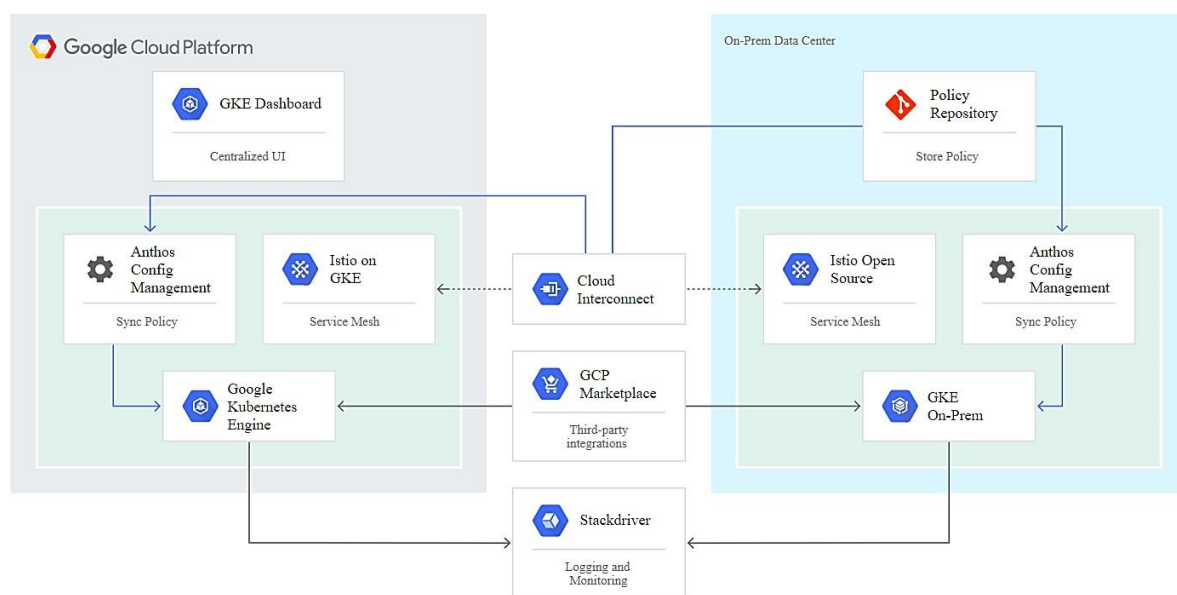


Рисунок 11 - Компоненты гибридной облачной архитектуры Anthos

Гибридные облачные реализации, как правило, сложны из-за необходимости интеграции между центрами обработки данных и общедоступных облачных центров обработки данных, проблемы сетевой безопасности / Интернета, сложные двунаправленные маршрутизация трафика, требования к доступу к данным и их подготовке и т.д. Следовательно, гибридные облачные реализации, как правило, требуют несколько инструментов и служб сторонних производителей в зависимости от ожидаемых конкретных возможностей. Google Anthos от Cloud упрощает гибридное облако, предоставляя необходимые инструменты и сервисы для безопасной и масштабируемой работы реализации гибридного облака. На рисунке 11 показаны основные элементы архитектуры гибридного облака с Google Cloud и Anthos. Показаны компоненты, работающие в общедоступном облаке Google, в основном совпадают с компонентами, работающими в обычном облаке.

Эталонная архитектура: Anthos от Google Cloud с Lenovo ThinkAgile VX оперативное облако. Следовательно, кластеры Anthos, работающие в оперативном центре обработки данных, по сути, являются расширением

общедоступного облака. Поэтому подход гибридного облака Google отличается от подхода других поставщиков.

Согласованность архитектуры в общедоступных и локальных облаках упрощает развертывание и что еще более важно, заказчикам не нужно выполнять никакой дополнительной работы для внедрения гибридного облака. Как только встроенный кластер Anthos развернут и подключен к вычислительной платформе Google, гибридное облако готово к использованию.

Движок Google Kubernetes Engine (GKE) является общим знаменателем между общедоступными и локальными облаками. Поскольку Anthos в первую очередь ориентирована на создание контейнерных рабочих нагрузок, работающих поверх Kubernetes, гибридного облака, реализованное с помощью Anthos, обеспечивает межоблачную координацию контейнеров и микросервисов по всему миру.

Кластеры Kubernetes. Миграция контейнерных рабочих нагрузок между локальными кластерами и кластерами GKE, работающими в другие облака могут быть созданы с помощью общедоступного реестра контейнеров, такого как Google container registry (GCR), или через ваши собственные реестры, защищенные централизованной идентификацией и контролем доступа только к разрешить аутентифицированным пользователям или учетным записям служб отправлять или извлекать изображения контейнеров из реестра. Отсутствует необходимость конвертировать изображения контейнеров, запущенные на предварительном этапе, для запуска на управляемом Google движке Kubernetes в общедоступном облаке.

### **Движок Google Kubernetes (GKE)**

Kubernetes (сокращенно известный как K8s) - проект с открытым исходным кодом, разработанный Google для обеспечения масштабируемая организация контейнерных рабочих нагрузок. Kubernetes обычно работает на кластере компьютеров как единая целая распределенная система, состоящая из одного или нескольких главных узлов и одного или нескольких рабочих узлов. Мастер узлы запускают основные службы Kubernetes, такие как сервер API, планировщик, база данных конфигурации кластера. (etcd), службы аутентификации / авторизации, виртуальные сети и т.д. Рабочие узлы выполняют действия пользователей.

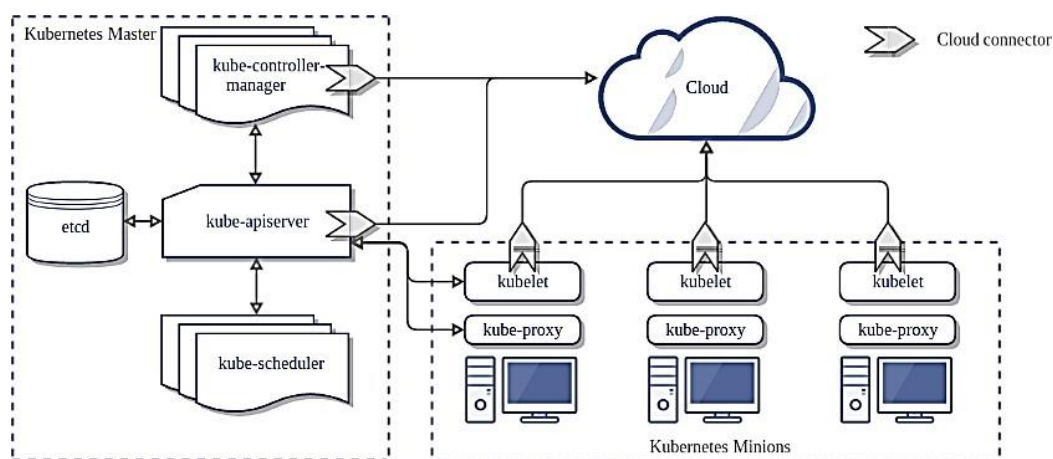


Рисунок 12 - Архитектура сервиса Google Kubernetes

Как показано на рисунке 12, главный узел (узлы) запускает основные службы управления кластером, такие как kube-apiserver, планировщик kube и т.д. Рабочие узлы взаимодействуют с главными узлами через службу kubelet, которая отвечает за управление модулями Kubernetes на локальных серверах. Модули запускают один или несколько контейнеров внутри. Служба kube-proxy предоставляет простой сетевой прокси для обычного входящего трафика в модули.

### Управление несколькими кластерами

Предприятиям, использующим облако, приходится использовать множество решений, как локальных, так и общедоступных облако по разным причинам – контроль затрат, потребности в рабочей нагрузке конкретных приложений, требования пользователей, соглашения об уровне обслуживания и т. д. далее. Из-за отсутствия общих стандартов у различных облачных провайдеров для управления несколькими облаками требуется специальное обучение и административные навыки. Поскольку Kubernetes становится стандартом де-факто для контейнерных управление приложениями для многих клиентов было бы полезно предоставить единую плоскость управления для управление кластерами Kubernetes из любого места с единой консоли.

С внедрением Anthos облачная платформа Google также позволяет управлять кластерами Kubernetes, работающими под управлением в любом месте – в Google Cloud, в готовых центрах обработки данных или у других облачных провайдеров, таких как Amazon AWS, из в одном месте. Кроме того, возможность создания нескольких кластеров также позволяет управлять конфигурацией в облаке и готовые среды, а также рабочие нагрузки, выполняемые в различных средах.

Основными компонентами мультикластерного управления являются GKE connection hub (сокращенно Connect), Google и системой управления конфигурацией Anthos.



## Google Cloud Connect

Connect позволяет подключать к Google cloud platform готовые кластеры Kubernetes, а также кластеры Kubernetes, работающие в других общедоступных облаках. Connect использует зашифрованное соединение между кластерами Kubernetes и Google cloud platform project и позволяет авторизованным пользователям входить в кластеры, получать доступ к сведениям о своих ресурсах, проектах и кластерах, а также управлять инфраструктурой кластера и рабочими нагрузками, независимо от того, работают ли они на оборудовании Google или где-либо еще. (Рисунок 14)

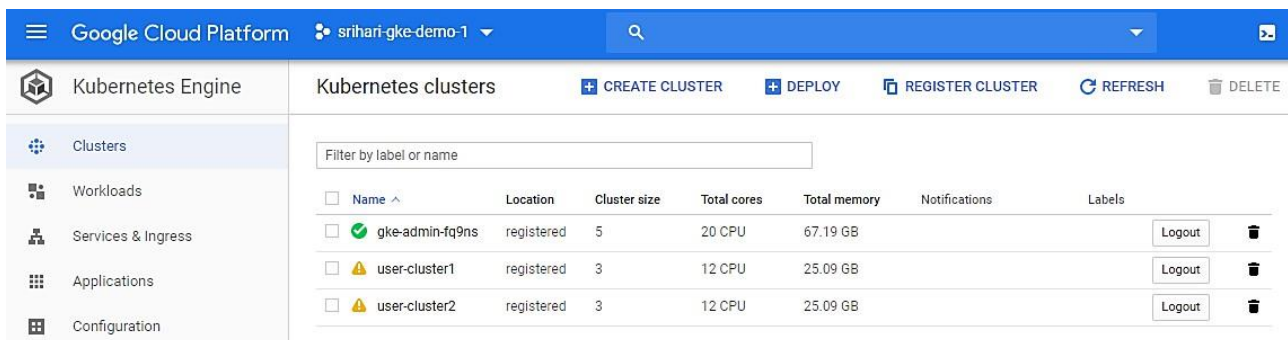


Рисунок 13 - Google cloud connect для управления несколькими кластерами

Агент GKE Connect устанавливается в вашем удаленном кластере.

Для кластера не требуется общедоступный IP-адрес.

Аутентифицированное и зашифрованное соединение между кластером Kubernetes и GCP .

Может работать с NAT и брандмауэрами.

Взаимодействие пользователей с кластерами видно в журналах аудита Kubernetes.

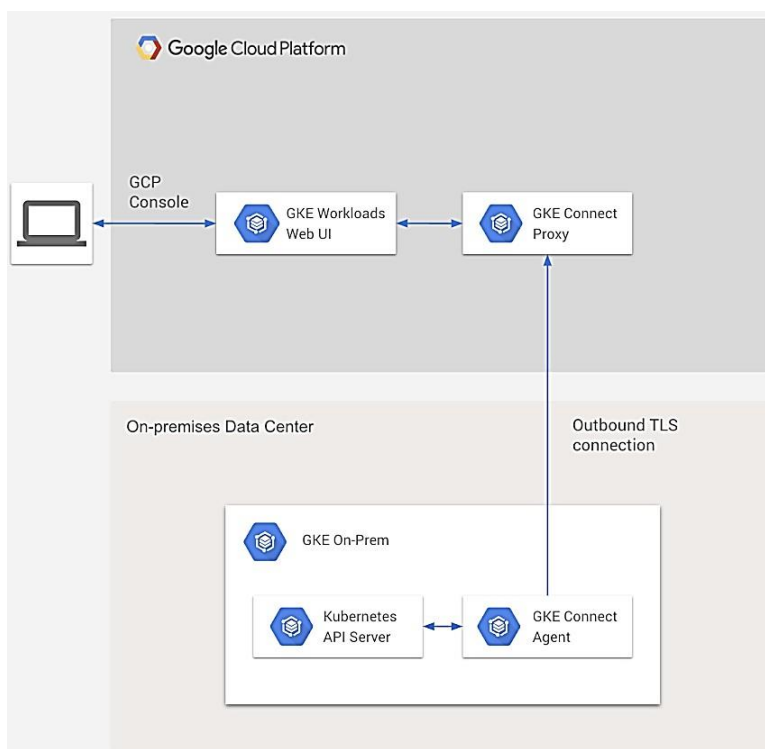


Рисунок 14 - Безопасное подключение на основе протокола TLS к облачной платформе Google с предварительной установки

## GCP

Консоль Google Cloud Platform Console действует как единая точка управления и мониторинга кластеров Kubernetes работает в разных местах. Это тот же веб-интерфейс, который используется для управления всеми ресурсами Google Cloud. Например, кластеры compute Engine, хранилища, сети, кластеры Kubernetes и так далее. (Рисунок 15)

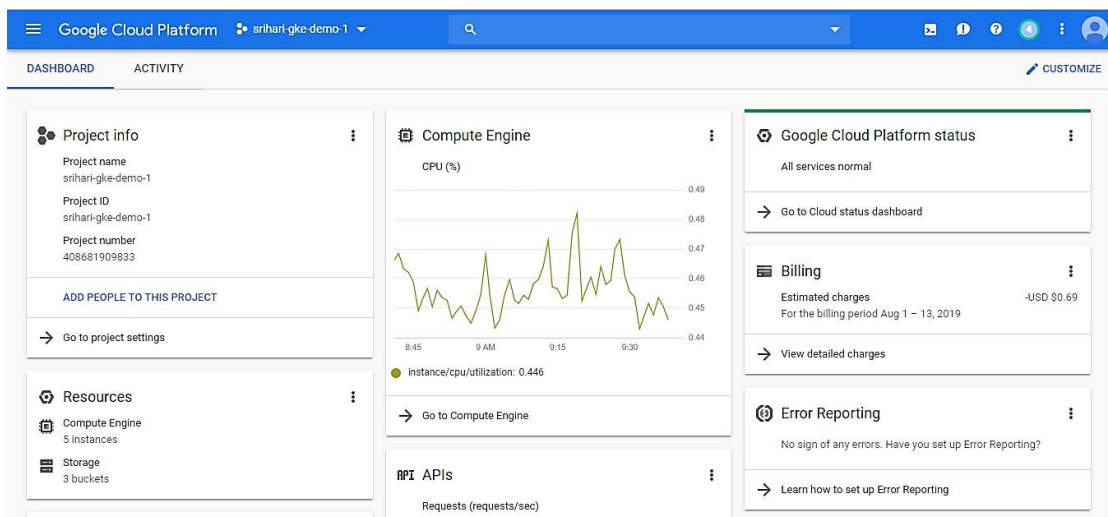
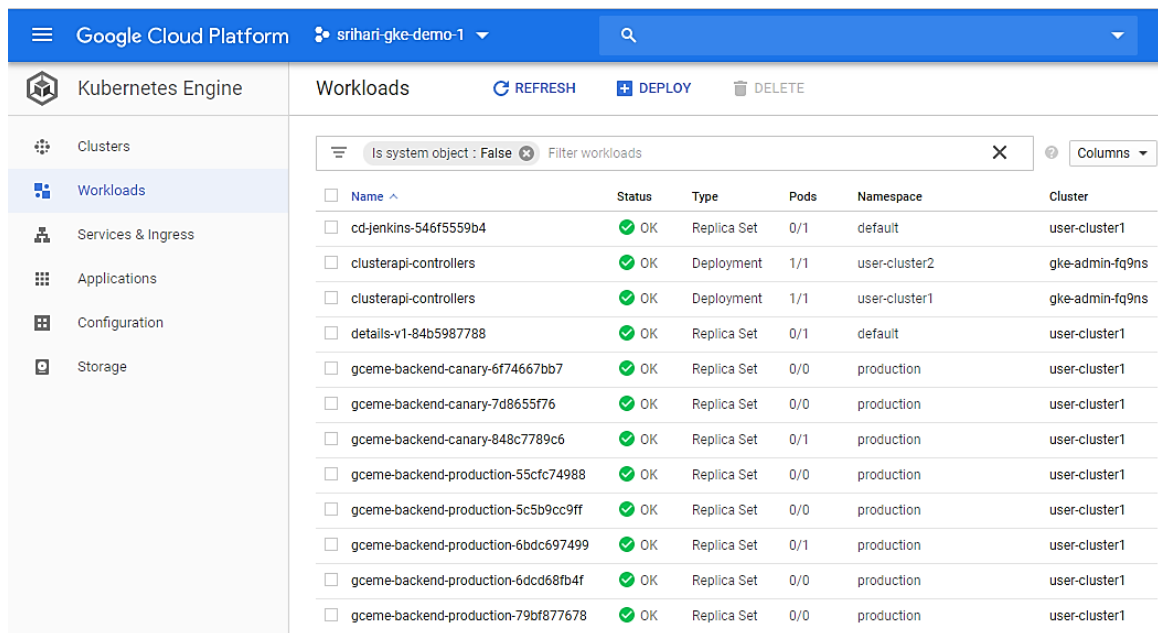


Рисунок 15 - Консоль облачной платформы Google

## Управление кластерами Anthos из GCP

Управление несколькими кластерами Kubernetes и рабочими нагрузками, выполняемыми в разных местах, становится простым с помощью консоли GCP. Например, с помощью консоли можно проверить работоспособность запущенного компьютера и внести в них любые изменения конфигурации. (Рисунок 16)

Обратите внимание, что кластеры Anthos GKE, работающие в вашем локальном центре обработки данных, должны быть подключены и зарегистрированы в GCP для доступа к Google и отображения в консоли GCP. Кластеры GKE on-prem приложения, развернутые через Anthos, автоматически регистрируются и подключаются к GCP в процессе настройки. (Рисунок 17)



Name	Status	Type	Pods	Namespace	Cluster
cd-jenkins-546f5559b4	OK	Replica Set	0/1	default	user-cluster1
clusterapi-controllers	OK	Deployment	1/1	user-cluster2	gke-admin-fq9ns
clusterapi-controllers	OK	Deployment	1/1	user-cluster1	gke-admin-fq9ns
details-v1-84b5987788	OK	Replica Set	0/1	default	user-cluster1
gceme-backend-canary-6f74667bb7	OK	Replica Set	0/0	production	user-cluster1
gceme-backend-canary-7d8655f76	OK	Replica Set	0/0	production	user-cluster1
gceme-backend-canary-848c7789c6	OK	Replica Set	0/1	production	user-cluster1
gceme-backend-production-55cfc74988	OK	Replica Set	0/0	production	user-cluster1
gceme-backend-production-5c5b9cc9ff	OK	Replica Set	0/0	production	user-cluster1
gceme-backend-production-6bdc697499	OK	Replica Set	0/1	production	user-cluster1
gceme-backend-production-6cdc68fb4f	OK	Replica Set	0/0	production	user-cluster1
gceme-backend-production-79bf877678	OK	Replica Set	0/0	production	user-cluster1

Рисунок 16 - Управление рабочими нагрузками в кластерах GKE с консоли GCP

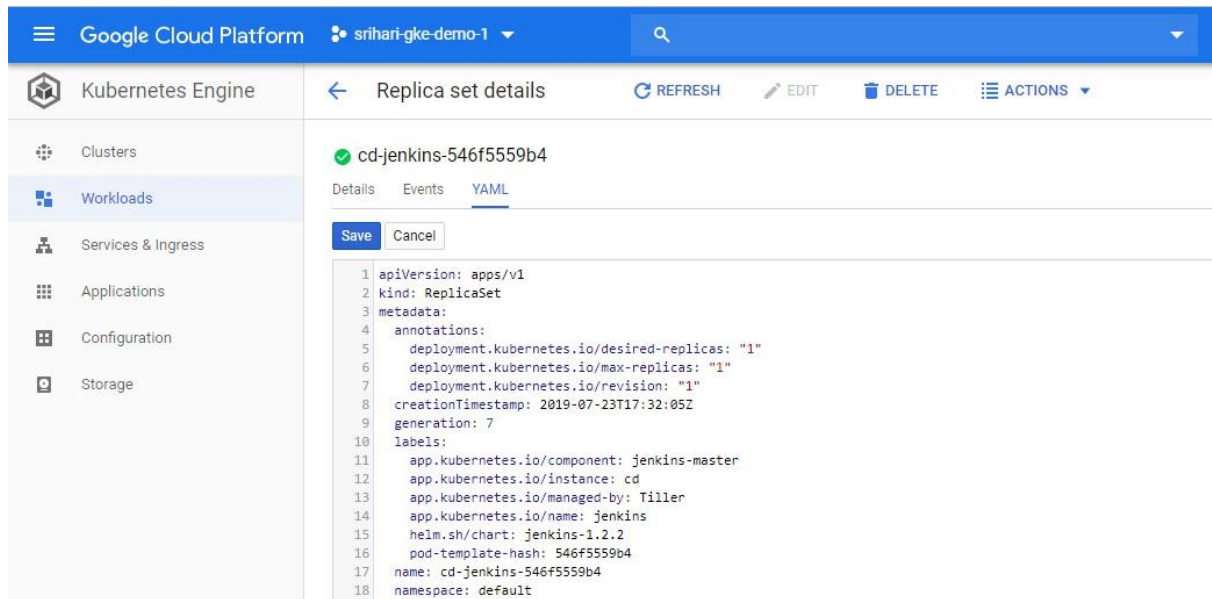


Рисунок 17 - Редактирование определения рабочей нагрузки из консоли GCP

## Конвейеры DevOps и CI / CD

DevOps - один из основных вариантов использования Anthos. Внедрена современная практика разработки программного обеспечения. Следуя методологиям Agile / Scrum и DevOps. В этом разделе описывается на высоком уровне, как реализовать непрерывную интеграцию и непрерывный конвейер развертывания (CI / CD) поверх кластеров Anthos Kubernetes. Непрерывная интеграция - это процесс, в ходе которого код, разрабатываемый несколькими разработчиками одновременно, постоянно извлекается из репозитория исходного кода, интегрируется, создается и тестируется.

## Развертывание и интеграция Jenkins с GKE on-prem

Для предварительной реализации конвейера CI / CD в кластере GKE можно использовать популярный инструмент CI / CD с открытым исходным кодом называется Jenkins. На рынке доступны другие популярные инструменты с открытым исходным кодом и коммерческие CI / CD, такие как хорошо, но у Jenkins есть широкая экосистема с открытым исходным кодом и коммерческие плагины для различных CI / CD включая интеграцию с Kubernetes и Docker, что делает его хорошо подходящим для Anthos.

Сам Jenkins может быть развернут в контейнере поверх кластера GKE on-prem, что упрощает развертывание. Далее подробно описывается развертывание Jenkins, пошаговое внедрение Jenkins в Kubernetes.

```

ubuntu@anthos-prod-admin-ws:~$ kubectl --kubeconfig=user-cluster1-kubeconfig get pods
NAME                                READY   STATUS    RESTARTS   AGE
cd-jenkins-546f5559b4-68c5p         1/1     Running   0           19d
details-v1-84b5987788-tjn9z         2/2     Running   0           13d
helloworld-v1-898dfb97c-g7m9l       2/2     Running   0           13d
helloworld-v2-74689c97d4-h9qh2      2/2     Running   0           13d
productpage-v1-8658f9948-f782h      2/2     Running   0           13d
ratings-v1-95d8d9c56-j7vnv         2/2     Running   0           13d
reviews-v1-78dfc5f4c6-84scs        2/2     Running   0           13d
reviews-v2-6f8c65db9-hm99j         2/2     Running   0           13d
reviews-v3-6dddd94f8b-2qpl7        2/2     Running   0           13d
ubuntu@anthos-prod-admin-ws:~$ kubectl get svc
NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
cd-jenkins                          LoadBalancer        10.99.193.61    10.0.10.200     8080:32558/TCP   25d
cd-jenkins-agent                    ClusterIP            10.110.31.51    <none>          50000/TCP        25d
details                              ClusterIP            10.103.4.221    <none>          9080/TCP         13d
helloworld                          ClusterIP            10.99.140.67    <none>          5000/TCP         13d
kubernetes                          ClusterIP            10.96.0.1       <none>          443/TCP          38d
productpage                          ClusterIP            10.106.164.149  <none>          9080/TCP         13d
ratings                              ClusterIP            10.100.176.139  <none>          9080/TCP         13d
reviews                              ClusterIP            10.96.48.162    <none>          9080/TCP         13d
ubuntu@anthos-prod-admin-ws:~$

```

Рисунок 18 - Инструмент Jenkins CI / CD, развернутый в виде контейнера в кластере GKE on-prem

После развертывания Jenkins в кластере увидите, что модули-контейнеры Jenkins успешно созданы и работают. На рисунке 18 приведены команды kubectl для проверки состояния модуля Jenkins pod и доступа. (Рисунок 19)



Welcome to Jenkins!




Рисунок 19 - Главный экран входа в систему Jenkins

После успешной установки Jenkins необходимо настроить Jenkins и подключить кластер Anthos GKE с Jenkins master для запуска конвейеров CI / CD. На портале Jenkins выберите “настроить Jenkins” и нажмите создайте конфигурацию “облака” (Рисунок 20). Здесь вам нужно указать “kubernetes” в качестве имени облака, потому что это же облако необходимо указать позже в определении конвейера. Кроме того, необходимо указать URL-адрес для Jenkins master и агента Kubernetes.

**Cloud**

**Kubernetes**

Name:

Kubernetes URL:

Kubernetes server certificate key:

Disable https certificate check:

Kubernetes Namespace:

Credentials:

Jenkins URL:

Jenkins tunnel:

Рисунок 20 - Определение облака Kubernetes в конфигурации Jenkins

Для того, чтобы ведущее устройство Jenkins могло успешно развернуть и запустить подчиненные устройства Jenkins в кластере Kubernetes, необходимо настроить учетные данные для кластера Kubernetes в глобальных учетных данных Jenkins. Можно скопировать и вставить файл kubeconfig из кластера Anthos GKE. (Рисунок 21, 22).

**Kubernetes Pod Template**

Name:

Namespace:

Labels:

Usage:

Pod template to inherit from:

**Containers**

**Container Template**

Name:

Docker image:

Always pull image:

Working directory:

Command to run:

Arguments to pass to the command:

Allocate pseudo-TTY:

EnvVars:

List of environment variables to set in agent pod

Рисунок 21 - Шаблон модуля Kubernetes pod для Jenkins slave

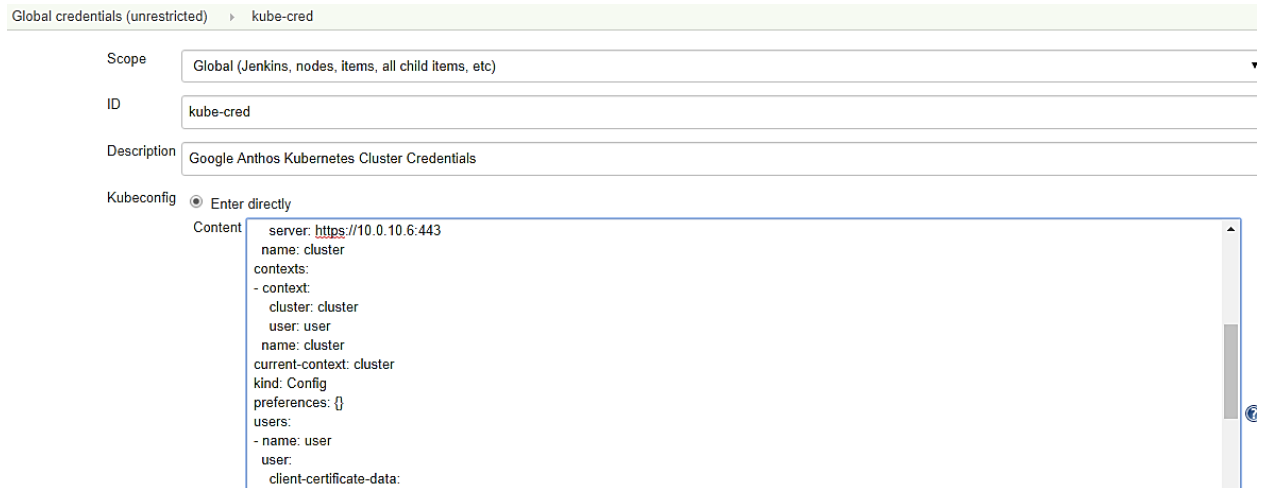


Рисунок 22 - Файл Kubeconfig для кластера GKE on-prem с Jenkins

## Интеграция Jenkins с репозиторием исходного кода

Чтобы реализовать CI-часть конвейера, необходимо интегрировать репозиторий исходного кода с Jenkins и настроить агент конвейера, который будет периодически проверять репозитории на наличие обновлений и автоматически планировать сборки. В этом примере используем Github. Пример кода приведенного ниже был клонирован в другой персональный репозиторий на Git. (Рисунок 23)

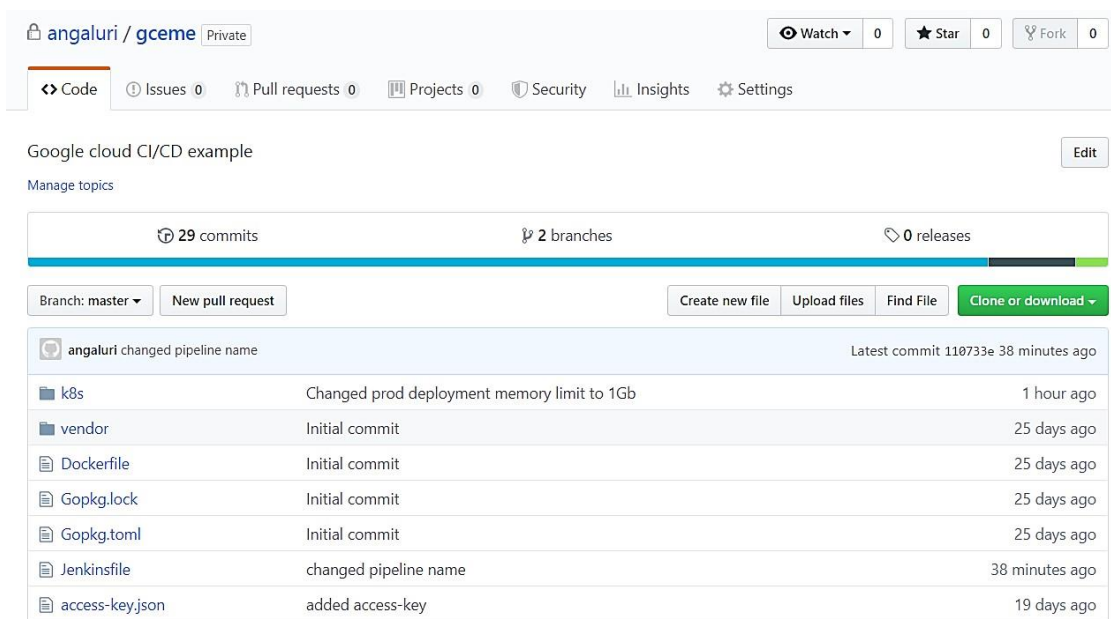


Рисунок 23 - Репозиторий Git для примера приложения CI/CD

Также нужно будет зарегистрировать свои учетные данные для репозитория Git в Jenkins, чтобы агент сборки Jenkins мог получить доступ к репозиторию и извлечь код. Кроме того, чтобы персональная рабочая станция разработчика могла извлекать и отправлять код в репозиторий Git, необходимо зарегистрировать SSH-ключи в репозитории Git и включить их.

## Создание конвейера CI / CD

В Jenkins создайте новый многоотраслевой конвейерный проект. Репозиторий Git, в котором размещен код, должен быть указан с соответствующими учетными данными. (Рисунок 24)

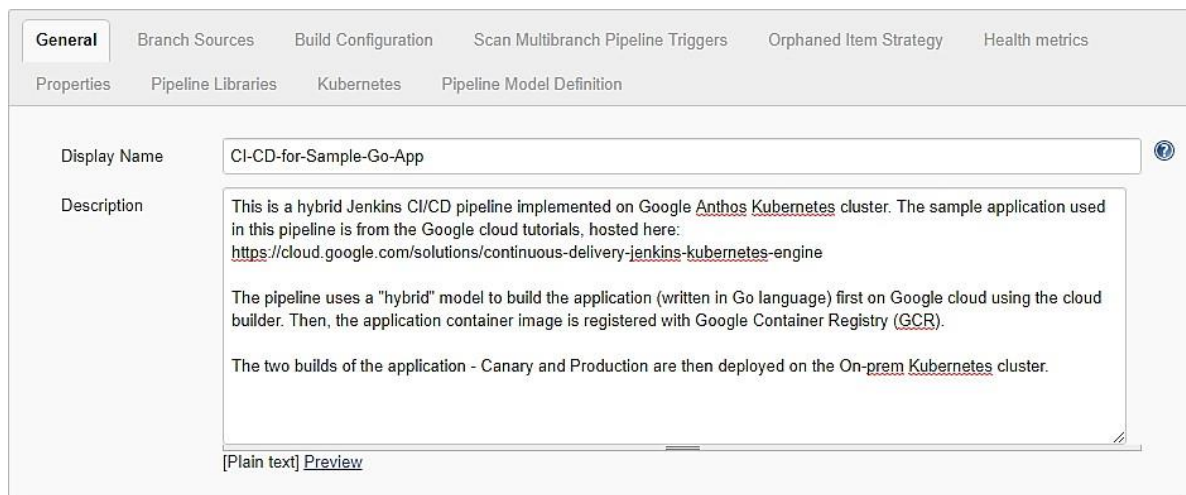


Рисунок 24 - Создание многоотраслевого конвейера для примера приложения CI/CD

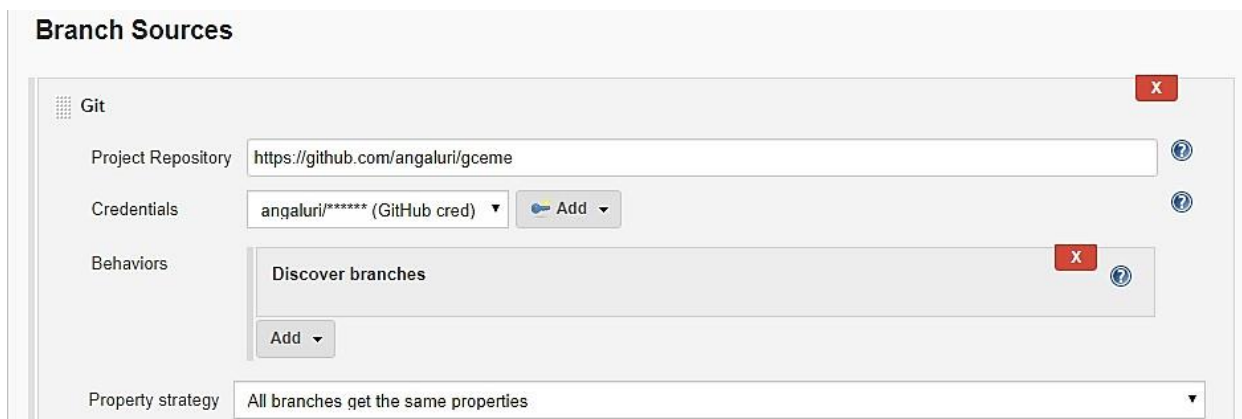


Рисунок 25 - Репозиторий Github и учетные данные для конвейера CI/CD

Для настройки непрерывной интеграции необходимо, чтобы репозиторий исходного кода периодически проверялся на наличие изменений.

Также можно настроить триггеры сборки с помощью веб-перехватов в конфигурации Git таким образом, чтобы Git запускал конвейерную сборку, когда новый код передается в репозитории. Однако, как правило, кластеры Anthos GKE будут за корпоративными брандмауэрами, которые не позволят Git web перехватывать трафик на сервер Jenkins. (Рисунок 25)

Далее настроим периодический сканер репозитория в конвейере Jenkins. (Рисунок 26). Указываем одну минуту в качестве интервала для сканирования репозитория на Git hub. Каждую минуту, Дженкинс агент



просканирует репозиторий на наличие любых обновлений кода, а затем запустит конвейерную сборку

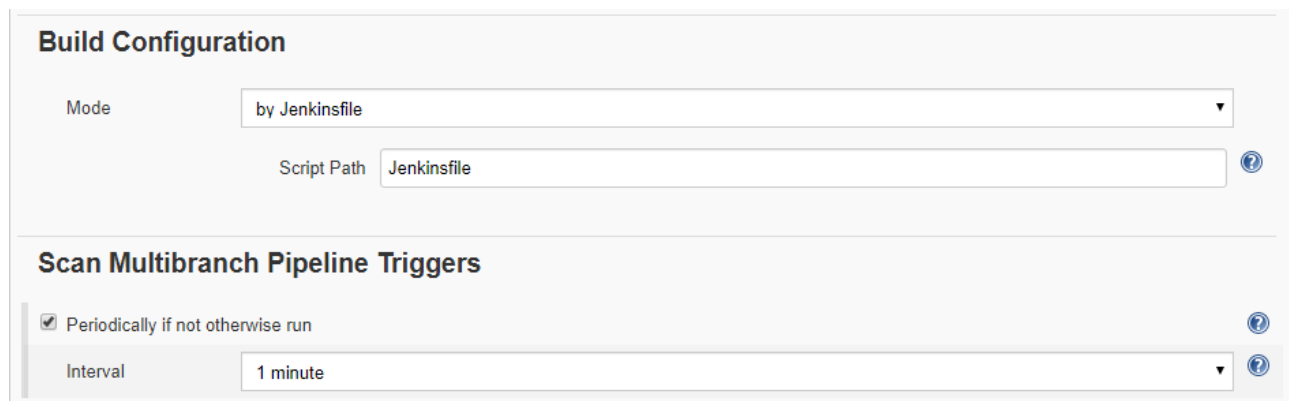


Рисунок 26 - Конвейерный сканер для запуска сборки

После создания многоветвленного конвейера можно увидеть его на панели управления Jenkins, как показано. (Рисунок 27). Можно видеть, что Дженкинс обнаружил две ветви - Canary и Master. Дженкинс запустит сборку на одной или обеих ветвях по отдельности, когда сканер кода обнаруживает изменения.

 **CI-CD-for-Sample-Go-App**

Folder name: sample-pipeline  
 This is a hybrid Jenkins CI/CD pipeline implemented on Google Anthos Kubernetes cluster. The sample application used in this pipeline is from the Google cloud tutorials, hosted here: <https://cloud.google.com/solutions/continuous-delivery-jenkins-kubernetes-engine>

The pipeline uses a "hybrid" model to build the application (written in Go language) first on Google cloud using the cloud builder. Then, the application container image is registered with Google Container Registry (GCR).

The two builds of the application - Canary and Production are then deployed on the On-prem Kubernetes cluster.

**Branches (2)**

S	W	Name ↓	Last Success	Last Failure	Last Duration
		canary	16 days - #53	16 days - #50	2 min 27 sec
		master	16 days - #31	24 sec - #32	2 min 38 sec

Рисунок 27 - Конвейер с несколькими ответвлениями для примера приложения CI/CD

### Запуск конвейерных сборок

Когда изменения в коде будут внесены и зафиксированы в соответствующей ветке, сканер конвейерной сборки увидит изменения и запускает автоматическую сборку. В приведенных ниже выходных данных внесли изменения в один из файлов YAML, создали репозиторий и внесли изменения в репозиторий origin, который находится на Github. (Рисунок 28)

```
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

```

$git commit -a -m 'Changed prod deployment memory limit
to 1Gb'
[master c7dfee6] Changed prod deployment memory limit to
1Gb
1 file changed, 1 insertion(+), 1 deletion(-)
$ git push origin master
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 588 bytes | 588.00 KiB/s,
done.
Total 5 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2
local objects.
To git+ssh://github.com/angaluri/gceme.git
41fb4b2..c7dfee6 master -> master

```

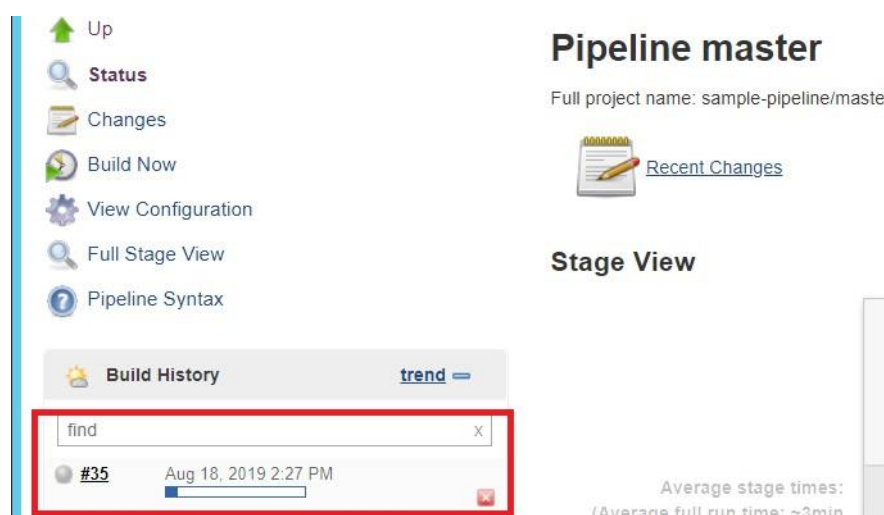


Рисунок 28 - запуск автоматической сборки после проверки кода

## Выполнение конвейера сборки на CI / CD

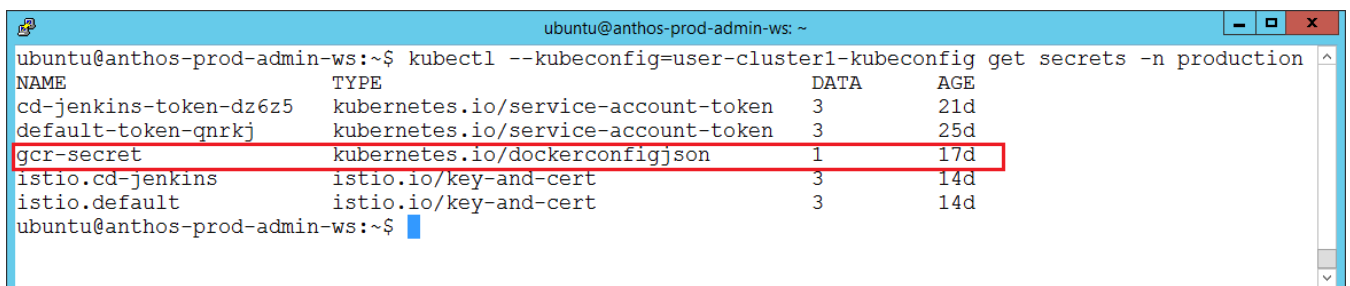
Jenkins поддерживает множество сценариев и декларативных языков для создания конвейеров. Сценарий обычно указывается в файле Jenkins и включается в исходный репозиторий или предоставляется вместе с определением конвейера. Различные этапы конвейера и соответствующие шаги по извлечению, сборке, тестированию и развертыванию кода указаны в файле Jenkinsfile.

На этапе инициализации задаются различные настройки для сборки, включая идентификатор проекта для проекта GCP, в котором зарегистрирован кластер Anthos on-prem, параметры приложения, тег сборки для образа docker для приложения, учетную запись облачной службы IAM и учетную запись

Jenkins, авторизованную для подчиненного устройства Jenkins запущен в кластере Kubernetes.

```
def project = 'srihari-gke-demo-1'
def appName = 'gceme'
def feSvcName = "${appName}-frontend"
def imageTag =
"gcr.io/${project}/${appName}:${env.BRANCH_NAME}.${env.BUILD_NUMBER}"
def gcloud_account = 'sa-connect-sri-gke-1@srihari-gke-demo-1.iam.gserviceaccount.com'
def jenkins_account = 'cd-jenkins'
```

Учетная запись Google cloud service используется для доступа к реестру контейнеров Google, куда будем помещать созданные изображения контейнеров docker. Эта учетная запись должна иметь доступ на чтение и запись к корзине облачных хранилищ Google, которая используется GCR в качестве хранилища изображений. Затем эта же учетная запись должна быть указана на более позднем этапе при развертывании кода в готовом кластере Kubernetes. Учетные данные учетной записи службы должны быть зарегистрированы в секретном хранилище cluster image pull и затем использоваться как часть определения модулей для развертывания. (Рисунок 29)



```
ubuntu@anthos-prod-admin-ws:~$ kubectl --kubeconfig=user-cluster1-kubeconfig get secrets -n production
NAME                                TYPE                                DATA  AGE
cd-jenkins-token-dz6z5              kubernetes.io/service-account-token 3      21d
default-token-qnrkj                kubernetes.io/service-account-token 3      25d
gcr-secret                          kubernetes.io/dockerconfigjson      1      17d
istio.cd-jenkins                    istio.io/key-and-cert                3      14d
istio.default                        istio.io/key-and-cert                3      14d
ubuntu@anthos-prod-admin-ws:~$
```

Рисунок 29 - Хранение учетных данных для доступа к реестру контейнеров Google

```
spec:
  containers:
  - name: backend
    image: gcr.io/cloud-solutions-images/gceme:1.0.0
  resources:
    limits:
      memory: "1000Mi"
      cpu: "100m"
    imagePullPolicy: Always
```

```

readinessProbe:
  httpGet:
    path: /healthz
    port: 8080
  command: ["sh", "-c", "app -port=8080"]
ports:
- name: backend
  containerPort: 8080
imagePullSecrets:
- name: gcr-secret

```

Ниже показан пример результатов процесса сборки конвейера.

На первом этапе Дженкинс извлекает исходный код из Git и извлекает файл Jenkins, в котором описаны этапы сборки конвейера.

```

Setting origin to https://github.com/angaluri/gceme
> git config remote.origin.url
https://github.com/angaluri/gceme # timeout=10
Fetching origin...
Fetching upstream changes from origin
> git --version # timeout=10
> git config --get remote.origin.url # timeout=10
using GIT_ASKPASS to set credentials GitHub cred
> git fetch --tags --progress origin
+refs/heads/*:refs/remotes/origin/*
Seen branch in repository origin/canary
Seen branch in repository origin/master
Seen 2 remote branches
Obtained Jenkinsfile from
110733ed0c14d360089a0f892908154853dbad6a

```

Как только файл Jenkins будет получен, начнутся этапы сборки конвейера.

На этапе тестирования будут выполнены модульные тесты (и любые другие дополнительные проверки качества) для кода. Ниже приведены выходные данные консоли в Jenkins из конвейерного выполнения.

```

[Pipeline] { (Test)
[Pipeline] container
[Pipeline] {
[Pipeline] sh
+ pwd

```

```
+ ln -s /home/jenkins/workspace/sample-pipeline_master
/go/src/sample-app
+ cd /go/src/sample-app
+ go test
PASS
ok sample-app 0.015s
```

Как только этап тестирования будет завершен и пройдет успешно, следующим этапом будет создание образа контейнера для приложения и отправка его в реестр контейнеров на GCR.

```
[Pipeline] { (Build and push image with Container
Builder)
[Pipeline] container
[Pipeline] {
[Pipeline] sh
+ PYTHONUNBUFFERED=1 gcloud builds submit -t
gcr.io/srihari-gke-demo-1/gceme:master.35 .
Creating temporary tarball archive of 34 file(s)
totalling 83.3 KiB before compression.
Uploading tarball of [.] to [gs://srihari-gke-demo-
1_cloudbuild/source/1566138521.78-
fc2917dcd2f34da3b8e792e055650e30.tgz]
Created
[https://cloudbuild.googleapis.com/v1/projects/srihari-
gke-demo-1/builds/b56222e6-4140-40c1-a5b8-61abd8b5acb5] .
Logs are available at
[https://console.cloud.google.com/gcr/builds/b56222e6-
4140-40c1-a5b8-61abd8b5acb5?project=408681909833] .
..
Successfully built 81f4bb7c3874
Successfully tagged gcr.io/srihari-gke-demo-
1/gceme:master.35
PUSH
Pushing gcr.io/srihari-gke-demo-1/gceme:master.35
DONE
```

### **Разработка микросервисов и service mesh**

Микросервисы - это архитектура, которая структурирует приложение как набор сервисов. Преимущества микросервисов заключаются в следующем:

- Микросервисы упрощают интеграцию бизнеса, процессов, технологий и персонала, разбивая монолитное приложение на меньший набор, который может обрабатываться независимо.

- Они помогают создать приложение в виде набора небольших сервисов, каждый из которых работает в своем собственном процессе и может развертываться независимо.

- Микросервисы могут быть написаны на разных языках программирования и могут использовать разные методы хранения данных.

- Микросервисы масштабируемы и гибки и подключаются через API-интерфейсы,

- Используются многие инструменты и решения многократного использования в экосистеме RESTful и веб-сервисов.

- Архитектура микросервисов обеспечивает быструю, частую и надежную доставку больших и сложных приложений.

- Позволяет организации быстро развивать свой технологический стек.

Приложения для микросервисов развертываются в виде набора контейнеров в кластере Kubernetes. Istio - это платформа service mesh для подключения микросервисов. Istio упрощает управление балансировкой нагрузки, аутентификацией от службы к службе, мониторингом и т.д. в сети сервисов. На рисунке 30 показана официальная схема архитектуры istio 1.1.

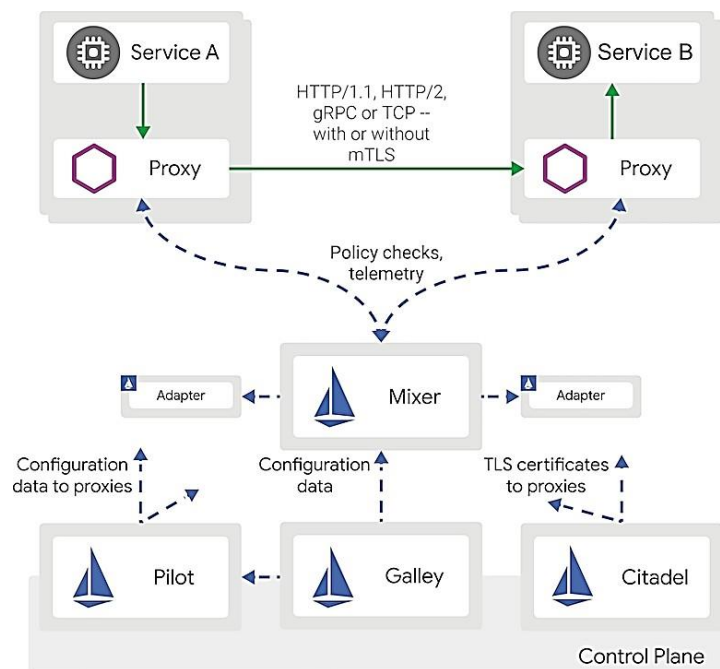


Рисунок 30 - Сетевая архитектура сервиса Istio

В среде приложения Edition используется специальный прокси-сервер `sidecar`, который перехватывает все сетевые взаимодействия между микросервисами с минимальным количеством изменений кода или вообще без изменений в коде сервиса. Это упрощает управление развертыванием микросервисов.

На рисунке 31 показана архитектура Istio, развернутая в Anthos GKE On-Prem на платформе ThinkAgile VX.

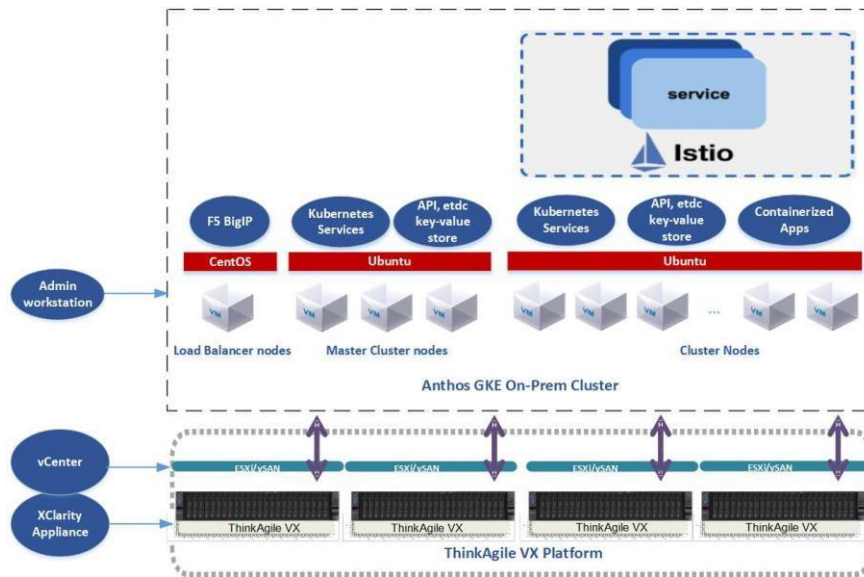


Рисунок 31 - Сервисная сетка Istio в кластере GKE on-prem

Edition устанавливается в пользовательских кластерах на платформе Anthos GKE On-Prem на платформе Lenovo Think Agile VX. Пользователи могут использовать Istio для развертывания приложений и предоставления услуг своим клиентам.

## Заключение

Данное учебное пособие о системах автоматизации управления предприятием и облачных технологиях представляет собой ценный ресурс для всех заинтересованных в эффективном управлении и развитии своего бизнеса. Представленные в нем концепции, методы и практические рекомендации отражают современные тенденции в области цифровой трансформации и подчёркивают важность внедрения инновационных решений для повышения конкурентоспособности предприятия.

Автоматизация управления предприятием и применение облачных технологий играют ключевую роль в современном бизнесе. Не только они улучшают эффективность и производительность предприятия, но и значительно повышают его конкурентоспособность. Системы управления, основанные на облачных технологиях, обеспечивают гибкость и масштабируемость, позволяя компаниям легко адаптироваться к изменяющимся рыночным условиям и требованиям клиентов.

Успешная реализация автоматизации и облачных технологий требует не только технического понимания, но и стратегического подхода. Организации должны тщательно планировать свой путь к цифровой трансформации, учитывая индивидуальные потребности и особенности своего бизнеса. Кроме того, важно обеспечить безопасность данных и учитывать риски, связанные с цифровыми технологиями.

Данное пособие послужит полезным ресурсом для руководителей и специалистов, стремящихся повысить эффективность и конкурентоспособность своих предприятий в условиях быстро меняющегося цифрового мира. Реализация автоматизации управления и использование облачных технологий могут стать ключевыми факторами успеха и дальнейшего развития бизнеса в XXI веке.

Учебное пособие предназначено для студентов образовательных программ «Информационно-коммуникационные технологии», однако может быть использовано и для других областей образования в высших учебных заведениях.



### Список использованных источников

1. Джон Арундел и Джастин Домингус "Облачный DevOps с Kubernetes: Создание, развертывание и масштабирование современных приложений в облаке"
2. William Denniss «Kubernetes for Developers», February 2024 ISBN 9781617297175
3. Билли Юэн, Александр Матюшенцев, Тодд Экенстам и Джесси Суэн «GitOps and Kubernetes», февраль 2021 г. ISBN 9781617297274 344 с.
4. Документация от Google Cloud и GitHub Портал облачных вычислений Google Cloud [Электронный ресурс]. - Режим доступа: <https://cloud.google.com>
5. Kief Morris «Infrastructure as Code: Managing Servers in the Cloud» ISBN-978-1491924358, Publisher O'Reilly Media, June 27, 2016
6. Облачные технологии [Электронный ресурс]: учеб. пособие / Никульчев Е.В., Лукьянчиков О.И., Ильин Д.Ю. — М. : РТУ МИРЭА, 2019. [https://www.researchgate.net/publication/334151736\\_Oblacnye\\_tehnologii](https://www.researchgate.net/publication/334151736_Oblacnye_tehnologii)
7. Технологии облачных вычислений : учебное пособие / И.Л. Андреевский. – СПб. : Изд-во СПбГЭУ, 2018. – 79 с.
8. Кошурин К. Облачные технологии. Основные понятия и типы облачных сервисов [Электронный ресурс]. - Режим доступа: <http://profit.kz/articles/10305/Oblachnie-tehnologii-Osnovnie-ponyatiya-itipi-oblachnih-servisov>
9. Батура Т.В., Мурзин Ф.А., Семич Д.Ф. Облачные технологии: основные понятия, задачи и тенденции развития [Электронный ресурс] // Программные продукты, системы и алгоритмы. – Режим доступа: <http://swsysweb.ru/cloud-computing-basic-concepts-problems.html> (дата обращения: 26.03.2016).
10. Кононюк А. Е. K213 Фундаментальная теория облачных технологий. — В 18-и книгах. Кн.1. —К. : Освіта України. 2018.—620 с.
11. Макарчук И. А. Построение ерп-систем на базе облачных вычислений для компаний малого и среднего бизнеса / И. А. Макарчук, О. С. Лобанов, П. П. Томша // Международный научно-исследовательский журнал.- 2015. - №4 (35). - URL: <https://research-journal.org/archive/4-35-2015-may/postroenie-erp-sistem-na-baze-oblachnyh-vychislenij-dlya-kompanij-malogo-i-srednego-biznesa>
12. Г. В. Пицхелаури, Т. М. Владимирова // Инновационные подходы к управлению в экономических, технических и правовых системах : материалы Всерос. науч.-практич. конф. с междунар. участ (Москва, 23 апр. 2024 г.) / редкол.: Н. Л. Филатова [и др.] – Чебоксары: ИД «Среда», 2024. – С. 254-257. – ISBN 978-5-907830-37-0.
13. Методологии и технологии системного проектирования информационных систем / Платова Э.Р. - Москва : ФЛИНТА, 2016. -

- [Электронный ресурс]. -  
<http://www.studentlibrary.ru/book/ISBN9785893499780.html>
14. Управление процессами. Часть 1. Подготовка бизнес-процессов к моделированию. Инструменты моделирования: учеб. пособие / Мамонова В. Г. - Новосибирск : Изд-во НГТУ, 2014. - [Электронный ресурс]. - <http://www.studentlibrary.ru/book/ISBN9785778224391.html>
15. Технология разработки программного обеспечения: учебное пособие / Зубкова Т.М. - Оренбург: ОГУ, 2017. - [Электронный ресурс]. - <http://www.studentlibrary.ru/book/ISBN9785741017852.html>