



Программное обеспечение
«Базис.Virtual Protect».
Документация жизненного
цикла

RU.ИРФЛ.00009-01

Москва
23/06/2023

Содержание

Аннотация.....	4
Введение.....	5
Идентификационные данные.....	5
Описание системы	5
Назначение.....	5
Функциональные возможности.....	5
Используемые технологии.....	5
Структура Системы	6
Процессы организационного обеспечения проекта.....	8
Процесс менеджмента модели жизненного цикла.....	8
Виды деятельности и задачи	8
Процесс менеджмента инфраструктуры.....	8
Процесс менеджмента людских ресурсов.....	9
Процесс менеджмента качества.....	10
Процессы проекта	11
Процесс планирования проекта.....	11
Оценка проекта и процесс управления	11
Процесс менеджмента решений.....	11
Процесс менеджмента рисков.....	11
Процесс менеджмента конфигурации	12
Контроль доступа к управлению конфигурацией.....	12
Процесс менеджмента информации	13
Процесс измерений	13
Технические процессы	14
Процесс определения требования правообладателей	14
Процесс анализа системных требований.....	14
Процесс проектирования архитектуры системы.....	14
Процесс реализации.....	15
Процесс комплексирования системы.....	15
Процесс квалификационного тестирования системы.....	15
Процесс инсталляции программного средства.....	15
Процесс поддержки приемки программного средства.....	15
Процедуры приемки.....	16
Процесс функционирования программного средства.....	16
Процесс сопровождения программного средства	16
Процессы реализации ПС	17
Процесс реализации программного средства.....	17
Процесс анализа требований программного средства	17
Процесс проектирования архитектуры программного средства.....	17

Процесс детального проектирования программного средства.....	17
Процесс конструирования программного средства	18
Верификация элементов конфигурации	18
Процесс комплексирования программного средства.....	18
Процесс квалификационного тестирования программного средства	18
Процессы поддержки ПС.....	20
Процесс менеджмента программной документации	20
Процесс менеджмента конфигурации	20
Процесс обеспечения гарантий качества программных средств	20
Процесс верификации программных средств	20
Верификация элементов конфигурации	21
Процесс валидации программных средств.....	21
Процесс ревизии программных средств.....	21
Процесс аудита программных средств.....	21
Процесс решения проблем в программных средствах.....	21
Процедуры устранения недостатков	22
Термины и определения.....	23
Перечень сокращений	24

Аннотация

Настоящий документ является документацией жизненного цикла и описывает процесс, который может использоваться при определении, управлении и совершенствовании процессов жизненного цикла изделия «Программное обеспечение «Базис.Virtual Protect» (далее по тексту — программное обеспечение (ПО), изделие, программное средство, ПО «Базис.Virtual Protect»). Данный документ разработан с целью удовлетворения требований к выбору средств, применяемых при разработке средства, требований к управлению конфигурацией средства, а также требований ГОСТ Р 12207-2010 «Процессы жизненного цикла программного средства», о чем свидетельствует таблица 1.

Таблица 1 – Соответствие требований разделам документа.

Содержание требования	Раздел документа
Процессы организационного обеспечения проекта	2
Процессы проекта	3
Технические процессы	4
Процессы реализации ПО	5
Процессы поддержки ПО	6

Введение

Настоящий документ предназначен для представления определенной совокупности процессов, облегчающих связи между приобретающими сторонами, поставщиками и другими правообладателями в течение жизненного цикла программного обеспечения «VirtualProtect».

Идентификационные данные

Идентификационные данные ПО	Программа для ЭВМ «Базис.Virtual Protect»
Название документа	«Программное обеспечение «Базис.Virtual Protect». Документация жизненного цикла»
Версия документа	1.0
Обозначение документа	RU.НРФЛ.00009-01
Автор документа	ООО «БАЗИС»

Описание системы

Назначение

ПО «Базис.Virtual Protect» (далее Базис.Virtual Protect, ПО, система) предназначено для управления резервным копированием кластеров kubernetes, а также восстановлением из резервных копий.

Резервному копированию и восстановлению из резервных копий доступны:

- системные компоненты Kubernetes (конфигурация самого кластера);
- рабочие нагрузки (оркестрируемые контейнеры);
- персистентные данные рабочих нагрузок.

Для хранения резервных копий используется любое S3-совместимое хранилище, что снижает риск потери данных за счет использования независимого совместимого с API S3 хранилища данных.

Функциональные возможности

Помимо прямых функций по резервному копированию и восстановлению, ПО «Базис.Virtual Protect» предоставляет пользователю следующие функциональные возможности:

- полнофункциональный графический интерфейс;
- встроенная возможность создавать S3-совместимое хранилище minIO с возможностью добавления к одному кластеру несколько независимых хранилищ;
- хранение данных в резервных копиях в сжатом виде, в определенных бакетах данного хранилища;
- автоматическое обслуживание копий архива (удаление копий, срок хранения которых истек);
- создание резервных копий нагрузок выбранных пространств имен (namespaces) и нагрузок определенных групп (groups) в кластерах K8s;
- выборочное восстановление из резервных копий нагрузок из определенных пространств имен и принадлежащим определенной группе API;
- поддержка мультикластерности, в рамках одного интерфейса, что позволяет добавлять в систему на обслуживание кластера k8s, а также удалять эти кластера из обслуживания;
- возможность интеграции с другими продуктами ООО "БАЗИС".

Используемые технологии

ПО «Базис.Virtual Protect» использует следующие технологии:

- Kubernetes – проект с открытым исходным кодом, предназначенным для управления кластером контейнеров Linux как единой системой. Kubernetes управляет и запускает контейнеры Docker на большом количестве хостов, а так же обеспечивает совместное размещение и репликацию большого количества контейнеров.
- Docker – проект с открытым исходным кодом для автоматизации развертывания приложений в виде переносимых автономных контейнеров, выполняемых в облаке или локальной среде.

- В системе реализованы два подхода для создания резервных копий PV/PVC — с использованием и без использования снапшотов из CSI;
- Расписание создания резервных копий выводится в формате CRON;
- MinIO – высокопроизводительное хранилище объектов MinIO. MinIO – высокопроизводительное решение для хранения объектов, которое предоставляет API, совместимый с S3, и поддерживает все основные функции S3. Одновременно настройки кластеров и учетные данные пользователей хранятся в документоориентированной базе mongo. Данные о резервных копиях, запланированных резервных копиях и хранилищах хранятся в целевых кластерах. Конечным, обобщающим все данные, является объектное хранилище совместимое с API S3, что позволяет осуществлять миграцию на новые кластера.
- Объектное хранилище S3 (Object storage S3) – облачный сервис, позволяющий хранить файлы любого типа и объема, используемый для хранения неструктурированных данных. В объектном хранилище файлы представлены в виде объектов. Обычно каждый объект состоит из трех основных компонентов: содержимого объекта, метаданных объекта и его идентификатора. Уникальный идентификатор позволяет быстро находить файл в хранилище и управлять им. Метаданные содержат расширенную информацию об объекте:
 - имя;
 - размер;
 - тип содержимого;
 - атрибуты жизненного цикла;
 - URL-адрес;
 - любую другую, определенную пользователем.

Метаданные позволяют управлять объектами нужного типа (выгрузка, политики хранения, удаления для определенных объектов и т. д.).

Объекты хранятся в специальных контейнерах — корзинах (buckets) с уникальным ID.

В S3 объекты хранятся в плоском адресном пространстве, как в файловом хранилище. Доступ к объектам возможен через API или HTTP/HTTPS.

Структура Системы

ПО «Базис.Virtual Protect» включает следующие программные компоненты:

- backend (написан на Golang);
- decort-go-sdk – компонент, представляющий библиотеку, написанную на языке GO, позволяющую взаимодействовать с API облачной платформы DECORT. Библиотека содержит в себе структуры и методы, необходимые для отправки запросов. Decort SDK имеет встроенный http-клиент и поддерживает разные способы авторизации на платформе. Библиотека так же содержит в себе модели ответов от платформы;
- frontend – компонент, представляющий собой фреймворк с открытым исходным кодом для создания пользовательских интерфейсов, интегрирующийся в проекты с использованием других JavaScript-библиотек.

На рисунке 1 отображена схема взаимодействия ПО «Базис.Virtual Protect» с внешними приложениями:

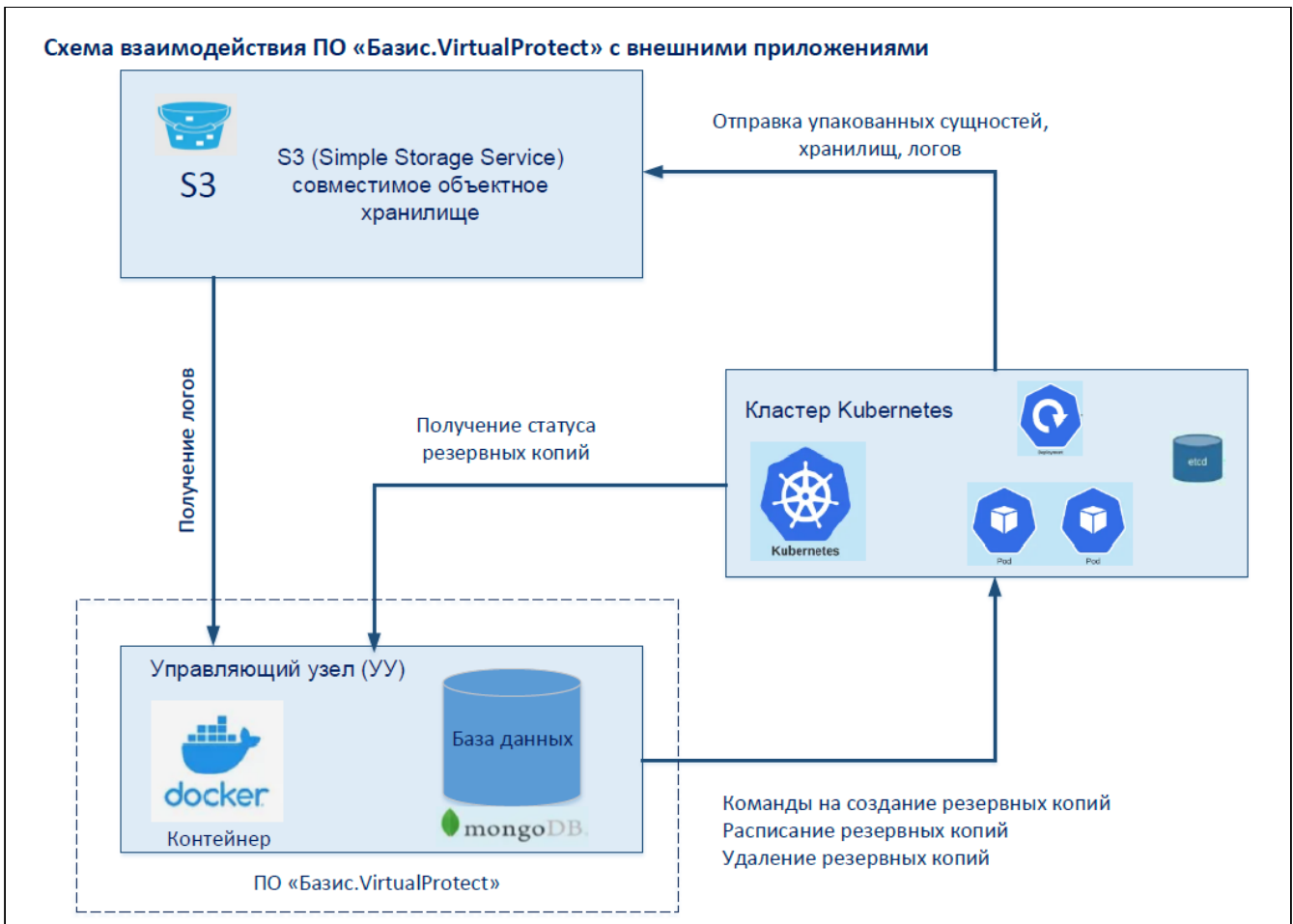


Рисунок 1 – Схема взаимодействия ПО «Базис.Virtual Protect» с внешними приложениями

Процессы организационного обеспечения проекта

Процесс менеджмента модели жизненного цикла

Цель процесса менеджмента модели жизненного цикла заключается в определении, сопровождении и обеспечении гарантии наличия политик, процессов жизненного цикла, моделей жизненного цикла и процедур для использования ООО «БАЗИС» в пределах области применения настоящего стандарта.

Данный процесс предусматривает политики, процессы и процедуры жизненного цикла, согласованные с целями ООО «БАЗИС», которые определяются, адаптируются, совершенствуются и сопровождаются для поддержки отдельных потребностей проекта в пределах задач и функций ООО и которые готовы к применению с использованием эффективных испытанных методов и инструментария.

В результате успешного осуществления процесса менеджмента модели жизненного цикла:

1. Предоставляются политики и процедуры менеджмента и развертывания моделей и процессов жизненного цикла.
2. Определяются обязанности, ответственность и полномочия менеджмента жизненного цикла.
3. Определяются, сопровождаются и совершенствуются процессы, модели и процедуры жизненного цикла для применения организацией.
4. осуществляется процесс усовершенствований в порядке установленных приоритетов.

Виды деятельности и задачи

ООО «БАЗИС» осуществляет следующие виды деятельности в соответствии с принятыми политиками и процедурами в отношении процесса менеджмента модели жизненного цикла:

- создание процессов;
- оценка процессов;
- совершенствование процессов.

Процесс менеджмента инфраструктуры

Цель процесса менеджмента инфраструктуры заключается в снабжении проекта обеспечивающей инфраструктурой и услугами для поддержки организации и целей проекта в течение всего жизненного цикла.

Данный процесс определяет, предоставляет и обслуживает средства, инструментарий, активы коммуникационных и информационных технологий, необходимые для деловой деятельности организации в соответствии с областью применения настоящего стандарта.

Для разработки ПО были использованы следующие средства разработки, представляющую инфраструктуру:

- Git 2.34.1
- Go 1.19.4
- Visual Studio Code 1.74.3
- docker 20.10.23
- node.js 16.16.0
- vue.js 3.2.37

Документация по использованным средствам разработки является общедоступной:

- <https://git-scm.com/doc> – документация по git;
- <https://go.dev/doc/> – документация по языку программирования Go;
- <https://code.visualstudio.com/docs> – редактор исходного кода с поддержкой Windows, macOS и Linux, поставляется со встроенной поддержкой JavaScript, TypeScript и Node.js, имеющий экосистему расширений для других языков и сред выполнения (таких как C ++, C #, Java, Python, PHP, Go, .NET);
- <https://docs.docker.com/> – документация по Docker;
- <https://nodejs.org/en/docs/> – справочная документация по API;
- <https://vuejs.org/guide/introduction.html> – документация по фреймворку JavaScript для создания пользовательских интерфейсов, построенному поверх стандартных HTML, CSS и JavaScript.

Информация по инструментальным средствам разработки ПО представлена в таблице 2.

Таблица 2 – Сводная информация по инструментальным средствам разработки

Система	Продукт	Версия	Разработчик
---------	---------	--------	-------------

Система хранения документов проекта	Git	2.34.1	Некоммерческий проект с открытым исходным кодом.
Система регистрации обнаруженных ошибок и проблем	Wekan и Jira	7.2	Некоммерческий проект с открытым исходным кодом. Atlassian. Коммерческий проект
Средства тестирования	Go (встроенные средства тестирования)	1.19.4	Некоммерческий проект с открытым исходным кодом.
Средства сборки продукта	Jenkins	2.204.1	Некоммерческий проект с открытым исходным кодом.
Средства разработки	Go	1.19.4	Некоммерческий проект с открытым исходным кодом.
	docker	20.10.23	Некоммерческий проект с открытым исходным кодом.
	Visual Studio Code	1.74.3	Некоммерческий проект с открытым исходным кодом.
	node.js	16.16.0	Некоммерческий проект с открытым исходным кодом.
	Git	2.34.1	Некоммерческий проект с открытым исходным кодом.
	vue.js	3.2.37	Некоммерческий проект с открытым исходным кодом.
Средства разработки документации	LibreOffice	6.0	Некоммерческий проект с открытым исходным кодом.
	Confluence	7.18.1	Atlassian. Коммерческий проект

Процесс менеджмента людских ресурсов

Целью процесса менеджмента людских ресурсов является обеспечение ООО «БАЗИС» необходимыми людскими ресурсами и поддержание их компетентности согласно потребностям деловой деятельности.

Процесс гарантирует обеспечение поддержки персонала, обладающего навыками, опытом и квалификацией для выполнения процессов жизненного цикла, направленных на достижение целей организации, проекта и заказчика.

В среде разработки ПО поддерживаются следующие роли и обязанности лиц, требуемые для выполнения операций по организационному обеспечению проекта:

- Руководитель проекта - осуществляет общее руководство разработкой, осуществляет контроль соблюдения организационно-технических мер по безопасности выпуска очередной версии.
- DevOps Инженер - отвечает за автоматизацию процессов и задач, в т.ч. автоматизацию развертывания и конфигурирования системного и прикладного ПО. Обеспечивает улучшение платформы благодаря повышению производительности.

- Системный аналитик – выполняет предварительный сбор информации, необходимой для анализа, описывает и согласовывает постановки.
- Старший технический писатель – выполняет разработку технической и проектной документации.
- Технический писатель – разрабатывает инструкции по работе с программами, оформляет техническую документацию.
- Разработчик – выполняет оценку взаимосвязи изменения части программного кода и функционирования программного модуля целиком. Отвечает за написание кода и отладку модулей программы с использованием соответствующих программных и аппаратных средств, и технологий программирования.
- Разработчик-внедренец – разрабатывает и внедряет системы автоматической проверки правильности программ, составляет технологию обработки информации. Выполняет работу по унификации и типизации вычислительных процессов.
- Разработчик-отладчик – выполняет работу по подготовке программ к отладке и проводит отладку. Определяет объем и содержание данных контрольных примеров, обеспечивающих наиболее полную проверку соответствия программ их функциональному назначению. Осуществляет запуск отлаженных программ и ввод исходных данных, определяемых условиями поставленных задач. Проводит корректировку разработанной программы на основе анализа выходных данных.
- Разработчик-администратор – выполняет работы по администрированию, резервному копированию, заполнению баз данных.
- Инженер аппаратного обеспечения – выполняет работы по организации и поддержанию работы стенда, локальной информационной системы, обеспечению бесперебойной и корректной работы информационных баз данных, вычислительной техники.
- Тестировщик – осуществляет тестирование очередной версии ПО на наличие скрытых ошибок в соответствии с тестовой документацией;
- Системный архитектор – Разработчик – выполняет анализ требований к информационной системе и отвечает за разработку архитектуры программных решений. Отвечает за написание кода и отладку модулей программы.

Процесс менеджмента качества

Целью процесса менеджмента качества является гарантия того, что продукты, услуги и реализации процессов жизненного цикла соответствуют целям ООО «БАЗИС» в области качества и удовлетворяют заказчика.

В результате успешного осуществления процесса менеджмента качества:

1. Определяются политики и процедуры в области менеджмента качества;
2. Определяются цели ООО «БАЗИС» в области качества;
3. Определяются обязанности и полномочия менеджмента качества;
4. Осуществляется мониторинг степени удовлетворенности заказчика;
5. Предпринимаются соответствующие действия, когда цели в области качества не достигаются.

Процессы проекта

Процесс планирования проекта

Цель процесса планирования проекта состоит в составлении и доведении до заинтересованных сторон эффективного и выполнимого плана.

Данный процесс определяет область применения менеджмента проекта и технических мероприятий, результаты процесса, проектные задачи и поставки, устанавливает графики для выполнения задач проекта, включая критерии достижения и ресурсы, необходимые для выполнения задач проекта.

В результате успешного осуществления процесса планирования проекта:

1. Определяется область проведения работ по проекту.
2. Оценивается возможность достижения конечных целей проекта с имеющимися ресурсами и ограничениями.
3. Определяются размеры и оцениваются задачи и ресурсы, необходимые для выполнения работы.
4. Идентифицируются интерфейсы между элементами в проекте и с другими проектами и подразделениями ООО «БАЗИС».
5. Разрабатываются планы реализации проекта.
6. Активизируются планы реализации проекта.

Оценка проекта и процесс управления

Цель оценки проекта и процесса управления заключается в определении состояния проекта и гарантии того, что проект выполняется в соответствии с планами и графиками работ в пределах бюджета и удовлетворяет техническим параметрам.

При необходимости этот процесс включает в себя переориентацию деятельности в рамках проекта, корректировку выявленных отклонений и изменений, связанных с менеджментом других проектов или с техническими процессами. Соответственно переориентация может включать в себя перепланирование.

В результате успешного осуществления оценки проекта и процесса управления:

- предпринимаются действия по корректировке отклонений от плана и для предотвращения повторения проблем, выявленных в проекте, если проектные задания не достигнуты;
- цели проекта достигаются и регистрируются.

Процесс менеджмента решений

Цель процесса менеджмента решений заключается в выборе из существующих альтернатив наиболее предпочтительного направления проектных действий.

Данный процесс является реакцией на возникающие в течение жизненного цикла системы запросы на принятие решений, направленных на достижение заданных, желаемых или оптимальных результатов вне зависимости от происхождения или источника таких запросов. Альтернативные действия анализируются, а также выбирается и указывается направление действий. Решения и их обоснования документируются для поддержки принятия будущих решений.

В результате успешного осуществления процесса менеджмента решений:

1. Определяется стратегия принятия решений.
2. Определяются альтернативные направления действий.
3. Выбирается наиболее предпочтительное направление действий.
4. Принятое решение, его обоснование и допущения документируются и доводятся до сведения заинтересованных сторон.

Процесс менеджмента рисков

Цель процесса менеджмента рисков заключается в постоянном определении, анализе, обработке и мониторинге рисков. Процесс менеджмента рисков является непрерывным процессом для систематичной адресации риска по всему жизненному циклу системного программного продукта или услуги. Это может быть применимо к рискам, связанным с приобретением, разработкой, сопровождением или применением по назначению системы.

В результате успешного осуществления процесса менеджмента рисков:

1. Определяется область применения выполняемого менеджмента рисков.
2. Определяются и выполняются соответствующие стратегии менеджмента рисков.
3. Определяются риски по мере их выявления и в течение проведения проекта.

4. Риски анализируются и определяются приоритеты использования ресурсов для обработки этих рисков.
5. Определяются, применяются и оцениваются степени риска для установления изменений состояния риска и прогресса в действиях по его обработке.
6. Предпринимается обработка для исправления или уклонения от воздействия риска, основанная на его приоритете, вероятности и последствиях или другом определенном пороговом значении риска.

Процесс менеджмента конфигурации

Цель процесса менеджмента конфигурации состоит в установлении и поддержании целостности всех идентифицированных выходных результатов проекта или процесса обеспечения доступа к ним любой заинтересованной стороны.

В результате успешного осуществления процесса менеджмента конфигурации:

1. Определяется стратегия менеджмента конфигурации.
2. Определяются составные части, нуждающиеся в менеджменте конфигурации.
3. Устанавливается базовая линия конфигурации.
4. Осуществляется управление изменениями в составных частях, находящихся под менеджментом конфигурации.
5. Осуществляется управление конфигурацией составных частей, входящих в выпуск.
6. Статус составных частей, на которые распространяется менеджмент конфигурации, становится доступным на протяжении всего жизненного цикла.

Контроль доступа к управлению конфигурацией

Процедуры доступа и внесения изменений в элементы ПО

Доступ к серверу получают пользователи в соответствии с административным решением о включении в группу разработчиков ПО. Доступа к серверу хранилища осуществляется с помощью личных сертификатов или выполнения процедуры идентификации и аутентификации.

Доступ к серверу получают пользователи в соответствии с административным решением о включении в группу разработчиков ПО. Доступа к серверу хранилища осуществляется с помощью личных сертификатов или выполнения процедуры идентификации и аутентификации.

Все элементы хранятся в Git и на рабочих станциях в виде клонов веток репозитория, разграничение доступа осуществляется с помощью учётных записей, права предоставляются по решению руководителей отдела.

Для создания локальной копии репозитория на рабочей станции разработчика необходимо выполнить команду:

```
git clone <адрес репозитория>
```

Далее система запросит имя пользователя и пароль. Доступ к активам хранилища получают только пользователи, успешно прошедшие процедуру идентификации и аутентификации.

Сервер хранилища размещен в физически охраняемом помещении. Физический доступ к серверу ограничен и регламентируется административными мерами компании изготовителя ПО.

Доступ к среде разработки осуществляется в соответствии с идентификаторами пользователей, наделенных полномочиями в соответствии с заданными ролями.

Система маркировки ветвей разработки

Разработка производится в отдельной ветви. После принятия решения о выпуске новой версии создаётся новый тег, указывающий на образ ветви на момент выпуска версии.

При разработке могут создаваться временные ветви, которые позднее интегрируются в главную ветвь и удаляются.

Выпуск новых версий

Выпущенные версии помечаются отдельными метками (иногда также используется термин «GIT-тег»). Каждая выпущенная версия проходит этап тестирования, в ходе которого проверяется базовая функциональность.

При каждой сборке продукта всегда повышается номер подвесии или номер исправления. В случае внеплановых релизов к номеру версии необходимо приписать текстовую аббревиатуру.

Журнал изменений

Для основных функциональных модулей ПО ведутся журналы, куда заносятся все изменения в репозитории.

Процесс прекращения применения программного средства

Цель процесса прекращения применения программного средства состоит в обеспечении завершения существования системного программного объекта.

Этот процесс прекращает деятельность организации по поддержке функционирования и сопровождения или деактивирует, демонтирует и удаляет поврежденные программные продукты, отправляя их в финальное состояние. В ходе данного процесса происходит уничтожение или сохранение программных элементов системы и связанных с ними продуктов обычным способом в соответствии с действующим законодательством, соглашениями, организационными ограничениями и требованиями правообладателей.

В результате успешного осуществления процесса прекращения применения программного средства:

1. Определяется стратегия прекращения применения.
2. Ограничения по прекращению применения служат в качестве входных данных к требованиям.
3. Системные программные элементы уничтожаются или сохраняются.
4. Окружающая среда оставляется в согласованном состоянии.
5. Обеспечивается доступ к записям, хранящим знания о действиях по прекращению применения, и результатам анализа долговременных воздействий.

Процесс менеджмента информации

Цель процесса менеджмента информации состоит в своевременном предоставлении заинтересованным сторонам релевантной, своевременной, полной, достоверной и, если требуется, конфиденциальной информации в течение и соответственно после завершения жизненного цикла системы.

В рамках данного процесса реализуется создание, сбор, преобразование, хранение, поиск, распространение и использование информации. Процесс управляет информацией, включая техническую, проектную, организационную, пользовательскую информацию, а также информацию, содержащуюся в соглашениях.

В результате успешного осуществления процесса менеджмента информации:

1. Определяется информация, подлежащая управлению.
2. Определяются формы представления информации.
3. Информация преобразуется и распределяется в соответствии с требованиями.
4. Документируется статус информации.
5. Информация является актуальной, полной и достоверной.
6. Информация становится доступной для уполномоченных сторон.

Процесс измерений

Цель процесса измерений заключается в сборе, анализе и составлении отчетов о данных, относящихся к разработанным продуктам и процессам, реализованным в пределах определенного организационного подразделения, для поддержки эффективного менеджмента процессов и объективной демонстрации качества этих продуктов.

В результате успешного осуществления процесса измерений:

1. Идентифицируются информационные потребности технических процессов и процессов менеджмента.
2. Идентифицируется и (или) разрабатывается соответствующая совокупность единиц измерения, управляемых информационными потребностями.
3. Определяются и планируются действия по измерениям.
4. Необходимые данные собираются, сохраняются, анализируются и интерпретируются результаты.
5. Используются информационные продукты для поддержки решений и обеспечения объективной основы для коммуникаций.
6. Оцениваются единицы измерений и процесс измерений.
7. Сведения об усовершенствованиях сообщаются владельцу процесса измерений.

Технические процессы

Процесс определения требования правообладателей

Цель процесса определения требований правообладателей состоит в выявлении требований к системе, выполнение которых может обеспечивать предоставление услуг, необходимых пользователям и другим правообладателям в заданной среде применения.

Этот процесс позволяет определять правообладателей или классы правообладателей, которые связаны с системой на протяжении всего ее жизненного цикла, а также их потребности и пожелания. В рамках процесса они анализируются и преобразуются в общую совокупность требований правообладателей, которые описывают желаемое поведение системы в процессе взаимодействия со средой применения. Она служит в качестве ссылки, по отношению к которой каждая предоставляемая услуга подвергается валидации для подтверждения того, что система полностью удовлетворяет заявленным требованиям.

В результате успешного осуществления процесса определения требований правообладателей:

1. Задаются требуемые характеристики и условия использования услуг.
2. Определяются ограничения для системных решений.
3. Достигается возможность прослеживания от требований правообладателей к правообладателям и их потребностям.
4. Описывается основа для определения системных требований.
5. Определяется основа для валидации соответствия услуг.
6. Формируется основа для ведения переговоров и заключения соглашений о поставке услуги или продукции.

Процесс анализа системных требований

Цель анализа системных требований состоит в преобразовании определенных требований правообладателей в совокупность необходимых системных технических требований, которыми будут руководствоваться в проекте системы.

В результате успешного осуществления анализа системных требований:

1. Устанавливается определенная совокупность системных функциональных и нефункциональных требований, описывающих проблему, подлежащую решению.
2. Выполняются соответствующие технические приемы оптимизации предпочитаемого проектного решения.
3. Системные требования анализируются на корректность и тестируемость.
4. Осмысливается воздействие системных требований на среду применения.
5. Требования расставляются по приоритетам, утверждаются и обновляются.
6. Устанавливается согласованность и прослеживаемость между системными требованиями и базовой линией требований заказчика.
7. Оцениваются изменения базовой линии по стоимости, графикам работ и воздействию технических решений.
8. Системные требования доводятся до сведения всех участвующих сторон и включаются в базовую линию.

Процесс проектирования архитектуры системы

Цель процесса проектирования архитектуры системы заключается в определении того, как системные требования следует распределить относительно элементов системы.

В результате успешного осуществления процесса проектирования архитектуры системы:

1. Определяется архитектурный проект системы, в соответствии с которым выполняется идентификация элементов системы и удовлетворяются заданные требования.
2. Устанавливаются функциональные и нефункциональные системные требования.
3. Требования распределяются по элементам системы.
4. Определяются внутренние и внешние интерфейсы каждого системного элемента.
5. Выполняется верификация между системными требованиями и архитектурой системы.
6. Требования, распределенные по системным элементам и их интерфейсам, становятся прослеживаемыми к базовой линии требований заказчика.
7. Поддерживается согласованность и прослеживаемость между системными требованиями и архитектурным проектом системы.
8. Системные требования, конструкция, архитектурный проект системы и их взаимосвязи отражаются в базовой линии и сообщаются всем участвующим сторонам.
9. В системный проект включается человеческий фактор, эргономические знания, технические приемы, методы и средства.
10. Определяются и выполняются действия по проектированию, ориентированные на человека.

Процесс реализации

Цель процесса реализации заключается в создании заданных элементов системы.

Процесс реализации программного средства является соответствующим примером процесса реализации, приспособленного к частным потребностям реализации программного продукта или услуги.

Процесс комплексирования системы

Цель процесса комплексирования системы заключается в объединении системных элементов (включая составные части технических и программного средства, ручные операции и другие системы, при необходимости) для производства полной системы, которая будет удовлетворять системному проекту.

В результате успешного осуществления процесса комплексирования системы:

1. Определяется стратегия комплексирования системы в соответствии с приоритетами системных требований.
2. Разрабатываются критерии для верификации соответствия с системными требованиями, распределенными по элементам системы, включая интерфейсы между ними.
3. Верифицируется комплексированная система с применением определенных критериев.
4. Разрабатывается и применяется стратегия регрессии для повторного тестирования системы в случае, если выполняются изменения.
5. Устанавливается согласованность и прослеживаемость между системным проектом и интегрированными элементами системы.
6. Конструируется комплексированная система, демонстрирующая соответствие с системным проектом.
7. Конструируется комплексированная система, демонстрирующая существование полной совокупности пригодных для применения поставляемых системных элементов.

Процесс квалификационного тестирования системы

Цель процесса квалификационного тестирования системы заключается в подтверждении того, что реализация каждого системного требования тестируется на соответствие и система готова к поставке.

В результате успешного осуществления процесса квалификационного тестирования системы:

1. Разрабатываются критерии для оценки соответствия системным требованиям.
2. Комплексированная система тестируется, используя определенные критерии.
3. Документируются результаты тестирования.
4. Гарантируется готовность системы для поставки.

Процесс инсталляции программного средства

Цель процесса инсталляции программного средства заключается в установке программного продукта, удовлетворяющего заданным требованиям, в целевую среду применения.

В результате успешного осуществления процесса инсталляции программного средства:

1. Разрабатывается стратегия инсталляции программного средства;
2. Разрабатываются критерии для инсталляции программного средства, предназначенные для демонстрации соответствия с требованиями к инсталляции программного средства;
3. Программный продукт инсталлируется в целевую среду.
4. Обеспечивается готовность программного продукта для использования в среде его применения.

Процесс поддержки приемки программного средства

Цель процесса поддержки приемки программного средства заключается в содействии приобретающей стороне в обеспечении уверенности в том, что продукт соответствует заданным требованиям.

В результате успешного осуществления процесса поддержки приемки программного средства:

1. Продукт комплектуется и поставляется приобретающей стороне.
2. Поддерживаются приемочные тесты, проводимые приобретающей стороной.
3. Продукт применяется по назначению в среде заказчика.
4. Проблемы, обнаруженные в течение приемки, идентифицируются и передаются ответственным за их решение.

Процедуры установки ПО изложены в документе RU.НПФЛ.00009-01 95 01 «Руководство администратора», RU.НПФЛ.00009-01 96 01 «Руководство по установке»

Ввод в эксплуатацию и эксплуатация ПО должны производиться в соответствии с указаниями в формуляре и эксплуатационных документах на ПО.

Процедуры приемки

План приемки состоит из следующих процедур:

- Принятие решения на уровне руководителя проекта о создании или изменении части ПО;
- Определение круга лиц, выполняющих проектирование и разработку создаваемой/модифицируемой части ПО;
- Определение необходимых ресурсов для сопровождения выполняемых работ;
- После окончания работ проводится тестирование созданной/модифицированной части ПО в составе ПО;
- Проводится технический контроль исходного кода силами испытательной лаборатории организации разработчика;
- Рассмотрение результатов тестирования на уровне руководителя проекта. Принятие решения о возможном выпуске новой версии. В этом случае производится создание новой GIT-метки для зафиксированного образа репозитория.

Сопроводительная документация на вносимые изменения в ПО хранится в Confluence.

Процесс функционирования программного средства

Цель процесса функционирования программного средства заключается в применении программного продукта в предназначенной для него среде и обеспечении поддержки заказчиков программного продукта.

В результате успешного осуществления процесса функционирования программного средства:

1. Определяется стратегия функционирования.
2. Определяются и оцениваются условия корректного функционирования программного средства в предназначенной для них среде.
3. Программные средства тестируются и настраиваются в предназначенной для них среде.
4. Программные средства функционируют в предназначенной для них среде.
5. Обеспечиваются содействие и консультации заказчикам программных продуктов в соответствии с условиями соглашения.

Процесс сопровождения программного средства

Цель процесса сопровождения программного средства заключается в обеспечении эффективной по затратам поддержки поставляемого программного продукта.

Виды деятельности по сопровождению программного средства перед поставкой включают планирование операций после поставки, обеспечения поддержки и логистики. Виды деятельности после поставки включают в себя модификацию программного средства и поддержку функционирования, такую как обучение или работа в режиме диспетчерской связи.

В результате успешного осуществления процесса сопровождения программного средства:

1. Разрабатывается стратегия сопровождения для управления модификацией и перемещением программных продуктов согласно стратегии выпусков.
2. Выявляются воздействия изменений в существующей системе на организацию, операции или интерфейсы.
3. По мере необходимости обновляется связанная с изменениями системная и программная документация.
4. Разрабатываются модифицированные продукты с соответствующими тестами, демонстрирующими, что требования не ставятся под угрозу.
5. Обновленные продукты помещаются в среду заказчика.
6. Сведения о модификации системных программного средства доводятся до всех затронутых обновлениями сторон.

Процессы реализации ПС

Процесс реализации программного средства

Цель процесса реализации программного средства заключается в создании заданных элементов системы, выполненных в виде программных продуктов или услуг.

В ходе этого процесса происходит преобразование заданных поведенческих, интерфейсных и производственных ограничений в действия, которые создают системный элемент, выполненный в виде программного продукта или услуги, известный как «программный элемент».

Результатом процесса является создание программной составной части, удовлетворяющей как требованиям к архитектурным решениям, что подтверждается посредством верификации, так и требованиям правообладателей, что подтверждается посредством валидации.

В результате успешного осуществления процесса реализации программного средства:

1. Определяется стратегия реализации.
2. Определяются ограничения по технологии реализации проекта.
3. Изготавливается программная составная часть.
4. Программная составная часть упаковывается и хранится в соответствии с соглашением о ее поставке.

Процесс анализа требований программного средства

Цель процесса анализа требований к программному средству заключается в установлении требований к программным элементам системы.

В результате успешного осуществления процесса анализа требований к программному средству:

1. Определяются требования к программным элементам системы и их интерфейсам.
2. Требования к программным средствам анализируются на корректность и тестируемость.
3. Осознается воздействие требований к программным средствам на среду функционирования.
4. Устанавливается совместимость и прослеживаемость между требованиями к программным средствам и требованиями к системе.
5. Определяются приоритеты реализации требований к программным средствам.
6. Требования к программным средствам принимаются и обновляются по мере необходимости.
7. Оцениваются изменения в требованиях к программным средствам по стоимости, графикам работ и техническим воздействиям.
8. Требования к программным средствам воплощаются в виде базовых линий и доводятся до сведения заинтересованных сторон.

Процесс проектирования архитектуры программного средства

Цель процесса проектирования архитектуры программного средства заключается в обеспечении проекта для программного средства, которые реализуются и могут быть верифицированы относительно требований.

В результате успешной реализации процесса проектирования архитектуры программного средства:

1. Разрабатывается проект архитектуры программного средства и устанавливается базовая линия, описывающая программные составные части, которые будут реализовывать требования к программным средствам.
2. Определяются внутренние и внешние интерфейсы каждой программной составной части.
3. Устанавливаются согласованность и прослеживаемость между требованиями к программным средствам и программным проектом.

Процесс детального проектирования программного средства

Цель процесса детального проектирования программного средства заключается в обеспечении проекта для программного средства, которые реализуются и могут быть верифицированы относительно установленных требований и архитектуры программного средства, а также существенным образом детализируются для последующего кодирования и тестирования.

В результате успешного осуществления процесса детального проектирования программного средства:

1. Разрабатывается детальный проект каждого программного компонента, описывающий создаваемые программные модули.
2. Определяются внешние интерфейсы каждого программного модуля и устанавливается совместимость и прослеживаемость между детальным проектированием, требованиями и проектированием архитектуры.

Процесс конструирования программного средства

Цель процесса конструирования программного средства заключается в создании исполняемых программных блоков, которые должным образом отражают проектирование программного средства.

В результате успешного осуществления процесса конструирования программного средства:

1. Определяются критерии верификации для всех программных блоков относительно требований.
2. Изготавливаются программные блоки, определенные проектом.
3. Устанавливается совместимость и прослеживаемость между программными блоками, требованиями и проектом.
4. Завершается верификация программных блоков относительно требований и проекта.

Верификация элементов конфигурации

Пользователь имеет возможность верифицировать версию ПО следующими способами:

- Пользователь имеет возможность проверить версию ПО «Базис.Cloud» в административной консоли управления.
- Версию ПО «Базис.Cloud» можно проверить в наименовании составных модулей.

Процесс комплексирования программного средства

Цель процесса комплексирования программного средства заключается в объединении программных блоков и программных компонентов, создании интегрированных программных элементов, согласованных с проектом программного средства, которые демонстрируют, что функциональные и нефункциональные требования к программным средствам удовлетворяются на полностью укомплектованной или эквивалентной ей операционной платформе.

В результате успешного осуществления процесса комплексирования программного средства:

1. Разрабатывается стратегия комплексирования для программных блоков, согласованная с программным проектом и расположенными по приоритетам требованиями к программным средствам.
2. Разрабатываются критерии верификации для программных составных частей, которые гарантируют соответствие с требованиями к программным средствам, связанными с этими составными частями.
3. Программные составные части верифицируются с использованием определенных критериев.
4. Программные составные части, определенные стратегией комплексирования, изготавливаются.
5. Регистрируются результаты комплексного тестирования.
6. Устанавливаются согласованность и прослеживаемость между программным проектом и программными составными частями.
7. Разрабатывается и применяется стратегия регрессии для повторной верификации программных составных частей при возникновении изменений в программных блоках (в том числе в соответствующих требованиях, проекте и кодах).

Процесс квалификационного тестирования программного средства

Цель процесса квалификационного тестирования программного средства заключается в подтверждении того, что комплектованный программный продукт удовлетворяет установленным требованиям.

В результате успешного осуществления процесса квалификационного тестирования программного средства:

1. Определяются критерии для комплектованных программного средства с целью демонстрации соответствия с требованиями к программным средствам.
2. Комплектованные программные средства верифицируются с использованием определенных критериев.
3. Записываются результаты тестирования.

4. Разрабатывается и применяется стратегия регрессии для повторного тестирования комплектующего программного средства при проведении изменений в программных составных частях.

Процессы поддержки ПС

Процесс менеджмента программной документации

Цель процесса менеджмента документации программных средств заключается в разработке и сопровождении зарегистрированной информации по программным средствам, созданной некоторым процессом.

В результате успешного осуществления процесса менеджмента документации программных средств:

1. Разрабатывается стратегия идентификации документации, которая реализуется в течение жизненного цикла программного продукта или услуги.
2. Определяются стандарты, которые применяются при разработке программной документации.
3. Определяется документация, которая производится процессом или проектом.
4. Указываются, рассматриваются и утверждаются содержание и цели всей документации.
5. Документация разрабатывается и делается доступной в соответствии с определенными стандартами.
6. Документация сопровождается в соответствии с определенными критериями.

Процесс менеджмента конфигурации

Цель процесса менеджмента конфигурации программных средств заключается в установлении и сопровождении целостности программных составных частей процесса или проекта и обеспечении их доступности для заинтересованных сторон.

В результате успешного осуществления процесса менеджмента конфигурации программных средств:

а) разрабатывается стратегия менеджмента конфигурации программных средств;

1. Составные части, порождаемые процессом или проектом, идентифицируются, определяются и вводятся в базовую линию.
2. Контролируются модификации и выпуски этих составных частей.
3. Обеспечивается доступность модификаций и выпусков для заинтересованных сторон.
4. Регистрируется и сообщается статус составных частей и модификаций.
5. Гарантируются завершенность и согласованность составных частей.
6. Контролируются хранение, обработка и поставка составных частей.

Процесс обеспечения гарантий качества программных средств

Цель процесса обеспечения гарантии качества программных средств заключается в предоставлении гарантии соответствия рабочей продукции и процессов предварительно определенным условиям и планам.

В результате успешного осуществления процесса гарантии качества программных средств:

1. Разрабатывается стратегия обеспечения гарантии качества.
2. Создается и поддерживается свидетельство гарантии качества.
3. Идентифицируются и регистрируются проблемы и (или) несоответствия с требованиями.
4. Верифицируется соблюдение продукцией, процессами и действиями соответствующих стандартов, процедур и требований.

Процесс верификации программных средств

Цель процесса верификации программных средств заключается в подтверждении того, что каждый программный рабочий продукт и (или) услуга процесса или проекта должным образом отражают заданные требования.

В результате успешного осуществления процесса верификации программных средств:

1. Разрабатывается и осуществляется стратегия верификации.
2. Определяются критерии верификации всех необходимых программных рабочих продуктов.
3. Выполняются требуемые действия по верификации.
4. Определяются и регистрируются дефекты.
5. Результаты верификации становятся доступными заказчику и другим заинтересованным сторонам.

Верификация элементов конфигурации

Пользователь имеет возможность верифицировать версию ПО следующими способами:

- Пользователь имеет возможность проверить версию ПО «Базис.Virtual Protect» в административной консоли управления.
- Версию ПО «Базис.Virtual Protect» можно проверить в наименовании составных модулей.

Процесс валидации программных средств

Цель процесса валидации программных средств заключается в подтверждении того, что требования выполняются для конкретного применения рабочего программного продукта.

В результате успешного осуществления процесса валидации программных средств:

1. Разрабатывается и реализуется стратегия валидации.
2. Определяются критерии валидации для всей требуемой рабочей продукции.
3. Выполняются требуемые действия по валидации.
4. Идентифицируются и регистрируются проблемы.
5. Обеспечиваются свидетельства того, что созданные рабочие программные продукты пригодны для применения по назначению.
6. Результаты действий по валидации делаются доступными заказчику и другим заинтересованным сторонам.

Процесс ревизии программных средств

Цель процесса ревизии программных средств заключается в поддержке общего понимания с правообладателями прогресса относительно целей соглашения и того, что именно необходимо сделать для помощи в обеспечении разработки продукта, удовлетворяющего правообладателей. Ревизии программных средств применяются как на уровне менеджмента проекта, так и на техническом уровне и проводятся в течение всей жизни проекта.

В результате успешного осуществления процесса ревизии программных средств:

1. Выполняются технические ревизии и ревизии менеджмента на основе потребностей проекта.
2. Оцениваются состояние и результаты действий процесса посредством ревизии деятельности.
3. Объявляются результаты ревизии всем участвующим сторонам.
4. Отслеживаются для закрытия позиции, по которым необходимо предпринимать активные действия, выявленные в результате ревизии.
5. Идентифицируются и регистрируются риски и проблемы.

Процесс аудита программных средств

Цель процесса аудита программных средств заключается в независимом определении соответствия выбранных продуктов и процессов требованиям, планам и соглашениям.

В результате успешного осуществления процесса аудита программных средств:

1. Разрабатывается и осуществляется стратегия аудита.
2. Согласно стратегии аудита, определяется соответствие отобранных рабочих программных продуктов и (или) услуг или процессов требованиям, планам и соглашениям.
3. Аудиты проводятся соответствующими независимыми сторонами.
4. Проблемы, выявленные в процессе аудита, идентифицируются, доводятся до сведения ответственных за корректирующие действия и затем решаются.

Процесс решения проблем в программных средствах

Цель процесса решения проблем в программных средствах заключается в обеспечении гарантии того, что все выявленные проблемы идентифицируются, анализируются, контролируются и подвергаются менеджменту для осуществления их решения.

В результате успешной реализации процесса решения проблем в программных средствах:

1. Разрабатывается стратегия менеджмента проблем.
2. Проблемы регистрируются, идентифицируются и классифицируются.
3. Проблемы анализируются и оцениваются для определения приемлемого решения (решений).
4. Выполняется решение проблем.
5. Проблемы отслеживаются вплоть до их закрытия.
6. Известно текущее состояние всех зафиксированных проблем.

Процедуры устранения недостатков

В период обеспечения технической поддержки продукта ООО «БАЗИС» безвозмездно устраняет все неисправности ПО при условии соблюдения Потребителем правил и условий хранения, транспортировки, монтажа и эксплуатации. Контакты службы поддержки: info@basistech.ru¹ или многоканальный телефон +7 (495) 645-68-89. В случае выявления в ПО дефектов, не связанных с нарушением правил эксплуатации, транспортирования и хранения, ПО подлежит рекламации. Рекламации предъявляются предприятию-изготовителю: ООО «БАЗИС» (109316, г. Москва, вн. тер. г. Муниципальный округ Текстильщики, проезд Остаповский, дом 22, стр. 16, этаж 2, комн. 52). Предприятие-изготовитель обязуется при получении рекламации в возможно короткий срок принять меры по устранению дефектов.

Ошибки обнаруживаются в процессе внутреннего автоматического и ручного тестирования ПО, либо в процессе эксплуатации. По мере обнаружения, ошибки заносятся в трекер Jira 7.2. Ошибки заносятся в трекер сотрудниками службы тестирования, либо сотрудниками службы поддержки. Далее ошибки ранжируются по критичности менеджерами продукта и руководителем технического направления, и соответствующие исправления распределяются по предстоящим релизам.

По мере модернизации ООО «БАЗИС» информирует Потребителя о новых возможностях ПО «Базис.Virtual Protect» и, по желанию Потребителя, производить поставку новых сертифицированных версий ПО «Базис.Virtual Protect» и программного обеспечения по электронным каналам связи. Информирование потребителей о выходе обновления производится путем отправки сообщений по электронной почте.

¹ <mailto:info@basistech.ru>

Термины и определения

Термин	Определение
CRON	Компьютерная программа в системах класса UNIX, использующаяся для периодического выполнения заданий в определённое время (автоматический запуск программ и скриптов на сервере в определённое время)
Docker	Проект с открытым исходным кодом для автоматизации развертывания приложений в виде переносимых автономных контейнеров, выполняемых в облаке или локальной среде
Golang	Компилируемый многопоточный язык программирования
Helm	Диспетчер пакетов, который упрощает настройку и развертывание приложений в кластерах Kubernetes (для разработчиков и операторов)
Kubernetes	Проект с открытым исходным кодом, предназначенным для управления кластером контейнеров Linux как единой системой. Kubernetes управляет и запускает контейнеры Docker на большом количестве хостов, а так же обеспечивает совместное размещение и репликацию большого количества контейнеров
MinIO	Высокопроизводительное хранилище объектов MinIO. MinIO - высокопроизводительное решение для хранения объектов, которое предоставляет API, совместимый с S3, и поддерживает все основные функции S3
Бакет	(англ. Bucket) – логическая сущность, которая помогает организовать хранение объектов. Название бакета используется как часть URL для доступа к данным. - хранилище объектов. Чтобы разместить файл в S3-хранилище, его необходимо поместить в бакет. Бакеты могут быть приватными или публичными и отличаются способом доступа к ним. Получить доступ к объектам в приватном контейнере может только авторизованный пользователь (администратор). Лимиты хранилища. Существует только одно ограничение — на объем занятого места
Чарт	Пакет для Helm, который содержит все определения ресурсов, необходимые для запуска приложения, инструмента или службы внутри кластера Kubernetes

Перечень сокращений

В документе использованы следующие сокращения:

Сокращение	Определение
LDAP	(англ. Lightweight Directory Access Protocol) - протокол прикладного уровня для доступа к службе каталогов
PV (Kubernetes)	PersistentVolume - постоянный объем (PV) - это часть хранилища в кластере, которая была подготовлена администратором или динамически подготовлена с использованием классов хранения. Это ресурс в кластере точно так же, как узел является ресурсом кластера. PVs - это плагины для томов, похожие на тома, но имеют жизненный цикл, независимый от любого отдельного модуля, который использует PV. Этот объект API фиксирует детали реализации хранилища, будь то NFS, iSCSI или система хранения, зависящая от облачного провайдера
PVC (Kubernetes)	PersistentVolumeClaim - запрос пользователя на хранение, похожий на модуль. Модули потребляют ресурсы узла, а PVCS потребляют ресурсы PV. Модули могут запрашивать определенные уровни ресурсов (процессор и память). Утверждения могут запрашивать определенный размер и режимы доступа (например, они могут быть подключены к ReadWriteOnce, ReadOnlyMany или ReadWriteMany, см. AccessModes)
USB	(англ. Universal Serial Bus) — «универсальная последовательная шина», последовательный интерфейс для подключения периферийных устройств к вычислительной технике
АМДЗ	Аппаратный модуль доверенной загрузки
ОТ	Охрана труда
ПО	Программное обеспечение
Пул	Логический объект СХД, объединяющий пространства нескольких физических накопителей в единое пространство хранения данных
СВТ	Средства вычислительной техники
СХД	Система хранения данных
ТБ	Техника безопасности
ЦОД	Центр обработки данных