

ООО «РусБИТех-Астра»

## **Платформа контейнеризации Боцман**

Описание функциональных характеристик программного обеспечения

Москва, 2025

## 1. Введение

Документ содержит описание функциональных характеристик программного обеспечения (ПО) платформы контейнеризации Боцман далее Боцман.

## 2. Общие сведения о программном обеспечении

Боцман позволяет управлять контейнеризированными приложениями в масштабируемой среде с минимальными затратами на настройку. Он поддерживает как государственные облака, так и частные центры обработки данных.

## 3. Область применения

**Целевая аудитория:** Разработчики ПО, DevOps-инженеры, системные администраторы и специалисты по машинному обучению.

**Области применения:** облачные вычисления, микросервисы, IT-инфраструктура больших и средних организаций.

Областью применения настоящего программного обеспечения являются любые сферы государственной или частной деятельности, автоматизирующие свою деятельность (использующие программное обеспечение при ведении деятельности).

## 4. Языки программирования

Исходный код Боцман написан на Golang;

## 5. Общее описание функциональных характеристик

**Боцман** - это платформа для автоматизации развертывания, масштабирования и управления контейнеризированными приложениями. Ключевые функциональные характеристики:

1. **Контейнерное развертывание:** Боцман позволяет управлять жизненным циклом контейнера от создания и внедрения, до масштабирования и удаления.

2. **Оркестрация контейнеров:** Боцман группирует контейнеры в одну логическую единицу. Это улучшает ресурсное использование и упрощает масштабирование.
3. **Создание кластера:** Боцман поддерживает создание множественных источников вычислений и управление ими как одной единицей, обеспечивая файлообмен между ними.
4. **Самодиагностика:** Боцман автоматически восстанавливает приложения после сбоев или проблем, таких как отключение узла или отказ программного обеспечения.
5. **Балансировка нагрузки:** Боцман распределяет трафик для обеспечения стабильности и может обнаруживать и связывать между собой контейнерные приложения с использованием DNS имени или своего собственного IP адреса.
6. **Автоматическое масштабирование:** Боцман автоматически масштабирует приложения на основе нагрузки CPU или других метрик, определенных пользователем.
7. **Управление конфигурациями и секретами:** Вместо развертывания отдельных контейнеров для управления конфигурациями и секретами, Nodus позволяет обновлять их без переразвертывания контейнеров.
8. **Хранилище:** Боцман позволяет автоматически монтировать систему хранения по выбору, будь то локальное хранилище или облачное (например, GCP, AWS).

## 6. Используемые технические средства Аппаратные требования

Минимальные требования:	Рекомендованные требования:
<b>Мастер-нода (Master Node) :</b>	
ОС Astra Linux 1.7.5.9 (x86_64) или новее	ОС Astra Linux 1.7.5.9 (x86_64) или новее
2 или более CPU	2 или более CPU
не менее 2 ГБ оперативной памяти	от 4 ГБ оперативной памяти
20 ГБ или более свободного места на диске	от 20 ГБ свободного места на диске

Рабочая нода (Worker Node) :	
ОС Astra Linux 1.7.5.9 (x86_64) или новее	ОС Astra Linux 1.7.5.9 (x86_64) или новее
1 или более CPU	1 или более CPU
не менее 1 ГБ оперативной памяти	от 4 ГБ оперативной памяти
20 ГБ или более свободного места на диске	20 ГБ или более свободного места на диске

Важно: все машины в вашем кластере должны иметь доступ к интернету для загрузки образов контейнеров.

## 7. Входные и выходные данные

### Входные данные для Kubernetes:

1. *Конфигурационные файлы:* Файлы YAML или JSON, описывающие ресурсы кластера и их настройки.
2. *Docker образы:* Бинарные файлы, содержащие все необходимое для запуска приложения.
3. *API запросы:* Межсетевые запросы, использующиеся для коммуникации между компонентами Kubernetes.

Многие из этих входных данных могут быть сгенерированы с помощью инструментов, таких как Helm (для управления конфигурацией), Docker (для создания образов) и Kubectl (для взаимодействия с API).

### Выходные данные из Kubernetes:

1. *Статус ресурсов:* Kubernetes предоставляет информацию о статусе различных ресурсов в кластере, таких как поды, узлы, сервисы и т.д., через его API и консоль управления.
2. *Логи:* Kubernetes собирает логи со всех контейнеров в кластере. Эти логи могут быть предоставлены пользователю через интерфейс командной строки или могут быть отправлены в систему управления логами.
3. *Метрики ресурсов:* Kubernetes также собирает метрики, такие как использование CPU и памяти, которые доступны через API метрик или с помощью инструментов мониторинга, таких как Prometheus.
4. *События:* Kubernetes сообщает об изменениях в системе, таких как запуск или остановка контейнеров, ошибки и другие события.

## **8. Подготовка к работе**

Состав работ по подготовке платформы контейнеризации Боцман к использованию изложен в Инструкции по установке и настройке, включая описание параметров конфигурационных файлов.