

Автор:
10.05.14 07:38 -

После 9 месяцев альфа-тестирования [объявлено](#) о выпуске первой бета-версии проекта [CoreOS](#), в рамках которого развивается не похожее на традиционные Linux-дистрибутивы серверное окружение, напоминающее по сути ChromeOS, но нацеленное на массовое развёртывание серверных систем. Бета-ветка будет развиваться параллельно и будет нацелена на стабилизацию уже реализованных в альфа-выпусках возможностей. После стабилизации альфа-выпуск будет получать статус бета-версии, т.е. каждый альфа-выпуск рассматривается как кандидат в релизы для бета-версии, а каждая бета-версия как кандидат для стабильного выпуска.

Готовые базовые образы CoreOS [подготовлены](#) для запуска с использованием PXE-загрузки, Amazon EC2, Google Compute Engine, OpenStack, VirtualBox, VMware, Vagrant и QEMU/KVM. Для обновления также можно использовать средства встроенного менеджера установки обновлений, в которой применяется техника выбора веток, похожая на каналы обновлений в Chrome (нужно [перейти](#) на beta-канал, установив в файле конфигурации настройку "GROUP=beta"). Наработки проекта [распространяются](#) под лицензией Apache 2.0.

CoreOS содержит только минимальный набор компонентов, достаточный для выполнения изолированных контейнеров (cgroups+namespaces), которые в свою очередь содержат произвольную начинку для запуска необходимых серверных приложений. По сути, в состав базовой системы входит только ядро Linux, системный менеджер systemd и ряд служебных сервисов для управления конфигурацией и установки обновлений. Системный раздел монтируется в режиме только для чтения и не изменяется в процессе работы.

Для установки обновлений используется подход ChromeOS, при котором одновременно создаётся два дисковых раздела. Один из разделов является активным, а второй используется для копирования обновления, после установки которого активным становится второй раздел, а первый остаётся для установки следующего обновления и предоставляет возможность быстрого отката изменений. Таким образом, при каждом обновлении разделы меняются местами. Обновления могут устанавливаться автоматически, по аналогии с ChromeOS. По задумке разработчиков, обновление серверов на базе CoreOS должно производиться также просто, как в настоящее время осуществляется обновление браузеров.

Автор:
10.05.14 07:38 -

Вместо традиционных пакетных менеджеров предлагается использовать преднастроенные изолированные контейнеры, содержащие все необходимые компоненты для выполнения того или иного серверного приложения. Упаковка приложений в произвольные обособленные контейнеры позволяет не задумываться об особенностях базовой ОС и свободно переносить контейнеры от одной ОС к другой и с сервера на сервер. В качестве системы управления контейнерами поддерживается [Docker](#)

, предоставляющий средства для автоматизации создания изолированных окружений для запуска произвольных процессов и возможности по переносу и клонированию окружений на другие серверы. При этом Docker не является обязательным, присутствует возможность создания контейнеров вручную или использования уже готовых образов, пригодных для использования с инструментарием LXC.

Другой особенностью CoreOS являются средства автоматического определения доступных сервисов, использования единой конфигурации для группы серверов и объединения набора серверов во взаимосвязанные кластерные системы. Для обмена и управления конфигурацией используется система [etcd](#), развиваемая специально для CoreOS. Код etcd написан на языке Go и поставляется под лицензией Apache. Etcd представляет собой высоконадёжное хранилище параметров конфигурации в форме ключ/значение. Для доступа к конфигурации предоставляется простой интерфейс, основанный на использовании HTTP и JSON (запросы могут отправляться при помощи утилиты curl или специальной утилиты etcdctl). Аутентификация выполняется на основе SSL-ключей. Хранилище конфигурации и логи реплицируются на все узлы и поддерживается в синхронизированном состоянии с использованием протокола [Raft](#)

На этапе инициализации задействована система [CloudInit](#), позволяющая задавать конфигурацию окружения на этапе загрузки в отдельном файле cloud-config.yml. В том числе, через заданные в cloud-config.yml параметры можно создавать пользователей, добавлять ssh-ключи, записывать юниты systemd и контролировать их запуск, создавать произвольные файлы в ФС, передавать параметры сети. Таким образом общий загрузочный образ может быть унифицирован и не привязан к конкретным конфигурациям.

В новом выпуске представлена подсистема Locksmith, позволяющая управлять перезапусками после применения обновлений. Среди предлагаемых стратегий:

- reboot - немедленная перезагрузка после получения обновления;

Автор:

10.05.14 07:38 -

- etcd-lock - перезагрузка после получения распределённой блокировки etcd, что позволяет гарантировать, что в один момент времени в состоянии перезагрузки будет находиться не более одного хоста в кластере. Метод позволяет избежать ситуации, когда несколько узлов одновременно получают обновление и уйдут в перезагрузку, нарушив работоспособность кластера (в режиме etcd-lock узлы будут перезагружаться по цепочке);
- best-effort (по умолчанию) - при выполнении etcd использование метода "etcd-lock", а в других случаях метода "reboot";
- off - отключение автоматической перезагрузки.

Read more <http://www.opennet.ru/opennews/art.shtml?num=39744>