

# Программирование в UNIX, лекция 1: “горячее” обновление контейнеров

Пожидаев М. С.

17 февраля 2021 г.

Компоненты рассматриваемой архитектуры:

1. Контроллер: центральный узел, получающий команды от внешних клиентских соединений.
2. Брокер сообщений: диспетчер обмена сообщениями между контроллером и остальными контейнерами.
3. Контейнеры-работники: динамически изменяемое множество контейнеров разных типов для выполнения работ, необходимых для обслуживания запросов.

# Задача “горячего” обновления

Требования к решению задачи “горячих” обновлений (rolling updates):

1. Контейнеры, не являющиеся контроллером и брокером, периодически получают обновления. Чем чаще, тем лучше.
2. Работу сервиса останавливать нельзя, пользователи должны наблюдать непрерывное функционирование с нормальной обработкой запросов.
3. Близкой является задача динамического изменения набора контейнеров-работников для увеличения или уменьшения задействованных вычислительных мощностей.

# Порядок выполнения обновления

1. Механизм реализуется в непосредственной связи с используемым брокером, выполняющим обслуживание межконтейнерного взаимодействия. Именно он чаще всего хранит информацию о конфигурации системы.
2. Инструмент оркестрации получает команду изменить конфигурацию системы, определяет набор контейнеров, которые должны быть остановлены.
3. Контейнеры получают сигнал завершить свою работу (SIGTERM), после чего уведомляют брокер о своём новом статусе.
4. Контейнеры завершают выполнение тех задач, которые были в работе на момент получения сигнала завершить работу.
5. Контейнеры либо добровольно завершают свою работу, либо их убивают по timeout при помощи SIGKILL.

Основная команда обновления конфигурации:

```
docker service update --image
```

Пара значимых параметров:

- ▶ `update-delay`: задержка между успешными обновлениями контейнеров;
- ▶ `stop-grace-period`: задержка после отправления SIGTERM до отправления SIGKILL.

# Запуск контейнера

1. Брокер должен быть уведомлён о готовности контейнера не ранее того момента, когда контейнер фактически становится готовым принять задачи для обработки. Этот момент наступает с задержкой после старта контейнера.
2. Контейнер должен корректно идентифицировать себя в системе, чтобы точно передать эту информацию брокеру. Проще всего это сделать на основе содержимого `/etc/hosts`.

# Обработка сигнала завершения работы

Java:

```
Runtime.getRuntime()  
.addShutdownHook(new Thread(()->{...}));
```

Python с AIOHttp:

```
app.on_shutdown.append(on_shutdown)
```

Спасибо за внимание!

Веб-сайт: <http://marigostra.ru/>

E-mail: [mSP@luwrain.org](mailto:mSP@luwrain.org)