



Онлайн-образование



Меня хорошо видно && слышно?

Ставьте , если все хорошо
Напишите в чат, если есть проблемы

НЕ ЗАБЫТЬ ВКЛЮЧИТЬ
ЗАПИСЬ!!!

Linux High Availability. Pacemaker.

Цель занятия

- Ответить на вопрос: как наиболее просто реализовать высокую доступность сервисов на linux?

План занятия

- Базовая терминология;
- Расетакер и его архитектура;
- Простейший НА-кластер на базе Расетакер.

Термины. Свойства систем

- Отказоустойчивость (fault tolerance) - свойство системы сохранять свою работоспособность после отказа одного или нескольких её компонентов.

Термины. Свойства систем

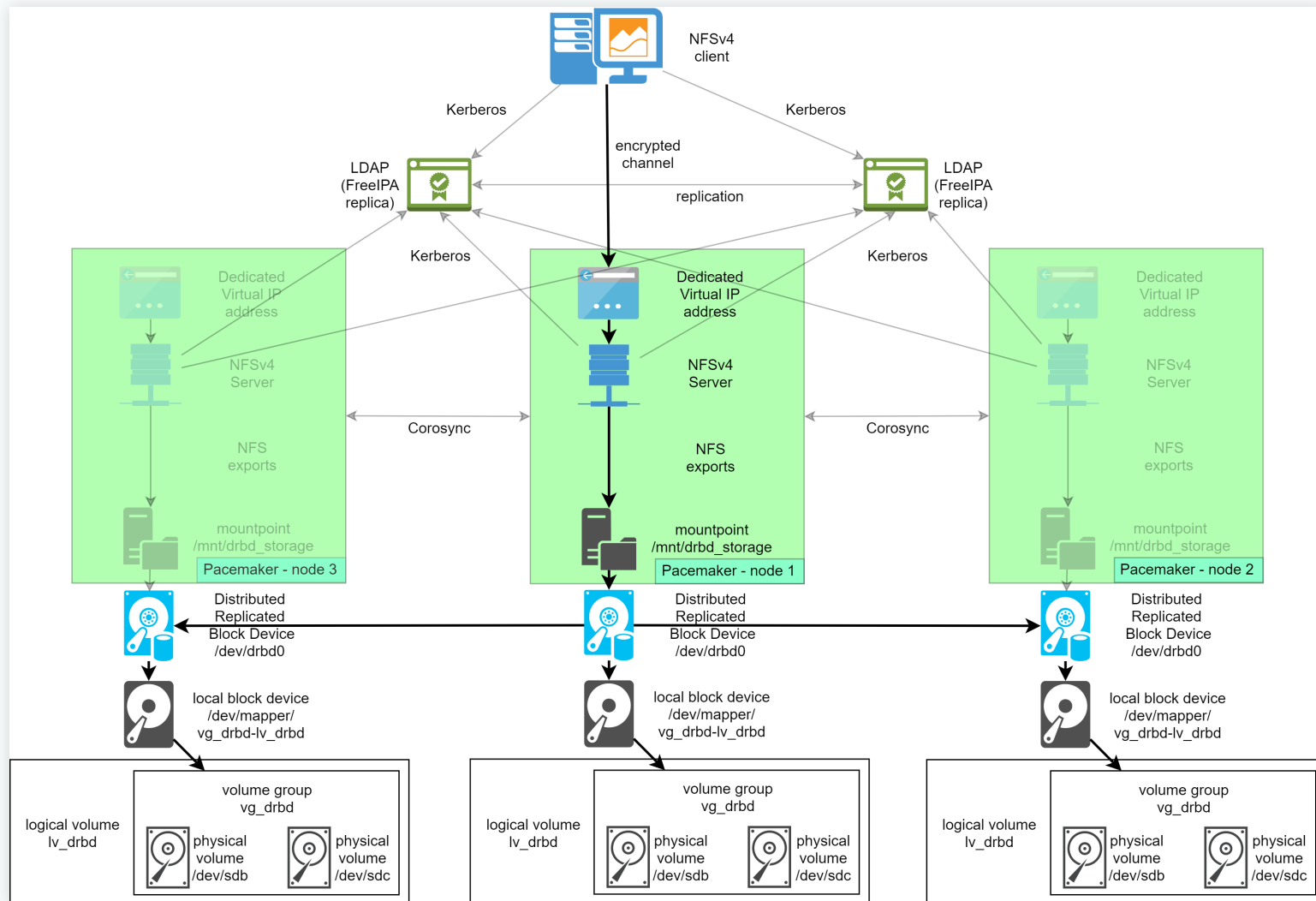
Уровни доступности:

- Высокая доступность (high availability) - свойство системы, которое достигается избеганием(!) невыполненного обслуживания клиентов путём управления сбоями и минимизацией времени плановых/внеплановых простоев.
- Непрерывный режим работы (continuous operations) - свойство системы, которое достигается отсутствием невыполненного обслуживания и отсутствием плановых простоев (но не внеплановых).
- Постоянная доступность (continuous availability) - свойство системы, которое достигается отсутствием невыполненного обслуживания и отсутствием как плановых, так и внеплановых простоев. Достигается одновременным использованием методов высокой доступности и непрерывного режима работы.

Кластеры

- Кластер - группа компьютеров, объединённых высокоскоростными каналами связи, представляющая с точки зрения пользователя единый ресурс.
- Отказоустойчивый кластер (Fault tolerant cluster) - кластер, в котором отказ сервера или приложения не приводит к полной неработоспособности всего кластера или недоступности приложения.
- Кластер высокой доступности (High available cluster) - такой кластер, в котором отказ сервера или приложения не приводит к полной неработоспособности всего кластера, а недоступное приложение (или группа приложений, если вышел из строя целый сервер) будет запущено на других узлах такого кластера.

Пример HA-кластера



Расетmaker. Архитектура

Менеджер ресурсов кластера высокой доступности:

- Распределяет ресурсы по узлам кластера;
- Отрабатывает сбои ресурсов и узлов;
- "Ограждает" сбойные узлы от размещения на них ресурсов.

Ресурс (resource) - это сервис, доступность которого поддерживается кластером.

Расетmaker. Архитектура

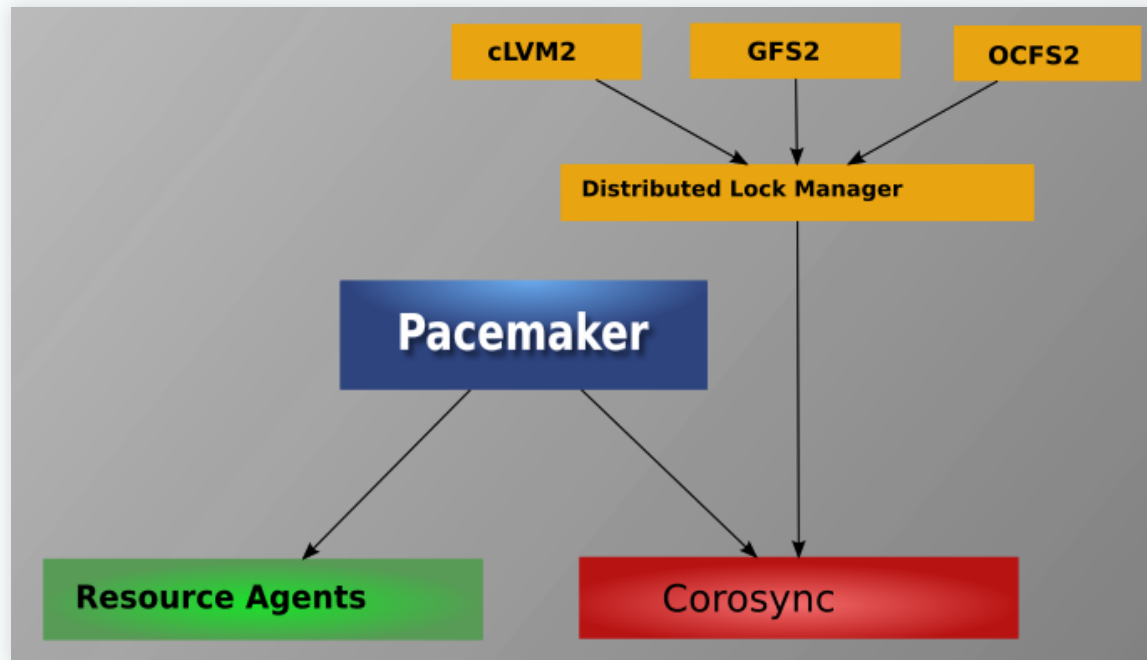
Ресурсные агенты (Resource agents) - скрипты и утилиты для управления ресурсами и контроля над ними.

- Open Cluster Framework (OCF)
- Linux Standard Base (LSB)
- Systemd
- Upstart
- System (LSB init script > Systemd unit > Upstart job)
- Fence (STONITH)
- Nagios plugin

```
pcs resource list  
ls /usr/lib/ocf/
```

Расетmaker. Архитектура

- менеджер ресурсов
- кластеронезависимый уровень
- информационная шина



Расетmaker. Архитектура

Corosync обеспечивает сетевое взаимодействие между узлами кластера:

- состав кластера, кворум;
- сервисные сообщения.

Кворум достижим, если работает более половины узлов.

Сейчас в качестве информационной шины поддерживается только Corosync.

Corosync Cluster Engine

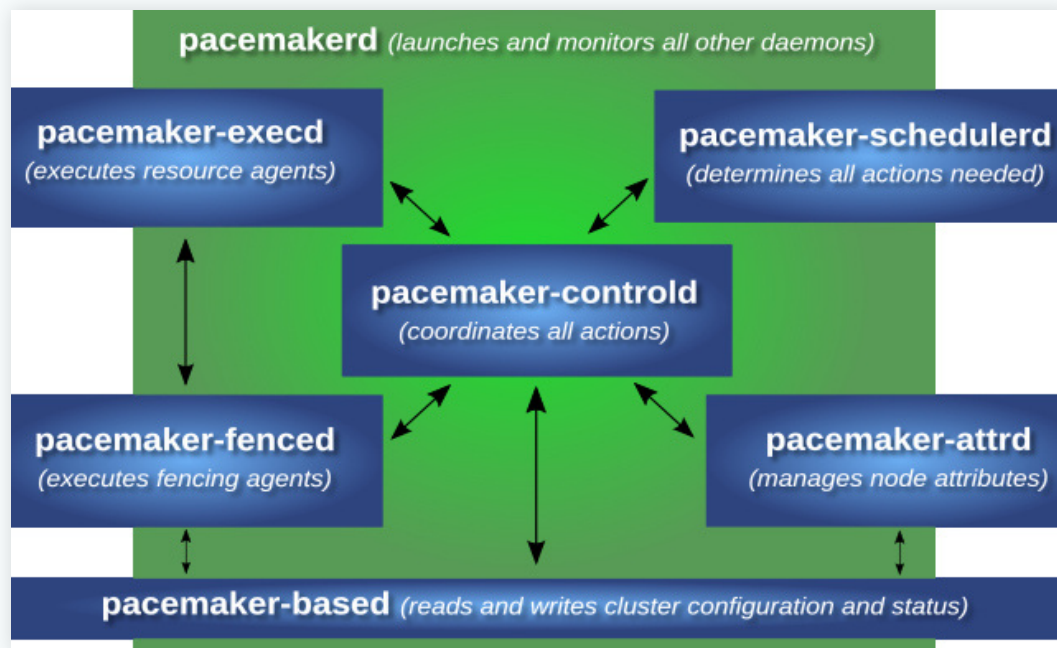
Предоставляет несколько API для приложений. С их помощью обеспечивается обмен информацией (CPG), получение статуса кворума (quorum), отслеживание состояния приложений (sam) и работа с самим Corosync (confdb).

При этом сам Corosync использует протокол "Totem Single Ring Ordering and Membership", межпроцессное взаимодействие через shared memory, хранение объектов в in-memory database, собственную систему маршрутизации сетевых (kronosnet) и межпроцессных сообщений.

```
man corosync.conf
more /etc/corosync/corosync.conf
tail -n 100 /var/log/cluster/corosync.log
corosync-keygen
more /etc/pacemaker/authkey
```

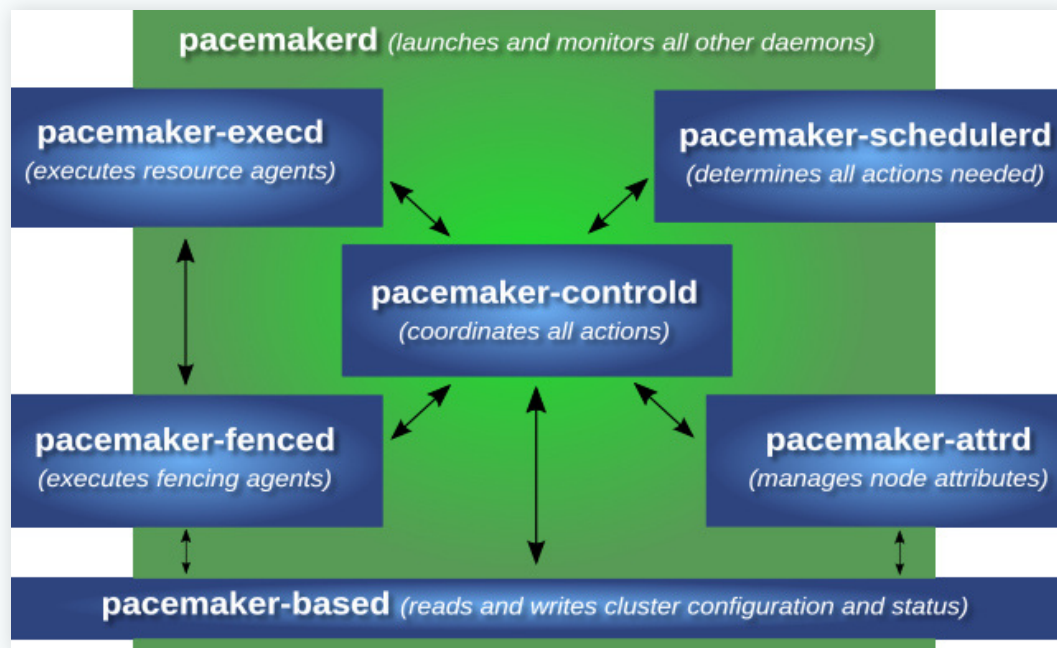
Расemaker. Демоны

- **racemakerd** - запускает остальные процессы;
- **racemaker-based (cib)** - хранит Cluster Information Base и обрабатывает запросы на изменение;
- **racemaker-attd (attd)** - обслуживает базу атрибутов, синхронизирует её и обрабатывает запросы на изменение;
- **racemaker-schedulerd (scheduler)** - определяет какие действия нужно сделать для получения желаемого состояния кластера.



Расemaker. Демоны

- **pacemaker-controld (controller)** - координатор, поддерживает членство в кластере, управляет всеми компонентами и состоянием кластера вообще, если он Designated Controller (DC);
- **pacemaker-execd (lrmd)** - обслуживает запросы на выполнение локальных ресурсных агентов и возвращает результат;
- **pacemaker-fenced (stonithd)** - обслуживает запросы на "ограждение" (как локальные, так и удалённые).

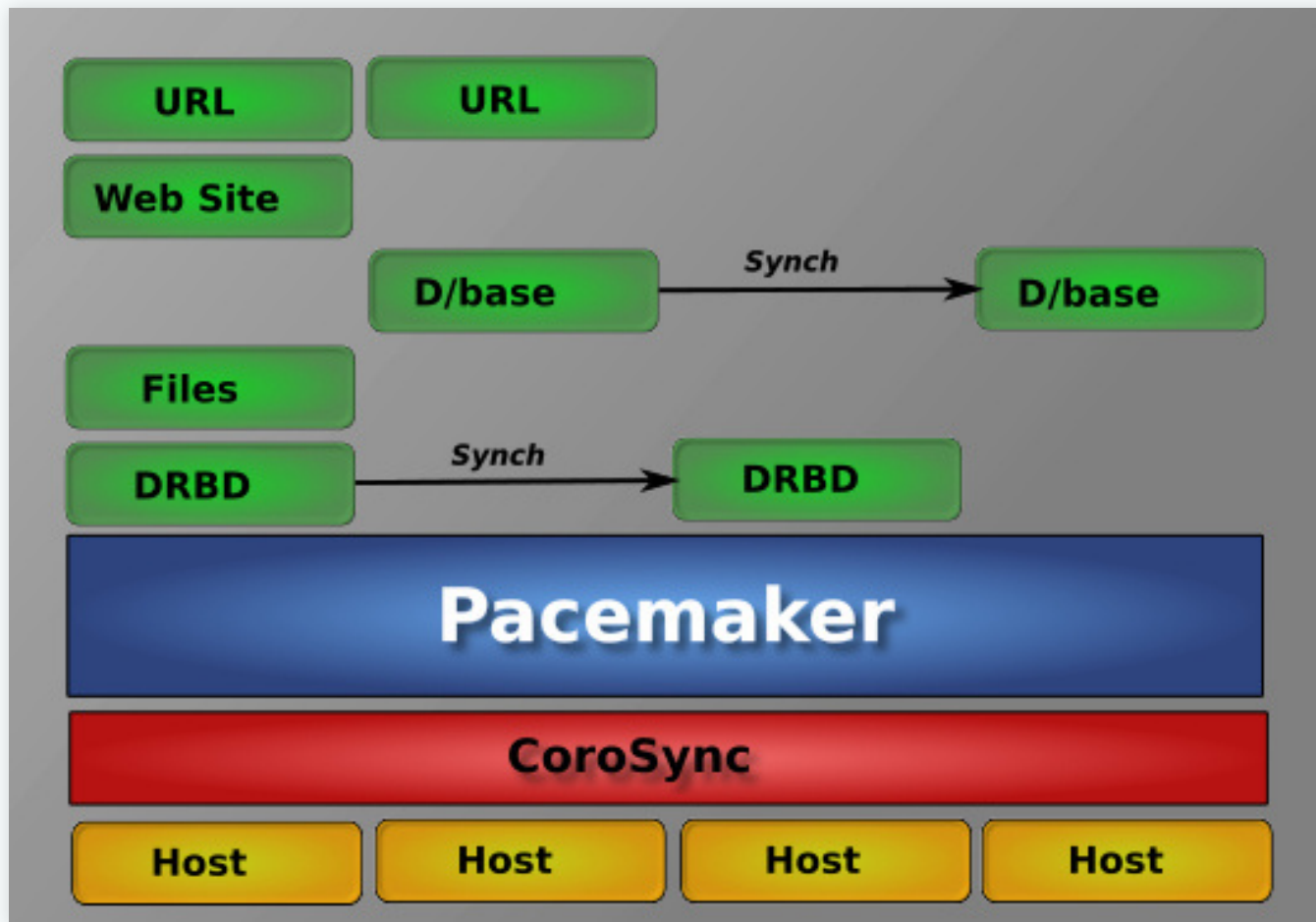


Расетmaker. Свойства

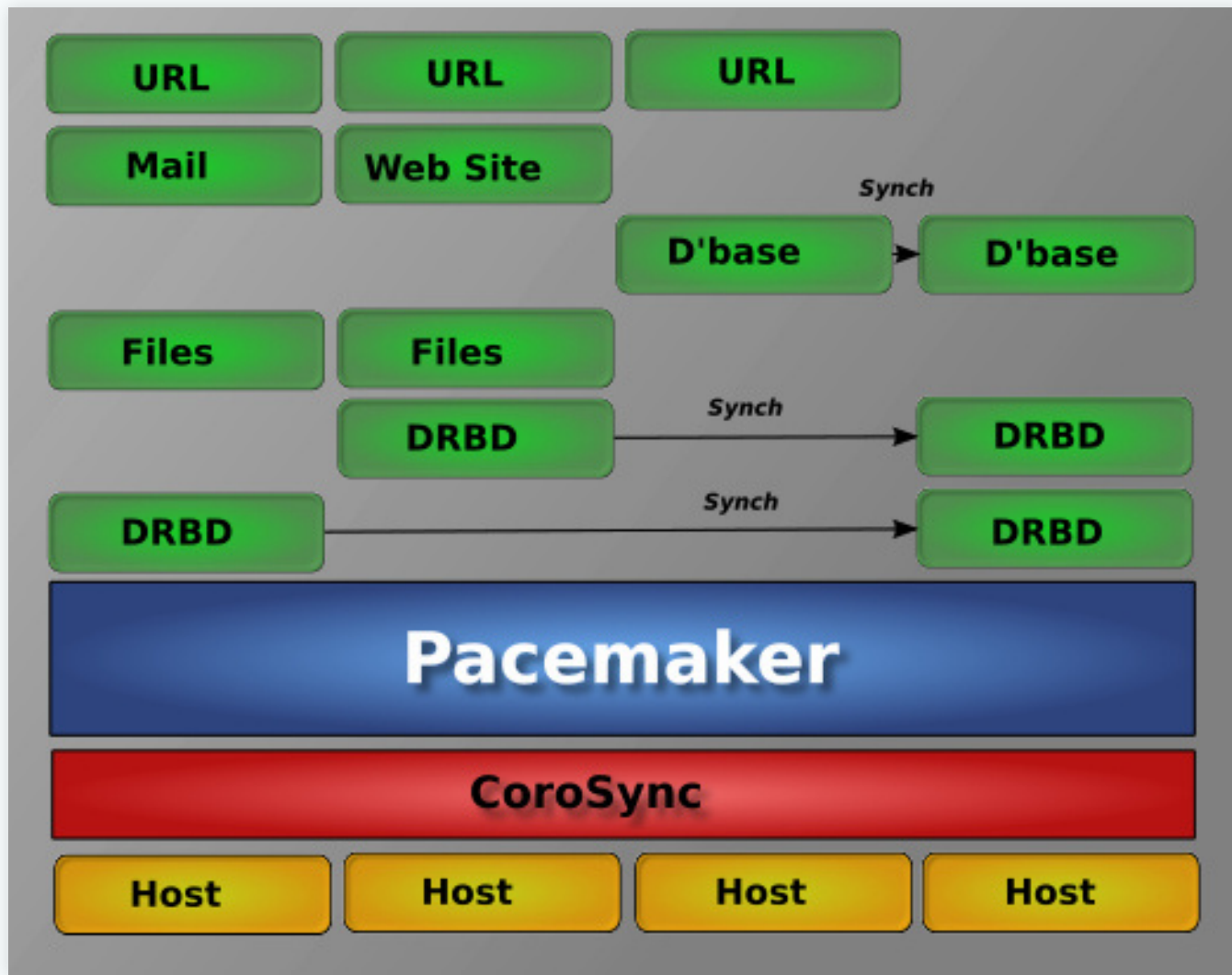
- Обнаруживает и отрабатывает сбои на уровне узлов и сервисов;
- Ограждает сбойные узлы во избежание повреждения чего-либо;
- Всё, что может быть заскриптовано, может быть кластеризовано;
- Не требует общего хранилища данных для своей работы;
- Автоматическая репликация конфигурации на все узлы кластера;
- Унифицированное средство управления кластером (pcs);
- Поддерживает множество видов отказоустойчивых конфигураций;
- Поддерживает расширенные типы ресурсов: клоны и ресурсы с состояниями;
- Гибкая система настройки взаимоотношений между ресурсами.

```
man pcs
```


Pacemaker. Active/Backup



Pacemaker. Shared failover



Pacemaker. Active/Active (N to N)



Расетmaker. Типы ресурсов

- Primitive - простой ресурс, например, nginx или IP адрес (можно объединять в группы);
- Clone - ресурс, который может выполняться на нескольких узлах одновременно;
- Promotable clone - как Clone, но имеет состояние;
- Bundle - контейнеры docker, podman и rkt.

```
pcs config
pcs resource list
pcs resource show NGINX
pcs resource describe ocf:linbit:drbd
```

Расетmaker. Атрибуты ресурсов

- Метаатрибуты - применимы кластером к любому ресурсному агенту определенного типа ресурса;
 - priority (0)
 - target-role (Started)
 - maintenance (False)
 - resource-stickiness (0 для всех, 1 для клонов)
 - migration-threshold (INFINITY)
 - multiple-active (stop_start)
- Атрибуты, специфичные для конкретного агента.

```
pcs resource defaults resource-stickiness=100
pcs resource op defaults timeout=10s
pcs resource describe ocf:linbit:drbd
```


Расemaker. Constraints (ограничения)

Можно наложить на ограничения на запуск определенных ресурсов:

- location constraints - на каких узлах можно размещать ресурс;
- order constraints - в каком порядке размещать ресурсы на узле;
- colocation constraints - ограничение запуска определенных ресурсов на одном узле;

Политика задаётся числовым значением от $-\text{INFINITY}$ до INFINITY .

Строим кластер. Перед началом

На всех узлах должны быть настроены:

- DNS (как минимум /etc/hosts);
- Синхронизация времени (chronyd, ntpd, etc.);
- Firewall.

```
more /usr/lib/firewalld/services/high-availability.xml  
firewall-cmd --permanent --add-service=high-availability
```

Строим кластер

Установка Pacemaker

```
yum install -y pacemaker pcs fence-agents-all  
systemctl enable pcsd.service --now  
passwd hacluster
```

Настройка Corosync

```
pcs cluster auth node-1 node-2 node-3  
pcs cluster setup --name otuscluster node-1 node-2 node-3  
pcs cluster enable --all  
pcs cluster start --all
```

Диагностические команды

```
corosync-cfgtool -s  
corosync-cmapctl | grep members  
pacemakerd --features  
pcs cluster cib  
pcs cluster status  
pcs status  
crm_verify -L -V  
tail /var/log/pacemaker.log  
tail /var/log/pcsd/pcsd.log  
tail /var/log/cluster/corosync.log
```

Настраиваем fence agents

```
pcs stonith list
pcs stonith describe fence_vbox
pcs cluster cib tmp_cfg
pcs -f tmp_cfg stonith create pcs1_fence_dev fence_vbox \
  ipaddr="{{ virtualbox_host }}" \
  login="{{ virtualbox_host_username }}" \
  passwd="{{ virtualbox_host_password }}" \
  power_wait="10" secure="1" port="pcs1" \
  pcmk_host_list="pcs1.mydomain"
pcs -f tmp_cfg constraint location pcs1_fence_dev \
  avoids pcs1.mydomain
pcs cluster cib-push tmp_cfg --config
```

На производственных системах выключать STONITH НЕ РЕКОМЕНДУЕТСЯ:

```
pcs property set stonith-enabled=false
```


Добавляем простой ресурс

```
pcs resource list
pcs resource standards
pcs resource providers
pcs resource agents ocf:heartbeat
pcs resource describe ocf:heartbeat:IPaddr2
pcs cluster cib tmp_cfg
pcs -f tmp_cfg resource defaults resource-stickiness=100
pcs -f tmp_cfg resource op defaults timeout=10s
pcs -f tmp_cfg resource create ClusterIP ocf:heartbeat:IPaddr2 \
    nic=eth0 ip=10.0.1.2 cidr_netmask=32 \
    op monitor interval=30s

pcs cluster cib-push tmp_cfg --config
```

Добавляем ещё ресурс и ограничения

```
pcs resource describe ocf:heartbeat:nginx
pcs cluster cib tmp_cfg
pcs -f tmp_cfg resource create NGINX ocf:heartbeat:nginx \
    configfile=/etc/nginx/nginx.conf \
    op monitor interval=5s timeout=20s

pcs -f tmp_cfg constraint colocation \
    add NGINX with ClusterIP INFINITY

pcs -f tmp_cfg constraint order \
    start ClusterIP then start NGINX

pcs cluster cib-push tmp_cfg --config
```

Простейший кластер высокой доступности
построен и готов к работе!

Полезные ссылки

- Pacemaker. Домашняя страничка проекта
- HA Addon for RHEL7
- Демостенд pacemaker под virtualbox
- Fence agents от ClusterLab
- Corosync. Домашняя страничка проекта
- Не забываем про man. Актуальная информация по конкретной используемой версии ПО доступна там.

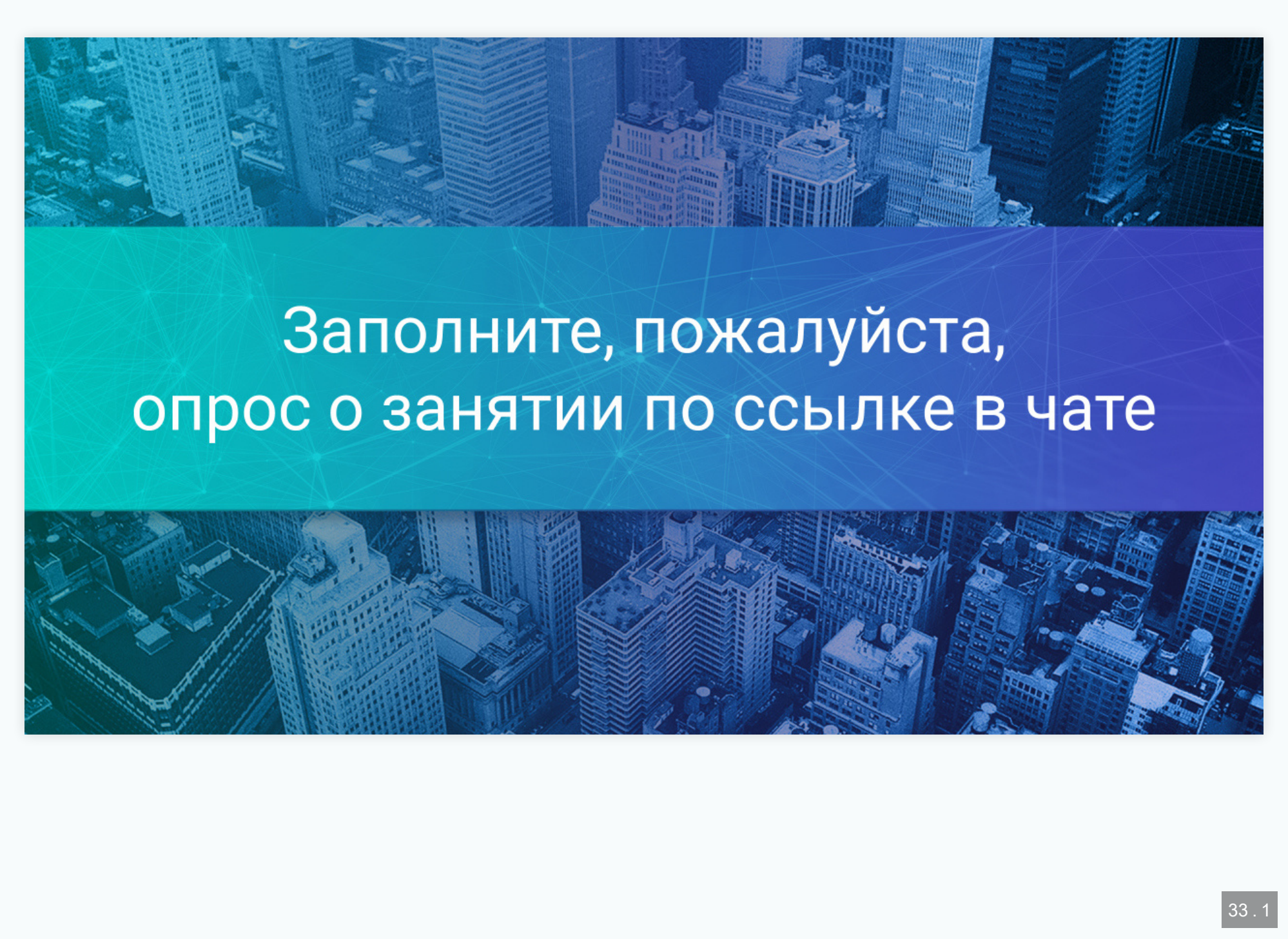
Рефлексия



Отметьте 3 пункта, которые вам запомнились с вебинара



Что вы будете применять в работе из сегодняшнего вебинара?

The background of the slide is an aerial photograph of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer that features a white network pattern of interconnected dots and lines. The text is centered within this blue area.

Заполните, пожалуйста,
опрос о занятии по ссылке в чате