

Runbook: восстановление tpark-it.* после boot тестового srv-app.olvr

Область: только тестовый srv-app.olvr. На production это решение не распространяется.

Источник и решение: [OLV-4169](#), Card 80NT (причинный разбор YiyxBP, решение Бориса 9fsx0A, позиция архитектора MA1sXZ). Эксплуатационное закрепление — Card a0V6. Существующий reconciler принят штатным механизмом; требование «без infra-reconciler» снято.

Причина и механизм

Standalone-контейнеры tpark-it.*, созданные через `docker run --restart=always`, подключены к swarm-overlay. При boot dockerd делает одну попытку их запуска после инициализации swarm. Если attach к overlay не завершился, процесс приложения не запускается: контейнер остаётся Exited с exit 255 и RestartCount=0; повтор этой boot-попытки не выполняется. Retry RabbitMQ/БД внутри приложения этот отказ не устраняет.

Существующий владелец восстановления:

- unit: /etc/systemd/system/tpark-it-reconcile.service;
- script: /usr/local/sbin/tpark-it-reconcile.sh, интерпретатор /bin/bash;
- After=docker.service network-online.target, Requires=docker.service;
- Type=oneshot, RemainAfterExit=yes, TimeoutStartSec=600;
- включён в multi-user.target; 24 раунда с паузой 15 секунд, только Exited-контейнеры по name=^/tpark-it..

Историческое наблюдение сохранено в 80NT/YiyxBP: boot 07.09.2026, dockerd не запустил четыре контейнера из-за сетевого attach; reconciler поднял их в 20:08:28–34. Это уже полученное evidence, а не новый тест. Специальная перезагрузка или restart Docker для квалификации не нужны.

SSH-маршрут Olvery test

Прямой маршрут — **root@srv-app.olvr**, без перехода через gw1. Используйте собственный разрешённый project SSH key и штатный OpenSSH known_hosts своего operator; чужую .ssh не изменяйте.

Действует правило владельца команды из /home/olvr/HANDBOOK.md, раздел «SSH: правило владельца», решение 2u1L/hrBEmT. Независимую сверку fingerprint и подтверждение через доверенную консоль не выполнять, не запрашивать и не делать условием работы. Новый target не требует предварительно подтверждённой записи.

Перед командами задайте PROJECT_SSH_KEY как путь к своему ключу, а PROJECT_KNOWN_HOSTS — как штатный путь known_hosts своего operator. Клиент использует StrictHostKeyChecking=accept-new: новый ключ принимается штатно, изменение уже

известного ключа не скрывается.

```
: "${PROJECT_SSH_KEY:?Укажите путь к собственному project SSH key}"
: "${PROJECT_KNOWN_HOSTS:?Укажите штатный путь OpenSSH known_hosts своего
operator}"
ssh -F /dev/null -i "$PROJECT_SSH_KEY" \
  -o BatchMode=yes -o ConnectTimeout=10 \
  -o StrictHostKeyChecking=accept-new \
  -o UserKnownHostsFile="$PROJECT_KNOWN_HOSTS" \
  root@srv-app.olvr 'hostname'
```

Единственная оговорка — изменение уже известного host key: проверить, что переинсталляция этого сервера нам известна, затем перейти на новый ключ в своём штатном known_hosts и сохранить краткую запись «ключи сменились, перешли на новые». Если известного факта переинсталляции нет, не обходить штатный отказ и не переходить молча; сообщить РП только этот факт, без запроса fingerprint, доверенной консоли или нового поручения на верификацию. StrictHostKeyChecking=no/off и отбрасывание known_hosts не заменяют этот рецепт.

Отказ авторизации — отдельный факт доступа; он не создаёт поручения на верификацию SSH-ключей. Адресная проверка hostname ниже относится к правильности target и фактическому доступу.

Сохранённое историческое наблюдение. Точный пример: zIBX/fP8y5V, 01.10.2026 23:40:25 UTC; исполнитель использовал собственные /home/zIBX/.ssh/id_ed25519 и /home/zIBX/.ssh/known_hosts-Z0Ix. Эти пути — provenance выполненной команды, не инструкция использовать чужие файлы или создавать новый реестр. Успешный прямой маршрут со строгой проверкой наблюдался в Z0Ix/ojhIxI от 01.10.2026 23:59:05 UTC; нынешний клиентский рецепт выше не переписывает результат той команды.

Наблюдённый тогда ED25519 fingerprint:

SHA256:Ge9TS9DP0M09BK4unYE8GbV3v0ni7ZmjfKv3IqzKbNQ. Применение переданной публичной записи для test было разрешено РП OLVR: zIBX/USeeDT, 01.10.2026 23:39:06 UTC. Независимое подтверждение подлинности хоста тогда не выполнялось; новое поручение из этого исторического факта не следует.

Документационная запись сохранена в Card Kt2U; нового подключения для её публикации не выполнялось. vIup по production srv-prod-app.olvr отменена, не является prerequisite этого маршрута и не объявляется выполненной.

Адресная проверка (read-only)

Выполнять с операционного рабочего места с разрешённым SSH-доступом; сначала подтвердить exact host:

```
ssh -F /dev/null -i "$PROJECT_SSH_KEY" -o BatchMode=yes -o ConnectTimeout=10
-o StrictHostKeyChecking=accept-new -o
UserKnownHostsFile="$PROJECT_KNOWN_HOSTS" root@srv-app.olvr 'hostname'
ssh -F /dev/null -i "$PROJECT_SSH_KEY" -o BatchMode=yes -o ConnectTimeout=10
-o StrictHostKeyChecking=accept-new -o
```

```
UserKnownHostsFile="$PROJECT_KNOWN_HOSTS" root@srv-app.olvr 'systemctl cat
tpark-it-reconcile.service'
ssh -F /dev/null -i "$PROJECT_SSH_KEY" -o BatchMode=yes -o ConnectTimeout=10
-o StrictHostKeyChecking=accept-new -o
UserKnownHostsFile="$PROJECT_KNOWN_HOSTS" root@srv-app.olvr 'systemctl is-
enabled tpark-it-reconcile.service'
ssh -F /dev/null -i "$PROJECT_SSH_KEY" -o BatchMode=yes -o ConnectTimeout=10
-o StrictHostKeyChecking=accept-new -o
UserKnownHostsFile="$PROJECT_KNOWN_HOSTS" root@srv-app.olvr 'systemctl show
tpark-it-reconcile.service -p ActiveState -p SubState -p Result -p After'
ssh -F /dev/null -i "$PROJECT_SSH_KEY" -o BatchMode=yes -o ConnectTimeout=10
-o StrictHostKeyChecking=accept-new -o
UserKnownHostsFile="$PROJECT_KNOWN_HOSTS" root@srv-app.olvr 'cat
/usr/local/sbin/tpark-it-reconcile.sh; /bin/bash -n /usr/local/sbin/tpark-
it-reconcile.sh'
ssh -F /dev/null -i "$PROJECT_SSH_KEY" -o BatchMode=yes -o ConnectTimeout=10
-o StrictHostKeyChecking=accept-new -o
UserKnownHostsFile="$PROJECT_KNOWN_HOSTS" root@srv-app.olvr 'docker ps -a --
filter "name=~/tpark-it\." --format "{{.Names}}" state={{.State}}"'
ssh -F /dev/null -i "$PROJECT_SSH_KEY" -o BatchMode=yes -o ConnectTimeout=10
-o StrictHostKeyChecking=accept-new -o
UserKnownHostsFile="$PROJECT_KNOWN_HOSTS" root@srv-app.olvr 'docker ps -a --
filter "name=~/tpark-it\." --filter status=exited --format "{{.Names}}"'
ssh -F /dev/null -i "$PROJECT_SSH_KEY" -o BatchMode=yes -o ConnectTimeout=10
-o StrictHostKeyChecking=accept-new -o
UserKnownHostsFile="$PROJECT_KNOWN_HOSTS" root@srv-app.olvr 'journalctl -u
tpark-it-reconcile.service -b --no-pager'
```

Нормальное состояние завершённого oneshot: enabled, active/exited, Result=success. active/exited у этого unit не означает Exited-контейнер.

Проверять именно состояния контейнеров: отсутствие строк с status=exited само по себе не доказывает, что все они running. Историческое wording «зависимость ещё не готова?» в логе скрипта не является диагностикой RabbitMQ/БД.

Восстановление существующим механизмом

Если на **test** после boot остались Exited tpark-it.*, сначала прочитать журнал Docker и reconciler. Когда Docker и сеть доступны, повторить уже существующий oneshot (это write-операция, не read-only проверка):

```
ssh -F /dev/null -i "$PROJECT_SSH_KEY" -o BatchMode=yes -o ConnectTimeout=10
-o StrictHostKeyChecking=accept-new -o
UserKnownHostsFile="$PROJECT_KNOWN_HOSTS" root@srv-app.olvr 'systemctl
restart tpark-it-reconcile.service'
ssh -F /dev/null -i "$PROJECT_SSH_KEY" -o BatchMode=yes -o ConnectTimeout=10
-o StrictHostKeyChecking=accept-new -o
UserKnownHostsFile="$PROJECT_KNOWN_HOSTS" root@srv-app.olvr 'systemctl show
tpark-it-reconcile.service -p ActiveState -p SubState -p Result; journalctl
```

```
-u tpark-it-reconcile.service -b --no-pager; docker ps -a --filter "name=^/tpark-it\" --format "{{.Names}}" state="{{.State}}"
```

Из-за `RemainAfterExit=yes` обычный `start` уже active unit не повторит выполнение; для повторного запуска нужен `restart` этого unit, **не Docker**. В а0V6 эта операция не выполнялась.

При ошибке unit или оставшихся non-running контейнерах сохранить время, имена и сетевую/стартовую диагностику без секретов, передать incident owner **ПП OLVR**. Не менять данные, topology, olvery/cicd, backend retry; не создавать второй watchdog.

Observation и обратный ход

На ближайшем **уже плановом** reboot можно сохранить: время старта Docker, журнал reconciler, все `tpark-it.*` running не позже 6 минут после старта Docker без ручного вмешательства. Новая перезагрузка для этой работы не запрашивается.

Изменение а0V6 — только комментарий-обоснование скрипта; исполняемая логика и unit не изменены. Откат — восстановить прежний комментарий по diff и before-state в handoff а0V6, проверить `/bin/bash -n`. Перезапуск/daemon-reload не требуется. Не отключать принятый механизм как «откат документации».

Этот runbook не хранит копию исполняемого скрипта: фактический source поведения остаётся на указанном host, решения и датированное evidence — в Cards.

From:

<https://wiki.olvery.ru/> - Olvery

Permanent link:

<https://wiki.olvery.ru/doku.php?id=runbooks:tpark-it-reconcile-test>

Last update: **2026/10/02 04:11**

