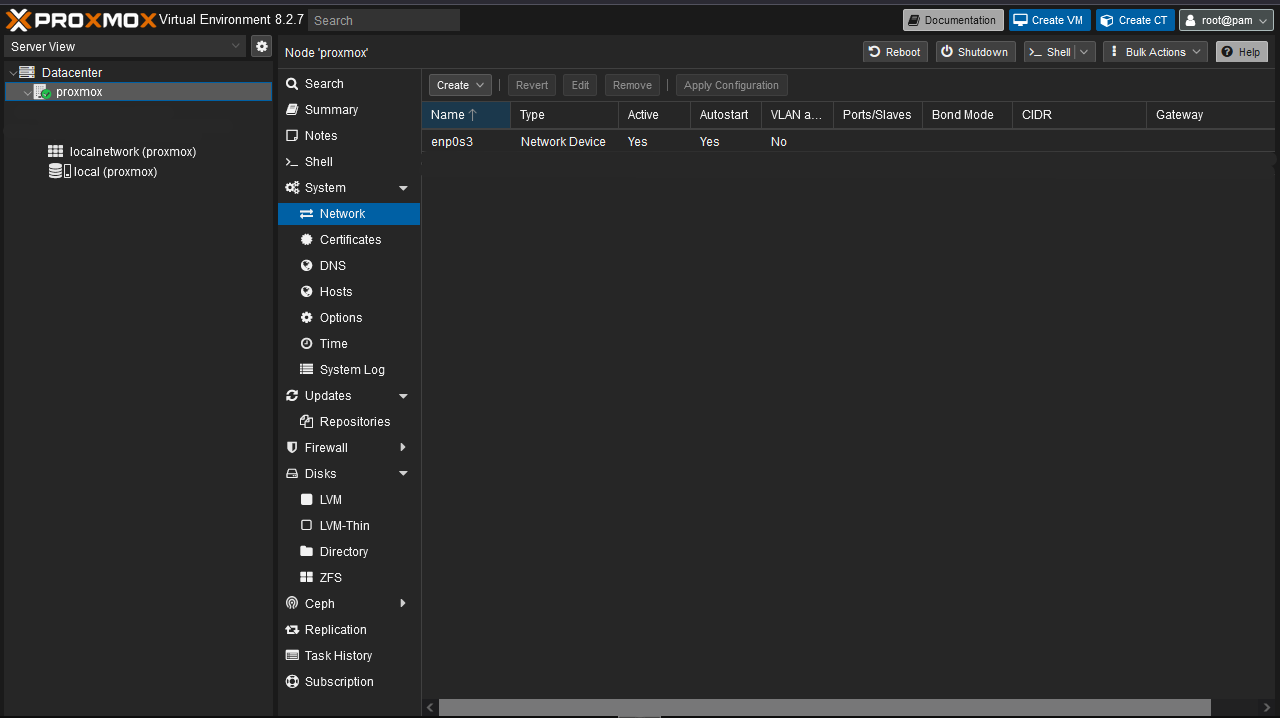
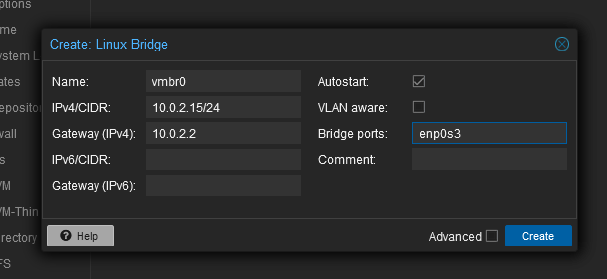
|  |
| --- |
| Университет ИТМО |
| Лабораторная работа №2 |
| Знакомство с LXC-контейнерами |

1. Кликнуть на proxmox, перейти в раздел System -> Network. Там будет лишь один сетевой интерфейс по умолчанию, он же основной сетевой интерфейс ранее созданной виртуальной машины, будет называться enp3s0:

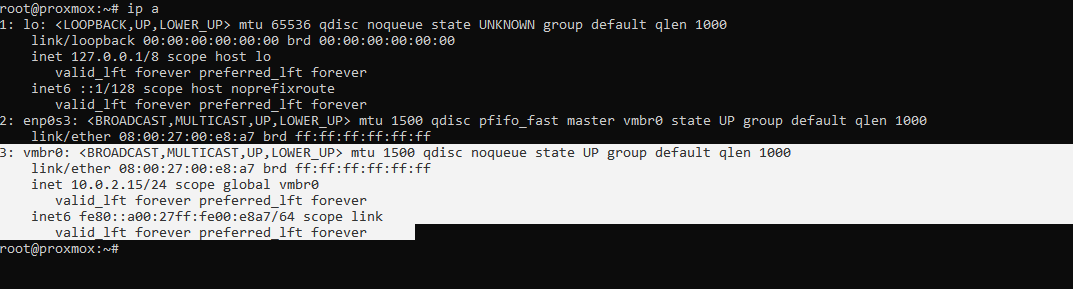


1. Для создания ВМ и контейнеров в Proxmox необходимо использовать сетевой интерфейс типа bridge (мост). Создать новый бридж можно, нажав Create -> Linux Bridge. При создании важны три вещи:

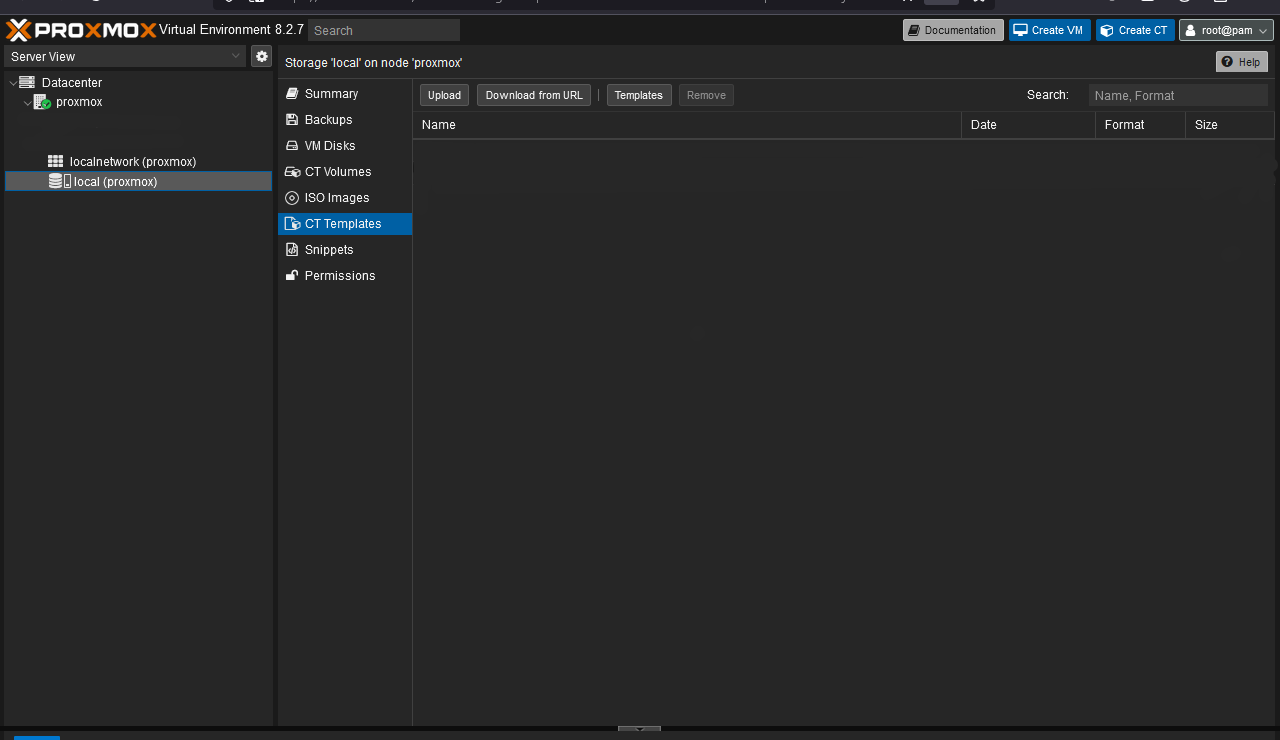
* IP-адрес нужно указать такой же, как установлен на базовой ВМ, в нашем случае 10.0.2.15/24
* В качестве шлюза необходимо указать 10.0.2.2 (его можно проверить заранее, выполнив на ВМ команду ip r и взять значение из строчки default via)
* В качестве порта бриджа прописать имя сетевого интерфейса по умолчанию.



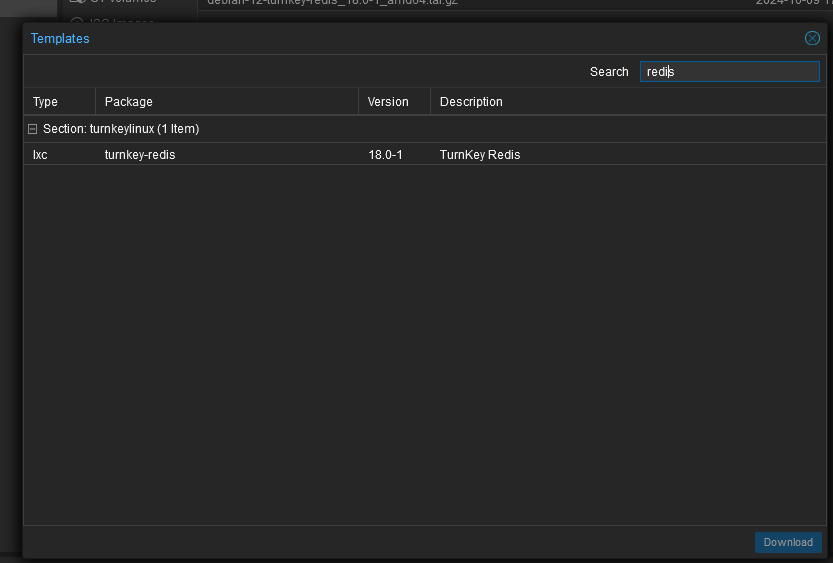
Если все правильно, то сетевой доступ не пропадет и основной интерфейс станет интерфейсом типа bridge. В этом можно убедиться, например, залогинившись на ВМ по SSH и выполнив команду *ip a*



1. Создать контейнер с Redis, для этого необходимо провалиться внутрь ноды proxmox и выбрать пункт меню *CT Templates*

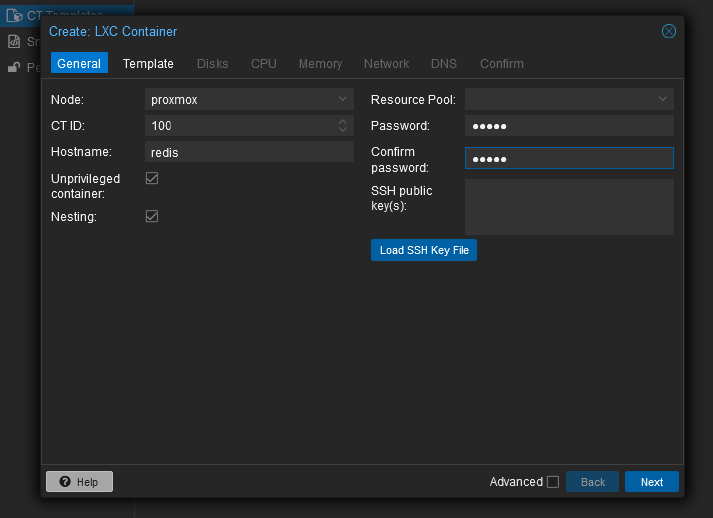


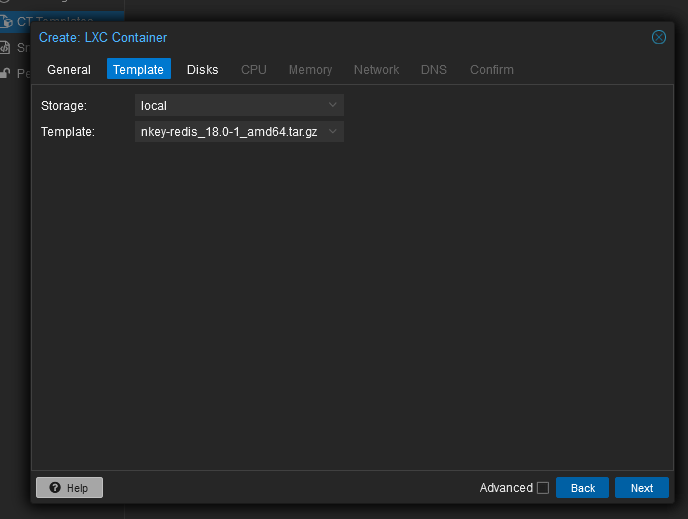
1. Для тренировки подойдет любой образ из готовых, публичных. Поэтому достаточно кликнуть кнопку Templates и через поиск найти нужны темплейт с redis, затем нажать *Download.*

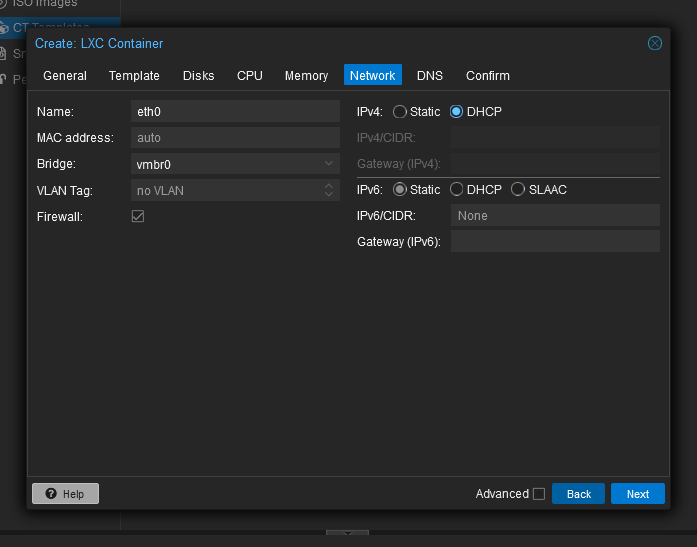


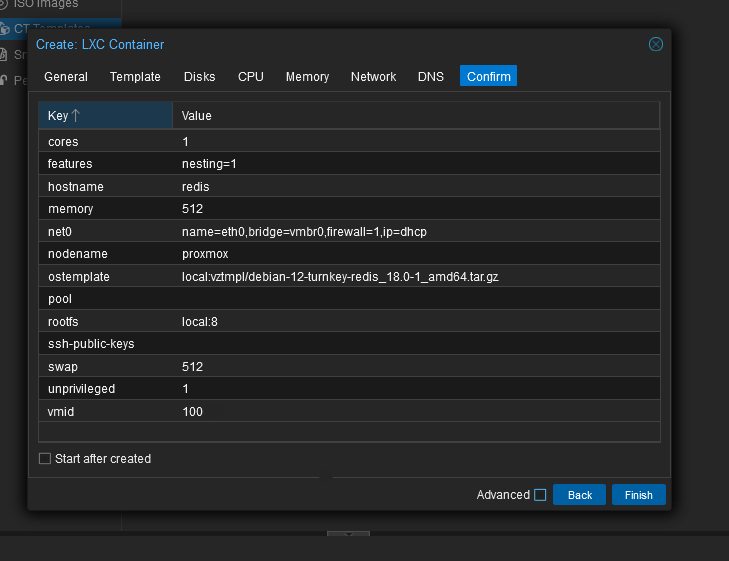
1. Чтобы не уходить далеко, начать создавать контейнер можно сразу в этом же разделе. В правом верхнем углу необходимо нажать заветную кнопку *Create CT*

Все настройки можно оставить по умолчанию, кроме IPv4 в разделе *Network* – там необходимо выбрать селектор **DHCP**. В качестве темплейта, понятно, выбираем только что скачанный postgresql. *Также важно запомнить/записать Hostname и пароль на самой первой вкладке, они пригодятся нам далее*

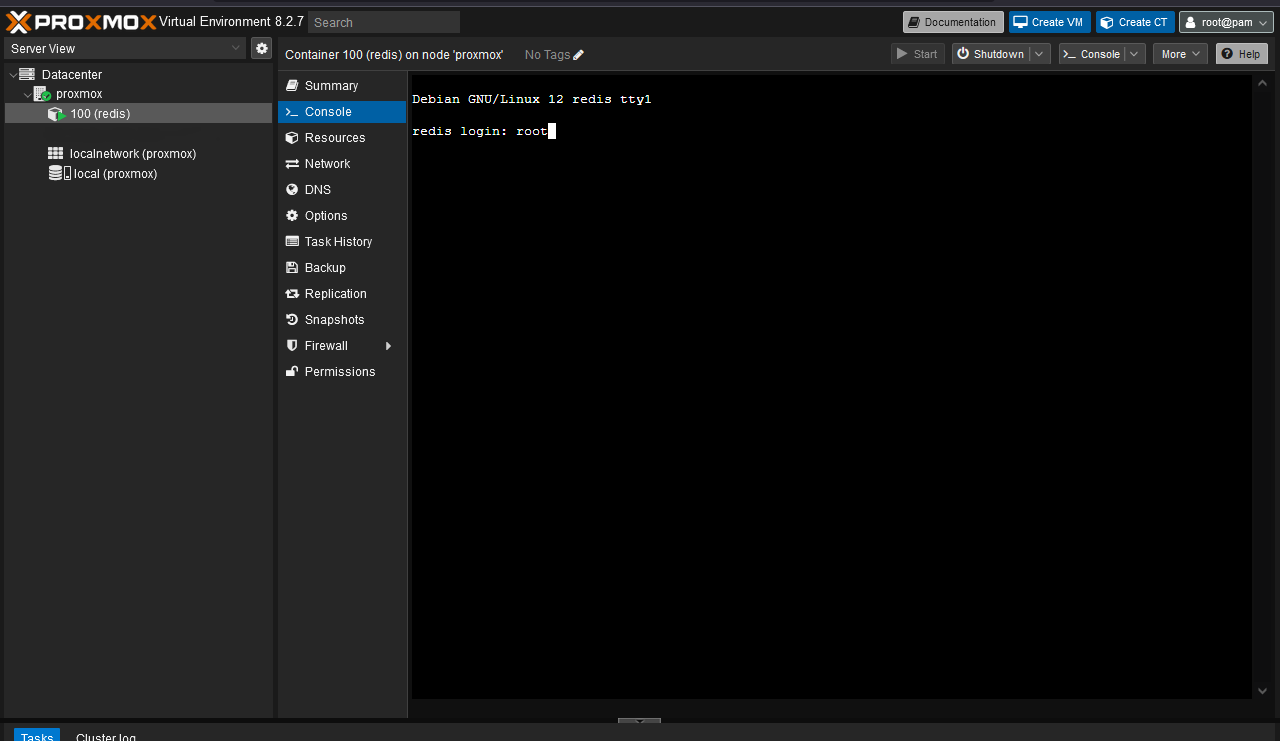








1. Теперь можно залогиниться на контейнер для его первоначальной настройки (init). Для этого необходимо кликнуть на свежесозданный контейнер в меню слева и выбрать пункт *Console* и затем нажимаем Start чтобы запустить контейнер. Пишем console *Логин – root, пароль указывался на первой вкладке при создании контейнера*



1. На самом первом шаге инита просят придумать пароль для будущей админки Redis, но нам он не понадобится, поэтому можно ввести любой. Интерфейс для подключения выбираем all. Protected-mode – выключить. Все остальное можно прожать skip (фокус на кнопках меняется клавишей Tab или стрелками)

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Автоматически созданное описание

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание\

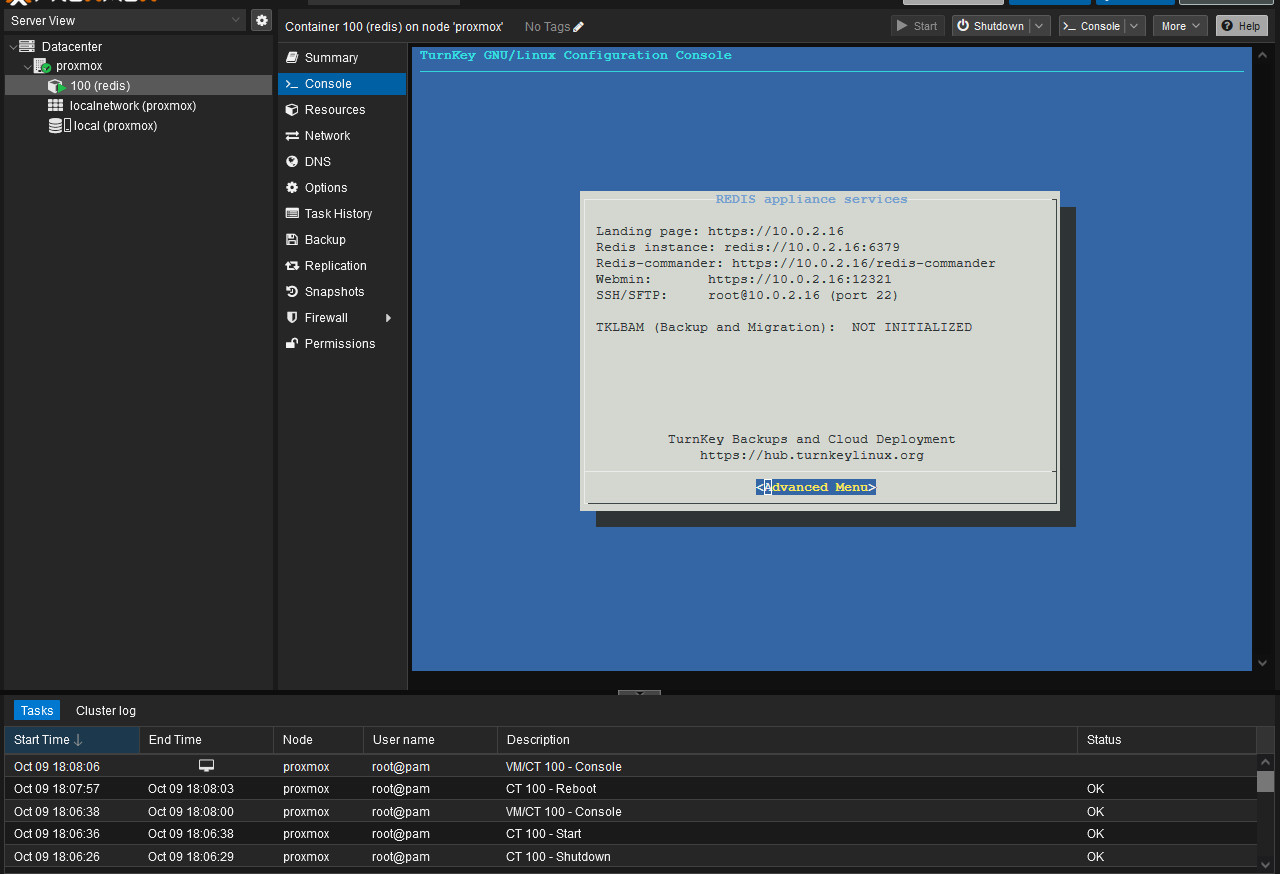
Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

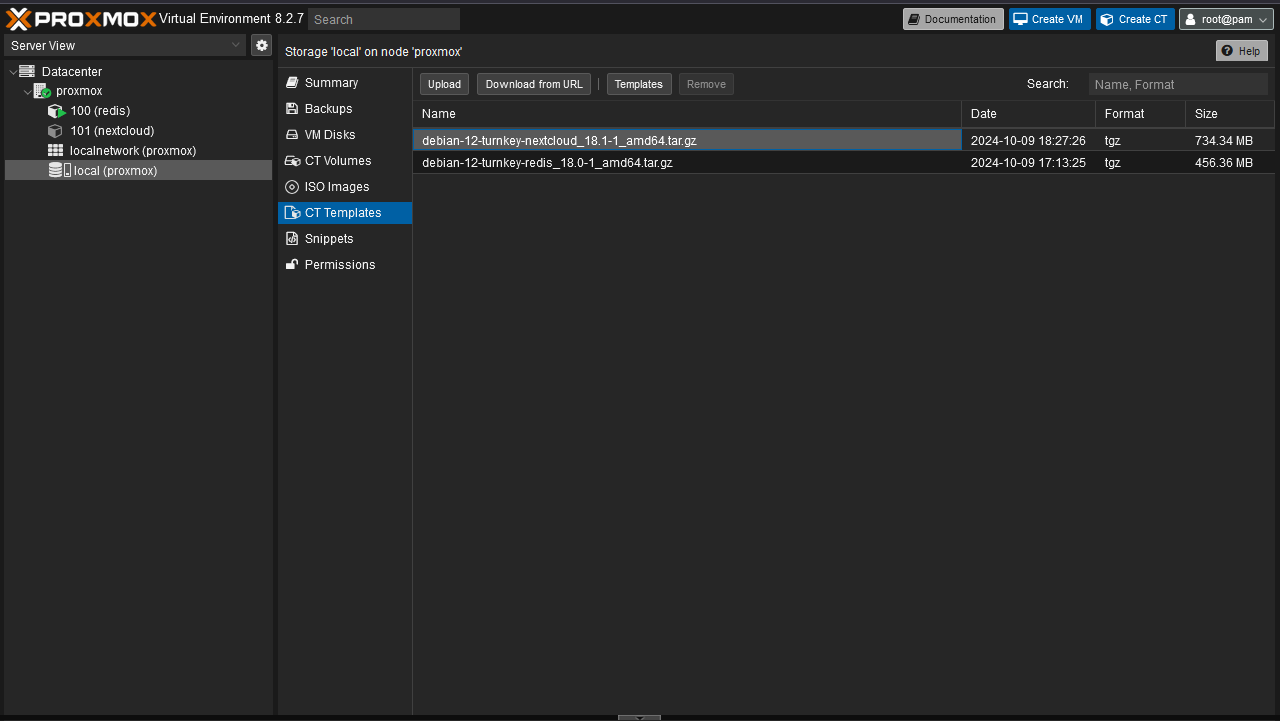
Последний экран важен, его лучше сохранить/заскринить: на нем перечислены адреса и порты доступов до redis (адреса и порты могут отличаться от скриншота). На этом этапе настройка считается завершенной, готовым контейнером с redis можно пользоваться по назначению (выход через пункт Quit)



Изображение выглядит как текст, электроника, дисплей, снимок экрана

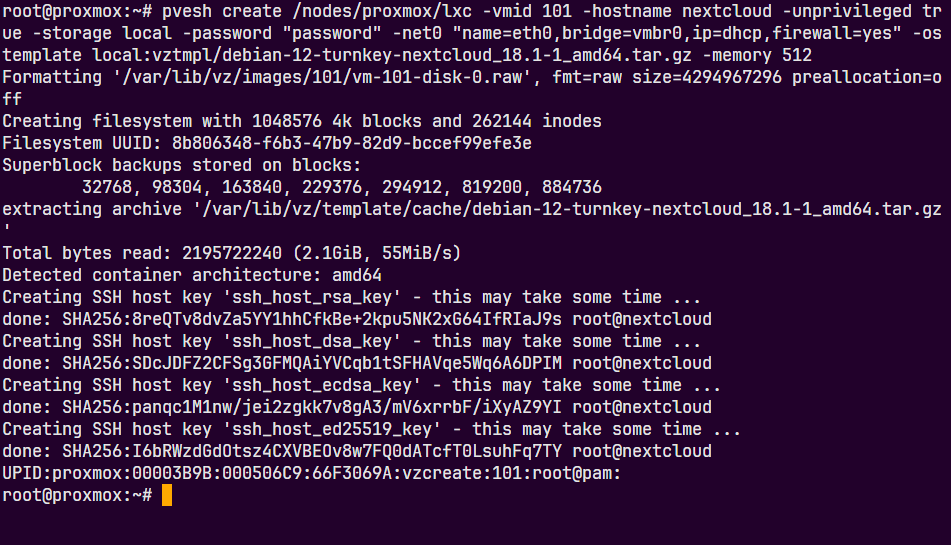
Автоматически созданное описание

1. Загрузим еще один дефолтный образ для контейнера, на этот раз ищем по слову nextcloud

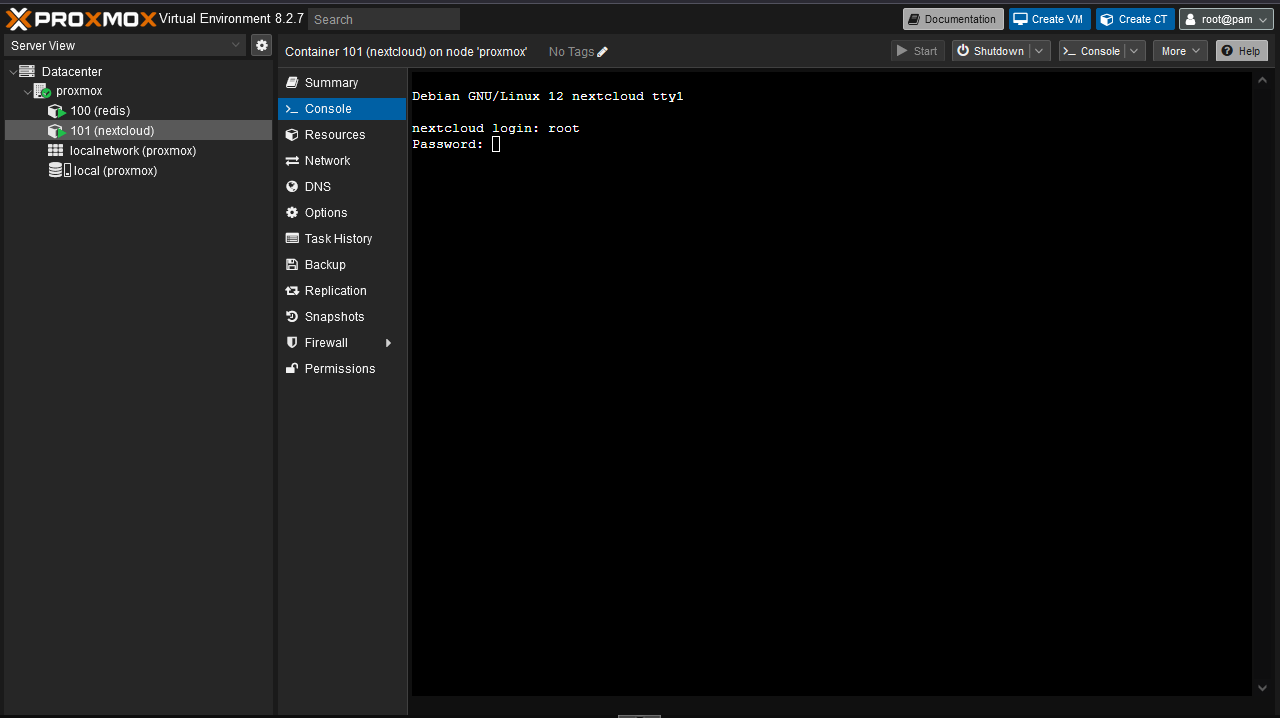


1. Создаем еще один контейнер из только что скачанного образа, но на этот раз с помощью CLI. Для этого необходимо подключиться к ВМ по SSH аналогично шагам в лабораторной № 1, либо просто залогиниться через встроенную консоль VirtualBox. Далее выполнить команду с помощью встроенной утилиты **pvesh**, например такую (параметры можно менять на свое усмотрение, поиграться):

pvesh create /nodes/proxmox/lxc -vmid 101 -hostname nextcloud -unprivileged true -storage local -password "какой\_нибудь\_пароль" -net0 "name=eth0,bridge=vmbr0,ip=dhcp,firewall=yes" -ostemplate local:vztmpl/debian-12-turnkey-nextcloud\_18.1-1\_amd64.tar.gz -memory 512



1. Возвращаемся в вебморду, проверяем что новый контейнер действительно успешно создался. Подключаемся через *Console* аналогично шагу 6.



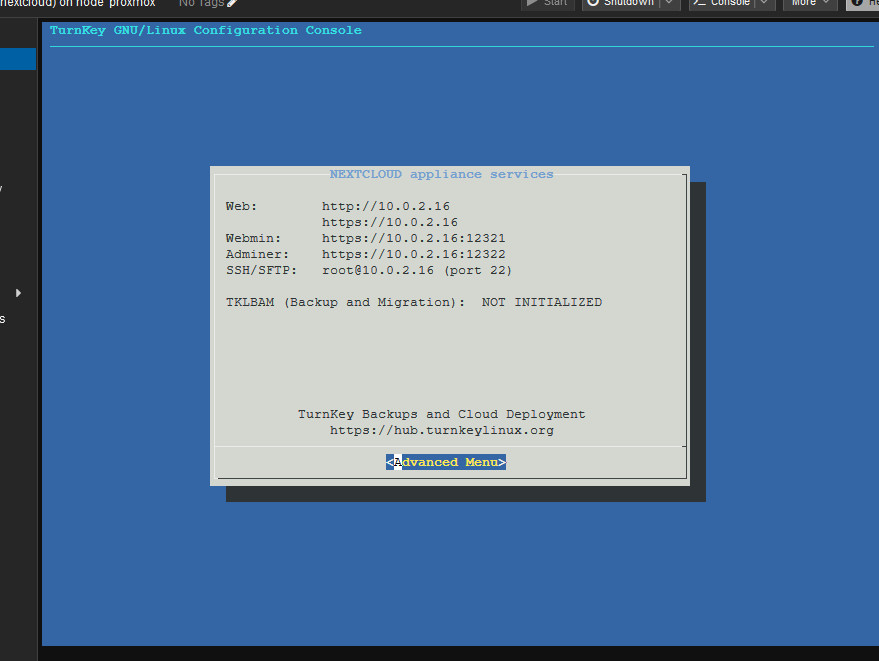
1. Проинициализировать контейнер с Nextcloud аналогично шагу 7. На моменте с Redis выдаст ошибку – ожидаемо, мы настроим подключение к нему позже, пока что выбираем *Skip*. В качестве *Nextcloud Domain* желательно указать localhost

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

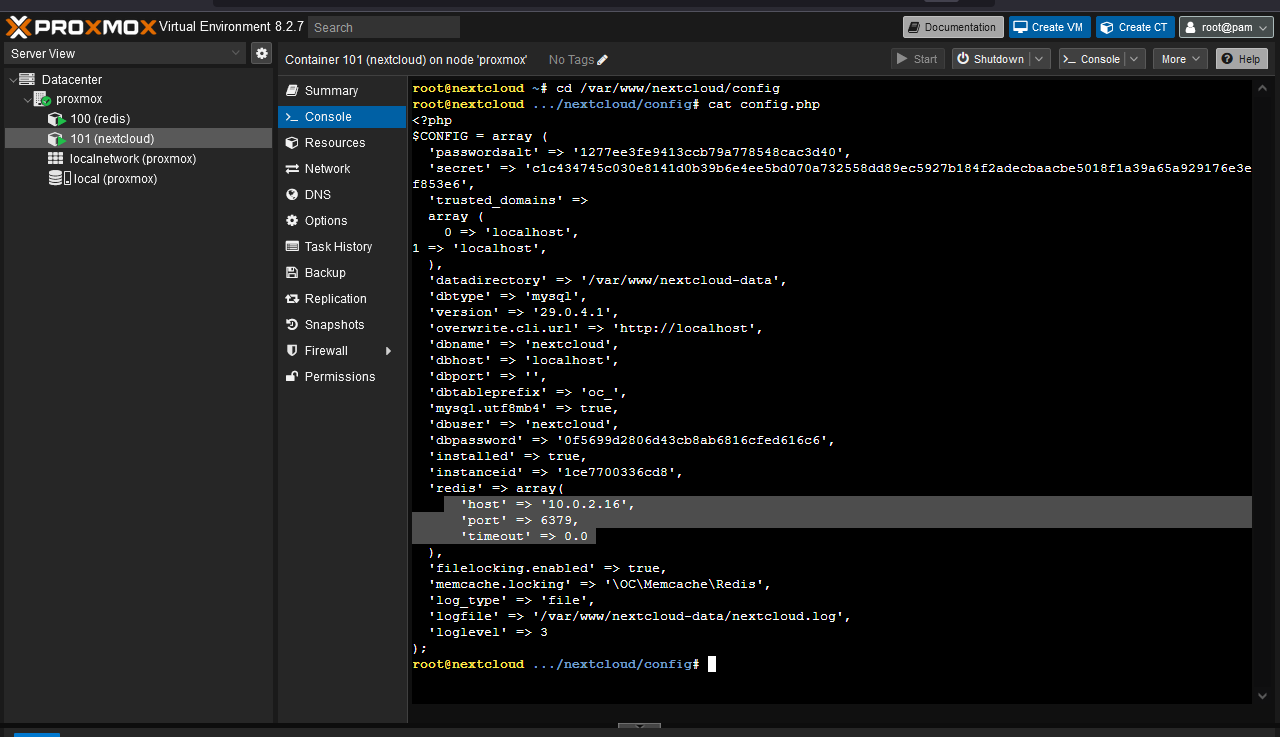


1. Аналогично шагам в лабораторной №1, сделать проброс портов (в этот раз 127.0.0.1:443-> адрес\_некстклауда:443) и проверить, действительно ли приложение запустилось. Для этого достаточно просто перейти в браузере по адресу <https://localhost/> - там мы [ожидаемо] увидим ошибку сервиса, т.к. некстклауд не смог найти редис

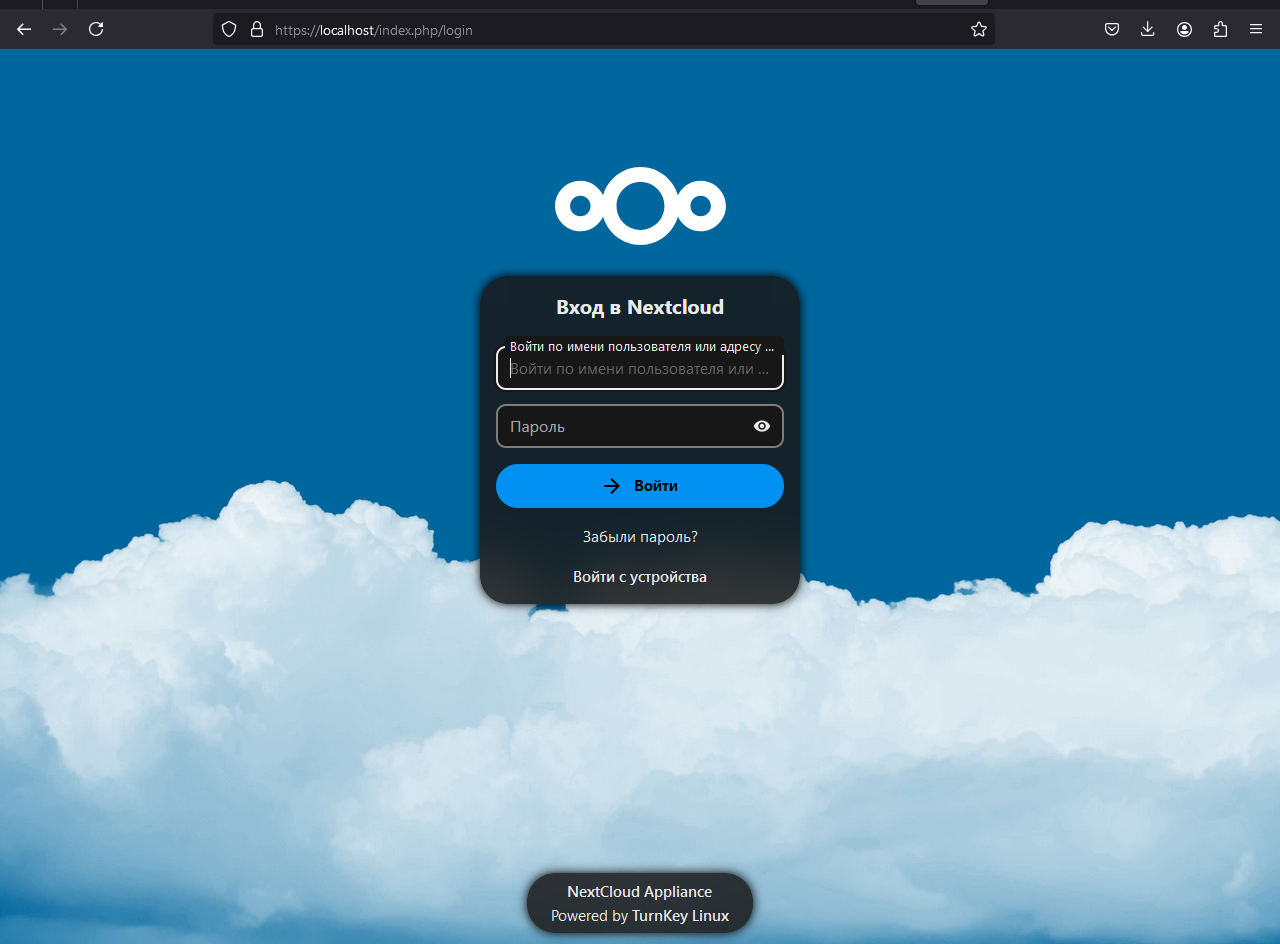
Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

1. Возвращаемся в консоль некстклауда, открываем файл **/var/www/nextcloud/config/config.php** и заменяем адрес редиса на ip/port из шага 7. Также отключаем Memcache, удалив целиком строчку с *‘memcache.local’*



1. Возвращаемся в браузер и убеждаемся, что теперь форма входа в Nextcloud открывается без ошибок



**Ответы:**

1. Неправильная конфигурация приведет к потере сетевой доступности, поскольку она нарушает принципы работы сетевого моста (bridge) в Proxmox.

Когда вы создаете новый бридж и указываете IP-адрес, шлюз и порт бриджа, вы фактически создаете новую сетевую конфигурацию, которая будет конфликтовать с существующей конфигурацией базовой ВМ.

1. В Proxmox адрес шлюза (default gateway) виртуальной машины обычно установлен на 10.0.2.2, а не 10.0.2.1. Это происходит из-за способа, которым Proxmox настраивает виртуальную сеть.

Proxmox использует технику под названием "NAT" (Network Address Translation) для обеспечения доступа к интернету для виртуальных машин. В NAT виртуальная машина получает частный IP-адрес, а Proxmox работает как шлюз, переводящий трафик между виртуальной машиной и внешним миром.