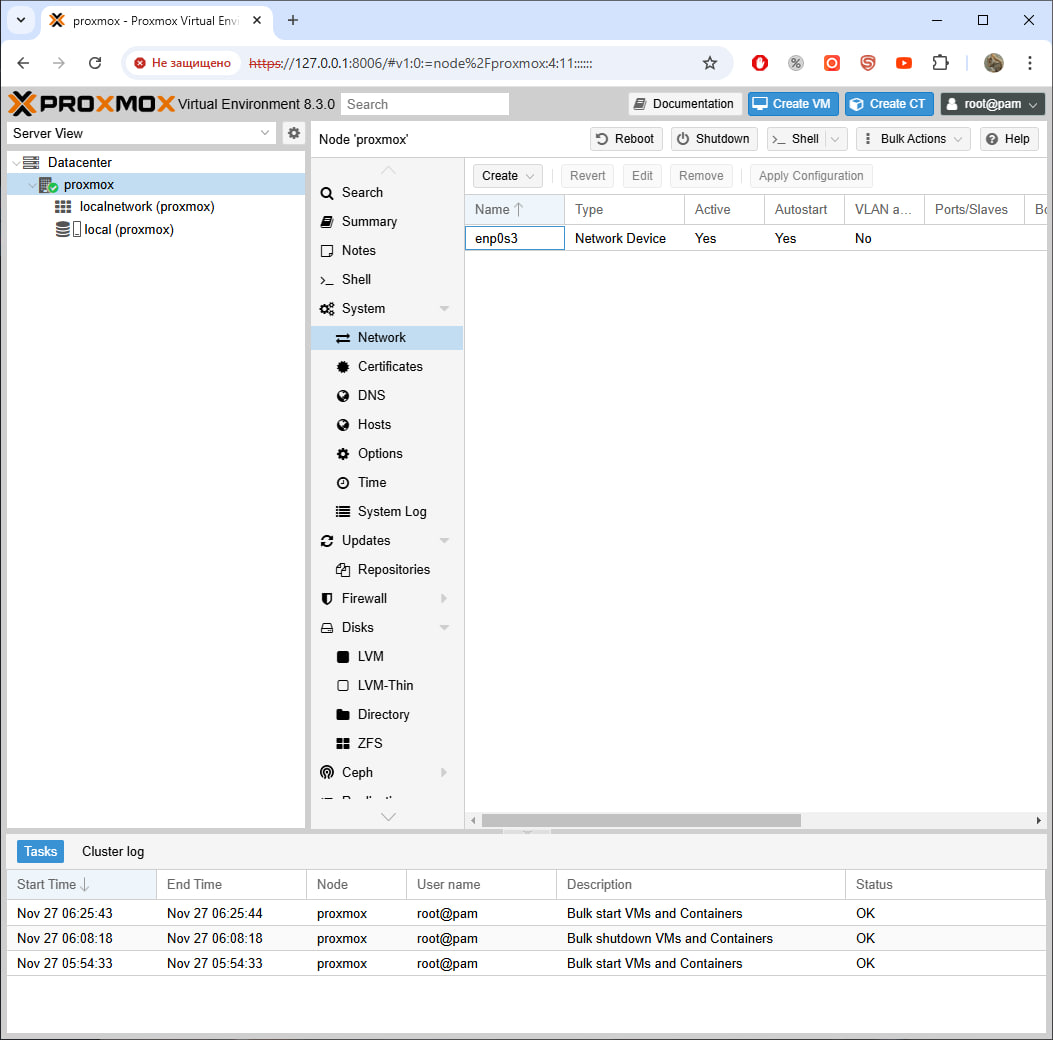
|  |
| --- |
| Университет ИТМО |
| Лабораторная работа №1 |
| Знакомство с LXC-контейнерами |

|  |
| --- |
| Факультет ИКТ  24.9.2024  Выполнил студент Петров Д. |

**Шаг 1. Кликнуть на ноду в нашем случае *proxmox*, перейти в раздел *System -> Network*. Там будет лишь один сетевой интерфейс по умолчанию, он же основной сетевой интерфейс ранее созданной виртуальной машины**



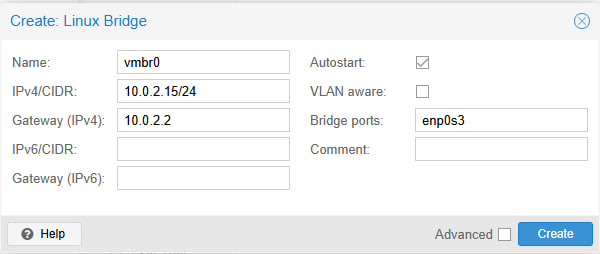
**Шаг 2. Для создания ВМ и контейнеров в Proxmox необходимо использовать сетевой интерфейс типа bridge (мост), поэтому существующий интерфейс нам не подойдет. Создаем новый бридж**

* 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.

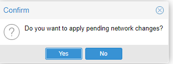
Смотрим default via



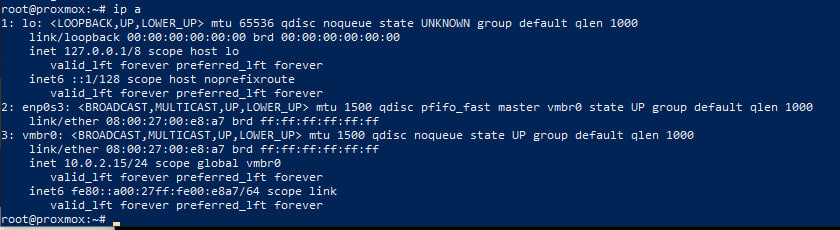
Создание бриджа



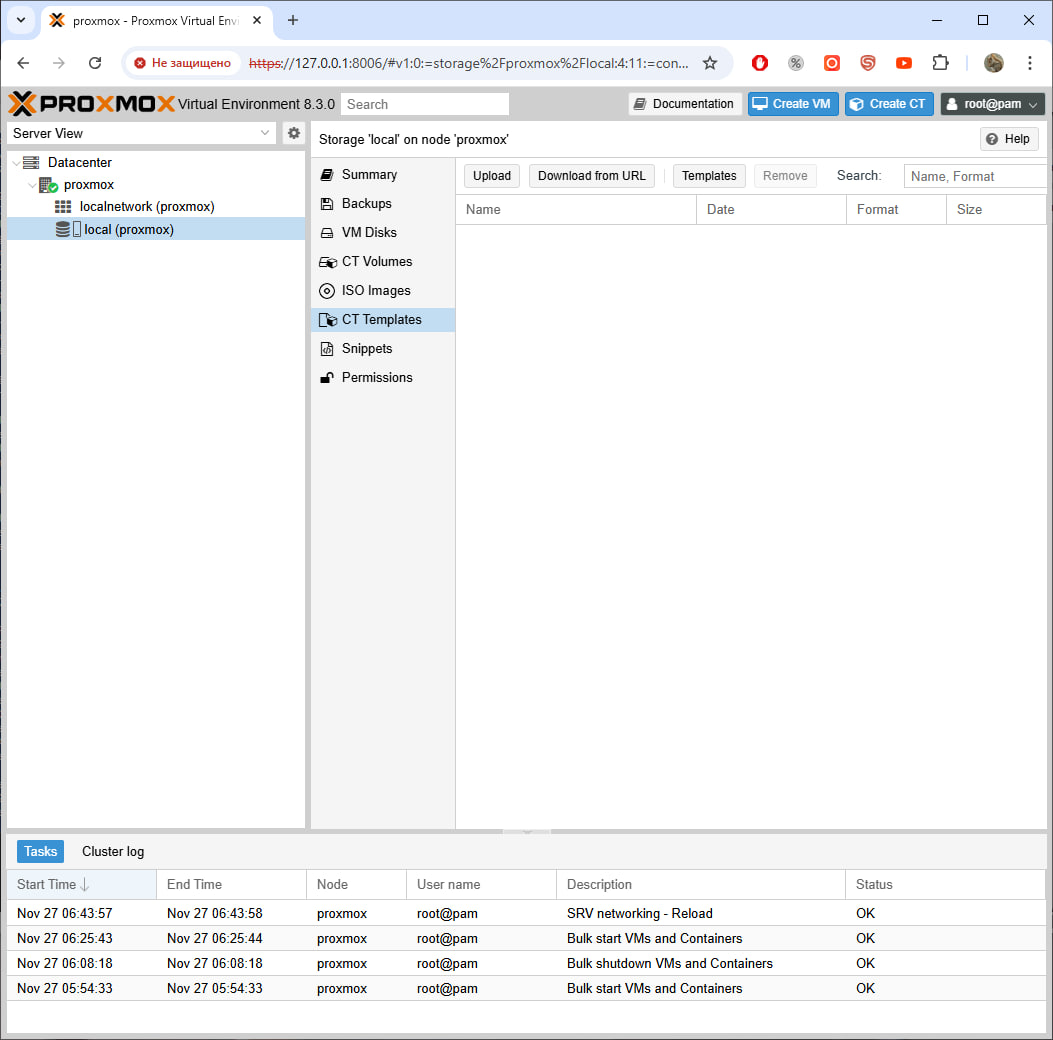
Проверяем по 10 раз все настройки, чтоб не было ошибок, ибо это критически важное место и применяем настройки



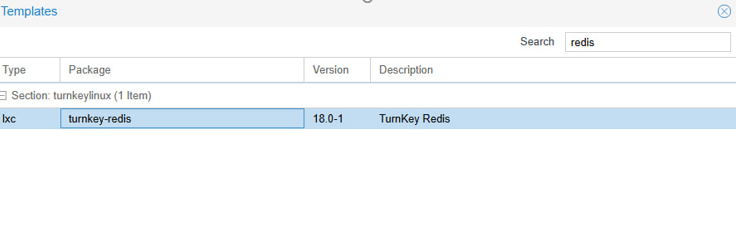
Если все правильно, то сетевой доступ не пропадет и основной интерфейс станет интерфейсом типа bridge. В этом можно убедиться, например, залогинившись на ВМ по SSH и выполнив команду *ip a*

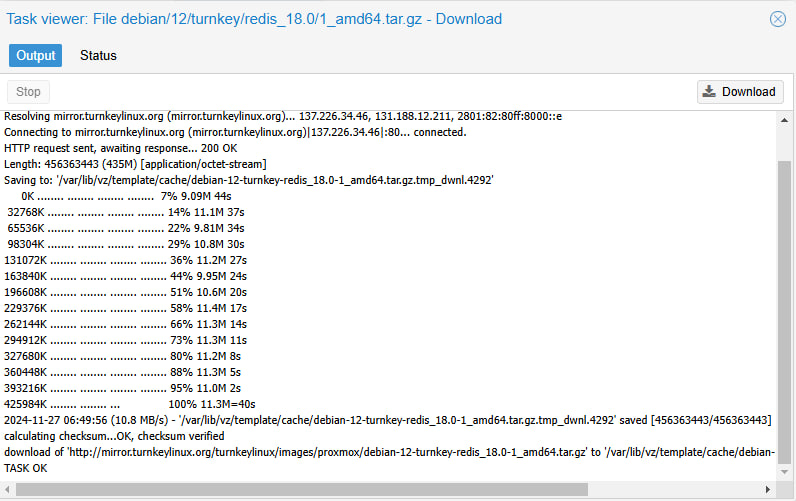


**Шаг 3. Для создания контейнеров необходим, как минимум, сам базовый образ для контейнера. Это может быть обычный Linux-образ, а может быть и готовый образ с предустановленным ПО. В Proxmox это называется template (шаблон). Попробуем создать контейнер с Redis, для этого необходимо провалиться внутрь ноды proxmox и выбрать пункт меню *CT Templates***



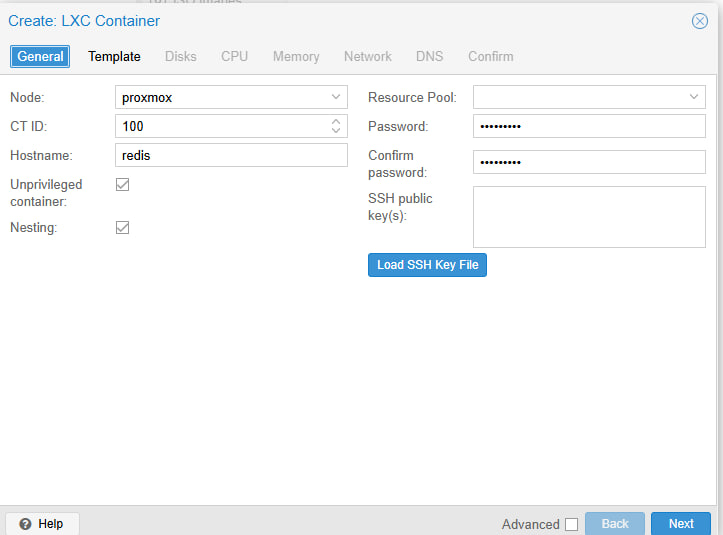
Шаг 4. Для тренировки подойдет любой образ из готовых, публичных. Поэтому достаточно кликнуть кнопку Templates и через поиск найти нужны темплейт с redis, затем нажать *Download*

Запускаем его установку и ждем окончания (TASK OK)



**Шаг 5. Начинаем создавать контейнер в этом же разделе. В правом верхнем углу необходимо нажать заветную кнопку *Create CT***

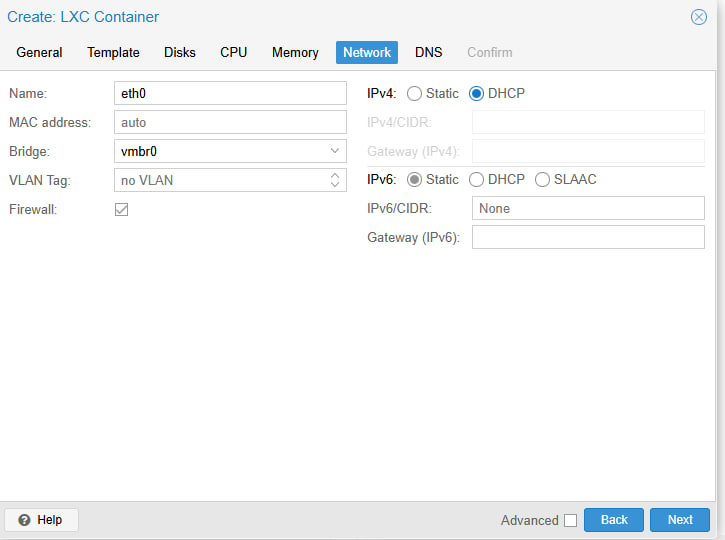
**Заполняем и запоминаем Hostname и пароль на самой первой вкладке, они пригодятся нам далее**



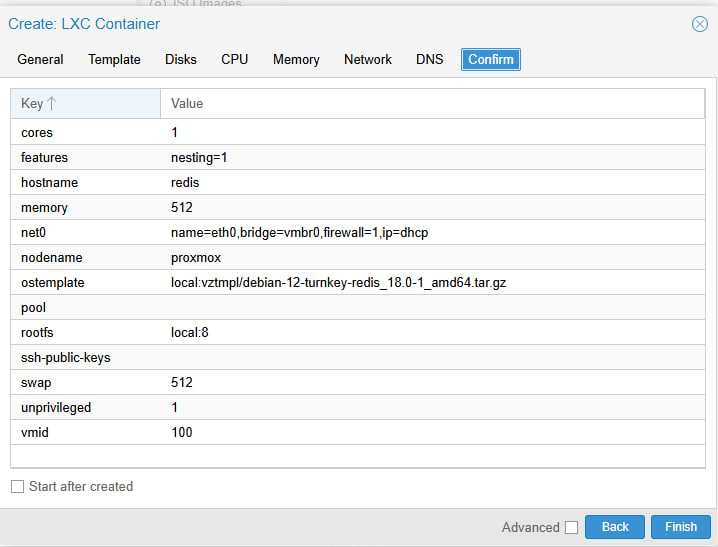
В качестве темплейта выбираем только что скачанный postgresql.



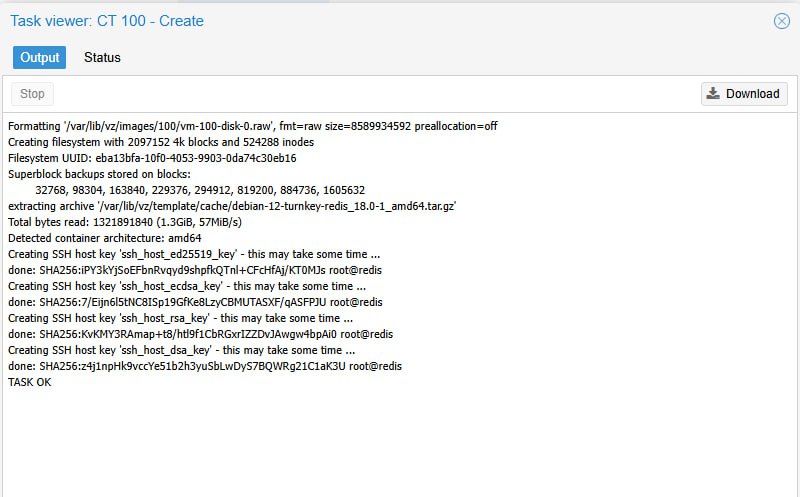
В разделе *Network* – там необходимо выбрать селектор **DHCP**



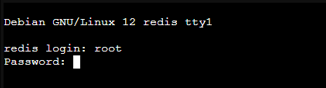
Итоговые настройки



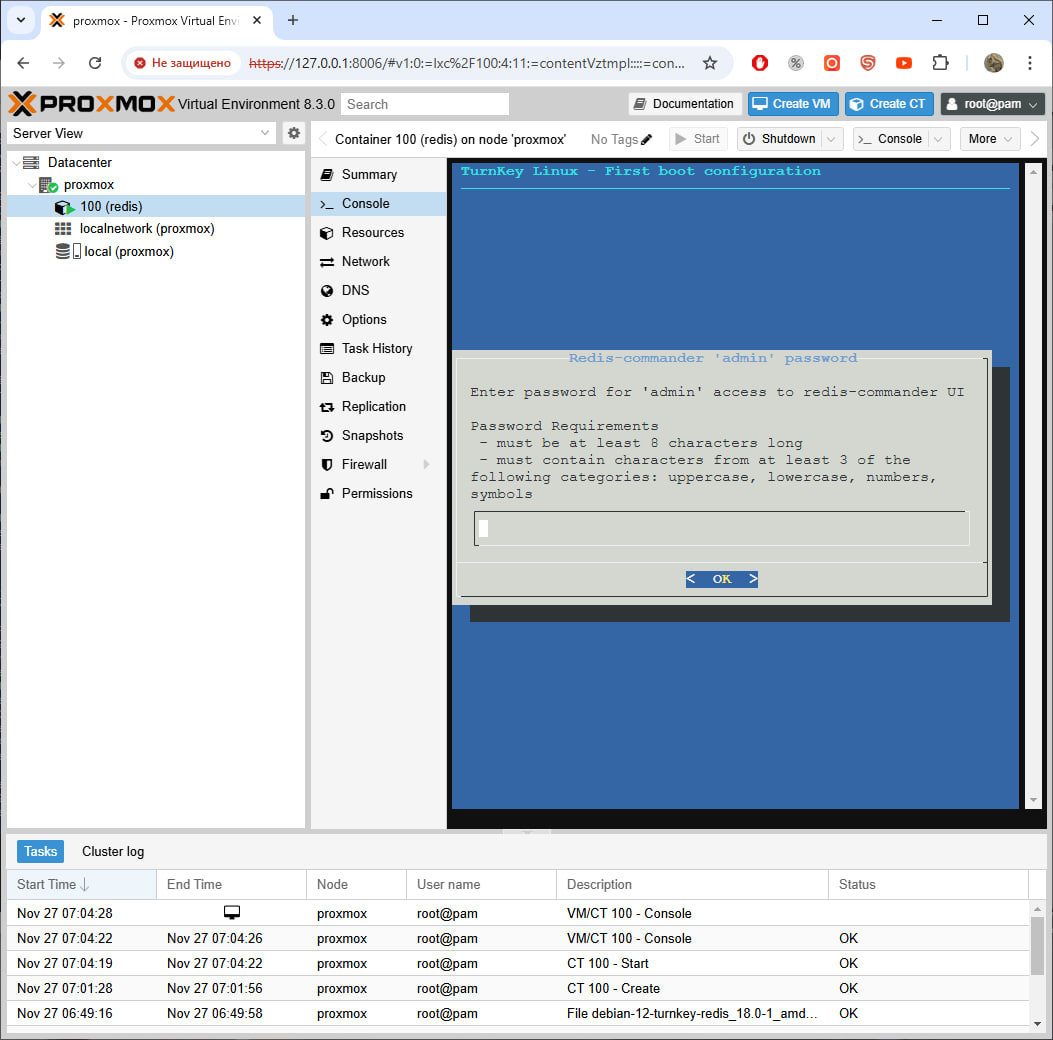
Ждем TASK OK



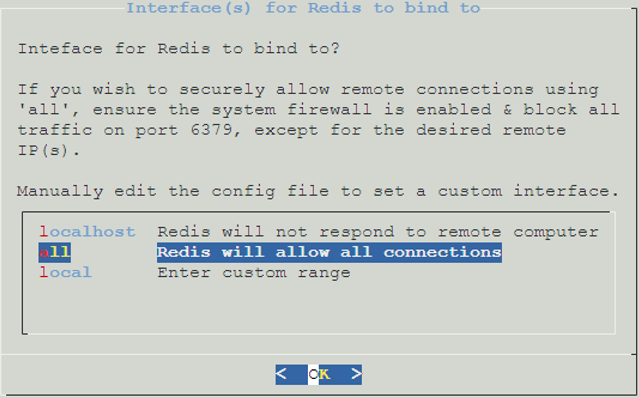
**Шаг 6. Если мы получили в итоге TASK OK, значит все создалось успешно и можно залогиниться на контейнер для его первоначальной настройки (init). Для этого необходимо кликнуть на свежесозданный контейнер в меню слева и выбрать пункт *Console*. *Логин – root, пароль указывался на первой вкладке при создании контейнера***



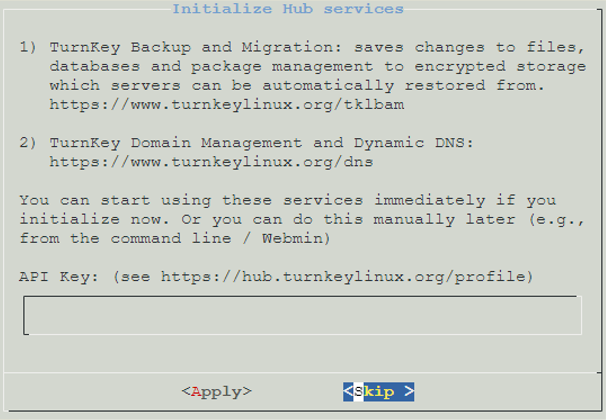
**Шаг 7. На самом первом шаге инита просят придумать пароль для будущей админки Redis, но нам он не понадобится, поэтому можно ввести любой.**



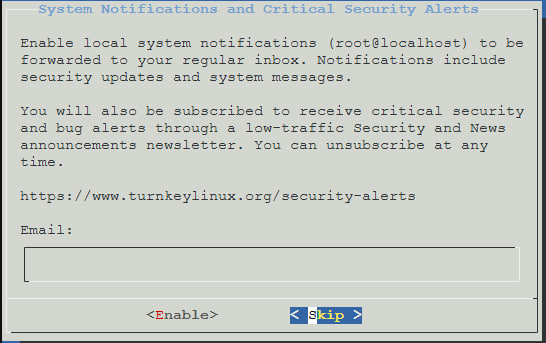
Интерфейс для подключения выбираем all. Protected-mode – выключить.



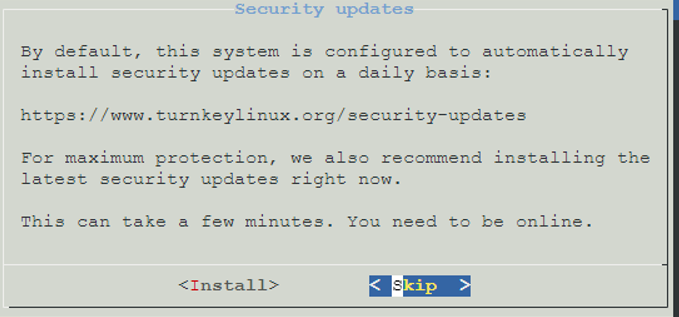
Все остальное скипаем



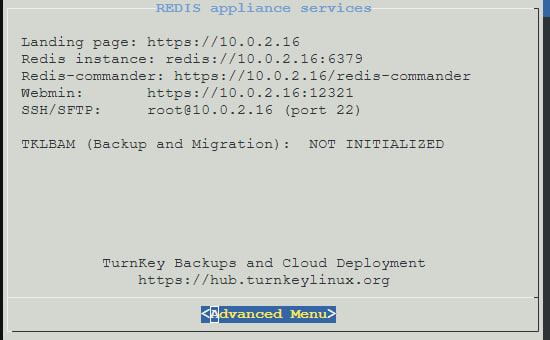
Это тоже скип



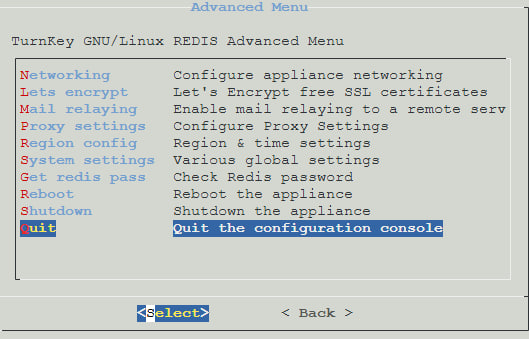
Аналогично скип



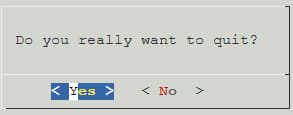
Последний экран важен скриним его на нем перечислены адреса и порты доступов до redis



После скрина нажимаем на выход



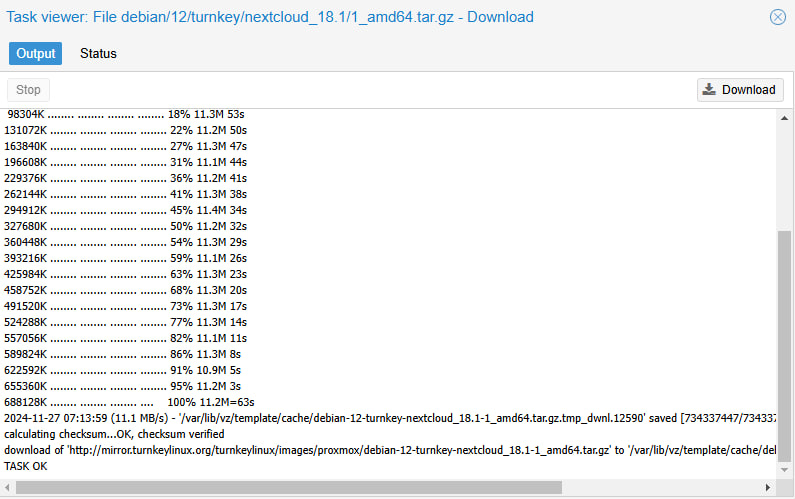
Подтверждаем



**Шаг 8. Загрузим еще один дефолтный образ для контейнера, на этот раз ищем по слову nextcloud**

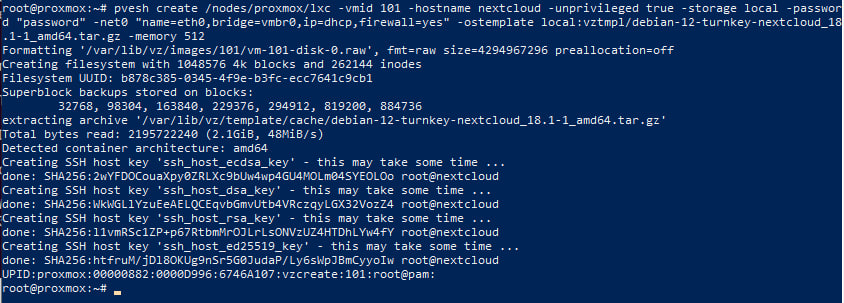


Ждем окончания установки



**Шаг 9. Создаем еще один контейнер из только что скачанного образа, но на этот раз с помощью CLI. Для этого необходимо подключиться к ВМ по SSH, либо просто залогиниться через встроенную консоль VirtualBox. Далее выполнить команду с помощью встроенной утилиты pvesh, например такую:**

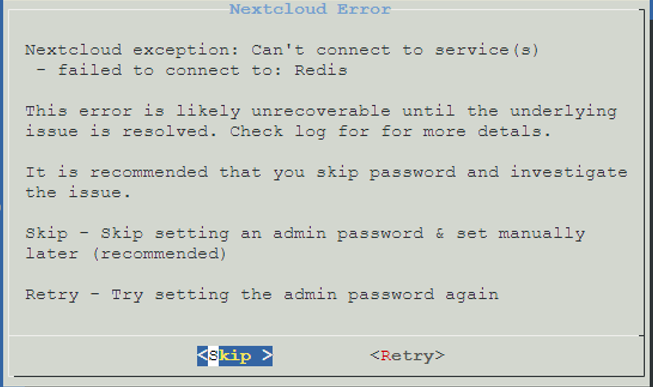
pvesh create /nodes/proxmox/lxc -vmid 101 -hostname nextcloud -unprivileged true -storage local -password "password" -net0 "name=eth0,bridge=vmbr0,ip=dhcp,firewall=yes" -ostemplate local:vztmpl/debian-12-turnkey-nextcloud\_18.1-1\_amd64.tar.gz -memory 512



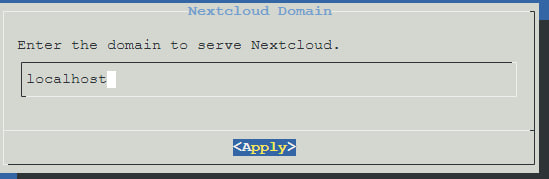
**Шаг 10. Возвращаемся в вебморду, проверяем что новый контейнер действительно успешно создался. Подключаемся через *Console* аналогично шагу 6.**

**Шаг 11. Проинициализировать контейнер с Nextcloud аналогично шагу 7.**

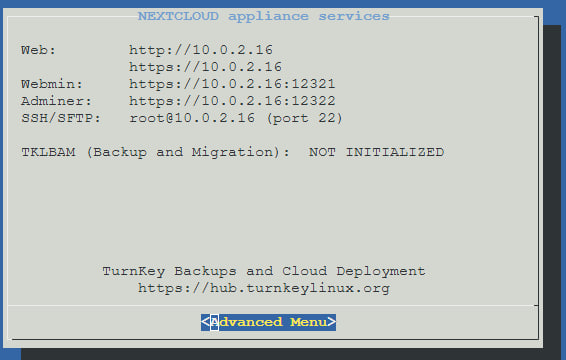
На моменте с Redis выдаст ошибку – ожидаемо, мы настроим подключение к нему позже, пока что выбираем *Skip*.



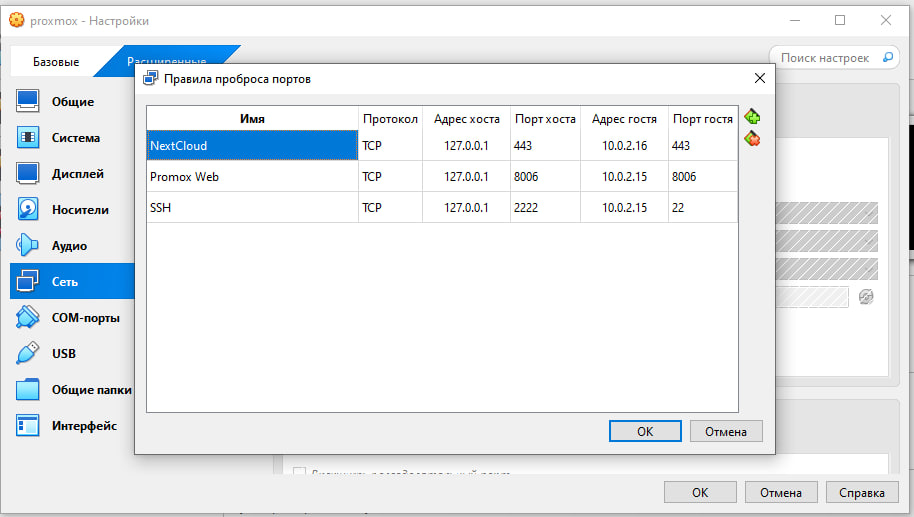
В качестве Nextcloud Domain указываем localhost(желательно)



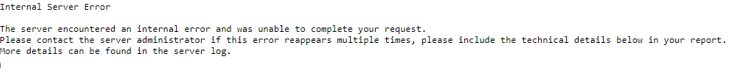
Аналогично как и с предыдущим финальным экраном скриним этот



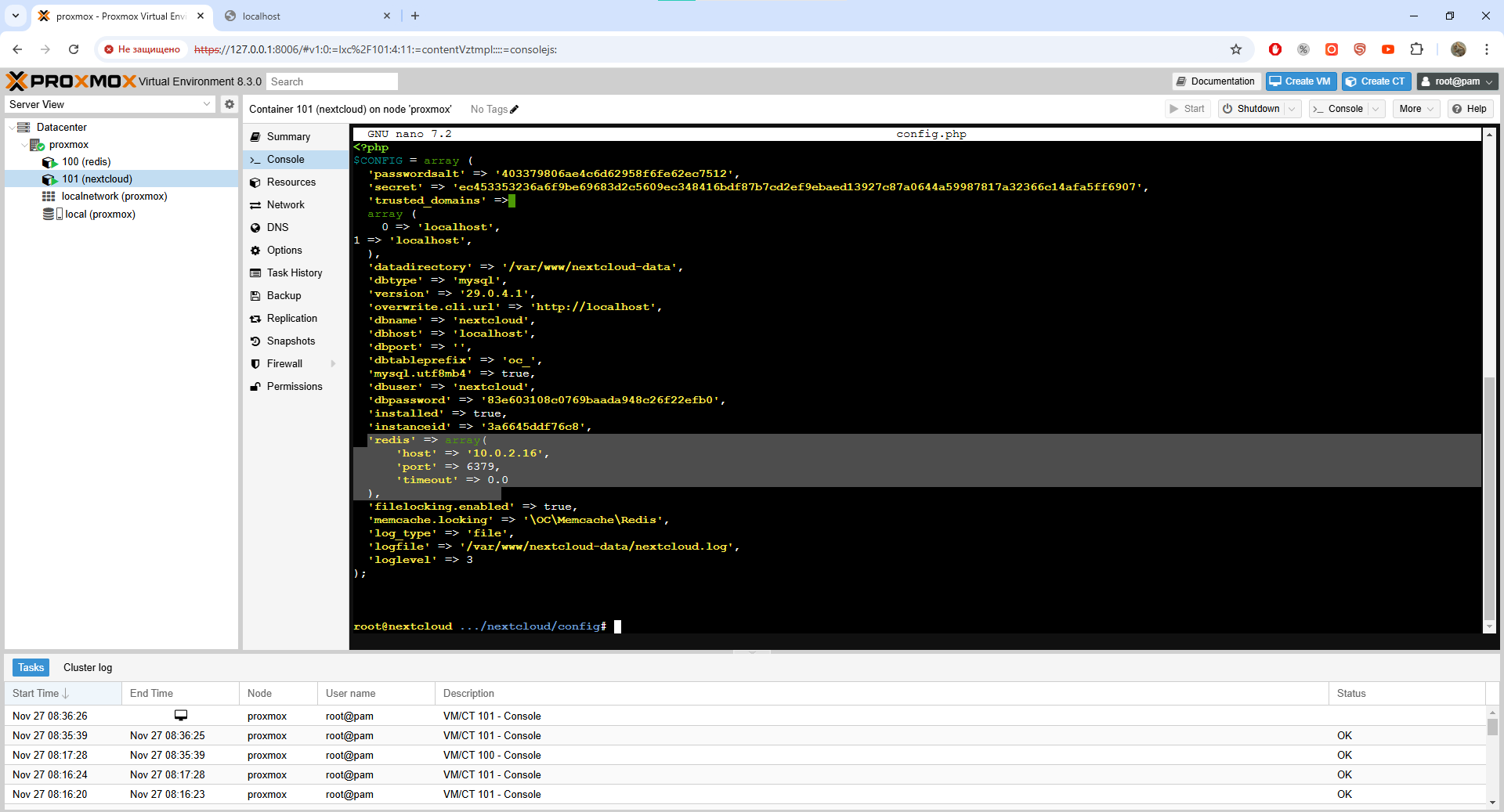
**Шаг 12. Делаем проброс портов**



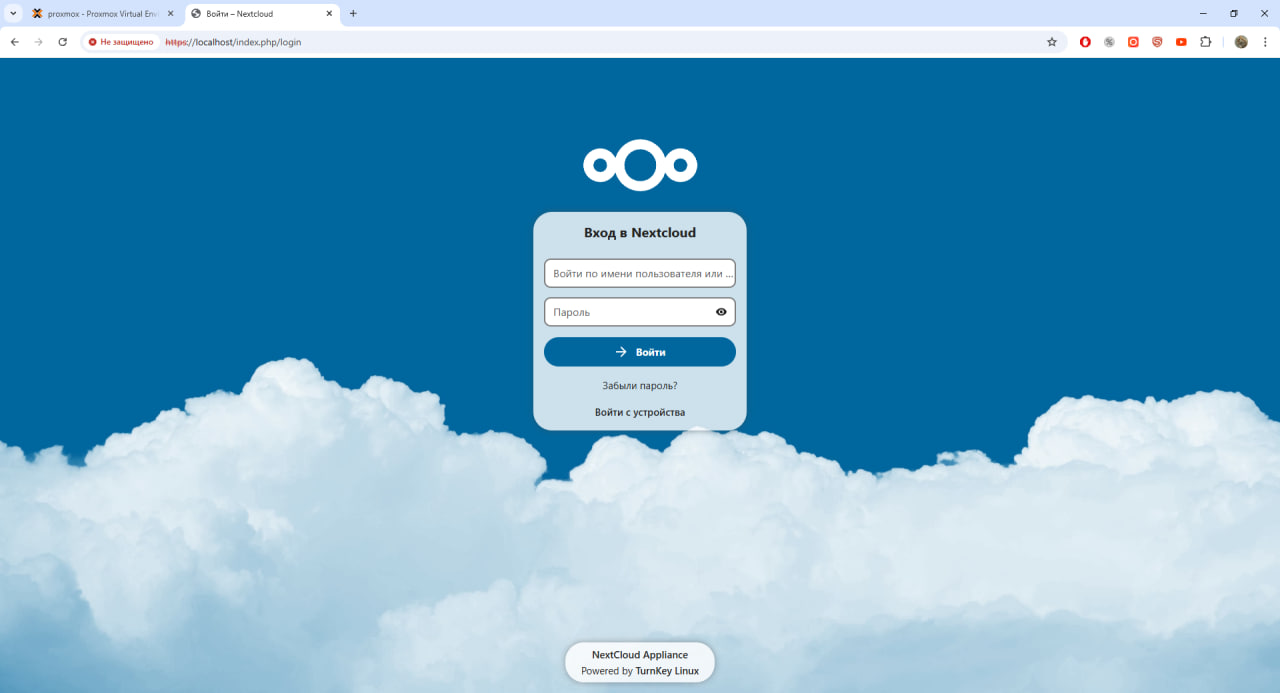
Проверить, действительно ли приложение запустилось. Для этого переходим по адресу <https://localhost/> - там мы увидим ошибку сервиса, т.к. некстклауд не смог найти редис



**Шаг 13. 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’***



**Шаг 14. Возвращаемся в браузер и убеждаемся, что теперь форма входа в Nextcloud открывается без ошибок**



Вопросы для размышления:

* *Почему неправильная конфигурация на шаге 2 приведет (вероятнее всего) к полной потере сетевой доступности?   
  - Неправильная конфигурация сети в Proxmox может привести к потере доступности из-за ошибок в привязке моста, IP-настроек или конфликтов, что разрывает соединение с хостом*
* *Почему адрес шлюза (default gateway) виртуальной машины выглядит как 10.0.2.2, а не 10.0.2.1?   
  – 10.0.2.1 занят NAT, поэтому 10.0.2.2 является шлюзом для виртуальной машины*