

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ»

Факультет информационных технологий
Кафедра информационных систем и технологий

Лабораторная работа № 4
«Инфраструктура Docker»

Выполнил:
Студент: Водчиц Анастасия
ФИТ 3 курса 1 группы
Преподаватель: Сазонова Д.В.

Минск 2025

Цель: освоить навыки развертывания контейнера из образа и его настройки в среде docker, передачи файлов между операционной системой и контейнером docker, запуска интерактивных приложений внутри контейнера и управления контейнером.

Docker — это платформа, которая **упаковывает** приложение и все его зависимости (библиотеки, настройки) в **изолированный контейнер**.

Главная идея: Приложение в контейнере будет работать **одинаково** где угодно, решая проблему "у меня работает, а у тебя нет".

Задачи:

1. на виртуальной машине через утилиту apt установить клиентский пакет **docker** и серверный пакет **dockerd**;

Обновляем пакеты на виртуальной машине

```
vm-admin@server:~$ sudo apt update
[sudo] password for vm-admin:
0шб:1 http://by.archive.ubuntu.com/ubuntu bionic InRelease
  Что-то странное произошло при разрешении «172.16.0.101:3128» (-9 - Address family for hostname not supported)
0шб:2 http://by.archive.ubuntu.com/ubuntu bionic-updates InRelease
  Что-то странное произошло при разрешении «172.16.0.101:3128» (-9 - Address family for hostname not supported)
0шб:3 http://by.archive.ubuntu.com/ubuntu bionic-backports InRelease
  Что-то странное произошло при разрешении «172.16.0.101:3128» (-9 - Address family for hostname not supported)
0шб:4 http://by.archive.ubuntu.com/ubuntu bionic-security InRelease
  Что-то странное произошло при разрешении «172.16.0.101:3128» (-9 - Address family for hostname not supported)
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Все пакеты имеют последние версии.
W: Не удалось получить http://by.archive.ubuntu.com/ubuntu/dists/bionic/InRelease Что-то странное произошло при разрешении «172.16.0.101:3128» (-9 - Address family for hostname not supported)
W: Не удалось получить http://by.archive.ubuntu.com/ubuntu/dists/bionic-updates/InRelease Что-то странное произошло при разрешении «172.16.0.101:3128» (-9 - Address family for hostname not supported)
W: Не удалось получить http://by.archive.ubuntu.com/ubuntu/dists/bionic-backports/InRelease Что-то странное произошло при разрешении «172.16.0.101:3128» (-9 - Address family for hostname not supported)
W: Не удалось получить http://by.archive.ubuntu.com/ubuntu/dists/bionic-security/InRelease Что-то странное произошло при разрешении «172.16.0.101:3128» (-9 - Address family for hostname not supported)
W: Некоторые индексные файлы скачать не удалось. Они были проигнорированы, или вместо них были использованы старые версии.
vm-admin@server:~$
```

Эта команда устанавливает на Linux) с правами администратора (sudo):

1. **docker.io**: Сам Docker (основной движок для запуска контейнеров).
2. **docker-compose**: Инструмент для удобного управления несколькими связанными Docker-контейнерами (например, веб-сервер + база данных).

```

v@xxx:~$ sudo apt install docker.io docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz python3-compose python3-docker python3-dockerpty python3-dockerpty python3-dockerpty python3-dockerpty python3-dockerpty python3-pack
  python3-texttable python3-websocket runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount l cgroup-lite debotstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse l zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker-compose docker.io pigz python3-compose python3-docker python3-dockerpty python3-dockerpty python3-dockerpty python3-pack
  python3-texttable python3-websocket runc ubuntu-fan
0 upgraded, 17 newly installed, 0 to remove and 128 not upgraded.
Need to get 78.9 MB of archives.
After this operation, 304 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

sudo systemctl start docker - Запустите службу Docker
 sudo systemctl enable docker - Включите службу

```

v@xxx:~$ sudo systemctl start docker
v@xxx:~$ sudo systemctl enable docker
v@xxx:~$

```

```

v@xxx:~$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
v@xxx:~$ _

```

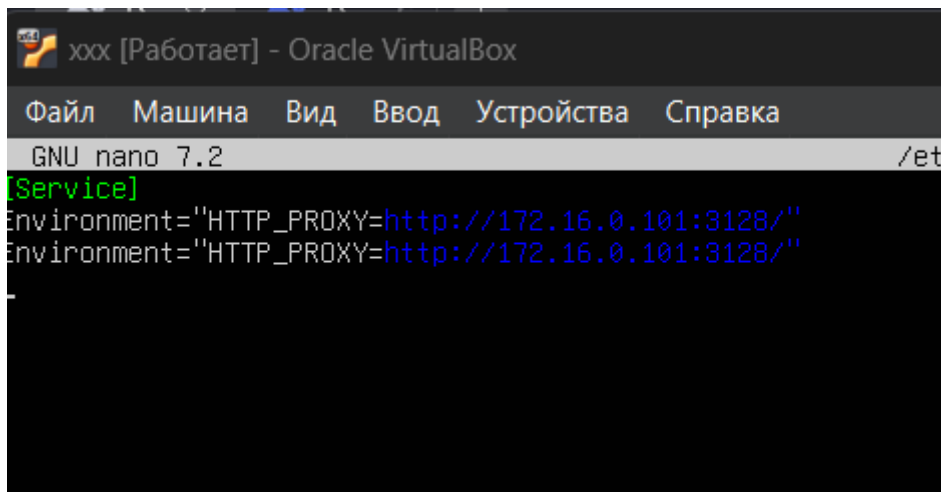
2. . при использовании университетской сети сконфигурировать dockerd на использование прокси-сервера 172.16.0.101:3128;

1. sudo mkdir -p /etc/systemd/system/docker.service.d:
 - Создает директорию /etc/systemd/system/docker.service.d.
 - Эта директория используется для дополнительных (переопределяющих) настроек системного сервиса Docker, управляемого systemd. Опция -p создает родительские директории, если их нет.
2. sudo nano /etc/systemd/system/docker.service.d/http-proxy.conf:
 - Открывает (или создает) файл http-proxy.conf в этой директории с помощью текстового редактора nano.
 - Цель — добавить или изменить конфигурацию Docker, в данном случае (http-proxy.conf), для настройки HTTP/HTTPS прокси, который Docker будет использовать (например, для скачивания образов).

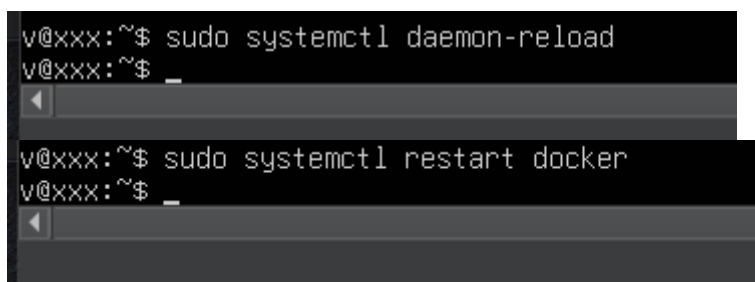
```

Docker version 26.1.3, build 26.1.3-0ubuntu1 24.04.1
v@xxx:~$ sudo mkdir -p /etc/systemd/system/docker.service.d
v@xxx:~$

```



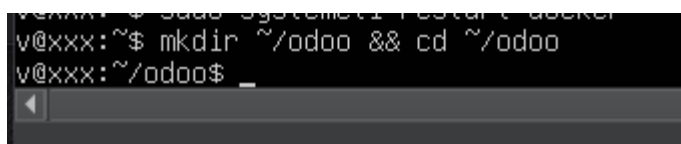
1. `sudo systemctl daemon-reload`: Заставляет систему перечитать измененные файлы конфигурации служб.
2. `sudo systemctl restart docker`: Перезапускает службу Docker, чтобы она начала работать с примененными новыми настройками.



3. выполнить установку ERP-системы odoo с помощью docker и инструкций на сайте hub.docker.com, согласиться на внесение в базу данных демонстрационных данных;

Odoo — это ERP-система (Enterprise Resource Planning), то есть комплекс интегрированных бизнес-приложений для управления основными процессами компании.

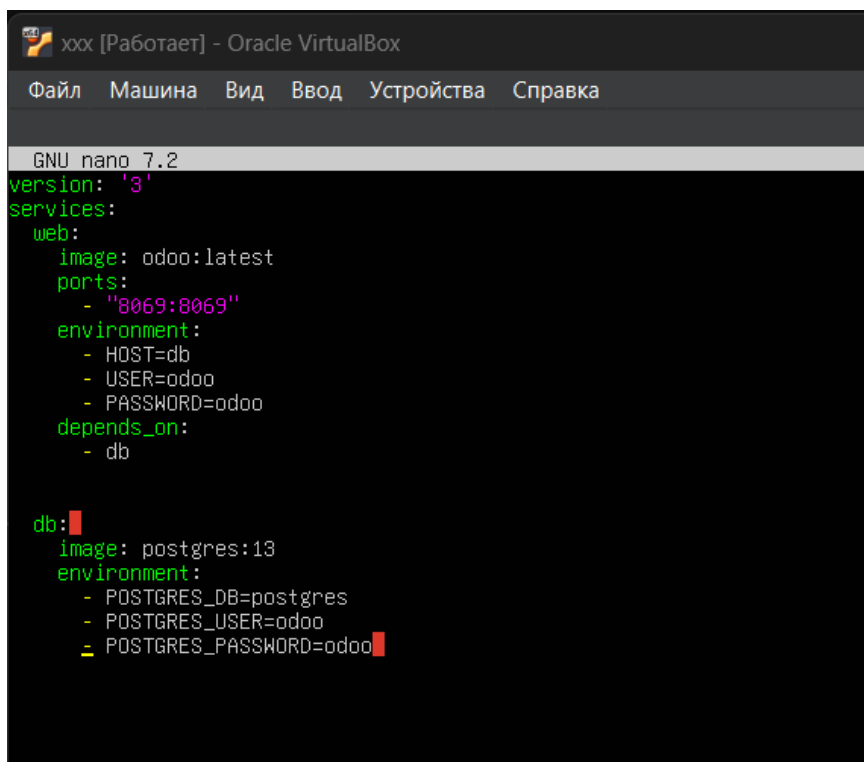
`mkdir ~/odoo && cd ~/odoo` - Создайте директорию для проекта Odoo



Что такое `docker-compose.yml`?

Это конфигурационный файл для инструмента Docker Compose. Он описывает, как запустить одно или несколько связанных Docker-контейнеров (сервисов), их настройки, связи между ними (сети), используемые диски (тома) и другие параметры.

- web: Запускает Odoo (odoo:latest), делает его доступным на порту 8069 и настраивает подключение к базе данных с именем хоста db и пользователем/паролем odoo/odoo. Зависит от запуска сервиса db.
- db: Запускает базу данных PostgreSQL версии 13 (postgres:13) и создает пользователя odoo с паролем odoo.
- Odoo (web) будет автоматически подключаться к базе данных (db), так как они настроены для работы в одной сети Docker Compose.



```

GNU nano 7.2
version: '3'
services:
  web:
    image: odoo:latest
    ports:
      - "8069:8069"
    environment:
      - HOST=db
      - USER=odoo
      - PASSWORD=odoo
    depends_on:
      - db
  db:
    image: postgres:13
    environment:
      - POSTGRES_DB=postgres
      - POSTGRES_USER=odoo
      - POSTGRES_PASSWORD=odoo

```

Эта команда запускает все сервисы (контейнеры), описанные в файле docker-compose.yml в текущей директории, в фоновом режиме (-d).

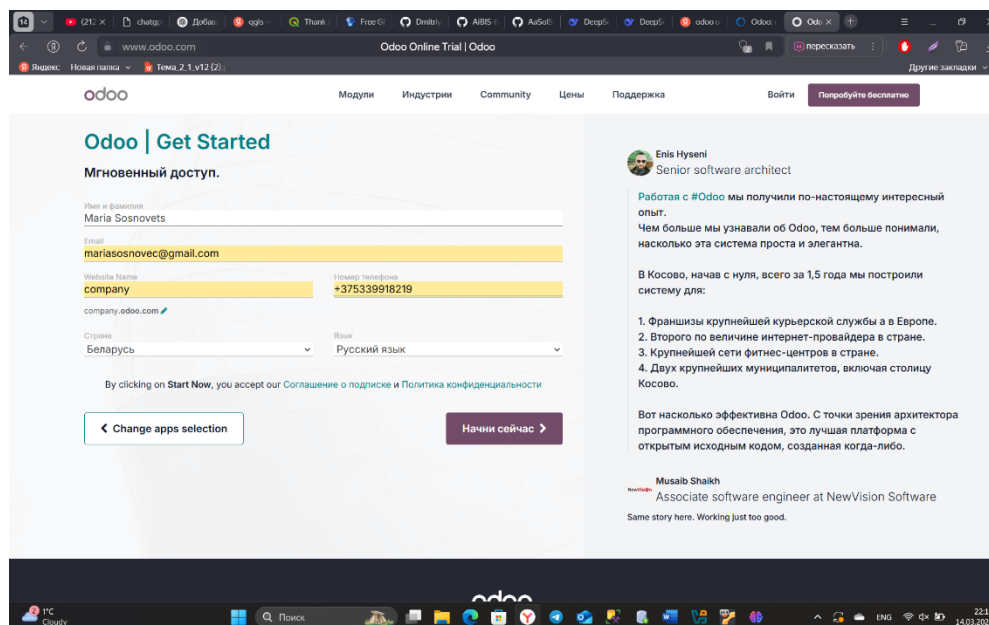
```

v@xxx:~$ docker-compose up -d

```

4. сконфигурировать ERP-систему odoo на использование SMTP-сервера postfix, входящего в состав пакета iRedMail, для отправки e-mail;

1. Войдите в админ-панель Odoo.
2. Перейдите в Настройки > Технические > Электронная почта > SMTP-серверы.
3. Добавьте новый SMTP-сервер с параметрами:
 - SMTP-сервер: postfix (или IP-адрес вашего сервера Postfix)
 - Порт: 25
 - Логин и Пароль: если требуется аутентификация, укажите данные от почтового ящика.



5. сконфигурировать SMTP-сервер postfix на прием писем из docker-контейнера с ERP-системой odoo;

Настройте Postfix на сервере, чтобы он принимал соединения от Docker-контейнера. Отредактируйте файл /etc/postfix/main.cf

/etc/postfix/main.cf — это главный конфигурационный файл почтового сервера **Postfix**.

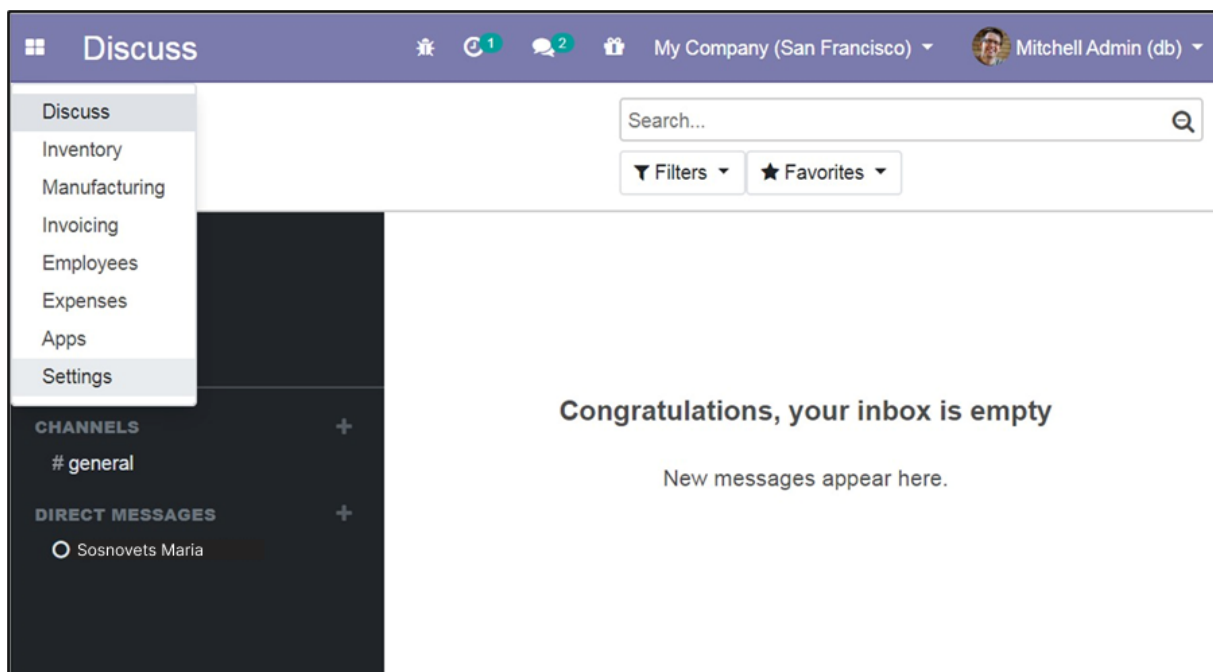
Эта строка в файле /etc/postfix/main.cf **разрешает** почтовому серверу Postfix **принимать и пересылать (релеить) почту без аутентификации** от клиентов, находящихся в следующих сетях:

1. 127.0.0.0/8: Локальный хост (IPv4) - т.е. сама машина, где работает Postfix.
2. [::1]/128: Локальный хост (IPv6).
3. 172.17.0.0/16: **Стандартная сеть Docker по умолчанию (bridge).**

```
mynetworks = 127.0.0.0/8 [::1]/128 172.17.0.0/16
```

```
sudo systemctl restart postfix
```

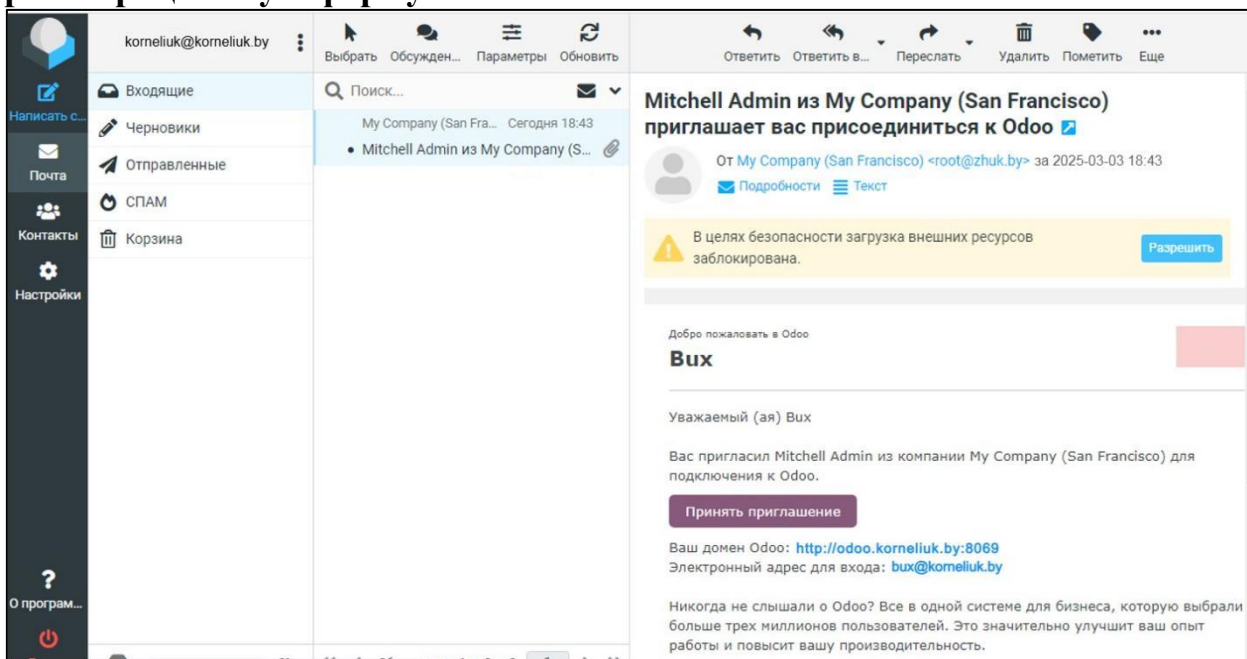
6. создать почтовый ящик на почтовом сервере с помощью панели iRedAdmin;

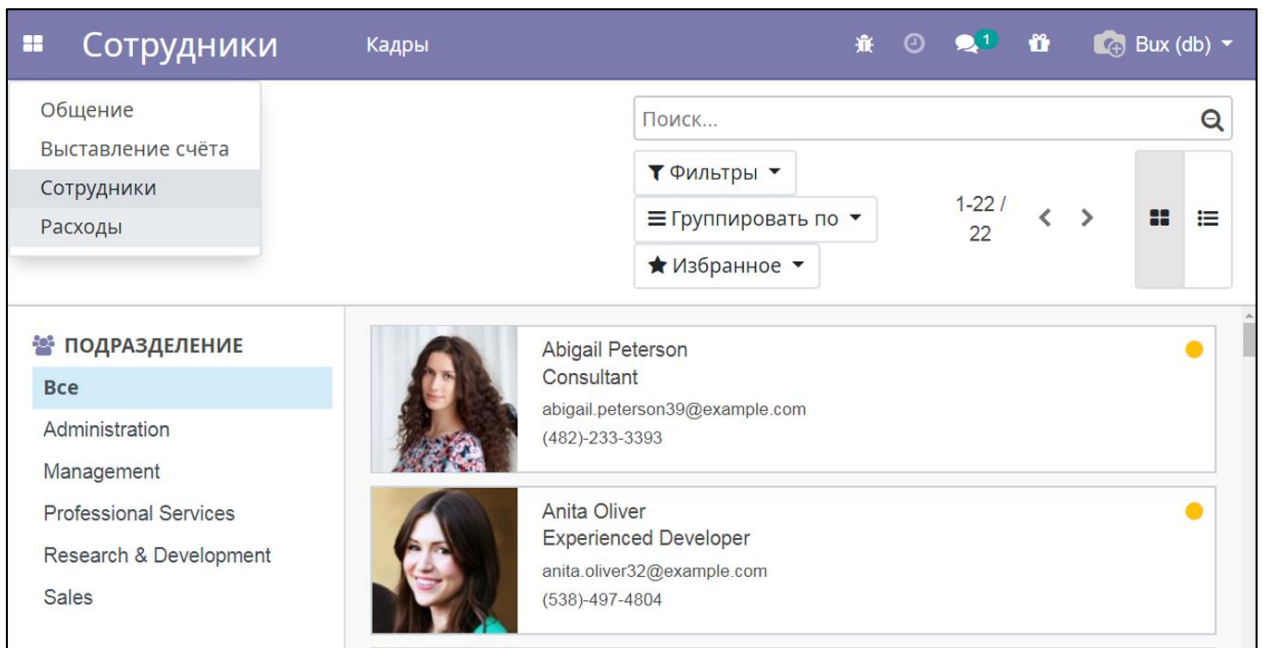


7. создать пользователя в ERP-системе odoo, назначить ему права на использование только бухгалтерских компонентов, выслать подтверждение на созданный почтовый ящик;

1. Войдите в Odoo как администратор.
2. Перейдите в Пользователи > Создать.
3. Заполните данные пользователя, назначьте права только на бухгалтерские компоненты.
4. Укажите почтовый адрес, созданный в iRedAdmin, и отправьте приглашение.

8. выполнить регистрацию пользователя в ERP-системе odoo, перейдя по ссылке в полученном на почтовый ящик письме и заполнив регистрационную форму.





9. определите пути в ОС Ubuntu и внутри контейнера odoo к папке, используемой odoo для размещения дополнительных модулей, с помощью команды

docker inspect odoo | less

```
docker inspect odoo | less
v@xxx:~$ docker inspect odoo | less
[]
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/containers/odoo/json": dial unix /var/run/docker.sock: connect: permission denied
[END]
```

10. запустите командную строку bash внутри контейнера командой

docker exec -it odoo /bin/bash

```
docker exec -it odoo /bin/bash
v@xxx:~$ docker exec -it odoo /bin/bash
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/containers/odoo/json": dial unix /var/run/docker.sock: connect: permission denied
v@xxx:~$
```

11. создайте заготовку модуля в папке для дополнительных модулей командой

odoo scaffold test_module /mnt/extra-addons/

12. выйдите из командной строки внутри контейнера командой

13. на физической машине для удобства редактирования исходного кода модуля скачайте исходный код odoo с github:

<https://github.com/odoo/odoo>

```
D:\masha\university\6_sem\AIBIS\Lab_05>git clone https://github.com/odoo/odoo.git
Cloning into 'odoo'...
remote: Enumerating objects: 5171261, done.
remote: Counting objects: 100% (996/996), done.
remote: Compressing objects: 100% (661/661), done.
Receiving objects: 0% (18738/5171261), 7.04 MiB | 4.61 MiB/s
```

14. распакуйте скачанный исходный код в отдельную папку;

15. скопируйте по FTP с виртуальной машины созданную ранее заготовку модуля (папку) test_module на физическую машину в папку /addons

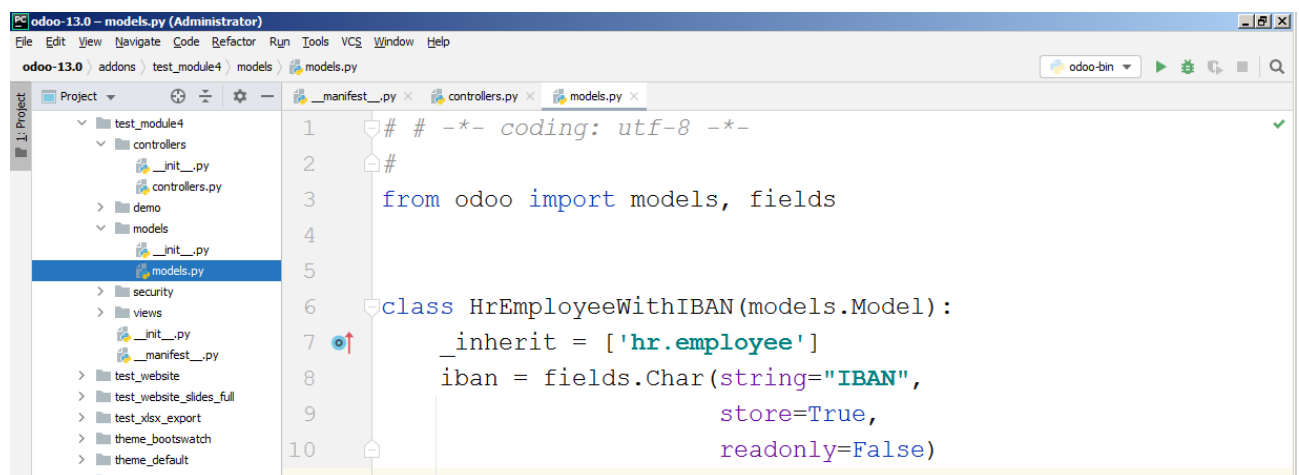
внутри папки с исходным кодом odoo; не перепутайте с папкой /odoo/addons;

16. для редактирования исходного кода модуля на языке Python целесообразно скачать и установить среду разработки PyCharm Community Edition, при установке поставьте галочку «Add “Open Folder as Project”»;

17. нажмите правой кнопкой мыши по папке с исходным кодом odoo и выберите пункт «Open Folder as PyCharm Community Edition Project»;

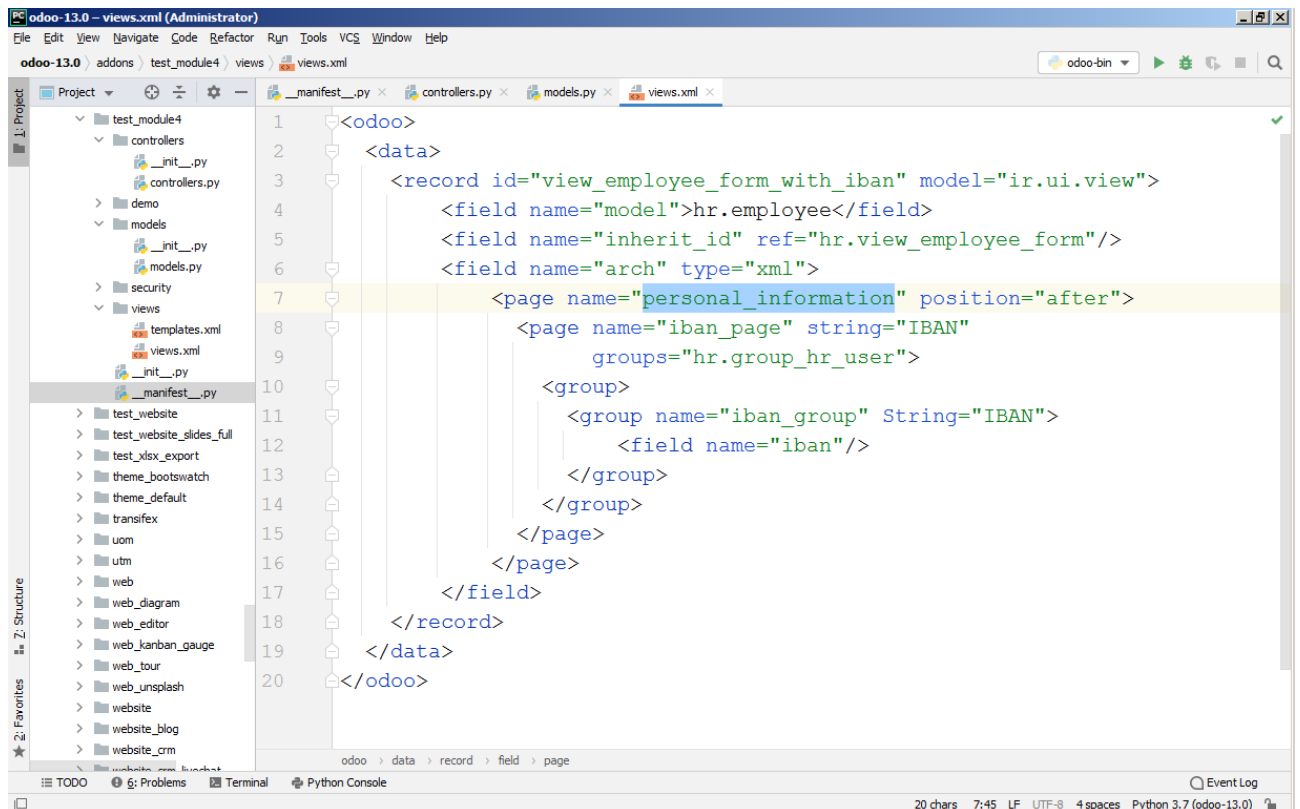
18. на панели Project найдите заготовку модуля (папку);

19. в первую очередь необходимо обеспечить хранение номеров счетов работников с помощью модели, поэтому отредактируйте файл /models/models.py внутри папки с заготовкой модуля следующим образом для добавления поля iban к существующей модели hr.employee:



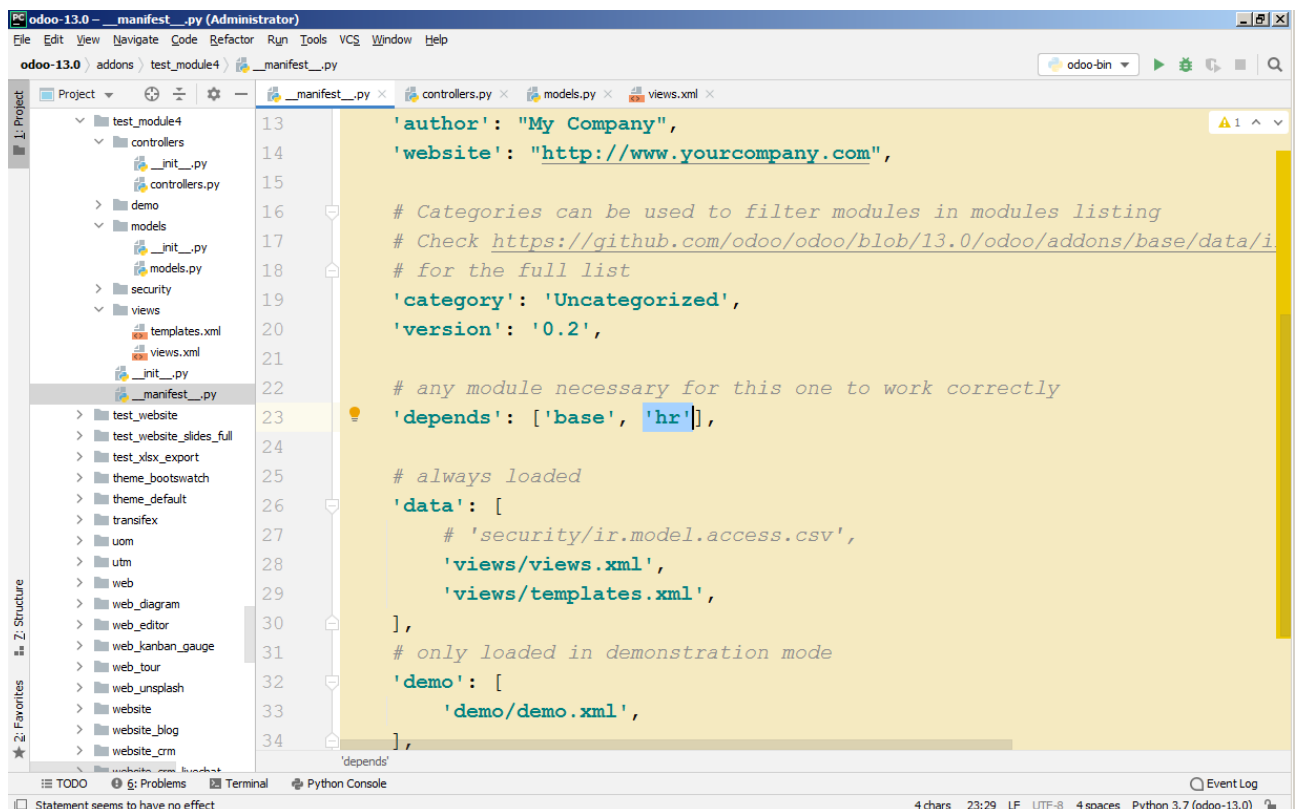
```
1  # -*- coding: utf-8 -*-
2  #
3  from odoo import models, fields
4
5
6  class HrEmployeeWithIBAN(models.Model):
7      _inherit = ['hr.employee']
8      iban = fields.Char(string="IBAN",
9                          store=True,
10                         readonly=False)
```

20. для обеспечения возможности редактирования номера счета на веб-страницах odoo добавьте вкладку с полем для ввода/вывода номера счета в существующее представление hr.view_employee_form, например, после вкладки personal_information, отредактировав файл /views/views.xml следующим образом:



```
1 <odoo>
2 <data>
3   <record id="view_employee_form_with_iban" model="ir.ui.view">
4     <field name="model">hr.employee</field>
5     <field name="inherit_id" ref="hr.view_employee_form"/>
6     <field name="arch" type="xml">
7       <page name="personal_information" position="after">
8         <page name="iban_page" string="IBAN"
9           groups="hr.group_hr_user">
10          <group>
11            <group name="iban_group" String="IBAN">
12              <field name="iban"/>
13            </group>
14          </group>
15        </page>
16      </page>
17    </field>
18  </record>
19 </data>
20 </odoo>
```

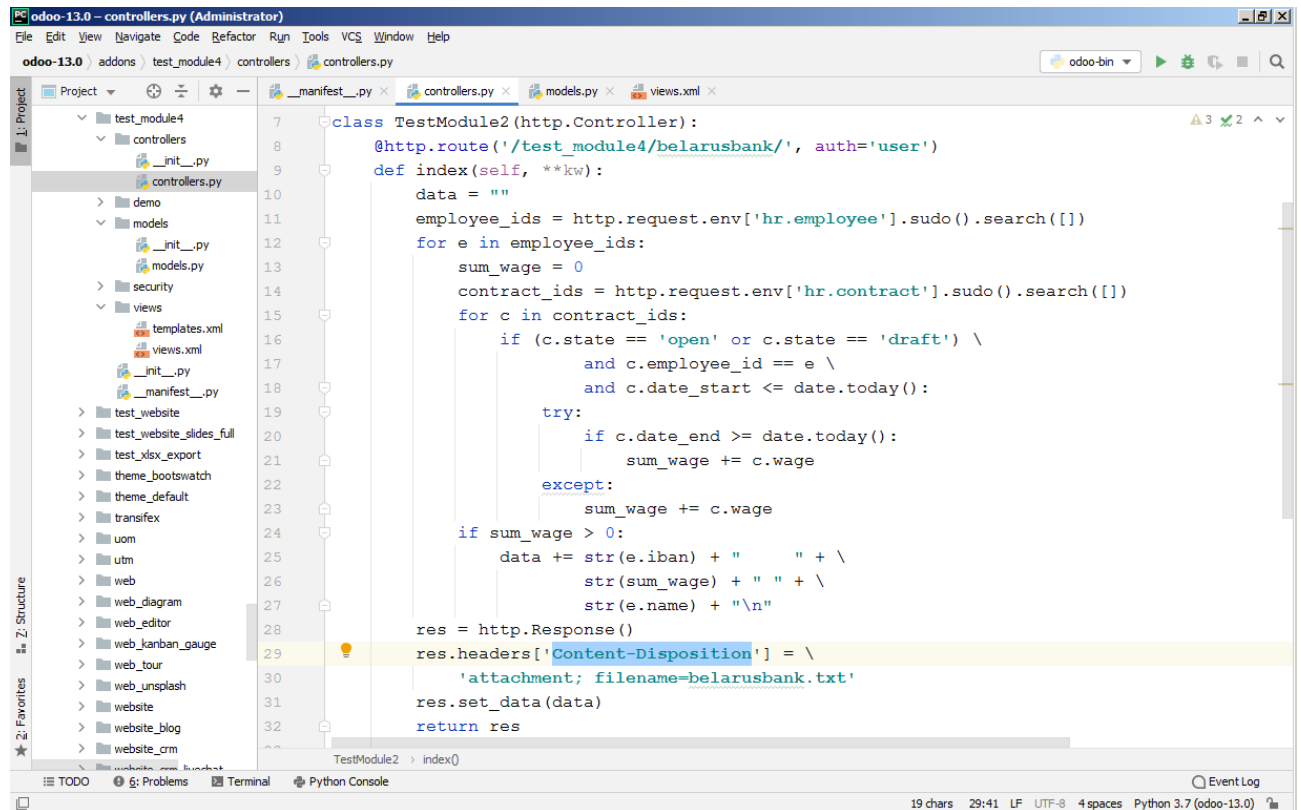
21. поскольку наше расширяющее представление привязывается к модели из модуля hr, укажите данный модуль в списке зависимостей нашего модуля в файле `__manifest__.py`:



```
13 'author': "My Company",
14 'website': "http://www.yourcompany.com",
15
16 # Categories can be used to filter modules in modules listing
17 # Check https://github.com/odoo/odoo/blob/13.0/odoo/addons/base/data/i
18 # for the full list
19 'category': 'Uncategorized',
20 'version': '0.2',
21
22 # any module necessary for this one to work correctly
23 'depends': ['base', 'hr'],
24
25 # always loaded
26 'data': [
27     # 'security/ir.model.access.csv',
28     'views/views.xml',
29     'views/templates.xml',
30 ],
31 # only loaded in demonstration mode
32 'demo': [
33     'demo/demo.xml',
34 ],
```

22. для импорта в клиент-банк требуется генерировать простой текстовый файл, поэтому требуется использовать не представление, а контроллер; создайте в файле `/controllers/controllers.py` контроллер с

привязанным к определенному URL методом `index`, перебирающим всех работников, суммирующим для каждого работника оплату по всем действующим на текущую дату контрактам, формирующим файл из трех колонок, разделенных пробелами (IBAN, сумма выплат по контрактам, ФИО работника) и добавляющим перед файлом заголовок `Content-Disposition` для загрузки этого файла, а не открытия в браузере:



```
7 class TestModule2(http.Controller):
8     @http.route('/test_module4/belarusbank/', auth='user')
9     def index(self, **kw):
10         data = ""
11         employee_ids = http.request.env['hr.employee'].sudo().search([])
12         for e in employee_ids:
13             sum_wage = 0
14             contract_ids = http.request.env['hr.contract'].sudo().search([])
15             for c in contract_ids:
16                 if (c.state == 'open' or c.state == 'draft') \
17                     and c.employee_id == e \
18                     and c.date_start <= date.today():
19                     try:
20                         if c.date_end >= date.today():
21                             sum_wage += c.wage
22                     except:
23                         sum_wage += c.wage
24             if sum_wage > 0:
25                 data += str(e.iban) + " " + \
26                     str(sum_wage) + " " + \
27                     str(e.name) + "\n"
28         res = http.Response()
29         res.headers['Content-Disposition'] = \
30             'attachment; filename=belarusbank.txt'
31         res.set_data(data)
32         return res
```

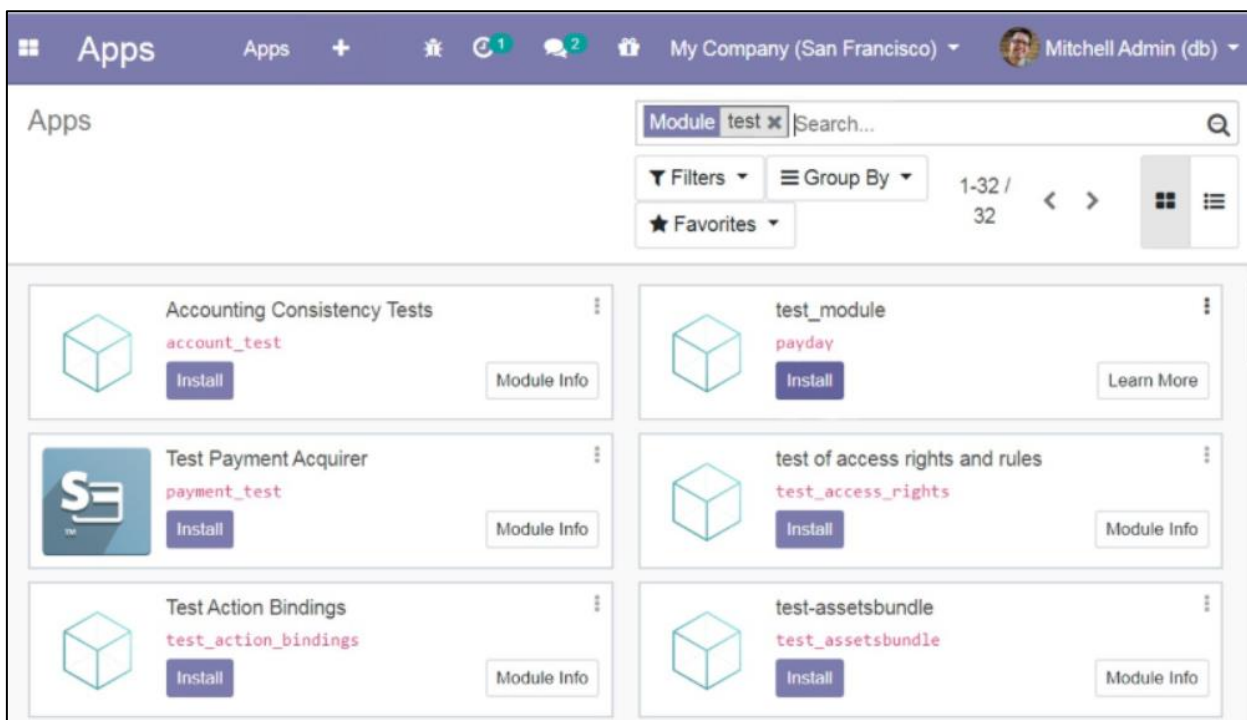
23. добавьте в созданное ранее представление в файле `/views/views.xml` ссылку на метод созданного контроллера:

The screenshot shows the Odoo-13.0 web editor interface. The left sidebar displays a project tree with folders like 'test_module4', 'controllers', 'models', 'security', and 'views'. The main editor area shows the XML code for a view named 'view_employee_form_with_iban'. The code is as follows:

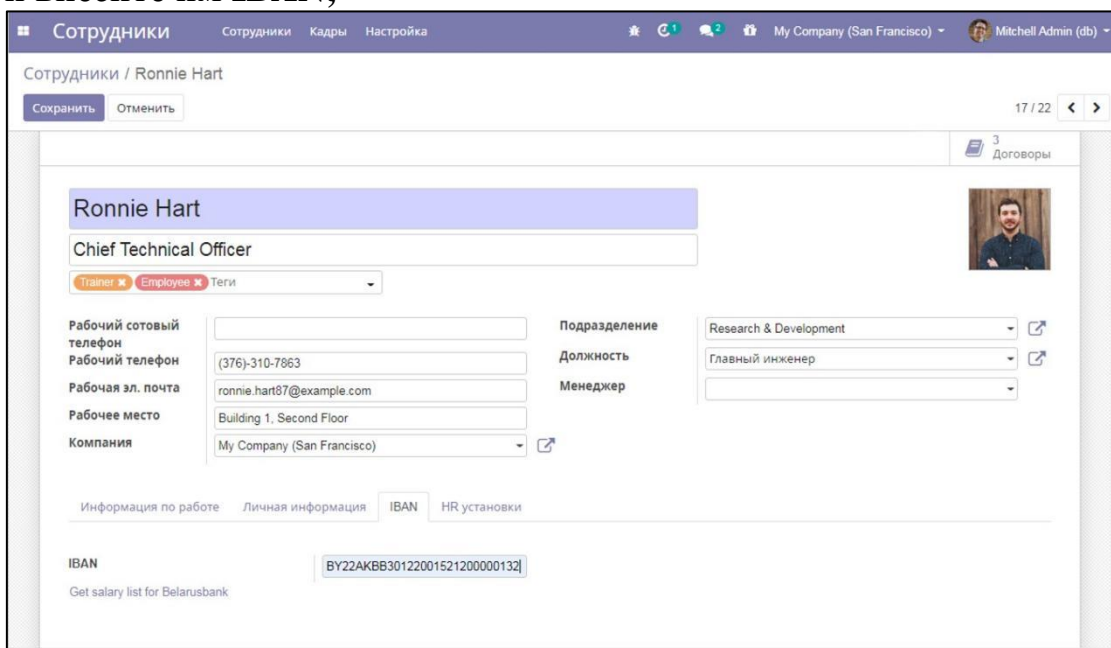
```
<odoo>
<data>
  <record id="view_employee_form_with_iban" model="ir.ui.view">
    <field name="model">hr.employee</field>
    <field name="inherit_id" ref="hr.view_employee_form"/>
    <field name="arch" type="xml">
      <page name="personal_information" position="after">
        <page name="iban_page" string="IBAN"
          groups="hr.group_hr_user">
          <group>
            <group name="iban_group" String="IBAN">
              <field name="iban"/>
              <a href="/test_module4/belarusbank">
                Get salary list for Belarusbank</a>
            </group>
          </group>
        </page>
      </page>
    </field>
  </record>
</data>
</odoo>
```

The status bar at the bottom indicates 120 chars, 1 line break, 13:1 LF, UTF-8, 4 spaces, Python 3.7 (odoo-13.0).

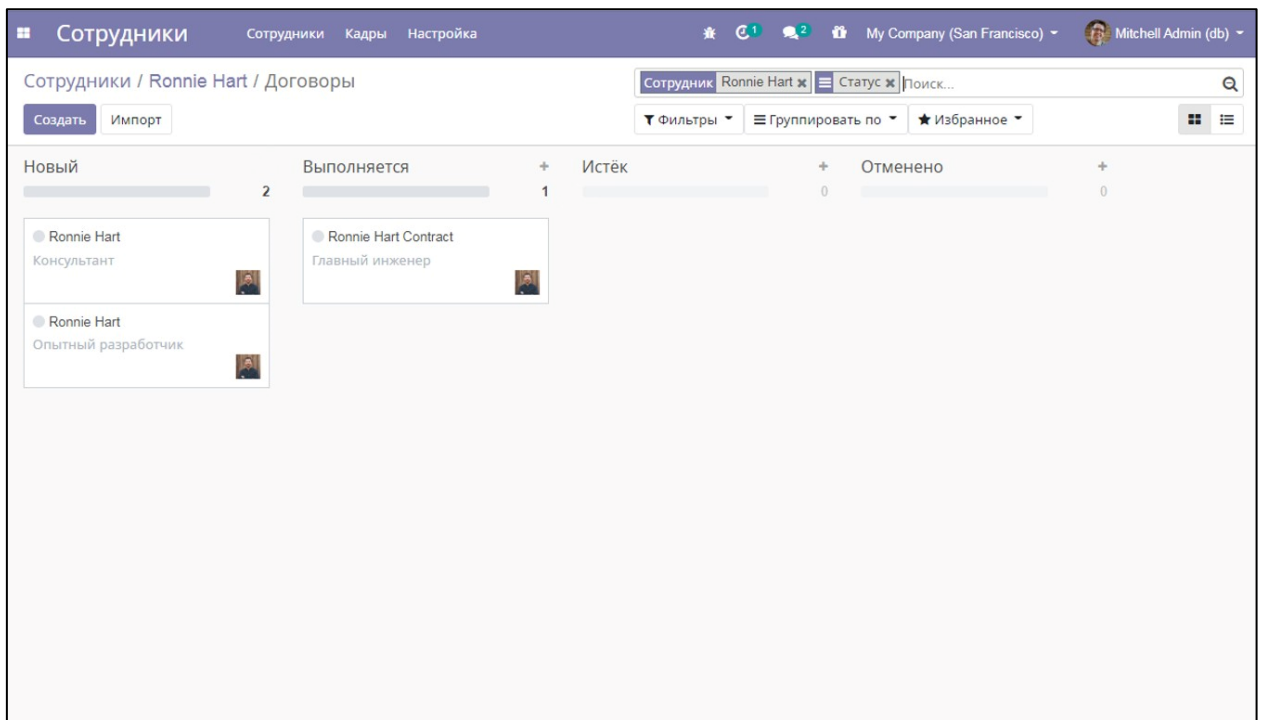
24. сохраните все отредактированные файлы;
25. передайте отредактированный модуль обратно на виртуальную машину, в папку для дополнительных модулей odoo; при отсутствии прав на запись выдайте права на папку командой `chmod` всем пользователям;
26. для загрузки информации о новом модуле в odoo перезапустите контейнер командами
`docker stop odoo`
`docker start odoo`
27. для установки собственного модуля включите режим разработчика в настройках odoo;
28. установите созданный модуль через пункт главного меню «Приложения», найти свой модуль можно по названию с помощью панели поиска;



29. найдите сотрудников с действующими на текущую дату контрактами и внесите им IBAN;



30. найдите хотя бы одного сотрудника с несколькими действующими на текущую дату контрактами или добавьте новый контракт сотруднику с одним действующим контрактом;



31. нажмите на ссылку «Get salary list for Belarusbank» и убедитесь в корректности формирования списка на выплату зарплат: сотрудник с несколькими действующими контрактами не должен повторяться, для него должна быть указана сумма выплат по всем действующим контрактам.

