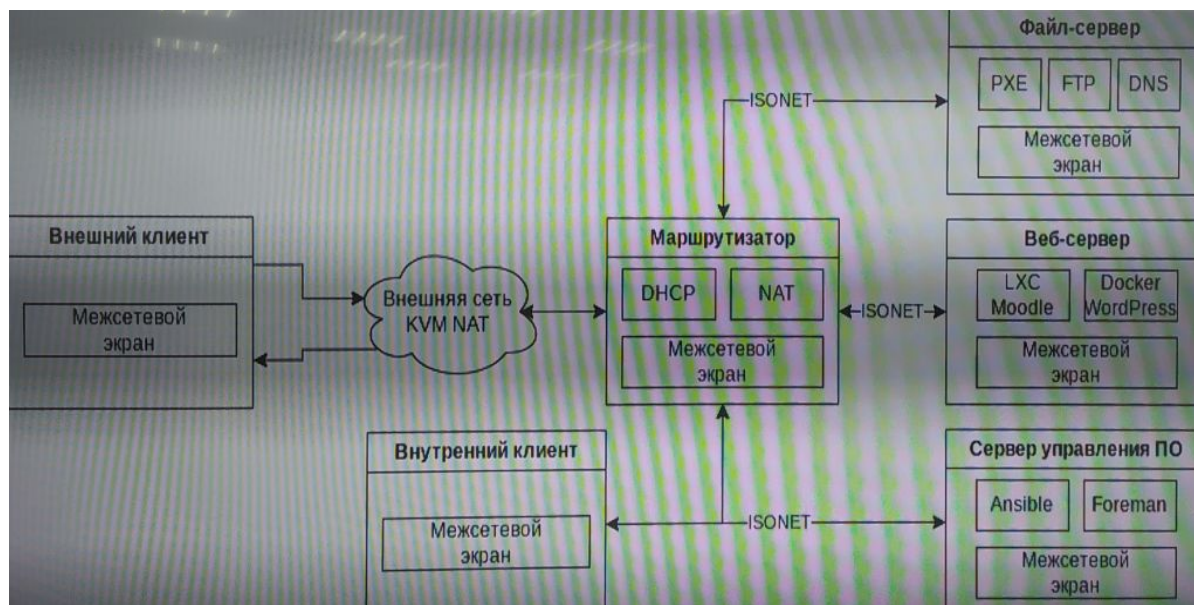


## ЭКЗАМЕН АСОС



### Условия

1. Сервер управления ПО должен иметь сценарий обновления внутреннего клиента после установки по PXE. Все процессы должны быть оптимизированы в полном объёме (preseed.cfg + postinstall.script)
2. Веб-приложения Moodle и WordPress и сетевой ресурс FTP должны быть доступны по доменному имени через внешний адрес маршрутизатора (NAT). Внутренний клиент должен обращаться к сетевым ресурсам также по домену через внешний IP адрес. (Правила dstNAT + DNS настроенный на внешний адрес в MT)
3. На всех устройствах должен быть включён и настроен межсетевой экран. Рекомендуется при настройке сети отключать МСЭ для проверки работоспособности система и включать его только тогда, когда все сетевые сервисы работают в штатном режиме.
4. Домен второго уровня (пример, необходимо указать свою фамилию) – ivanov.sa3. Таблица доменных имён:

Сетевой ресурс	Доменное имя (на базе примера)
FTP	ftp.ivanov.sa3
Moodle	moodle.ivanov.sa3
WordPress	wp.ivanov.sa3

Адрес-FTP: ftp://10.0.24.3

Для прокси (с FTP): apt.conf → /etc/apt/

Для гуляния по сети (с FTP): environment → /etc/

Разархивирование из командной строки: tar -xvf <путь до архива>

Выгрузка с FTP из командной строки: wget -P /home/\$user/Загрузки/  
/ftp://10.0.24.3/<путь к нужному файлу>

## **ВЫПОЛНЕНИЕ**

### **1. Маршрутизатор (MikroTik)**

#### *a. DHCP*

Первоначальная настройка в QuickSet:

RouterOS v7.2.1 (stable) Quick Set WebFig Terminal

Ethernet Quick Set

active

Configuration

Mode ☒ Router ☐ Bridge

Internet

Address Acquisition ☒ Static ☐ Automatic ☐ PPPoE

IP Address 192.168.217.252

Netmask 255.255.255.0 (/24)

Gateway 192.168.217.1

DNS Servers ▼

MAC Address 00:0C:29:24:90:BA

Local Network

IP Address 192.168.16.1

Netmask 255.255.255.0 (/24)

Bridge All LAN Ports ☐

DHCP Server ☒

DHCP Server Range ▲ 192.168.16.100-192.168.168

NAT ☒

Port Mapping

VPN

VPN Access ☐

VPN Address 192.168.217.252

System

Router Identity MikroTik

Check For Updates Reboot Reset Configuration

Password...

Apply Configuration

*рис.1 у меня NAT подсеть имела вид 192.168.217.0/24, вы изменяете всё под свою подсеть*

В Internet – настраивайте **IP Address** и **Gateway** исходя из подсети NAT.

В Local Network – **IP Address** и **DHCP Server Range** исходя из изолированной сети

В DNS Servers проставьте первый айпи вашего NAT – 192.168.122.1

Поставьте галочку на NAT и Apply Configuration

Далее в WebFig заходим в IP → DHCP Server → Leases  
Если здесь уже что-то есть – нажимаем “-”, затем снова заходим на Сервер и в терминале прописываем

`sudo dhclient`

Так мы отправляем запрос для получения ip адреса и как только он появляется во вкладке Leases на MikroTik – заходим в него и нажимаем на кнопку Make Static. Это необходимо, чтобы задать статический ip адрес серверу.

Для PXE сервера понадобится следующая настройка:

IP → DHCP Server → Network

Создаем новую или изменяем текущую запись и добавляем:

Next Server: 192.168.88.2

Boot File Name: pxelinux.0

Если это не поможет, то можно заморочиться с созданием Options и Options Set:

IP → DHCP Server → Options

Опция 66:

Name: option66

Code: 66

Value: s'192.168.88.2'

Опция 67:

Name: option67

Code: 67

Value: 'pxelinux.0'

IP → DHCP Server → Option Sets

Name: set-pxe

Options: option66, option67

### *b. NAT*

В том же WebFig заходим в IP → Firewall → NAT и в единственное правило:

Chain: srcnat

Out.interface: ether1 – интерфейс NAT

Action: masquerade

Для перенаправления адресов или портов используются следующие правила:

IP → Firewall → Nat → Add new

Chain: dst-nat

Dst. Address: <адрес Mikrotik>

Protocol: 6 (tcp) / 17 (udp)

Dst. Port: 53 (dns) / 80(http) / 20-21(ftp) / 22(ssh)

Action: dst-nat

Dst. Address: <адрес вашего сервера>

Dst. Port: 80

*b(1). don.DNS*

WebFig → IP → DNS

К уже имеющемуся NAT-овскому адресу добавьте адрес своего сервера, который вы сделали ему статичным – 192.168.16.? и поставьте галочку в пункте, который начинается с “Allow”

*c. Межсетевой экран*

## **2. Файл-сервер**

*a. PXE*

# Перед настройкой PXE система должна быть обновлена, должен быть настроен DHCP-сервер (в данном случае используется адрес узла 192.168.16.1). Также необходимо в пути файлов заменить имя пользователя "oge" на своё и изменить IP адреса на адрес своего устройства, куда планируется ставить PXE сервер. **ISO образ** и **архив grub загрузчика** должны быть **разархивированы** в каталог ~/Загрузки/ (ПКМ по файлу и “Распаковать во вложенную папку”. Образ AL\_SE загрузить с FTP и архив GRUB и preseed.cfg взять с [wiki.astralinux.ru](http://wiki.astralinux.ru) “Установка Astra Linux по сети”), или же вы можете заменить на путь к своим архивам.

# Установка нужных компонентов

```
apt install tftpd-hpa pxelinux syslinux apache2 -y
```

# Настройки Web-сервера Apache2

```
sed -i 's/# AstraMode on/AstraMode off/'
```

```
/etc/apache2/apache2.conf
```

```
chmod 755 /var/www/html
chown www-data:www-data /var/www/html
```

```
# Добавить папку для распакованного образа ОС (AstraLinux)
mkdir -p /var/www/html/repo
mkdir -p /var/www/html/repo/se
```

```
# Распакованный ISO образ ОС (AstraLinux) перекинуть из
Загрузок на web-сервер Apache2
cp /home/ore/Загрузки/AL_SE_1.7.3/* /var/www/html/repo/se -r
```

```
# Перекинуть linux и initrd.gz из ISO-netinst (закинуть образ
AstraLinux в Загрузки)
mkdir /srv/tftp/se/
cp /home/ore/Загрузки/AL_SE_1.7.3/netinst/linux /srv/tftp/se/
cp /home/ore/Загрузки/AL_SE_1.7.3/netinst/initrd.gz /srv/tftp/se/
```

```
# Скопировать в каталог /srv/tftp файл pxelinux.0 и
необходимые библиотеки syslinux из установленной ОС:
cp /usr/lib/PXELINUX/pxelinux.0 /srv/tftp/
cp
/usr/lib/syslinux/modules/bios/{chain.c32,ldlinux.c32,libcom32.c32,libuti
l.c32,menu.c32} /srv/tftp/
```

```
# Настройка pxelinux.cfg
mkdir -p /srv/tftp/pxelinux.cfg
touch /srv/tftp/pxelinux.cfg/default
```

```
# Как должен выглядеть файл default
ui menu.c32
PROMPT 0
#
MENU TITLE PXE Boot menu
TIMEOUT 50
DEFAULT local
#
LABEL local
```

```
MENU LABEL Boot from local drive
kernel chain.c32
append hd0
# Astra Linux Special Edition
LABEL netinstall
MENU LABEL Start Astra Linux SE net install
kernel se/linux
append initrd=se/initrd.gz modprobe.blacklist=evbug
auto=true priority=critical debian-installer/locale=en_US
console-keymaps-at/keymap=ru hostname=student domain=dedov.ded
astra-license/license=true url=tftp://192.168.16.1/se/preseed.cfg
interface=auto netcfg/dhcp_timeout=60 astra_install=1 vga=788
debian-installer/allow_unauthenticated=true
```

```
# Переписать конфиг загрузчика GRUB
sed -i 'd'
/home/ore/Загрузки/netinst/debian-installer/amd64/grub/grub.cfg
```

```
# Настроить конфиг загрузчика GRUB для ОС Astra Linux
(grub.cfg)
```

```
if loadfont $prefix/font.pf2 ; then
    set gfxmode=800x600
    set gfxpayload=keep
    insmod efi_gop
    insmod efi_uga
    insmod video_bochs
    insmod video_cirrus
    insmod gfxterm
    insmod png
    terminal_output gfxterm
fi
#
if background_image /isolinux/splash.png; then
    set color_normal=light-gray/black
    set color_highlight=white/black
else
    set menu_color_normal=cyan/blue
```

```

        set menu_color_highlight=white/blue
    fi
#
menuentry "Boot default EFI bootloader" {
    set background_color=black
    regexp -s root "\\((.+))" "$cmdpath"
    chainloader /EFI/Boot/Bootx64.efi
}
# Установка Astra Linux Special Edition
menuentry 'Install Astra Linux SE' {
    set background_color=black
    linux /se/linux modprobe.blacklist=evbug
    debian-installer/allow_unauthenticated=true auto=true priority=critical
    debian-installer/locale=en_US console-keymaps-at/keymap=ru
    hostname=student domain=dedov.ded astra-license/license=true
    url=tftp://192.168.16.1/se/preseed.cfg interface=auto
    netcfg/dhcp_timeout=60
    initrd /se/initrd.gz
}

# Отправить конфиг в tftp
cp /home/ore/Загрузки/netinst/* /srv/tftp/ -r

# Перезапустим службы
systemctl restart isc-dhcp-server.service
systemctl restart tftpd-hpa
systemctl restart apache2.service

# Настройка автоматической установки ОС в конфиге
preseed.cfg
touch /home/ore/Загрузки/preseed.cfg

# Настройка языка и страны согласно локации
d-i debian-installer/locale string ru_RU
d-i debian-installer/locale select ru_RU.UTF-8
d-i debian-installer/language string ru
d-i debian-installer/country string RU

```



```
d-i debian-installer/keymap string ru
#
```

```
# Выбор клавиатуры
```

```
d-i console-tools/archs select at
d-i console-keymaps-at/keymap select ru
d-i console-setup/toggle string Alt+Shift
d-i console-setup/layoutcode string ru
d-i keyboard-configuration/toggle select Alt+Shift
d-i keyboard-configuration/layoutcode string ru
d-i keyboard-configuration/xkb-keymap select ru
d-i languagechooser/language-name-fb select Russian
d-i countrychooser/country-name select Russia
#
```

```
# netcfg выберет активный сетевой интерфейс, если такой
имеется. Если есть более одного интерфейса, то при данной
настройке список интерфейсов выводится не будет.
```

```
d-i netcfg/choose_interface select auto
#
```

```
# Чтобы использовать нужный интерфейс:
```

```
#d-i netcfg/choose_interface select eth1
#
```

```
#non-free,contrib
```

```
d-i apt-setup/non-free boolean true
d-i apt-setup/contrib boolean true
#
```

```
# Имена хоста и домен, назначенные с помощью dhcp имеют
приоритет над значениями, установленными здесь. Однако,
установка значений всё ещё предотвращает показ вопросов, даже
если значения получены от dhcp.
```

```
d-i netcfg/get_hostname string astra
d-i netcfg/get_hostname seen true
d-i netcfg/get_domain string cct.rbt
#
```

```
d-i apt-setup/services-select none
d-i apt-setup/security_host string
#
```

# Выключить показ надоедливого диалога с WEP ключом.

```
d-i netcfg/wireless_wep string
#
```

# Репозиторий для установки

```
d-i mirror/protocol string http
d-i mirror/country string manual
d-i mirror/http/hostname string 192.168.16.1
d-i mirror/http/directory string /repo/se/
d-i mirror/http/proxy string
#
```

# Задаёт, установлены или нет аппаратные часы по Гринвичу.

```
d-i clock-setup/utc boolean true
#
```

# Здесь вы можете указать любое допустимое значение для \$TZ; смотрите допустимые значения в файле /usr/share/zoneinfo/.

```
d-i time/zone string Europe/Yekaterinburg
#
```

```
d-i partman-auto/method string regular
#
```

# Если один из дисков, который нужно разметить с помощью автоматической разметки, содержит старую конфигурацию LVM, то пользователю выдаётся предупреждение. Ответ можно автоматизировать...

```
d-i partman-auto/purge_lvm_from_device boolean true
# И подтвердить запись lvm разделов.
d-i partman-lvm/confirm boolean true
#
```

# Для режима UEFI:

```
d-i partman-auto/choose_recipe select
/lib/partman/recipes-amd64-efi/30atomic
#
```

# Этот параметр заставляет partman выполнить разметку автоматически без подтверждения.

```
d-i partman/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman-auto-crypto/erase_disks boolean true
d-i partman-basicfilesystems/no_swap boolean true
d-i partman-target/mount_failed boolean true
d-i partman-partitioning/unknown_label boolean true
d-i partman-auto/purge_lvm_from_device string true
d-i partman-lvm/vgdelete_confirm boolean true
d-i partman/confirm_write_new_label string true
d-i partman-lvm/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
#
```

# Устанавливаемый пакет (мета) с образом ядра; можно указать "none", если ядро устанавливать не нужно.

```
d-i base-installer/kernel/image string linux-image-generic
#
```

# Пропустить создание учётной записи суперпользователя (в обычной учётной записи пользователя можно использовать sudo).

```
#d-i passwd/root-login boolean false
```

# Или же пропустить создание обычной учётной записи пользователя.

```
d-i passwd/make-user boolean true
#
```

# Пароль суперпользователя, любой открытым текстом

```
d-i passwd/root-password password 12345678
```

```
d-i passwd/root-password-again password 12345678
```

# Пароль суперпользователя в виде MD5 хеша (P@ssw0rd)

# Для создания хэш пароля впишите в терминал: openssl  
passwd и результат, который вам выдаст и будет вашим  
зашифрованным паролем, главное его не забыть

```
d-i passwd/root-password-crypted password
```

```
$1$OdrGUEj5$DIiSWm9HkShgT7cZ7dIj//
```

```
#
```

# Создать учётную запись обычного пользователя.

```
d-i passwd/user-fullname string student
```

```
d-i passwd/username string student
```

```
#
```

# Пароль обычного пользователя открытым текстом

```
d-i passwd/user-password password 12345678
```

```
d-i passwd/user-password-again password 12345678
```

# Пароль обычного пользователя в виде MD5 хеша (P@ssw0rd)

```
d-i passwd/user-password-crypted password
```

```
$1$OdrGUEj5$DIiSWm9HkShgT7cZ7dIj//
```

```
#
```

# По умолчанию программа установки требует, чтобы  
репозитории аутентифицировались с помощью известного ключа  
gpg. Этот параметр выключает данную аутентификацию.

```
d-i debian-installer/allow_unauthenticated string true
```

# (выбрать необходимое ПО можно прописав в терминале  
команду tasksel - -list)

```
tasksel tasksel/first multiselect Base packages, Fly desktop,  
Internet suite, Ufw firewall, SSH server
```

```
tasksel tasksel/astra-feat-setup multiselect
```

```
#
```

```
d-i astra-additional-setup/os-check select Base security level Orel
```

```
d-i astra-additional-setup/additional-settings-orel multiselect
```

```
Disable ptrace capability
```

```
#
```

```
#astra-additional-setup astra-additional-setup/additional-settings  
multiselect
```

```
#d-i astra-additional-setup/additional-settings multiselect
```

```

#astra-additional-setup
astra-additional-setup/automatic-network-disable multiselect
#d-i astra-additional-setup/automatic-network-disable
multiselect
#d-i desktop-tablet-mode-switch/tablet-mode multiselect
#
tripwire tripwire/use-localkey boolean false
ripwire tripwire/use-sitekey boolean false
tripwire tripwire/installed note ok
portsentry portsentry/warn_no_block note ok
astra-license astra-license/license boolean true
krb5-config krb5-config/kerberos_servers string
libnss-ldapd libnss-ldapd/ldap-base string
libnss-ldapd libnss-ldapd/ldap-uris string
libnss-ldapd libnss-ldapd/nsswitch multiselect services
ald-client ald-client/make_config boolean false
ald-client ald-client/manual_configure false
astra-feat-setup astra-feat-setup/feat multiselect kiosk mode false
astra-feat-setup astra-feat-setup/feat multiselect Служба ALD false
d-i console-cyrillic/switch select "Клавиша Menu"
d-i console-cyrillic/toggle select Control+Shift
d-i samba-common/dhcp boolean false
d-i samba-common/workgroup string testgroup1
#

```

# Некоторые версии программы установки могут отсылать отчёт об установленных пакетах. По умолчанию данная возможность выключена, но отправка отчёта помогает проекту определить популярность программ и какие из них включать на CD.

```

popularity-contest popularity-contest/participate boolean false
#

```

# Эту переменную можно безопасно устанавливать, она указывает, что grub должен устанавливаться в MBR, если на машине не найдено другой операционной системы.

```

d-i grub-installer/only_debian boolean true
#

```

# Это заставляет grub-installer устанавливать в MBR, даже если на машине есть другая ОС, что менее безопасно, так как может привести к отказу загрузки этой другой ОС.

```
d-i grub-installer/with_other_os boolean true
#
```

# Необязательный пароль на grub, или в виде открытого текста в шифрованном виде с помощью хеша MD5, смотрите grub-md5-crypt(P@ssw0rd).

```
d-i grub-installer/password-crypted password
$1$OdrG$Uj5$DliSWm9HkShgT7cZ7dIj//
#
```

# Не показывать последнее сообщение о том, что установка завершена.

```
d-i finish-install/reboot_in_progress note
d-i finish-install/exit/poweroff boolean true
#
```

# Остановить систему после завершения установки, а не перегружаться в установленную систему.

```
#d-i debian-installer/exit/halt boolean true
```

# Эта настройка позволяет выключить питание машины, а не просто остановить её.

```
#d-i debian-installer/exit/poweroff boolean true
```

```
d-i preseed/late_command string in-target systemctl enable ssh ; \
                                in-target wget -O /usr/local/bin/update.sh
ftp://<адрес вашего FTP сервера>/update.sh; \
                                in-target chmod +x /usr/local/bin/update.sh; \
                                in-target wget -O
/lib/systemd/system/update.service ftp://<адрес вашего FTP
сервера>/update.service; \
                                in-target systemctl enable update.service;
```

```
#Скопировать preseed.cfg в каталог tftp  
cp /home/ore/Загрузки/preseed.cfg /srv/tftp/se/preseed.cfg
```

### *b. FTP*

Для FTP самое главное что необходимо, так это репозитории оперативного обновления, которые необходимо добавить в /etc/apt/sources.list на КАЖДОМ компьютере в системе во избежание различного рода ошибок.

```
# Основной репозиторий  
deb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-main/  
1.7_x86-64 main contrib non-free
```

```
# Оперативное обновление 1.7.3 основного репозитория  
deb  
https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.3/repository-update/  
1.7_x86-64 main contrib non-free
```

```
# Оперативное обновление 1.7.3.UU.1 основного репозитория  
deb  
https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.3/uu/1/repository-upda  
te/ 1.7_x86-64 main contrib non-free
```

После того как обновили систему – устанавливаем пакет FTP  
sudo apt install fly-admin-ftp

И в программе (sudo fly-admin-ftp) необходимо только разрешить анонимный доступ и перезагрузить службу.

Чтобы проверить работает ли он – создаём пару каталогов в корневой папке FTP

```
sudo mkdir /srv/ftp/salosalos -p  
sudo mkdir /srv/ftp/ukrainskiysalo -p
```

Когда нужно будет закинуть какой-либо файл на FTP пользуйтесь командой-образцом.

```
sudo cp <путь_к_файлу_который_надо_перекинуть> /srv/ftp/
```

Также здесь необходимо создать некоторые файлы, необходимые для автоматического обновления системы после установки по PXE

В корне создаём файл update.service

```
sudo touch /srv/ftp/update.service
```

И этот файл должен выглядеть так:

```
[Unit]
Description=Update system
After=rc-local.service network-online.target
Before=getty.target
Before=systemd-user-sessions.service
```

```
[Service]
StandardOutput=tty
Type=oneshot
ExecStart=/usr/local/bin/update.sh (если не работает, то
пропишите путь до своего FTP сервера)
```

```
[Install]
WantedBy=multi-user.target
```

Тут же создаём скрипт для автоматического обновления системы:

```
sudo touch /srv/ftp/update.sh
```

Который должен выглядеть примерно так:

```
#!/bin/sh
sed -i 'd' /etc/apt/sources.list
```

```
echo 'deb
https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-main/
1.7_x86-64 main contrib non-free' >> /etc/apt/sources.list
echo 'deb
https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-update/
1.7_x86-64 main contrib non-free' >> /etc/apt/sources.list
```



```

echo 'deb
https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-base/
1.7_x86-64 main contrib non-free' >> /etc/apt/sources.list
echo 'deb
https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/
1.7_x86-64 main contrib non-free' >> /etc/apt/sources.list
echo 'deb
https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/
1.7_x86-64 astra-ce' >> /etc/apt/sources.list

echo 'deb
https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-main/
1.7_x86-64 main contrib non-free #1' >> /etc/apt/sources.list
echo 'deb
https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.3/repository-update/
1.7_x86-64 main contrib non-free #1' >> /etc/apt/sources.list
echo 'deb
https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.3/uu/1/repository-upda
te/ 1.7_x86-64 main contrib non-free #1' >> /etc/apt/sources.list

apt update && apt full-upgrade -y
sed -i '/#1/d' /etc/apt/sources.list
/sbin/reboot

```

### с. DNS

Устанавливаем пакеты

```
sudo apt install fly-admin-bind dnsutils
```

sudo fly-admin-bind – запуск программы

Перед созданием новых зон – выключите bind9 (красный квадратик)

Создать новую зону → тип прямой → wr.<фамилия>.sa3 (ваш домен для nginx) → имя сервера имён wr.<фамилия>.sa3 ; почтовый адрес: wr.<фамилия>.sa3 → всё оставляем → Адрес: 192.168.16.? (адрес вашего сервера) → Завершить

Создать новую зону → тип обратный → 192.168.16.? → имя сервера

имён и почтовый адрес такие же → Адрес тот же 192.168.16.? →  
Завершить

`sudo nslookup wr.<фамилия>.sa3` – проверяем работу

Таких зон должно быть три согласно методичке:

`wr.<фамилия>.sa3`

`ftp.<фамилия>.sa3`

`moodle.<фамилия>.sa3`

После того как проверите работу DNS сервера на других машинах – возвращаетесь fly-admin-bind и все адреса в зонах заменяете на адрес своего Mikrotik

#### *d. Межсетевой экран*

### **3. Веб-сервер** (*выделите хотя бы 70ГБ места на ЖД*)

Перед любой из следующих настроек – осознаём, что все настройки будут крутиться на виртуальных машинах внутри виртуальной машины => необходимо провести преднастройочные работы, а именно:

Устанавливаем недостающие пакеты

`sudo apt install astra-kvm qemu-utils, qemu-system,  
gir1.2-spiceclientgtk-3.0`

Добавляем себя в группы для виртуализации

`sudo usermod -a -G kvm, libvirt, libvirt-qemu, libvirt-admin <имя  
пользователя>`

#### *a. nginx*

Заходим в конфиг nginx

`sudo nano /etc/nginx/sites-available/default`

Если отбросить лишнюю информацию, то конфиг должен выглядеть примерно так:

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /var/www/html;
```

```

# Add index.php to the list if you are using PHP
index index.php index.html index.htm
index.nginx-debian.html;

server_name wp.<фамилия>.sa3;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_pass http://<адрес виртуальной машины с
WordPress>;
    # proxy_cache_bypass $http_upgrade;
}

#
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    #
    # With php7.0-cgi alone:
    fastcgi_pass <адрес виртуальной машины с
WordPress>;
    # With php7.0-fpm:
    # fastcgi_pass unix:/run/php/php7.0-fpm.sock;
}

}

server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;

```

```

# Add index.php to the list if you are using PHP
index index.php index.html index.htm
index.nginx-debian.html;

server_name moodle.<фамилия>.sa3;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_pass http://<адрес виртуальной машины с
Moodle>;
    # proxy_cache_bypass $http_upgrade;
}

#
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    #
    # With php7.0-cgi alone:
    fastcgi_pass <адрес виртуальной машины с Moodle>;
    # With php7.0-fpm:
    # fastcgi_pass unix:/run/php/php7.0-fpm.sock;
}
}

```

#### *b. WordPress (AstraLinux1.7.3)*

# Устанавливаем базу

```

sudo apt install apache2 mariadb-server php7.3 php7.3-mysql
php7.3-fpm

```

# Настройка mariadb (sudo mariadb или sudo mysql)

```
# Создание пользователя: create user '<имя пользователя>'@'localhost'
identified by '<пароль>';
# Создание БД: create database <имя БД>;
# Выдача привелегий: grant select, insert, update, delete, index, create, alter,
drop, grant option on <имя БД>.* to '<имя пользователя>'@'localhost';
# Выдача всех привелегий кроме grant option: grant all privileges on <имя
БД>.* to '<имя пользователя>'@'localhost';
# Лишить привилегий: revoke delete, drop on <имя БД>.* from '<имя
пользователя>'@'localhost'
# Применение привелегий: flush privileges;
```

```
# Установка WordPress
```

```
    wget -c http://wordpress.org/latest.tar.gz
```

```
# Распаковка архива
```

```
    tar -xzf latest.tar.gz
```

```
# Скопировать файлы WordPress в каталог сайта
```

```
    rsync -av wordpress/* /var/www/html/
```

```
# Переименовать и настроить конфиг WordPress
```

```
    cp /var/www/html/wp-config-sample.php
```

```
    /var/www/html/wp-config.php
```

```
    rm /var/www/html/wp-config-sample.php
```

```
# Настройка БД для WordPress
```

```
    sed -i 's/database_name_here/<имя БД>/'
```

```
    /var/www/html/wp-config.php
```

```
    sed -i 's/username_here/<имя пользователя>/'
```

```
    /var/www/html/wp-config.php
```

```
    sed -i 's/password_here/<пароль>/' /var/www/html/wp-config.php
```

```
# Выдать права папке сайта
```

```
    chown -R www-data:www-data /var/www/html/
```

```
    chmod -R 755 /var/www/html/
```

```
# Перезапустить службы apache2
```

```
    systemctl restart apache2.service
```

*c. Moodle (Ubuntu 22.02)*

# Устанавливаем базу

```
sudo apt install apache2 mariadb-server php7.3 php7.3-mysql  
php7.3-fpm
```

# Настройка mariadb (sudo mariadb или sudo mysql)

# Создание пользователя: create user '<имя пользователя>'@'localhost'  
identified by '<пароль>';

# Создание БД: create database <имя БД>;

# Выдача привелегий: grant select, insert, update, delete, index, create, alter,  
drop, grant option on <имя БД>.\* to '<имя пользователя>'@'localhost';

# Выдача всех привелегий кроме grant option: grant all privileges on <имя  
БД>.\* to '<имя пользователя>'@'localhost';

# Лишить привилегий: revoke delete, drop on <имя БД>.\* from '<имя  
пользователя>'@'localhost'

# Применение привелегий: flush privileges;

# Загрузить архив moodle.tar с FTP в Загрузки

# Разархивировать архив в /var/www/html/ (Распаковать во Вложенную  
папку)

```
sudo cp ~/Загрузки/moodle/* /var/www/html/ -R
```

# Выдать права папке сайта

```
chown -R www-data:www-data /var/www/html/  
chmod -R 755 /var/www/html/
```

# Перезапустить службы apache2

```
systemctl restart apache2.service
```

*d. Межсетевой экран*

#### **4. Сервер управления ПО**

*a. Ansible*

Установите ansible

```
sudo apt install ansible
```

Откройте конфиг с пользователями

```
sudo nano /etc/ansible/hosts
```

Добавьте в таком формате адреса ваших компьютеров:

[fileservers]

<имя пользователя>@<адрес пользователя>

[webservers]

<имя пользователя>@<адрес пользователя>

[clients]

<имя пользователя>@<адрес пользователя>

### Простой путь:

На клиентах и сервере устанавливаем sshpass

```
sudo apt install sshpass
```

В конфиге /etc/ansible/ansible.cfg изменяем следующий параметр

```
host_key_checking = False
```

Далее проверяем

```
sudo ansible -m ping all --ask-pass
```

Если всё верно настроено, то должны вывестись зелёные строки ping pong

### Сложный путь:

Беспарольный доступ через ключи.

Подробная инструкция на [wiki.astralinux.ru](http://wiki.astralinux.ru) “Подключение по SSH без ввода пароля” (В помощь администратору)

### *b. Foreman*

Установить пакет foreman-installer:

```
sudo apt install foreman-installer
```

Запустить установщик:

```
sudo foreman-installer
```

Установить необходимые плагины:

```
sudo foreman-installer --enable-foreman-plugin-ansible
```

```
--enable-foreman-proxy-plugin-ansible
```

```
--enable-foreman-plugin-remote-execution
```

```
--enable-foreman-proxy-plugin-remote-execution-ssh
```

Создать и скопировать ключ для работы каждого клиента с сервера (команда выполняется на сервере от имени пользователя foreman-proxy, подключение выполняется от имени пользователя для удаленного доступа):

```
sudo -u foreman-proxy ssh-keygen -f  
~foreman-proxy/.ssh/id_rsa_foreman_proxy -N "  
sudo -u foreman-proxy ssh-copy-id -i  
~foreman-proxy/.ssh/id_rsa_foreman_proxy  
<имя_пользователя>@agent1.astra.lan
```

с. Межсетевой экран