Руководство по установке и настройке iBatyr WAF 2.15

г. Астана

ОГЛАВЛЕНИЕ

1. Схема работы и типовые конфигурации WAF

Интеллектуальный сетевой экран для защиты веб-приложений iBatyr (далее МЭ, iBatyr, iBatyr WAF) позволяет обеспечить эффективную защиту критичных веб-ресурсов от внешних атак, а также дает возможность осуществлять полный контроль над использованием приложений в разрешенных сценариях.

МЭ iBatyr обеспечивает выполнения ряда следующих функций:

- контроль и фильтрацию;
- идентификацию и аутентификацию пользователей, являющихся работниками оператора;
- регистрацию событий безопасности (аудит);
- бесперебойное функционирование и восстановление;
- тестирование и контроль целостности;
- управление (администрирование);
- взаимодействие с другими средствами защиты информации.
- iBatyr WAF работает в нескольких режимах:
- режим обратного прокси;
- режим пассивного захвата трафика;
- гибридный режим;
- распределенная конфигурация;
- распределенная конфигурация с кластером узлов управления.

1.1. Режим обратного прокси

В данном режиме входящий трафик проходит через iBatyr WAF, что дает возможность влиять на http-запросы и ответы (блокировать запросы, изменять запросы и ответы). Типовая схема включения iBatyr WAF в инфраструктуру «в разрыв» с возможностью блокирования атак показана на схеме ниже (см. схема 1).

Примечание: Несмотря на то, что iBatyr WAF установлен в разрыв, по умолчанию для создаваемых в системе приложений установлен режим мониторинга, в котором не происходит реального блокирования атак. Переключение в режим блокирования осуществляется отдельно для каждого защищаемого

приложения в разделе «Приложения» основного меню консоли управления.

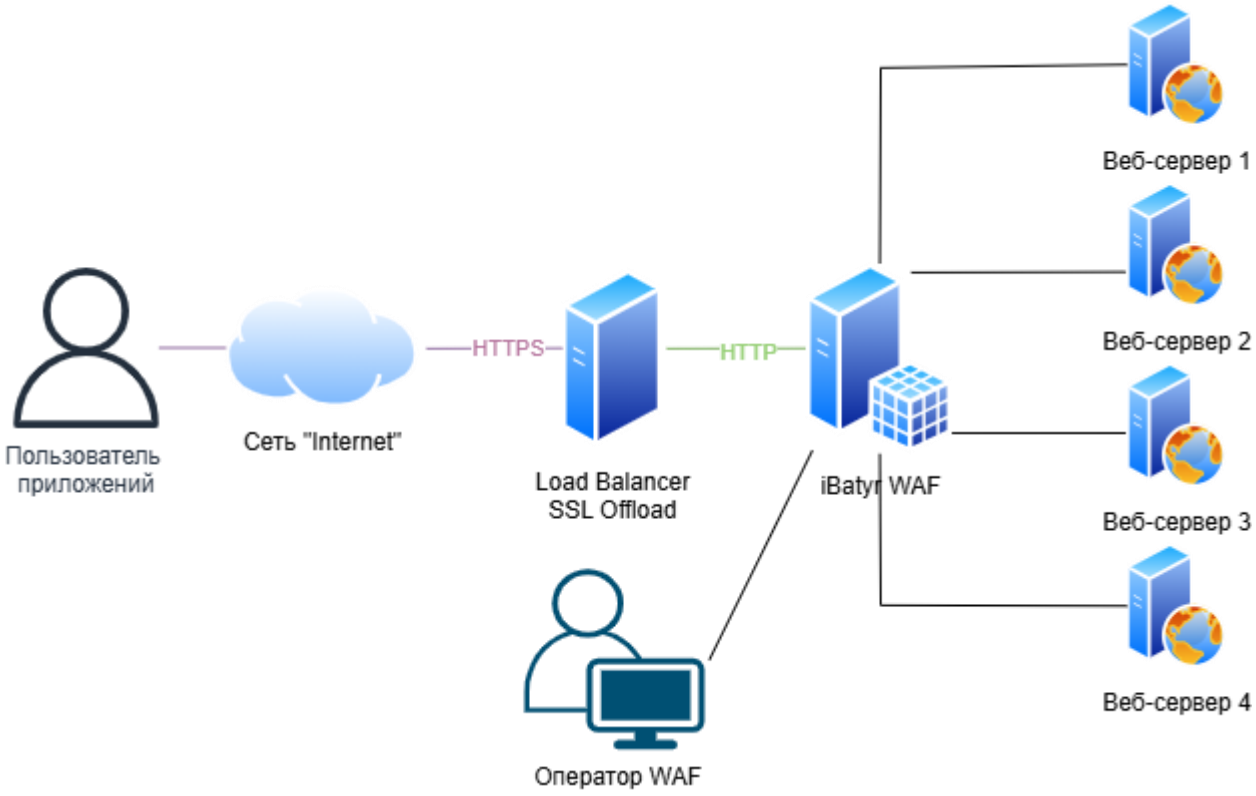


Схема 1 – Схема работы iBatyr WAF в режиме обратного прокси

Описание процесса установки iBatyr WAF в режиме обратного прокси дано в разделе 4.2.

1.2. Режим пассивного захвата трафика

Характерной особенностью режима пассивного захвата трафика (пассивный режим) является то, что работоспособность и доступность защищаемых приложений не зависит от работоспособности iBatyr WAF. iBatyr WAF получает копию трафика защищаемых приложений с сетевого устройства (это может быть, например, коммутатор или устройство распределения нагрузки сети) через SPAN-порт.

Схема включения iBatyr WAF в инфраструктуру в режиме работы с копией трафика представлена на схеме 2.

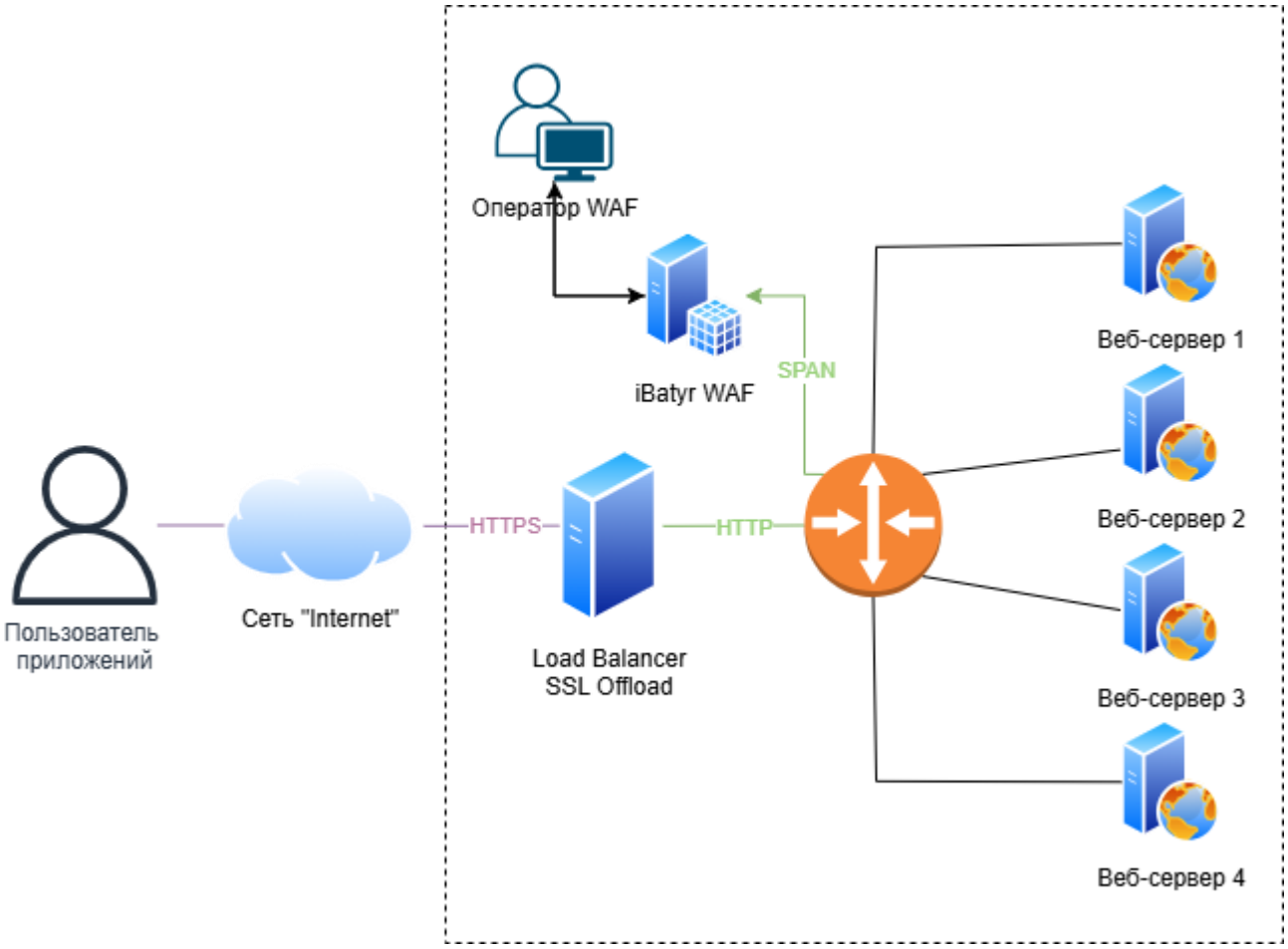


Схема 2 – Схема работы iBatyr WAF в режиме пассивного захвата трафика

В режиме мониторинга проводится пассивный анализ трафика без возможности блокирования запросов. Для работы iBatyr WAF в режиме мониторинга необходимы два сетевых интерфейса. Один интерфейс служит для приема копии трафика, второй – для управления, подключения к веб-интерфейсу и SSH консоли сервера.

Описание процесса установки iBatyr WAF в режиме пассивного захвата трафика дано в разделе 4.3.

Данные режим не рекомендуется использовать в промышленной эксплуатации.

1.3. Гибридный режим

Гибридный режим позволяет настроить работу МЭ как в режиме обратного прокси, так и в пассивном режиме для различных сетевых интерфейсов. Таким образом создаются отдельные контуры защиты для различных сервисов, что позволяет использовать один iBatyr WAF для сервисов с разным уровнем предоставления услуг. Схема работы iBatyr WAF в гибридном режиме представлена на схеме 3.

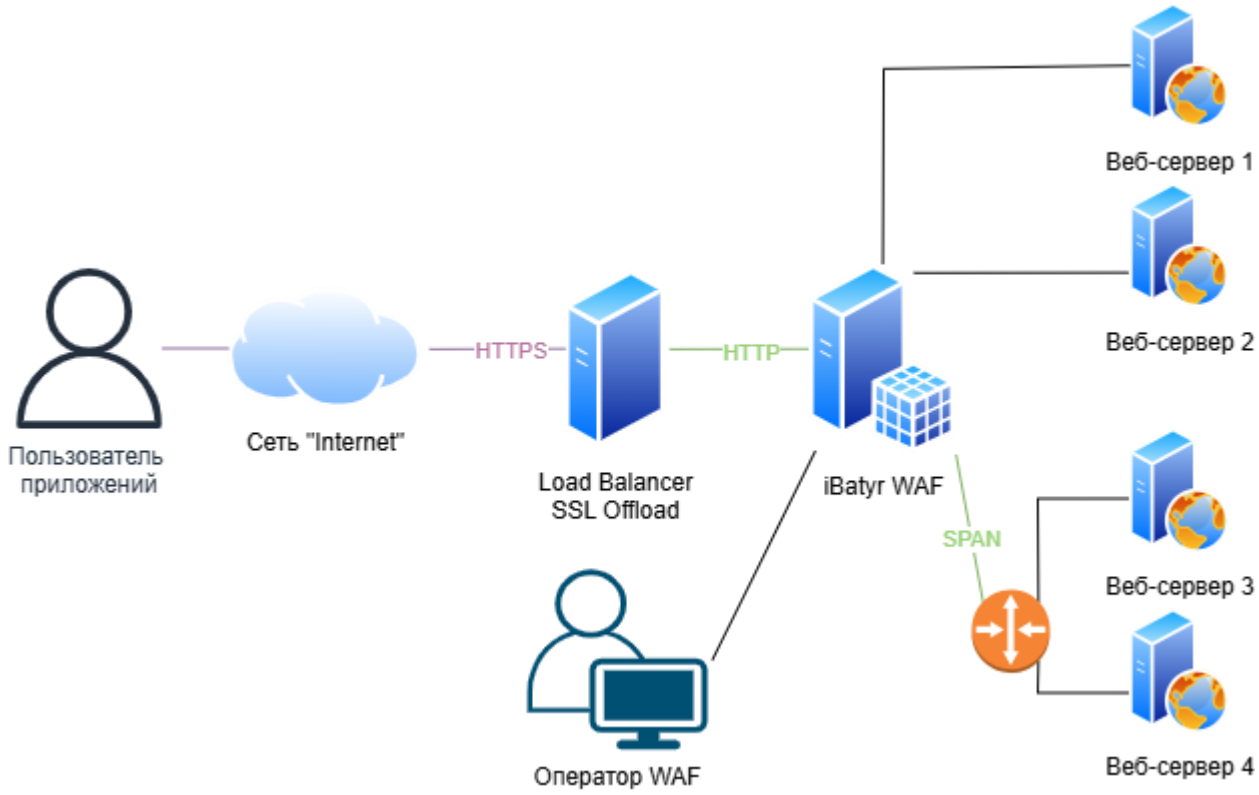


Схема 3 – Схема работы iBatyr WAF в гибридном режиме

Описание процесса установки iBatyr WAF в гибридном режиме дано в разделе 4.4.

Данный режим не рекомендуется использовать в процессе промышленной эксплуатации.

1.4. Распределенная конфигурация

Для балансировки нагрузки на серверы возможна установка iBatyr WAF в распределенной конфигурации (см. схема 4). Помимо этого, распределение позволяет объединить работу различных контуров под одной базой и системой управления. Еще одним преимуществом распределения является выделение узлов анализа под тестовые приложения и объединение их с продуктовыми приложениями под одним защищаемым ресурсом, что позволяет минимизировать временные затраты на дообучение профиля приложения WAF.

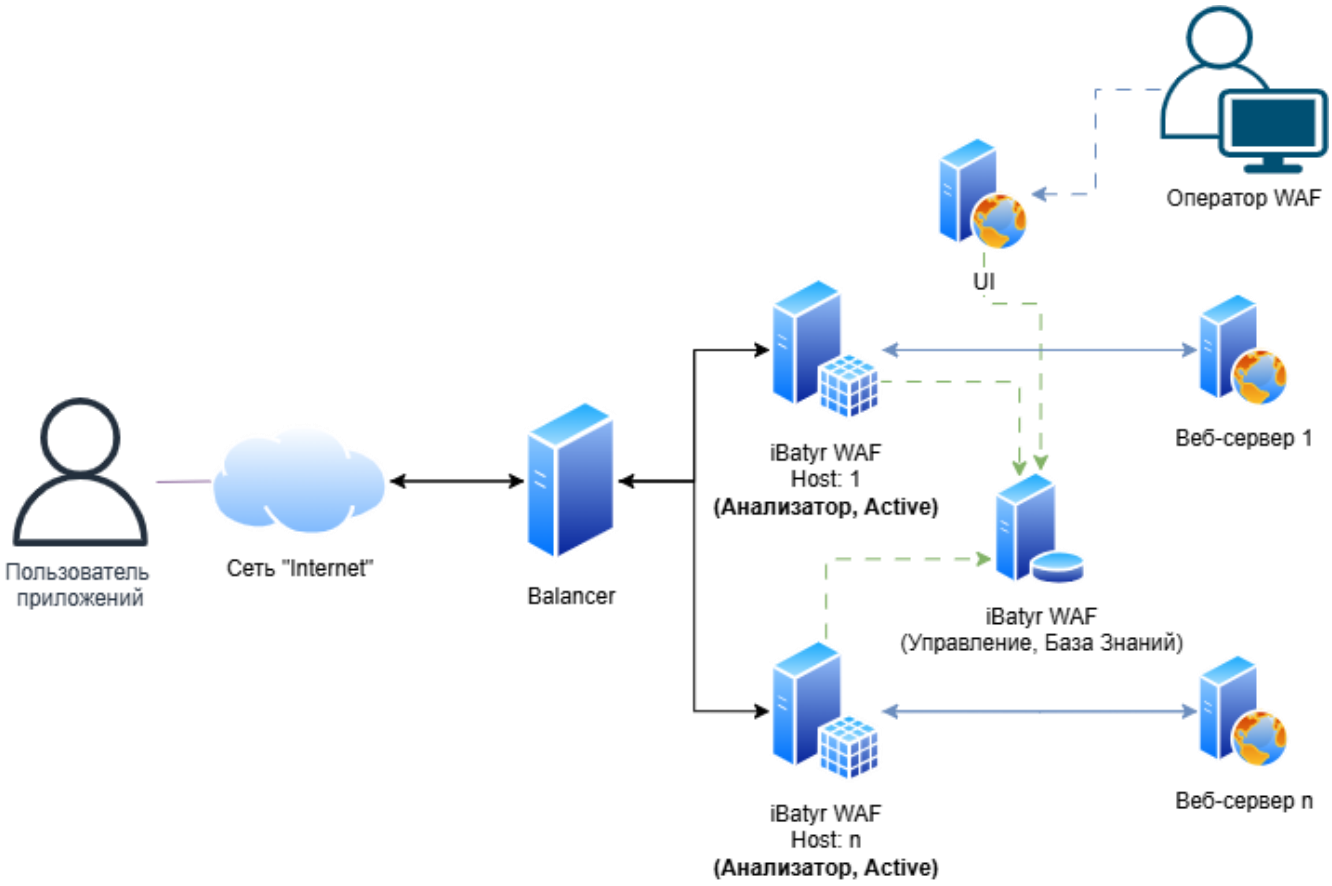


Схема 4 – Распределенная конфигурация

Распределенную конфигурацию можно настроить при инсталляции iBatyr WAF (см. раздел 4.5).

Распределенная конфигурация позволяет вынести на отдельные узлы модули анализа, БД и управления (веб-интерфейс), которые будут связаны между собой.

К службам iBatyr WAF относятся:

- ibatyr-analyzer, ibatyr-nginx (ibatyr-suricata в случае пассивного режима) и ibatyr-redis@0 к узлу Анализа;
- Postgres SQL, mongod, ibatyr-celerybeat, ibatyr-celery и ibatyr-redis@mgmt к узлу БД (База знаний);
- ibatyr-dashboard, nginx к узлу Управления (UI).

1.5. Распределенная конфигурация с кластером узлов Управления

Для балансировки нагрузки на серверы возможна распределенная конфигурация с кластером узлов Управления (см. схема 5). При такой схеме можно выделить 2 разных типа узлов WAF: узел Управления и узел Анализатора. Два узла управления могут объединяться в отказоустойчивый кластер (active-passive). Все узлы анализа – активные. Для их работы требуется балансировщик нагрузки HTTP-трафика (Balancer). Специальных требований к балансировщикам не предъявляется. Рекомендуется использовать балансировщик в режиме Sticky Session, чтобы HTTP-запросы в рамках одного соединения обрабатывались каким-то одним Анализатором. Это требование является рекомендуемым, но необязательным.

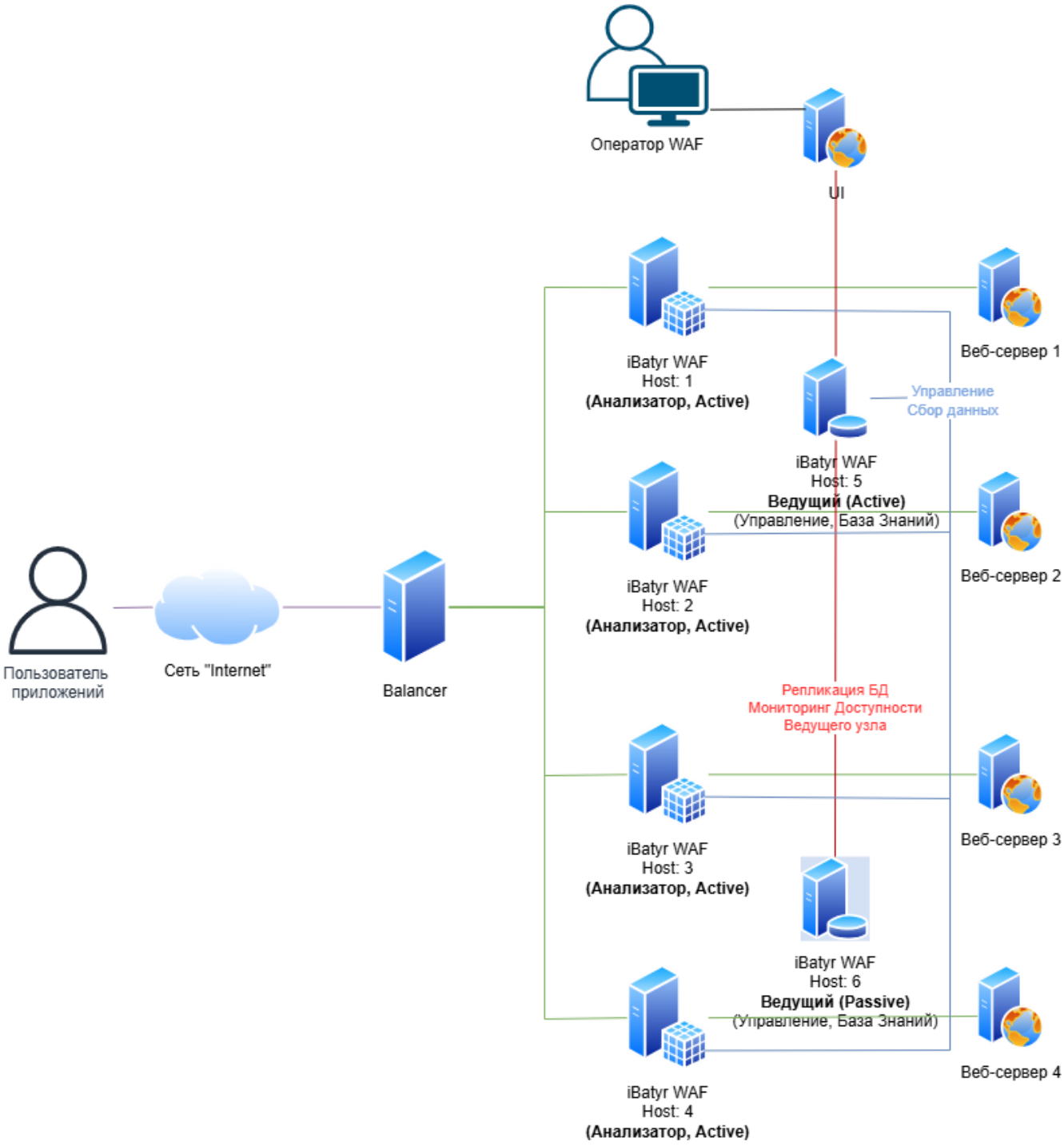


Схема 5 – Распределенная конфигурация с кластером узлов Управления

Описание процесса установки iBatyr WAF в распределенной конфигурации с кластером узлов Управления дано в разделе 4.7.

1.6. Отказоустойчивый кластер с репликацией на два узла

Два узла iBatyr WAF могут объединяться в отказоустойчивый кластер (active-passive). Каждый узел кластера содержит все необходимые для работы WAF сервисы.

При отказоустойчивой схеме весь трафик проходит через общий адрес кластера. Его еще называют – Virtual IP или VIP.

В такой конфигурации трафик обрабатывается только одним, ведущим узлом кластера (его еще называют master или active). Второй (также ведомый, резервный, slave или passive) узел идентичен первому по аппаратным требованиям. На него непрерывно реплицируются все изменения с ведущего узла. Оба узла непрерывно контролируют доступность друг друга и состояние критичных служб на узлах. В случае какого-либо критичного для работы WAF сбоя на ведущем узле, резервный узел автоматически берет на себя роль мастер-узла. На его интерфейсе поднимается Virtual IP, и, таким образом, трафик защищаемых приложений перенаправляется на него.

Далее необходимо локализовать и устранить причину и последствия сбоя на первом узле и вернуть его в штатный режим функционирования.

Необходимым требованием для работы кластера является L2-связность между узлами кластера. Или, иными словами, узлы должны находиться в одной подсети.

Процесс установки описан в разделе 4.6.

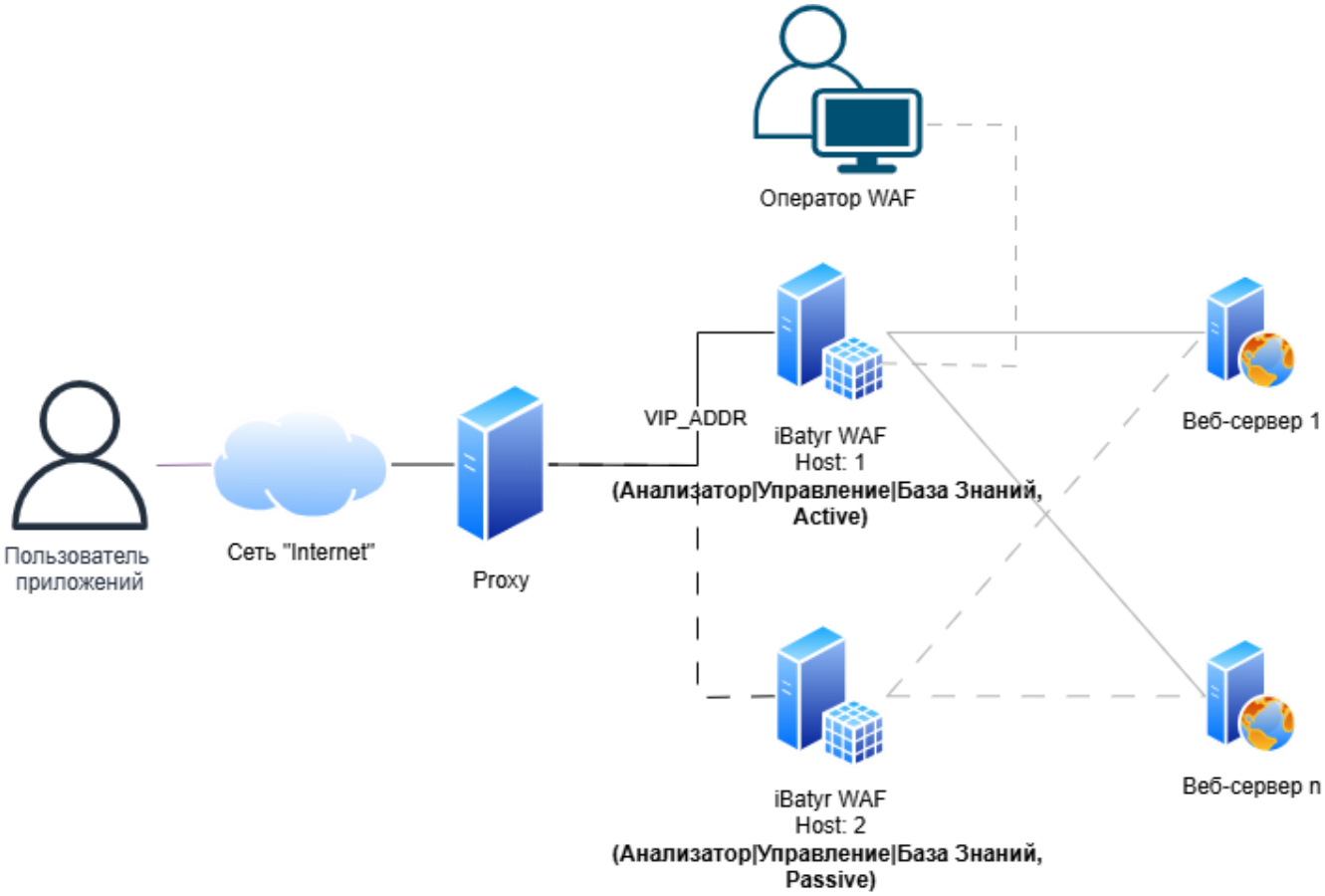


Схема 6 – Схема отказоустойчивого кластера из двух узлов с репликацией.

2. Требования к среде установки

Установка iBatyr WAF осуществляется в соответствии с требованиями к оборудованию и программному обеспечению описанными в данном разделе.

2.1. Минимальные системные требования

Минимальные требования должны соответствовать приведенным в таблице 1.

Таблица 1 – Минимальные системные требования

Компонент	Минимальное требование
Объём оперативной памяти (RAM)	Не менее 8 ГБ
Процессор (CPU)	4-ядерный и более с архитектурой x86-64, тактовая частота не ниже 2.4 ГГц
Объём жёсткого диска (HDD)	Не менее 500 ГБ
Сетевые интерфейсы	Два сетевых интерфейса 1 ГБит/с Ethernet для пассивного режима. Один сетевой интерфейс 1 ГБит/с Ethernet для активного режима

Примечание: Данная конфигурация может обрабатывать до 100 rps.

В качестве системного программного обеспечения поддерживаются следующие операционные системы (ОС): Ubuntu 22.04 Server 64-bit, Debian 12 64-bit, Astra Linux 1.7 Smolensk.

Установка может осуществляться на физические сервера и виртуальные машины.

Со списком поддерживаемых гипервизоров можно ознакомиться в требованиях на выбранную для установки WAF операционную систему.

2.2. Требования к разметке диска

В случае когда установка производится на узел Управления или в режиме stand-alone (все службы располагаются на одном узле) дисковое пространство должно быть размечено в следующих пропорциях: минимум 30 ГБ в «/», минимум 50 ГБ в «/var/log/», минимум 400 ГБ в «/var/lib/postgresql/».

В случае, когда установка производится на узел Анализа, дисковое пространство должно быть размечено в следующих пропорциях: минимум 30 ГБ в «/», минимум 50 ГБ в «/var/log/».

Пример развертывания ОС Ubuntu в соответствии с требованиями данного раздела приведён в приложении Б.

2.3. Антивирус

Установка антивируса на узлы с компонентами iBatyr WAF может привести к существенной деградации производительности и к отказам в работе модулей WAF. Устанавливать на узлы WAF антивирус крайне не рекомендуется и может приводить к отказу в сервисной поддержке.

3. Требования к инженеру

Рекомендуемая квалификация инженера:

- Знание основ веб-технологий, веб-протоколов и стандартов.
- Опыт администрирования UNIX-систем.
- Знание принципов работы сетей и сетевого оборудования, в том числе стека протоколов TCP/IP, семейства веб-протоколов и стандартов.
- Уметь проводить работы по настройке сети в среде Linux.
- Знать команды оболочки Bash на уровне администратора.
- Знать принципы работы в виртуальных средах и уметь ими оперировать.
- Разбираться в сервисе nginx, уметь создавать/редактировать конфигурационные файлы.
- Знать принципы работы Docker, PostgreSQL, MongoDB.
- Желательно иметь опыт работы с пограничными файрволами и маршрутизаторами.

4. Установка для различных режимов

4.1. Общий порядок установки

iBatyr WAF поставляется в виде дистрибутивов двух видов: вместе с операционной системой Ubuntu в формате ISO и в виде TGZ-архивов, которые можно разворачивать на совместимую ОС.

Общий порядок установки WAF или его компонентов на узел следующий:

1. Устанавливается поддерживаемая ОС из дистрибутива, который поставляется вендором данной ОС (пример данного этапа описан в приложении Б), либо дистрибутива WAF в формате ISO с соблюдением требований, описанных выше.
2. Далее при необходимости производится настройка установленных компонентов и их интеграция между собой согласно инструкциям, приведенным в следующих разделах.

4.2. Обратный прокси

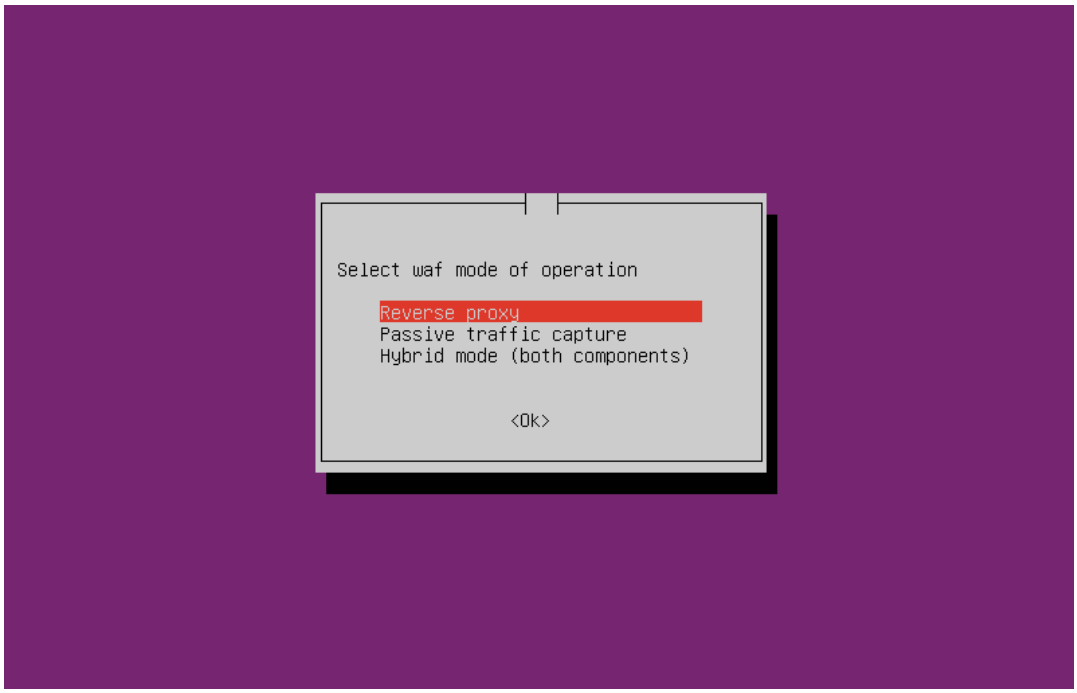
Установка осуществляется из образа `ibatyr-ubuntu-2X.04-v2.XX.0.iso` (ОС+WAF) на виртуальный или физический сервер. Сервера, на которые планируется устанавливать iBatyr WAF, должны отвечать минимальным системным требованиям из раздела 2.1.

Для начала нужно установить ОС (пример установки ОС приведён в приложении Б). После установки и аутентификации в ОС автоматически запустится установка WAF. Далее необходимо выполнить следующие действия:

1. В появившемся окне выбрать установку на один узел (см. снимок экрана 1).

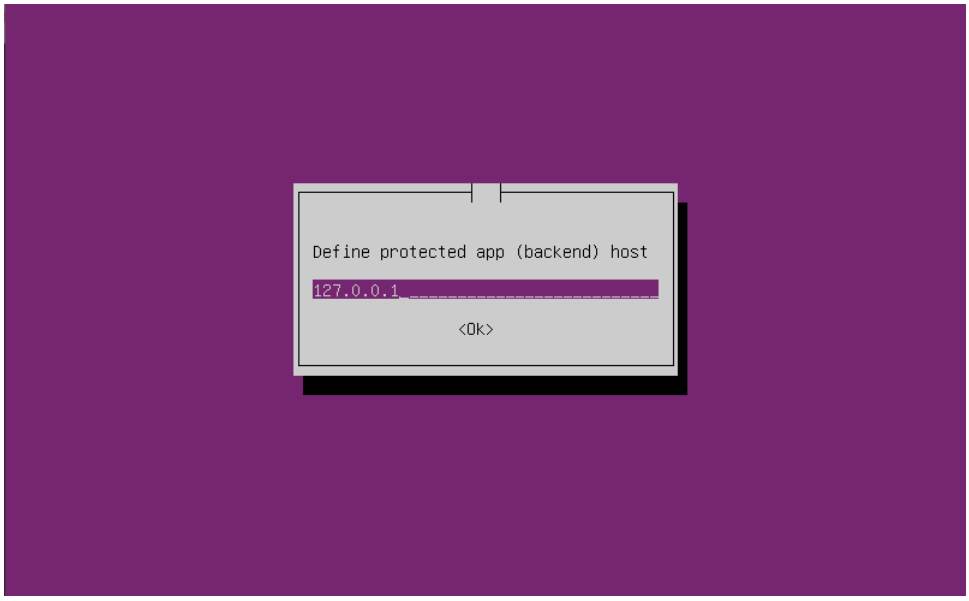
Снимок экрана 1 – Окно выбора установки на один узел

2. Затем в окне выбора режима установки выбрать пункт `Reverse proxy` (см. снимок экрана 2).

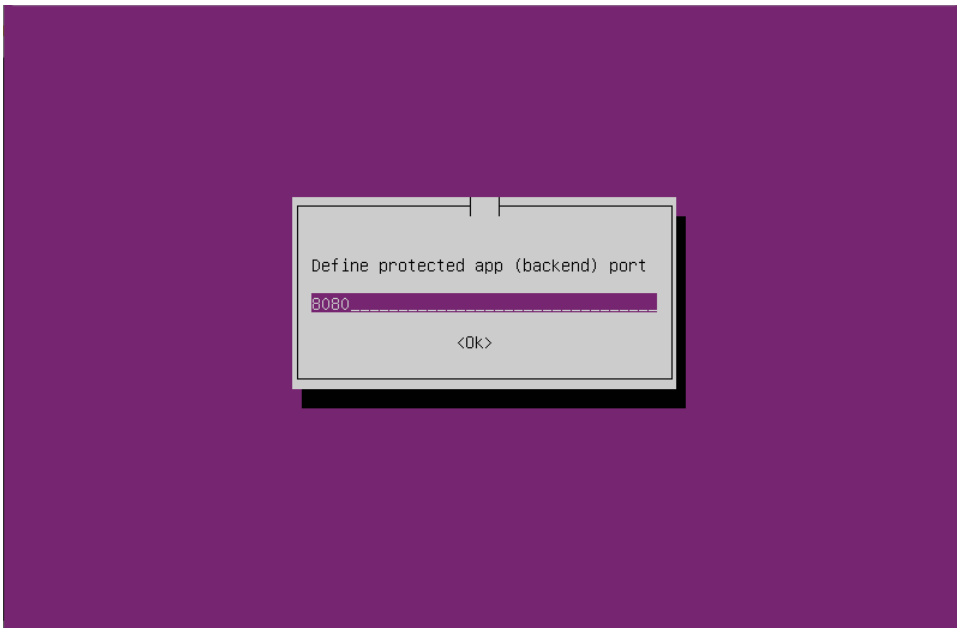


Снимок экрана 2 – Выбор режима установки

3. После, в соответствующих окнах, указать IP-адрес (см. снимок экрана 3) и порт защищаемого приложения (см. снимок экрана 4). Введенные на данном этапе значения можно будет изменить после установки.



Снимок экрана 3 –IP-адреса защищаемого приложения



Снимок экрана 4 – Порт защищаемого приложения

4. Вывести для проверки обзорную информацию по результатам установки с помощью команды (см. снимок экрана 5):

```
sudo grep -A 1 RECAP /var/tmp/install.log
```

```
waf@ibatyr :~$ sudo grep -A 1 RECAP /var/tmp/install.log
2021-12-01 07:47:25,904 p=2754 u=waf n=ansible | PLAY RECAP *********************************************************
*****************************
2021-12-01 07:47:25,906 p=2754 u=waf n=ansible | localhost                  : ok=174  changed=107  u
nreachable=0    failed=0    skipped=212  rescued=0    ignored=0
```

Снимок экрана 5 – Команда для проверки

Индикатором успешной установки будет являться отсутствие ошибок установки (failed=0).

5. Затем для проверки соответствия портов сервисам необходимо воспользоваться командой (см. снимок экрана 6):

```
ss -ntl
```

```
waf@ibatyr :~$ ss -ntl
State      Recv-Q     Send-Q         Local Address:Port          Peer Address:Port     Process
LISTEN     0          511                  0.0.0.0:8443               0.0.0.0:*
LISTEN     0          2048             127.0.0.1:5088               0.0.0.0:*
LISTEN     0          4096             127.0.0.1:27017              0.0.0.0:*
LISTEN     0          100              127.0.0.1:6666               0.0.0.0:*
LISTEN     0          511              127.0.0.1:6378               0.0.0.0:*
LISTEN     0          511              127.0.0.1:6379               0.0.0.0:*
LISTEN     0          511                  0.0.0.0:80               0.0.0.0:*
LISTEN     0          4096          127.0.0.53%lo:53                0.0.0.0:*
LISTEN     0          128                  0.0.0.0:22               0.0.0.0:*
LISTEN     0          224              127.0.0.1:5432               0.0.0.0:*
LISTEN     0          511                     [::]:80                  [::]:*
LISTEN     0          128                     [::]:22                  [::]:*
```

Снимок экрана 6 – Соответствие проверки портов

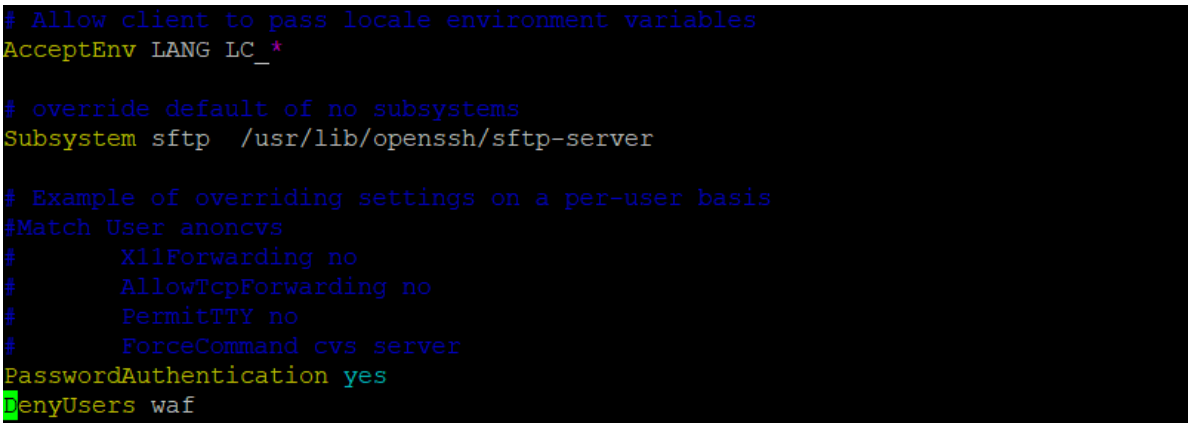
Порты для сервисов должны соответствовать:

- nginx: port 8443;

- ibatyr-dashboard: port 5088;
- mongod: port 27017;
- ibatyr-analyzer@0: port 6666;
- ibatyr-redis@mgmt: port 6378;
- ibatyr-redis@0: port 6379;
- ibatyr-nginx: порт, который вы указали при установке WAF;
- sshd: port 22;
- postgresql: port 5432.

Для безопасности во время установки был запрещен SSH доступ пользователю `waf`. Если необходимо разрешить такой доступ, то в конфигурационном файле `/etc/sshd/sshd_config` следует удалить последнюю строчку `DenyUsers waf` и перезапустить сервис с помощью команды (см. снимок экрана 7):

```
sudo systemctl restart sshd
```



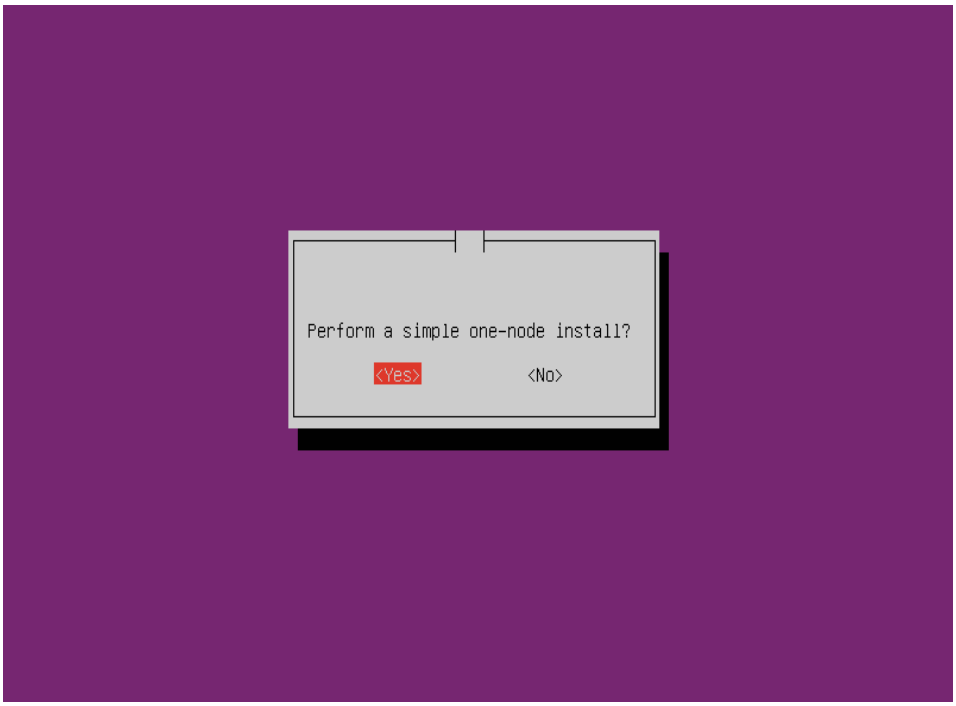
Снимок экрана 7 – Разрешение SSH

4.3. Пассивный режим

Установка осуществляется из образа `ibatyr-ubuntu-2X.04-v2.XX.0.iso` (ОС+WAF) на виртуальный или физический сервер. Сервера, на которые планируется устанавливать iBatyr WAF, должны отвечать минимальным системным требованиям из раздела 2.1.

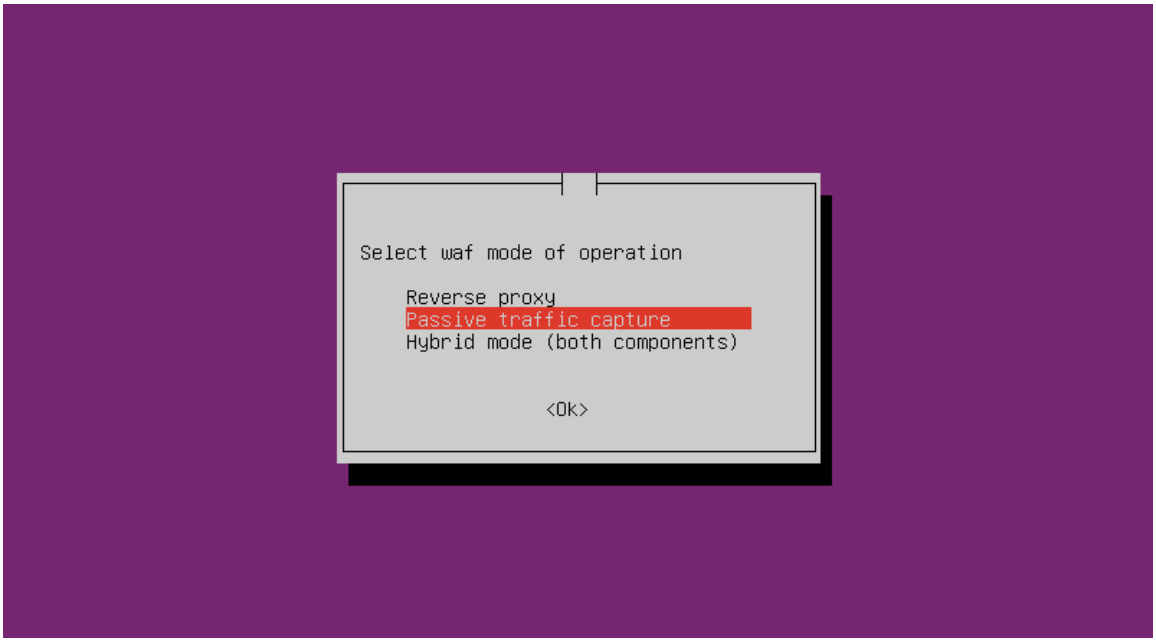
Для начала нужно установить ОС (пример установки ОС приведён в приложении Б). После установки и аутентификации в ОС автоматически запустится установка WAF. Далее необходимо выполнить следующие действия:

1. В появившемся окне выбрать установку на один узел (см. снимок экрана 8).



Снимок экрана 8 – Выбор установки на один узел

2. Затем в окне выбора режима установки выбрать пункт `Passive traffic capture` (см. снимок экрана 9).



Снимок экрана 9 – Выбор режима установки

3. Убедиться в корректности установки (см. раздел 5.1 и снимок экрана 10).

```
waf@ibatyr:~$ sudo grep -A 1 RECAP /var/tmp/install.log
2021-12-01 07:47:25,904 p=2754 u=waf n=ansible | PLAY RECAP *********************************************
**********************************
2021-12-01 07:47:25,906 p=2754 u=waf n=ansible | localhost                  : ok=174  changed=107  u
nreachable=0    failed=0    skipped=212  rescued=0    ignored=0
```

Снимок экрана 10 – Проверка успешности установки

Для безопасности во время установки был запрещен SSH доступ пользователю `waf`. Если необходимо разрешить такой доступ, то в конфигурационном файле `/etc/sshd/sshd_config` следует удалить последнюю строчку `DenyUsers waf` и перезапустить сервис с помощью команды (см. снимок экрана 11):

```
sudo systemctl restart sshd
```

```
# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp  /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
PasswordAuthentication yes
DenyUsers waf
```

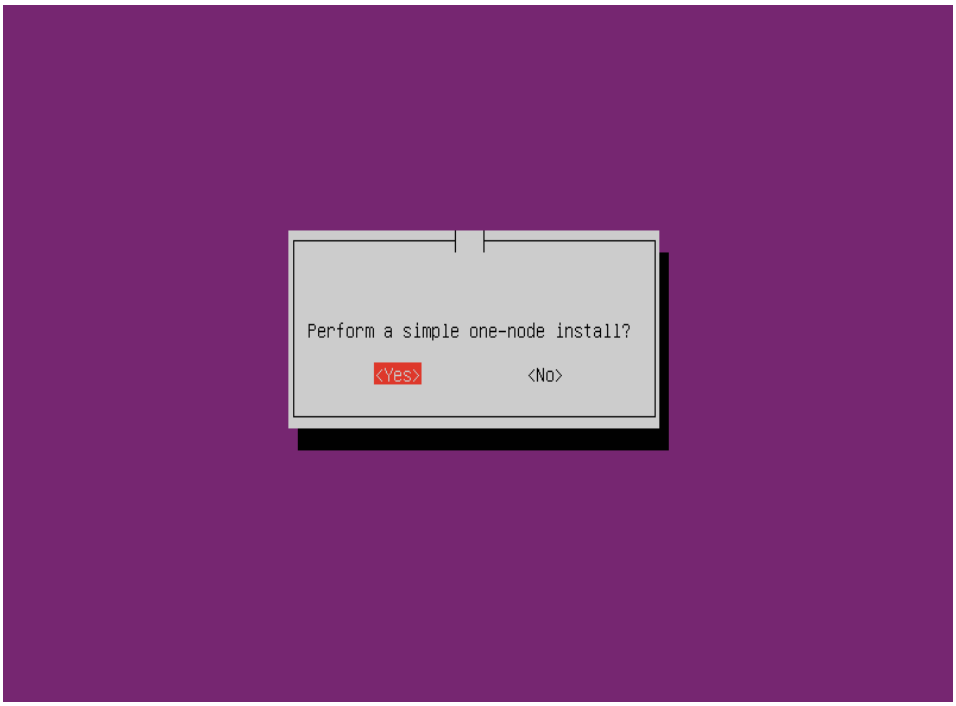
Снимок экрана 11 – Разрешение SSH

4.4. Гибридный режим

Установка осуществляется из образа `ibatyr-ubuntu-2X.04-v2.XX.0.iso` (ОС+WAF) на виртуальный или физический сервер. Сервера, на которые планируется устанавливать iBatyr WAF, должны отвечать минимальным системным требованиям из раздела 2.1.

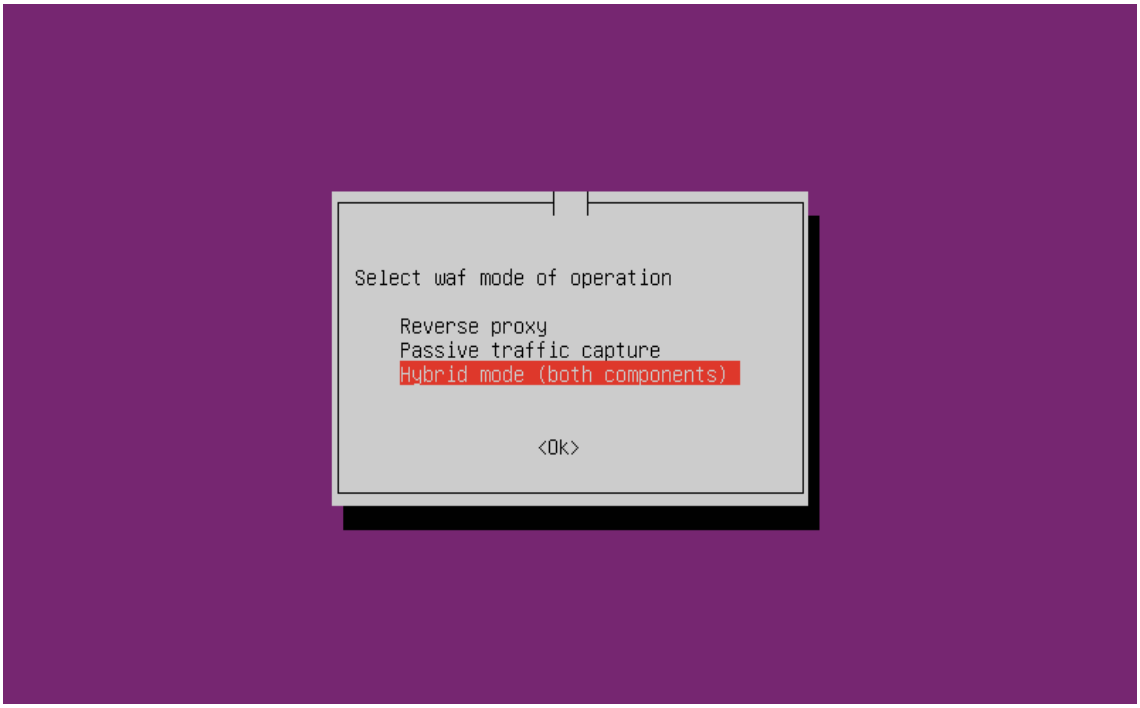
Для начала нужно установить ОС (пример установки ОС приведён в приложении Б). После установки и аутентификации в ОС автоматически запустится установка WAF. Далее необходимо выполнить следующие действия:

1. В появившемся окне выбрать установку на один узел (см. снимок экрана 12).



Снимок экрана 12 – Выбор установки на один узел

2. Затем в окне выбора режима установки выбрать пункт `Hybrid mode (both components)` (см. снимок экрана 13).



Снимок экрана 13 – Выбор режима установки

3. После, в соответствующих окнах, указать IP-адрес (см. снимок экрана 14) и порт защищаемого приложения (см. снимок экрана 15). Введенные на данном этапе значения можно будет изменить после установки.

Снимок экрана 14 – IP-адреса защищаемого приложения

Снимок экрана 15 – Порт защищаемого приложения

4. Убедиться в корректности установки (см. раздел 5.1 и снимок экрана 16).

```
waf@ibatyr:~$ sudo grep -A 1 RECAP /var/tmp/install.log
2021-12-01 07:47:25,904 p=2754 u=waf n=ansible | PLAY RECAP *********************************************************
**********************************
2021-12-01 07:47:25,906 p=2754 u=waf n=ansible | localhost                  : ok=174  changed=107  u
nreachable=0    failed=0    skipped=212  rescued=0    ignored=0
```

Снимок экрана 16 – Проверка корректности установки

Для безопасности во время установки был запрещен SSH доступ пользователю `waf`. Если необходимо разрешить такой доступ, то в конфигурационном файле `/etc/sshd/sshd_config` следует удалить последнюю строчку `DenyUsers waf` и перезапустить сервис с помощью команды (см. снимок экрана 17):

```
sudo systemctl restart sshd
```

```
# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp  /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
PasswordAuthentication yes
DenyUsers waf
```

Снимок экрана 17 – Разрешение SSH

4.5. Распределенная конфигурация

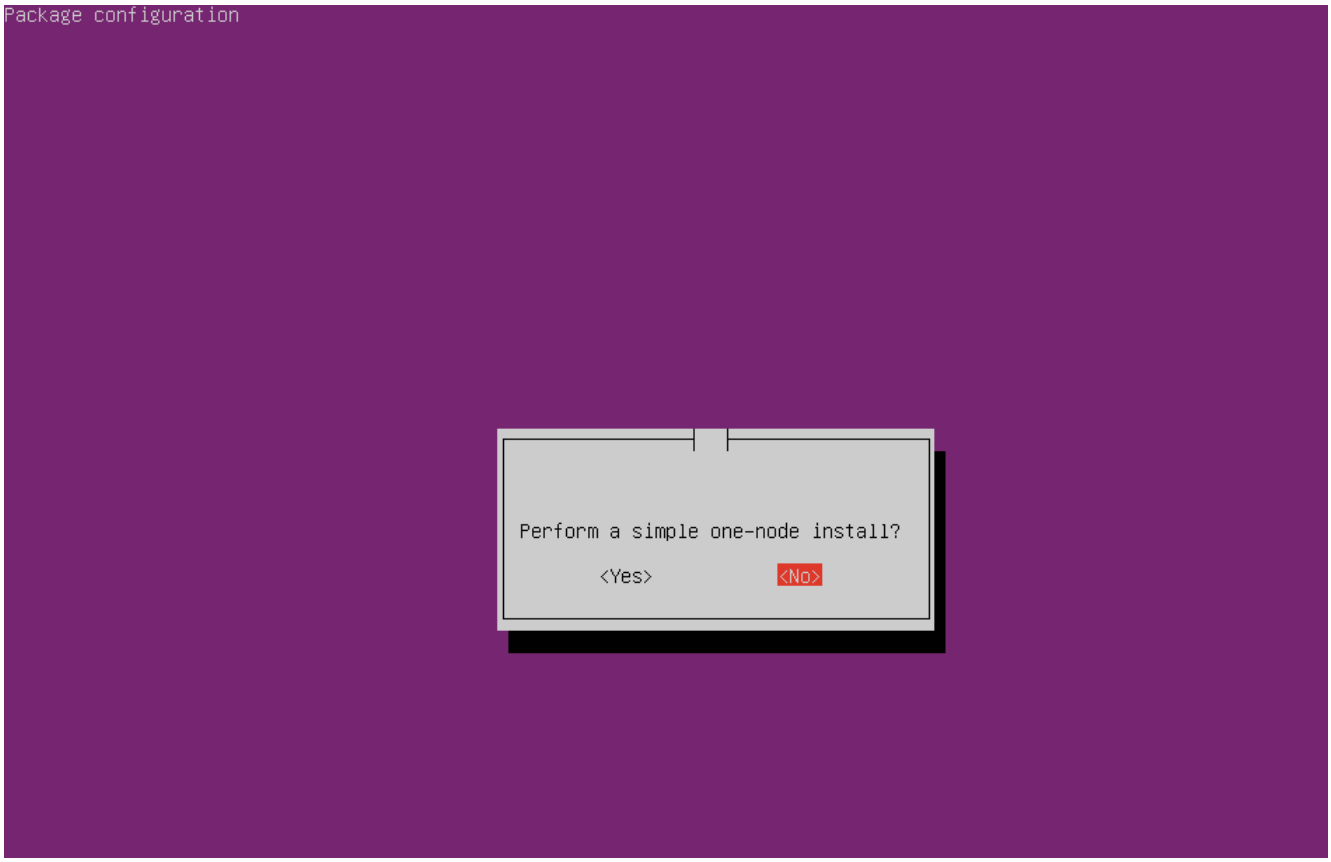
4.5.1. Узел БД и узел менеджмента

Зачастую модуль менеджмента (веб-интерфейс) не выносят на отдельный узел и оставляют вместе с узлом БД на одном узле.

Установка осуществляется из образа `ibatyr-ubuntu-2X.04-v2.XX.0.iso` (ОС+WAF) на виртуальный или физический сервер. Сервера, на которые планируется устанавливать iBatyr WAF, должны отвечать минимальным системным требованиям из раздела 2.1.

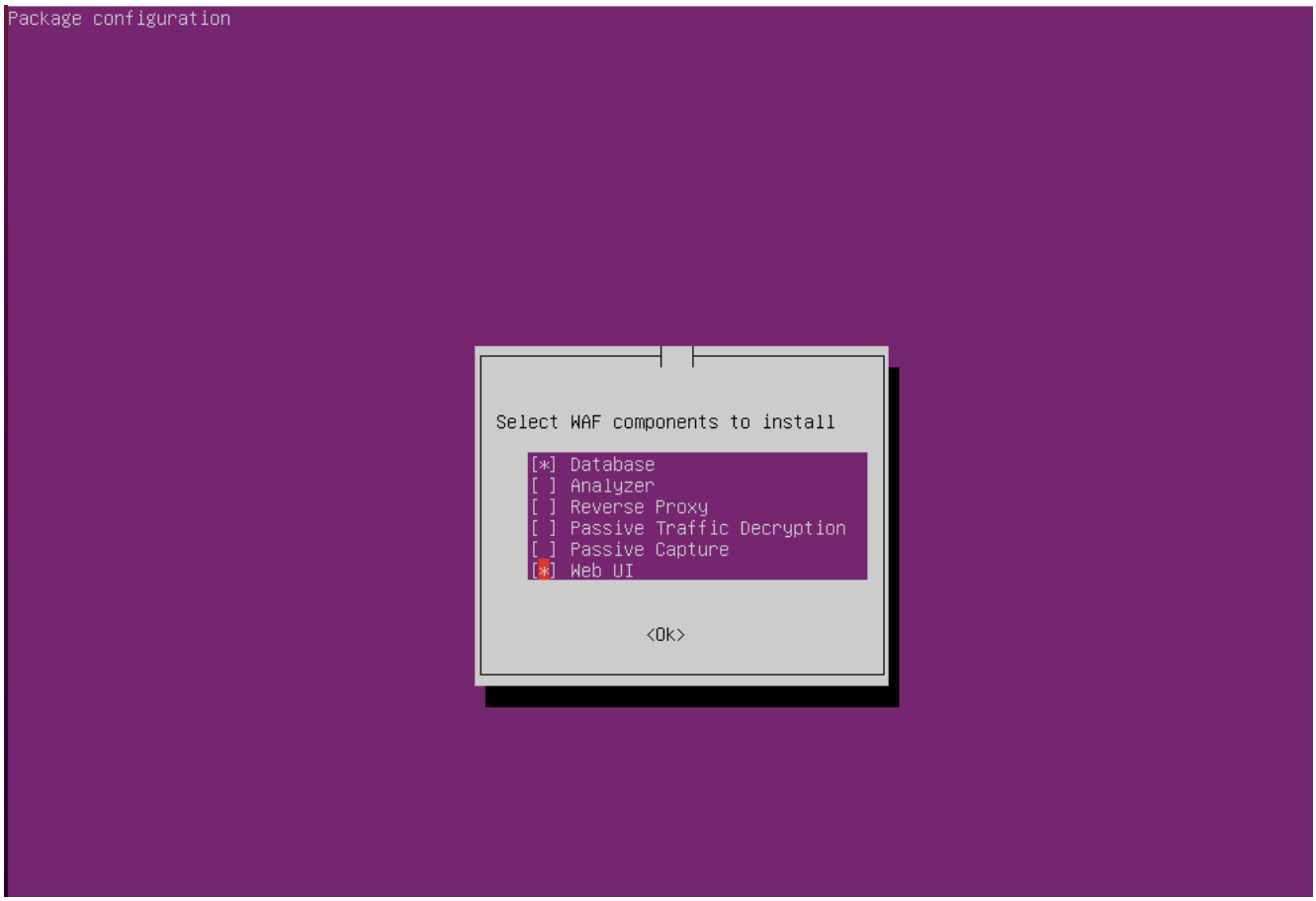
Для начала нужно установить ОС (пример установки ОС приведён в приложении Приложение Б. После установки и аутентификации в ОС автоматически запустится установка WAF. Далее необходимо произвести следующие действия:

1. Указать «No» и выбрать установку не на один узел (см. снимок экрана 18).



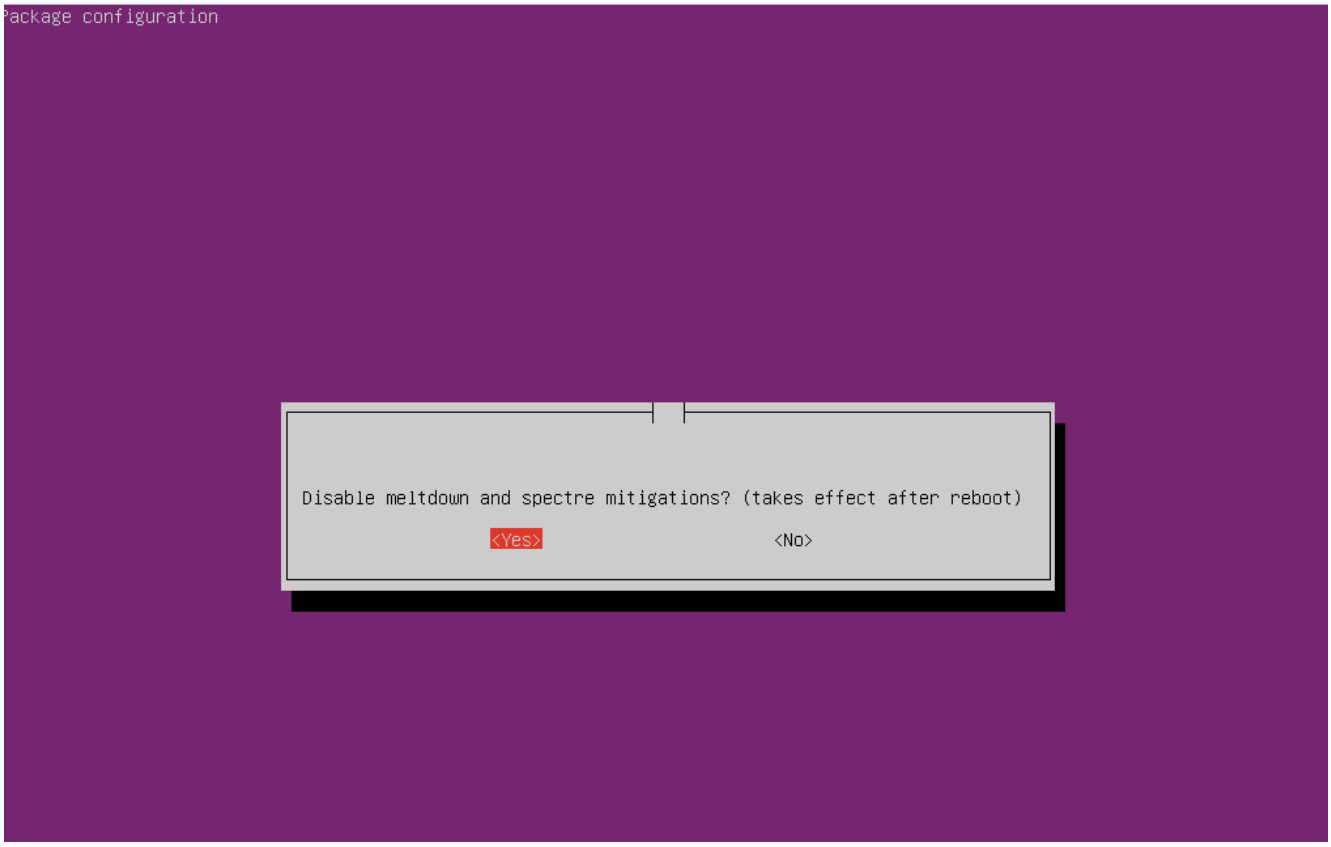
Снимок экрана 18 – Выбор установки

2. В открывшемся окне выбрать компоненты установки `Database` и `WebUI` (см. снимок экрана 19).



Снимок экрана 19 – Компоненты установки

3. В следующем появившемся окне рекомендуется отключить защиту от эксплуатации уязвимостей процессоров `meltdown` и `spectre`, т.к. это значительно снизит производительность узлов кластера и актуально только на многопользовательских облачных сервисах (см. снимок экрана 20).



Снимок экрана 20 – Компоненты защиты процессоров

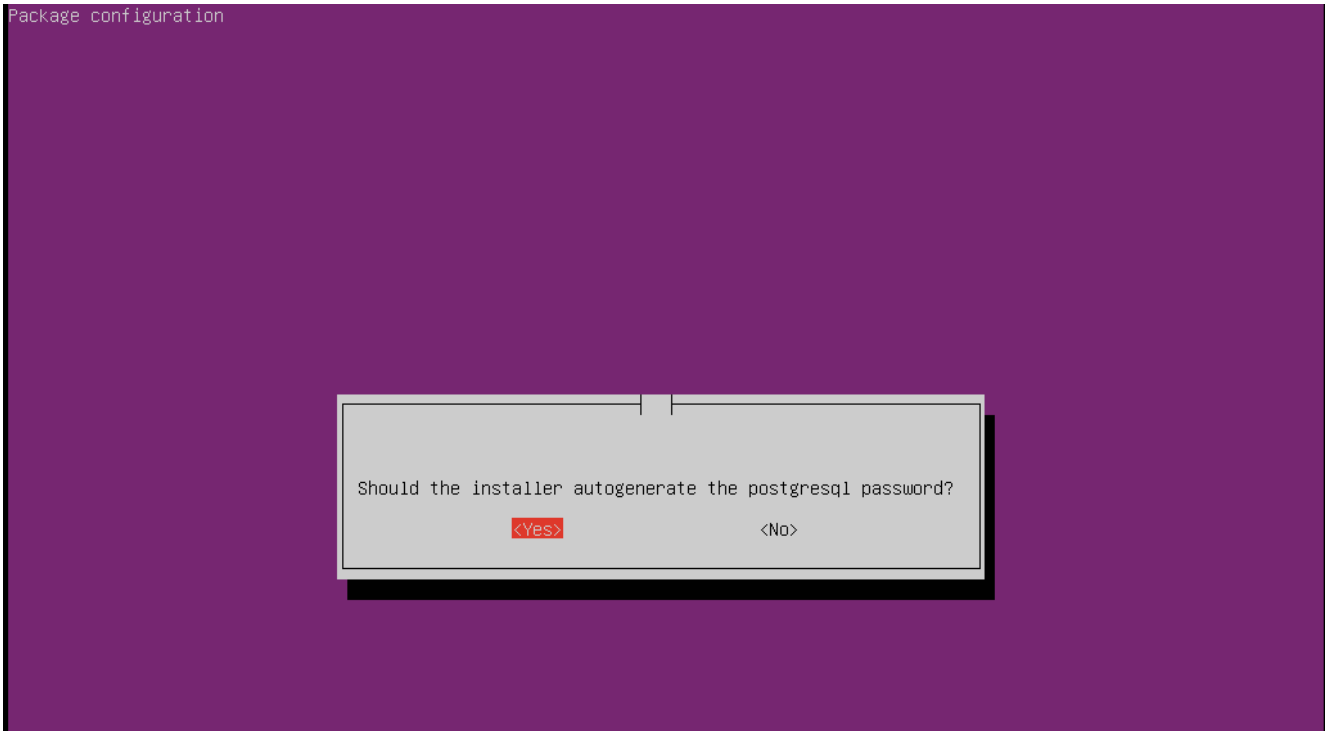
4. Далее следующем окне выбрать количество дней хранения записи о заблокированных транзакциях в БД PostgreSQL (см. снимок экрана 21).

Снимок экрана 21 – Время хранения записи о заблокированных транзакциях в БД PostgreSQL

5. В следующем окне выбрать количество дней хранения обычных транзакций в БД PostgreSQL (см. снимок экрана 22).

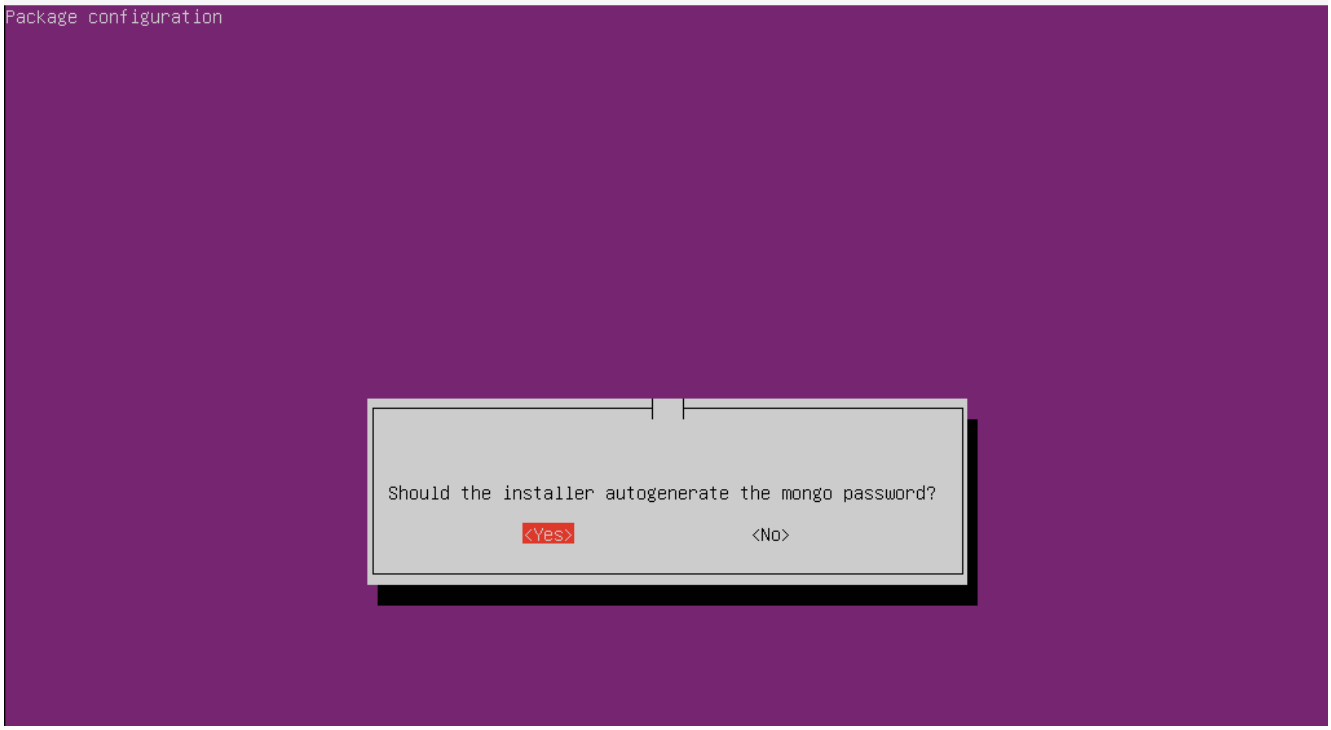
Снимок экрана 22 – Время хранения обычных транзакций в БД PostgreSQL

6. В следующем появившемся окне выбрать автоматическую генерацию пароля БД PostgreSQL (см. снимок экрана 23).



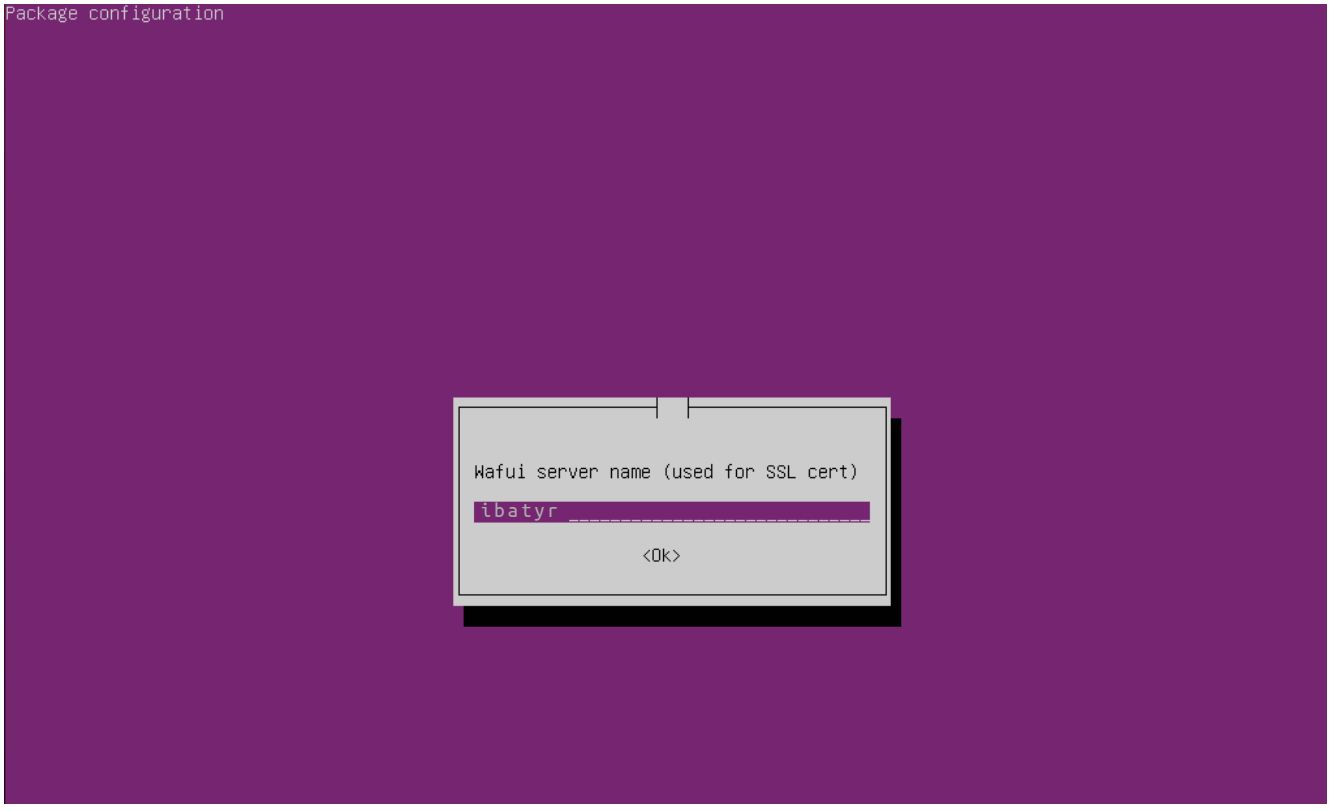
Снимок экрана 23 – Генерация пароля БД PostgreSQL

7. В следующем окне выбрать автоматическую генерацию пароля `mongo` (см. Снимок экрана 24).



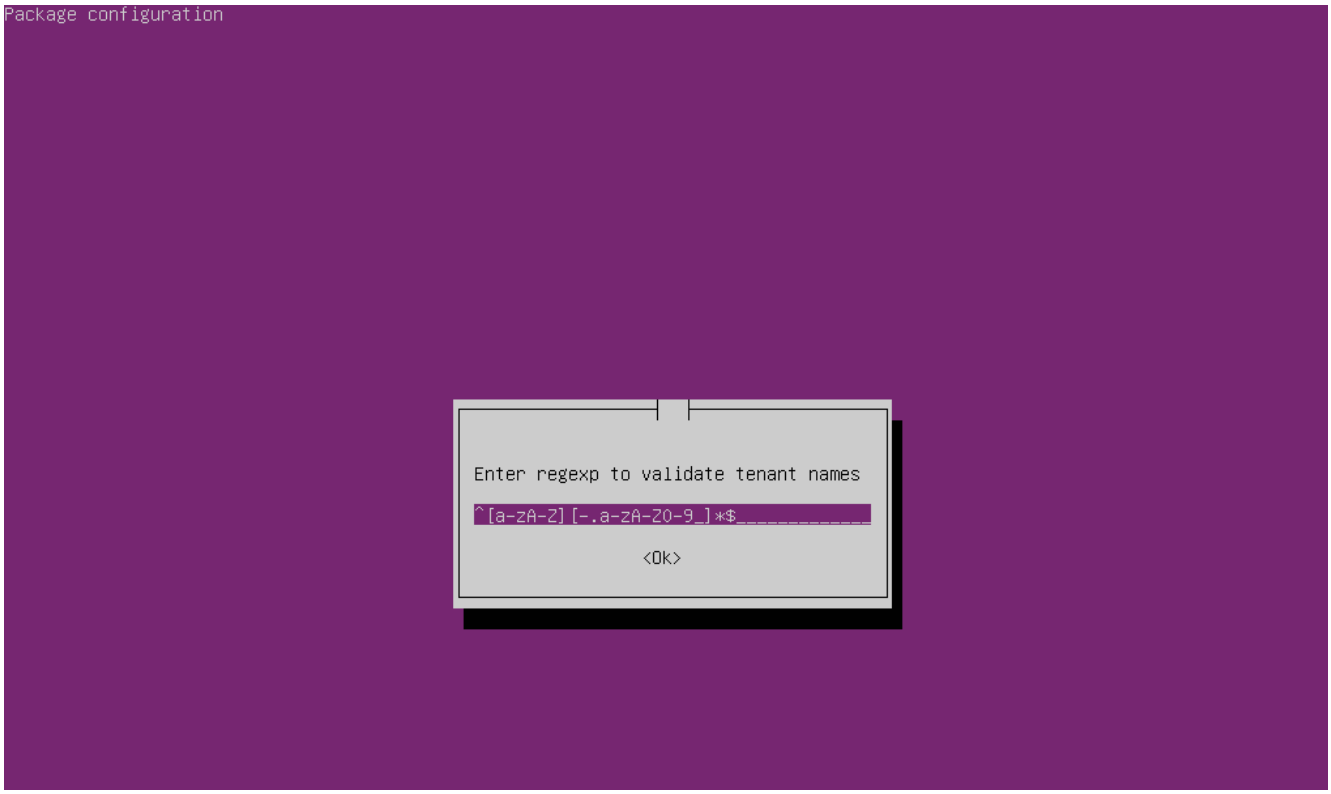
Снимок экрана 24. Генерация пароля `mongo`

8. В следующих двух появившихся окнах возможно дублирование запросов данных из пунктов 4 и 5 настоящего раздела.
9. Затем в следующем появившемся окне задать имя сервера `WebUI` (см. снимок экрана 25).



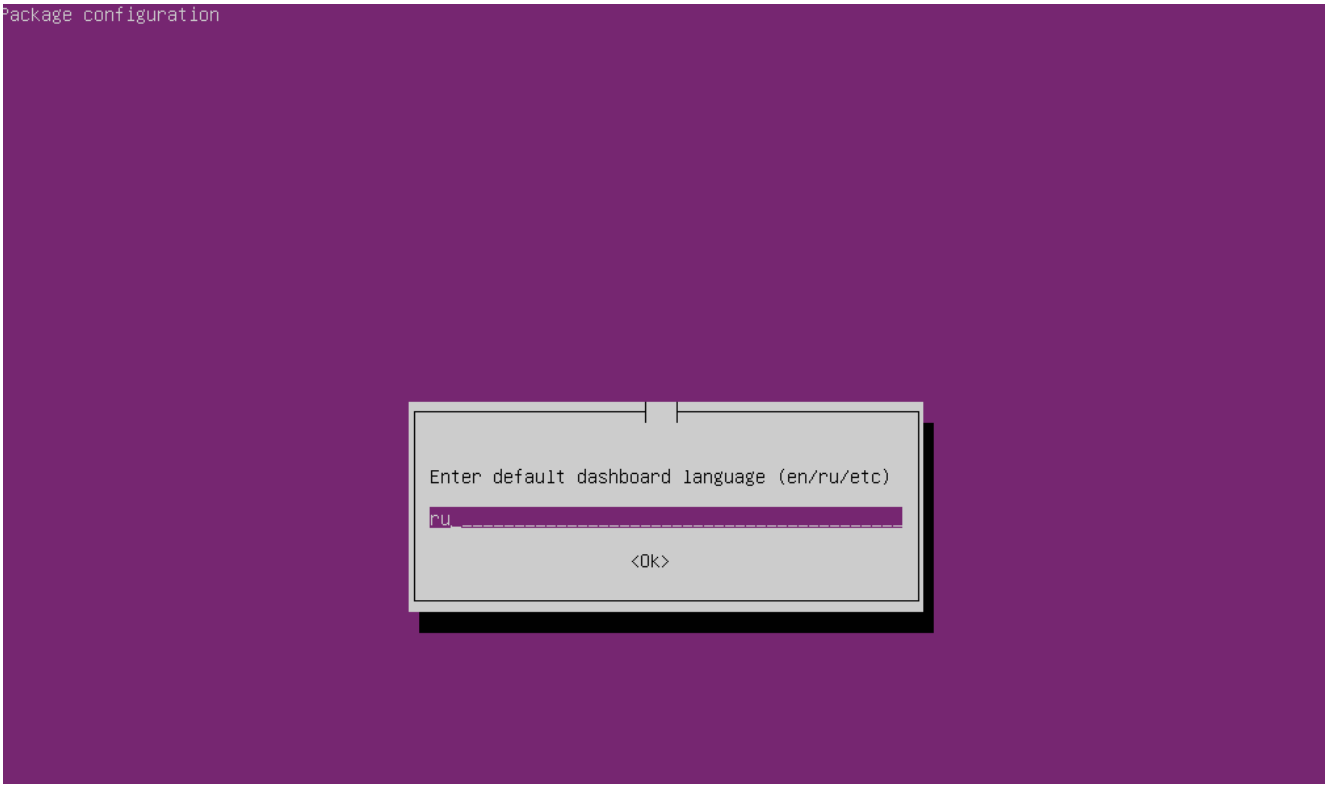
Снимок экрана 25 – Имя сервера `WebUI`

10. В следующем появившемся окне выбрать «Ok» (см. снимок экрана 26).



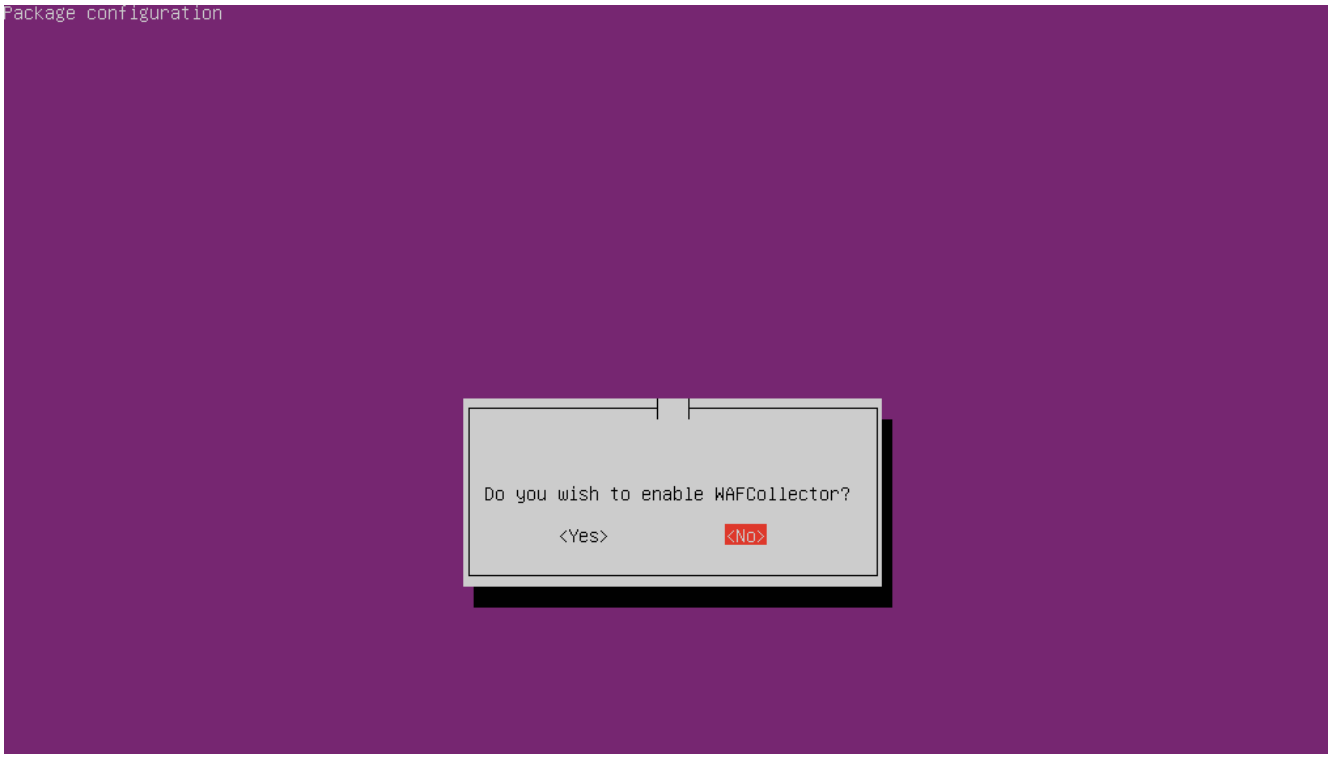
Снимок экрана 26 – Enter regexp to validate tenant names

11. В следующем появившемся окне выбрать язык дашборда по дефолту (см. Снимок экрана 27).



Снимок экрана 27 – Язык дашборда

12. Далее в появившемся окне выбрать «No» - не включать `WAFCollector` (см. снимок экрана 28).

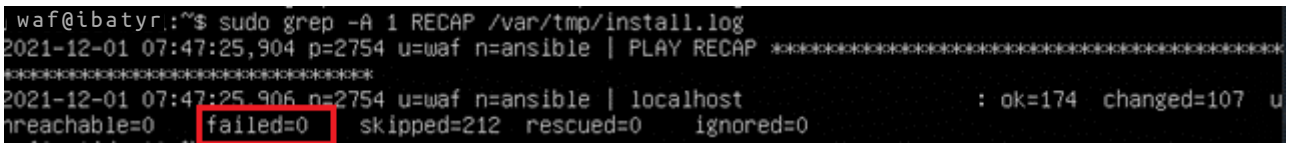


Снимок экрана 28 – `WAFCollector`

13. Затем вывести для проверки обзорную информацию по результатам установки с помощью команды:

```
sudo grep -A 1 RECAP /var/tmp/install.log
```

Индикатором успешной установки будет являться отсутствие ошибок установки «failed=0» (см. снимок экрана 29).



Снимок экрана 29 – Установка без ошибок

Для безопасности во время установки был запрещен SSH доступ пользователю `waf`. Если необходимо разрешить такой доступ, то в конфигурационном файле `/etc/sshd/sshd_config` следует удалить последнюю строчку `DenyUsers waf` и перезапустить сервис с помощью команды (см. снимок экрана 30):

```
sudo systemctl restart sshd
```

```
# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp  /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
PasswordAuthentication yes
DenyUsers waf
```

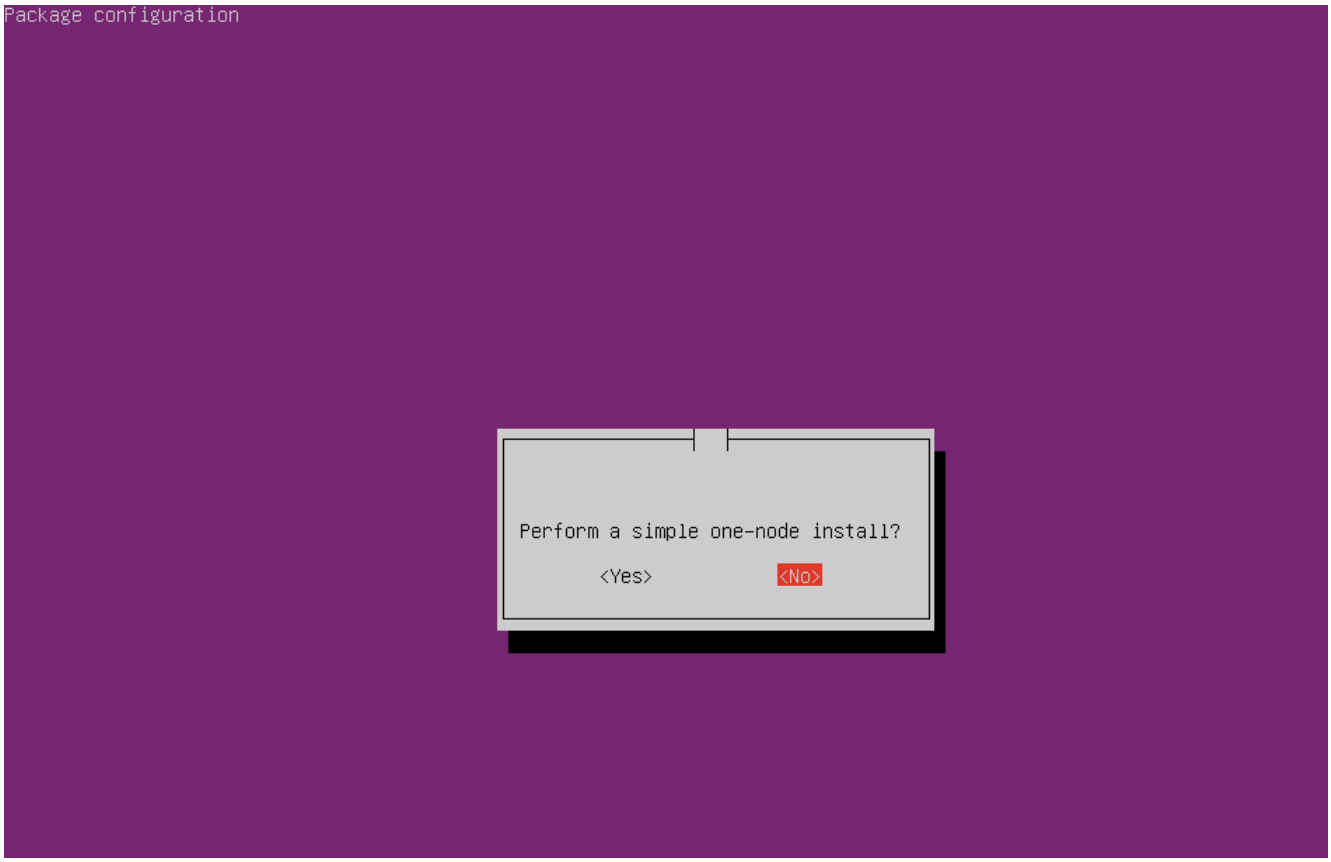
Снимок экрана 30 – Разрешение SSH

4.5.2. Узел веб-интерфейса

Установка осуществляется из образа ibatyr-ubuntu-2X.04-v2.XX.0.iso (ОС+WAF) на виртуальный или физический сервер. Сервера, на которые планируется устанавливать iBatyr WAF, должны отвечать минимальным системным требованиям из раздела 2.1.

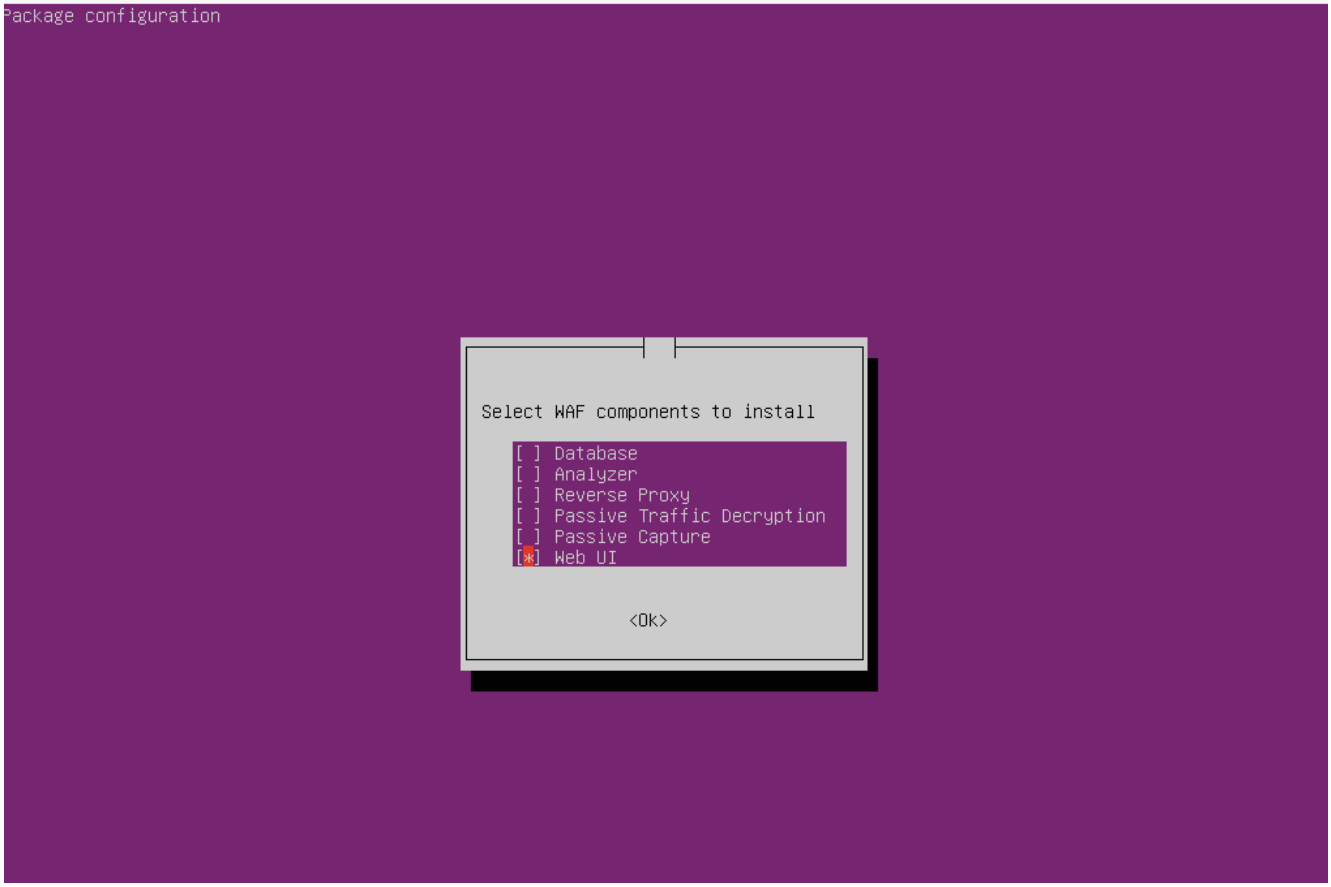
Для начала нужно установить ОС (пример установки ОС приведён в приложении Б). После установки и аутентификации в ОС автоматически запустится установка WAF. Далее необходимо произвести следующие действия:

1. Указать «No» и выбрать установку не на один узел (см. снимок экрана 31).



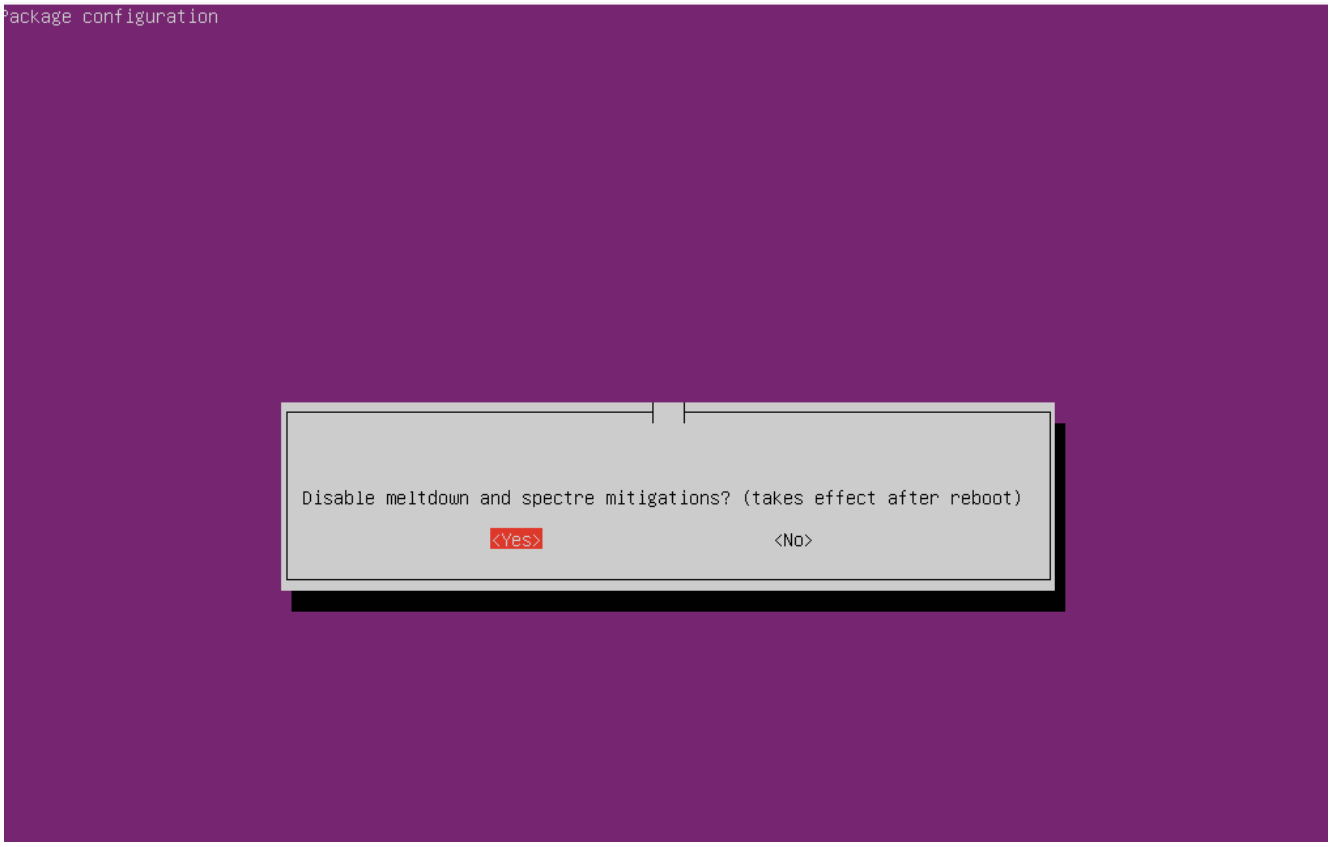
Снимок экрана 31 – Выбор установки

2. В следующем появившемся окне выбрать компонент установки `Web UI` (см. снимок экрана 32).



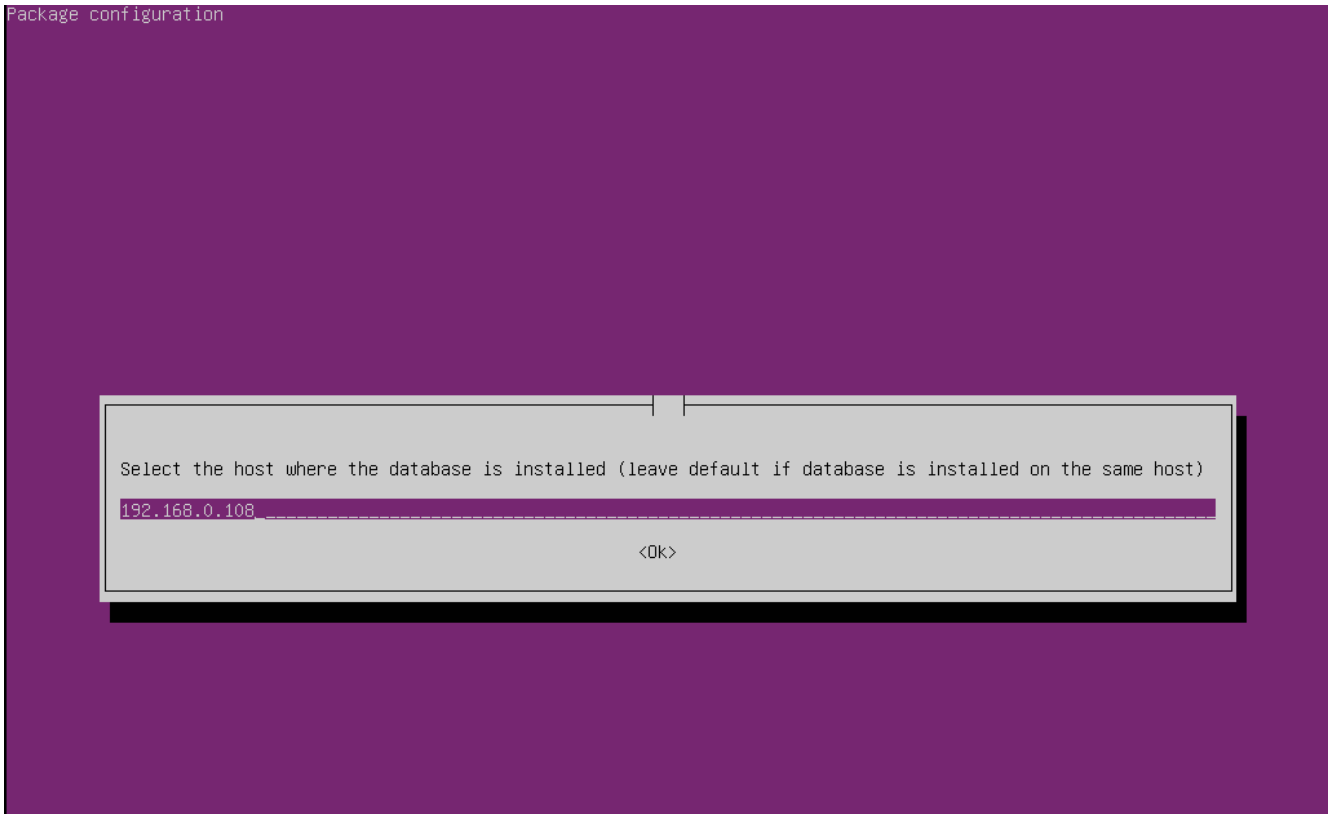
Снимок экрана 32 – Выбор компонентов установки

3. В следующем появившемся окне рекомендуется отключить защиту от эксплуатации уязвимостей процессоров `meltdown` и `spectre`, т.к. это значительно снизит производительность узлов кластера и актуально только на многопользовательских облачных сервисах (см. снимок экрана 33).



Снимок экрана 33 – Компоненты защиты процессоров

4. В следующем окне ввести IP-адрес сервера с базой данных PostgreSQL (см. снимок экрана 34).



Снимок экрана 34 – IP-адрес сервера с базой данных PostgreSQL

5. Далее в открывшемся окне ввести пароль подключения к PostgreSQL сервера (см. снимок экрана 35).

Снимок экрана 35 – Пароль PostgreSQL

6. В следующем появившемся окне подтвердить пароль подключения к PostgreSQL сервера (см. снимок экрана 36).

Снимок экрана 36 – Подтверждение пароля PostgreSQL

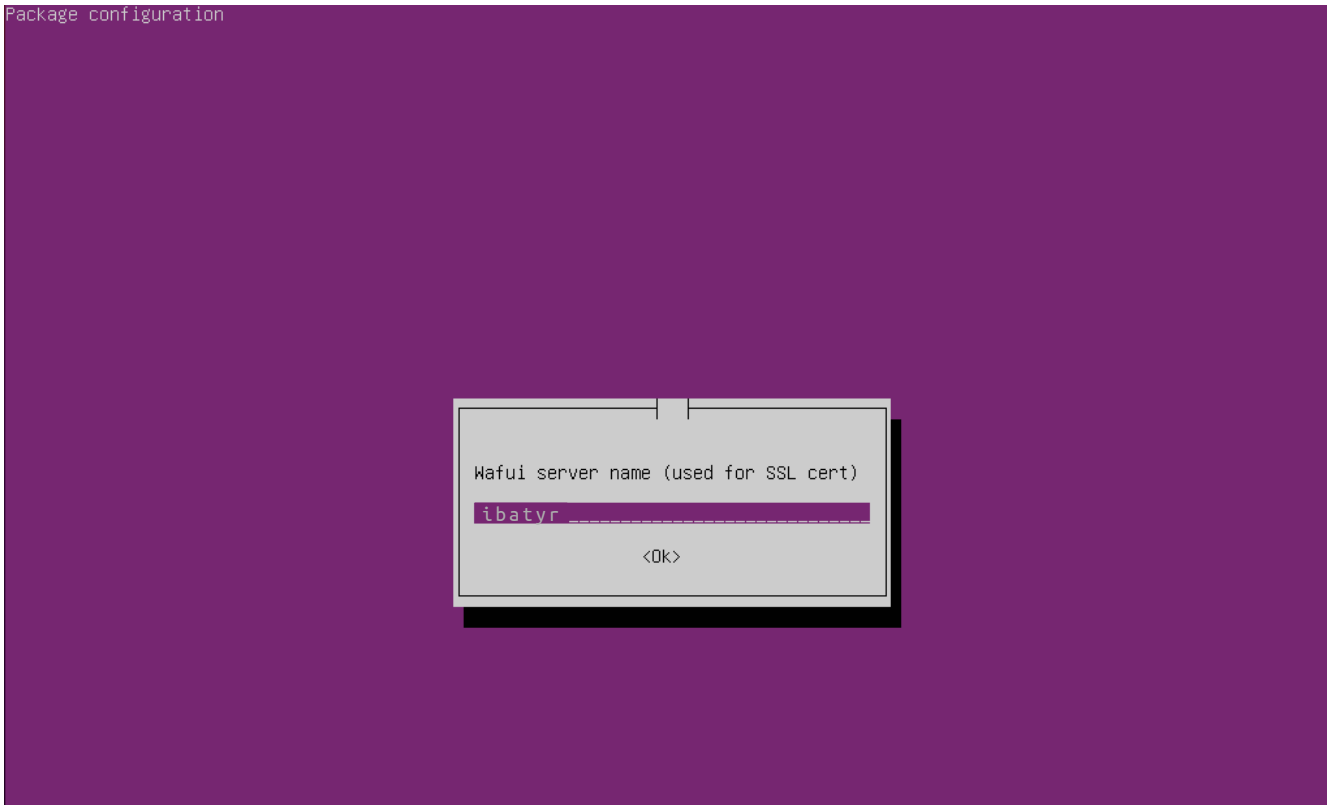
7. В следующем окне ввести пароль подключения к базе данных `mongo` сервера (см. снимок экрана 37).

Снимок экрана 37 – Пароль mongo

8. Затем подтвердить пароль подключения к базе данных `mongo` сервера (см. снимок экрана 38).

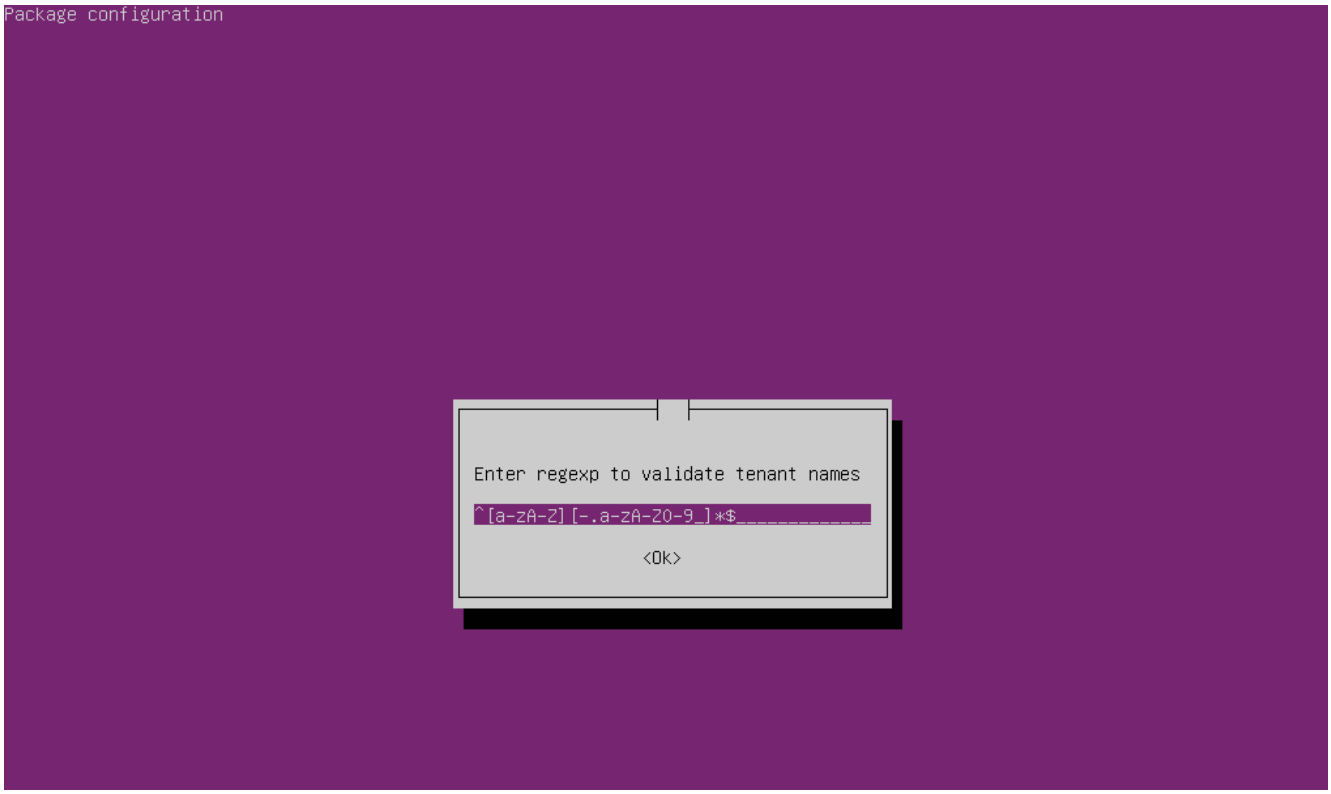
Снимок экрана 38 – Подтверждение пароля mongo

9. В следующем появившемся окне задать имя сервера `WebUI` (см. Снимок экрана 39).



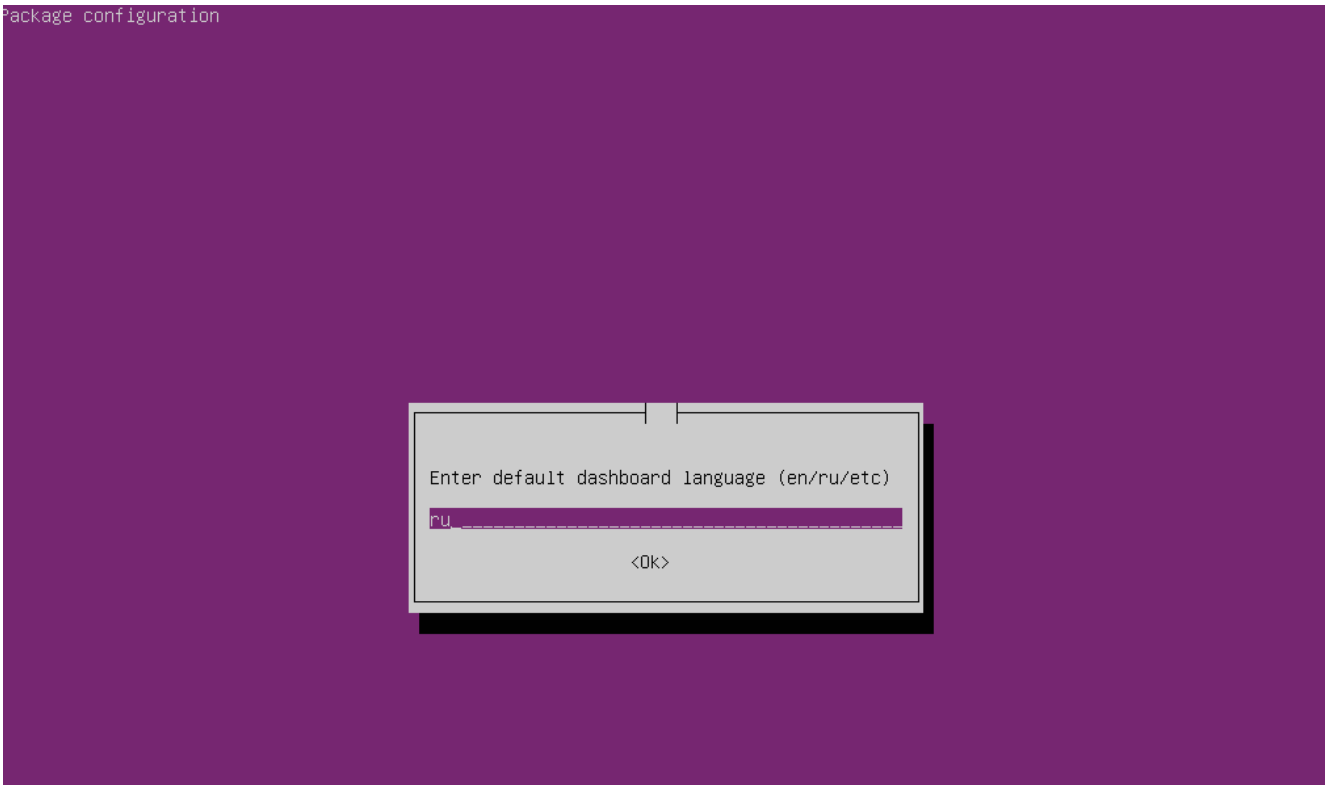
Снимок экрана 39 – Имя сервера WebUI

10. Далее выбрать «Ok» (см. снимок экрана 40).



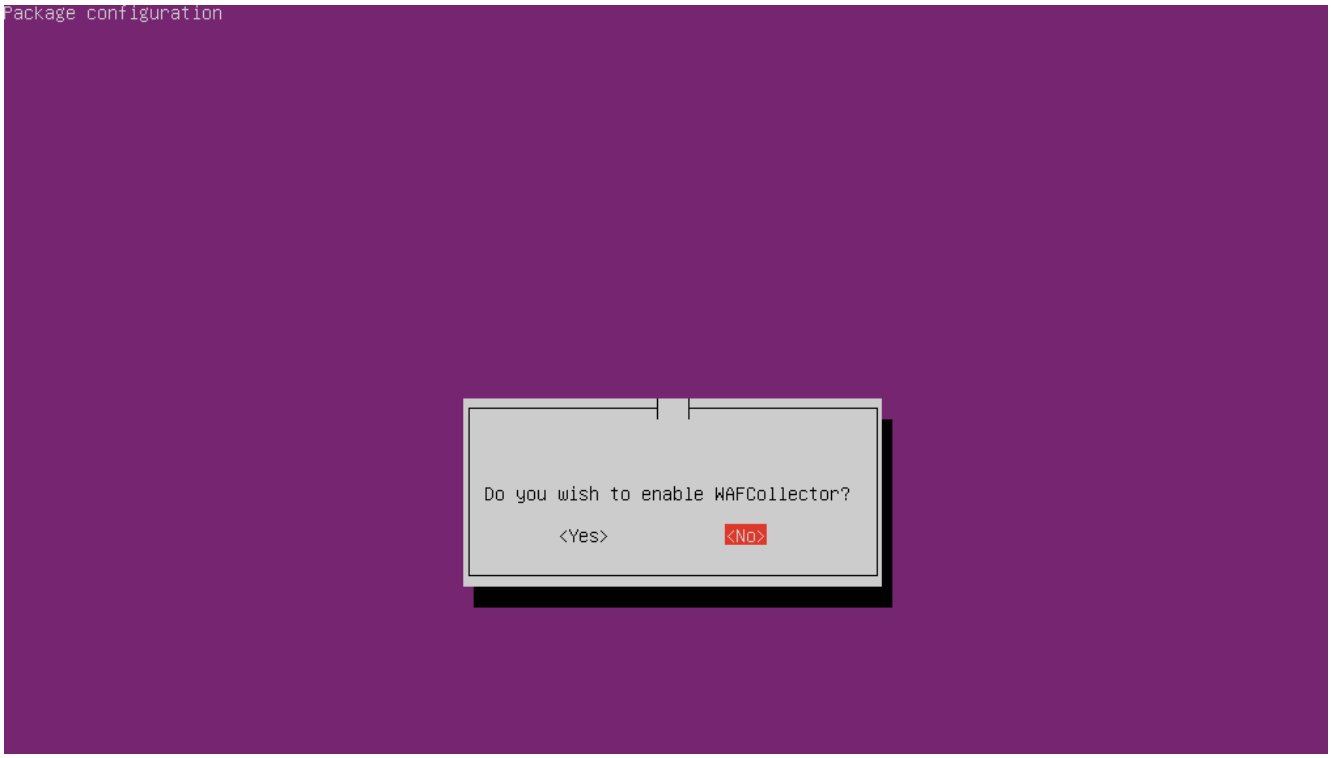
Снимок экрана 40 – Enter regexp to validate tenant names

11. В следующем появившемся окне выбрать язык дашборда по дефолту (см. снимок экрана 41).



Снимок экрана 41 – Язык дашборда

12. В следующем окне выбрать «No» - не включать `WAFCollector` (см. снимок экрана 42**Ошибка! Источник ссылки не найден.**).

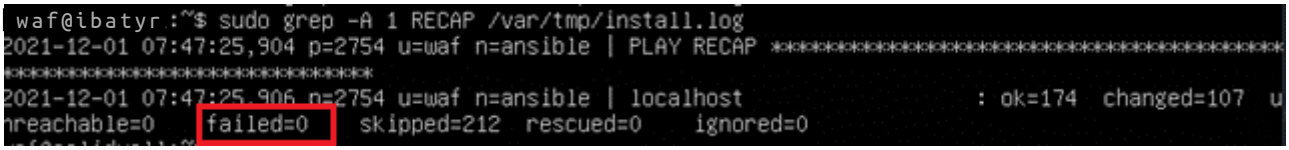


Снимок экрана 42 – `WAFCollector`

13. Затем вывести для проверки обзорную информацию по результатам установки с помощью команды:

```
sudo grep -A 1 RECAP /var/tmp/install.log
```

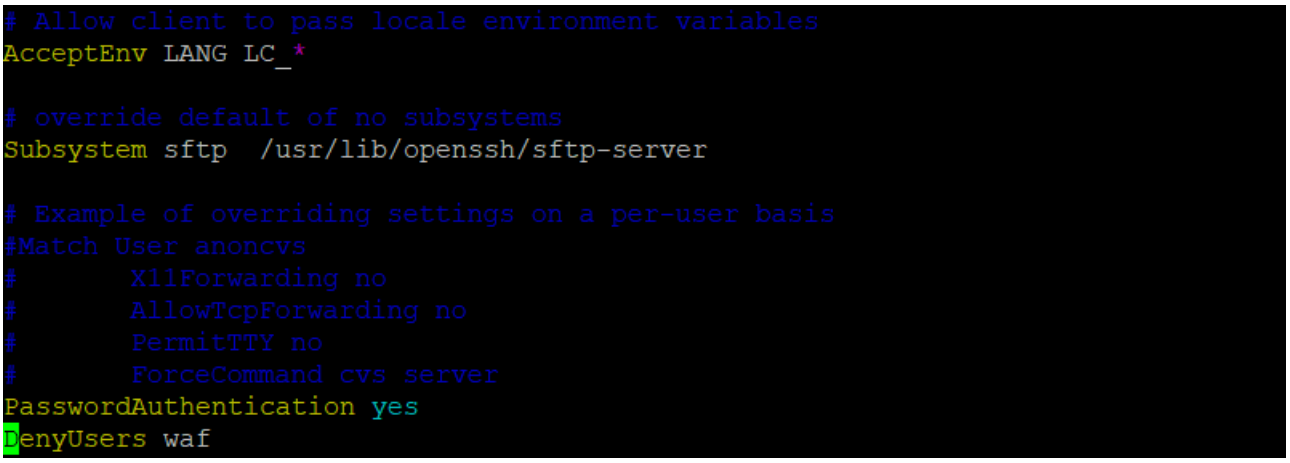
Индикатором успешной установки будет являться отсутствие ошибок установки «failed=0» (см. снимок экрана 43).



Снимок экрана 43 – Установка без ошибок

Для безопасности во время установки был запрещен SSH доступ пользователю `waf`. Если необходимо разрешить такой доступ, то в конфигурационном файле `/etc/sshd/sshd_config` следует удалить последнюю строчку `DenyUsers waf` и перезапустить сервис с помощью команды (см. снимок экрана 44):

```
sudo systemctl restart sshd
```



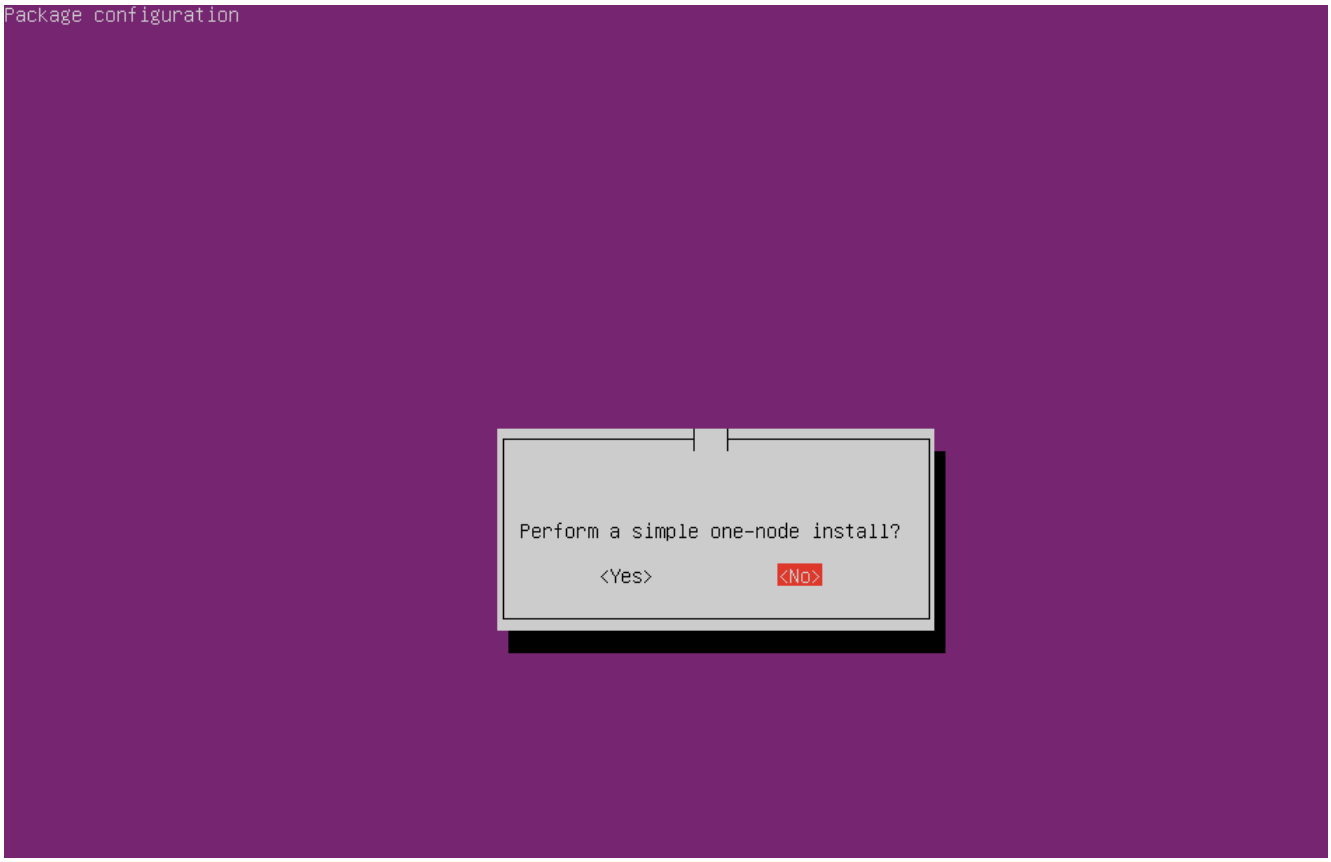
Снимок экрана 44 – Разрешение SSH

4.5.3. Узел анализа

Установка осуществляется из образа ibatyr-ubuntu-2X.04-v2.XX.0.iso (ОС+WAF) на виртуальный или физический сервер. Сервера, на которые планируется устанавливать iBatyr WAF, должны отвечать минимальным системным требованиям из раздела 2.1.

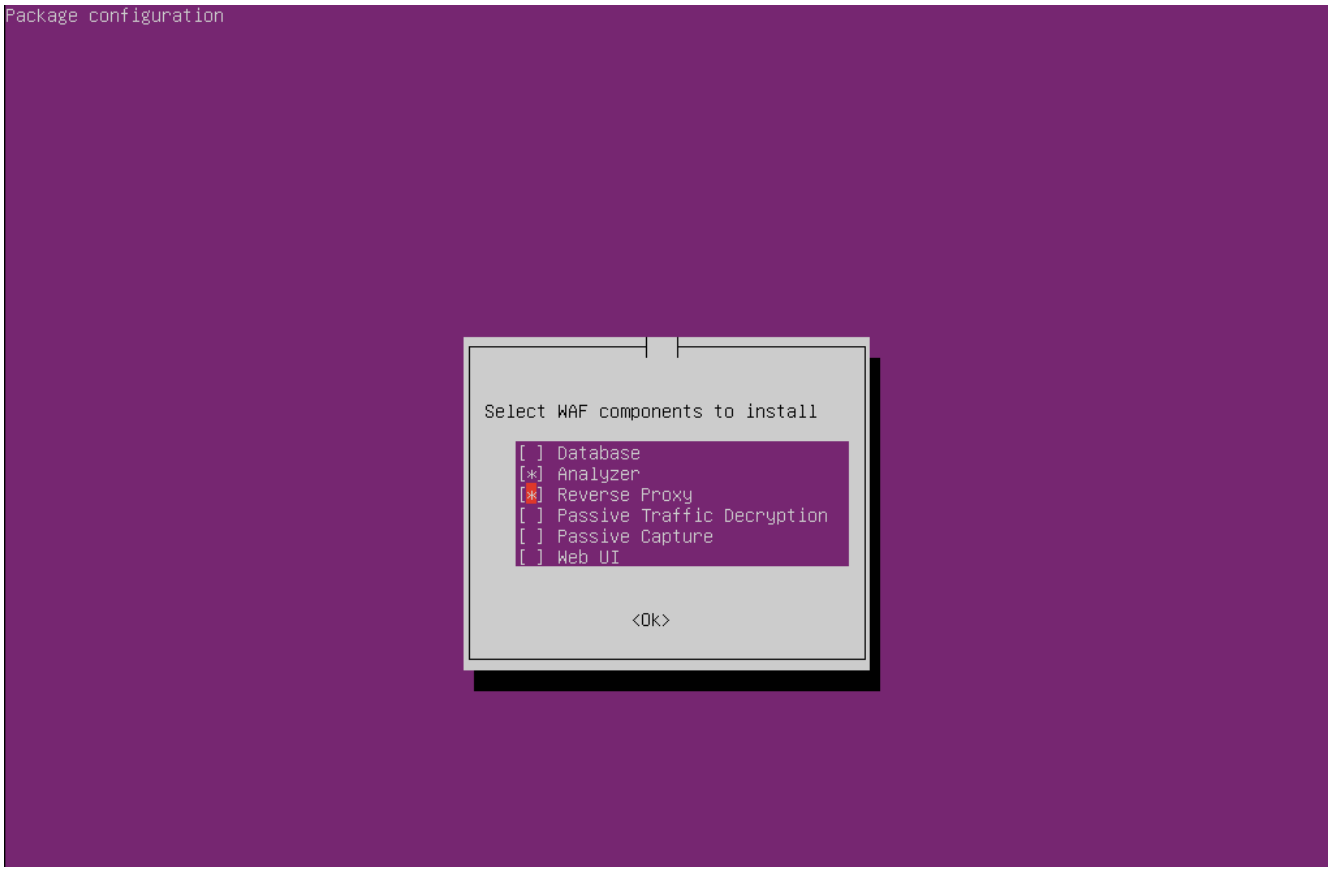
Для начала нужно установить ОС (пример установки ОС приведён в приложении Б. После установки и аутентификации в ОС автоматически запустится установка WAF. Далее необходимо произвести следующие действия:

1. Указать «No» и выбрать установку не на один узел (см. снимок экрана 45).



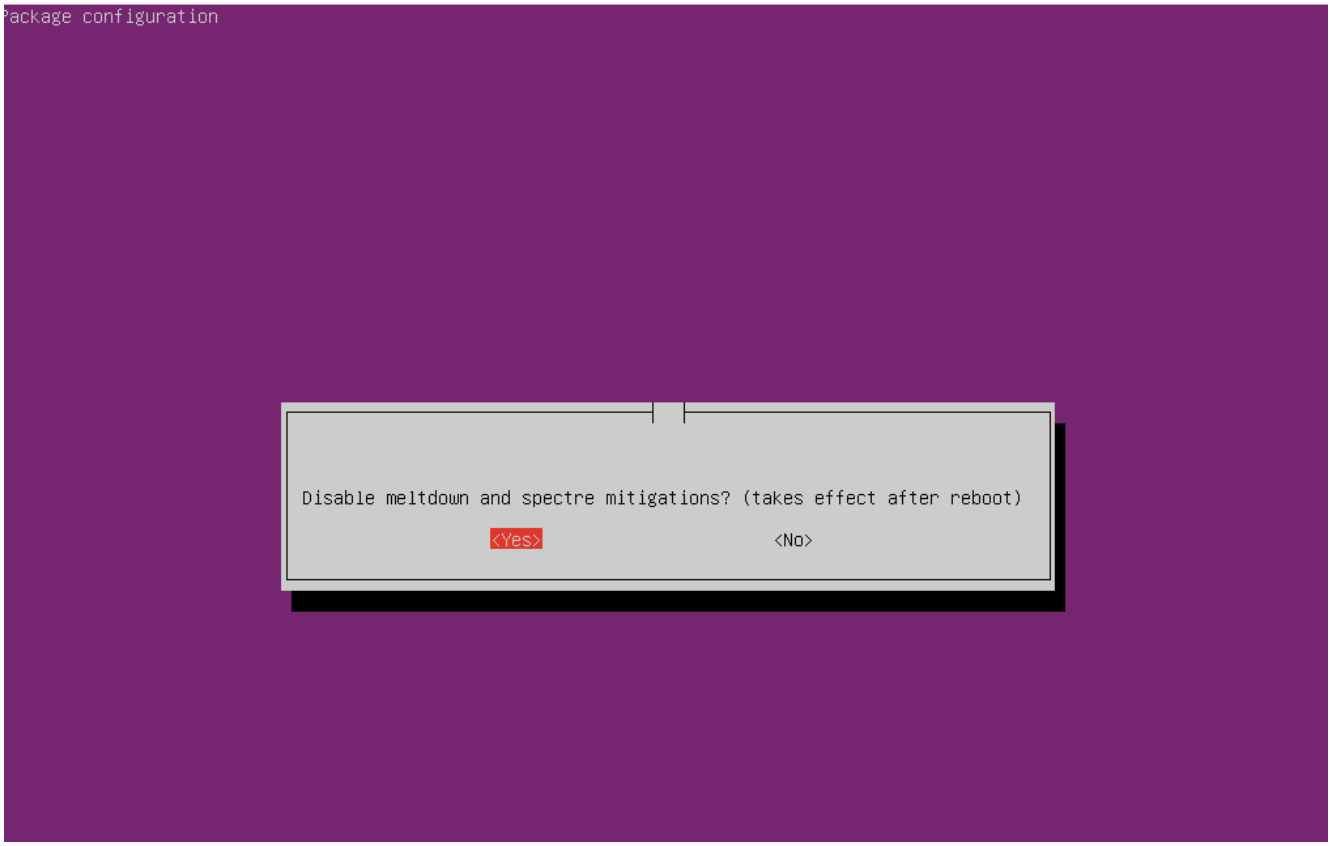
Снимок экрана 45 – Выбор установки

2. В следующем появившемся окне выбрать компоненты установки `Analyzer` и `Reverse Proxy` (см. снимок экрана 46).



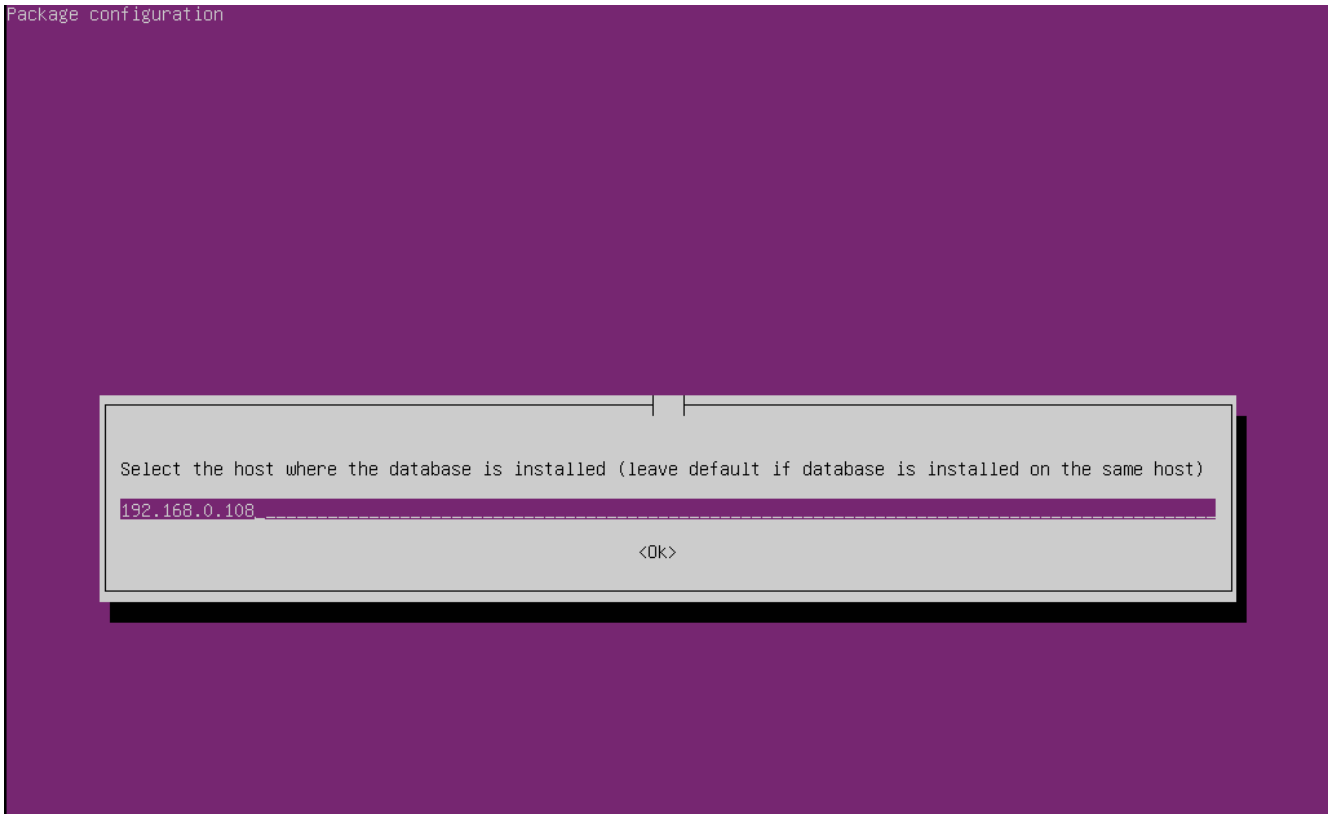
Снимок экрана 46 – Выбор компонентов установки `Analyzer` и `Reverse Proxy`

3. В следующем появившемся окне рекомендуется отключить защиту от эксплуатации уязвимостей процессоров `meltdown` и `spectre`, т.к. это значительно снизит производительность узлов кластера и актуально только на многопользовательских облачных сервисах (см. снимок экрана 47).



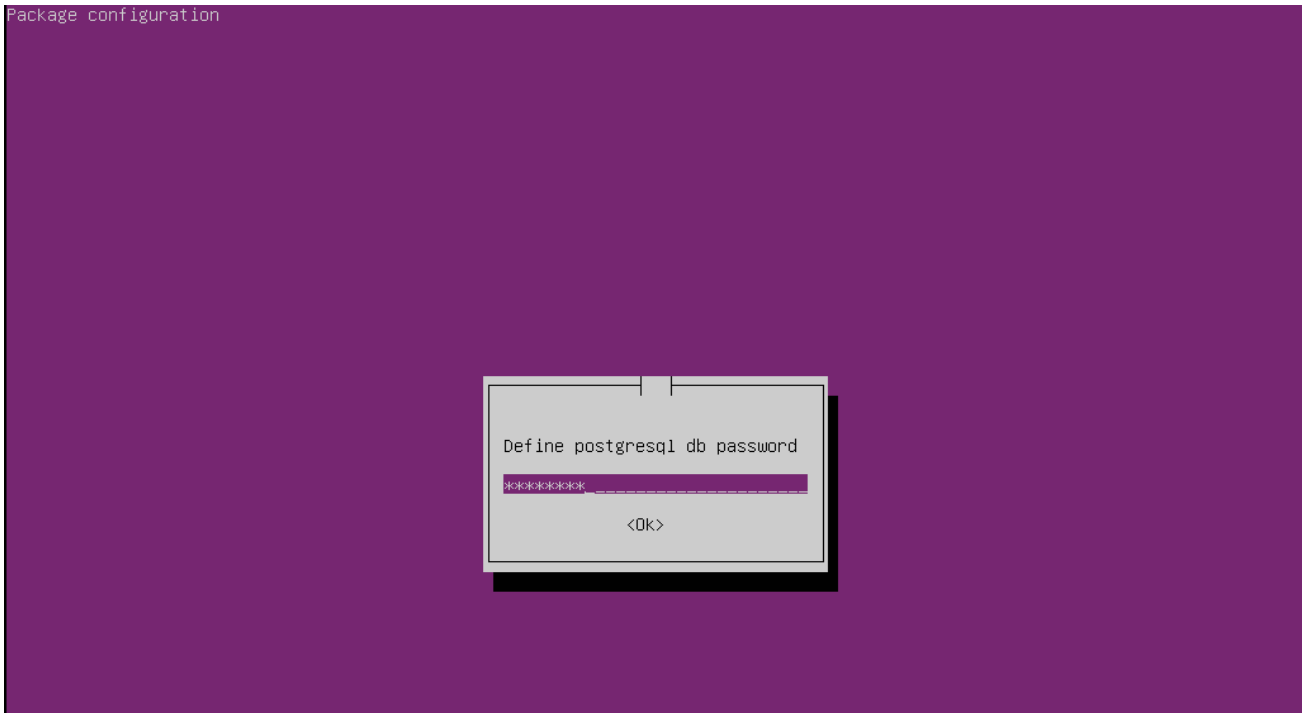
Снимок экрана 47 – Компоненты защиты процессоров

4. В следующем окне ввести IP-адрес сервера с базой данных PostgreSQL (см. снимок экрана 48).



Снимок экрана 48 – IP-адрес сервера с базой данных PostgreSQL

5. Далее в открывшемся окне ввести пароль подключения к PostgreSQL сервера (см. снимок экрана 49).



Снимок экрана 49 – Пароль PostgreSQL

6. В следующем появившемся окне подтвердить пароль подключения к PostgreSQL сервера (см. снимок экрана 50).

Снимок экрана 50 – Подтверждение пароля PostgreSQL

7. В следующем окне ввести пароль подключения к базе данных `mongo` сервера (см. снимок экрана 51).

Снимок экрана 51 – Пароль mongo

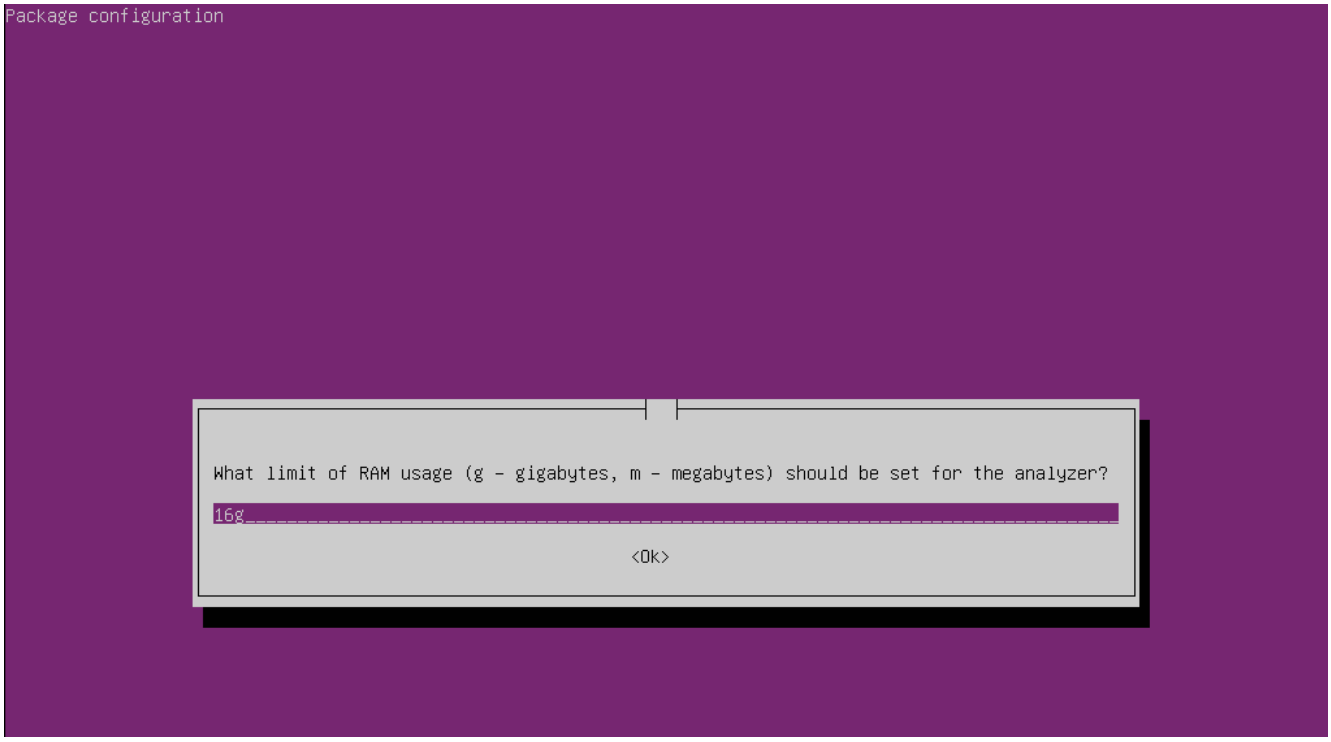
8. Затем подтвердить пароль подключения к базе данных `mongo` сервера (см. снимок экрана 52).

Снимок экрана 52 – Подтверждение пароля mongo

9. В следующем появившемся ввести имя анализатора (см. снимок экрана 53).

Снимок экрана 53 – Имя анализатора

10. В следующем появившемся окне ограничить потребление памяти анализатором (см. снимок экрана 54). Для выделенного анализатора - 70% от объема всей ОЗУ. По умолчанию задано 16ГБ.



Снимок экрана 54 – Лимит памяти

11. Повторить шаги 3 и 4 из подраздела 4.2.

Для безопасности во время установки был запрещен SSH доступ пользователю `waf`. Если необходимо разрешить такой доступ, то в конфигурационном файле `/etc/sshd/sshd_config` следует удалить последнюю строчку `DenyUsers waf` и перезапустить сервис с помощью команды (см. снимок экрана 55):

```
sudo systemctl restart sshd
```

```
# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp  /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
PasswordAuthentication yes
DenyUsers waf
```

Снимок экрана 55 – Разрешение SSH

4.6. Отказоустойчивый кластер с репликацией на два узла

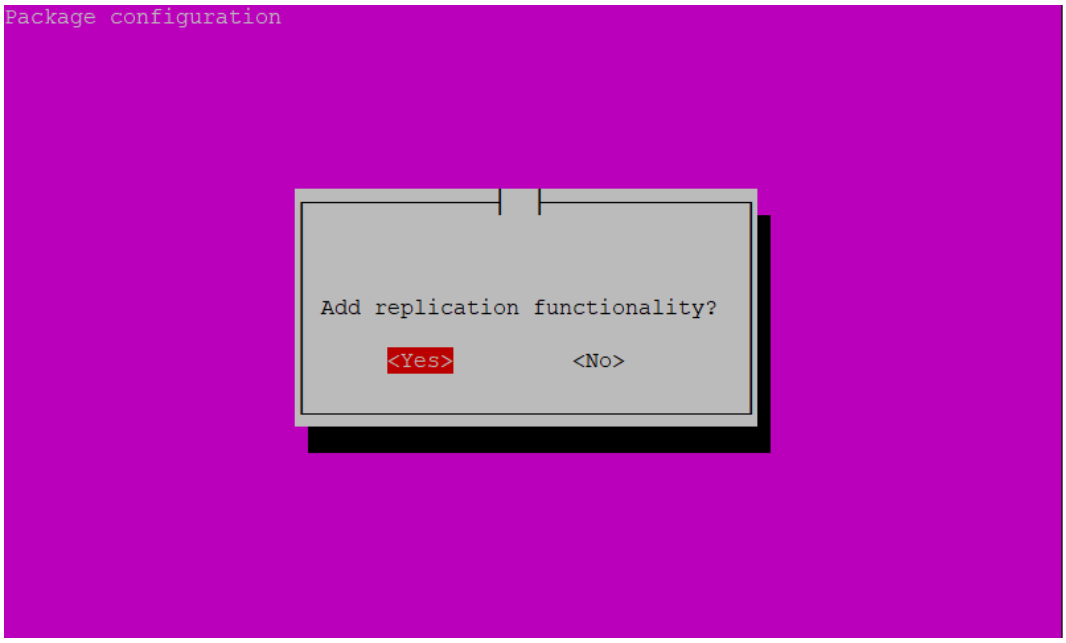
Установка осуществляется из образа `ibatyr-ubuntu-2X.04-v2.XX.0-replication.iso` (ОС+WAF) на виртуальный или физический сервер. Сервера, на которые планируется устанавливать iBatyr WAF, должны отвечать минимальным системным требованиям из раздела 2.1.

Для начала нужно установить ОС (пример установки ОС приведён в приложении б. После установки и аутентификации в ОС автоматически запустится установка WAF.

Перед началом установки кластера Управления из двух узлов следует выделить три IP-адреса из одной подсети – по одному на каждый узел и одному для Virtual IP, который будет всегда привязан к данному master-узлу.

Далее необходимо выполнить следующие действия:

1. В появившемся окне выбрать установку с репликацией (см. снимок экрана 56).

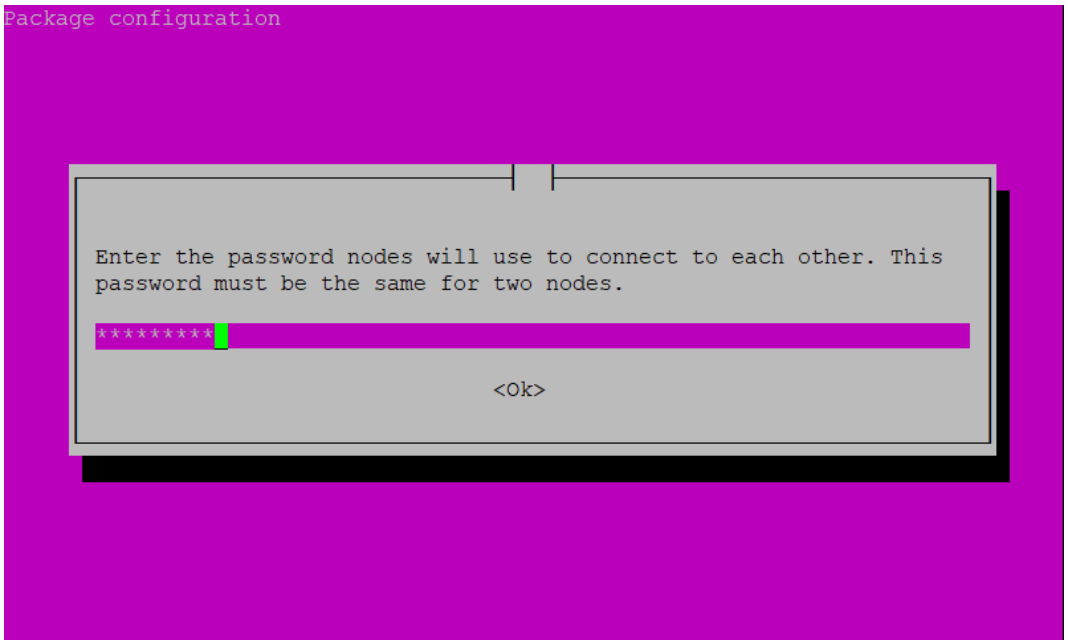


Снимок экрана 56 – Выбор типа установки

2. В следующем появившемся окне выбрать основной узел (см. снимок экрана 57).

Снимок экрана 57 – Выбор основного узла

3. В следующем окне ввести пароль для взаимодействия основного (master) узла с резервным (slave) (см. снимок экрана 58).



Снимок экрана 58 – Пароль для взаимодействия основного и резервного узлов

4. Далее нужно подтвердить пароль (см. снимок экрана 59).

Снимок экрана 59 – Подтверждение пароля

5. Затем ввести IP-адрес основного узла (см. снимок экрана 60).

Снимок экрана 60 – IP-адрес основного узла

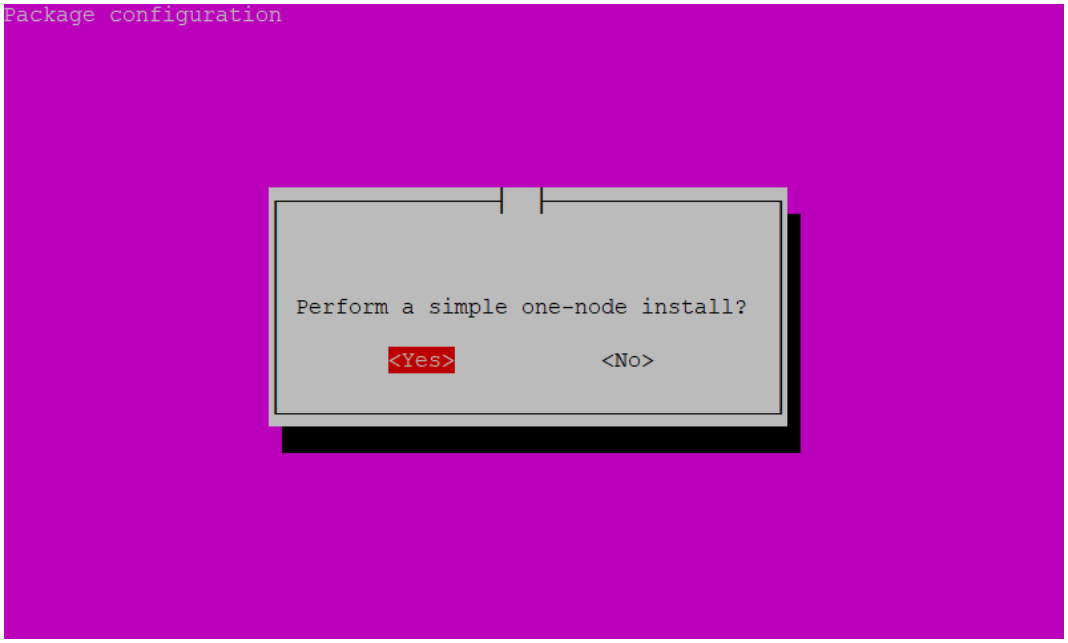
6. Затем ввести адрес резервного узла (см. снимок экрана 61).

Снимок экрана 61 – IP-адрес резервного узла

7. Далее ввести VIP-адрес (см. снимок экрана 62).

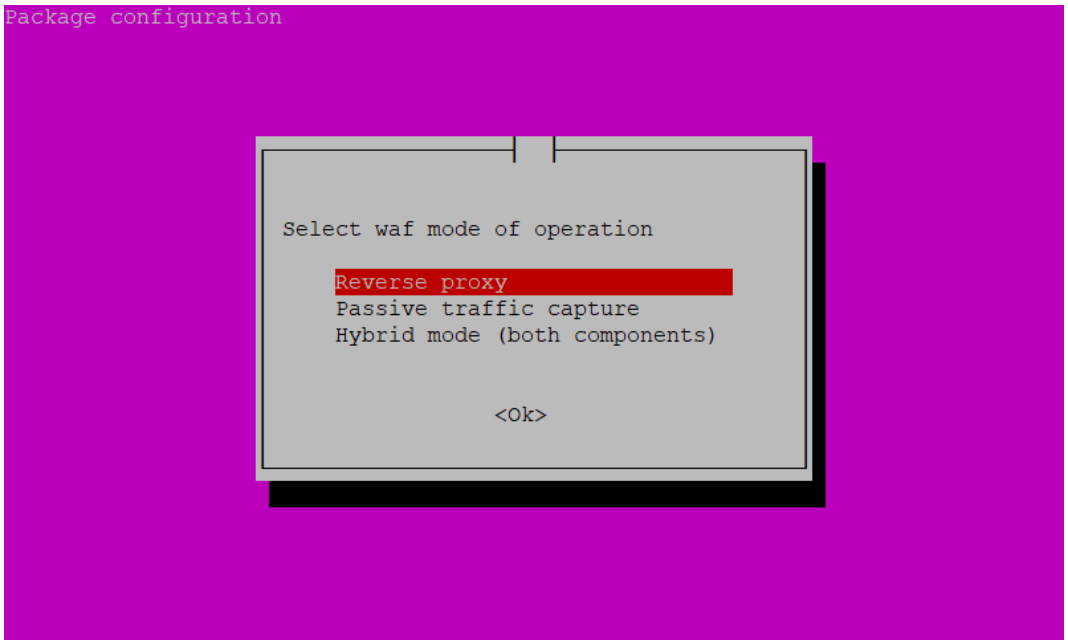
Снимок экрана 62 – VIP-адрес

8. В появившемся окне выбрать простую установку на один узел (см. снимок экрана 63).



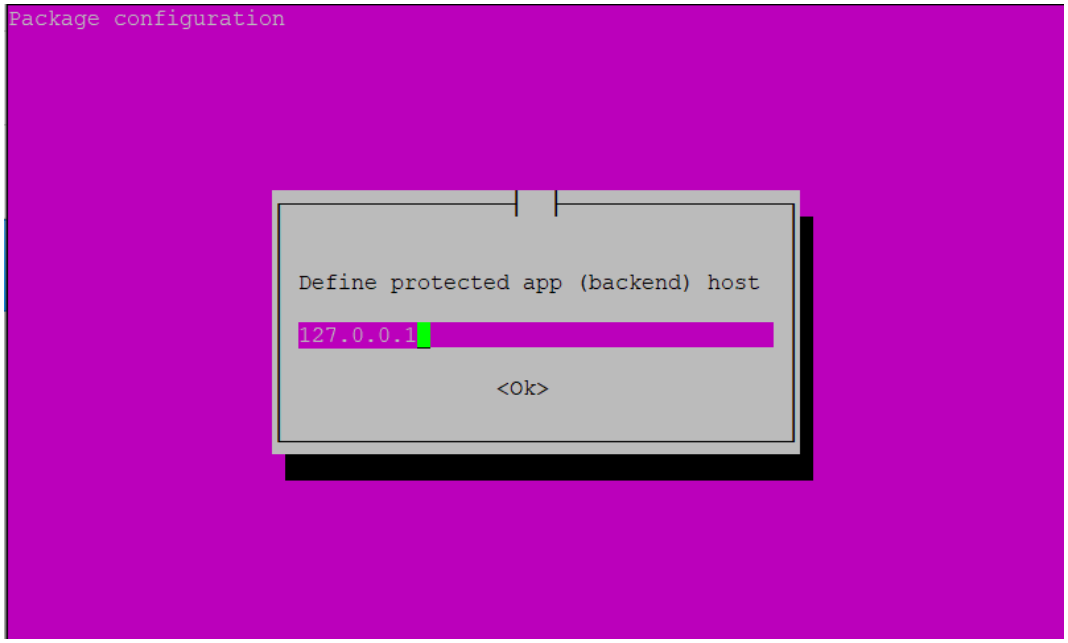
Снимок экрана 63 – Выбор типа установки

9. Далее в окне выбора режима установки выбрать `Reverse proxy` (см. снимок экрана 64).

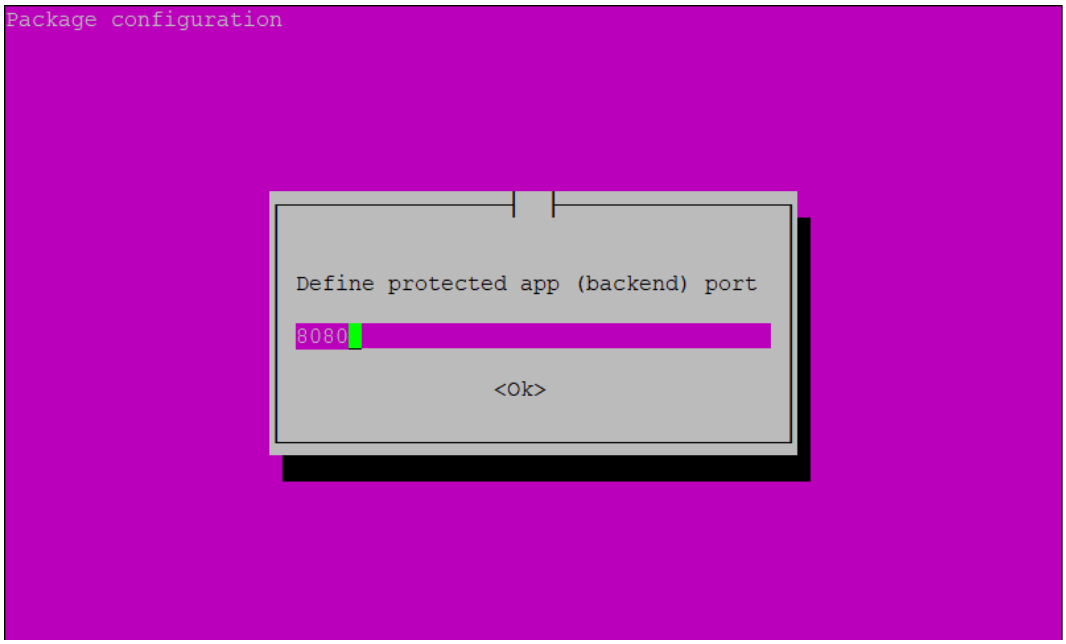


Снимок экрана 64 – Выбор режима установки

10. После, в соответствующих окнах, указать IP-адрес и порт защищаемого приложения (см. снимок экрана 65, снимок экрана 66). Введенные на данном этапе значения можно будет изменить после установки.



Снимок экрана 65 – IP-адреса защищаемого приложения

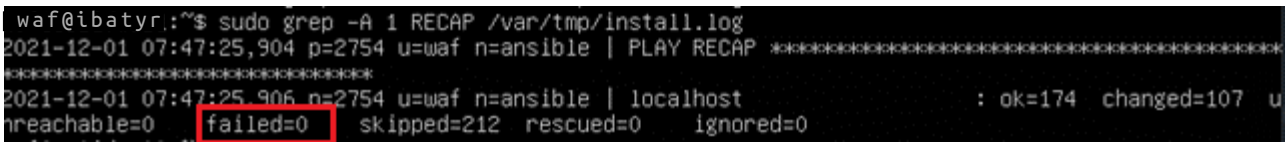


Снимок экрана 66 – Порт защищаемого приложения

11. Вывести для проверки обзорную информацию по результатам установки с помощью команды:

```
sudo grep -A 1 RECAP /var/tmp/install.log
```

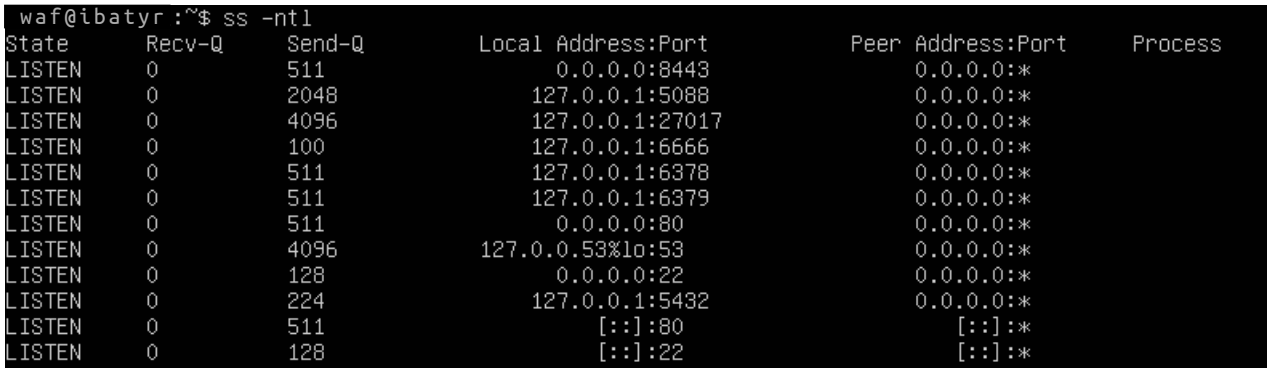
Индикатором успешной установки будет являться отсутствие ошибок установки «failed=0» (см. снимок экрана 67).



Снимок экрана 67 – Установка без ошибок

12. Для проверки соответствия портов сервисам необходимо воспользоваться командой (см. Снимок экрана 68):

```
ss -ntl
```



Снимок экрана 68 – Соответствие портов сервисов

Порты для сервисов должны соответствовать:

- nginx: port 8443;

- ibatyr-dashboard: port 5088;
- mongod: port 27017;
- ibatyr-analyzer@0: port 6666;
- ibatyr-redis@mgmt: port 6378;
- ibatyr-redis@0: port 6379;
- ibatyr-nginx: порт, который вы указали при установке WAF;
- sshd: port 22;
- postgresql: port 5432.

13. Далее нужно проверить состояние кластера командой `sudo crm status` (см. снимок экрана 69).

```
Cluster Summary:
  * Stack: corosync
  * Current DC: waf-repl-node-1 (version 2.0.3-4b1f869f0f) - partition WITHOUT q
uorum
  * Last updated: Tue May 17 16:57:14 2022
  * Last change:  Tue May 17 16:52:38 2022 by root via crm_attribute on waf-repl
-node-1
  * 2 nodes configured
  * 16 resource instances configured

Node List:
  * Online: [ waf-repl-node-1 ]
  * OFFLINE: [ waf-repl-node-2 ]

Full List of Resources:
  * Resource Group: master:
    * waf-virtual-ip    (ocf::heartbeat:IPaddr2):        Started waf-repl-node-1
    * mongo-arbiter     (systemd:mongod-arbiter):        Started waf-repl-node-1
    *     ibatyr-celery  (systemd:    ibatyr-celery):      Started waf-repl-node-1
    *     ibatyr-celerybeat      (systemd:    ibatyr-celerybeat):  Started waf-rep
l-node-1
  * Clone Set: ReplicatedPg [pgsql] (promotable):
    * Masters: [ waf-repl-node-1 ]
```

Снимок экрана 69 – Результаты проверки состояния кластера

14. После инициализировать слот репликации путем перезапуска ресурса `pgsql` командой (см. снимок экрана 70):

```
sudo crm resource restart pgsql
```

```
waf@wafsrv:~/  ibatyr  -distrib$ sudo crm status
waf@wafsrv:~/  ibatyr  -distrib$ sudo crm resource restart pgsql
```

Снимок экрана 70 – Инициализация слота репликации

Во время выполнения данной команды ресурс сначала перейдет в статус Slaves (см. снимок экрана 71). Нужно дождаться, когда он вернется в состояние Masters (см. снимок экрана 72). Проверка состояния кластера выполняется следующей командой:

```
sudo crm status
```

```
Node List:
  * Online: [ waf-repl-node-1 ]
  * OFFLINE: [ waf-repl-node-2 ]

Full List of Resources:
  * Resource Group: master:
    * waf-virtual-ip    (ocf::heartbeat:IPaddr2):        Started waf-repl-node-1
    * mongo-arbiter     (systemd:mongod-arbiter):        Started waf-repl-node-1
    *    ibatyr-celery  (systemd:   ibatyr-celery):      Started waf-repl-node-1
    *    ibatyr-celerybeat      (systemd:   ibatyr-celerybeat):  Started waf-rep
l-node-1
  * Clone Set: ReplicatedPg [pgsql] (promotable):
    * Slaves: [ waf-repl-node-1 ]
    * Stopped: [ waf-repl-node-2 ]
  * Clone Set: ReplicatedRedis [  ibatyr-redis-mgmt] (promotable):
    * Masters: [ waf-repl-node-1 ]
    * Stopped: [ waf-repl-node-2 ]
  * Clone Set: mongo-multi [mongo]:
    * Started: [ waf-repl-node-1 ]
    * Stopped: [ waf-repl-node-2 ]
  * Clone Set:    ibatyr-analyzer-clone [   ibatyr-analyzer]:
    * Started: [ waf-repl-node-1 ]
```

Снимок экрана 71 – Статус Slaves

```
Node List:
  * Online: [ waf-repl-node-1 ]
  * OFFLINE: [ waf-repl-node-2 ]

Full List of Resources:
  * Resource Group: master:
    * waf-virtual-ip    (ocf::heartbeat:IPaddr2):        Started waf-repl-node-1
    * mongo-arbiter     (systemd:mongod-arbiter):        Started waf-repl-node-1
    *    ibatyr-celery  (systemd:   ibatyr-celery):      Started waf-repl-node-1
    *    ibatyr-celerybeat      (systemd:   ibatyr-celerybeat):  Started waf-rep
l-node-1
  * Clone Set: ReplicatedPg [pgsql] (promotable):
    * Masters: [ waf-repl-node-1 ]
    * Stopped: [ waf-repl-node-2 ]
  * Clone Set: ReplicatedRedis [  ibatyr-redis-mgmt] (promotable):
    * Masters: [ waf-repl-node-1 ]
    * Stopped: [ waf-repl-node-2 ]
  * Clone Set: mongo-multi [mongo]:
    * Started: [ waf-repl-node-1 ]
    * Stopped: [ waf-repl-node-2 ]
  * Clone Set:    ibatyr-analyzer-clone [   ibatyr-analyzer]:
    * Started: [ waf-repl-node-1 ]
```

Снимок экрана 72 – Статус Master

Если выделенные адреса имеют маску не 255.255.255.0 (24), то необходимо изменить маску в конфигурационном файле для VIP-ресурса `waf-virtual-ip`. Для этого следует:

1. Выполнить команду `sudo crm configure edit`.
2. В открывшемся редакторе `vim`, в файле найти ресурс и сменить `cidr_netmask=24` на нужную маску (см. снимок экрана 73).

```
        op monitor timeout=30s interval=10s on-fail=restart \
        op stop timeout=60s interval=0
primitive solidwall-redis-mgmt redis \
        params config="/etc/ ibatyr /redis/mgmt.conf" rundir="/var/run/ibatyr
l/redis" pidfile_name=mgmt.pid socket_name=mgmt.sock port=6378 \
        op start timeout=120s interval=0s on-fail=restart \
        op promote timeout=120s interval=0s on-fail=restart \
        op stop timeout=120s interval=0s on-fail=block \
        op notify timeout=90s interval=0s \
        op monitor timeout=120s interval=90s on-fail=restart \
        op monitor timeout=120s interval=80s on-fail=restart role=Master \
        op demote timeout=120s interval=0s on-fail=stop
primitive waf-virtual-ip IPaddr2 \
        params ip=192.168.0.108 cidr_netmask=24 iflabel=waf0 \
        op start timeout=60s interval=0s on-fail=stop \
        op stop timeout=60s interval=0s on-fail=block \
        op monitor timeout=60s interval=10s on-fail=restart
group master waf-virtual-ip mongo-arbiter    ibatyr-celery    ibatyr-celerybeat
\
```

Снимок экрана 73 – Смена сетевой маски

Для безопасности во время установки был запрещен SSH доступ пользователю `waf`. Если необходимо разрешить такой доступ, то в конфигурационном файле `/etc/sshd/sshd_config` следует удалить последнюю строчку `DenyUsers waf` и перезапустить сервис с помощью команды (см. снимок экрана 74):

```
sudo systemctl restart sshd
```

```
# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp  /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
PasswordAuthentication yes
DenyUsers waf
```

Снимок экрана 74 – Разрешение SSH

4.6.1. Установка резервного узла (slave) отказоустойчивого кластера

Чтобы установить резервный узел (slave) отказоустойчивого кластера, нужно:

1. На резервном узле кластера повторить шаги с 1 по 11 из подраздела 4.6, при этом на шаге 2 указать вместо «Master» – «Slave» (см. снимок экрана 75).

Снимок экрана 75 – Развёртывание резервного узла кластера

2. Проверить состояние кластера командой `sudo crm status`: оба узла должны быть в статусе Online (см. Снимок экрана 76).

```
  * Online: [ waf-repl-node-1 waf-repl-node-2 ]

Full List of Resources:
  * Resource Group: master:
    * waf-virtual-ip    (ocf::heartbeat:IPaddr2):        Started waf-repl-node-1
    * mongo-arbiter     (systemd:mongod-arbiter):        Started waf-repl-node-1
    *    ibatyr-celery  (systemd:   ibatyr-celery):      Started waf-repl-node-1
    *    ibatyr-celerybeat      (systemd:   ibatyr-celerybeat):  Started waf-rep
l-node-1
  * Clone Set: ReplicatedPg [pgsql] (promotable):
    * Masters: [ waf-repl-node-1 ]
    * Slaves: [ waf-repl-node-2 ]
  * Clone Set: ReplicatedRedis [   ibatyr-redis-mgmt] (promotable):
    * Masters: [ waf-repl-node-1 ]
    * Slaves: [ waf-repl-node-2 ]
  * Clone Set: mongo-multi [mongo]:
    * Started: [ waf-repl-node-1 waf-repl-node-2 ]
  * Clone Set:    ibatyr-analyzer-clone [   ibatyr-analyzer]:
    * Started: [ waf-repl-node-1 waf-repl-node-2 ]
  * Clone Set:    ibatyr-nginx-clone [   ibatyr-nginx]:
    * Started: [ waf-repl-node-1 waf-repl-node-2 ]
  * Clone Set:    ibatyr-dashboard-clone [   ibatyr-dashboard]:
    * Started: [ waf-repl-node-1 waf-repl-node-2 ]
```

Снимок экрана 76 – Проверка состояния кластера

3. При необходимости в конфигурационном файле `/etc/sshd/sshd_config` удалить последнюю строчку `DenyUsers waf` и перезапустить сервис с помощью команды (см. снимок экрана 77):

```
sudo systemctl restart sshd
```

```
# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp  /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
PasswordAuthentication yes
DenyUsers waf
```

Снимок экрана 77 – Конфигурационный файл службы SSH

4.7. Установка распределенной конфигурации с репликацией узлов Управления

Установка осуществляется из образа `ibatyr-ubuntu-20.04-v2.XX.0-replication.iso` (ОС+WAF) на виртуальный или физический сервер. Сервера, на которые планируется устанавливать iBatyr WAF, должны отвечать минимальным системным требованиям из раздела 2.1.

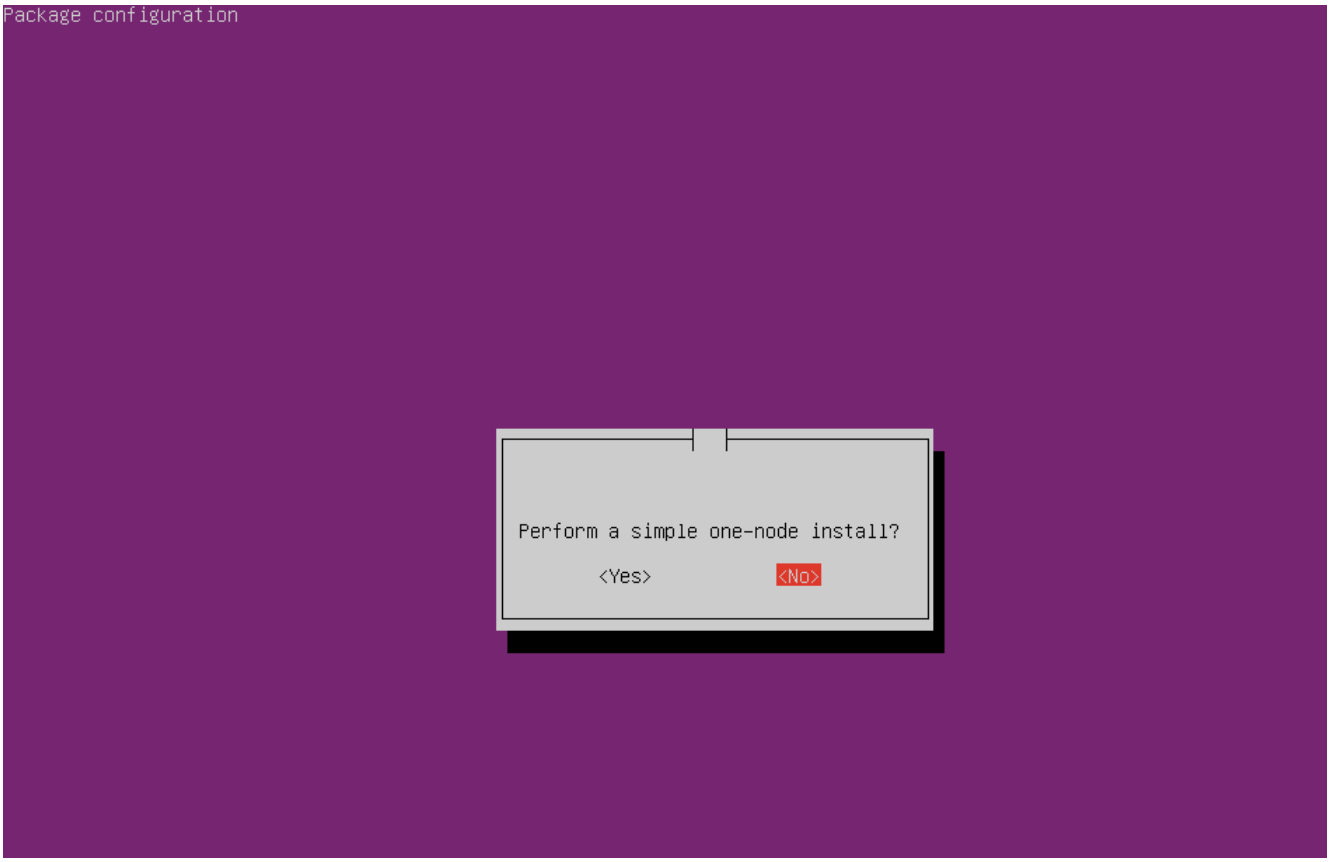
4.7.1. Установка основного узла (master) кластера узлов Управления

Для начала нужно установить ОС (пример установки ОС приведён в приложении Приложение Б).

Перед началом установки кластера узлов Управления из двух узлов следует выделить три IP-адреса в одной подсети – по одному на каждый узел и одному для Virtual IP, который будет всегда привязан к данному master-узлу.

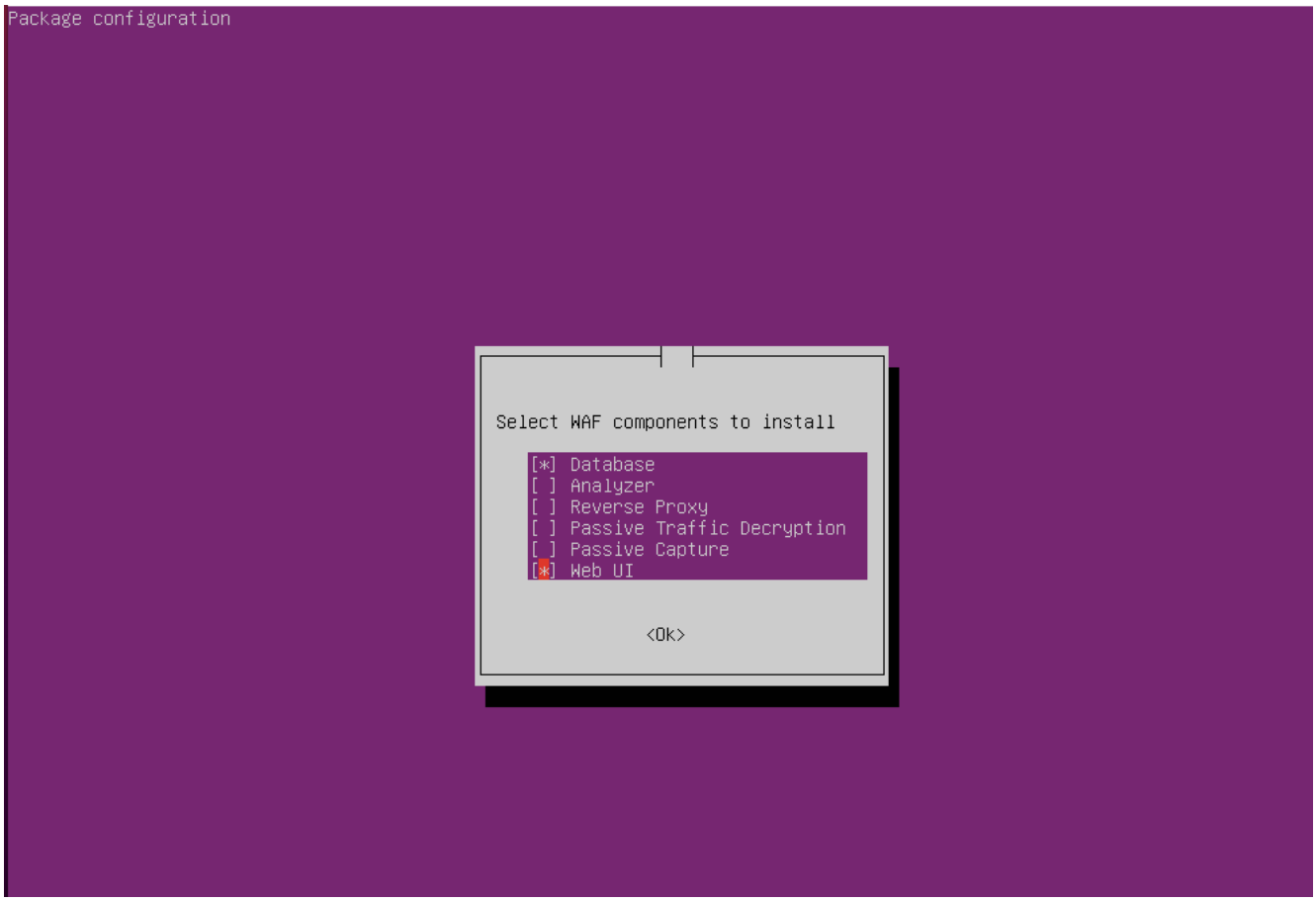
Далее необходимо произвести следующие действия:

1. Повторить шаги с 1 по 7 из подраздела 4.6.
2. На шаге 8 указать «No» и выбрать установку не на один узел (см. снимок экрана 78).



Снимок экрана 78 – Выбор установки

3. В следующем появившемся окне выбрать компоненты установки `Database` и `WebUI` (см. снимок экрана 79).

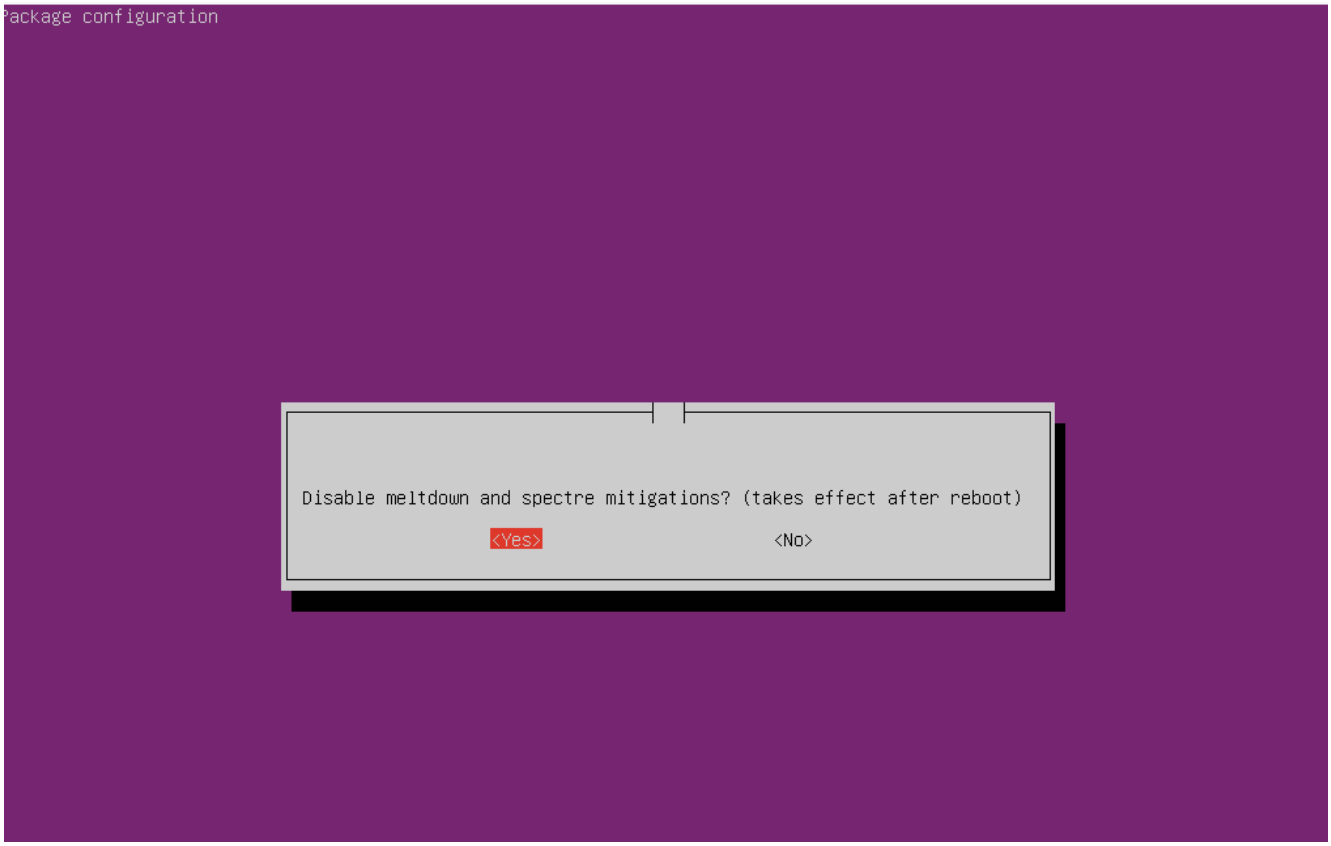


Снимок экрана 79 – Компоненты установки

4. В следующем появившемся окне выбрать «Ok» на предупреждение «Если вы решите вручную вводить пароли к базам данных, обратите внимание, что при репликации пароли для обоих узлов ДОЛЖНЫ БЫТЬ ОДИНАКОВЫМИ» (см. снимок экрана 80).

Снимок экрана 80 – Предупреждение

5. Далее в появившемся окне рекомендуется отключить защиту от эксплуатации уязвимостей процессоров `meltdown` и `spectre`, т.к. это значительно снизит производительность узлов кластера и актуально только на многопользовательских облачных сервисах (см. снимок экрана 81).



Снимок экрана 81 – Компоненты защиты процессоров

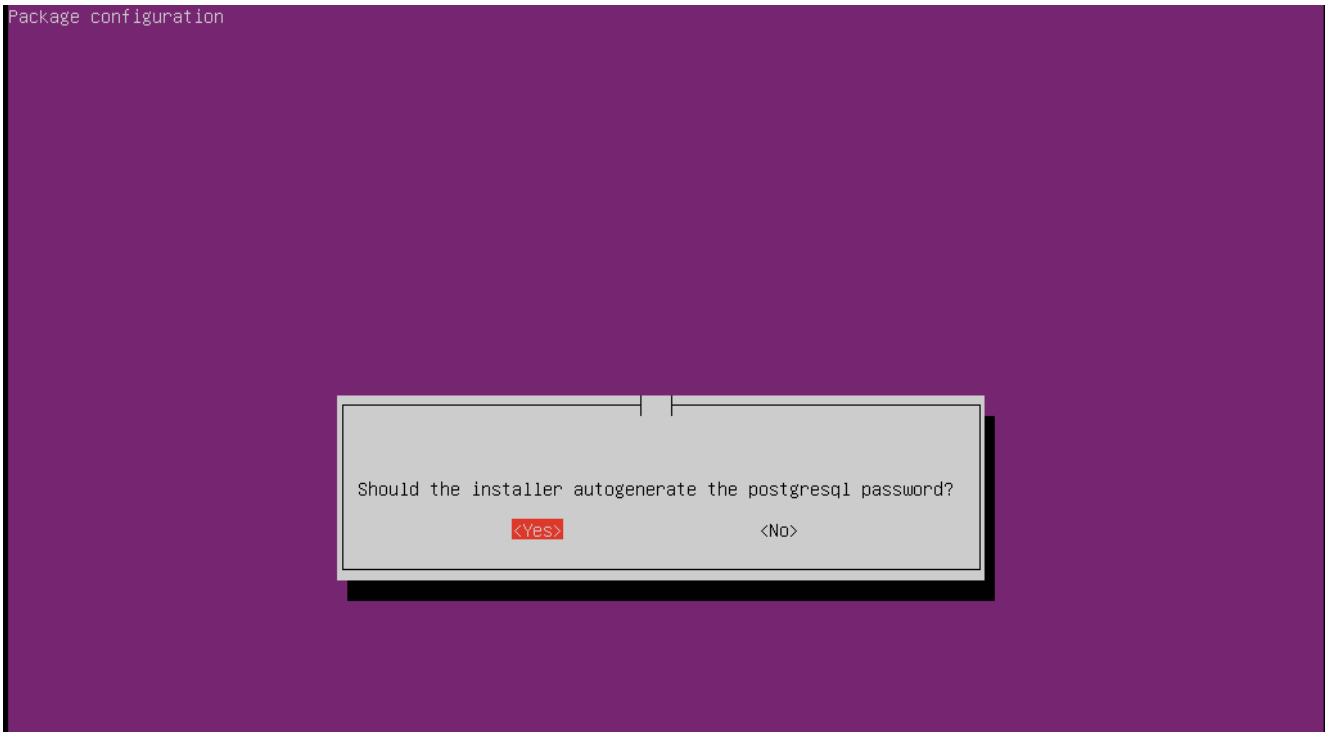
6. В следующем окне выбрать количество дней хранения записи о заблокированных транзакциях в БД PostgreSQL (см. снимок экрана 82).

Снимок экрана 82 – Время хранения о заблокированных транзакциях в БД PostgreSQL

7. Далее выбрать количество дней хранения обычных транзакций в БД PostgreSQL (см. снимок экрана 83).

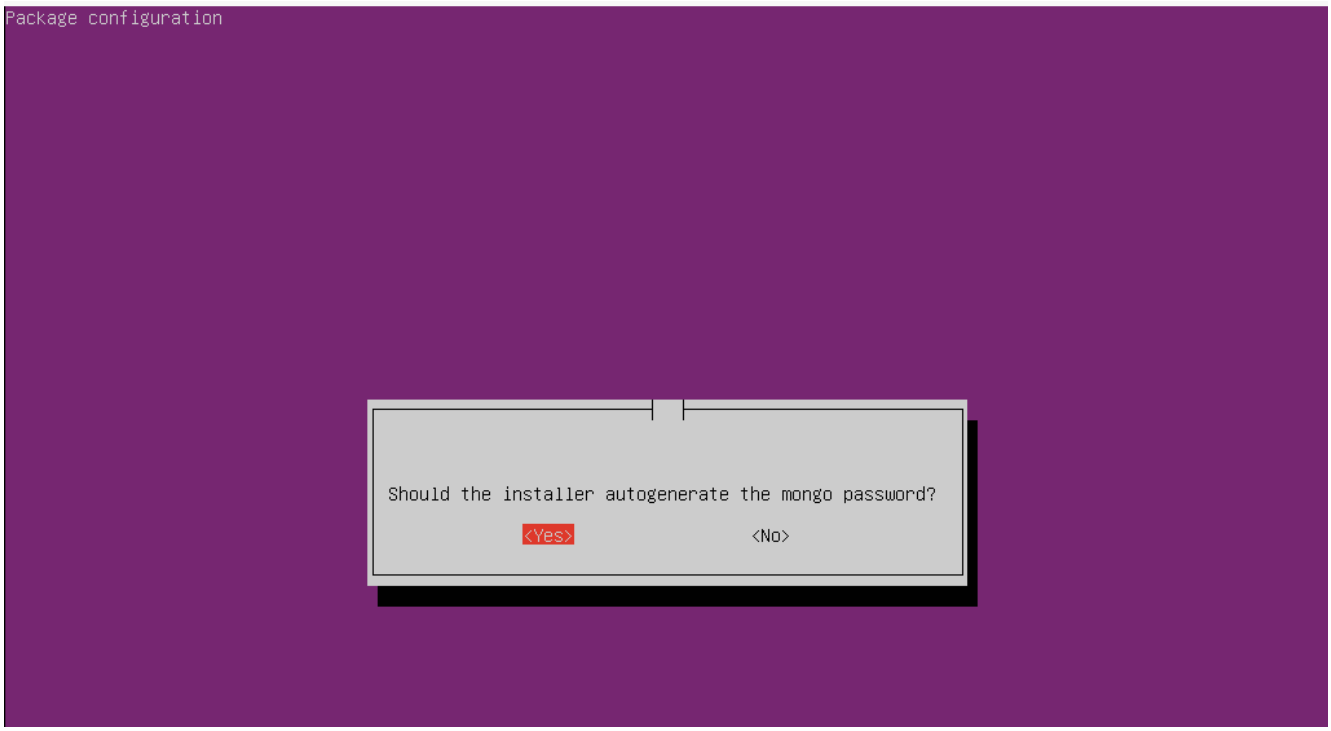
Снимок экрана 83. Время хранения обычных транзакций в БД PostgreSQL

8. В следующем появившемся окне выбрать автоматическую генерацию пароля PostgreSQL (см. снимок экрана 84).



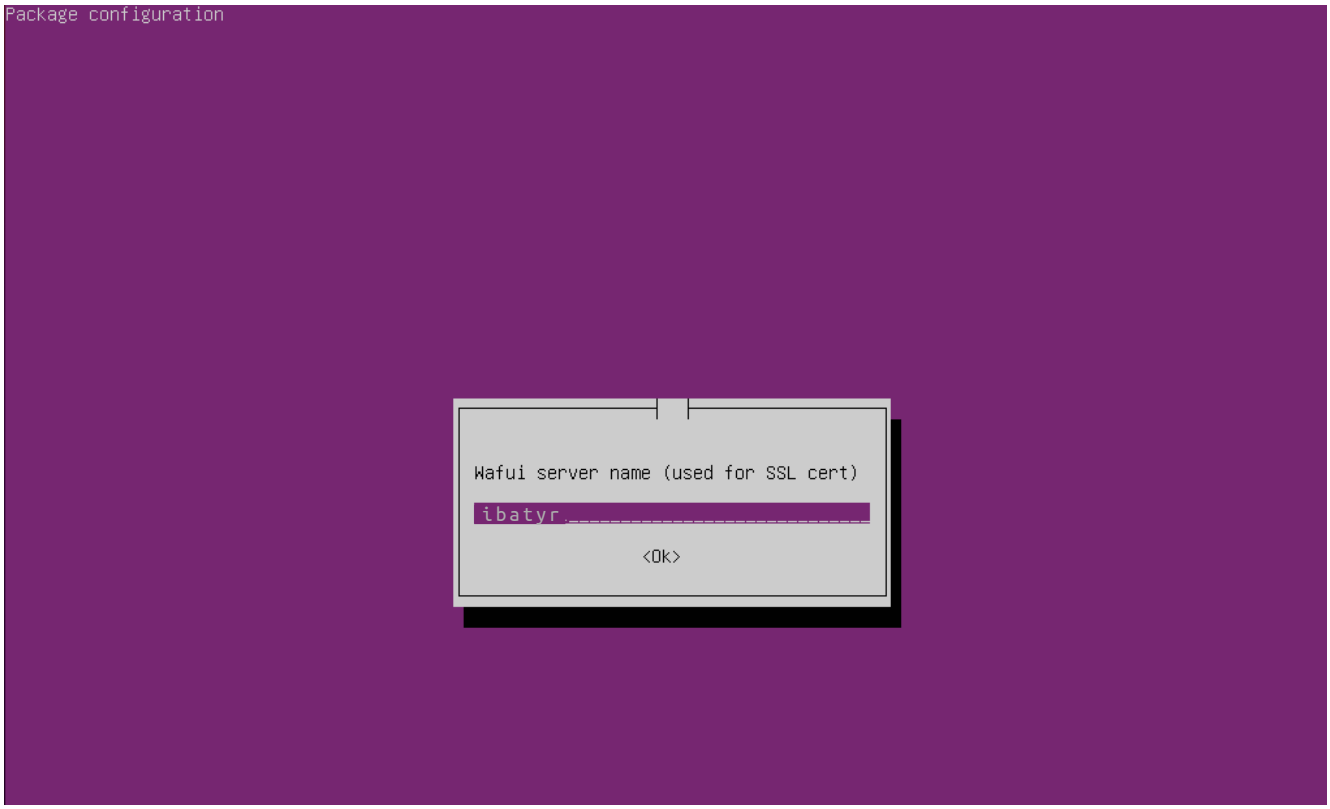
Снимок экрана 84. Генерация пароля PostgreSQL

9. В следующем появившемся окне выбрать автоматическую генерацию пароля `mongo` (см. снимок экрана 85).



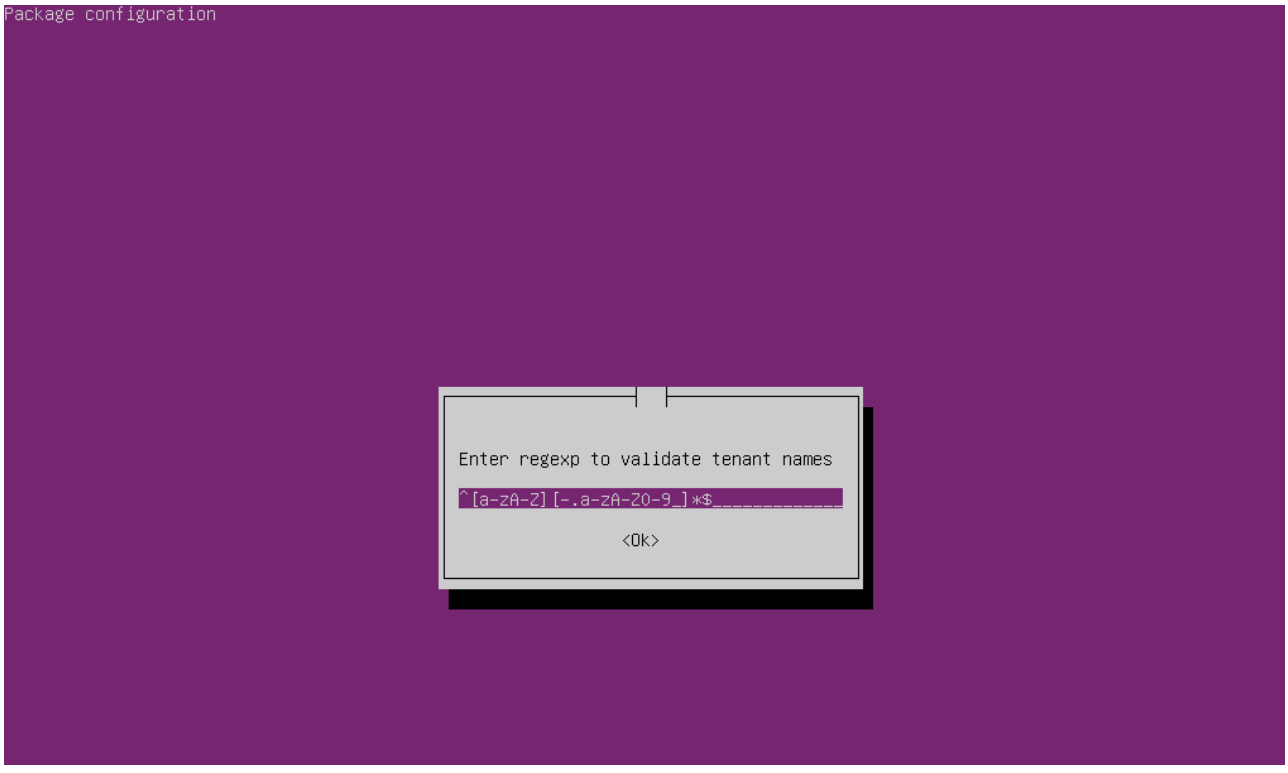
Снимок экрана 85 – Генерация пароля `mongo`

10. В следующих двух окнах возможно дублирование запросов данных из пунктов 6 и 7 настоящего раздела.
11. Далее нужно задать имя сервера `WebUI` (см. снимок экрана 86).



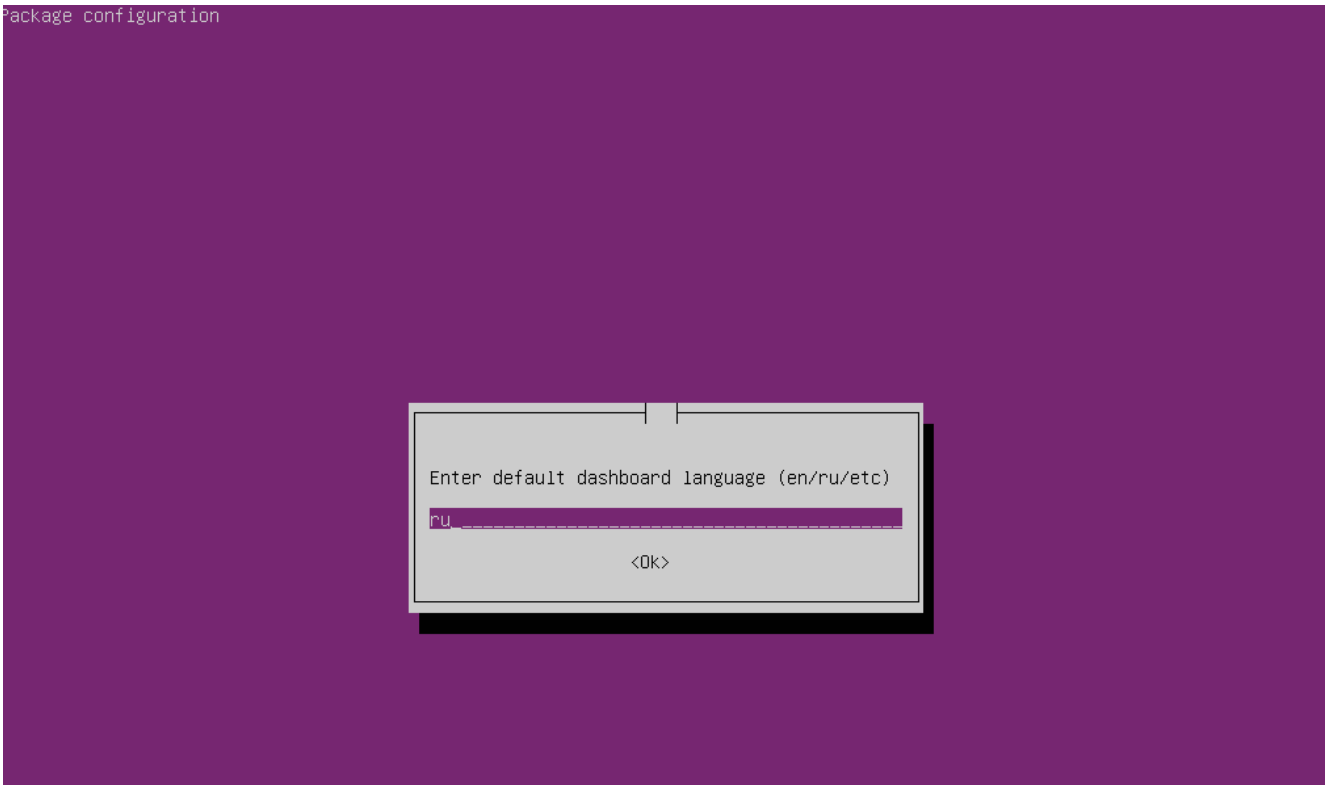
Снимок экрана 86 – Имя сервера `WebUI`

12. В следующем окне выбрать «Ok» (см. снимок экрана 87).



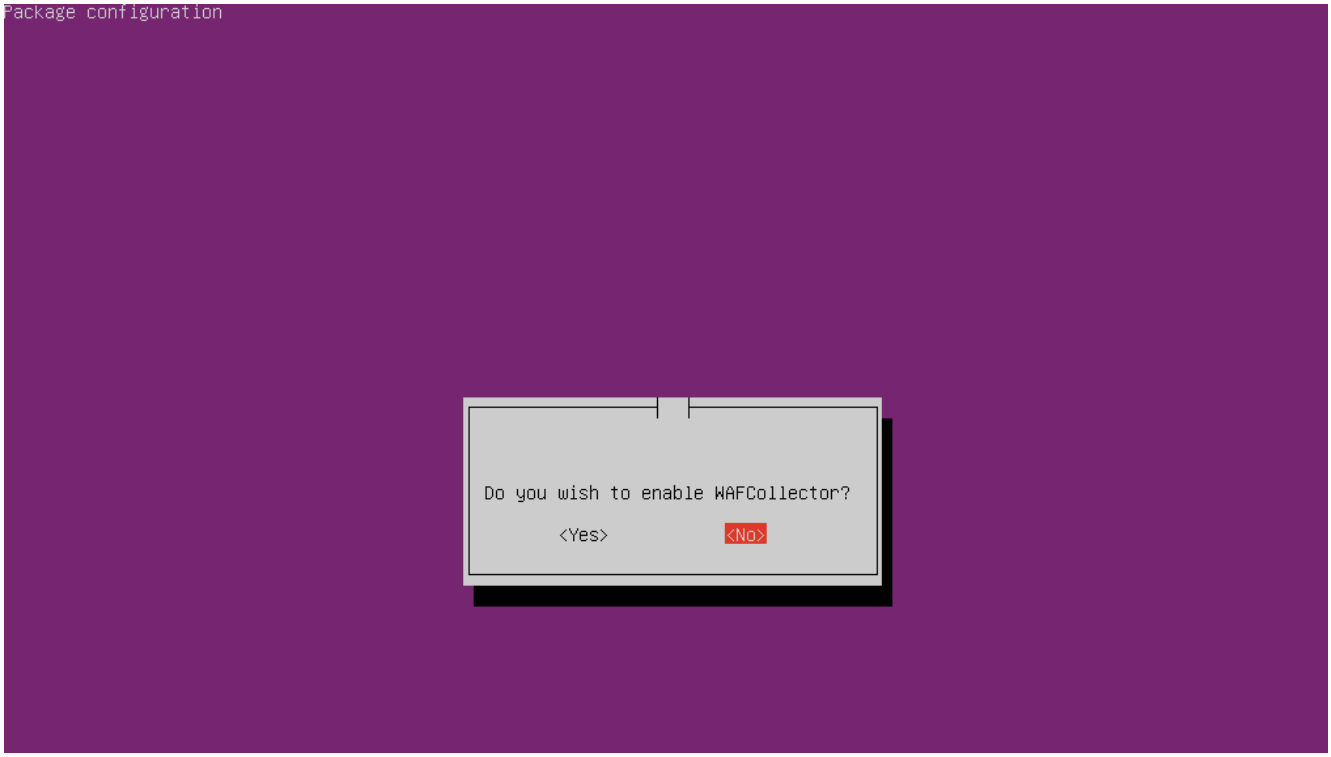
Снимок экрана 87 – Enter regexp to validate tenant names

13. Затем нужно выбрать язык дашборда по дефолту (см. снимок экрана 88).



Снимок экрана 88 – Язык дашборда

14. В следующем появившемся окне выбрать «No» - не включать `WAFCollector` (см. снимок экрана 89).

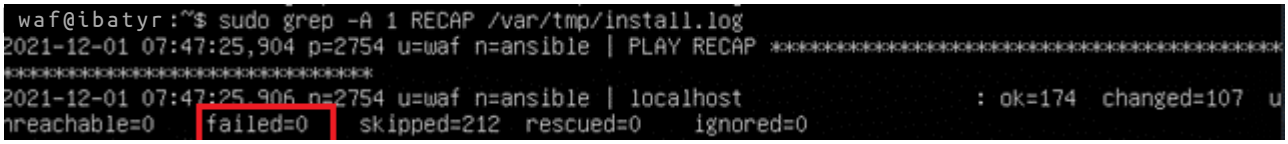


Снимок экрана 89 – WAFCollector

15. Далее вывести для проверки обзорную информацию по результатам установки с помощью команды:

```
sudo grep -A 1 RECAP /var/tmp/install.log
```

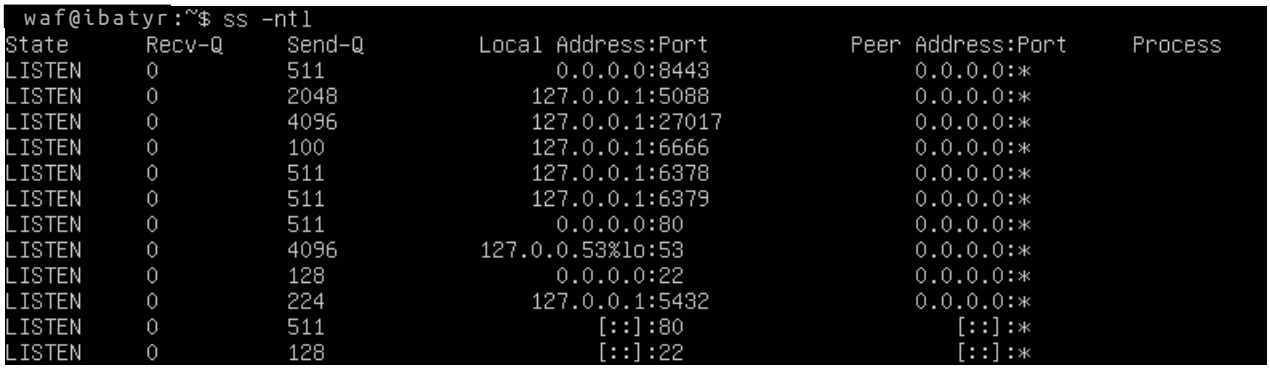
Индикатором успешной установки будет являться отсутствие ошибок установки «failed=0» (см. снимок экрана 90).



Снимок экрана 90 – Установка без ошибок

16. Затем для проверки соответствия портов сервисам необходимо воспользоваться командой (см. снимок экрана 91):

```
ss -ntl
```



Снимок экрана 91 – Соответствие портов сервисов

Порты для сервисов должны соответствовать:

- nginx: port 8443;
- ibatyr-dashboard: port 5088;
- mongod: port 27017;
- ibatyr-redis@mgmt: port 6378;
- ibatyr-redis@0: port 6379;
- sshd: port 22;
- postgresql: port 5432.

17. Проверить состояние кластера командой `sudo crm status` (см. снимок экрана 92).

```
Cluster Summary:
  * Stack: corosync
  * Current DC: waf-repl-node-1 (version 2.0.3-4b1f869f0f) - partition WITHOUT q
uorum
  * Last updated: Tue May 17 16:57:14 2022
  * Last change:  Tue May 17 16:52:38 2022 by root via crm_attribute on waf-repl
-node-1
  * 2 nodes configured
  * 16 resource instances configured

Node List:
  * Online: [ waf-repl-node-1 ]
  * OFFLINE: [ waf-repl-node-2 ]

Full List of Resources:
  * Resource Group: master:
    * waf-virtual-ip    (ocf::heartbeat:IPaddr2):        Started waf-repl-node-1
    * mongo-arbiter     (systemd:mongod-arbiter):        Started waf-repl-node-1
    *   ibatyr-celery  (systemd:   ibatyr-celery):      Started waf-repl-node-1
    *   ibatyr-celerybeat      (systemd:   ibatyr-celerybeat):  Started waf-rep
l-node-1
  * Clone Set: ReplicatedPg [pgsql] (promotable):
    * Masters: [ waf-repl-node-1 ]
```

Снимок экрана 92 – Результаты проверки состояния кластера

18. Инициализировать слот репликации путем перезапуска ресурса pgsql командой (см. снимок экрана 93):

```
sudo crm resource restart pgsql
```

```
waf@wafsrv:~/  ibatyr_distrib   $ sudo crm status
waf@wafsrv:~/  ibatyr_distrib   $ sudo crm resource restart pgsql
```

Снимок экрана 93 – Инициализация слота репликации

Во время выполнения данной команды ресурс сначала перейдет в статус Slaves (см. снимок экрана 94). Нужно дождаться, когда вернется в состояние Masters (см. снимок экрана 95). Проверка состояния кластера выполняется следующей командой:

```
sudo crm status
```

```
Node List:
  * Online: [ waf-repl-node-1 ]
  * OFFLINE: [ waf-repl-node-2 ]

Full List of Resources:
  * Resource Group: master:
    * waf-virtual-ip    (ocf::heartbeat:IPaddr2):        Started waf-repl-node-1
    * mongo-arbiter     (systemd:mongod-arbiter):        Started waf-repl-node-1
    *  ibatyr -celery  (systemd:  ibatyr -celery):      Started waf-repl-node-1
    *  ibatyr -celerybeat      (systemd:  ibatyr-celerybeat):  Started waf-rep
l-node-1
  * Clone Set: ReplicatedPg [pgsql] (promotable):
    * Slaves: [ waf-repl-node-1 ]
    * Stopped: [ waf-repl-node-2 ]
  * Clone Set: ReplicatedRedis [  ibatyr-redis-mgmt] (promotable):
    * Masters: [ waf-repl-node-1 ]
    * Stopped: [ waf-repl-node-2 ]
  * Clone Set: mongo-multi [mongo]:
    * Started: [ waf-repl-node-1 ]
    * Stopped: [ waf-repl-node-2 ]
  * Clone Set:  ibatyr -analyzer-clone [  ibatyr -analyzer]:
    * Started: [ waf-repl-node-1 ]
```

Снимок экрана 94 – Статус Slaves

```
Node List:
  * Online: [ waf-repl-node-1 ]
  * OFFLINE: [ waf-repl-node-2 ]

Full List of Resources:
  * Resource Group: master:
    * waf-virtual-ip    (ocf::heartbeat:IPaddr2):        Started waf-repl-node-1
    * mongo-arbiter     (systemd:mongod-arbiter):        Started waf-repl-node-1
    *   ibatyr-celery  (systemd:  ibatyr -celery):      Started waf-repl-node-1
    *   ibatyr-celerybeat      (systemd:  ibatyr -celerybeat):  Started waf-rep
l-node-1
  * Clone Set: ReplicatedPg [pgsql] (promotable):
    * Masters: [ waf-repl-node-1 ]
    * Stopped: [ waf-repl-node-2 ]
  * Clone Set: ReplicatedRedis [  ibatyr-redis-mgmt] (promotable):
    * Masters: [ waf-repl-node-1 ]
    * Stopped: [ waf-repl-node-2 ]
  * Clone Set: mongo-multi [mongo]:
    * Started: [ waf-repl-node-1 ]
    * Stopped: [ waf-repl-node-2 ]
  * Clone Set:  ibatyr -analyzer-clone [  ibatyr -analyzer]:
    * Started: [ waf-repl-node-1 ]
```

Снимок экрана 95 – Статус Master

Если выделенные адреса имеют маску не 255.255.255.0 (24), то необходимо изменить маску в конфигурационном файле для VIP-ресурса `waf-virtual-ip`. Для этого следует:

1. Выполнить команду `sudo crm configure edit`.
2. В открывшемся редакторе `vim`, в файле найти ресурс и сменить `cidr_netmask=24` на нужную маску (см. снимок экрана 96).

```
        op monitor timeout=30s interval=10s on-fail=restart \
        op stop timeout=60s interval=0
primitive   ibatyr -redis-mgmt redis \
        params config="/etc/  ibatyr /redis/mgmt.conf" rundir="/var/run/ibatyr
l/redis" pidfile_name=mgmt.pid socket_name=mgmt.sock port=6378 \
        op start timeout=120s interval=0s on-fail=restart \
        op promote timeout=120s interval=0s on-fail=restart \
        op stop timeout=120s interval=0s on-fail=block \
        op notify timeout=90s interval=0s \
        op monitor timeout=120s interval=90s on-fail=restart \
        op monitor timeout=120s interval=80s on-fail=restart role=Master \
        op demote timeout=120s interval=0s on-fail=stop
primitive waf-virtual-ip IPaddr2 \
        params ip=192.168.0.108 cidr_netmask=24 iflabel=waf0 \
        op start timeout=60s interval=0s on-fail=stop \
        op stop timeout=60s interval=0s on-fail=block \
        op monitor timeout=60s interval=10s on-fail=restart
```

Снимок экрана 96 – Смена сетевой маски

Для безопасности во время установки был запрещен SSH доступ пользователю `waf`. Если необходимо разрешить такой доступ, то в конфигурационном файле `/etc/sshd/sshd_config` следует удалить последнюю строчку `DenyUsers waf` и перезапустить сервис с помощью команды (см. снимок экрана 97):

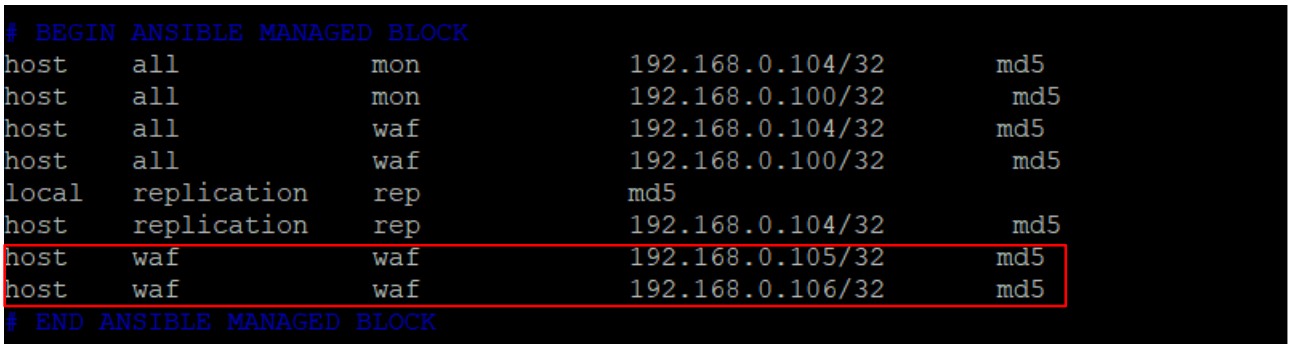
```
sudo systemctl restart sshd
```

```
# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp  /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
PasswordAuthentication yes
DenyUsers waf
```

Снимок экрана 97 – Разрешение SSH

3. В конфигурационном файле `/etc/postgresql/11/main/pg_hba.conf` добавить записи для подключения анализаторов к БД. Затем применить настройки доступа командой (см. снимок экрана 98):

```
sudo -u postgres psql -c 'SELECT pg_reload_conf();'
```



Снимок экрана 98 – Добавление записи для подключения анализаторов к БД

4. В файле `/etc/ibatyr/redis/0.conf` добавить код:

```
bind 0.0.0.0
requirepass f6f89916eb9881b5f0dc
masterauth  f6f89916eb9881b5f0dc
```

Пароли в примере выбраны случайным образом. Пароли можно сгенерировать следующей командой в консоли Линукс:

```
dd if=/dev/urandom bs=512 count=1 2>/dev/null | md5sum | awk '{printf $1}'
| fold -w20 | head -n1
```

Перезапуск сервиса осуществляется командой:

```
sudo systemctl restart ibatyr-redis@0
```

Примечание: В процессе установки в папке `/tmp` создаются два файла с расширением `conf`. Рекомендуется их сохранить, например, в домашней папке пользователя `waf`.

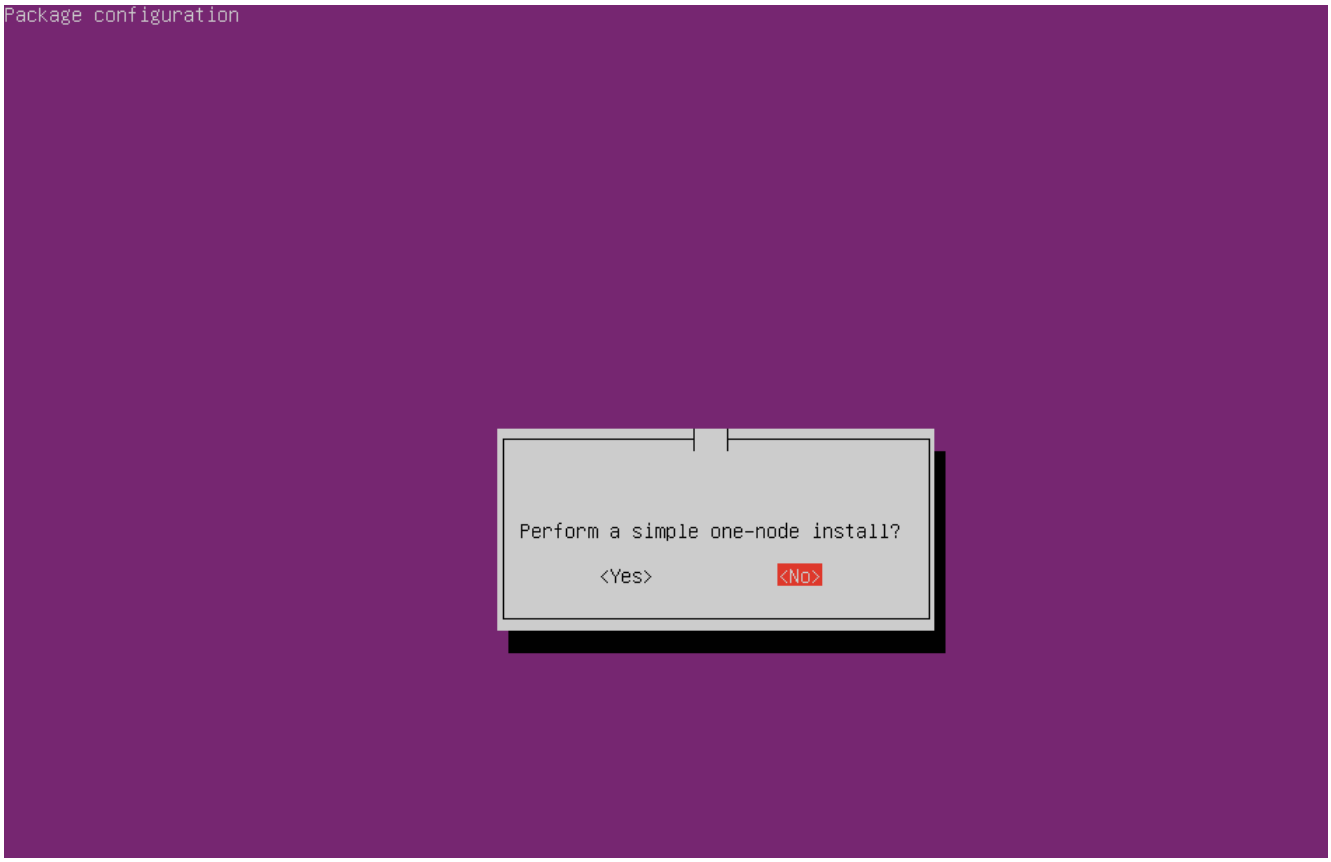
4.7.2. Установка резервного узла (slave) кластера БД

Для установки кластера на резервном узле следует:

1. На резервном узле кластера повторить шаг 1 из подраздела 4.6.
2. На шаге 2 указать вместо «Master» – «Slave» (см. снимок экрана 99).

Снимок экрана 99 **– Развёртывание резервного узла**

3. Повторить шаги с 3 по 7 из подраздела 4.6
4. Выбрать установку не на один узел (см. снимок экрана 100).



Снимок экрана 100 – Выбор установки

5. Повторить шаги с 3 по 15 из подраздела 4.7.1
6. Проверить состояние кластера командой `sudo crm status`. Оба узла должны быть в статусе Online (см. снимок экрана 101).

```
  * Online: [ waf-repl-node-1 waf-repl-node-2 ]

Full List of Resources:
  * Resource Group: master:
    * waf-virtual-ip    (ocf::heartbeat:IPaddr2):        Started waf-repl-node-1
    * mongo-arbiter     (systemd:mongod-arbiter):        Started waf-repl-node-1
    *   ibatyr-celery  (systemd:  ibatyr-celery):       Started waf-repl-node-1
    *   ibatyr-celerybeat      (systemd:  ibatyr-celerybeat):  Started waf-rep
l-node-1
  * Clone Set: ReplicatedPg [pgsql] (promotable):
    * Masters: [ waf-repl-node-1 ]
    * Slaves: [ waf-repl-node-2 ]
  * Clone Set: ReplicatedRedis [  ibatyr-redis-mgmt] (promotable):
    * Masters: [ waf-repl-node-1 ]
    * Slaves: [ waf-repl-node-2 ]
  * Clone Set: mongo-multi [mongo]:
    * Started: [ waf-repl-node-1 waf-repl-node-2 ]
  * Clone Set:   ibatyr-analyzer-clone [  ibatyr-analyzer]:
    * Started: [ waf-repl-node-1 waf-repl-node-2 ]
  * Clone Set:   ibatyr-nginx-clone [  ibatyr-nginx]:
    * Started: [ waf-repl-node-1 waf-repl-node-2 ]
  * Clone Set:   ibatyr-dashboard-clone [  ibatyr-dashboard]:
    * Started: [ waf-repl-node-1 waf-repl-node-2 ]
```

Снимок экрана 101 – Проверка состояния кластера

7. При необходимости в конфигурационном файле `/etc/sshd/sshd_config` удалить последнюю строчку `DenyUsers waf` и перезапустить сервис с помощью команды (см. снимок экрана 102):

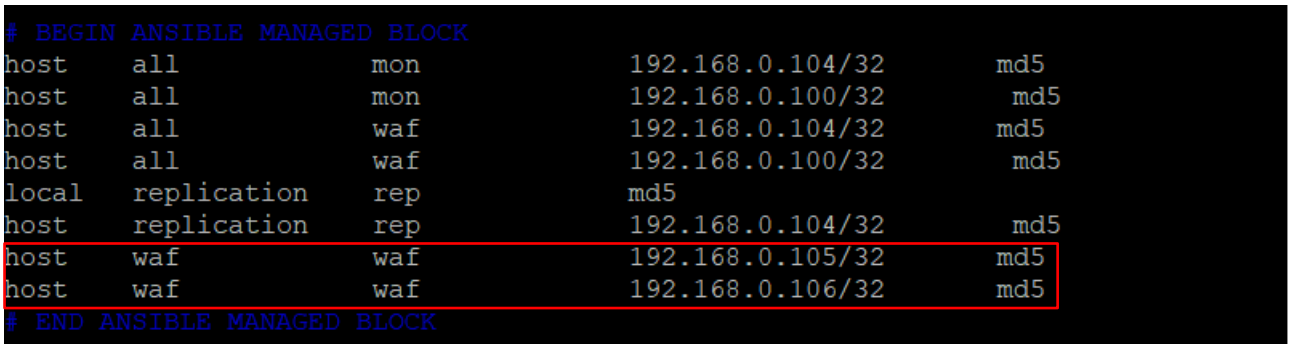
```
sudo systemctl restart sshd
```

```
# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp  /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
PasswordAuthentication yes
DenyUsers waf
```

Снимок экрана 102 – Перезапуск сервиса

8. В конфигурационном файле `/etc/postgresql/11/main/pg_hba.conf` добавить записи для подключения анализаторов к БД (см. снимок экрана 103). Затем применить настройки доступа командой:

```
sudo -u postgres psql -c 'SELECT pg_reload_conf();'
```



Снимок экрана 103 – Добавление записи для подключения анализаторов к БД

9. В файле `/etc/ibatyr/redis/0.conf` добавить код:

```
bind 0.0.0.0
requirepass f6f89916eb9881b5f0dc
masterauth  f6f89916eb9881b5f0dc
```

Пароль для `redis@0` должен быть такой же, как на master-узле.

Перезапуск сервиса осуществляется командой:

```
sudo systemctl restart ibatyr-redis@0
```

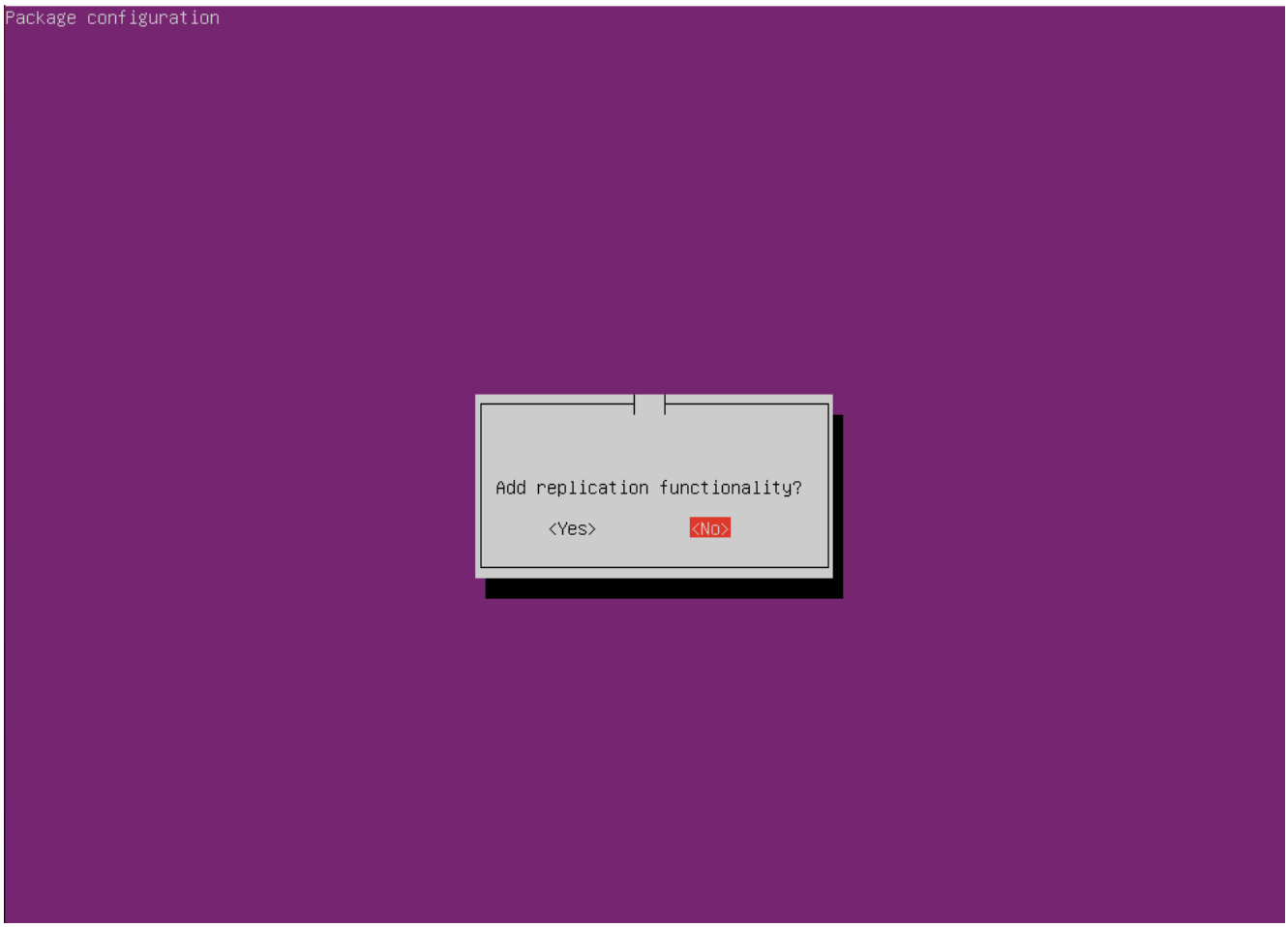
4.7.3. Установка узла Анализа

Установка осуществляется из образа `ibatyr-ubuntu-2X.04-v2.XX.0-replication.iso` (ОС+WAF) на виртуальный или физический сервер. Сервера, на которые планируется устанавливать iBatyr WAF, должны отвечать минимальным системным требованиям из раздела 2.1.

Для начала нужно установить ОС (пример установки ОС приведён в приложении Приложение Б).

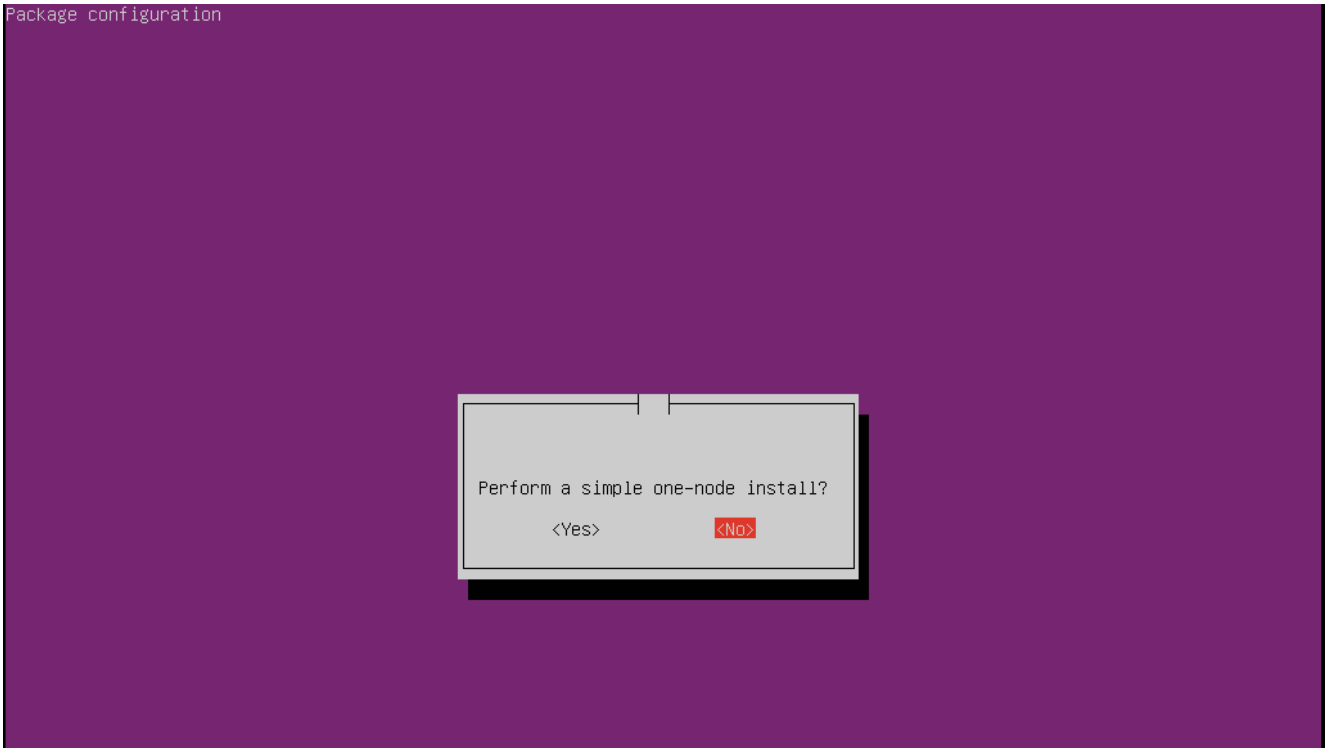
Далее необходимо произвести следующие действия:

1. В появившемся окне выбрать установку без репликации (см. снимок экрана 104).



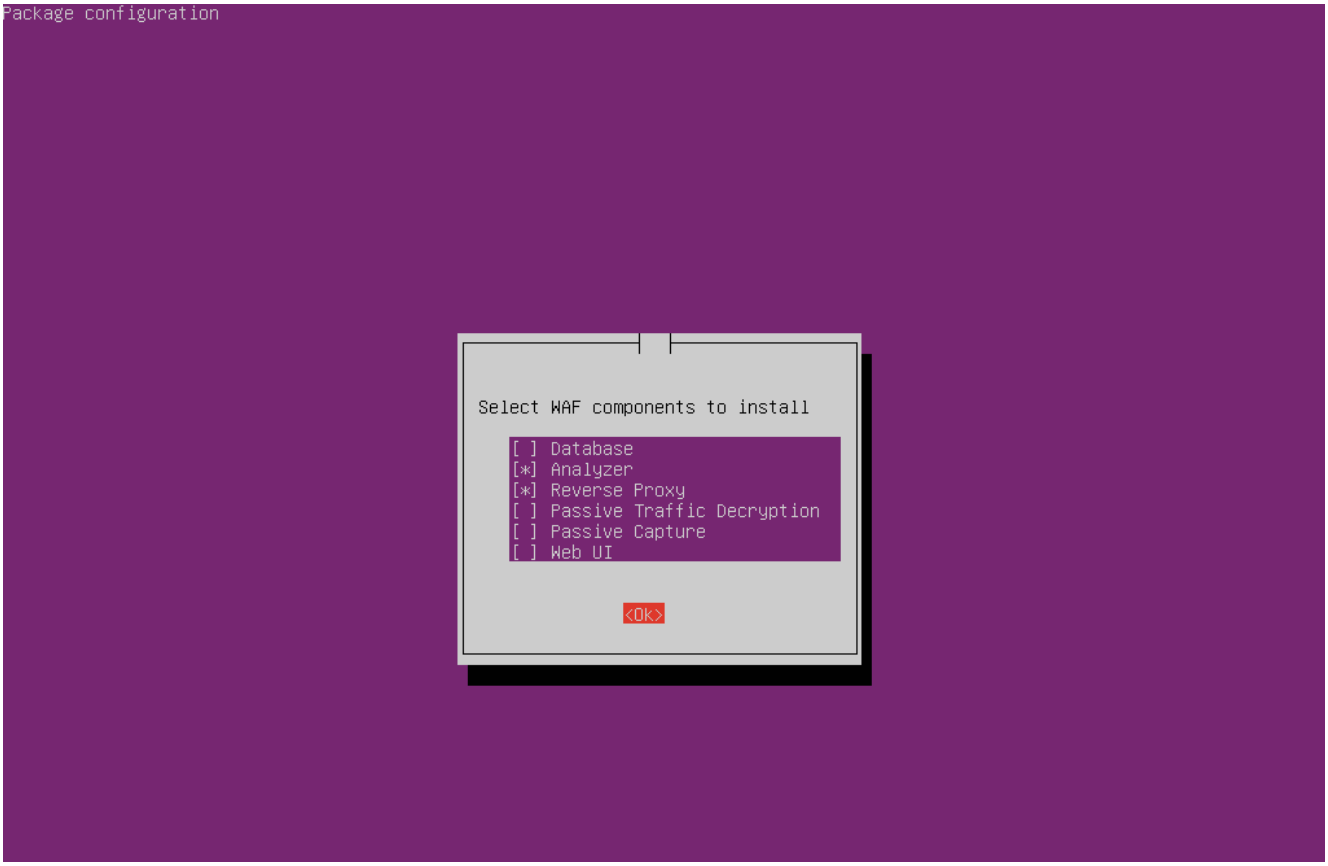
Снимок экрана 104 – Выбор типа установки

2. В следующем появившемся окне выбрать установку НЕ на один узел (см. снимок экрана 105).



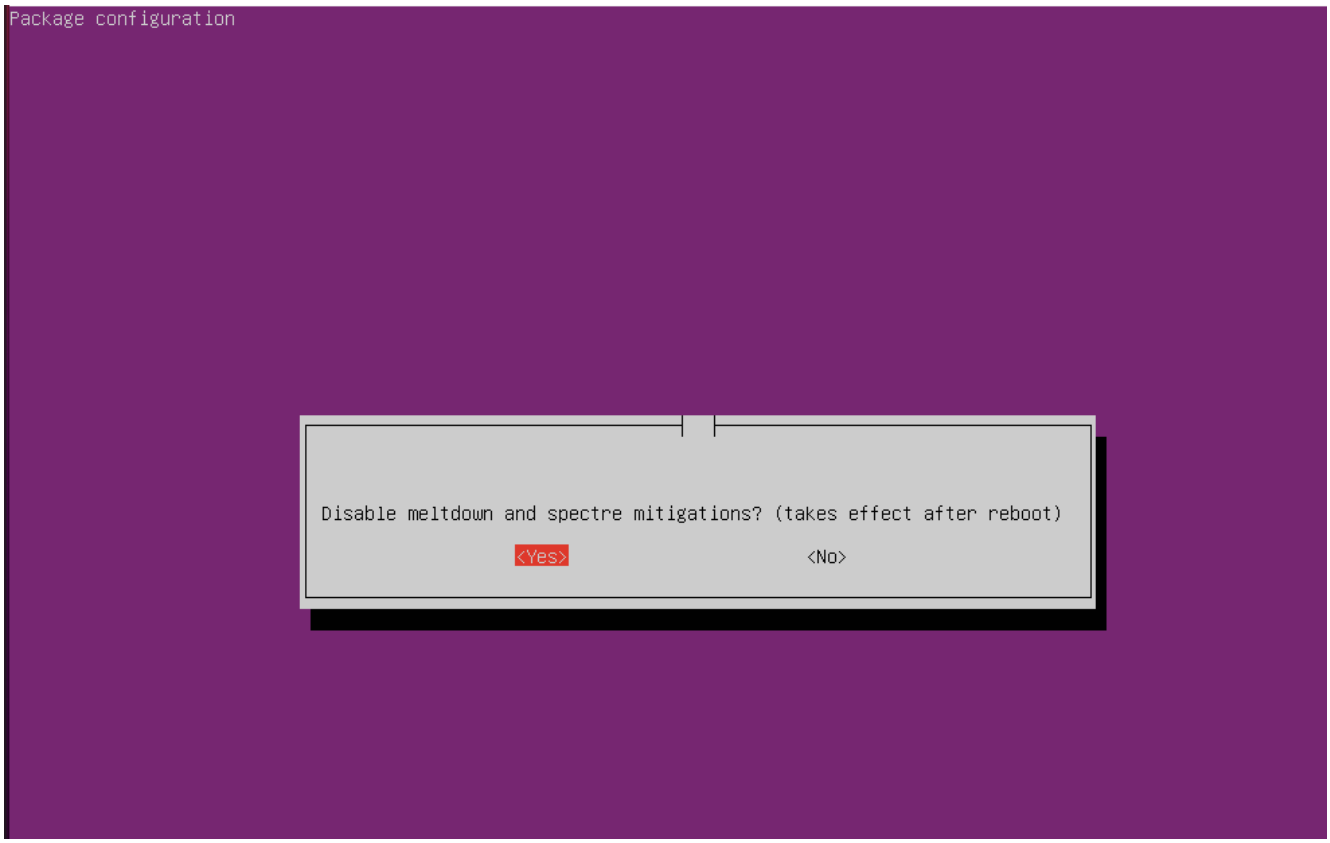
Снимок экрана 105 – Окно выбора установки НЕ на один узел

3. Далее нужно выбрать компоненты установки `Analyzer` и `Reverse Proxy` (см. снимок экрана 106).



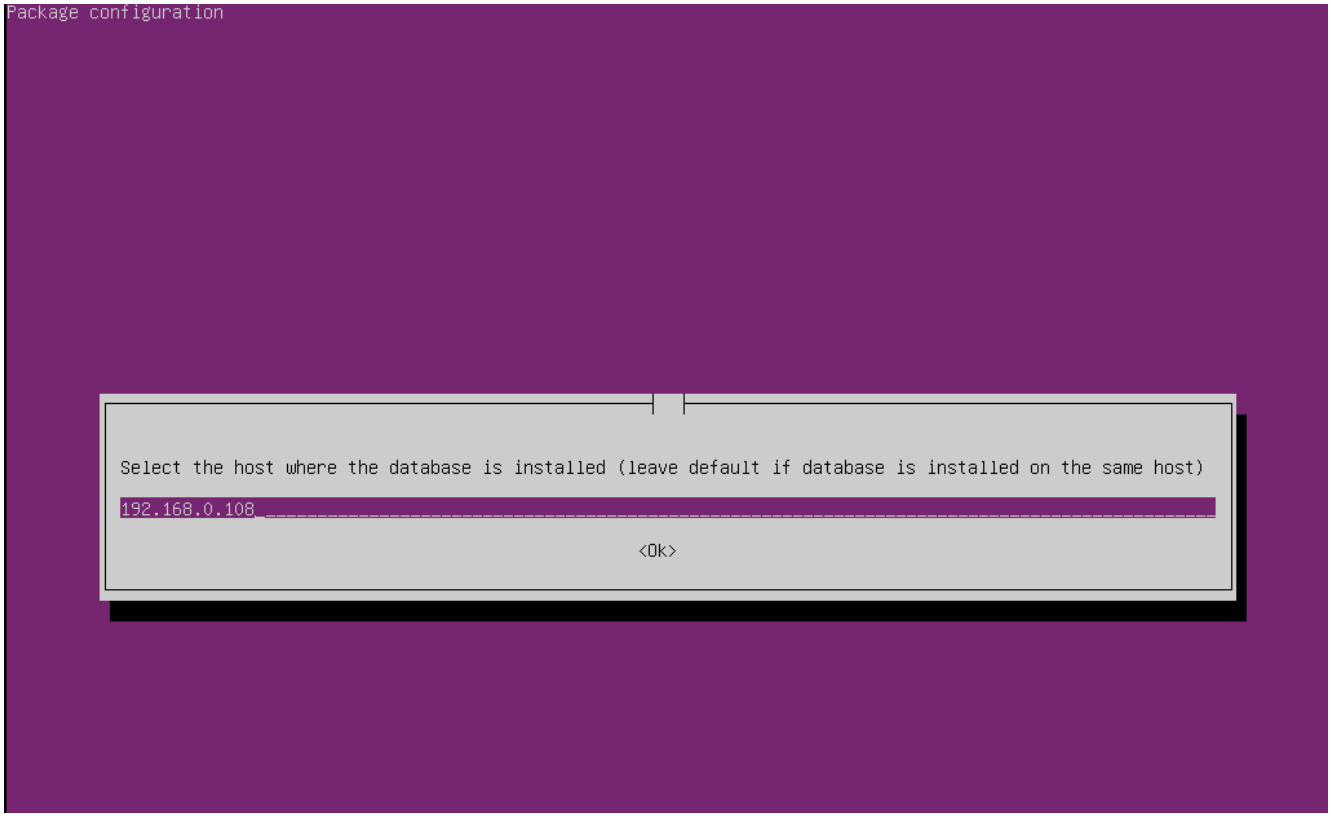
Снимок экрана 106 – Компоненты установки

4. В следующем появившемся окне рекомендуется отключить защиту от эксплуатации уязвимостей процессоров `meltdown` и `spectre`, т.к. это значительно снизит производительность узлов кластера и актуально только на многопользовательских облачных сервисах (см. снимок экрана 107).



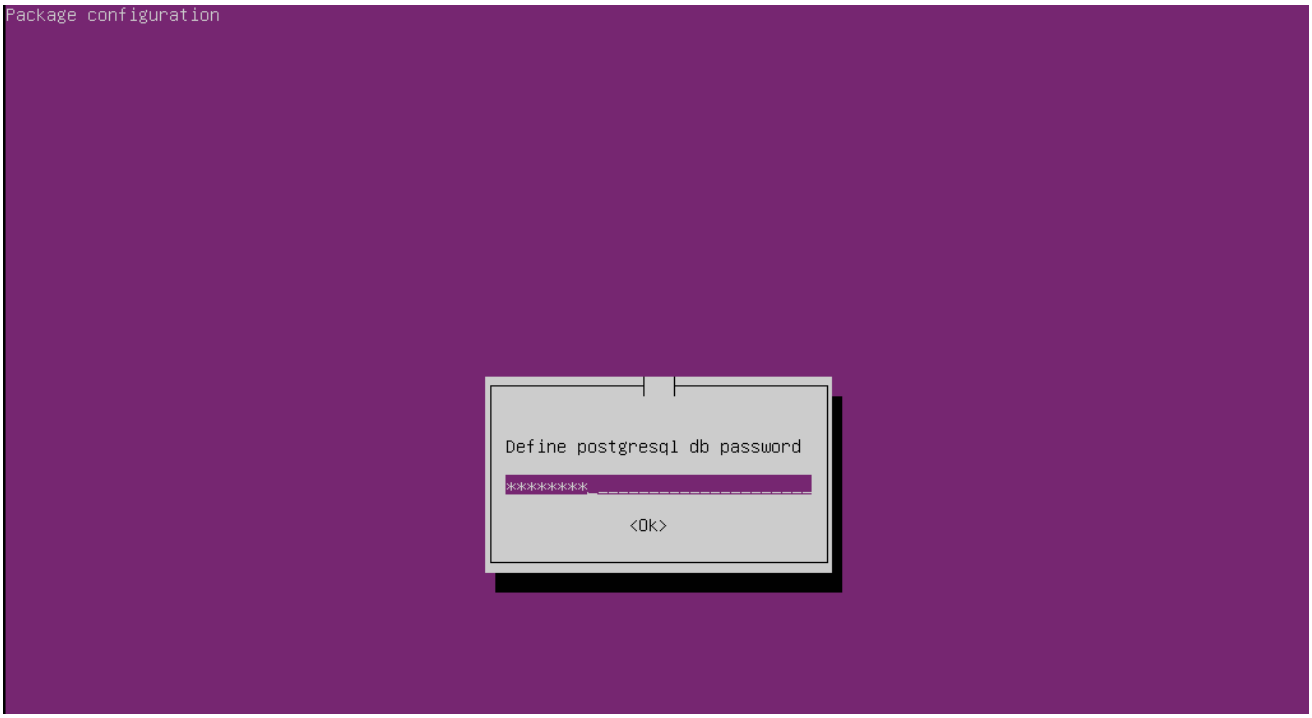
Снимок экрана 107 – Отключение `meltdown` and `spectre`

5. Затем ввести VIP-адрес кластера серверов с базой данных PostgreSQL (см. снимок экрана 108).



Снимок экрана 108 – VIP-адрес кластера серверов с базой данных PostgreSQL

6. Далее ввести пароль подключения к PostgreSQL кластера серверов (см. снимок экрана 109).



Снимок экрана 109 – Пароль PostgreSQL

7. В следующем появившемся окне подтвердить пароль подключения к PostgreSQL кластера серверов (см. снимок экрана 110).

Снимок экрана 110 – Подтверждение пароля PostgreSQL

8. В следующем появившемся окне ввести пароль подключения к базе данных `mongo` кластера серверов (см. снимок экрана 111).

Снимок экрана 111 – Пароль mongo

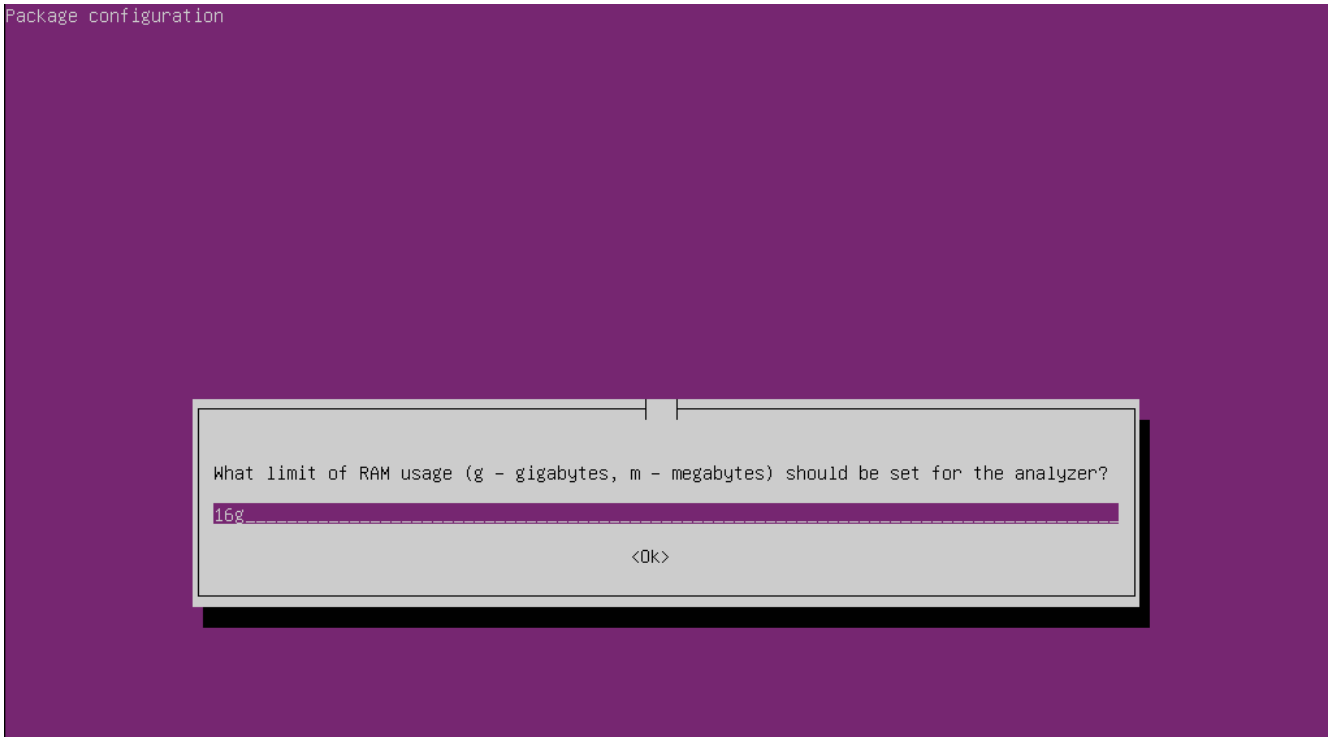
9. Затем подтвердить пароль подключения к базе данных `mongo` кластера серверов (см. снимок экрана 112).

Снимок экрана 112. Подтверждение пароля mongo

10. Далее ввести имя анализатора (см. снимок экрана 113).

Снимок экрана 113 – Имя анализатора

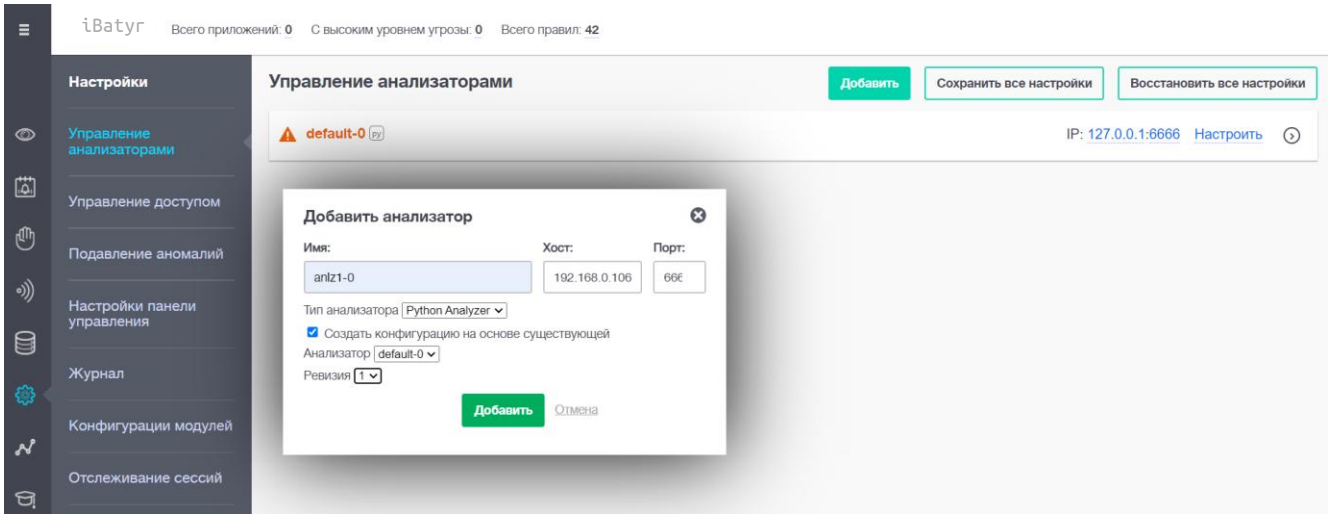
11. В следующем появившемся окне ограничить потребление памяти анализатором (см. снимок экрана 114). Для выделенного анализатора - 70% от объема всей ОЗУ. По умолчанию задано 16ГБ.



Снимок экрана 114 – Лимит памяти

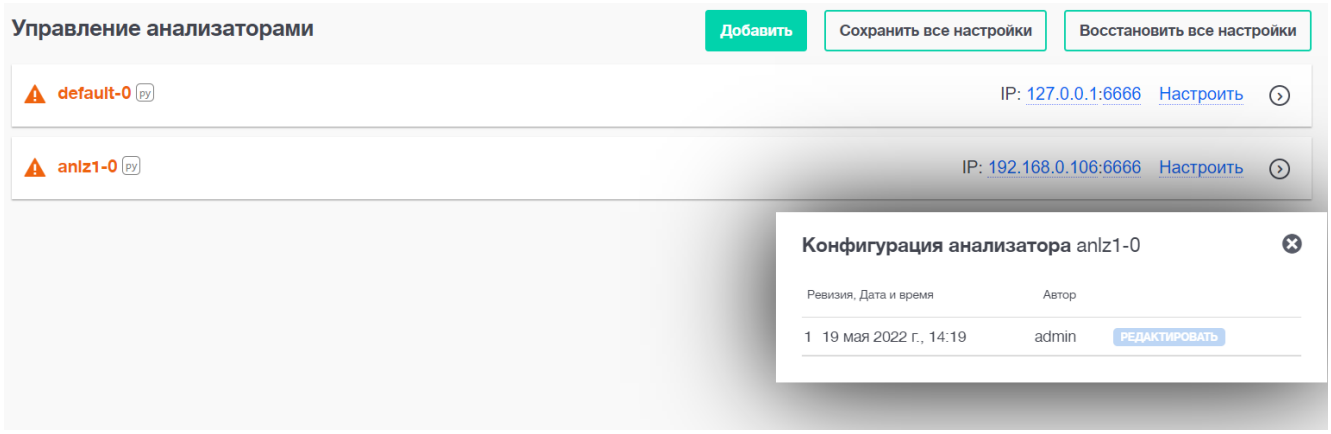
12. Повторить шаги с 3 по 5 из подраздела 4.2

13. В веб-интерфейсе перейти в «Настройки» -> «Управление анализаторами». Добавить новый анализатор. В настройках указать ранее заданное имя анализатора и добавить «-0», указать IP-адрес узла Анализа и порт 6666. Указать создание конфигурации, на основе данной и выбрать анализатор по умолчанию с последней доступной ревизией. (см. снимок экрана 115).



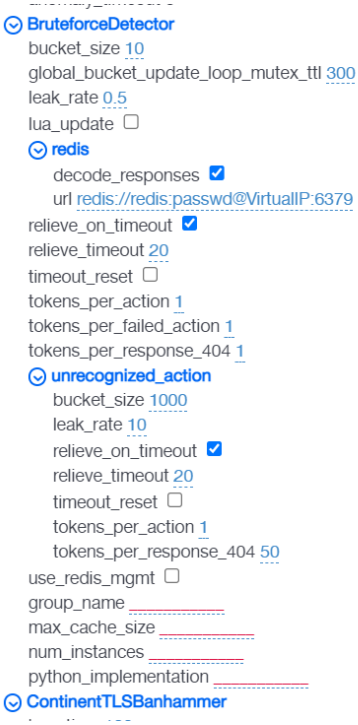
Снимок экрана 115 – Управление анализаторами

14. Отредактировать настройки добавленного анализатора (см. снимок экрана 116).



Снимок экрана 116 – Редактирование настроек добавленного анализатора

15. Привести данные настройки в соответствие со снимками экрана 117, 118, 119.

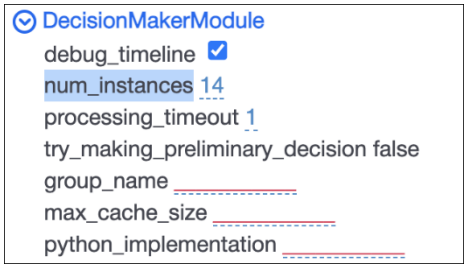


Снимок экрана 117 – Настройки 1

Снимок экрана 118 – Настроки 2

Снимок экрана 119 – Настройки 3

16. Задать в настройках анализатора количество пайплайнов равное суммарному количеству ядер ЦПУ на узле анализа – 2. Например, для 16 ядер это будет 14 пайплайнов. Для этого в настройках `DecisionMakerModule` необходимо задать значение `num_instances` равное нужному количеству пайплайнов (см. снимок экрана 120).

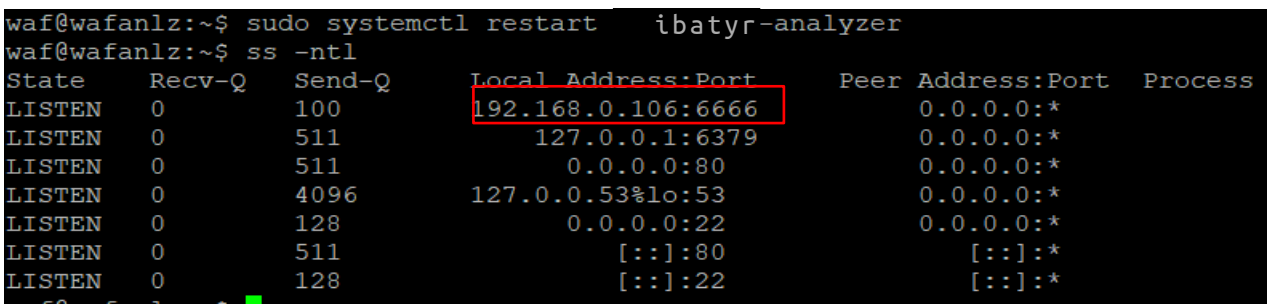


Снимок экрана 120 – Настройка `DecisionMakerModule`

17. Удалить в веб-интерфейсе старый анализатор по умолчанию (default).
18. В терминале анализатора перезапустить службу анализатора командой:

```
sudo systemctl restart ibatyr-analyzer
```

19. Проверить открытые порты. Должен подняться порт 6666 (см. снимок экрана 121).



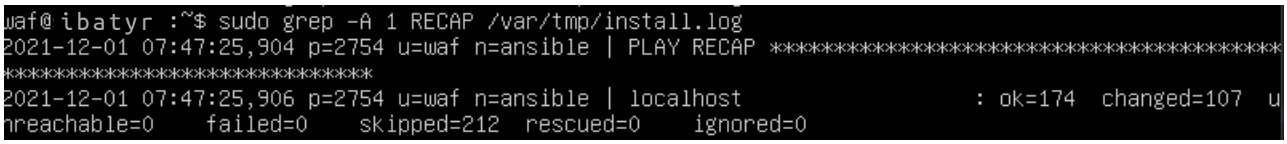
Снимок экрана 121 – Проверка портов

5. Способы контроля корректности установки

5.1. Проверка успешности процесса установки

Для проверки успешности процесса установки необходимо вывести обзорную информацию по результатам установки с помощью команды (см. снимок экрана 122):

```
sudo grep -A 1 RECAP /var/tmp/install.log
```



Снимок экрана 122 – Проверка успешности установки

Индикатором успешной установки будет являться отсутствие ошибок установки (failed=0).

Примечание: Все команды в данном разделе должны выполняться от имени суперпользователя (root).

5.2. Проверка сервисов

Для того, чтобы убедиться в успешной установке сервисов, первым делом нужно проверить их статус и порты, на которых они работают.

Для проверки соответствия портов сервисам необходимо воспользоваться командой:

```
ss -ntl
```

Порты для сервисов должны соответствовать:

- nginx: port 8443;
- ibatyr-dashboard: port 5088;
- mongod: port 27017;
- ibatyr-analyzer@0: port 6666;
- ibatyr-redis@mgmt: port 6378;
- ibatyr-redis@0: port 6379;
- ibatyr-nginx: port 80/443;
- sshd: port 22;
- postgresql: port 5432.

Для проверки статуса нужно выполнить команду `systemctl status service_name`. В случае если сервис недоступен или находится в выключенном состоянии, нужно выполнить от имени `root` команду `systemctl restart service_name`.

5.3. Проверка сетевой связности

В ходе диагностики неполадок может потребоваться проверка сетевой доступности служб WAF. В таблице 2 представлены TCP порты, используемые службами WAF.

Таблица 2 – TCP порты

Источник	Служба источника	Порт	Назначение	Служба назначения	Комментарий
Анализатор	ibatyr-analyzer@0	tcp6379	Управление	ibatyr-redis@0	BruteforceDetector
Анализатор	ibatyr-analyzer@0	tcp5432	Управление	postgresql	TX data save
Анализатор	ibatyr-analyzer@0	tcp27017	Управление	mongodb	get settings
Управление	ibatyr-dashboard	tcp6666	Анализатор	ibatyr-analyzer@0	restart, status
Балансировщик нагрузки		tcp80, tcp443	Анализатор	ibatyr-nginx	http(s)
Анализатор	ibatyr-nginx	tcp80, tcp443	Защищаемые приложения		http(s)

5.4. Проверка веб-интерфейса

Для проверки доступности веб-интерфейса необходимо убедиться, что запущены сервисы ibatyr-dashboard и nginx. Для доступа к веб-интерфейсу необходимо ввести в адресную строку веб-браузера адрес WAF с указанием порта (например, https://192.168.0.108:8443) и пройти процесс аутентификации, после чего пользователь будет перемещен на страницу с дашбордами.

Логин/пароль по умолчанию: **admin/p0wnp0wnp0wn**

5.5. Возможные проблемы и пути решения

В ходе установки могут возникнуть следующие проблемы:

- Установка завершилась ошибкой.
- Не запускается один из сервисов.
- Ошибка настройки интерфейсов.
- Веб-интерфейс недоступен по указанному порту.

5.5.1. Завершение установки с ошибкой

В случае завершения установки ошибкой необходимо прислать в службу поддержки журнал установки `/var/tmp/install.log`.

5.5.2. Не запускаются сервисы

При возникновении ошибок запуска сервисов необходимо проверить журналы незапустившихся сервисов:

- Для сервисов `ibatyr-celery`, `ibatyr-celerybeat` и `ibatyr-dashboard` журналы хранятся в директории `/var/log/waf/`.
- Для сервисов базы данных PostgreSQL, `mongod` журналы хранятся в директориях `/var/log/service_name/`.
- Для сервиса `ibatyr-redis` журнал находится в директории `/var/log/ibatyr/redis/`.

Помимо этого, журнал для `ibatyr-nginx` и `ibatyr-analyzer` можно получить путем выполнения команды от сервиса `journald`:

```
journalctl -xeu ibatyr-nginx
journalctl -xeu ibatyr-analyzer@0
```

После получение журнала попытаться решить проблему, если она указана в явном виде или же обратиться в техническую поддержку.

Примечание: Для более подробной информации по фильтрам команд следует обращаться к официальному руководству ОС.

5.5.3. Ошибка настройки интерфейсов

Если один или несколько интерфейсов не настроены или не принимают соединения нужно убедиться в корректности сетевых настроек. Файл конфигурации находится в `/etc/netplan/00-installer-config.yaml`.

Если происходит блокировка портов на уровне хоста нужно проверить настройки локального межсетевого экрана (ufw, iptables). По умолчанию межсетевой экран отключен.

Примечание: Для более подробной информации по настройкам сети и файрвола следует обращаться к официальному руководству ОС.

5.5.4. Веб-интерфейс недоступен по указанному порту

Если веб-интерфейс недоступен, необходимо убедиться, что запущены службы `ibatyr-dashboard` и `nginx`. Для этого можно использовать команду:

```
sudo systemctl status nginx
sudo systemctl status ibatyr-dashboard
```

Если сервисы не запущены, следует попробовать перезапустить сервис и проверить журнал на наличие ошибок (см. раздел 5.5.2).

6. Настройки сервисов

Все настройки, описанные в данном разделе, выполняются после установки iBatyr WAF для оптимизации работы.

6.1. Настройки параметров Postgres

Рекомендуемая конфигурация PostgreSQL для узла управления. Изменение настроек требуют рестарта службы PostgreSQL. Для отказоустойчивого кластера перезагрузка осуществляется средствами `crm`. Добавить/изменить следующие настройки в файле `/etc/postgresql/14/main/postgresql.conf`. Настройки приведены в качестве примера для узла управления с 16 Гб ОЗУ.

```
timescaledb.telemetry_level=off
max_connections = 2048
shared_buffers = 4GB # Необходимо задать здесь ¼ от общего объема опера-
тивной памяти на узле.
effective_cache_size = 4GB
maintenance_work_mem = 128MB
checkpoint_completion_target = 0.9
default_statistics_target = 100
random_page_cost = 1.1
effective_io_concurrency = 300 # Только для диска ssd, nvme
work_mem = 32MB
min_wal_size = 1GB
max_wal_size = 4GB
checkpoint_timeout = 30min
checkpoint_warning = 25min
synchronous_commit = off

bgwriter_delay = 10ms
bgwriter_lru_maxpages = 900
bgwriter_lru_multiplier = 9.0

vacuum_cost_delay = 0
vacuum_cost_page_hit = 0
vacuum_cost_page_miss = 5
vacuum_cost_page_dirty = 5
vacuum_cost_limit = 2000
vacuum_freeze_min_age = 20000000
vacuum_freeze_table_age = 900000000
vacuum_multixact_freeze_min_age = 20000000
vacuum_multixact_freeze_table_age = 900000000
log_autovacuum_min_duration = 1000000
autovacuum_max_workers = 6
autovacuum_naptime = 1s
autovacuum_vacuum_threshold = 500
autovacuum_analyze_threshold = 5000
autovacuum_vacuum_scale_factor = 0
autovacuum_analyze_scale_factor = 0
autovacuum_freeze_max_age = 1000000000
autovacuum_multixact_freeze_max_age = 1000000000
autovacuum_vacuum_cost_delay = 0
autovacuum_vacuum_cost_limit = -1

idle_in_transaction_session_timeout = 600000 # ms

max_locks_per_transaction = 400
```

6.2. Настройки параметров службы MongoDB

На распределенных сетапах с центральным узлом менеджмента иногда возникает проблема с большим числом коннектов к MongoDB, в результате чего MongoDB может отказывать в обслуживании некоторым процессам. Чтобы этого избежать, надо MongoDB предоставить возможность порождать много тредов (системные настройки ядра Linux) и поменять настройки по умолчанию в самой MongoDB .

6.2.1. Оптимизация настроек ядра для MongoDB

Все изменения настроек ядра выполняется через `/etc/sysctl.conf`. Применять настройки можно с помощью команды `sudo sysctl -p`.

```
vm.max_map_count = 999999 # количество регионов памяти
kernel.pid_max = 999999 # пространство pid
fs.file-max = 20000000 # количество файловых дескрипторов
```

6.2.2. Настройки MongoDB

Все изменения настроек выполняем через `/etc/mongod.conf` в секции `net`:

1. `maxIncomingConnections: 999999` – поднимаем лимит входящих соединений, по умолчанию он равен 64×10^3.
2. `serviceExecutor: adaptive` – меняем модель работы пула, по умолчанию MongoDB создаёт тред на каждый коннект, `adaptive` переключает на модель с воркерами (один тред обслуживает много коннектов).

6.2.3. Рекомендуемые настройки ядра для высоконагруженной инсталляции

`/etc/sysctl.conf`:

```
vm.swappiness = 1
vm.overcommit_memory=1
vm.dirty_ratio=10
vm.dirty_background_ratio=3
kernel.sched_autogroup_enabled = 0
kernel.sched_migration_cost_ns = 5000000
net.netfilter.nf_conntrack_max=3000000
# net.ipv4.tcp_tw_reuse=1 ## Ставим только при рисках исчерпания коннектов
net.core.somaxconn=65535
net.ipv4.ip_local_port_range=10001 65535
```

Применить настройки можно командой `sudo sysctl -p&`.

6.3. Настройки параметров ibatyr-celery

Конфигурация настройки автоматических задач и машинного обучения доступна в файле `/etc/ibatyr/celery/config.yml`.

Настраиваемыми параметрами являются параметры хранения записей обычных (`days_to_keep_txs_normal`) и аномальных (`days_to_keep_txs_max`) транзакций в БД PostgreSQL:

```
remove_old_txs_and_anomalies:
```

```
    task:                   'scwaf_analyzer.common.db.db_cleanup.tasks.re-
move_old_txs_and_anomalies'
    schedule:    # crontab type
      minute: 30
      hour: 0
    kwargs:
      days_to_keep_txs_normal: 7
      days_to_keep_txs_max: 90
      days_to_keep_actions: 90
```

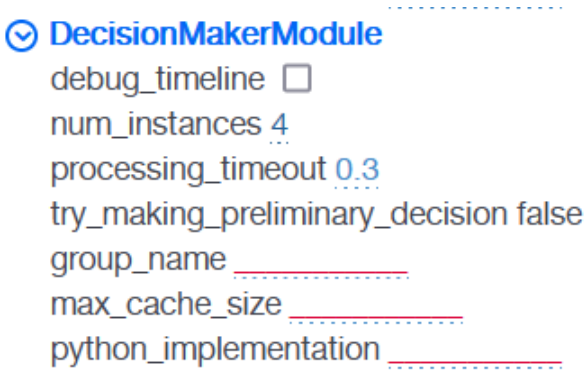
Проверка журнала на наличие ошибок.

```
tail -20 /var/log/waf/wafcelery-security_events.log
tail -20 /var/log/waf/wafcelerybeat.log
```

6.4. Настройки параметров ibatyr-analyzer

В настройке конфигурации сервиса (`/etc/default/ibatyr-analyzer`) задать верхнюю планку используемой памяти (MEMORY_LIMIT) для сервиса равную 70% от количества ОЗУ на узле Анализа. Например, для размера ОЗУ 32 Гб, выставить `MEMORY_LIMIT=22g`.

В веб-интерфейсе в настройках анализаторов задать количество воркеров (пайплайнов) равное количеству ядер ЦПУ минус 2 (по умолчанию выставлено 4, см. снимок экрана 123).



Снимок экрана 123 – Количество воркеров в настройках анализатора

6.5. Настройки параметров ibatyr-nginx

6.5.1. Настройка увеличенных таймаутов соединения с бекендами

Для настройки увеличенных таймаутов соединения с бекендами, нужно добавить в конфигурационный файл /etc/ibatyr-nginx/user/http.conf:

```
set $BACKEND_CONNECT_TIMEOUT 60s;
set $BACKEND_SEND_TIMEOUT 600s;
set $BACKEND_READ_TIMEOUT 600s;
```

6.5.2. Увеличение лимита загружаемых файлов в защищаемые приложения

Для настройки увеличенного лимита загружаемых файлов в защищаемые приложения нужно изменить директиву в конфигурационном файле `/etc/ibatyr-nginx/user/http.conf`:

```
client_max_body_size 1024m;
```

6.5.3. Настройка доверенных HTTP-прокси серверов для получения IP-источника запроса

Если перед WAF установлен HTTP-прокси сервер или сервера, нужно указать их адреса в качестве доверенных в `/etc/ibatyr-nginx/user/http.conf`, например:

```
set_real_ip_from 192.168.4.6;
set_real_ip_from 185.121.240.0/22;
```

6.5.4. Настройка параметров LRB-cache

LRB-cache (Least Resently Blocked Cache кэш блокировок) – механизм, позволяющий значительно экономить серверные ресурсы iBatyr WAF при большом количестве нелигитимных запросов с общими характеристиками источника в момент бот-атаки, DDoS-активности. Данный механизм является неотъемлемой частью iBatyr WAF и настраивается по умолчанию при развертывании.

После нескольких блокировок нелегитимных запросов (количество блокировок задается LRB_CACHE_BAN_AFTER) от одного источника с общими параметрами, задаваемыми IP, method, path, значение заголовка user-agent, данный набор параметров добавляется в LRB-cache. Все последующие запросы с данного источника блокируются автоматически без анализа на уровне `ibatyr-nginx` на несколько секунд (количество секунд задается параметром LRB_CACHE).

Как результат, это дает возможность справляться с атаками на исчерпание, сканированиями на уязвимости, фаззингом параметров какой-либо формы или функции защищаемого приложения и, в том числе, выдерживать атаки, которые по нагрузке могут кратно превышать пропускную способность iBatyr WAF в отношении легитимного трафика.

Настройка параметров LRB-cache выполняется в `/etc/ibatyr-nginx/user/server.conf`.

Дефолтные настройки приведены ниже:

```
set $LRB_CACHE_BAN_AFTER 5;
set $LRB_CACHE 300;
```

Примечание: При необходимости можно отключить LRB-cache, задав `LRB_CACHE 0;`.

6.5.5. Сброс LRB-кэша

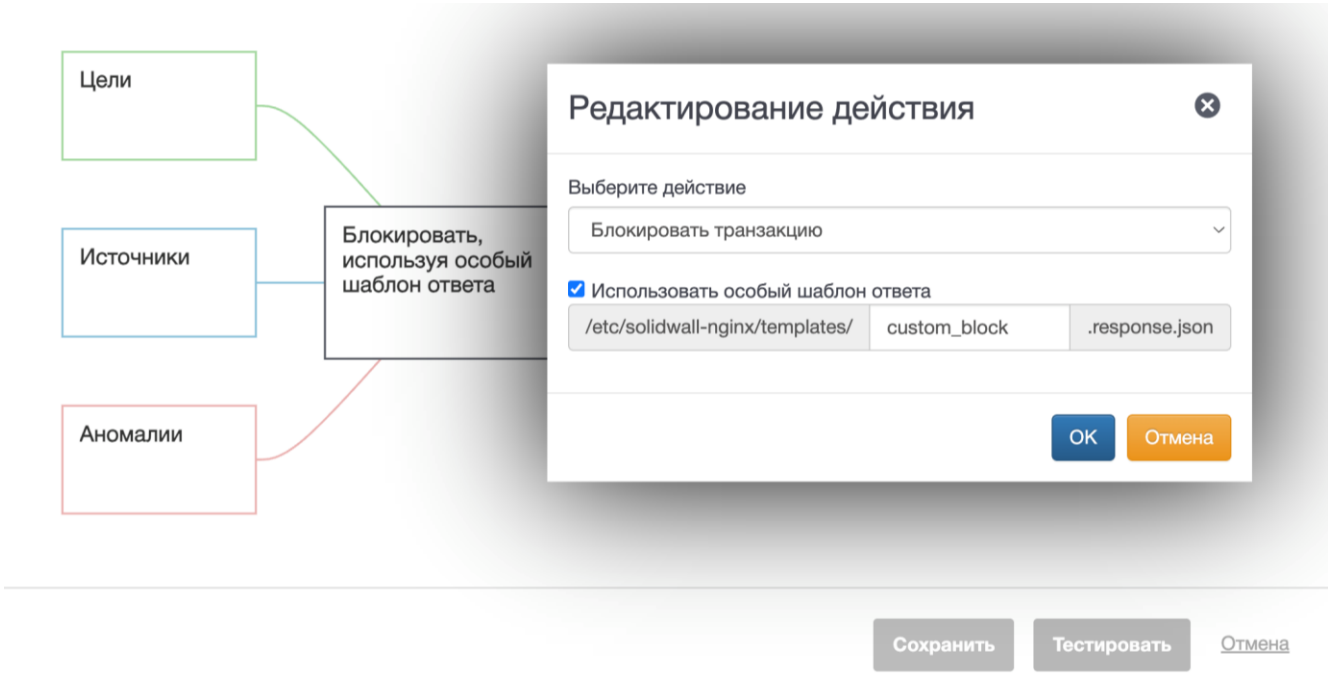
В случае, когда легитимный запрос попал в блок LRB-cache, подавление false positive такого запроса не сработает в течении времени, заданного в `$LRB_CACHE` (см. п. 6.5.4). Для немедленного сброса блокировки и применения подавления, необходимо сбросить LRB-cache ibatyr-nginx на анализаторах через API.

Для этого на каждом узле с `ibatyr-nginx` в командной строке нужно выполнить следующую команду.

```
curl http://127.0.0.1:19280/lrb-cache -X DELETE
```

6.5.6. Создание кастомных страниц блокировки

При блокировке транзакции правилом можно задать кастомную страницу с блокировкой. Для этого в действии нужного правила необходимо отметить опцию «Использовать особый шаблон ответа» и ввести в поле произвольное название файла, например, как показано на снимке экрана 124.



Снимок экрана 124 – Правило с особым шаблоном ответа

Далее необходимо на узлах с ролью Анализатор создать файл с данным шаблоном, например, `/etc/ibatyr-nginx/templates/custom_block.response.json` со следующим содержимым.

```
{
  "status": 401,
  "headers": {
    "Content-Type": "application/json"
  },
  "body": {
    "$apply": "json_encode",
    "value": {
      "message": "transaction blocked",
      "incident_id": {
        "$apply": "render",
        "string": "{* support_id *}"
      }
    }
  }
}
```

Для примера и демонстрации возможностей данного функционала ниже приведены еще примеры шаблонов ответа для кастомных страниц с блокировкой.

- `/etc/ibatyr-nginx/templates/default.response.json`:

```
{
  "status": {
    "$apply": "render",
    "string": "{* BLOCKED_STATUS_CODE or 403 *}"
  },
  "headers": {
    "Content-Type": {
      "$apply": "render",
      "string": "{* BLOCKED_CONTENT_TYPE or 'text/html' *}"
```

```
    }
  },
  "body": {
    "$apply": "render",
    "file": {
      "$apply": "render",
      "string": "{* BLOCKED_TEMPLATE or 'blocked.html' *}"
    }
  }
}
```

- `/etc/ibatyr-nginx/templates/example.response.json`:

```
{
  "status": 403,
  "headers": {
    "Content-Type": "application/json",
    "X-Multiple-Headers": ["one", "two", "three"],
    "X-The-Answer": {
      "$apply": "render",
      "string": "{* number + 1 *} is the answer",
      "context": {
        "number": 41
      }
    }
  },
  "body": {
    "$apply": "json_encode",
    "value": {
      "message": "You have been blocked.",
      "code": 1001,
      "incident_id": {
        "$apply": "render",
        "string": "{* support_id *}"
      },
      "time": {
        "$apply": "render",
        "string": "{* time_iso8601 *}"
      },
      "client_ip": {
        "$apply": "render",
        "string": "{* remote_addr *}"
      }
    }
  }
}
```

6.5.7. Установка и изменение значения заголовков запросов

Используя модуль `more_set_input_headers`, можно более гибко управлять заголовками запросов. Этот модуль позволяет устанавливать и изменять значения заголовков запросов.

Например, добавление заголовка x-request-waf со значением true

```
server {
    include static/server.conf;
    listen 80;
    server_name example.ru;
    set $BACKEND http://10.146.33.94:80;
```

```
    more_set_input_headers 'x-request-waf: true';
    location / {
        include static/waf.conf;
    }
}
```

Примечание: после добавления/редактирования конфигурационных файлов `ibatyr-nginx`, необходимо выполнить проверку конфигурации и применить настройки:

- `sudo ibatyr-nginx nginx test;`
- `sudo ibatyr-nginx nginx reload`.

7. Заведение приложений

7.1. Добавление конфигурационного файла NGINX

7.1.1. Настройка проксирования

Все файлы конфигурации должны быть созданы в каталоге `/etc/ibatyr-nginx/sites-available`.

Ключи и TLS-сертификаты нужно хранить в каталоге `/etc/ibatyr-nginx/ssl`.

Новый конфигурационный файл рекомендуется создавать по имени приложения.

Конфигурация файла состоит из блока\блоков server:

```
server {
    include static/server.conf;
    listen 80;
    server_name example.com www.example.com;
    # WAF configuration.
    set $BACKEND http://192.168.1.100;
    location / {
        include static/waf.conf;
    }
}
server {
    include static/server.conf;
    listen 443 ssl;
    ssl_certificate ssl/cert.pem;
    ssl_certificate_key ssl/key.pem;
    server_name example.com www.example.com;
    # WAF configuration.
    set $BACKEND https://192.168.1.100:443;
    location / {
        include static/waf.conf;
    }
}
```

Директива `listen` содержит параметры с номером порта, на который будет проксироваться бэкенд. Параметр `ssl` у данной директивы указывает на то, что трафик должен быть доступен по HTTPS.

Директивы `ssl_certificate*` в своих параметрах содержат пути к файлам с ssl-сертификатами и ключами. Это необходимо для того, чтобы терминировать зашифрованный TLS-трафик до его попадания на iBatyr WAF.

Директива **`server_name`** содержит доменные имена (заголовок Host), на которые будут идти запросы. Стоит отметить, что нужно указать все необходимые доменные имена, на которые они будут приходить. В противном случае, часть запросов может быть обработана некорректно.

В параметре директивы `set $BACKEND` передается полный адрес бэкенда с указанием протокола и номера порта (см. снимок экрана 125).

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Снимок экрана 125 – Результат успешной проверки конфигурации

7.1.2. Настройка проксирования для приложения с несколькими бекендами

Если у одного приложения несколько бекендов, между которыми трафик распределяется балансировщиком, то в этом случае для каждого бекенда создаются подобные блоки server. В каждом таком блоке в директиве `listen` указывается свой TCP-порт для каждого открытого порта на бэкенде. Ниже показан пример для приложения с двумя бэкендами.

На схеме 7 отражена балансировка без использования WAF.

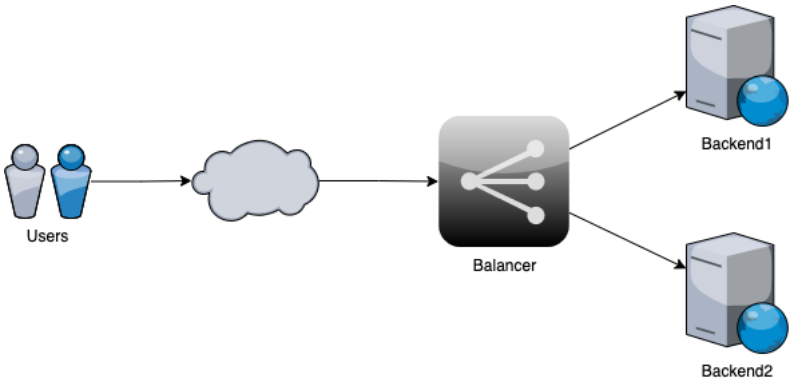


Схема 7 – Схема балансировки без WAF

Для внедрения WAF на каждом анализаторе открывается по порту на каждый бэкенд, и схема балансировки принимает вид, согласно примеру в таблице 3.

Таблица 3 – Схема балансировки

Балансировщик		WAF Analyzer		Backend
10.0.0.1	->	10.0.10.1:8081	->	10.0.100.1:80
	->	10.0.10.1:8082	->	10.0.100.2:80
	->	10.0.10.2:8081	->	10.0.100.1:80
	->	10.0.10.2:8082	->	10.0.100.2:80

Итоговая схема с балансировкой представлена на схеме 8.

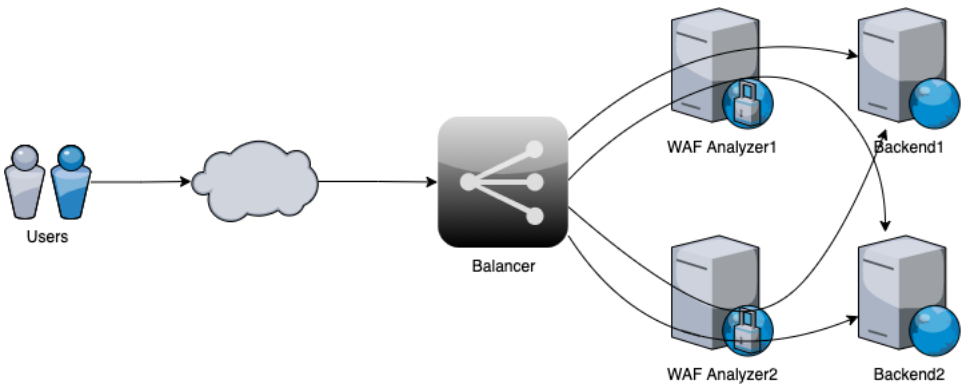


Схема 8 – Схема балансировки с WAF

Пример настроек проксирования на первом анализаторе WAF:

```
server {
    include static/server.conf;
    listen 10.0.10.1:8081;
    server_name example.com www.example.com;
    # WAF configuration.
    set $BACKEND http://10.0.100.1;
    location / {
        include static/waf.conf;
    }
}
server {
    include static/server.conf;
    listen 10.0.10.1:8082;
    server_name example.com www.example.com;
    # WAF configuration.
    set $BACKEND http://10.0.100.2;
    location / {
        include static/waf.conf;
    }
}
```

7.1.3. Применение конфигурации

После создания файла конфигурации на него необходимо создать символическую ссылку в каталог `/etc/ibatyr-nginx/sites-enabled/` для его активации. Например, для файла конфигурации `example.com` это можно сделать с помощью команды:

```
ln -s /etc/ibatyr-nginx/sites-available/example.com /etc/ibatyr-nginx/sites-enabled/example.com
```

Далее необходимо проверить корректность синтаксиса введенный изменений в конфигурацию командой:

```
ibatyr-nginx test
```

В случае если ответ не будет содержать ошибок, то можно применить конфигурацию. Стоит отметить, что данный способ перезагрузки конфигурации NGINX не приводит к разрыву текущих соединений и может быть применен на уже работающих в боевой среде узлах.

```
ibatyr-nginx reload
```

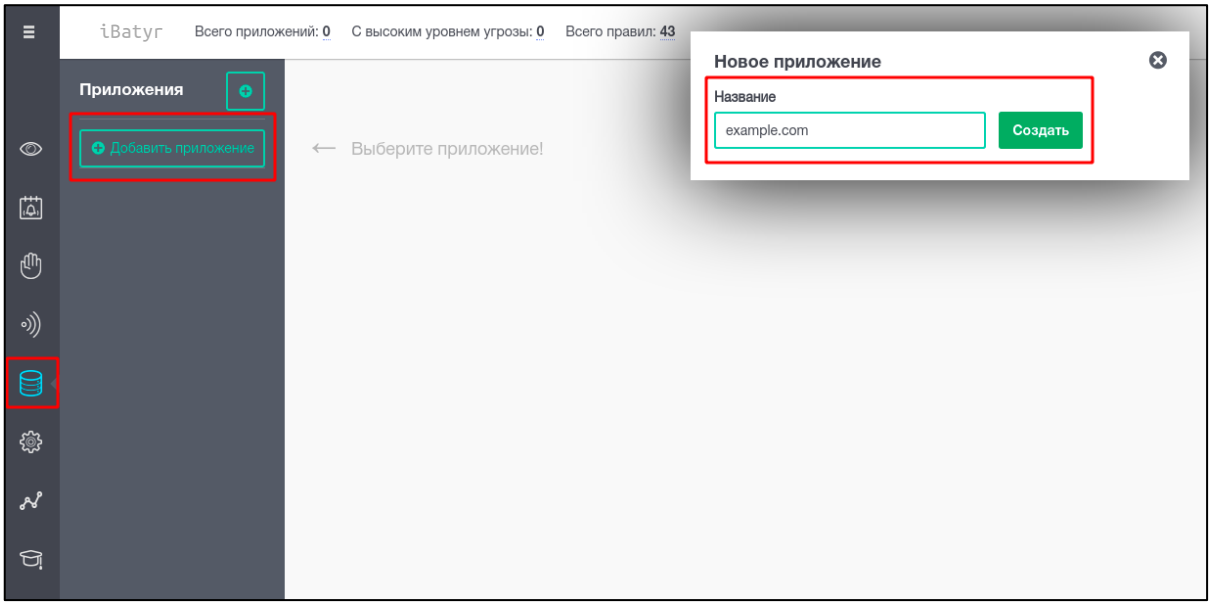
7.1.4. Проверка проксирования

Для проверки корректности настроенного проксирования желательно отправить запросы к защищаемому приложению напрямую, на бэкенд и через WAF. В обоих случаях ответы должны совпадать. Отправлять запросы можно любым удобным способом. Например, при помощи браузера, для чего необходимо настроить разрешение доменного имени проверяемого приложения в файле hosts операционной системы или при помощи утилиты curl, непосредственно в консоли сервера WAF.

7.2. Создание приложения

После того, как выполнены настройки проксирования, необходимо завести само приложение в веб-интерфейсе iBatyr WAF.

Для этого нужно зайти на веб-интерфейс iBatyr WAF по адресу https://ip_waf/, авторизоваться, после чего выбрать в левом навигационном меню «Приложения» -> «Добавить приложение». Далее в появившемся окне нужно указать название приложения (см. снимок экрана 125).



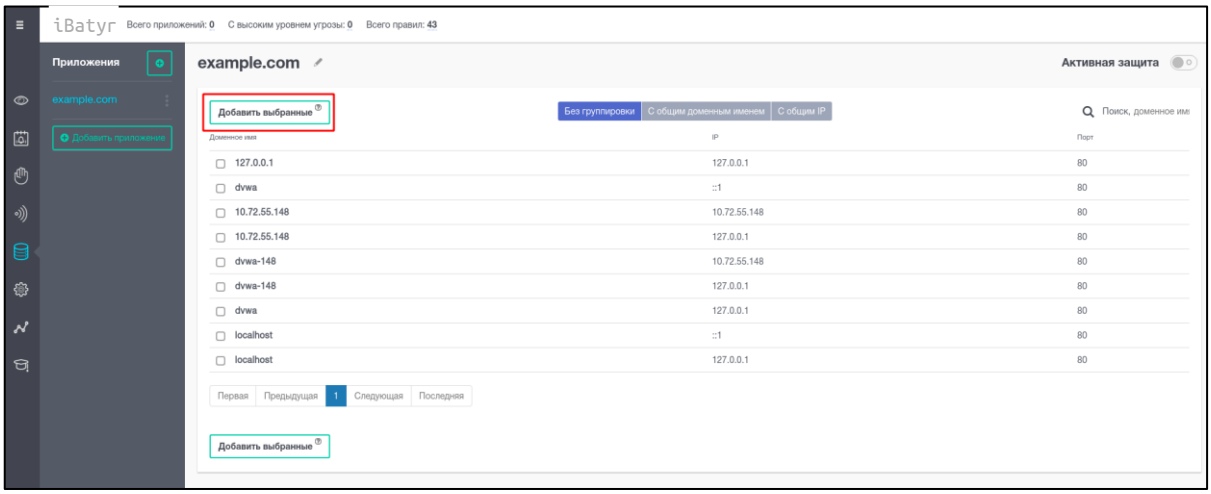
Снимок экрана 125 - Создание приложения

7.3. Добавление «троек»

В появившемся после создания приложения окне будут отображены те адреса\имена, трафик с которых был уже зафиксирован `ibatyr-nginx`.

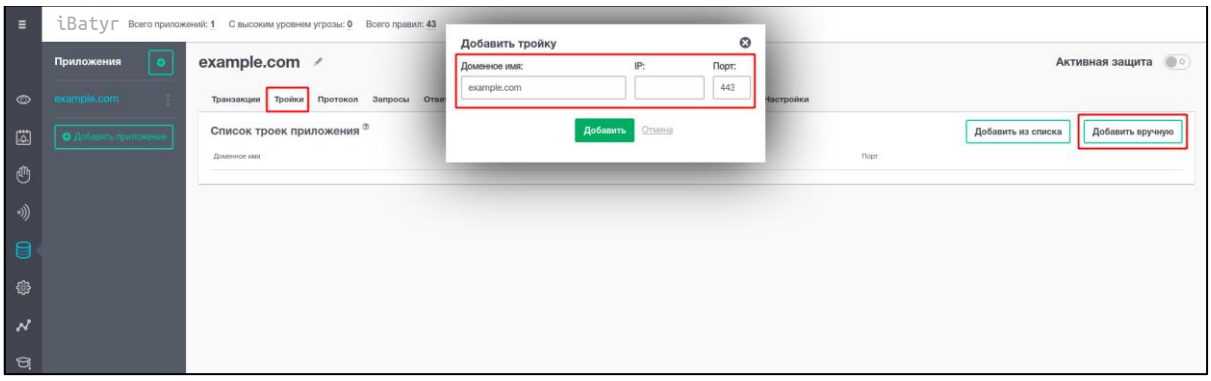
После того, как новое приложение было создано, к нему необходимо указать адреса, на которые настроено проксирование. Для этого следует:

1. Не выбирая ни один вариант, нажать кнопку «Добавить выбранные» (см. снимок экрана 126).



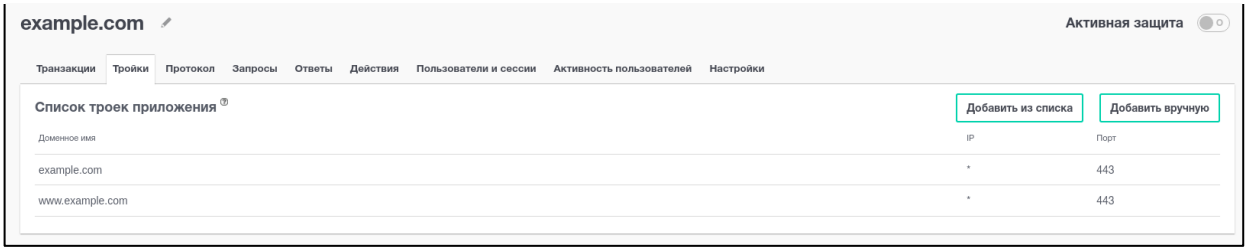
Снимок экрана 126 – Автоматический выбор «троек»

2. В следующем окне перейти на вкладку «Тройки» и нажать кнопку «Добавить вручную» (см. снимок экрана 127).



Снимок экрана 127 – Добавление «троек» вручную

3. В открывшемся окне «Добавить тройку» указать доменное имя защищаемого приложения и порт для приема трафика на WAF. IP-адрес указывать не нужно. Данную операцию необходимо проделать для всех доменных имен и поддоменов, а также портов, которые были добавлены в файл конфигурации ibatyr-nginx (см. снимок экрана 128).



Снимок экрана 128 – Добавленные «тройки»

7.4. Проверка прохождения трафика

Для проверки наличия сетевого доступа с сервера WAF к защищаемому приложению можно воспользоваться одной из следующих команд:

```
curl -v http://ip_addr_anlz:port
curl -X GET -H "host: webapp_name" http://ip_addr_anlz:port
curl -X GET -H "host: webapp_name" https://ip_addr_anlz:port
```

Где:

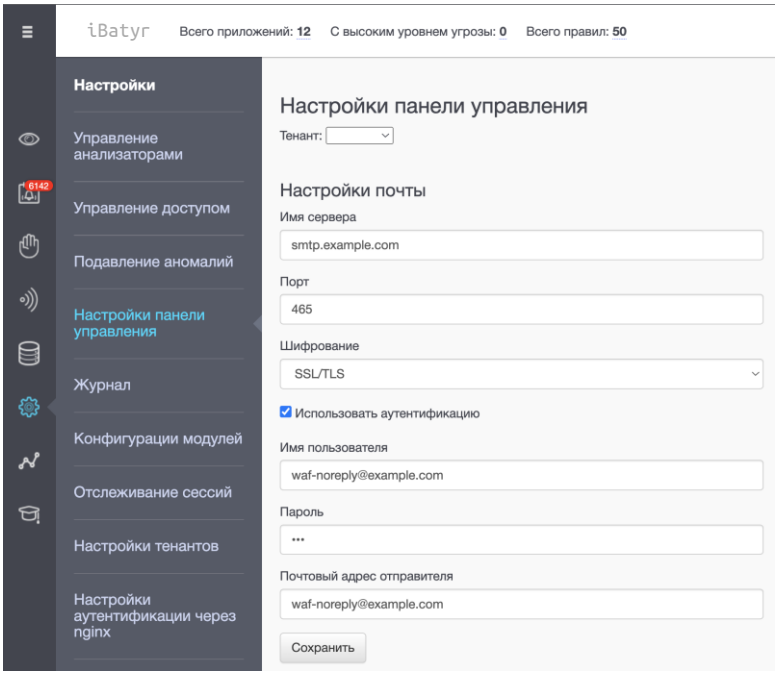
- `webapp_name` – имя защищаемого приложения;
- `ip_addr_anlz` – адрес анализатора;
- `port` – порт, на котором принимает соединения ibatyr-nginx.

8. Настройка интеграции со сторонними системами

8.1. Настройка уведомлений о событиях по электронной почте

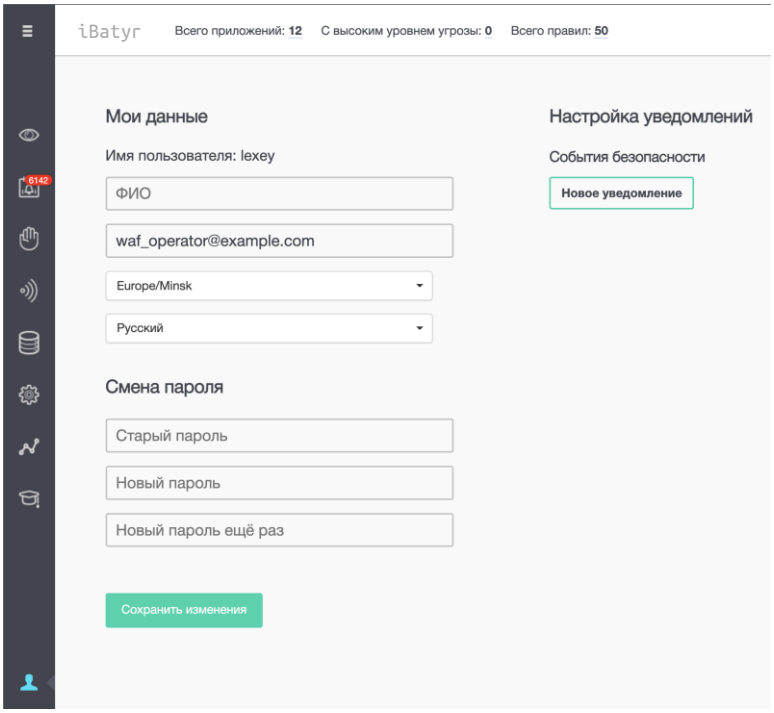
Предварительно необходимо создать почтовый ящик, от имени которого WAF будет отправлять уведомления. Любые действия с событиями, например, удаление, создание нового, закрытие будут сопровождаться отправкой уведомлений на электронную почту оператора WAF.

Для настройки отправки уведомлений о событиях на iBatyr WAF необходимо перейти на страницу «Настройки» панели управления» и заполнить настройки подключения к почтовому серверу, как показано на снимке экрана 129.



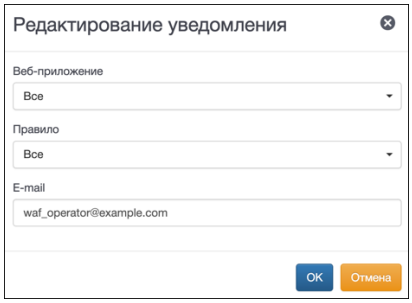
Снимок экрана 129 – Настройки уведомлений о событиях на электронную почту

Далее на странице «Мои данные» нужно настроить новое уведомление (см. снимок экрана 130).



Снимок экрана 130 – Создание нового уведомления

Выберите приложения и правила, для которых необходимо отправлять уведомления (см. снимок экрана 131).

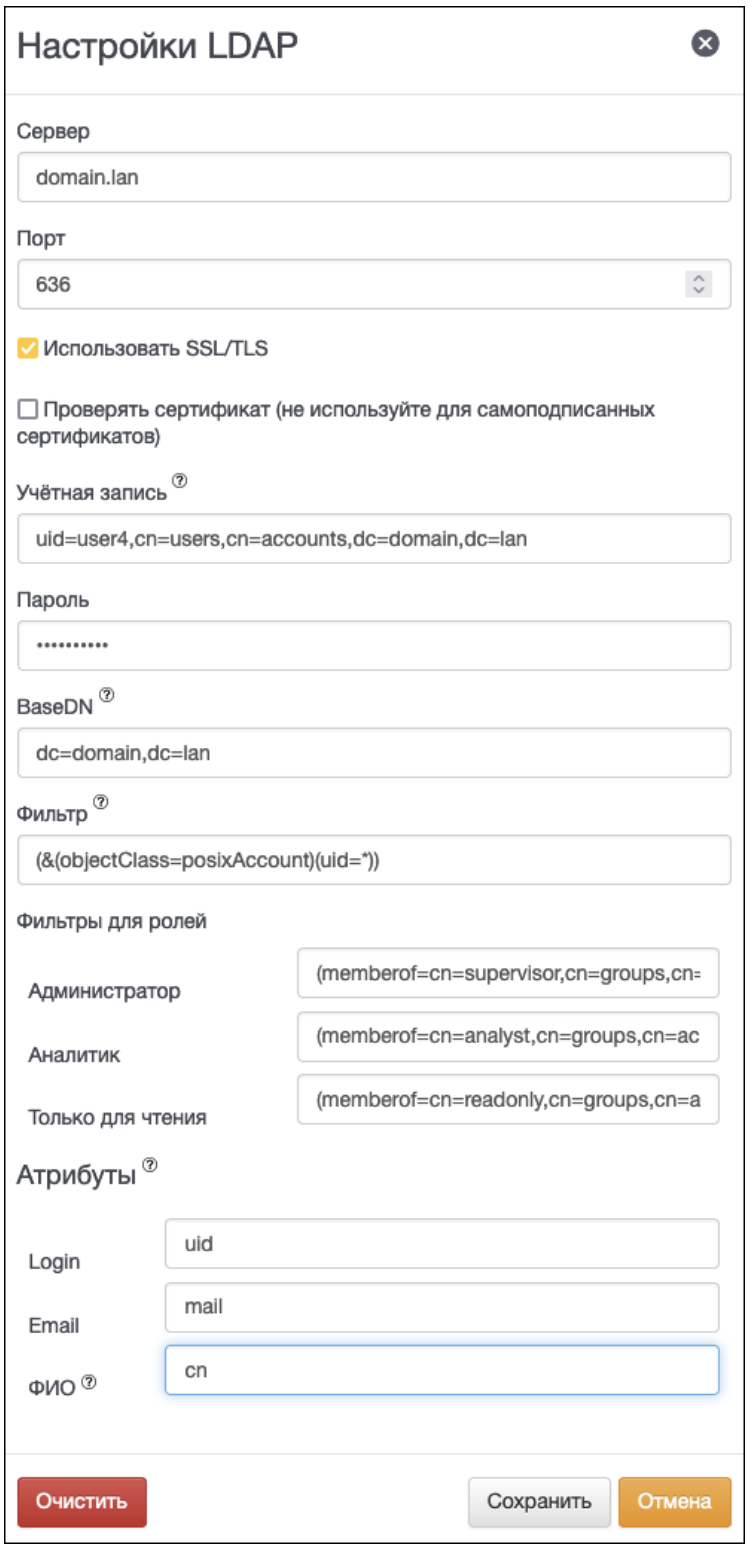


Снимок экрана 131 – Редактирование уведомления

Уведомления будут приходить на почтовый адрес текущего оператора WAF.

8.2. Настройка аутентификации по протоколу LDAP в веб-интерфейсе

Для настройки аутентификации по LDAP для веб-интерфейса необходимо перейти на страницу «Настройки/Настройки панели управления» и ввести необходимые данные (см. снимок экрана 132). Данные необходимо запрашивать у администратора домена организации.



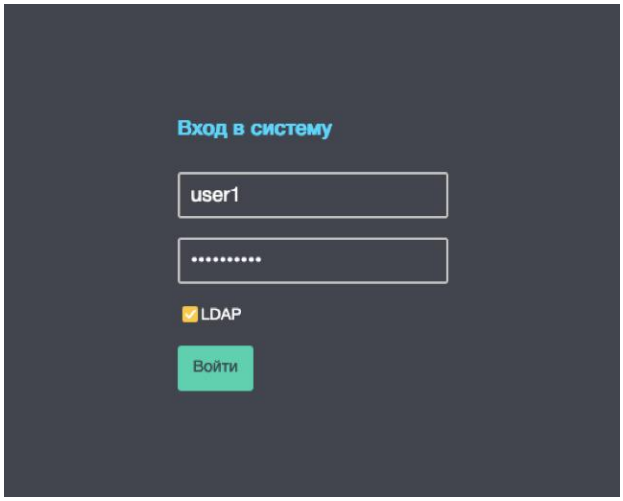
Снимок экрана 132 – Настройки LDAP

Пример для одного из полей «Фильтров», когда необходимо пускать только членов конкретной группы Active Directory (AD):

```
(&(objectCategory=person)(objectClass=user)(memberof=cn=supervi-
sor,cn=groups,cn=accounts,dc=domain,dc=lan))
```

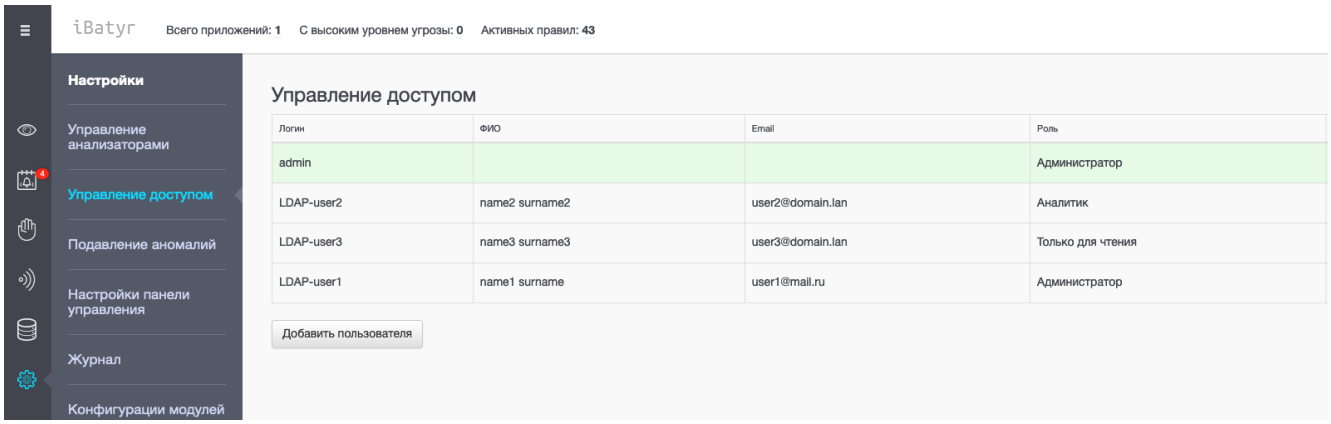
В качестве учетных данных для доступа веб-интерфейса в LDAP рекомендуется использовать служебную учетную запись (в примере выше — пользователь `user4`). Персональные учетные записи лучше не использовать, т.к. у них может истечь пароль, после чего LDAP-аутентификация может перестать работать для веб-интерфейса.

После сохранения данных можно авторизоваться в веб-интерфейсе, введя доменную учетную запись и пароль, и отметив настройку «LDAP» (см. снимок экрана 133).



Снимок экрана 133 – Вход в систему с доменной учётной записью

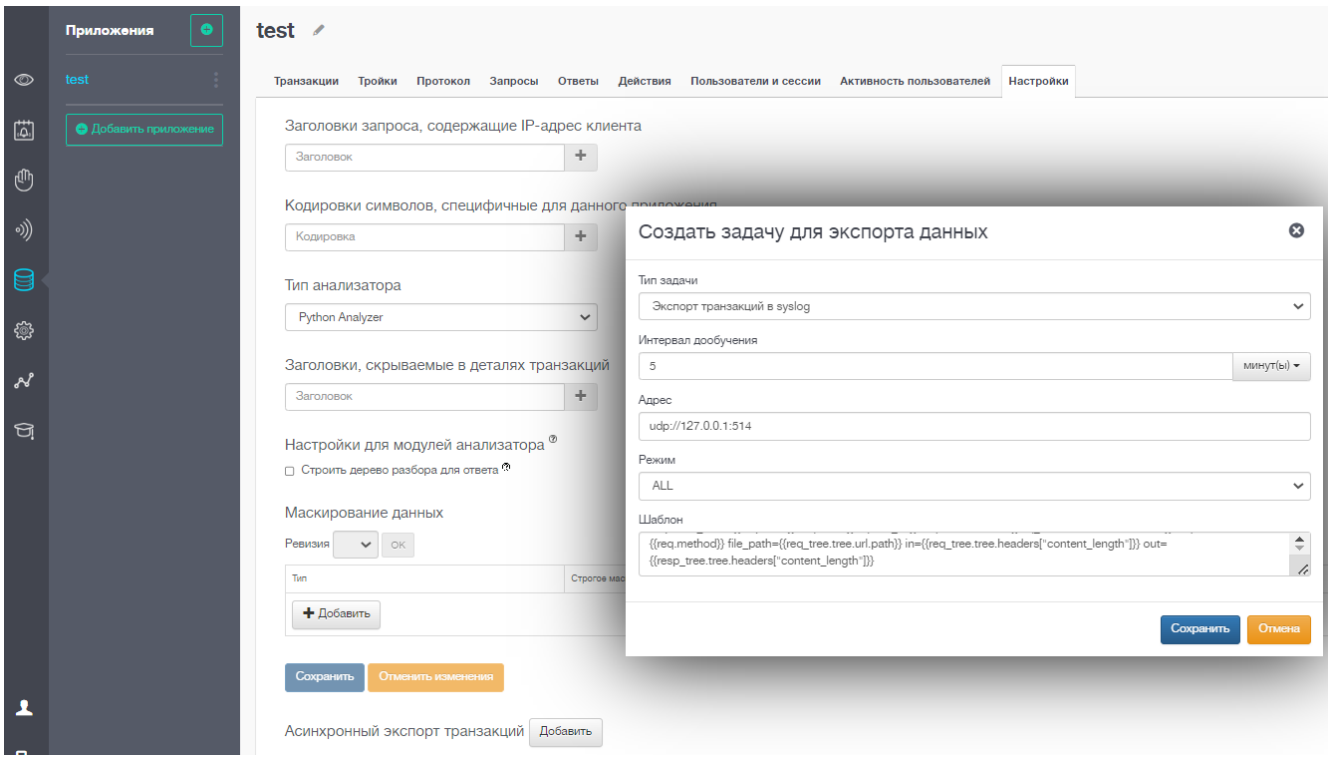
После успешной аутентификации в списке пользователей веб-интерфейса появится новая учетная запись с данными из LDAP-каталога (см. снимок экрана 134).



Снимок экрана 134 – Список аутентифицировавшихся пользователей

8.3. Настройка интеграции с SIEM системами

Для того чтобы настроить отправку журналов в стороннюю SIEM-систему, необходимо в веб-интерфейсе создать шаблон отправки (вкладка «Приложения» -> «Настройки» -> «Асинхронный экспорт транзакций» -> «Добавить») (см. снимок экрана 135).



Снимок экрана 135 – Создание шаблона отправки журналов в стороннюю SIEM-систему

Примеры шаблонов:

1. Шаблон LEEF-формата:

```
'{{time}}                    {{host_ip}}                    LEEF:2.0|iBatyr
WAF|1.0|{{req.obj_id}}|src={{req.src_ip}}\tspt={{req.src_port}}\tdst={{re
q.dst_ip}}\tdpt={{req.dst_port}}\tre-
quest_time={{req.time}}\tresponse_time={{resp.time}}\turl={{req.raw_uri}}
\tsev={{severity}}\treq_tree.headers.cookie={{req_tree.tree.headers.cooki
e}}\trequest-
Method={{req.method}}\treq_tree.url.path={{req_tree..tree.url.path}}\tdst
Bytes={{req_tree.tree.headers["content-
length"]}}\tsrcBytes={{resp_tree.tree.headers["content-length"]}}'
```

2. Шаблон CEEF-формата:

```
'{{req.time.strftime("%b    %d    %H:%M:%S")}}    localhost    CEF:0|iBatyr
WAF|1.0|{{req.obj_id}}|HTTP    Transaction|{{severity}}|src={{req.src_ip}}
scrPort={{req.src_port}} dst={{req.dst_ip}} dstPort={{req.dst_port}} re-
quest_time={{time}}  response_time={{resp.time}}  request={{req.raw_uri}}
requestCookies={{req_tree.tree.headers.cookie}}                    request-
Method={{req.method}}                file_path={{req_tree.tree.url.path}}
in={{req_tree.tree.headers["content_length"]}} out={{resp_tree.tree.head-
ers["content_length"]}}'
```

Ниже приведено описание параметров и возможные значения полей для отправки:

1. Режим:

- ALL – отправляется журнал обо всех транзакциях.
- BLOCKED – отправляется журнал только о заблокированных транзакциях.

2. Интервал дообучения – с какой частотой будет осуществляться выборка транзакций из базы данных и их отправка в SIEM.
3. Адрес:

- tcp://siem_ip:514 – отправка по tcp-соединению;
- udp://siem_ip:514 – отправка по udp-соединению;
- file:///dev/log – запись журнала в локальную файловую систему на узле WAF.

4. Допустимые плейсхолдеры:

- time - временная метка транзакции;
- host_ip - адрес узла, с которого осуществляется отправка журнала;
- severity – серьезность события;
- req – данные о запросе:
  - obj_id – идентификатор транзакции на WAF;
  - src_ip – адрес источника запроса;
  - src_port – порт источника запроса;
  - dst_ip – адрес назначения запроса;
  - dst_port – порт назначения запроса;
  - time – временная метка запроса;
  - raw_uri – адрес запрашиваемой страницы;
  - method – HTTP-метод;
  - protocol – протокол передачи.
- resp – данные об ответе:
  - time – временная метка ответа.
- req_tree – данные о разобранном дереве запроса:
  - tree:
    - time – временная метка запроса;
    - src_ip – адрес источника запроса;
    - src_port – порт источника запроса;
    - dst_ip – адрес назначение запроса;
    - dst_port – порт назначения запроса;
    - protocol – протокол передачи;
    - method – HTTP-метод;
    - url – адрес запрашиваемой страницы;
    - headers – заголовки.

5. Cookie – заголовок Cookie.
6. ... – прочие заголовки.

- resp_tree – данные о разобранном дереве ответа (по умолчанию разбор дерева ответа отключен):
  - tree:
    - time – временная метка ответа;
    - src_port – порт источника запроса;

- dst_ip – адрес назначения запроса;
- dst_port – порт назначения запроса;
- protocol – протокол передачи;
- method – HTTP-метод;
- url – адрес запрашиваемой страницы;
- headers – заголовки ответа.

7. Cookie:

- req_decision – решение WAF (пропустить/заблокировать):
  - decision – решение на запросе;
  - rule – сработавшее на запросе правило:
    - name – наименование правила;
    - obj_id – идентификатор правила.
- resp_decision – сработавшее на ответе правило:
  - decision – решение на ответе;
  - rule – сработавшее на ответе правило:
    - name – наименование правила;
    - obj_id – идентификатор правила.
- webapp_id – идентификатор защищаемого приложения на WAF;
- webapp_name – имя приложения;
- action_id – идентификатор действия позитивной модели бизнес-логики (подробности см. в руководстве оператора);
- action_name – имя действия модели бизнес-логики;
- action_parameters – параметры действия модели бизнес-логики;
- request_session_id – идентификатор сессии запроса;
- response_session_id – идентификатор сессии ответа.

Так как в качестве шаблонов используется `Jinja`, то в шаблонах имеется возможность писать выражения следующего вида.

1. Шаблон:

```
block_rule_name={{req_decision.rule_name if req_decision.decision == "BLOCK" else resp_decision.rule_name}}
```

Результат:

```
block_rule_name=Наименование правила блокировки запроса (если причиной была блокировка запроса), или ответа (если причиной была блокировка ответа)
```

2. Шаблон:

```
{% for rule in req_decision.reason.matching_rules %} rule={{rule.name}} {% endfor %}
```

Результат:

```
rule=Параметр не соответствует модели rule=Внедрение javascript-кода в страницы rule=Прочие сигнатуры
```

8.4. Настройка мониторинга zabbix

Для того чтобы настроить мониторинг узлов iBatyr WAF, необходимо установить `zabbix-agent` актуальной версии. В зависимости от выбранного режима отправки (пассивный/активный) произвести необходимые настройки доступов для взаимосвязи с zabbix-сервером. Настройки соединения находятся в `/etc/zabbix/zabbix_agentd.conf`. В файле необходимо прописать наименование конкретного узла, адрес сервера и порты (Hostname, Server, ServerActive).

8.4.1. Установка и настройка на стороне сервера

Установку можно проводить как вручную (для сетей без доступа в интернет), скачав и установив пакет с официального репозитория, так и через добавление самого репозитория Zabbix и последующую установку при помощи пакетного менеджера.

Для этого необходимо скачать пакет, содержащий данные о репозитории и GPG-ключ, и установить его в систему.

```
wget -P /tmp https://repo.zabbix.com/zabbix/5.4/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.4-1%2Bubuntu20.04_all.deb
sudo dpkg -i /tmp/zabbix-release_5.4-1+ubuntu20.04_all.deb
sudo apt update
sudo apt install zabbix-agent
```

После установки нужно сгенерировать PSK-ключ для шифрования данных от агента:

```
openssl rand -hex 32 > /etc/zabbix/secret.psk
```

Минимальная конфигурация `/etc/zabbix/zabbix-agent.conf`:

```
PidFile=/var/run/zabbix/zabbix_agentd.pid
LogFile=/var/log/zabbix/zabbix_agentd.log
LogFileSize=0

# Общие настройки для работы в активном и пассивном режимах. В HOSTNAME подставить имя текущего узла.
Server=ZABBIX_SERVER_IP
ServerActive=ZABBIX_SERVER_IP
Hostname=HOSTNAME
Include=/etc/zabbix/zabbix_agentd.d/*.conf

# Пользовательские параметры для выполнения проверок по службам
UserParameter=sysd.service[*],systemctl is-active --quiet '$1' && echo 1 || echo 0
UserParameter=sysd.service.instance[*],systemctl is-active --quiet '$1'@'$2' && echo 1 || echo 0

# Метод шифрования данных, если необходимо. В HOSTNAME подставить имя текущего узла.
TLSConnect=psk
TLSAccept=psk
TLSPSKIdentity=HOSTNAME
TLSPSKFile=/etc/zabbix/secret.psk
```

После изменения конфигурации необходимо перезагрузить сервис и включить его автозапуск с помощью команд:

```
sudo systemctl restart zabbix-agent
sudo systemctl enable zabbix-agent
```

8.4.2. Дополнительная настройка для мониторинга WAF в кластере

Для обеспечения мониторинга сервисов под управлением Corosync\Pacemaker, необходимо использовать внешний скрипт.

Поскольку данный функционал реализован на парсинге данных xml-формата CRM, в системе должен присутствовать python3.8 и библиотека python-lxml. Установить библиотеку можно с помощью команды:

```
apt update
apt install python-lxml
```

После установки необходимо скачать скрипт в каталог `/usr/local/bin`:

```
wget -P /usr/local/bin https://raw.githubusercontent.com/digiapulssi/zab-
bix-monitoring-scripts/master/etc/zabbix/scripts/pacemaker.py
chmod +x /usr/local/bin/pacemaker.py
```

Затем добавить дополнительный `UserParameter` в файл конфигурации `zabbix-agent`:

```
UserParameter=pacemaker.status[*],/usr/local/bin/pacemaker.py $1 $2 $3 $4
$5 $6 $7 $8
```

8.4.3. Шаблоны мониторинга

Шаблоны мониторинга, совместимые с zabbix-server 4.0 – 6.0, zabbix-agent (от версии 4.0), zabbix-agent2, поставляются отдельно по требованию. Для получения базовых системных метрик рекомендуется использовать шаблон `Linux by Zabbix agent active`, входящий в базовый набор шаблонов Zabbix-сервера. Полученные шаблоны мониторинга необходимо импортировать на Zabbix-сервер (файл с шаблоном, который содержит два шаблона – `Universal WAF State AA (Safe)` для активного режима и `Universal WAF State PA (Safe)` для пассивного режима `zabbix-agent`).

После импорта файла в конфигурации макросов хоста возможно указать `$WAF_TYPE (default ibatyr)`, а также, если необходим мониторинг кластера – имя ноды `Corosync $NODE_NAME (default waf-repl-node-1)`.

Также для получения базовых системных метрик рекомендуется добавить шаблоны `Template OS Linux by Zabbix agent active` для активного режима или `Template OS Linux by Zabbix agent` для пассивного режима мониторинга.

8.4.4. Настройка элементов данных шаблона

Поскольку шаблоны являются универсальными для всех типов инсталляций – часть элементов данных и триггеров может не использоваться и по умолчанию отключены, чтобы не увеличивать очередь сервера. Если какие-то из них планируется использовать, их необходимо включить. Это можно сделать через

WebUI – вкладку «Настройки» -> «Шаблоны» -> «Universal WAF State XX (Safe)». На вкладке «Элементы данных» нужно выбрать те элементы, которые требуется опрашивать, и нажать кнопку «Активировать».

По умолчанию отключенные элементы:

- ibatyr-analyzer@1;
- ibatyr-redis@1;
- postgresql@11-main (CLUSTER);
- ibatyr-celery (CLUSTER);
- ibatyr-celerybeat (CLUSTER);
- ibatyr-redis@mgmt (CLUSTER).

Элементы с пометкой «CLUSTER» работают только при выполнении шага, описанного в разделе 8.4.2.

9. Диагностика неполадок

9.1. Не запускаются сервисы

9.1.1. ibatyr-analyzer, ibatyr-dashboard

Если после запуска сервис отсутствует в списке процессов, необходимо посмотреть на причину ошибки в журнале journald с помощью команды:

```
journalctl -xe -u <service-name>
```

9.1.2. ibatyr-celery / ibatyr-celerybeat

Проверить журналы можно с помощью команды:

```
 /var/log/waf/wafcelery*.log
```

9.2. Заканчивается место на диске

9.2.1. /var/lib

Необходимо определить какой процесс занимает пространство.

Для этого выполните команду:

```
sudo du -hs /var/lib/* | sort -rh | head -n 10
```

Если основную часть раздела занимает папка `postgres`, проверьте какие таблицы в PostgreSQL занимают место следующими командами.

```
sudo -u waf psql
\d+
```

Для очистки места можно удалить партиции таблиц со старыми данными (таблицы `http_transaction` и `anomaly`) и уменьшить глубину хранения данных (см. п. 6.3).

9.2.2. /var/log

Необходимо определить процесс, который содержит файлы большого размера. Для этого выполните команду:

```
sudo du -hs /var/log/* | sort -rh | head -n 10
```

Проверьте корректность настройки правил `logrotate` компонент найденного процесса в `/etc/logrotate.d`.

9.3. Недоступность приложения

Если приложение через WAF недоступно, рекомендуется проверить корректность следующих аспектов:

- Сетевая адресация:

```
ip a
```

- Разрешение имен:

```
host example.com
```

- Маршрутизация:

```
ip ro
```

- Доступность приложения напрямую:

```
curl -v --resolve www.example.com:80:127.0.0.1 http://example.com
curl -k -v  --resolve www.example.com:443:127.0.0.1 https://example.com
```

- Состояние сервиса `ibatyr-nginx`:

```
sudo systemctl status ibatyr-nginx.service
```

- Перечень открытых портов:

```
ss -lnt
```

- Наличие ошибок в журнале сервиса `ibatyr-nginx`:

```
sudo journalctl -xefu ibatyr-nginx
```

- Наличие ошибок в конфигурации `ibatyr-nginx`:

```
sudo ibatyr-nginx test
```

Приложение А. Заведение дополнительного инстанса анализатора

В некоторых случаях полезно делать на одном узле анализатора несколько инстансов `ibatyr-analyzer`. Например, это нужно делать для узлов с большим количеством ядер и памяти, иначе однопоточный Redis, через который общаются `ibatyr-analyzer` и `ibatyr-nginx`, может не справиться с нагрузкой. Также можно делать инстансы с разными настройками, чтобы разные защищаемые ресурсы или отдельные их эндпоинты могли обрабатываться с разными настройками анализаторов, такими как таймауты принятия решений, вкл/выкл разного набора детекторов и др.

Все они управляются (запуск, рестарт и стоп) при помощи службы ibatyr-analyzer. Отдельным инстанс анализатора запускается, как служба ibatyr-analyzer@X, где X – порядковый номер, начиная с 0.

A.1. Рекомендации по количеству инстансов анализатора

На каждые 16 ядер рекомендуется создавать по одному инстансу анализатора с 14 пайплайнами. Итого: 14 ядер выделяется анализатору, 1 – `ibatyr-redis@0`, 1 – `ibatyr-nginx`.

На машине с 32 ядрами должно быть 2 инстанса анализаторов и 2 redis.

Для оценки производительности узла анализа можно ориентироваться на производительность 16-ти ядер ЦПУ ~ 1000 rps.

Изменить лимит потребления памяти у имеющегося инстанса так, чтобы памяти хватило второму. Настройки лимита хранятся в `/etc/default/ibatyr-analyzer (MEMORY_LIMIT)`. Лимит распространяется на один инстанс службы анализатора.Создать дополнительный инстанс можно с помощью команды:

```
sudo ibatyrctl rescale 2
```

В результате создадутся дополнительные инстансы ibatyr-analyzer@1 и ibatyr-redis@1.

Добавить в настройки `/etc/ibatyr-nginx/user/server.conf` запись про redis@1 можно с помощью команды:

```
set $REDIS "redis://127.0.0.1:6379 redis://127.0.0.1:6380";
```

Завести новый анализатор в веб-интерфейсе, изменить порт анализатора и порт в IP-адресе (watchdog -> endpoints -> controller) на незанятый, например 6667, изменить порт локального редиса на 6380 (см. снимок экрана А.1).

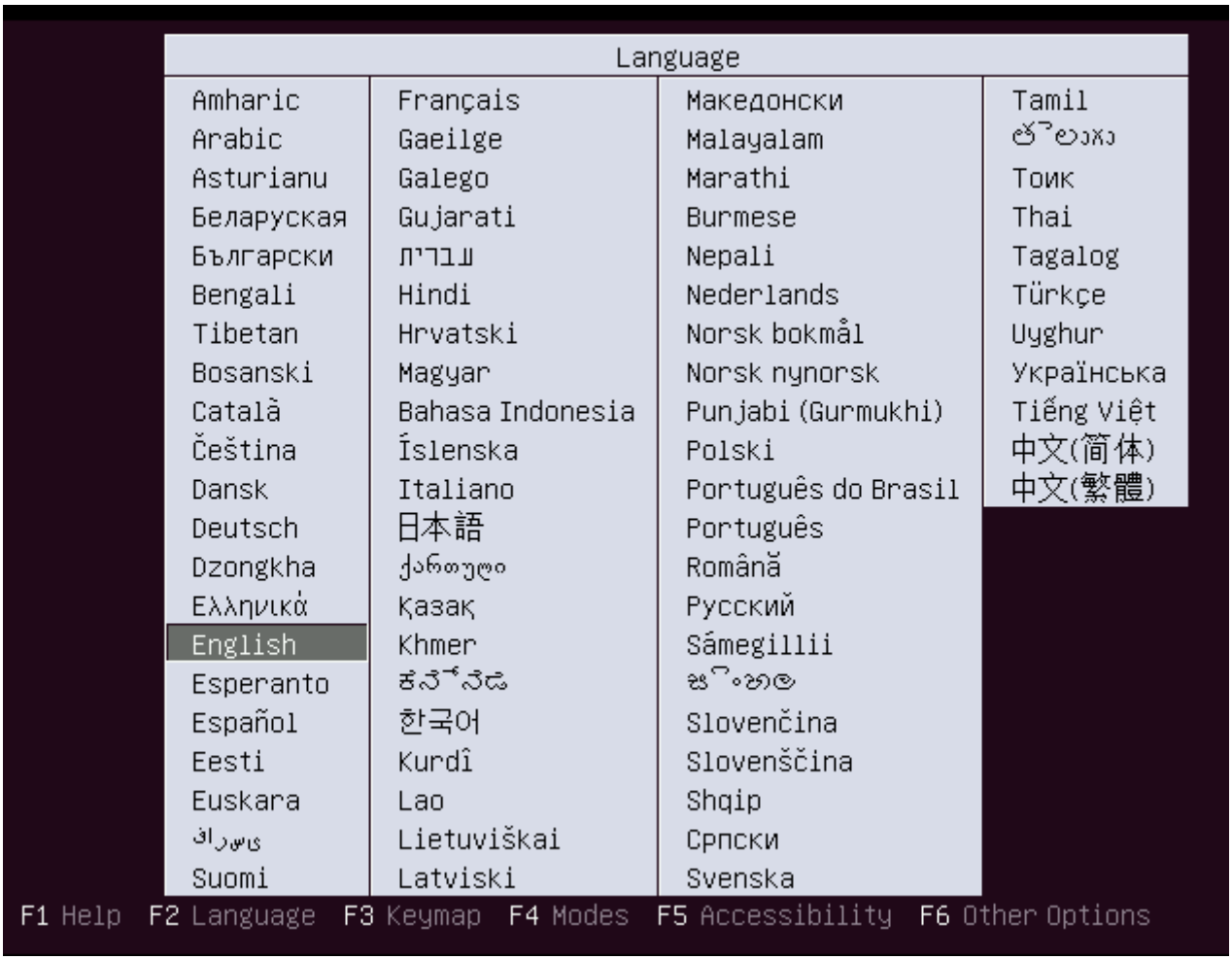
Снимок экрана А.1 – Порт анализатора

Перезапустить службу ibatyr-analyzer и применить настройки на ibatyr-nginx.

Приложение Б. Пример установки ОС Ubuntu 22.04

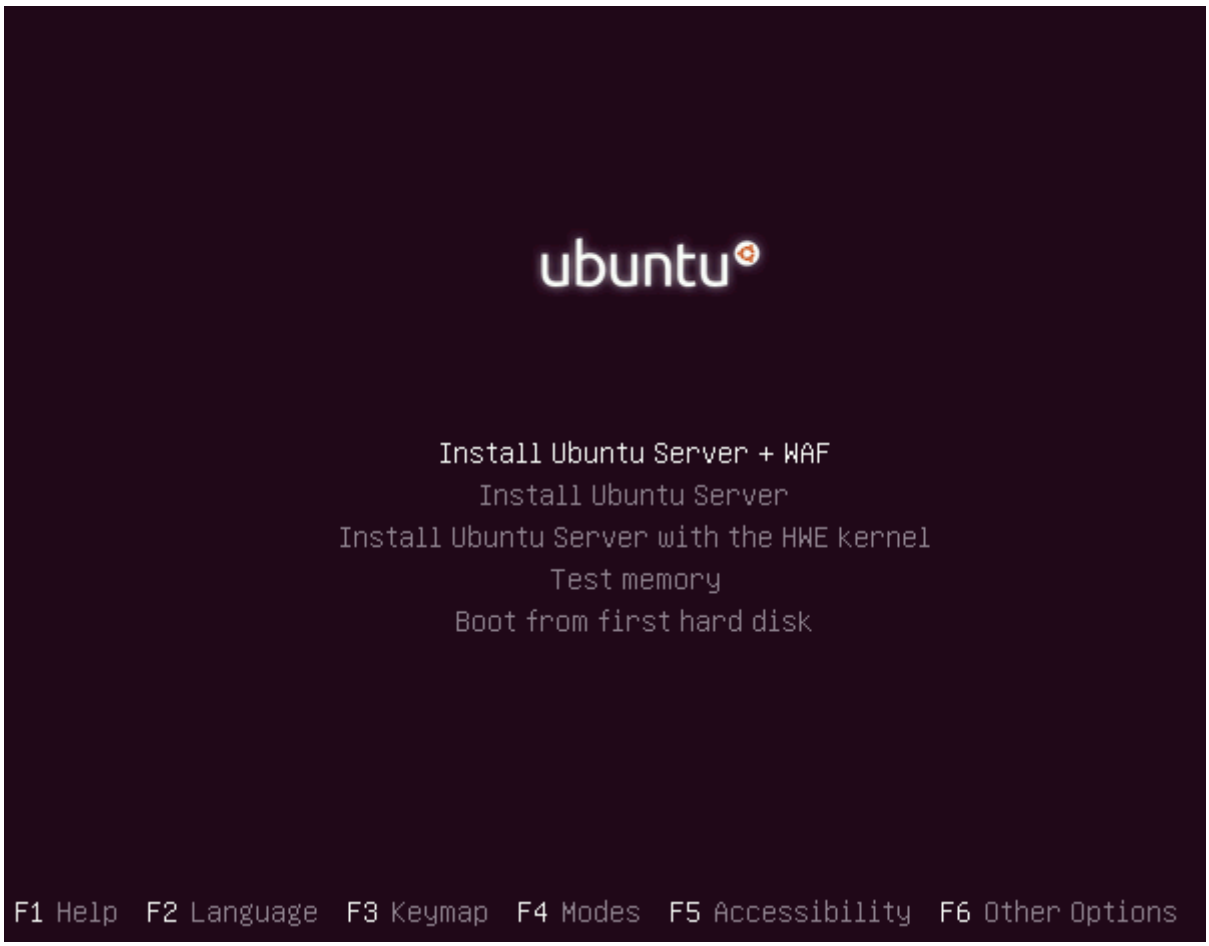
Ниже показана пошаговая установка ОС Ubuntu 2X.04 из образа ibatyr-ubuntu-2X.04-v2.XX.0.iso (ОС+WAF) с выбором опций при установке.

1. Рекомендуем выбирать язык English, чтобы на следующих шагах не возникло проблем с кодировкой (см. снимок экрана Б.1).



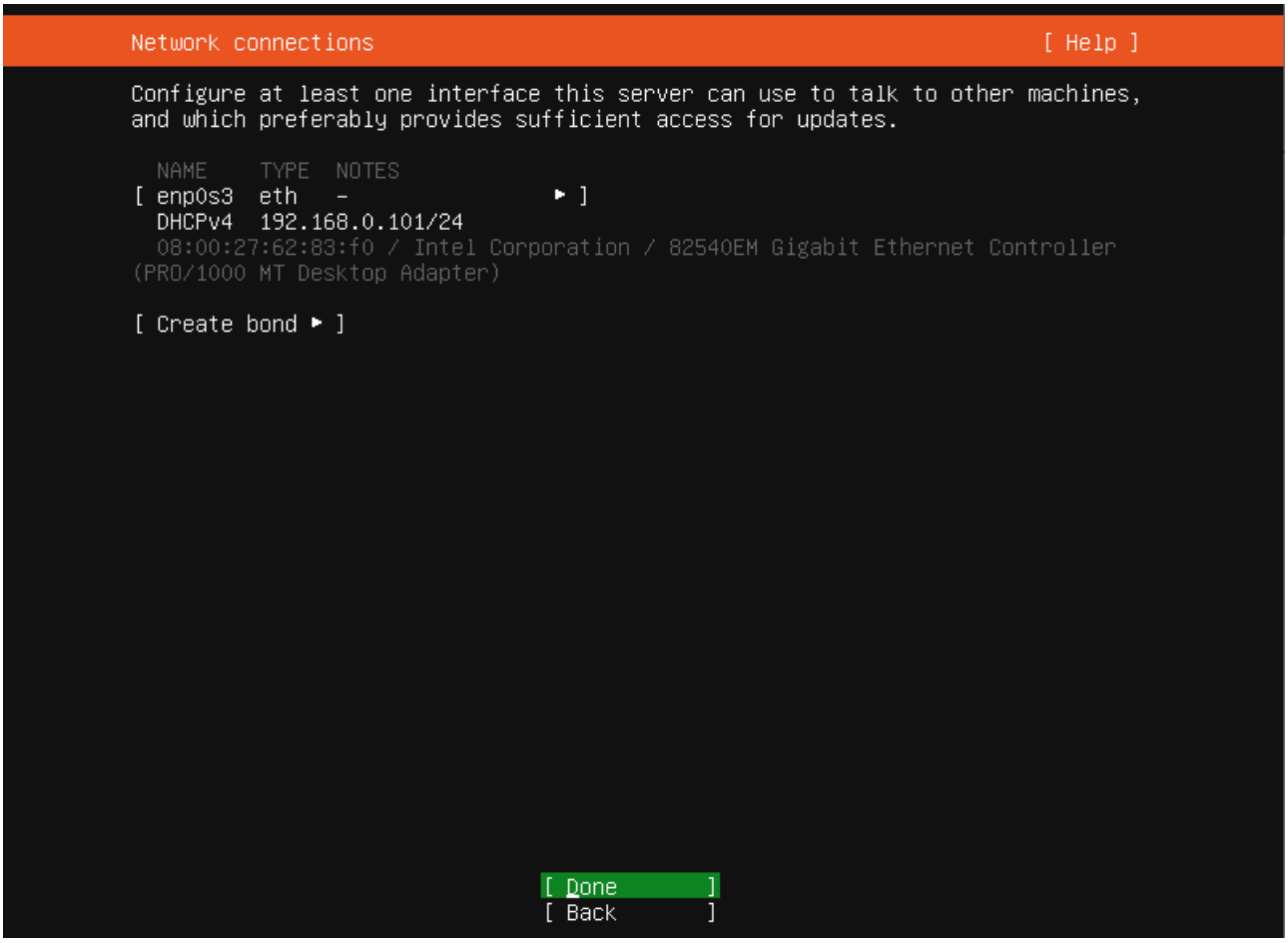
Снимок экрана Б.1 – Выбор языка

2. Далее выбрать установку `Install Ubuntu Server + WAF` (см. снимок экрана Б.2).



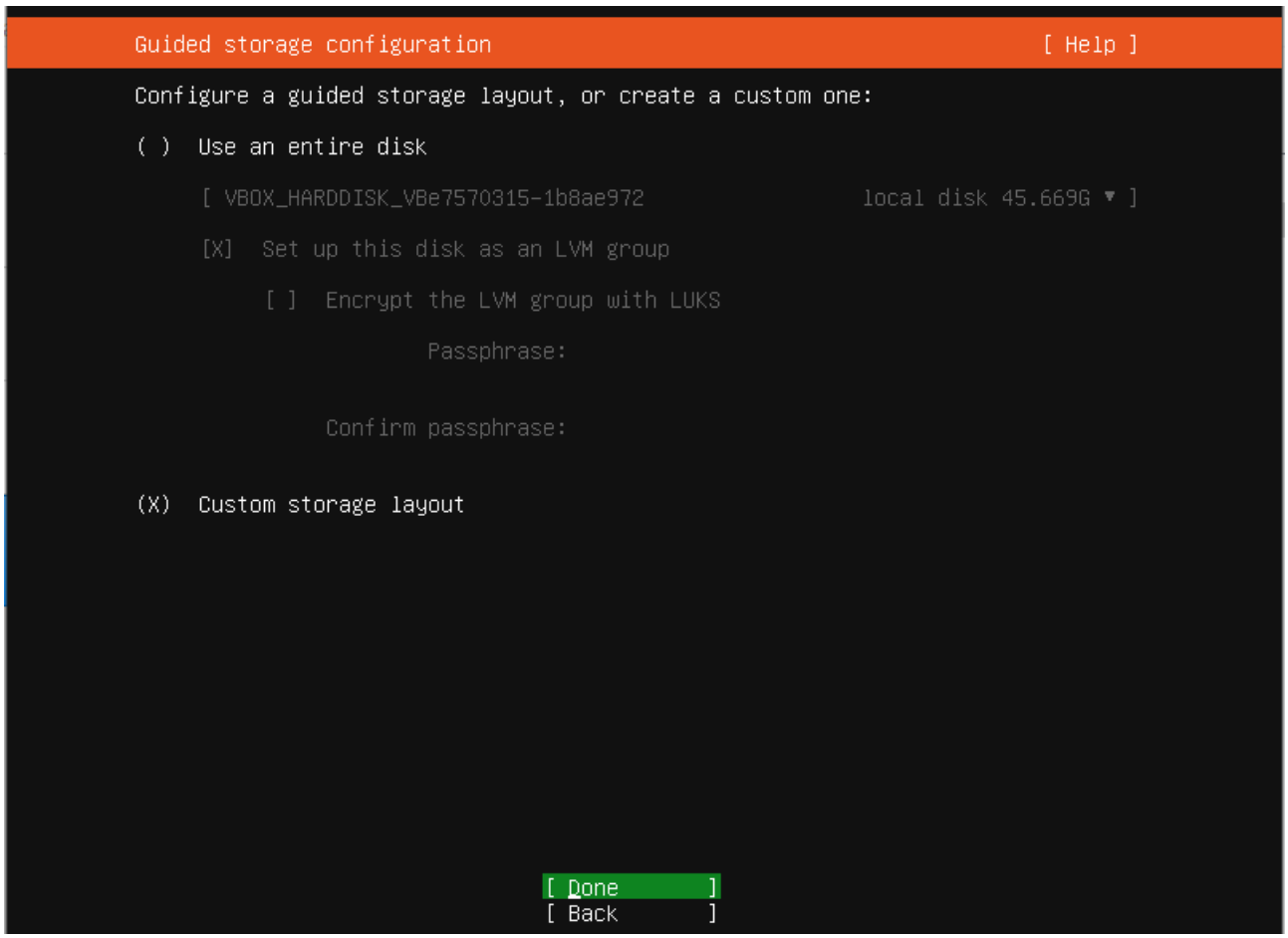
Снимок экрана Б.2 – Выбор установки

На этапе настройки сетевого окружения можно задать необходимые параметры для подключения по локальной сети (см. снимок экрана Б.3).

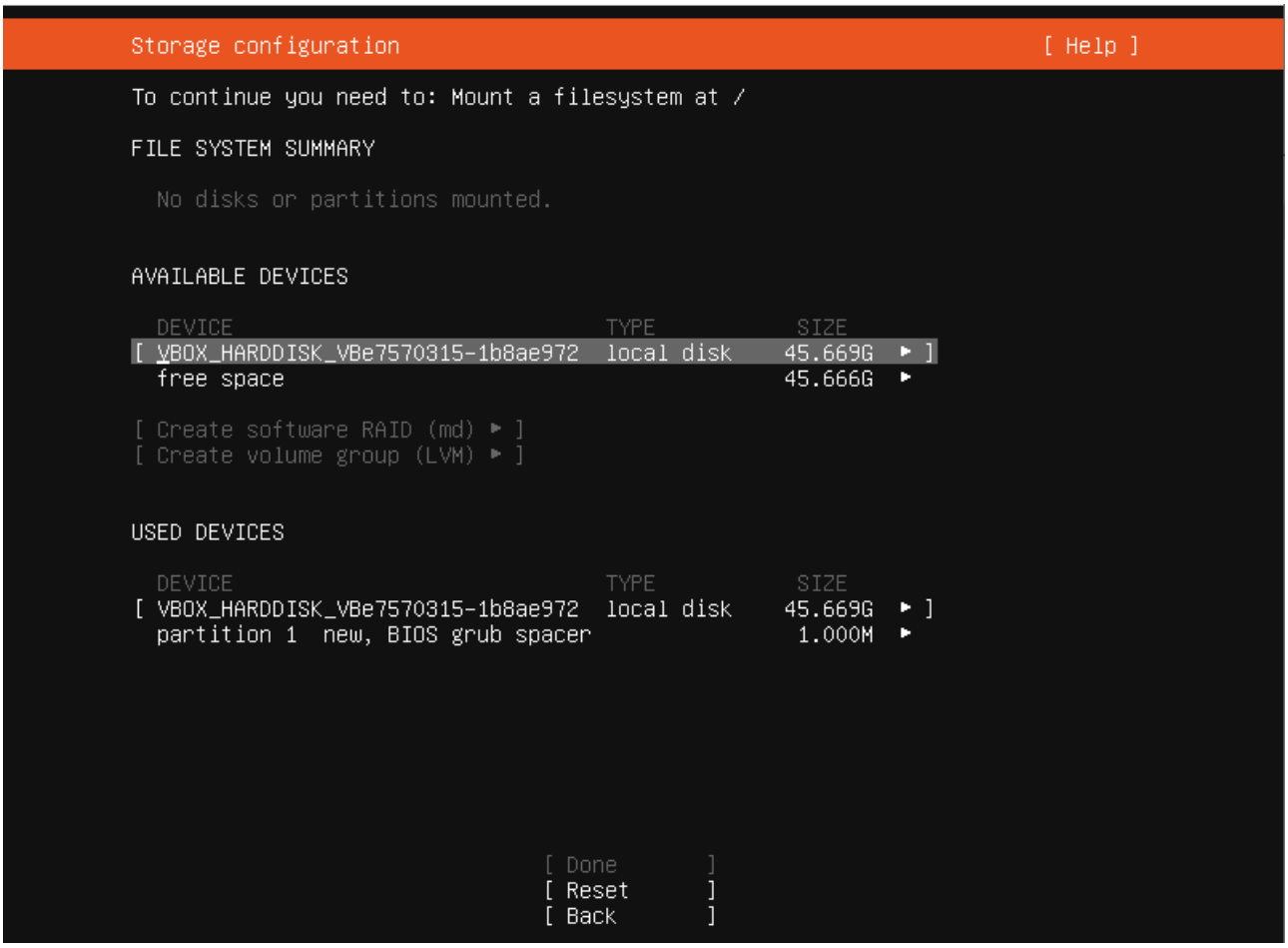


Снимок экрана Б.3 – Сетевые параметры

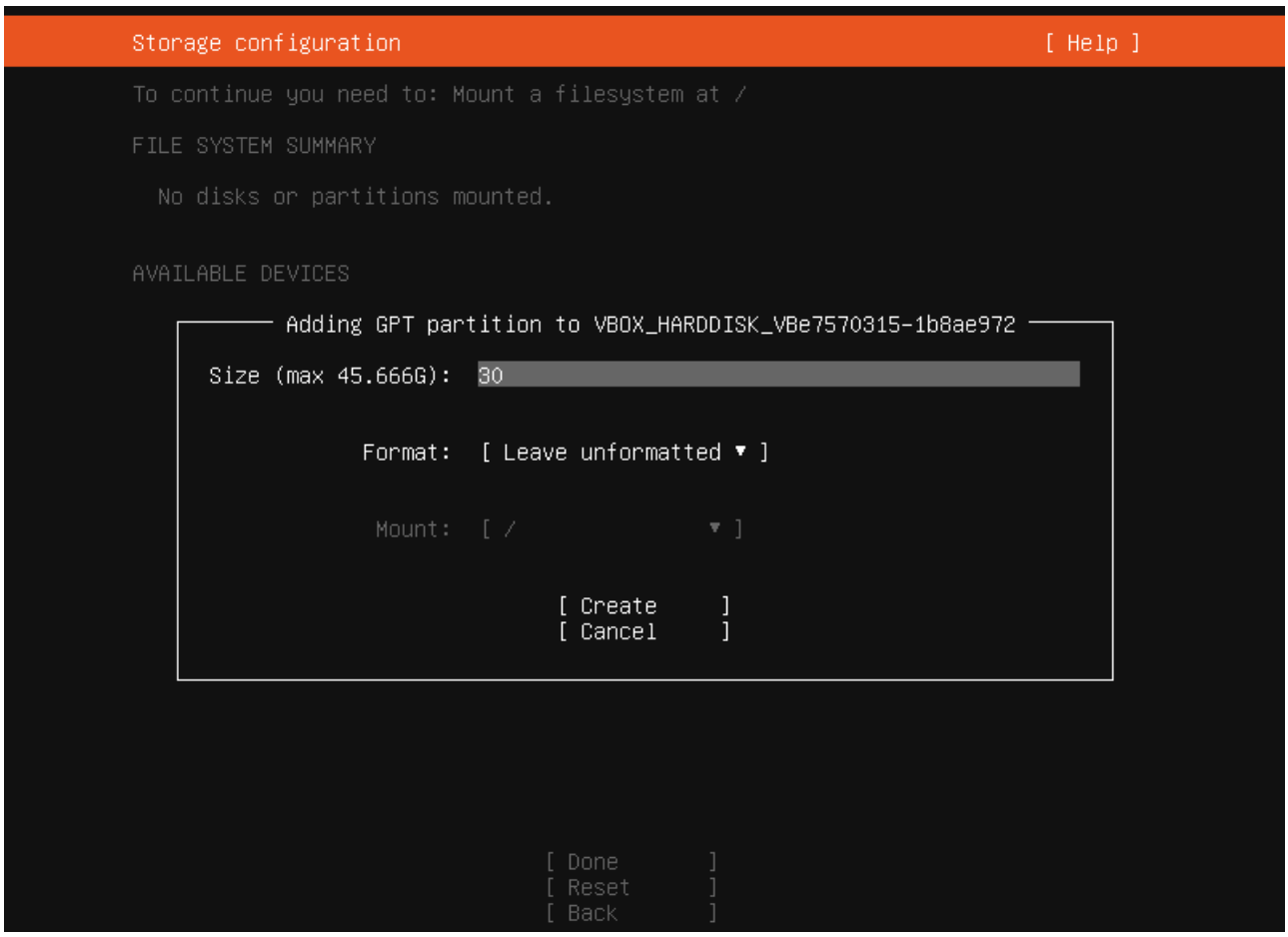
Настройка разметки диска предполагает создание виртуальной группы и логического тома и используется как эталонная модель, которую в последующем следует придерживаться для удобства управления и расширения дискового пространства. При установке ОС настраивается корневая группа, остальные манипуляции с разбиением дискового пространства можно проводить по аналогичной схеме уже после установки ОС, для чего нужно иметь необходимый опыт в создании групп логических томов (см. раздел 3, снимки экранов Б.4 - Б.12). Рекомендации по разбиению диска описаны в разделе 2.2.



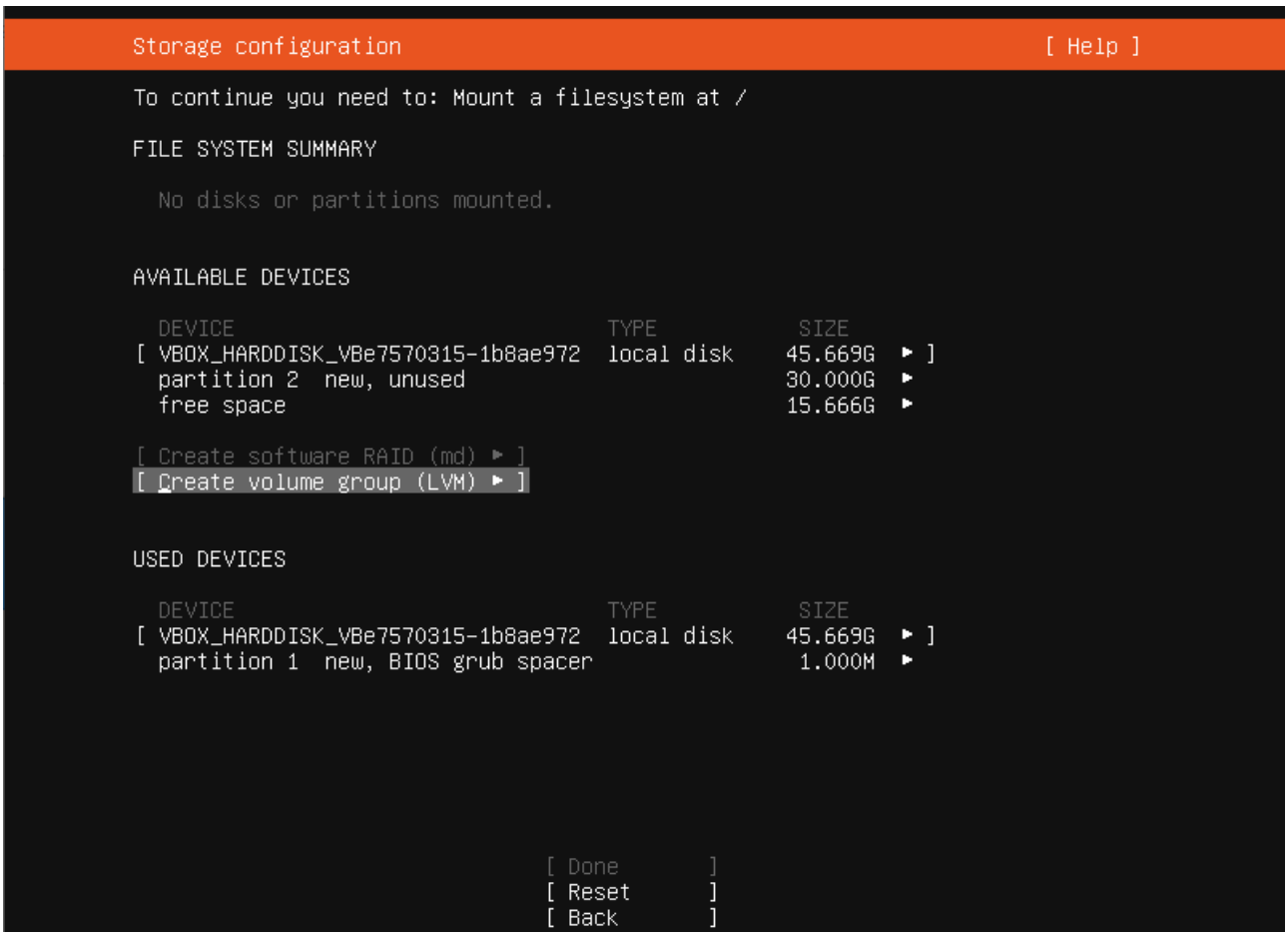
Снимок экрана Б.4 – Выбор ручной разметки диска



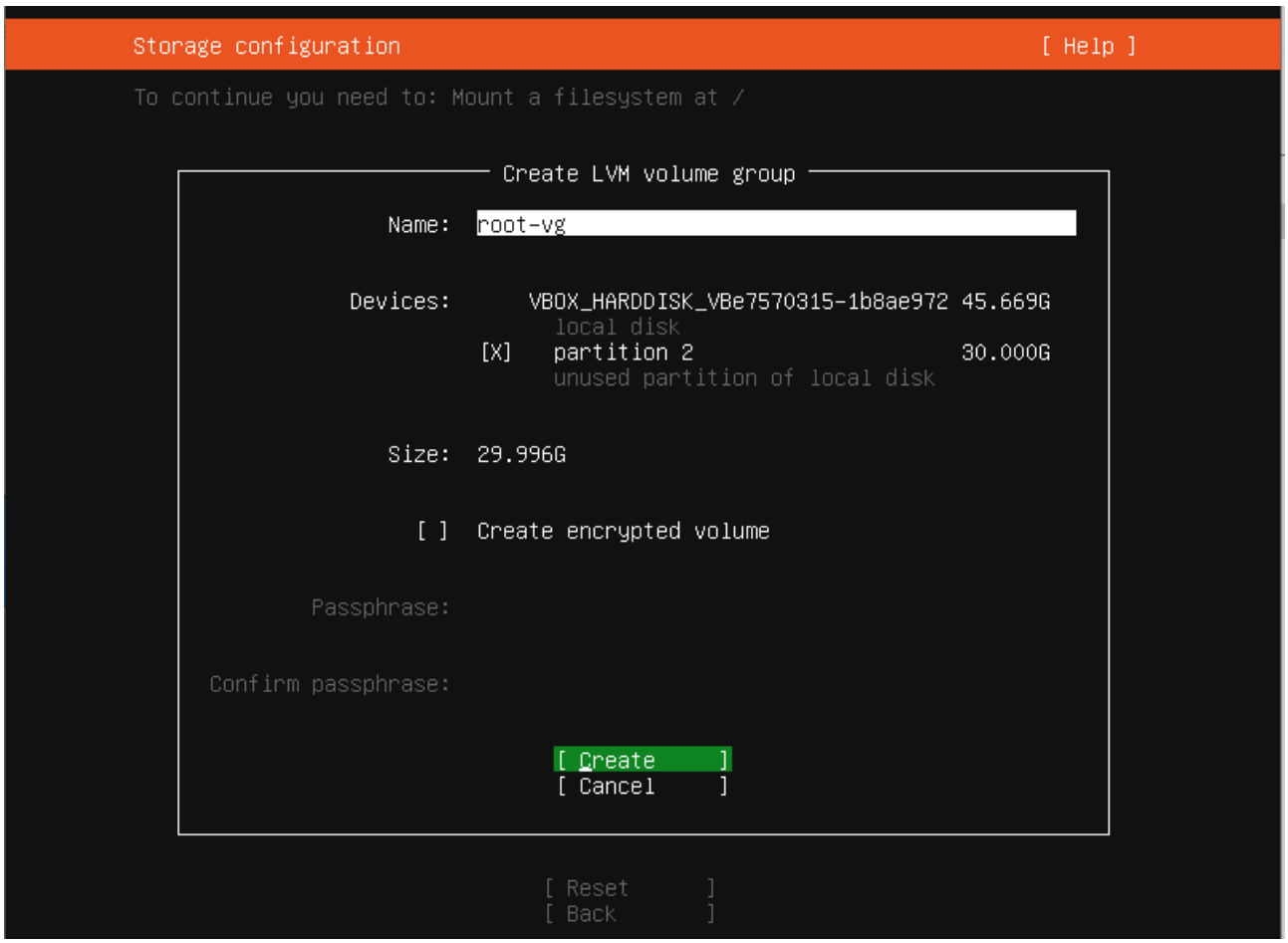
Снимок экрана Б.5 – Выбор диска



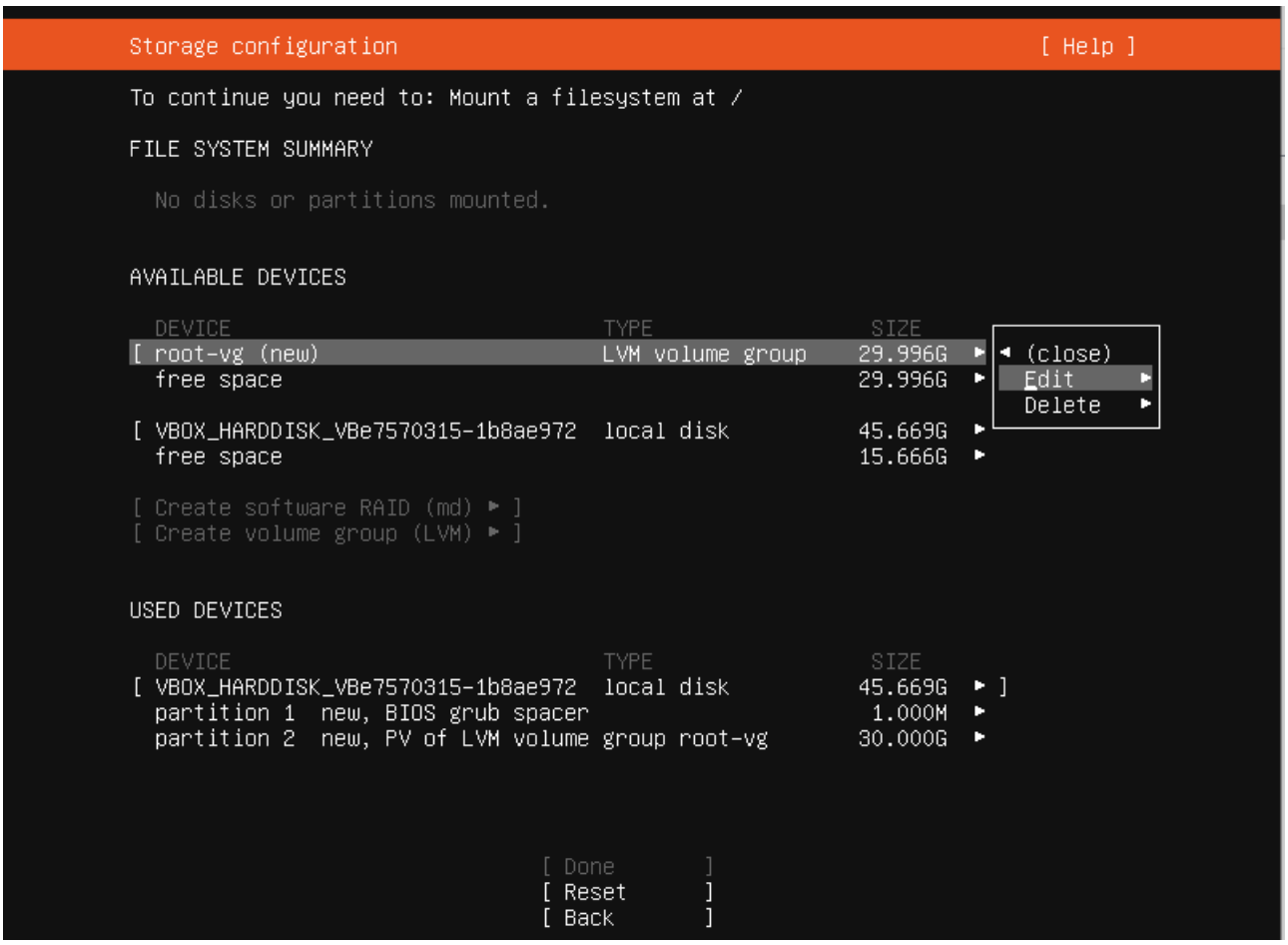
Снимок раздела Б.6 – Создание системного раздела



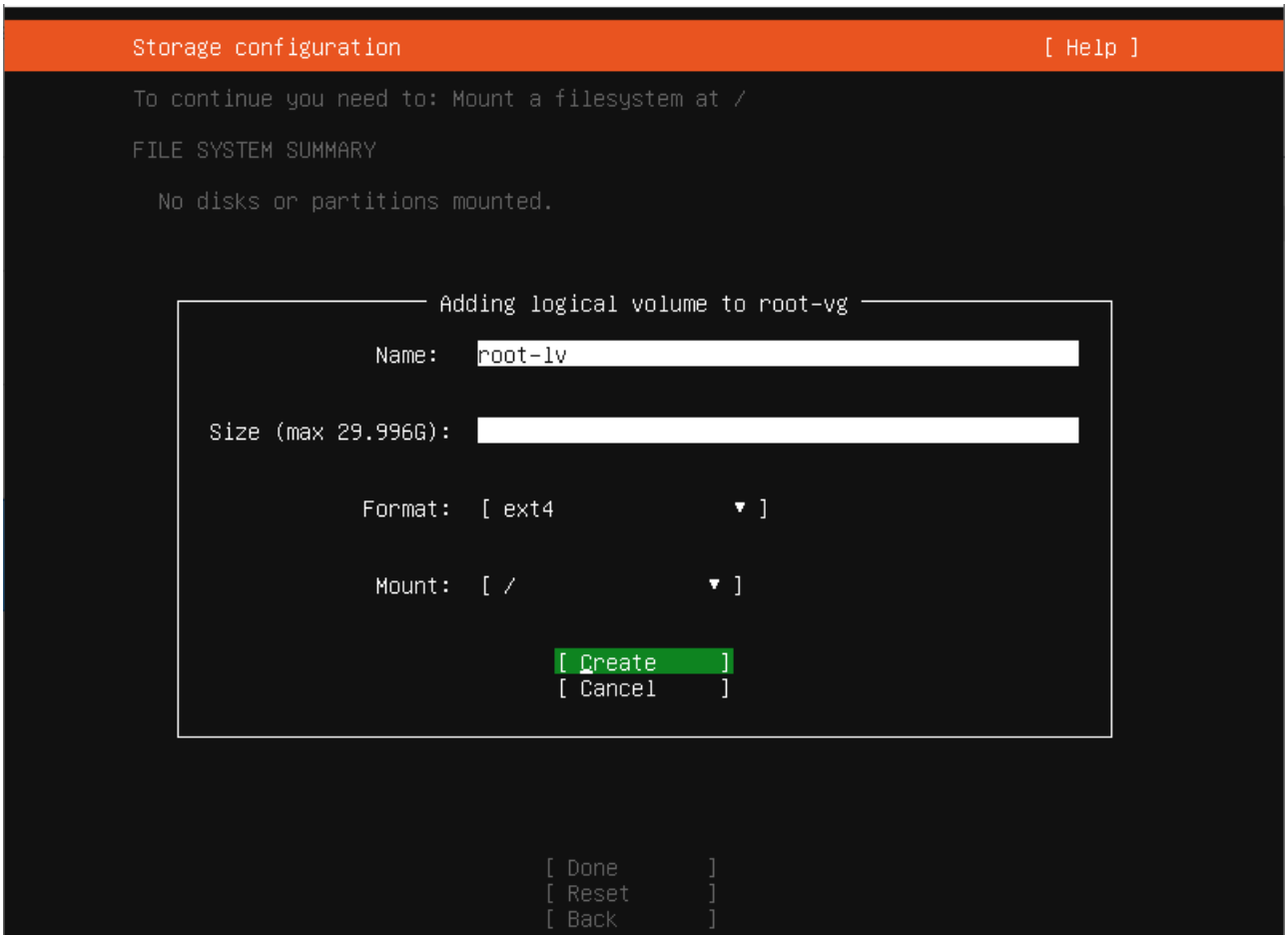
Снимок раздела Б.7 – Создание виртуальной группы



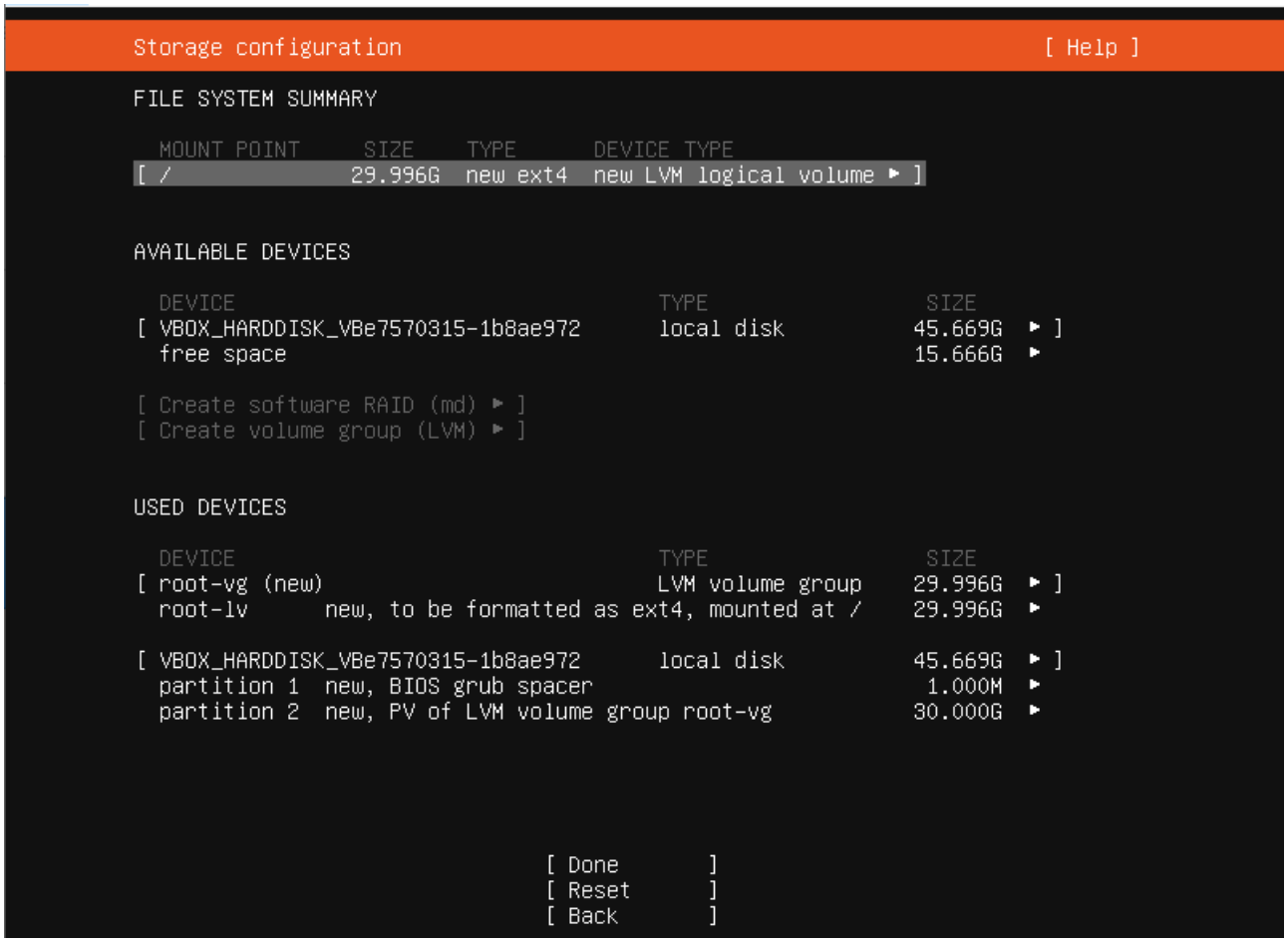
Снимок раздела Б.8 – Параметры виртуальной группы



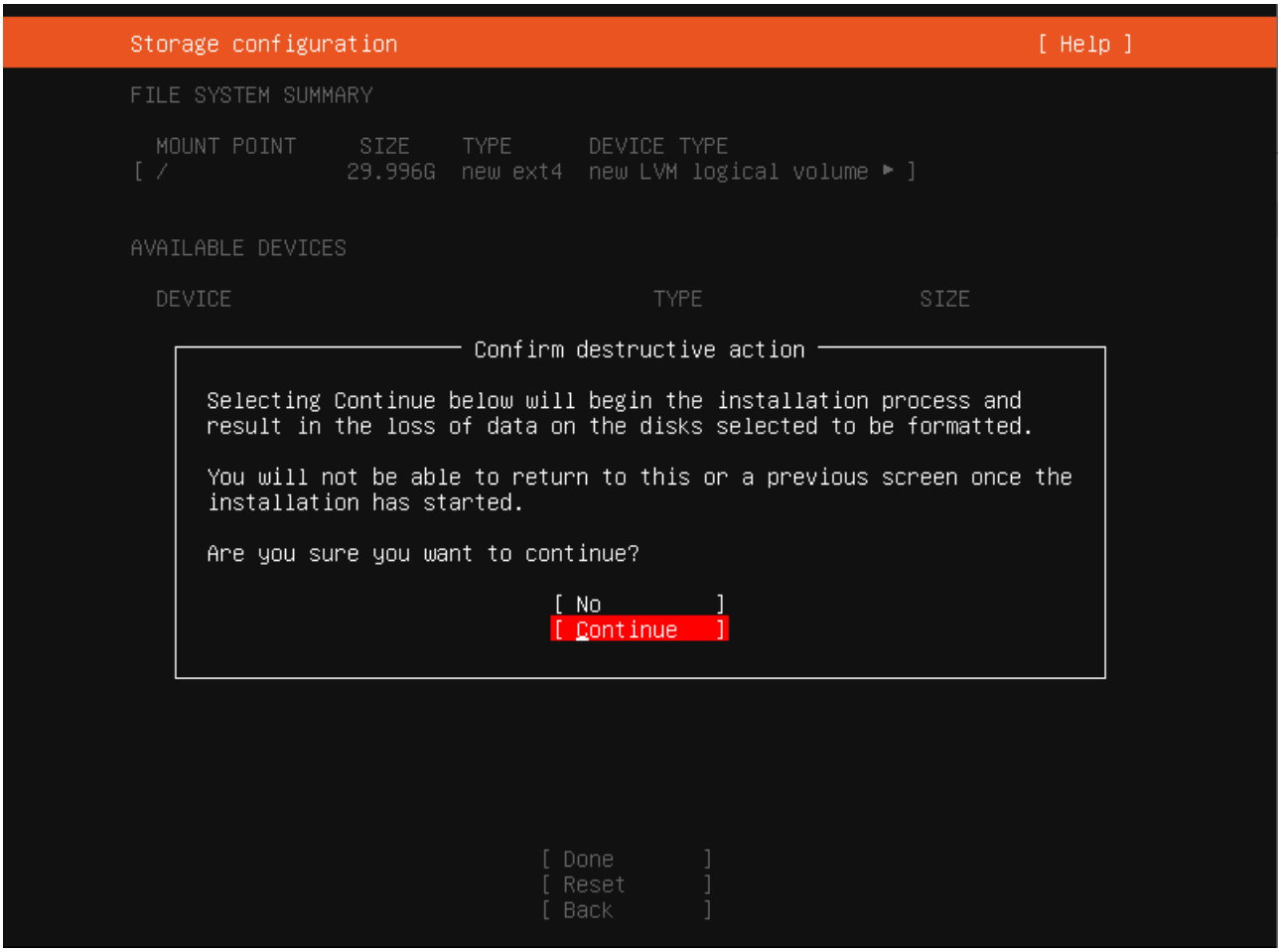
Снимок раздела Б.9 – Создание логической группы



Снимок раздела Б.10 – Параметры логической группы

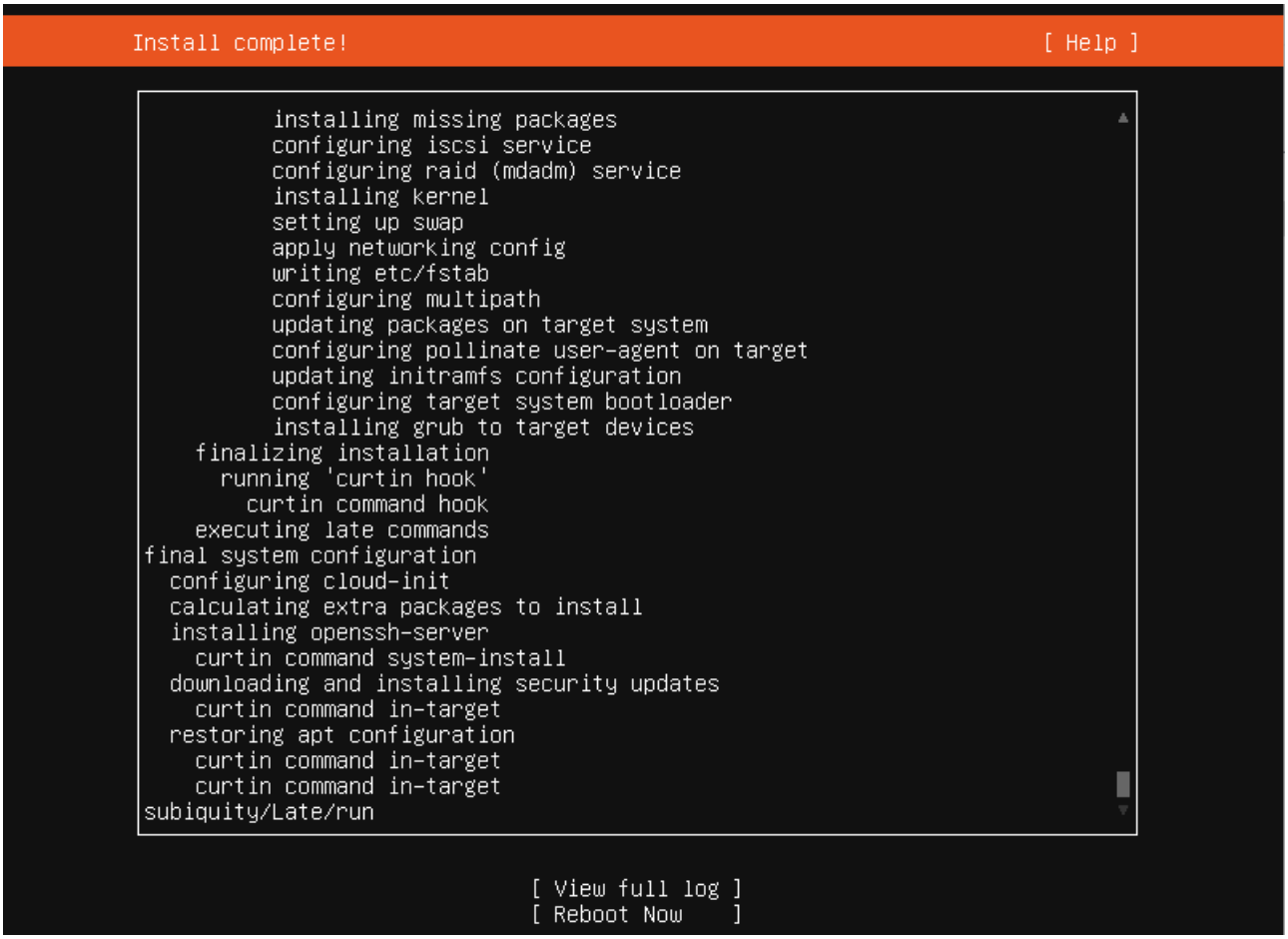


Снимок раздела Б.11 – Итоговая информация по конфигурации дисков



Снимок экрана Б.12 – Завершение разметки

Результат установки операционной системы представлен на снимке экрана Б.13.



Снимок экрана Б.13 – Результат установки

Теперь нужно перезагрузить ОС, выбрав пункт `Reboot Now`.

Следующий этап – установка WAF. Установка WAF начнется после авторизации с учетной записью `waf` (пароль – `waf`) в терминале, представленном на снимке экрана Б.14.

Снимок экрана Б.14 – Терминал для авторизации