Setup ISC BIND 9

Dmitriy Prigoda

https://github.com/D34m0nN0n3

Table of contents

1. Общее описание 3
1.1 Название ролей 3
1.2 Дополнительные материалы 3
2. Роли 4
2.1 Установка и настройка сервера разрешения имен ISC BIND 9 с помощью Ansible 4
2.2 Настройка DNS клиента в CentOS/RHEL 7 & 8 с помощью Ansible 10
2.3 Настройка DNS клиента в CentOS/RHEL 7 & 8 совместно с dnsmasq при помощью Ansible 12
3. Базовая структура проекта с предыдущей конфигурацией 14
4. Стратегия развертывания, пере развертывания 15
4.1 Развертывание 15
4.2 Пере развертывание 15
5. Инструкция сменного персонала DNS 16
5.1 Общее описание 16
5.2 Проверка работоспособности 16
6. 21
7. 21
7.1 Устранение сбоя 22
7.2 Необходимые права sudoers 23
8. Записи о соответствии имени и служебной информации в системе доменных имен 24
8.1 Доменные имена и IP адреса 24
8.2 Типы записей 25
8.3 Форма для генерации записей 25
9. Дополнительные материалы 26
9.1 Ссылки на документацию 26
9.2 Дополнительные ссылки 26

1. Общее описание

Данная документация описывает параметры при использование Ansible Roles для установки и настройки сервера разрешения имен ISC BIND 9, а так же настройки DNS клиента в CentOS/RHEL 7 & 8.

1.1 Название ролей

- infra-system.linux.isc-bind-setup
- infra-system.linux.os.resolv
- infra-system.linux.os.dnsmasq-resolver

> **Подсказка**
>
> Для создания закладки в браузере на данное руководство: `^ Ctrl` + `D`

1.2 Дополнительные материалы

- Git Documentations
- GitHub Documentations
- GitLab Documentations

Последнее обновление: July 10, 2024

2. Роли

2.1 Установка и настройка сервера разрешения имен ISC BIND 9 с помощью Ansible

2.1.1 Общее описание

Domain name server — приложение, предназначенное для ответов на DNS-запросы. По выполняемым функциям DNS-серверы делятся на несколько групп; сервер определённой конфигурации может относиться сразу к нескольким типам:

- *Авторитативный DNS-сервер* — сервер, отвечающий за какую-либо зону.
  - *Мастер* — имеет право на внесение изменений в данные зоны. Обычно зоне соответствует только один мастер-сервер. В случае Microsoft DNS-сервера и его интеграции с Active Directory мастер-серверов может быть несколько (так как репликация изменений осуществляется не средствами DNS-сервера, а средствами Active Directory, за счёт чего обеспечивается равноправность серверов и актуальность данных).
  - *Слейв* - не имеющий права на внесение изменений в данные зоны и получающий сообщения об изменениях от мастер-сервера. В отличие от мастер-сервера, их может быть (практически) неограниченное количество. Слейв также является авторитативным сервером (и пользователь не может различить мастер и слейв, разница появляется только на этапе конфигурирования/внесения изменений в настройки зоны).
- *Кэширующий DNS-сервер* — обслуживает запросы клиентов (получает рекурсивный запрос, выполняет его с помощью нерекурсивных запросов к авторитативным серверам или передаёт рекурсивный запрос вышестоящему DNS-серверу).
- *Перенаправляющий DNS-сервер* — перенаправляет полученные рекурсивные запросы вышестоящему кэширующему серверу в виде рекурсивных запросов. Используется преимущественно для снижения нагрузки на кэширующий DNS-сервер.
- *Корневой DNS-сервер* — сервер, являющийся авторитативным за корневую зону. Общеупотребительных корневых серверов в мире всего 13, их доменные имена находятся в зоне `root-servers.net` и называются `a.root-servers.net, b.root-servers.net, …, m.root-servers.net`. В определённых конфигурациях локальной сети возможна ситуация настройки локальных корневых серверов.
- *Регистрирующий DNS-сервер* - сервер, принимающий динамические обновления от пользователей. Часто совмещается с DHCP-сервером. В Microsoft DNS-сервере при работе на контроллере домена сервер работает в режиме регистрирующего DNS-сервера, принимая от компьютеров домена информацию о соответствии имени и IP-адреса компьютера и обновляя в соответствии с ней данные зоны домена.

При выполнение роли устанавливается и настраивается ISC BIND 9. BIND (Berkeley Internet Name Domain, до этого: Berkeley Internet Name Daemon) — открытая и наиболее распространённая реализация DNS-сервера, обеспечивающая выполнение преобразования DNS-имени в IP-адрес и наоборот. Исполняемый файл-демон сервера BIND называется named. BIND поддерживается организацией Internet Systems Consortium. 10 из 13 корневых серверов DNS работают на BIND, оставшиеся 3 работают на NSD.

Примечание

Роль настраивает два типа конфигураций: авторитативный DNS-сервер и кэширующий DNS-сервер, а так же их комбинацию.

Для информации

Так же включена поддержка raesponse policy zone (зона политики ответа) - это механизм для введения настраиваемой политики на серверах системы доменных имен, чтобы рекурсивные распознаватели возвращали возможно измененные результаты. Зоны и их политика: "rpz.passeddomain.hosts" policy passthru; "rpz.changeddomain.hosts" policy given; "rpz.blockeddomain.hosts" policy nxdomain;

Обратите внимание

Вместе с DNS сервером устанавливаются `node_exporter` и `bind_exporter`. Необходимо не забыть открыть в брандмауэр порты: `9100`, `9119`.

Подсказка

Поддерживается загрузка предыдущей конфигурации из `git` репозитория.

Примечание

Роль поддерживает статические и динамические `inventory`.

Примечание

Для редактирования файлов зон есть скрипт `/usr/local/bin/named-editzone`, который обновляет номер зоны при сохранение, проверяет синтаксис, перечитывает конфигурацию. Перед внесением изменений делает временный файл зоны, с сохранением оригинального файла данных во временную директорию `/tmp/named_zone/`, файлы старше 30 дней удаляются. Поддерживает работу с файлами форматов: raw, text.

2.1.2 Параметры

Название переменной	Тип переменной	Значения по умолчанию	Описание
prometheus_user	string	def in var (prometheus)	Пользователь для exporter'ов.
node_exporter_url	string	undef	URL на пакет с nede_exporter.
bind_exporter_url	string	undef	URL на пакет с bind_exporter.
bind_forwarders	array	undef	Список серверов куда пересылать запросы которые нельзя разрешить.
bind_acl_int	array	undef	Список контроля кому можно разрешать рекурсивные запросы.
bind_acl_int_exclude	array	undef	Список исключений кому можно разрешать рекурсивные запросы.
bind_acl_ext	array	def in var (any)	Список контроля кому можно разрешать записи из зон.
bind_acl_ext_exclude	array	undef	Список исключений кому можно разрешать записи из зон.
bind_acl_change	boolean	undef (false)	Для формирования новых списков контроля.
bind_cont_ph_num	string	undef	Контактный номер телефона администратора.
bind_cont_mail	string	undef	Контактный почтовый адрес администратора.
bind_srv_role	string	undef	Роль сервера: `master` или `slave`.
bind_srv_type	string	def in var (resolver)	Тип сервера: `resolver` - кэширующий, `authorized` - авторитативный, `mixed` - смешанный, `localroot` - локальный root hint.
bind_localroot_resolv	boolean	undef (false)	Заменяет root.hint на локальные stub сервера.
bind_ip_v6_on	boolean	undef (false)	Поддержка протокола IPv6.
bind_max_cache	string	256M	Максимальный размер кеша в Мб.
bind_max_journal	string	500M	Максимальный размер журнала в Мб.
alt_tranfer_src	boolean	undef (false)	Задает использования альтернативного интерфейса для передачи зон.
mf_format	string	undef	Задает формат зон: map, raw, text.
zero_ttl	boolean	undef	При возврате авторитетных отрицательных ответов на запросы SOA установите TTL записи SOA.
trust_clients	array	localhost	Задает список клиентов по умолчанию.
trust_servers	array	'ansible_all_ipv4_addresses'	Задает список серверов по умолчанию.
empty_zone_name	string	def in var (example.com)	Задает имя первой зоны.

Название переменной	Тип переменной	Значения по умолчанию	Описание
bind_backup_dir	string	def in var (/var/tmp/)	Директория для резервного копирования конфигурации.
bind_restore_last_conf	boolean	def in var (false)	Задает нужно ли восстановить предыдущую конфигурацию.
remote_git_repo	string	undef	Репозиторий от куда загружать предыдущую конфигурацию.
local_git_repo	string	def in var (/var/tmp/isc-bind-files)	Куда временно сохранить предыдущую конфигурацию.
bind_listen_ipv4	string	undef	Адрес IPv4 для статической конфигурации. Для заранее определенного состояния.
bind_listen_ipv6	string	undef	Адрес IPv6 для статической конфигурации. Для заранее определенного состояния.
bind_fqdn	string	undef	Имя хоста для статической конфигурации. Для заранее определенного состояния.

Для информации

Сценарий поддерживает масштабирование, добавляя новые slave сервера. Для этого хост настраивается как вторичный, и добавляется в еще одну группу `new_slaves` или ему присваивается переменная `new_slave=True`. После выполнения сценария переменную необходимо удалить, а хост исключить из группы.

2.1.3 Теги

Тег	Описание
bind_setupe	Установка `BIND`
bind_exporter_prometheus	Установка `exporter`
bind_configure	Создание конфигурационных файлов
bind_copy_configs	Копирование конфигурационных файлов
bind_ip_v6_enable	Настройка IPv6
bind_create_zone	Создание первой зоны
bind_restore_from_git	Копирование предыдущей конфигурации

2.1.4 Примеры

inventory/hosts

```
# Переменные которые необходимо заменить на свои значения указаны в '< >', значения указываются без них.
# Все узлы объединяются в группы ИС/АСУ.
# Данный фаил формируется в формате INI.
[all:vars]
bind_cont_ph_num='+7(000)111-22-33'
bind_cont_mail='mail@example.com'
bind_forwarders=['192.168.2.1','192.168.2.2']
alt_tranfer_src=True
bind_acl_int=['192.168.1.0/24','192.168.2.0/24']
bind_srv_type='mixed'
remote_git_repo='git@github.com:D34m0nN0n3/backup-isc-bind.git'
node_exporter_url='https://github.com/prometheus/node_exporter/releases/download/v1.1.1/node_exporter-1.1.1.linux-amd64.tar.gz'
bind_exporter_url='https://github.com/prometheus-community/bind_exporter/releases/download/v0.4.0/bind_exporter-0.4.0.linux-amd64.tar.gz'
ansible_connection=ssh

[master]
bootstrap.lab ansible_connection=local

[slaves]
rhel7.lab ansible_ssh_host=192.168.1.101
rhel8.lab ansible_ssh_host=192.168.1.102
```

2.1.5 Дополнительные материалы

- BIND 9 Administrator Reference Manual

Последнее обновление: July 10, 2024

2.2 Настройка DNS клиента в CentOS/RHEL 7 & 8 с помощью Ansible

2.2.1 Общее описание

DNS-клиент — программа (или модуль в программе), обеспечивающая определение IP-адреса узла по его полному имени. Для того чтобы программа-определитель могла выполнять свою задачу ей должен быть предоставлен доступ к серверам доменных имен. В фале `resolv.conf` содержатся адреса серверов имен, к которым имеет доступ данная система. После записей search идут записи `nameserver`, если таковые имеются. Для каждого сервера имен, к которому имеет доступ данная система, вводиься ключевое слово `nameserver` и IP-адрес.

> **Примечание**
>
> В процессе выполнения роли отключается настройка управление DNS через `NetworkManager`, настройка производится только через `resolv.conf`.

2.2.2 Параметры

Название переменной	Тип переменной	Значения по умолчанию	Описание
dns_attempts	string	3	Число запросов посылаемых клиентом до переключения на другой сервер при не ответе.
dns_timeout	string	2	Время в секундах ожидания ответа от сервера до повторной попытки.
dns_ndots	string	1	Число точек в аргументе, чтобы клиент произвел поиск по буквальному имени, прежде чем использовать список поиска.
dns_domain	string	'ansible_domain'	Задает локальное имя клиента.
dns_search	array	'ansible_domain'	Список доменов для поиска, при не полном указание имени.
nameserver	array	'1.1.1.1','8.8.8.8'	Список DNS серверов.

2.2.3 Примеры

inventory/hosts

```
[example-servers]
<host_name> ansible_ssh_host=<host_ip> ansible_ssh_user=<user_name_for_connect>

[example-servers:vars]
ansible_connection=ssh
dns_search=['foo.example.com','bar.example.com']
nameserver=['1.1.1.1','1.0.0.1','8.8.8.8','8.8.4.4']
```

2.2.4 Дополнительные материалы

- Linux manual page - resolv.conf(5)
- DNS configuration with Ansible

Последнее обновление: July 10, 2024

2.3 Настройка DNS клиента в CentOS/RHEL 7 & 8 совместно с dnsmasq при помощью Ansible

2.3.1 Общее описание

DNS-клиент — программа (или модуль в программе), обеспечивающая определение IP-адреса узла по его полному имени. Dnsmasq — легковесный и быстроконфигурируемый DNS, является более современным подходом к настройке сети.

Dnsmasq предоставляет следующие возможности:

- Простая конфигурация DNS-машин за межсетевым экраном, независимо от особенностей и доступности DNS-серверов провайдера.
- Мгновенная передача клиентам информации о недоступности сайта, если внешнее соединение прервано.
- Может переопределить другие имена для глобальных IP-адресов без необходимости исправлять файл /etc/hosts на каждой машине.
- Кэширует интернет-адреса (A-записи и записи AAAA) и PTR-записи, снижая нагрузки на внешние серверы и повышая производительность (особенно на модемных соединениях).
- Пользователи могут настроить Dnsmasq для отправки запросов для определённых доменов на обработку внешним серверам.
- Поддерживает MX-записи и может вернуть MX-записи для любой или всех локальных машин.
- Поддерживает NAPTR-запись, что позволяет использовать регулярные выражения, основанные на переписывании доменных имён, которые затем могут быть использованы в качестве URI, дополнительных доменных имён для поиска.
- Некоторые провайдеры переписывают NXDOMAIN-ответы (домен не существует) от DNS-серверов. Это заставляет веб-браузеры искать страницы в домене, который не существует. Dnsmasq может отфильтровать такие записи.

Примечание

В процессе выполнения роли отключается настройка управление DNS через `NetworkManager`, настройка производится только через `dnsmasq.conf` и `resolv.dnsmasq`.

Внимание, при ошибке

Так как DNSMASQ является альтернативным DNS сервером, данное решение не следует применять на серверах с другими DNS службами.

2.3.2 Параметры

Название переменной	Тип переменной	Значения по умолчанию	Описание
dns_cache	string	'1000'	Количество кешируемых записей.
dns_domain	string	'ansible_domain'	Задает локальное имя клиента.
nameserver	array	'1.1.1.1','8.8.8.8'	Список DNS серверов.

2.3.3 Примеры

inventory/hosts

```
[example-servers]
<host_name> ansible_ssh_host=<host_ip> ansible_ssh_user=<user_name_for_connect>

[example-servers:vars]
ansible_connection=ssh
dns_cache='1111'
nameserver=['1.1.1.1','1.0.0.1','8.8.8.8','8.8.4.4']
```

2.3.4 Дополнительные материалы

- Linux manual page - resolv.conf(5)
- DNSMASQ

Последнее обновление: July 10, 2024

3. Базовая структура проекта с предыдущей конфигурацией

```
.
|
|-masters
|       |_ ...
|
|_external-include-extanded.conf
|
|_internal-include-extanded.conf
|
|_named.zones
```

Директории и файлы	Описание
masters	Директория с файлами мастер зоне
external-include-extanded.conf	Конфигурационный файл с описанием специфичных зон для внешних клиентов.
internal-include-extanded.conf	Конфигурационный файл с описанием специфичных зон для внутренних клиентов.
named.zones	Конфигурационный файл с описанием зон с мастер сервера.

Последнее обновление: July 10, 2024

4. Стратегия развертывания, пере развертывания

Порядок действий необходимых для достижения конечного результата.

4.1 Развертывание

Проводится установка и настройка нового комплекса или тестового с восстановлением конфигурации, без замены узлов находящихся в эксплуатации.

Порядок действий:

1. Развернуть на узлах ОС и настроить сеть.
2. Описать параметры узлов в `inventory`.
3. Выполнить роль по установке и настройке.
4. Проверить корректность и работоспособность конфигурации.

4.2 Пере развертывание

Проводится установка и настройка комплекса с восстановлением конфигурации, с замены узлов находящихся в эксплуатации. Предполагается два вариант: единовременная замена всех узлов (динамическая на основе состояния параметров хоста), последовательна замена узлов (статически описанное состояние переменных для узлов).

4.2.1 Единовременная замена всех узлов (требуется простой сервиса).

1. Развернуть на узлах ОС и настроить сеть. Аналогично заменяемому комплексу.
2. Описать параметры узлов в `inventory`.
3. Выполнить роль по установке и настройке.
4. Проверить корректность и работоспособность конфигурации.
5. Выключение узлов или отключение от сети заменяемого комплекса.
6. На новых узлах настраивается имена узлов и параметры сети такие же как на заменяемом комплексе.
7. Выполнить роль по установке и настройке повторно с параметром `bind_acl_change = True`.
8. Проверить корректность и работоспособность конфигурации.

4.2.2 Последовательна замена узлов (без простоя сервиса).

1. Развернуть на узлах ОС и настроить сеть. Аналогично заменяемому комплексу.
2. Описать параметры узлов в `inventory`.
3. Выполнить роль по установке и настройке.
4. Проверить корректность и работоспособность конфигурации.
5. Определение переменных для переопределения значений конфигурационных файлах.
6. Выполнить роль по установке и настройке повторно с параметром `bind_acl_change = True`.
7. Проверить корректность и работоспособность конфигурации.
8. Заменить настройки (выполняется последовательно на узлах).
   8.1 Выключение узла или отключение от сети заменяемого комплекса. 8.2 На новых узлах настраивается имена узлов и параметры сети такие же как на заменяемом комплексе.

Последнее обновление: July 10, 2024

5. Инструкция сменного персонала DNS

5.1 Общее описание

Сбои разделены на два типа операционные ошибки и ошибки конфигурации.

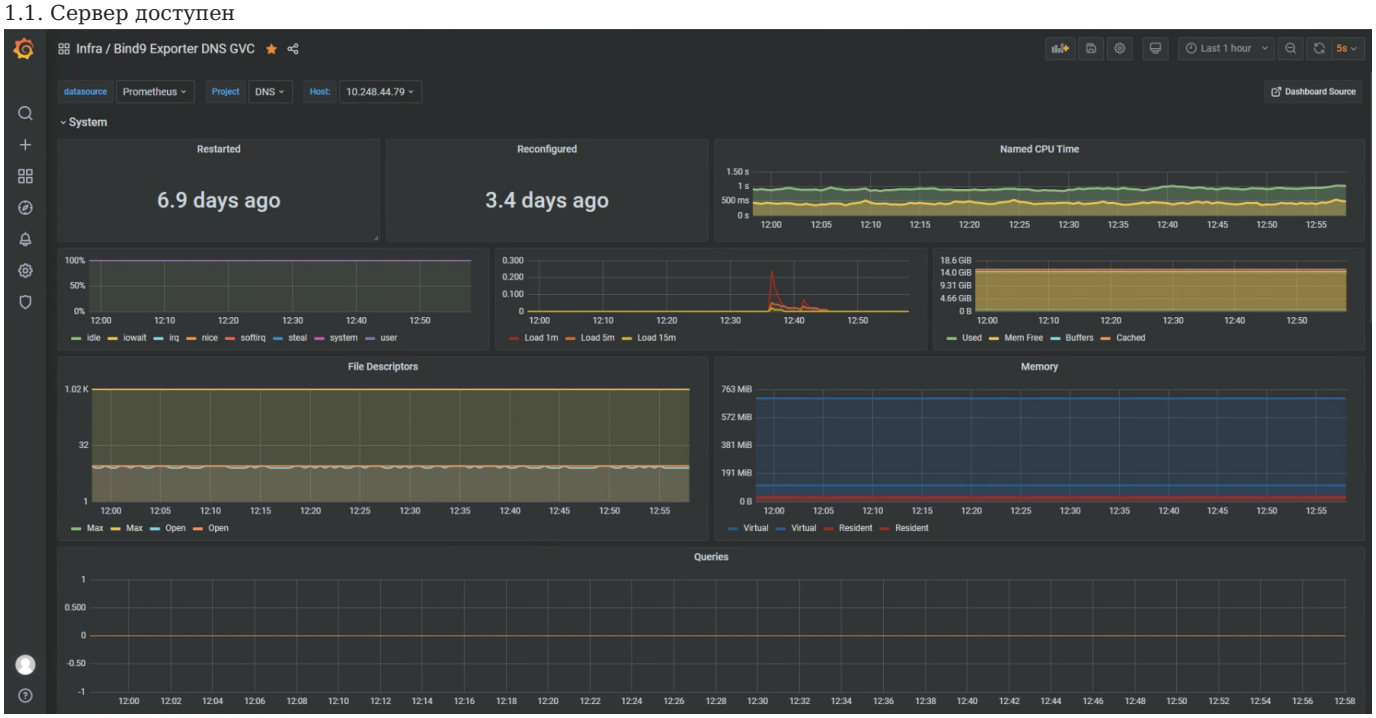
- Операционные ошибки - возникают в процессе выполнения различных операций, таких как формирование "негативного кеша". Устраняются без привлечения администратора.
- Ошибки конфигурации - связаны с неправильным внесением изменений в конфигурационные файлы. Устраняются с привлечением администратора.

Ниже приведено описание нахождения и устранения сбоев.

5.2 Проверка работоспособности

5.2.1 1. Необходимо проверить доступность и сбор метрик в Grafana.

1.1. Сервер доступен

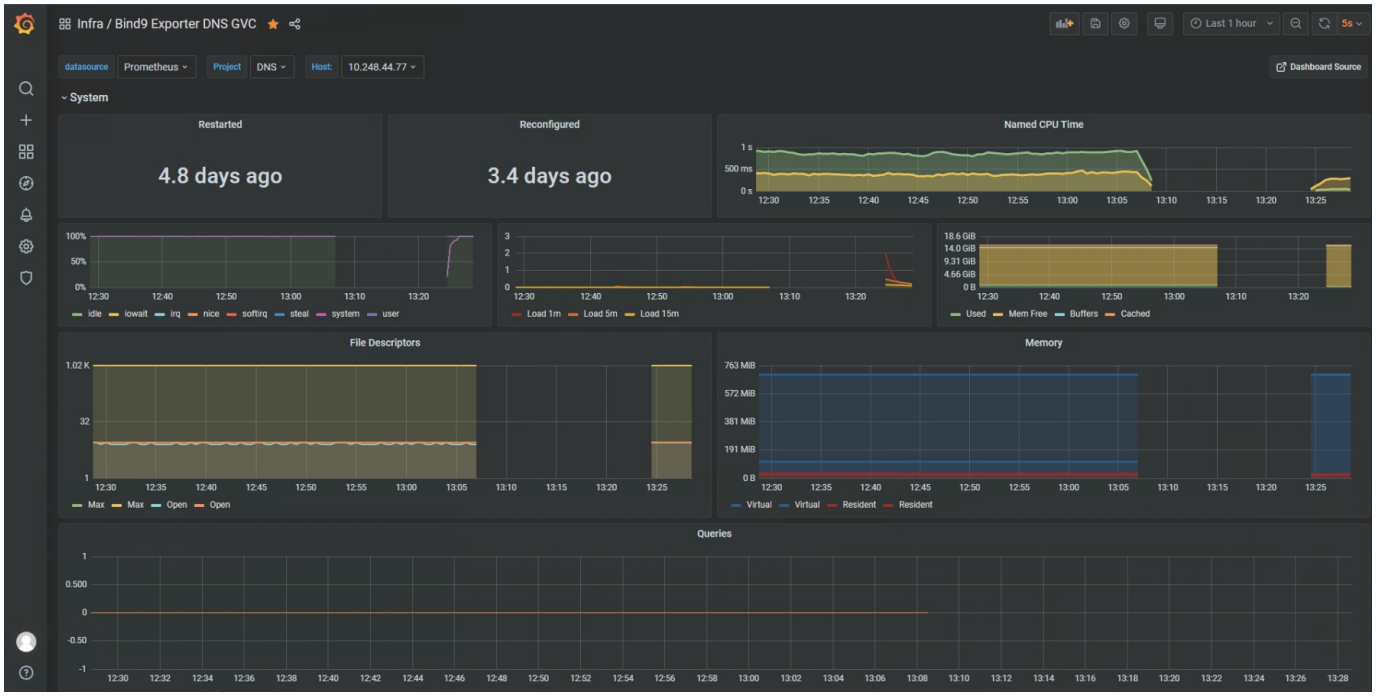


Если сервер доступен необходимо выполнить разрешение имен.

1.2. Сервер не доступен

Если сервер не доступен следует проверить проверить доступность по сети и зайти на сервер по SSH, и проверить состояние сервиса. При недоступности по сети проверить состояние сервера или виртуальной машины, должна находится в статусе `PowerOn`.

1.3. Убедившись в возобновление доступности



Необходимо выполнить разрешение имен.

5.2.2 2. Проверка состояния сервиса.

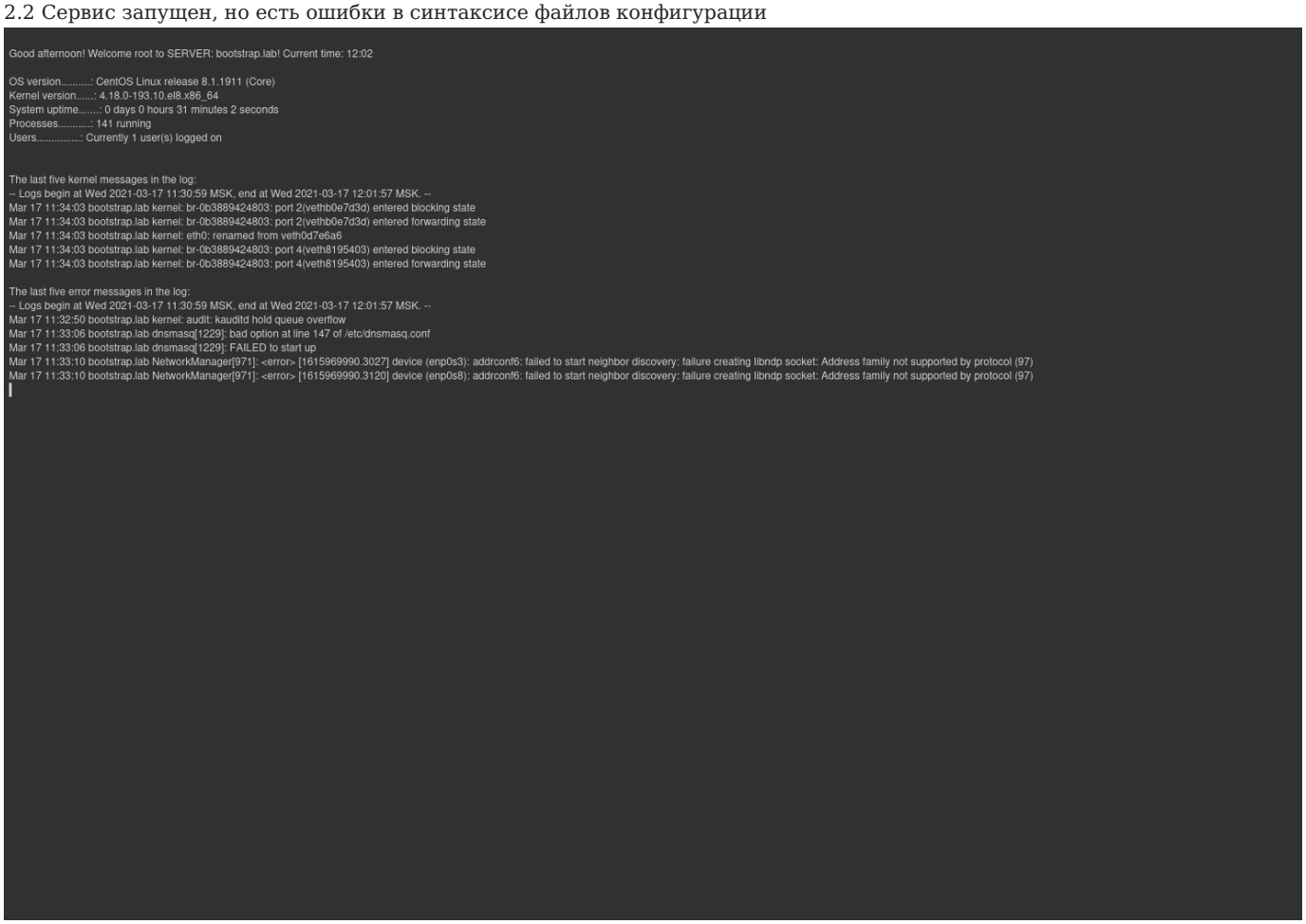
Для проверки необходимо запустить сценарий проверки: `named-healthy-check`. Так как сервис может быть запушен в разных режимах с подменой дерева каталогов (unit name: named-chroot.service) или без (unit name: named.service), сценарий проверяет какой из них активирован. Выводя состояние для каждого режима: ENABLE и DISABLE. Выполняет

проверки: запущен ли сервис, валидность синтаксиса файлов конфигурации, валидность файлов зон. Выводя состояние для каждой проверки: PASS и INVALID. Выводи 7 последних сообщений из системного журнала.

2.1 Сервис запещен и работает, ошибок не зафиксировано

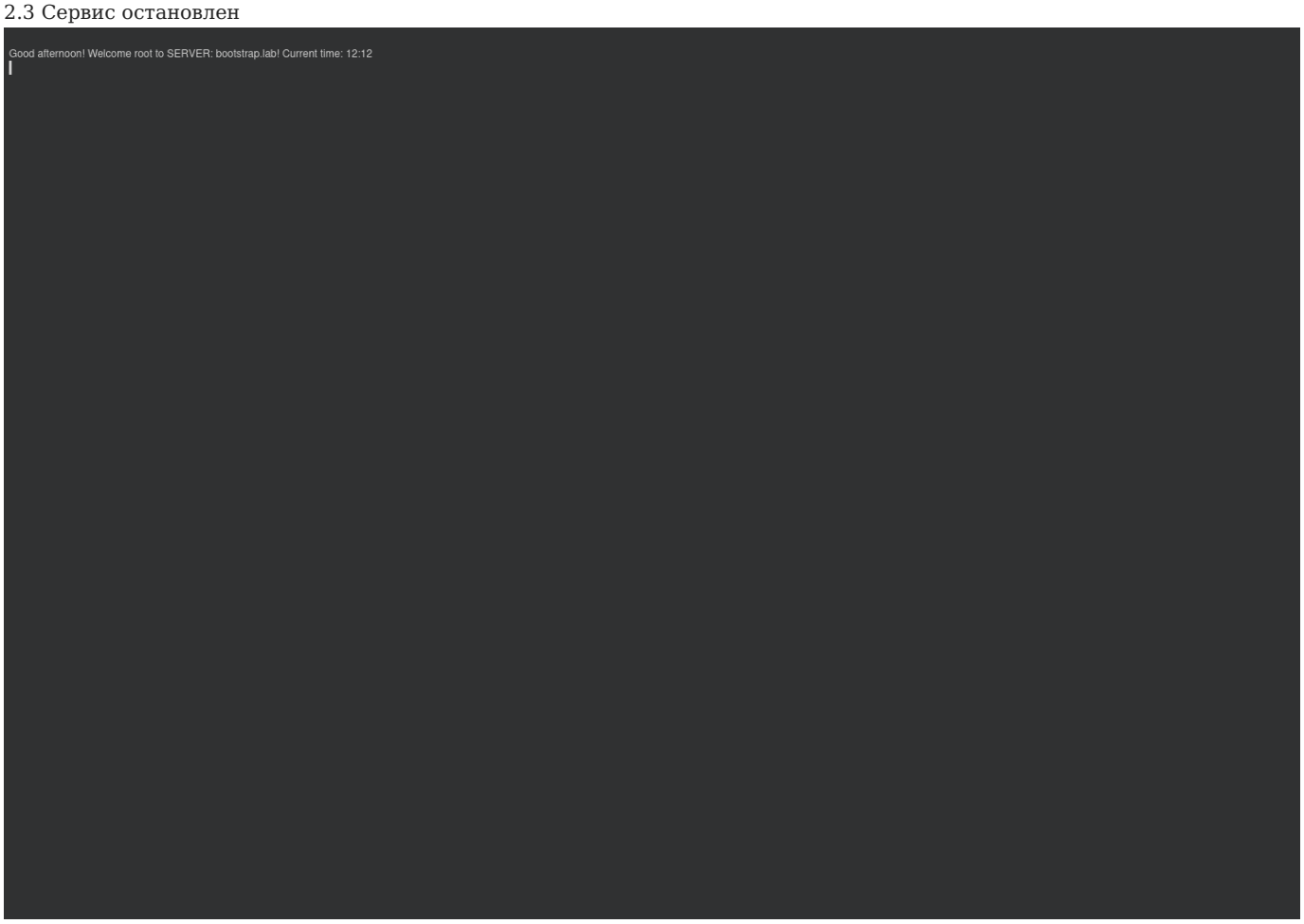
Необходимо выполнить разрешение имен.

2.2 Сервис запущен, но есть ошибки в синтаксисе файлов конфигурации



Необходимо запустить сбор диагностической информации и сообщить об этом ответственным администраторам.

2.3 Сервис остановлен



Необходимо выполнить запуск сервиса.

5.2.3 3. Проверка разрешения имен

Для проверки используется программа `nslookup`, в качестве аргументов необходимо передать: тип запроса, разрешаемое имя, сервер системы имен (DNS). Перед началом проверки необходимо сбросить DNS кеш на клиенте, выполнив команду:

> **Подсказка**
>
> В Linux `nslookup` требуется установить отдельно.

Очистка кеша на клиенте.

Для Windows

```
ipconfig \flushdns
```

Для Linux

```
sudo systemctl restart nscd.service
```

6.

7.

Выполнить разрешение имен:

Разрешения имен

```
nslookup -type=any <имя_записи> <ip_сервера>
```

Разрешения имен с расширенным выводом

```
nslookup -debug -type=any <имя_записи> <ip_сервера>
```

Если ответ не получен или содержит не верные данные, необходимо сбросить кеш на сервере. Или по согласованию с ответственным администратором выполнить перезагрузить конфигурацию.

7.0.1 4. Сбор диагностической информации

Для сбора диагностической информации необходимо запустить утилиту `sosreport`.

Пример

```
sudo sosreport  --log-size 25 --batch
```

Для информации

Так как сбор диагностической информации занимает длительное время, то запускать утилиту рекомендуется в `tmux` или как фоновое задание.

Для выполнения задания в `tmux`

```
tmux new -d -s SosReport && tmux send-keys -t SosReport.0 "sudo sosreport  --log-size 25 --batch"
```

Для выполнения задания в фоновом режиме

```
sudo sosreport  --log-size 25 --batch &
```

7.1 Устранение сбоя

7.1.1 1. Очистка кеша на сервере

После выполнения очистки кеша необходимо проверить текущий статус сервиса и выполнить разрешение имен.

Очистка всего кеша

```
sudo rndc flush
```

Проверка статуса

```
sudo rndc status
```

В последняя строка должна содержать: 'server is up and running'

7.1.2 2. Перезагрузка конфигурации

Пример

```
sudo rndc reload
```

Выполнить разрешение имен.

7.1.3 3. Запуск и перезапуск службы

В качестве следует передать имя службы со статусом ENABLE полученном при запуске сценария проверки `named-healthy-check`.

Запуск

```
sudo systemctl start <unit_name> && sudo systemctl status <unit_name>
```

Перезапуск

```
sudo systemctl restart <unit_name> && sudo systemctl status <unit_name>
```

Проверить журнал на наличие ошибок:

Пример

```
journalctl -xe _COMM=systemd -u <unit_name> -n 3 --no-pager
```

Если вывод заканчивается строкой: '-- The start-up result is done.', выполнить разрешение имен.

Если вывод заканчивается стракой: '-- The result is failed.', запустить сбор диагностической информации и сообщить об этом ответственным администраторам.

7.2 Необходимые права sudoers

Предлагаемое решение

Cmnd_Alias SERVICES = /usr/bin/systemctl (start|restart|status) (named|named-chroot).service, /usr/sbin/rndc (status|flus|reload), /usr/sbin/sosreport

Последнее обновление: July 10, 2024

8. Записи о соответствии имени и служебной информации в системе доменных имен

8.1 Доменные имена и IP адреса

- Доменное имя — символьное имя, служащее для идентификации областей, которые являются единицами административной автономии в сети.
- Доменная зона — совокупность доменных имён определённого уровня, входящих в конкретный домен.
- FQDN («полностью определённое имя домена») — имя домена, не имеющее неоднозначностей в определении. Включает в себя имена всех родительских доменов иерархии DNS.

Пример FQDN доменного имени пятого уровня — `sample.gtw-02.office4.example.com.`

- sample — пятый уровень;
- gtw-02 — четвертый уровень;
- office4 — третий уровень;
- example — второй уровень;
- com — первый (верхний) уровень;
- .(точка) — нулевой (корневой) уровень.

- IP адрес — уникальный числовой идентификатор устройства в компьютерной сети, работающий по протоколу TCP/IP.

IP адреса используемые в локальных сетях

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

- NAT (Network Address Translation) — это механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов.

Обратите внимание

Для доступа из Интернета на сервер в локальной сети необходимо указать преобразованный NAT адрес.

8.2 Типы записей

Тип	Описание	RFC
A	Адресная запись, соответствие между именем и IP-адресом	1035
CNAME	Каноническое имя для псевдонима (одноуровневая переадресация)	1035
MX	Адрес почтового шлюза для домена. Состоит из двух частей — приоритета (чем число больше, тем ниже приоритет), и адреса узла	1035
PTR	Соответствие адреса имени — обратное соответствие для A	1035
RP	Ответственный	1183
SRV	Указание на местоположение серверов для сервисов	2782
SSHFP	Отпечаток ключа SSH	4255
TXT	Запись произвольных двоичных данных, до 255 байт в размере	1035

8.3 Форма для генерации записей

> ⚠ **Обратите внимание**
>
> Столбцы помеченные `*` обязательны к заполнению.

> ⚠ **Обратите внимание**
>
> TTL (время жизни). Задаётся в секундах, типичное значение составляет 86 400 секунд, то есть 24 часа.

Ведите данные...

Имя *	Домен *	TTL	Тип	Зн
			A ˅	

Добавить веденные данные в таблицу

Данные для добавления в DNS...

Forward records	Reverse records	

Сохранить таблицу в CSV

Последнее обновление: July 10, 2024

9. Дополнительные материалы

9.1 Ссылки на документацию

- RHEL 7 Documentations
- RHEL 8 Documentations
- Ansible Documentations

9.2 Дополнительные ссылки

- Ansible installation guide
- Ansible tags
- BIND 9 Administrator Reference Manual
- Manually configuring the /etc/resolv.conf file

Последнее обновление: July 10, 2024