



МИНСТРОЙ РОССИИ
Федеральное автономное учреждение «РосКапСтрой»

УТВЕРЖДАЮ

Заместитель директора
ФАУ «РосКапСтрой»

_____ И.В.Кубышкин

МП

«_____» _____ 2022 г.

Система сбора сведений в чрезвычайных ситуациях и мониторинга

восстановительных работ

Руководство по развертыванию

Аннотация

Настоящий документ содержит инструкцию по установке и настройке Системы сбора сведений в чрезвычайных ситуациях и мониторинга восстановительных работ (далее – Система).

Содержание

1	Перечень сокращений.....	4
2	Состав и техническое обеспечение	5
2.1	Техническое обеспечение	5
2.2	Инфраструктурные назначения серверов.....	11
2.3	Общее программное обеспечение.....	11
2.4	Специальное программное обеспечение	14
2.5	Компоненты программного обеспечения	15
3	Установка подсистем	16
3.1	Требования к рабочему месту оператора развертывания.....	16
3.2	Подготовка рабочего места оператора развертывания	16
3.3	Описание структуры пакета автоматизированного развертывания Системы.....	17
3.4	Планирование инфраструктуры	19
3.5	Развертывание на инфраструктуру	20
3.6	Обновление системы	20
3.7	Горизонтальное масштабирование Системы.....	21
3.8	Откат к предыдущей версии.....	22
3.9	Тестирование системы развертывания	22
4	Настройка подсистем.....	23
5	Проверка работоспособности подсистем	24
6	Системные сообщения.....	25

1 Перечень сокращений

Перечень сокращений и терминов приведен в таблицах 1 и 2.

Таблица 1 – Перечень терминов.

Термин	Определение
Система	Система сбора сведений в чрезвычайных ситуациях и мониторинга восстановительных работ
Специальное программное обеспечение	Программа для электронно-вычислительных машин, создаваемая в рамках создания или развития Системы, в том числе посредством настройки (адаптации) программного обеспечения, принадлежащего третьим лицам или Исполнителю.

Таблица 2 – Перечень сокращений.

Сокращение	Описание
АРМ	Автоматизированное рабочее место, программно-технический комплекс АС, предназначенный для автоматизации деятельности определенного вида
БД	База данных
ИС	Информационная система
ОС	Операционная система - комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера (сервера) и организации взаимодействия с пользователем
ПО	Программное обеспечение, совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ
СУБД	Система управления базой данных, совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных
СМЭВ	Единая система межведомственного электронного взаимодействия

2 Состав и техническое обеспечение

2.1 Техническое обеспечение

В таблице 3 представлен состав и основные параметры серверного технического обеспечения для Системы сбора сведений в чрезвычайных ситуациях и мониторинга восстановительных работ.

Таблица 3 – Состав и параметры серверного технического обеспечения.

	Web Server (NGINX) & Cache DB (Redis)	Application Server (Accent)	Relational DB (PostgreSQL)	Relational DB (PgBouncer)	Non-Relational DB (MongoDB)	Message Broker (Kafka)	Monitoring & Logging Server (Prometheus & Grafana)	Workflow Manager (Airflow)
имя сервера (префикс) -hunt	gzhi-web01	gzhi-app01	gzhi-pgsql01	gzhi-pgb01	gzhi-mg01	gzhi-kafka01	gzhi-mon01	gzhi-airflow01
Тип выделяемого виртуального сервера (например, файловый сервер, сервер приложений, сервер баз данных, сервер доступа и пр.)	Сервер с веб-клиентом	Сервер приложений	Сервер базы данных postgresql	Сервер базы данных pgbouncer	Сервер базы данных MongoDB	Сервер шины данных Kafka	Сервер мониторинга и логирования	Сервер планировщик

	Web Server (NGINX) & Cache DB (Redis)	Application Server (Accent)	Relational DB (PostgreSQL)	Relational DB (PgBouncer)	Non-Relational DB (MongoDB)	Message Broker (Kafka)	Monitoring & Logging Server (Prometheus & Grafana)	Workflow Manager (Airflow)
Операционная система, устанавливаемая на виртуальный сервер	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64 (Orel), Astra Linux Special Edition 1.7	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64 (Orel), Astra	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64 (Orel), Astra	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64 (Orel), Astra Linux Special

	Web Server (NGINX) & Cache DB (Redis)	Application Server (Accent)	Relational DB (PostgreSQL)	Relational DB (PgBouncer)	Non-Relational DB (MongoDB)	Message Broker (Kafka)	Monitoring & Logging Server (Prometheus & Grafana)	Workflow Manager (Airflow)
	2.12 x86_64 (Orel), Astra Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64	(Orel), Astra Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64	x86_64, Alt Server 10 x86_64	Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64	x86_64 (Orel), Astra Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64	2.12 x86_64 (Orel), Astra Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64	Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64	Edition 1.7 x86_64, Alt Server 10 x86_64
Количество ядер каждого виртуального процессора, шт.	2	4	8	4	4	4	4	4
Объем оперативной памяти, ГБ	8G	8G	64G	8G	16G	32G	8G	16G
Количество виртуальных жестких дисков, шт.	1	2	3	1	2	2	2	2

	Web Server (NGINX) & Cache DB (Redis)	Application Server (Accent)	Relational DB (PostgreSQL)	Relational DB (PgBouncer)	Non-Relational DB (MongoDB)	Message Broker (Kafka)	Monitoring & Logging Server (Prometheus & Grafana)	Workflow Manager (Airflow)
Объем 1-го виртуального жесткого диска, ГБ. Разбиение диска на разделы	100 (SSD)	100 (SSD)	100 (SSD)	100 (SSD)	100 (SSD)	100 (SSD)	100 (SSD)	100 (SSD)
/boot	1G (xfs)	1G (xfs)	1G (xfs)	1G (xfs)	1G (xfs)	1G (xfs)	1G (xfs)	1G (xfs)
swap	8G	8G	8G	8G	8G	8G	8G	16G
/	91G (xfs)	91G (xfs)	91G (xfs)	91G (xfs)	91G (xfs)	91G (xfs)	91G (xfs)	91G (xfs)
Объем 2-го виртуального жесткого диска, ГБ Разбиение диска на разделы:		500 (HDD)	500 (SSD)		250 (SSD)	250 (HDD)	250 (HDD)	250 (HDD)
/data		500Gb (xfs)						
/db			500Gb (xfs)		250Gb (xfs)	250Gb (xfs)	250Gb (xfs)	250Gb (xfs)
Объем 3-го виртуального жесткого диска, ГБ Разбиение диска на разделы:		500 (HDD)	500 (HDD)					
/backup		500Gb (xfs)	500Gb (xfs)					

	Web Server (NGINX) & Cache DB (Redis)	Application Server (Accent)	Relational DB (PostgreSQL)	Relational DB (PgBouncer)	Non-Relational DB (MongoDB)	Message Broker (Kafka)	Monitoring & Logging Server (Prometheus & Grafana)	Workflow Manager (Airflow)
Количество виртуальных сетевых интерфейсов, шт.	1	1	1	1	1	1	1	1
внешний белый IP-адрес (да/нет)	да	нет	нет	нет	нет	нет	нет	нет
внутренний IP-адрес (да/нет)	да	да	да	да	да	да	да	да
Планируемое количество пользователей ИС, чел.								
Тип подключения пользователей ИС, размещаемой на виртуальном сервере (например, терминальный, программный клиент(толстый), веб-интерфейс)	http, https	нет	нет	нет	нет	нет	нет	нет
IP-адреса пользователей системы, для доступа	все	нет	нет	нет	нет	нет	нет	нет
Логин администратора виртуального сервера	root	root	root	root	root	root	root	root

	Web Server (NGINX) & Cache DB (Redis)	Application Server (Accent)	Relational DB (PostgreSQL)	Relational DB (PgBouncer)	Non-Relational DB (MongoDB)	Message Broker (Kafka)	Monitoring & Logging Server (Prometheus & Grafana)	Workflow Manager (Airflow)
Тип подключения администраторов ИС, размещаемой на виртуальном сервере (например, терминальный, программный клиент(толстый), веб-интерфейс)	ssh, 22 порт	ssh, 22 порт	ssh, 22 порт	ssh, 22 порт	ssh, 22 порт	ssh, 22 порт	ssh, 22 порт	ssh, 22 порт

2.2 Инфраструктурные назначения серверов

Сервера Системы сбора сведений в чрезвычайных ситуациях и мониторинга восстановительных работ классифицируются по инфраструктурному назначению. При этом, в целях обеспечения бесперебойной работы ИС для каждого ИН необходимо выделяется несколько однотипных серверов со схожими техническими характеристиками. Для каждого инфраструктурного назначения определен набор пакетов, ролей, сервисов и компонентов системы, которые разворачиваются на сервере. Для Системы определены следующие инфраструктурные назначения:

- WEB сервер (web). Поставляет клиентскому ПО (программное обеспечение) программный код, обеспечивающий непосредственное взаимодействие с пользователем. Принимает и авторизует запросы от клиентского ПО.
- APP сервер (app). Обеспечивает дополнительную обработку запросов клиентского ПО к сервисам ИСУП.
- Сервер реляционной БД (база данных) (pgsqlmain, pgsqladd2, pgb01, pgb02). Обеспечивает работу реляционной БД (PostgreSQL)
- Сервер нереляционной БД (mongodb). Обеспечивает работу документ-ориентированной БД (MongoDB).
- Сервер шины данных (Kafka). Брокер сообщений. Обеспечивает централизованный сбор, передачу и обработку сообщений.

2.3 Общее программное обеспечение

Средства общего программного обеспечения серверной части устанавливаются на комплекс технических средств, состав которого представлен в подразделе 2.1 в соответствии с таблицей 4.

Таблица 4 – Состав серверного общего программного обеспечения.

№	Имя сервера	Общее программное обеспечение	Назначение
1	web	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64 (Orel), Astra Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64, Docker, Openresty, Consul, Consul-template, Redis, GlusterFS	WEB сервер
2	app	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64 (Orel), Astra Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64, Docker, GlusterFS	APP сервер
3	pgsqlmain	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64 (Orel), Astra Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64, PostgreSQL 12, Patroni, pg_probackup	Сервер реляционной БД
4	pgb01	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64 (Orel), Astra Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64, PGbouncer, Keepalived	Сервер реляционной БД
5	mongodb	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04	Сервер нереляционной БД

№	Имя сервера	Общее программное обеспечение	Назначение
		LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64 (Orel), Astra Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64, MongoDB	
6	kafka	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64 (Orel), Astra Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64, Kafka, Zookeeper	Сервер Распределенный брокер сообщений
7	geoserver	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64 (Orel), Astra Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64, Django, Geoserver, TomCat	Сервер картографии
8	public	Centos 7 x86_64 minimal, Rocky Linux 8 x86_64 minimal, Ubuntu 20.04 LTS amd64 (Focal Fossa), Astra Linux Common Edition 2.12 x86_64 (Orel), Astra Linux Special Edition 1.7 x86_64, Alt Server 10 x86_64, Docker, Openresty, Consul, Consul-template, Redis, GlusterFS	WEB сервер

Состав клиентского общего программного обеспечения представлен в таблице 5.

Таблица 5 – Состав клиентского общего программного обеспечения.

№	Наименование	Назначение
1	Windows 7, 10 или Linux (Desktop Edition)	Операционная система
2	Mozilla Firefox 30 или новее, Google Chrome 29 или новее, Яндекс браузер 18 или новее, любой браузер на основе Blink 537 или новее	Браузер
3	Один из следующих офисных пакетов: - Microsoft Office; - Libre Office; - Мой офис.	Средства, позволяющие открывать файлы форматов офисных приложений (Word, Excel, PowerPoint, Visio, Project, «Мой Офис», OpenDocument Format), форматов цифровых файлов (изображения, аудио, видео), форматов PDF, TIFF, веб-форматов

2.4 Специальное программное обеспечение

В состав специального программного обеспечения входят следующие компоненты Системы:

- Модуль автоматизации деятельности по строительному надзору Московской области;
- Модуль автоматизации деятельности по инспектированию жилищного фонда Московской области;
- Модуль автоматизации контрольно-надзорной деятельности органов исполнительной власти Московской области.

2.5 Компоненты программного обеспечения

Система сбора сведений в чрезвычайных ситуациях и мониторинга восстановительных работ состоит из следующих компонентов специального программного обеспечения:

- Компонент UserWebClient – содержит и предоставляет клиентскому ПО (браузеру) клиентский программный код на языке JavaScript для взаимодействия с пользователем.
- Компонент SystemService – хранит и предоставляет конфигурационные параметры системы.
- Компонент AccessEditor – проводит аутентификацию пользователя, поставляет сведения о пользователе, роли и правах доступа остальным компонентам.
- Компонент ObjectEditor – обслуживает запросы разработчика на изменение структуры информационной базы системы, выполнение настроек системы через редактор.
- Компонент RegistryService – обслуживает запросы на выборку, добавление и актуализацию данных в системе.
- Компонент LogicEditor – выполняет предусмотренные регламентом процедуры при поступлении новых сведений в соответствии с настроенными бизнес-процессами.
- Компонент MenuEditor – Предоставляет гибкую настраиваемую структуру разделов АИС.
- Компонент InterfaceEditor – предоставляет настройки пользовательских интерфейсов в зависимости от роли пользователя и выбранного раздела.
- Компонент EtlEditor – предоставляет средства импорта/экспорта данных в системе, а также инструменты интеграции.

3 Установка подсистем

Развертывание Системы сбора сведений в чрезвычайных ситуациях и мониторинга восстановительных работ на серверах происходит удаленно по сети при помощи ansible в автоматическом режиме.

3.1 Требования к рабочему месту оператора развертывания

В качестве управляющей машины может использоваться компьютер под управлением различных дистрибутивов Linux или Mac OS. Рабочие места оператора развертывания под управлением операционной системы Windows не поддерживаются.

На рабочем месте оператора развертывания должны быть установлены следующие пакеты ПО и их зависимости:

- python 2.7 (интерпретатор языка программирования Python 2);
- python-pip (утилита управления пакета ПО на python);
- средство для работы с системой контроля версий git;
- средство удаленного администрирования ansible;
- средство обмена библиотеками описания инфраструктуры ролей ansible-galaxy.

3.2 Подготовка рабочего места оператора развертывания

Выполните следующие необходимые действия для подготовки управляющей машины к выполнению процедуры развертывания Системы сбора сведений в чрезвычайных ситуациях и мониторинга восстановительных работ:

- Получите исходный код описания инфраструктуры из системы контроля версий `git clone http://<адрес git>/<группа>/infrastructure2.0`

- Перейдите в каталог репозитория с описанием инфраструктуры `cd infrastructure2.0`

- Установите необходимые библиотеки python для корректной работы системы удаленного автоматизированного развертывания

```
pip install -r requirements.txt
```

- Скачайте общие части описания инфраструктуры из открытого репозитория ansible-galaxy

```
ansible-galaxy install -r requirements.yml
```

3.3 Описание структуры пакета автоматизированного развертывания Системы

Система имеет 3 слоя:

- Базовый слой: устанавливает все необходимые пакеты для корректной сборки приложения. Он должен быть применен ко всем серверам ИСУП ТО независимо от инфраструктурного назначения и сервисного набора.

Базовый слой содержит следующие роли:

- 1) ntp – протокол для синхронизации времени;
- 2) pip – система управления пакетами для Python;
- 3) def-packages – необходимые пакеты для установки библиотек (репозиторий postgresql, gcc, unzip, wget);
- 4) sysctl;
- 5) hosts-file добавляем в файл /etc/hosts рабочие имена других инфраструктурных единиц;
- 6) limits-conf;
- 7) firewalld-ports открывает необходимые порты в штатном firewall операционной системы настраиваемого сервера.

- Сервисный слой: устанавливает роли сервисов, необходимых для нормальной работы Системы ТО. Сервисные роли делятся по инфраструктурному назначению серверов. Ниже перечислены инфраструктурные назначения и список ролей, которые будут развернуты.

На серверах с инфраструктурным назначением backend устанавливаются следующие роли:

- glusterfs – служба для хранения и передачи файлов;
- docker – ПО для управления приложением с помощью контейнеров.

На серверах с инфраструктурным назначением frontend устанавливаются следующие роли:

- docker – ПО для распространения компонентов приложения совместно с зависимостями в форме контейнеров;
- consul – служба для поддержания сервисов;
- consul-template – служба для управления конфигурациями;
- openresty – веб сервер с поддержкой lua;
- redis – резидентная система управления БД;
- glusterfs – служба для хранения и передачи файлов.

На серверах с инфраструктурным назначением pgbouncer устанавливаются следующие роли:

- pgbouncer – программа, управляющая пулом соединений postgres;
- keepalived – ПО для обеспечения высокой доступности (плавающий ip-адрес).

На серверах с инфраструктурным назначением postgres устанавливаются следующие роли:

- postgresql – объектно-реляционная СУБД;
- partroni – отказоустойчивый кластер postgres с поддержкой контролируемого и автоматического аварийного переключения;
- pg_probackup – утилита для управления резервным копированием и восстановлением БД.

На серверах с инфраструктурным назначением mongodb устанавливаются следующие роли:

- mongodb – документоориентированная NoSQL СУБД.

На серверах с инфраструктурным назначением kafka устанавливаются следующие роли:

- kafka – распределённый программный брокер сообщений;
- zookeeper – централизованная служба для поддержки информации о конфигурации, именования, обеспечения распределенной синхронизации и предоставления групповых служб.

На серверах с инфраструктурным назначением monitoring устанавливаются следующие роли:

- Слой поставки deployment устанавливает компоненты системы на сервер в зависимости от его инфраструктурного назначения. Настройка осуществляется путем редактирования файла `site.yml`. Необходимо указать названия компонентов, группу хостов, количество копий. Пример развертывания компонента userwebclient на серверах с ИН frontend приведен на рисунке 1.

```

- import_playbook: app_level_playbooks/deployment.yml
vars:
  sname: userwebclient
  count: "{{ count }}"
  usr: "{{ usr }}"
  pwd: "{{ pwd }}"
  host_group: frontend
  sversion: 2.0.1

```

Рисунок 1 – Пример развертывания компонента userwebclient на серверах с ИН frontend.

3.4 Планирование инфраструктуры

Существует три готовых варианта инфраструктуры для развертывания Системы сбора сведений в чрезвычайных ситуациях и мониторинга восстановительных работ.

- Запуск на выделенном сервере всех компонентов системы и необходимых сервисов с целью тестирования либо изучения `#ansible-playbook site.yml -i /enviroments/test/inventory.yml`

- Запуск на резервных мощностях с развертыванием каждого инфраструктурного назначения на отдельном сервере. Все инфраструктурные назначения будут развернуты в единственном числе без резервирования и кластеризации

```
#ansible-playbook site.yml -i /enviroments/preprod/inventory.yml
```

```

---
all:
  children:
    backend:
      hosts:
        APP1:
          ansible_host: 192.168.1.221
    frontend:
      hosts:
        FE1:
          ansible_host: 192.168.1.225
    postgres:
      hosts:
        PG1:
          ansible_host: 192.168.1.231
    mongodb:
      hosts:
        MG1:
          ansible_host: 192.168.1.233
    monitoring:
      MON1:
        test-mon:
          ansible_host: 192.168.1.251
...

```

– Развертывание Системы ТО на кластере из 11 серверов с резервирование инфраструктурных назначений `#ansible-playbook site.yml -i /enviroments/prod/inventory.yml`

3.5 Развертывание на инфраструктуру

Для выполнения операции развертывания необходимо отредактировать файл `/environments/<infra-var>/inventory.yml`, где `<infra-var>` является выбранным на предыдущем шаге вариантом инфраструктуры ИС. Укажите ip адреса серверов каждого инфраструктурного назначения. Выполните следующую команду:

```
#ansible-playbook site.yml -i /enviroments/<infra-var>/inventory.yml
```

3.6 Обновление системы

Процесс непрерывной поставки подразумевает систематическую доработку компонентов ИС и скорейшую доставку обновлений в рабочую среду. Для обновления компонентов системы отредактируйте файл `site.yml` укажите в поле

`version` более свежую версию или ключевое слово `latest`, а затем выполните команду

```
#ansible-playbook site.yml -i /enviroments/<infra-var>/inventory.yml
```

, здесь `<infra-var>` наименование варианта инфраструктуры, на которой будет выполнено обновление.

Если есть необходимость обновить только некоторые компоненты системы – закройте комментарием части кода в `site.yml`, указывающие на компоненты, обновлять которые нежелательно.

3.7 Горизонтальное масштабирование Системы

Для оптимизации производительности Системы сбора сведений в чрезвычайных ситуациях и мониторинга восстановительных работ в ходе эксплуатации может возникнуть необходимость в горизонтальном масштабировании. Целью горизонтального масштабирования является более эффективное потребление системой вычислительных возможностей серверов, на которых развернуты её компоненты. Для выполнения горизонтального масштабирования в системе автоматизированного развертывания предусмотрены файлы описания настроек для ansible (плейбуки): `increase_replicas.yml` и `decrease_replicas.yml`, расположенные в каталоге `tree/master/playbooks/app_level_playbooks` Они предназначены для увеличения или уменьшения количества реплик сервиса-компонента системы в окружениях. На вход принимают следующие параметры: `count` – количество реплик сервиса, на которое нужно изменить текущее количество («1» по умолчанию), `name` – имя компонента обязательное к указанию.

Плейбук изменяет количество реплик на указанное, берёт текущее состояние (количество реплик и версию сервиса) из окружения. После завершения операции масштабирования плейбук обновляет указанные параметры.

3.8 Откат к предыдущей версии

В случае, когда после обновления система работает некорректно, может возникнуть необходимость в откате системы к предыдущей версии. Операция отката осуществляется запуском плейбука `tree/master/playbooks/app_level_playbooks/rollback.yml`. В качестве обязательного аргумента передается `name` – имя компонента системы, который нужно откатить.

3.9 Тестирование системы развертывания

Все роли протестированы с помощью `molecule`.

Для запуска тестов необходимо выполнить команду:

```
#molecule test -s vagrant
```

4 Настройка подсистем

Дополнительная настройка подсистем Системы сбора сведений в чрезвычайных ситуациях и мониторинга восстановительных работ не требуется. Система конфигурируется автоматически в результате выполнения действий, описанных в пункте 3 данного документа.

5 Проверка работоспособности подсистем

Проверить систему можно с помощью заранее подготовленных и установленных триггеров (слушателей) на сервере мониторинга. С каждого сервиса отправляются собранные данные (checkhealth) о состоянии самого сервиса, взаимодействии его как с другими сервисами, так и с серверами баз данных.

6 Системные сообщения

В ходе исполнения единой команды развертывания ПО могут появляться различные системные сообщения `ansible`. Как правило, данные сообщения носят информационный характер и отражают ход работы по развертыванию. Однако, при неверном указании IP адресов, пролей доступа к хостам, несуществующих ролей и других ошибок конфигурации `inventory.yml` возможно появление сообщений об ошибках. Их подробное описание существует в документации `ansible`. В случае появления сообщений об ошибках связанных с `inventory.yml`, исправьте файл `inventory.yml`, и выполните попытку развертывания повторно.

Система развертывания ПО проходит предварительное тестирование как при помощи `molecule`, так и на тестовых стендах с реальной ОС. Если в ходе развертывания ПО появились ошибки НЕ связанные с `inventory.yml`, они могут быть следствием динамического развития инфраструктурного окружения ИС. Выполните `git checkout <commit>` до предыдущего рабочего `commit`, и повторите попытку развернуть ПО.