**Задание:**

Поднять две ВМ под управлением Linux (желательно RHEL, CentOS, или Fedora).

На ВМ1 c помощью Ansible установить и запустить Jenkins.

В Jenkins создать и запустить Job свободной конфигурации, который выполнит на ВМ2:

3.1. Установку пакетов: htop, tmux, jq.

3.2. Сконфигурирует сервис sshd таким образом, чтобы запретить пользователю root соединения по ssh с удаленных хостов и парольную аутентификацию, включит аутентификацию по SSH ключам.

3.3. Перезапустит сервис sshd.

В качестве результатов необходимо прислать:

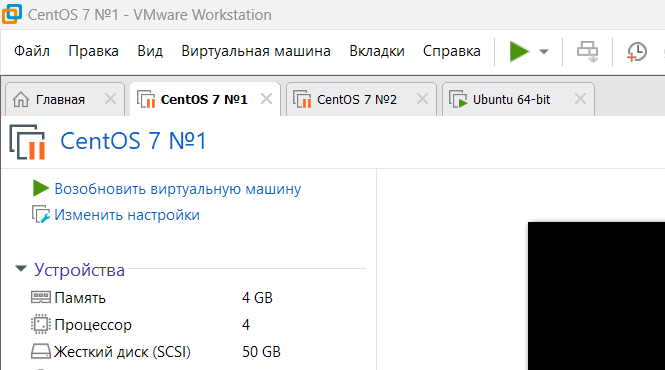
1. Ansible плэйбук и лог его выполнения.

2. Лог и конфигурацию (можно скриншот) успешно отработавшего Jenkins Job.

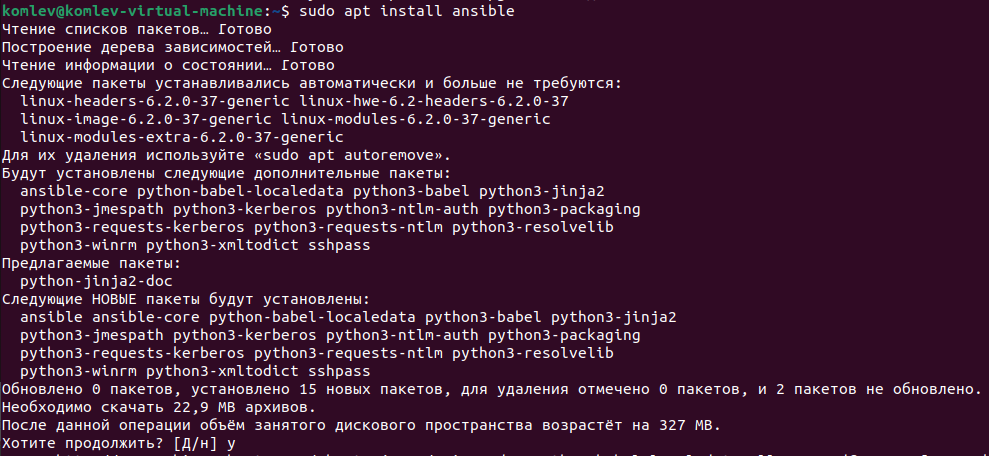
3. Файл конфигурации sshd.

**Реализация:**

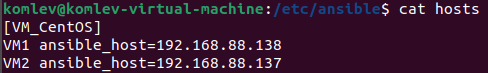
Создаю две ВМ CentOS 7 в VMware Workstation



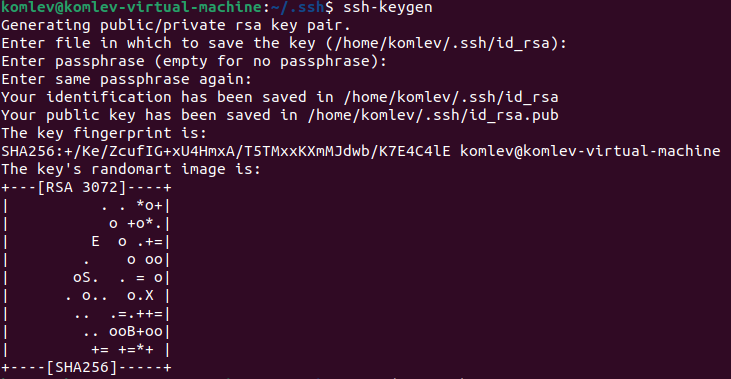
Устанавливаю Ansible на другую виртуалку (Ubuntu, которой пользуюсь в лчиных целях при необходимости).



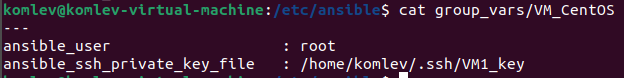
Далее настраиваю файл hosts (в нем прописаны только хосты, без пользователя и ключа, так как эти данные хранятся отдельно в group\_vars)



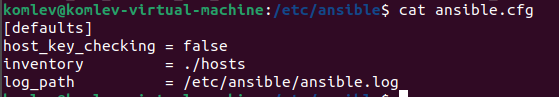
Генерирую пару ssh ключей на мастер-компьютере. Добавляю публичный ключ в authorized\_keys на ВМ1.



Соответственно добавляю group\_vars

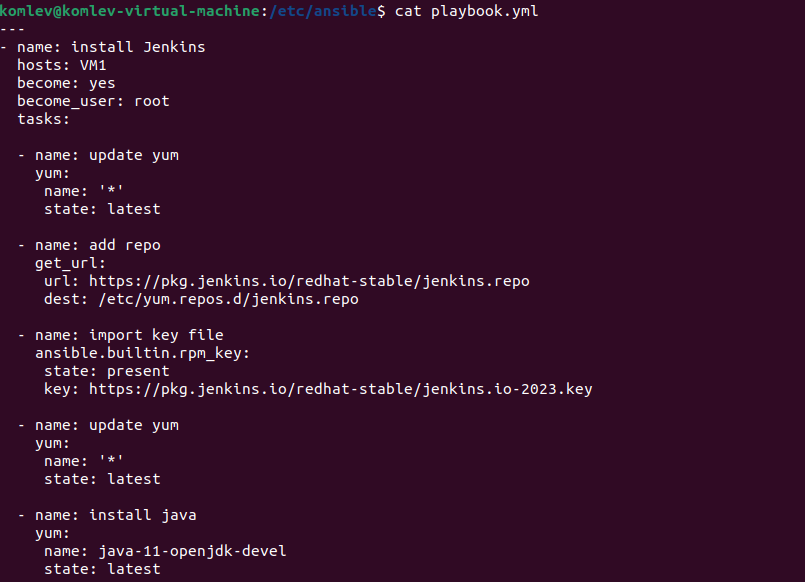


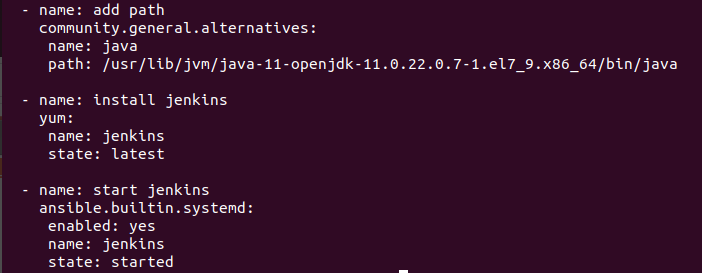
Также корректирую ansible.cfg



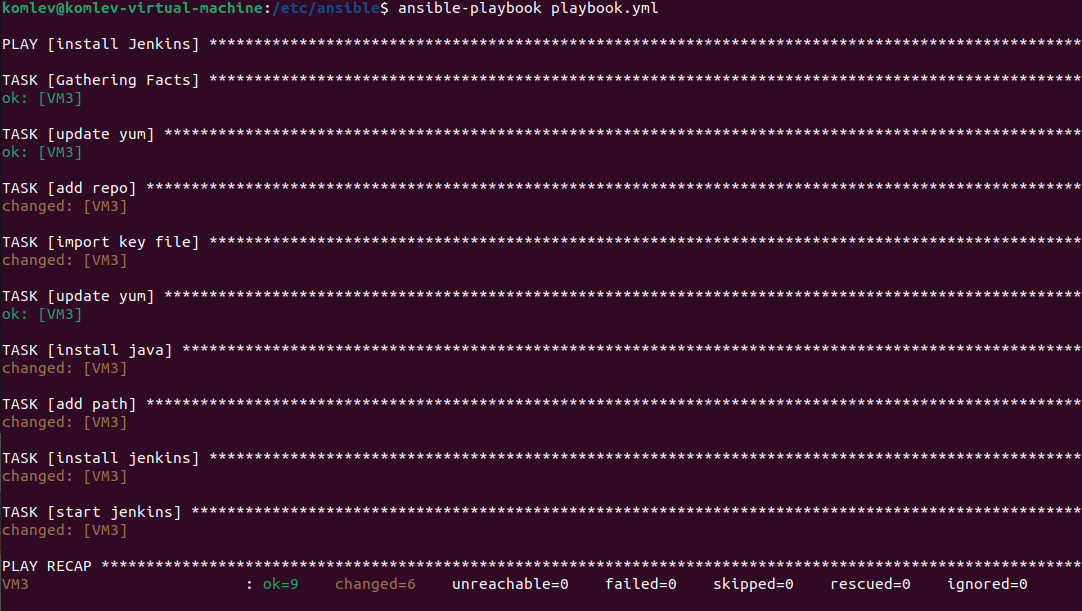
Создаю playbook с такими тасками:

Обновление yum -> добавление репозитория Jenkins -> добавление rpm-ключа для репозитория -> обновление yum -> установка java -> обновление альтернатив (выбор версии Java) -> установка Jenkins -> запуск Jenkins

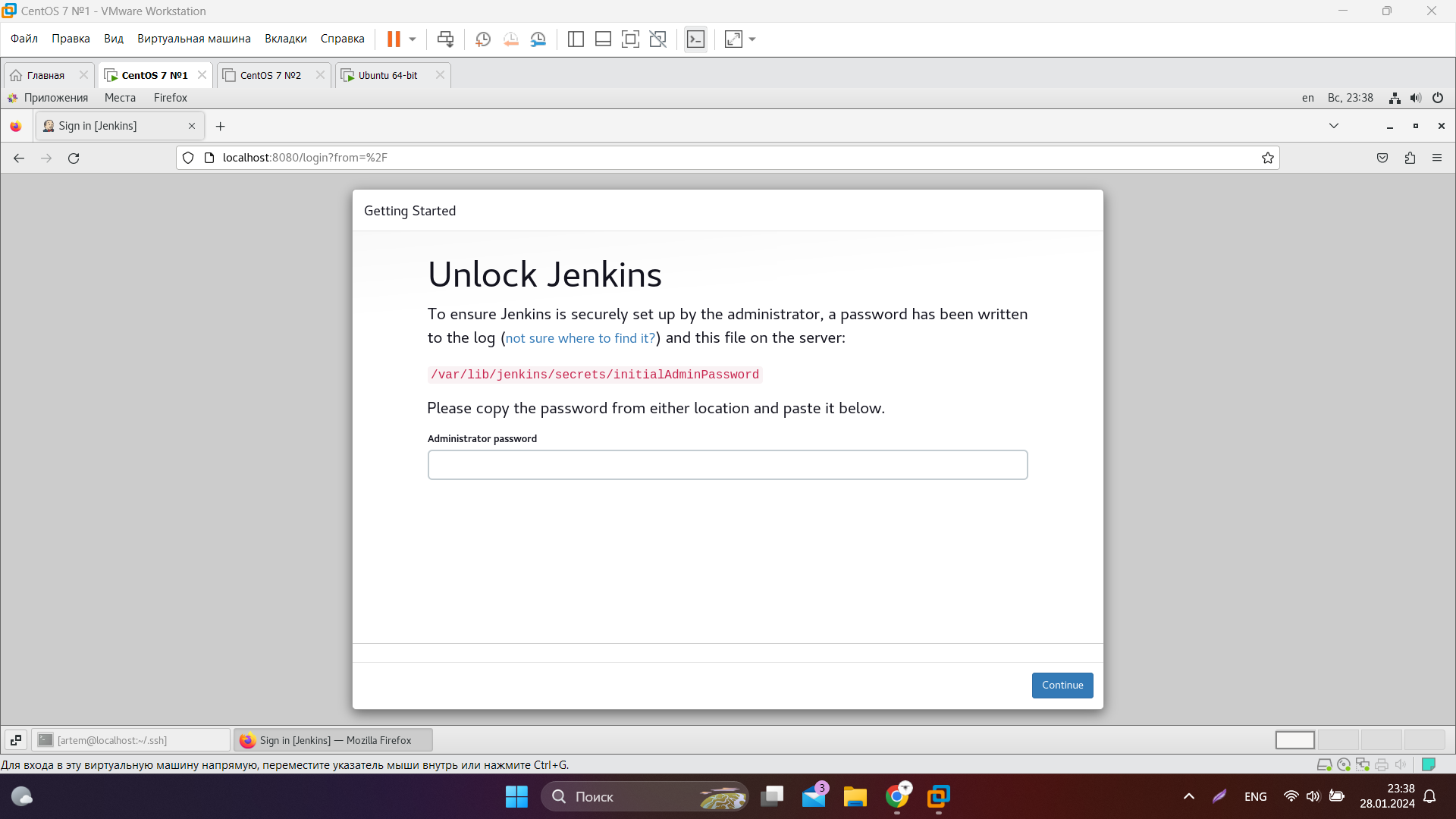




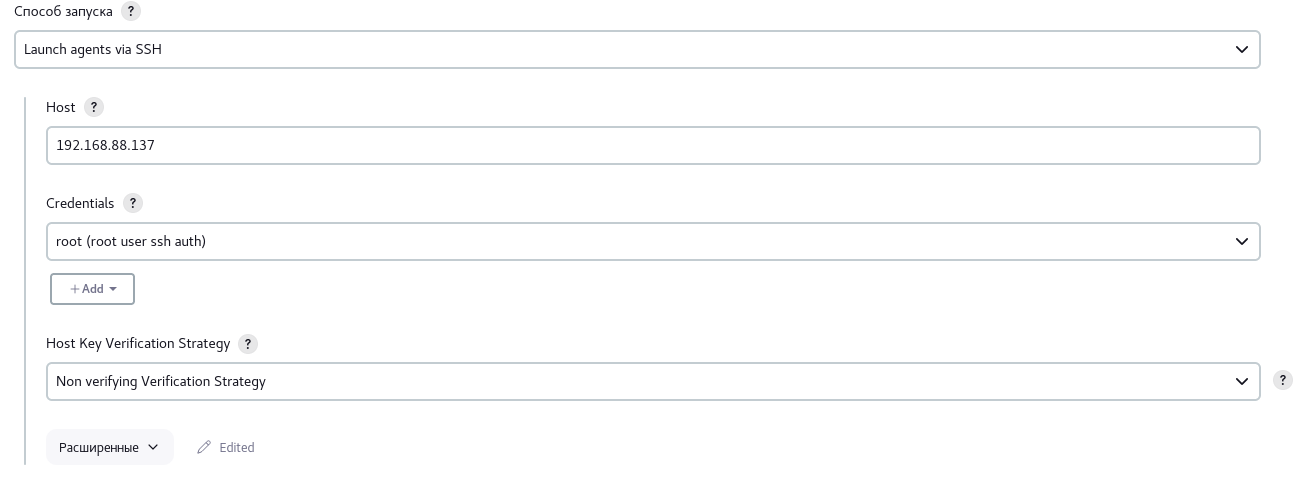
Процесс выполнения:



Теперь перехожу в Jenkins на ВМ1

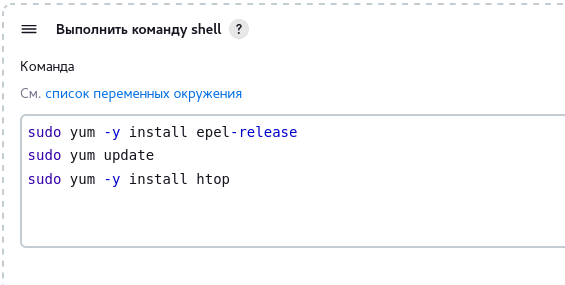


Добавляю агента VM2, чтобы запускать на нем джобу (в качестве способа авторизации выбираю ssh, генерирую и добавляю ключи)

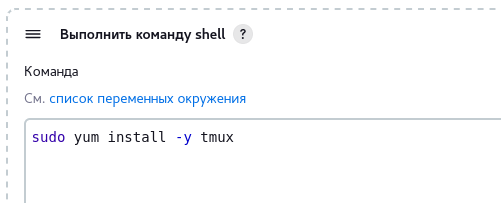


Запускаю агент и перехожу к создаю джобы Jenkins. При настройке выбираю label прописанный у агента. Шаги сборки:

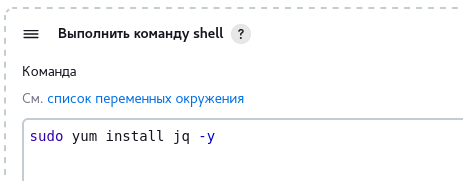
1. Установка epel пакетов, обновление yum, установка htop



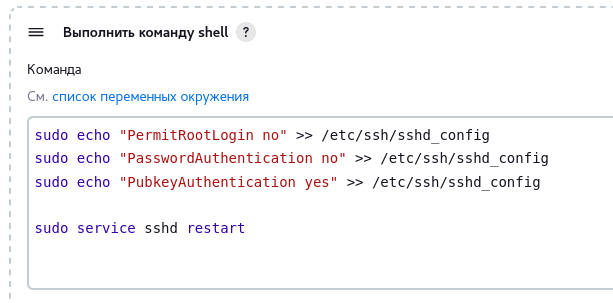
1. Установка tmux



1. Установка jq



1. Настройка sshd\_config, перезапуск сервиса



Результат:



**Задание:**

**Кейс 1**

"Представьте АС, уровня BO, в которой ежедневно работает до 20 тысяч пользователей. Доработку АС ведут разные команды из разных Блоков (более 15), внутри данной АС выделены несколько

ФП. Ближе к концу года службой мониторинга фиксировались регулярные замедления в работе АС. «Краснели» попеременно или вместе показатели времени отклика БД, переполнения памяти, утилизации процессоров, заканчивались свободные процессы.

Некоторые всплески завершались до того, как это замечали пользователи, некоторые длились часами и приводили к необходимости регистрации инцидентов со стороны пользователей.

Опишите, как бы Вы решали данные инциденты и исключали их повторение для данной АС в будущем."

Для решения данных инцидентов и исключения их повторения в будущем, я бы предложил следующие действия:

1. Анализ производительности и мониторинг системы: необходимо провести детальный анализ производительности и мониторинг системы, чтобы выявить причины замедлений в работе АС. Для этого можно использовать различные инструменты мониторинга производительности, такие как Nagios, Zabbix, Grafana и т.д.

2. Оптимизация базы данных: необходимо оптимизировать базу данных, чтобы уменьшить время отклика БД. Для этого можно использовать инструменты оптимизации БД, такие как MySQLTuner, Percona Toolkit и т.д.

3. Оптимизация кода: необходимо оптимизировать код приложения, чтобы уменьшить нагрузку на процессор и память. Для этого можно использовать инструменты оптимизации кода, такие как PHP CodeSniffer, PHPMD, PHP CS Fixer и т.д.

4. Увеличение ресурсов сервера: если причина замедлений в работе АС связана с недостаточными ресурсами сервера, то необходимо увеличить ресурсы сервера (память, процессор и т.д.).

5. Разделение функциональных блоков: если причина замедлений в работе АС связана с конфликтами между функциональными блоками, то необходимо разделить функциональные блоки на отдельные серверы.

6. Создание резервных копий: необходимо создать резервные копии базы данных и приложения, чтобы в случае сбоя можно было быстро восстановить работу системы.

7. Регулярное обновление: необходимо регулярно обновлять приложение и его компоненты, чтобы устранять уязвимости и улучшать производительность.

8. Тестирование изменений: перед внесением каких-либо изменений в систему необходимо проводить тестирование, чтобы убедиться, что изменения не повредят работу системы.

9. Обучение и документация: необходимо обучить пользователей и разработчиков работе с системой, чтобы они могли правильно использовать ее возможности и избегать ситуаций, которые могут привести к замедлению работы. Также необходимо обеспечить наличие документации, которая поможет пользователям и разработчикам быстро находить решения проблем.

10. Планирование и прогнозирование: необходимо проводить регулярное планирование и прогнозирование, чтобы предвидеть возможные проблемы и заранее принимать меры для их предотвращения. Это может включать в себя анализ нагрузки, планирование обновлений и резервирование ресурсов.

**Кейс 2**

"Вашей команде на сопровождение передают функциональную подсистему (часть от другой команды. Контроль процесса передачи этой функциональной подсистемы ложится на Ваши плечи.

Опишите свой чек лист процесса передачи. "

Чек-лист процесса передачи функциональной подсистемы может включать следующие шаги:

1. Определение состава функциональной подсистемы, которая будет передана на сопровождение.

2. Проверка наличия документации и инструкций по работе с функциональной подсистемой.

3. Оценка готовности функциональной подсистемы к передаче на сопровождение (например, проверка наличия всех необходимых компонентов, настройки, тестирование).

4. Подготовка инструкций для команды сопровождения по установке и настройке функциональной подсистемы.

5. Определение ответственных лиц за процесс передачи, анализ рисков и потенциальных проблем, связанных с передачей подсистемы.

6. Планирование времени и ресурсов для передачи функциональной подсистемы (например, выделение времени для обучения команды сопровождения).

7. Проверка доступа команды сопровождения к необходимым ресурсам (например, доступ к серверам, базам данных и т.д.).

8. Передача функциональной подсистемы команде сопровождения и проведение обучения по работе с ней.

9. Проверка работоспособности функциональной подсистемы после передачи и решение возникающих проблем.

10. Мониторинг работы функциональной подсистемы в течение некоторого времени после передачи.

**Кейс 3**

"В период технологического окна (с 00:00 до 01:00) произведено внедрение релиза АС. По итогу проведения работ никаких отклонений в работе АС не зафиксировано. С 04:00 до 06:00 поступает 3 пользовательских инцидента с жалобами на замедления в работе АС. В связи со схожим описанием инцидентов (подозрение на ВПИ) инженер дежурной смены осуществляет сбор ТКС и подключает администратора, который по результатам первичного анализа фиксирует блокировки на базе данных.

Опишите последовательность шагов по:

- классификации и уточнению периметра инцидента

- диагностике инцидента (выявлению причин инцидента и поиск способов его устранения)

- разработке мероприятий по устранению инцидента

- закрытию инцидента в системе автоматизации SM (с заполнением всех необходимых атрибутов инцидента)"

Классификация и уточнение периметра инцидента:

1. Инженер дежурной смены получает информацию о трех пользовательских инцидентах с жалобами на замедления в работе АС.

2. Инженер дежурной смены связывается с пользователями, чтобы уточнить детали инцидентов, такие как время возникновения, действия, предшествующие инциденту, и т.д.

3. Инженер дежурной смены анализирует полученную информацию и классифицирует инциденты по типу и приоритету.

4. Инженер дежурной смены уточняет периметр инцидента, определяя область, где происходят замедления в работе АС.

Диагностика инцидента:

1. Инженер дежурной смены собирает ТКС и подключает администратора для проведения первичного анализа.

2. Администратор проводит анализ системных журналов и баз данных, чтобы выявить причины замедлений в работе АС.

3. Администратор обнаруживает блокировки на базе данных и сообщает об этом инженеру дежурной смены.

Разработка мероприятий по устранению инцидента:

1. Инженер дежурной смены и администратор совместно разрабатывают план мероприятий по устранению блокировок на базе данных.

2. План мероприятий включает в себя проверку и исправление ошибок в коде, оптимизацию запросов, очистку таблиц от лишних данных и т.д.

3. Инженер дежурной смены и администратор проводят тестирование внесенных изменений, чтобы убедиться в их эффективности.

Закрытие инцидента в системе автоматизации SM:

1. Инженер дежурной смены закрывает инцидент в системе автоматизации SM, заполняя все необходимые атрибуты инцидента.

2. Инженер дежурной смены отправляет отчет о произошедшем инциденте руководству и команде разработки для предотвращения подобных инцидентов в будущем.