**1.1 Обоснование необходимости автоматизации процесса управления конфигурацией**

В связи с тенденцией перехода на свободное программное обеспечение

в деятельность органов государственной охраны активно внедряются различные программные продукты с открытым исходным кодом. Одним из таких продуктов является операционная система Astra Linux Special Edition. Она предназначена для создания на ее основе автоматизированных систем в защищенном исполнении, обрабатывающих информацию, составляющую государственную тайну. На ее основе возможно, как создание отдельных объектов ЭВТ, так и развертывание распределенных локальных вычислительных систем.

Для обеспечения эффективного и безопасного функционирования ИС на базе Astra Linux требуется обеспечить качественную настройку и конфигурирование программного обеспечения и самой операционной системы. Данные задачи решаются в процессе системного администрирования.

Системное администрирование - это процесс управления, технического обслуживания и проведения других технических и административных мероприятий, направленных на поддержание информационной системы в рабочем состоянии. В процессе администрирования компьютерных сетей часто возникают задачи выполнения большого количества однотипных операций на некотором числе машин, таких, как:

* установка, обновление, удаление программного обеспечения;
* управление пользователями;
* управление доступом к логическим томам, сетевым хранилищам;
* управление доступом к съемным носителям;
* настройка сетевых параметров;
* разграничение доступа пользователей;

Рассмотрим задачи, решаемые системным администратором на примере использования компьютерного класса Академии ФСО РФ в учебном процессе. В данном классе находиться 24 компьютера, используемые для обучения. Класс является общим, то есть, в нем проводятся занятия по разным дисциплинам преподавателями разных кафедр. Обычно, в учебном классе проводятся 3 пары занятий. Для каждого из занятий может потребоваться различная конфигурация группы или всех компьютеров в классе. Рассмотрим для примера расписание занятий на один день и необходимое состояние компьютеров для каждого занятия.

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п. | Название дисциплины | Число задействованных компьютеров | Требуемая конфигурация машин |
| 1 | Языки программирования | 20 | Наличие доступа к сетевому обучающему ресурсу, к среде программирования. |
| 2 | Вычислительные системы и сети | 24 | Наличие сетевого доступа, возможность изменять сетевые параметры |
| 3 | Системы баз данных | 24 | Наличие доступа к сетевому диску, включенный сервер СУБД на каждой машине, доступ к ПО администрирования СУБД. |

Конфигурировать компьютеры для занятий можно разными способами. Первый – это изменение параметров управляемых узлов без средств автоматизации. Данный способ можно применять при небольшом числе компьютеров в сети, но он имеет следующие существенные недостатки:

1. Для применения какого-либо набора требуемых изменений на большом количестве машин требуется большое количество времени, что в некоторых случаях является критичным;
2. Невозможно осуществить оперативный контроль применения набора конфигураций при большом числе машин;
3. Возможность совершения ошибок администратором.
4. Необходимо учитывать, насколько сильно вычислительная сеть территориально распределена. Разнесение узлов на большие расстояния значительно уменьшает оперативность выполнения необходимых задач.

Для решения данной проблемы администрирование сетевых узлов осуществляют удаленно при помощи специального программного обеспечения. Основными применяемыми инструментами являются программы получения доступа к удаленной командной строке и удаленному рабочему столу.

Несмотря на устранение данного недостатка, метод удаленного администрирования так же не дает возможность быстрого применения необходимых конфигураций на большом количестве машин; не позволяет осуществить оперативный контроль применения набора конфигураций и не защищает от возможных ошибок администратора. Для устранения первых двух проблем необходимо автоматизировать процесс применения конфигурации и контроля результата изменения состояния системы, путем применения планировщиков, подготовки специальных скриптов администрирования, которые бы осуществляли групповое применения необходимых изменений; или применение готовых программных продуктов, автоматизирующие эти задачи.

**1.2 Применение процесса управления конфигурацией в администрировании информационных систем**

Для уменьшения числа ошибок необходимо четко регламентировать операции процесса администрирования узлов:

1. Необходимо заранее определить исходное состояние машин; установить, какие действия необходимо произвести, для приведения состояния машины к требуемому;
2. Сформировать шаблоны, скрипты для конфигурирования;
3. Произвести отладку, исправление неточностей ошибок в скриптах;
4. По возможности, задокументировать набор применяемых изменений в системе;
5. Осуществлять контроль за результатом применения конфигурации.

Описанные операции для устранения недостатков представляют собой процессы, которые включают в себя управление конфигурациями.

Управление конфигурациями - в системной инженерии, это процесс установления и поддержания постоянства качества продукта, функциональных и физических свойств в соответствии с их требованиями, архитектурных и эксплуатационных сведений на всех жизненном цикле.

Управление конфигурациями это практика систематического управления конфигурациями так, что система сохраняет целостность на протяжении времени. Управление конфигурациями внедряет политики, процедуры и инструменты, которые необходимы для управления, оценки предложенных изменений, отслеживания состояния изменения и составление описи системы, и поддержка документирования системных изменений.

Управление конфигурациями составляет программу и план обеспечения технического и административного руководства разработкой и внедрением процедур, функций, служб, инструментов, процессов и ресурсов, необходимых для успешной разработки и поддержки сложных систем. В течение разработки системы, управление конфигурацией позволяет управлять программой для отслеживания требований на протяжении всего жизненного цикла путем принятия и эксплуатации и технического обслуживания. Если изменения неизбежны из требований и дизайна, они должны быть утверждены и документированы, создавая точный отчет о состоянии системы. В идеале процесс управления конфигурациями применяется на протяжении всего жизненного цикла системы.

Процесс управления конфигурациями описан в большом количестве стандартов. В Российской Федерации данный процесс определен в стандарте ГОСТ Р ИСО 10007-2007. В этом стандарте описаны пять процессов, которые выполняются в качестве политик и процедур для определения базовых конфигураций, и выполнения стандартного процесса управления изменениями.

В соответствии со стандартом, управление конфигурациями состоит из следующих процессов:

* Планирование управления конфигурацией. В ходе данного процесса определяются обязанности персонала, ответственность сотрудников; ресурсы, конфигурацией которых необходимо управлять; базовые конфигурации для ресурсов; применяемые процедуры и инструменты; порядок контроля статуса конфигураций;
* Идентификация конфигураций. Данный процесс состоит из установления и поддержания базовых конфигураций, которые определяют систему или подсистему, архитектуру, компоненты, а также события, происходящие в любой момент времени. Это является основой того, что изменение любой части системы определено, документировано и в дальнейшем отслеживается путем с помощью проектирования, разработки, тестирования и окончательного применения;
* Управление изменениями. Включает в себя оценку всех запросов и предложения на изменения и их последующего одобрения или отказа. Это процесс контроля изменений в архитектуре аппаратных, программно-аппаратных средств, программного обеспечения и документации системы.
* Учет статуса конфигураций. Включает процесс записи и представления описания элементов конфигурации (например, аппаратные средства, программное обеспечение, прошивки и т.д.) и все отклонения от базовой конфигурации в процессе проектирования и производства. В случае предполагаемых проблем, благодаря проверке базовой конфигурации и утвержденных модификаций, их можно быстро определить.
* Аудит конфигурации. Аудит конфигурации подтверждает то, что документация на конфигурации системы и подсистемы согласована с их функциональными и физическими характеристиками производительности до принятия в архитектуру базовой конфигурации.

Управление конфигурацией используется для поддержания понимания состояния сложных активов с целью поддержания высокого уровня удобства эксплуатации при самой низкой стоимости. В частности, она направлена на обеспечение непрерывности деятельность системы при превышении предела планируемого срока службы или снижения уровня качества этих активов (или части активов).

Применительно к операционным системам, в процессе управления конфигурациями обеспечивает:

* идентификацию начального состояния системы;
* контроль применения конфигураций (ограничения числа лиц, допущенных к управлению или числа программ и служб, которые доступны для конфигурирования);
* учет статуса конфигурации;
* отслеживание конфигураций и аудит примененных конфигураций. Данный процесс обеспечивает специализированное ПО, позволяющее автоматизировать этапы управления конфигурацией.

Система управления конфигурацией - это программный комплекс, обеспечивающий автоматизацию планирования, изменения, контроля и учета состояния конфигураций рабочих станций и серверов. Часто, в зависимости от масштабов системы УК, она может не включать некоторые функции процесса УК, перенося ответственность их выполнения на администратора.

**1.3 Управление конфигурацией и информационная безопасность**

В контексте информационной безопасности процесс Управления конфигурациями имеет наибольшее значение, так как он позволяет классифицировать конфигурационные единицы, то есть определить информационные активы и установить их критичность. Эта классификация определяет связи между конфигурационными единицами и предпринимаемыми мерами или процедурами безопасности.

Классификация конфигурационных единиц определяет их конфиденциальность, целостность и доступность. Каждая конфигурационная единица в конфигурационной базе данных должна быть классифицирована. Эта классификация связывает конфигурационную единицу с соответствующим комплексом мер безопасности или процедурой.

Виды работ, выполняемых в рамках процесса управления изменениями, часто бывают тесно связаны с безопасностью, так как управление изменениями и управление информационной безопасностью взаимозависимы. Если достигнут приемлемый уровень безопасности, который находится под контролем процесса управления изменениями, то можно гарантировать, что этот уровень безопасности будет обеспечиваться и после проведения изменении. Для поддержки этого уровня безопасности существует ряд стандартных операций. Каждый запрос на изменения связан с рядом параметров, которые определяют процедуру приемки. Параметры срочности и степени воздействия могут быть дополнены параметром, связанным с безопасностью. Если Запрос на изменения может оказать значительное воздействие на информационную безопасность, потребуются расширенные приемочные испытания и процедуры.

Любые меры безопасности, связанные с внесением изменений, должны реализовываться одновременно с проведением самих изменений, и они должны тестироваться совместно. Тесты безопасности отличаются от обычных функциональных тестов. Задачей обычных тестов является определение доступности определенных функций. При тестировании безопасности проверяют не только доступность функций безопасности, но также отсутствие других, нежелательных функций, которые могут снизить безопасность системы.

С точки зрения безопасности управление изменениями является одним из наиболее важных процессов. Это объясняется тем, что управление изменениями вводит новые меры безопасности в ИТ-инфраструктуру вместе с изменениями этой инфраструктуры.

Также, при внедрении управления конфигурациями, упрощается процесс приведения конфигурации операционных систем в соответствии с требованиями, предъявляемые некоторым профилем безопасности. В процессе управления конфигурацией будет определена базовая конфигурация для данного профиля, определены необходимые изменения, для приведения операционной системы в соответствие данному профилю, произведено применение этих изменений, и осуществление дальнейшего аудита и контроля соответствия конфигурации профилю безопасности.

Исходя из описанных выше положений внедрение процесса управления, конфигурацией позволяет значительно снизить количество ошибок и улучшить контроль статуса конфигурации при решении большого количества одинаковых задач. Дополнительно, внедрение управления конфигурацией решает часть важных задач обеспечения информационной безопасности. Но остается проблема необходимости большого количества времени для применения требуемых изменений в распределенной сети. Данную проблему решает применение автоматизированных систем управления конфигурацией.

**1.4 Системы управления конфигурацией**

Существует большое число различных систем управления конфигурациями. Типовая схема организации компонентов системы УК состоит из хранилища спецификаций, агента трансляции и управляемых узлов (Рисунок 1).



Рисунок 1. Типовая схема организации компонентов системы управления конфигурацией.

В хранилище находятся входные данные – спецификации, которые описывают инфраструктуру или сервер. Спецификация переводится в профиль настроек на Агенте трансляции. Эти данные передаются (или забираются) на управляемые узлы и переводятся в набор исполняемых команд при помощи компонента внедрения. Системы УК позволяют автоматизировать труд системных администраторов. Управляемые узлы при этом настраиваются быстро и безошибочно при условии составления исходного сценария без ошибок. Кроме того, система управления конфигурацией позволяет следить за изменениями состояния машины и поддерживать на ней стабильную эталонную конфигурацию. Решение подобной задачи необходимо, как правило, на довольно крупных системах, так как позволяет экономично эксплуатировать большой парк однотипных машин. Но эти системы также находят применение и в небольших вычислительных сетях.

Управление конфигурацией через сценарии, хранимые в базе данных, существенно повышает управляемость процессом настройки и администрирования IT-инфраструктуры. Парадигма управления информационными системами «Инфраструктура как код» получила аббревиатуру - «IaaS» - Infrastructure as a Code.

* 1. **Требования к системе управления конфигурацией**

В настоящий момент в составе Astra Linux не имеется системы управления конфигурациями. Процесс управления конфигурацией в вычислительных сетях, состоящих машин под управлением данной ОС, осуществляется вручную локально или удаленно.

Исходя из преимуществ внедрения систем управления конфигурацией, можно выделить следующие цели создания системы УК для ОС Astra Linux:

* Повышение оперативности процесса управления конфигурацией серверов и рабочих станций;

На создаваемую систему накладываются следующие ограничения:

1. На рабочих станциях и серверах применяется Astra Linux;
2. Управление конфигурацией должно осуществляться удаленно;
3. Для осуществления требуемых изменений необходимо обладать административными полномочиями.

Учитывая накладываемые ограничения, сформулируем требования к разрабатываемой системе. Система управления конфигурацией операционной системы Astra Linux SE должна отвечать следующим требованиям:

* Возможность удаленного управления целевыми машинами. Это позволяет осуществлять управление большим количеством машин с ограниченного числа узлов в сети.
* Возможность группового применения конфигураций. Требование подразумевает возможность создания шаблона некоторой конфигурации, которую можно применять к группе однотипных узлов с целью приведения их к единому требуемому состоянию.
* Осуществление контроля над процессом применения конфигурации, то есть, возможность отслеживать статус выполнения процесса применения конфигурации на одном или группе управляемых узлов.
* Наличие планировщика для изменения конфигурации в соответствии с некоторым планом, расписанием. Позволяет проводить операции по определенному плану, автоматизировать процесс применения конфигурации.
* Возможность решения большого числа задач администрирования за короткий промежуток времени.
* Широкий спектр решаемых административных задач
* Применение интерпретируемого языка программирования. Выполнение данного требования необходимо для облегчения процесса внедрения программного продукта.
* Наличие графического интерфейса администрирования.

В настоящее время разработано большое число программных продуктов для автоматизации процесса управления конфигурацией. Проведем сравнение существующих программ для определения соответствия их требованиям к системе. Для сравнения были выбраны следующие инструменты управления конфигурациями:

* CFEngine3,
* Puppet,
* Chef,
* SaltStack,
* Ansible.

*Таблица 1.*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название | Возможность удаленного управления | Возможность группового применения конфигураций | Осуществление контроля над процессом применения конфигурации | Наличие планировщика | Лицензия | Написана на | Архитектура | Язык сценариев | Наличие GUI |
| Ansible | Да | Да | Да | Нет | GPL | Python | Без агента | YAML | Да |
| Сhef | Да | Да | Да | Да | Apache | Ruby, Erlang | Клиент-Серверная | Ruby | Нет |
| CFEngine | Да | Да | Да | Да | Проприетарная | С | Клиент-Серверная | Собственный | Да |
| Puppet | Да | Да | Да | Нет | GPL, Apache | Ruby | Клиент-Серверная | Собственный/Ruby | Нет |
| SaltStack | Да | Да | Да | Нет | Apache | Python | Клиент-Серверная | YAML/Python | Да |

В ходе сравнения было выявлено, что большинство существующих программных продуктов отвечают предъявленным требованиям. Были выбраны дополнительные параметры для сравнения:

* лицензия распространения;
* язык программирования, на котором написано данное ПО; архитектура приложения;
* архитектура приложения;
* язык описания сценариев настройки;
* наличие графического интерфейса.

В ходе сравнения было определено, что все системы, кроме CFEngine3, распространяются по лицензии свободного программного обеспечения. Наличие свободной лицензии позволяет использовать исходный код программы для разработки собственных программных продуктов для решения специфических задач. Так же, это позволяет провести аудит безопасности исходного кода с целью выявления и устранения различных уязвимостей.

Приоритетным является использование интерпретируемых языков программирования, так как это облегчает процесс развертывания и внедрения данных систем. Для использования в Astra Linux SE использование программ, написанных на компилируемых языках программирования возможно только после проведения тематических исследований. При применении интерпретируемых языков данный процесс проходит быстрее.

Следует учитывать архитектуру построения этих программ. Архитектура без клиента предпочтительнее клиент-серверной архитектуру, так как ее применение уменьшает количество требуемого дополнительного программного обеспечения на управляемых узлах.

Во всех программных продуктах отсутствует встроенный планировщик задач. Большинство из них ориентировано на применение планировщика операционной системы.

Наличие графического интерфейса является хорошим преимуществом, так как облегчает процесс взаимодействия администратора с системой. Многие, из систем управления конфигурацией имеют пользовательский интерфейс администрирования, но при использовании открытой версии программы его функциональность сильно ограничена.

Значимой является возможности по масштабированию данного инструмента. Необходимо быть уверенным в том, что система будет работать стабильно при большой нагрузке.

Помимо параметров, описанных в таблице, можно выделить еще простоту освоения и использования; возможность предварительного тестирования сценариев; документированность системы; частота обновлений.

В ходе сравнения систем управления конфигурациями по приведенным параметрам выделяется система Ansible. Данный программный продукт написан на языке Python. Он распространяется по лицензии GPL, что означает:

* свободу запуска программы с любой целью;
* свободу изучения того, как программа работает, и её модификации (предварительным условием для этого является доступ к исходному коду);
* свободу распространения копий как исходного, так и исполняемого кода;
* свободу улучшения программы, и выпуска улучшений в публичный доступ (предварительным условием для этого является доступ к исходному коду).

Ansible создан по архитектуре без клиента, для управления узлом необходимо наличие на узле интерпретатора python версии не менее 2.4 и возможности получить доступ по протоколу SSH. Сценарии конфигурирования – плейбуки – формируются декларативно на языке YAML. Применение декларативного подхода облегчает описание требуемого состояния управляемой машины, за счет указания конечного требуемого результата, а не описания самого процесса конфигурирования, как при императивном подходе. Ansible поддерживает написание дополнительных модулей для решения каких-либо специфичных задач.

Встроенный планировщик отсутствует, возможна работа с планировщиком операционной системы.

Данная система очень хорошо документирована, проста в использовании, активно поддерживается разработчиком и сообществом.

Исходя из данных преимуществ, Ansible станет основой для построения системы управления конфигурацией для Astra Linux SE.

**1.7 Постановка инженерной задачи**

Требования к разрабатываемой системе определяются критериями оперативности, результативности и ресурсоемкости.

Сформулируем критерий оперативности.

Пусть – множество требуемых наборов конфигураций, – некоторый набор конфигураций, входящий в множество требуемых конфигураций.  *-* функция применения набора конфигураций на множестве машин.

Критерий оптимальности определяется тем, что время применения набора конфигурации не превышает некоторое допустимое время, т. е.:

Так как процесс применения конфигурации на целевых машинах является случайным процессом, то критерий оперативности преобразуется к следующему виду:

Вероятность того, что время применения набора конфигураций не превысит допустимое время, не меньше некоторого значения.

При рассмотрении применения данной системы в учебном процессе Академии допустимое время определяется длительностью перерыва между учебными занятиями, в течение которого возможно изменение конфигурации. Вероятность = 0.9.

Критерий результативности определяется как отношения числа успешно примененных наборов конфигураций к общему числу наборов.