Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Д. М. Романенко, И. А. Миронов

АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ И ВЕБ-ПОРТАЛОВ

**Лабораторный практикум
для студентов специальности 1-40 05 01-03
«Информационные системы и технологии
(издательско-полиграфический комплекс)»**

Минск 2018

УДК 004.7(076.5)
ББК 32.97я73
Р69

Рассмотрен и рекомендован к изданию редакционно-издательским советом Белорусского государственного технологического университета.

Р е ц е н з е н т ы :
кандидат технических наук, доцент,
заведующая кафедрой программного обеспечения
информационных технологий Белорусского государственного
университета информатики и радиоэлектроники
Н. В. Лапицкая;
кандидат технических наук, доцент,
заведующий кафедрой информационных систем и технологий
Белорусского государственного технологического университета
В. В. Смелов

Романенко, Д. М.

Р69 Администрирование информационных систем и веб-порталов : лаб. практикум для студентов специальности 1-40 05 01-03 «Информационные системы и технологии (издательско-полиграфический комплекс)» / Д. М. Романенко, И. А. Миронов. – Минск : БГТУ, 2018. – 190 с.

В лабораторном практикуме изложены теоретические основы изучаемой предметной области, связанной с построением и администрированием информационных систем на основе операционных систем Windows Server. Описаны практические примеры работы с сетевой (статической и динамической), а также символьной (DNS, NetBios) адресацией, приведены методы планирования и управления Active Directory, удаленного администрирования, построения надежных и безопасных информационных систем. Рассмотрены вопросы организации и администрирования наиболее популярных веб-серверов.

УДК 004.7(076.5)
ББК 32.97я73

ПРЕДИСЛОВИЕ

Дисциплина «Администрирование информационных систем и веб-порталов» представляет собой продолжение изучения сетевой тематики и дает теоретические и практические знания по организации и управлению распределенными информационными системами на основе операционной системы (ОС) Windows Server.

В данном лабораторном практикуме представлены основные технологии, применяемые для организации и администрирования компьютерной сети с помощью специализированного сетевого оборудования.

Рассмотрены методы настройки различных видов адресации (сетевой, символьной), построения и администрирования распределенных информационных систем, обеспечения надежности и безопасности их функционирования. При этом предполагается, что читатель знаком с основами организации и использования компьютерных сетей.

Основная задача лабораторного практикума – дать студентам общие систематизированные знания о методах организации и администрирования информационных систем.

В результате изучения дисциплины и выполнения заданий на лабораторных занятиях студент должен освоить:

– правила и методы настройки статической и динамической адресации в информационных системах;

– способы разделения ресурсов в информационных системах;

– правила и методы настройки символьной DNS-адресации в информационных системах;

– способы организации и удаленного администрирования доменной системы на базе Active Directory;

– методы обеспечения надежности доменной системы на базе Active Directory (репликация);

– способы обеспечения безопасности доменной системы на базе Active Directory (шифрование сетевого трафика).

Студент должен научиться применять рассматриваемые методы администрирования на практике.

Раздел 1

ТЕХНОЛОГИИ, ПРИМЕНЯЕМЫЕ ПРИ ПОСТРОЕНИИ СЕТЕЙ НА ОСНОВЕ КОММУТАТОРОВ

По мере развития сетевых технологий современные коммутаторы становятся все более сложными устройствами. Для успешного построения и обслуживания сетей ключевым моментом является понимание фундаментальных основ наиболее распространенных сетевых технологий, таких как коммутация второго уровня, третьего уровня, IEEE 802.1Q, IEEE 802.1p, RSTP, MSTP, IGMP и многих других, а также знание того, как данные технологии можно применить на практике наиболее эффективно.

1.1. Средства управления коммутаторами

Большинство современных коммутаторов поддерживают различные функции управления и мониторинга. К ним относятся дружественный пользователю веб-интерфейс управления, интерфейс командной строки (Command Line Interface, CLI), Telnet, SNMP-управление.

Для выполнения входа в веб-интерфейс компьютеру должен быть назначен IP-адрес из того же диапазона, в котором находится IP-адрес коммутатора. Например, если коммутатору назначен IP-адрес 10.90.90.90 с маской подсети 255.0.0.0, то компьютеру должен быть назначен IP-адрес вида 10.x.y.z (где x/y – число от 0 до 255; z – число от 1 до 254) с маской подсети 255.0.0.0.

Примечание. IP-адрес коммутатора по умолчанию – 10.90.90.90, маска подсети – 255.0.0.0, шлюз по умолчанию – 0.0.0.0.

Открываем веб-браузер (рис. 1.1) и вводим в адресной строке http://10.90.90.90.

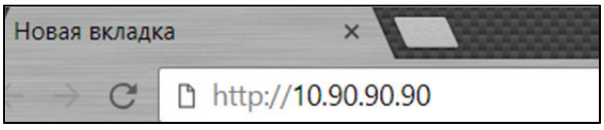


Рис. 1.1. Подключение
к веб-интерфейсу коммутатора

После появления окна аутентификации набираем *admin* в поле пароля. Нажимаем кнопку *OK*, чтобы перейти к главному окну настройки (рис. 1.2).

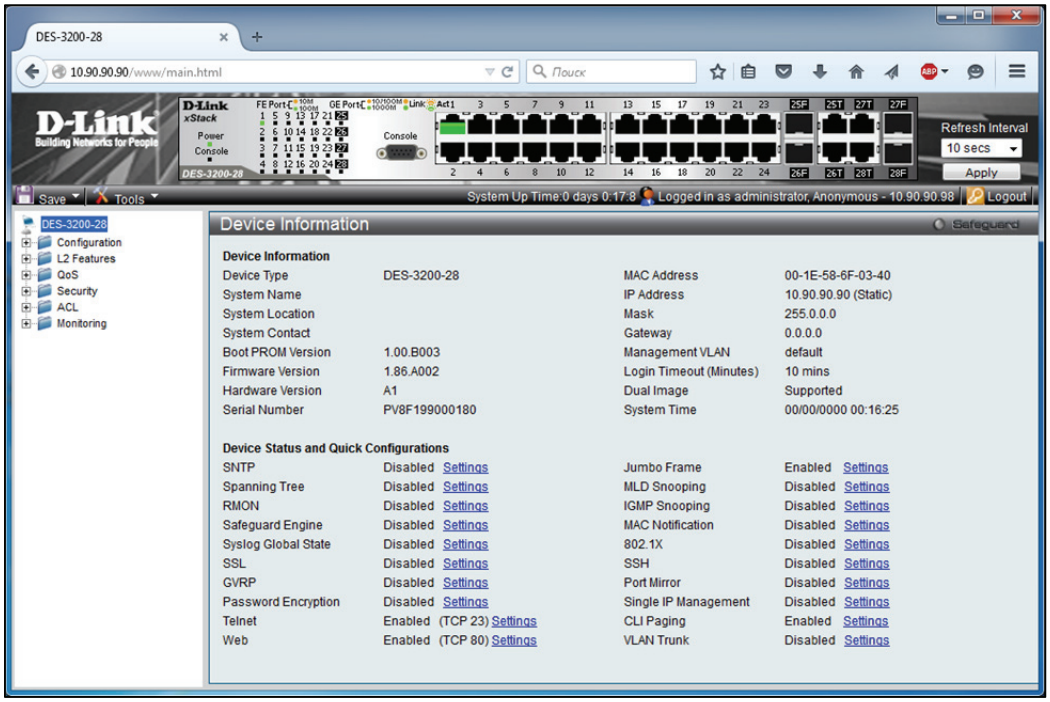


Рис. 1.2. Начальная настройка коммутатора. Пример веб-интерфейса

Прежде чем перейти в меню *Web-based Management* (*Управление на основе Веб-интерфейса*), с помощью *Мастера установки* (*Smart Wizard*) выполняем быструю настройку нескольких функций, таких как *Password Settings* (*Настройки пароля*), *SNMP Settings* (*Настройки SNMP*) и *System Settings* (*Настройки системы*). Если изменять эти настройки не требуется, нажимаем *Exit*, чтобы выйти из *Мастера установки* и перейти в меню *Web-based Management*.

1.2. Применение технологии виртуальных сетей (VLAN)

Поскольку коммутатор Ethernet является устройством канального уровня, то в соответствии с логикой работы он будет рассылать широковещательные кадры через все порты. Хотя трафик с конкретными адресами (соединения «точка – точка») изолирован парой портов, широковещательные кадры передаются во всю сеть (на каждый порт).

Широковещательные кадры – это кадры, передаваемые на все узлы сети. Они необходимы для работы многих сетевых протоколов, таких

как ARP, BOOTP или DHCP. С их помощью рабочая станция оповещает другие компьютеры о своем появлении в сети. Так же рассылка широковещательных кадров может возникать из-за некорректно работающего сетевого адаптера. Широковещательные кадры могут привести к нерациональному использованию полосы пропускания, особенно в крупных сетях.

Для того чтобы этого не происходило, важно ограничить область распространения широковещательного трафика (эта область называется широковещательным доменом) – организовать небольшие широковещательные домены, или виртуальные локальные сети (Virtual LAN, VLAN).

VLAN (Virtual Local Area Network – виртуальная локальная сеть) – это функция в роутерах и коммутаторах, позволяющая на одном физическом сетевом интерфейсе (Ethernet, Wi-Fi) создать несколько виртуальных локальных сетей. VLAN используют для создания логической топологии сети, которая никак не зависит от физической топологии.

1.2.1. Настройка VLAN на основе портов

При использовании VLAN на основе портов (Port-Based VLAN) каждый порт назначается в определенную VLAN, независимо от того, какой пользователь или компьютер подключен к этому порту. Это означает, что все пользователи, подключенные к этому порту, будут членами одной VLAN. Конфигурация портов статическая и может быть изменена только вручную.

Разделение в разные подсети компьютеров, подключенных к одному коммутатору, представлено на рис. 1.3. Компьютеры физически подключены к одному свитчу, но разделены в разные виртуальные сети VLAN 1 и VLAN 2. Компьютеры из разных виртуальных подсетей будут невидимы друг для друга. Как правило, одной VLAN соответствует одна подсеть. Компьютеры, находящиеся в разных VLAN, будут изолированы друг от друга. Каждая VLAN представляет отдельный широковещательный домен. Широковещательный трафик не будет транслироваться между разными VLAN.

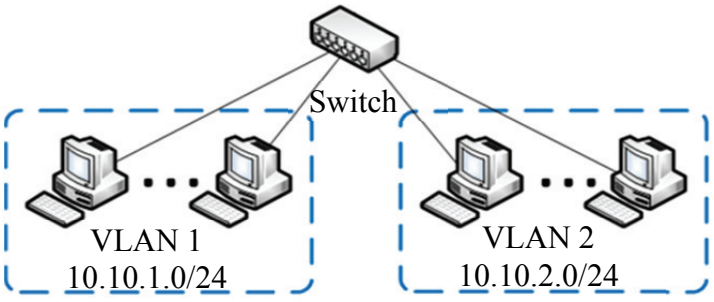


Рис. 1.3. VLAN на основе портов

Рассмотрим процесс настройки VLAN на основе портов через веб-интерфейс коммутатора D-Link. После подключения к коммутатору переходим в ветку VLAN – Port-Based VLAN. Изначально на коммутаторе настроена одна VLAN с VID = 1 (рис. 1.4), называемая default. По умолчанию все порты коммутатора входят в default VLAN. При настройке VLAN на основе портов соответствующие порты новых VLAN удаляются из default VLAN.

Рис. 1.4. VLAN default

Кликаем *Add VLAN* и указываем имя VLAN и порты (рис. 1.5).

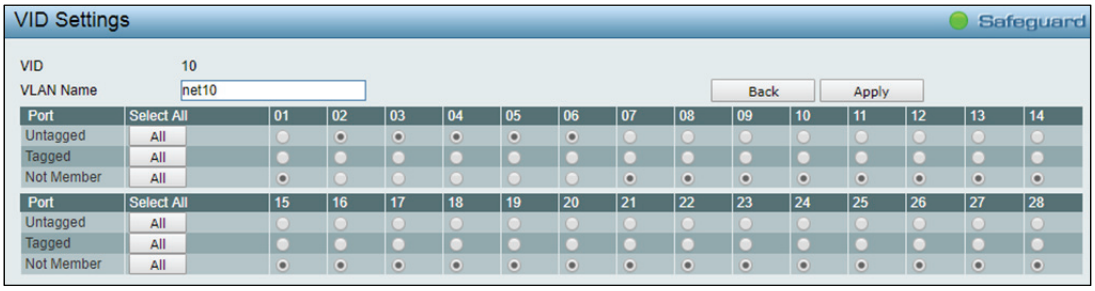


Рис. 1.5. Создание VLAN

Основные характеристики VLAN на основе портов:

1) применяются в пределах одного коммутатора. Если следует организовать несколько рабочих групп в пределах небольшой сети на основе одного коммутатора, например, необходимо разнести технический отдел и отдел продаж, то решение VLAN на базе портов оптимально подходит для данной задачи;

2) возможность изменения логической топологии сети без физического перемещения станций. Достаточно всего лишь изменить настройки порта с одной VLAN (например, VLAN технического отдела) на другую (VLAN отдела продаж), и рабочая станция сразу же получает возможность совместно использовать ресурсы с членами новой VLAN. Таким образом, VLAN обеспечивают гибкость при перемещениях, изменениях и наращивании сети;

3) каждый порт может входить только в одну VLAN. Для объединения виртуальных подсетей как внутри одного коммутатора, так и

между двумя коммутаторами нужно использовать сетевой уровень OSI-модели. Один из портов каждой VLAN подключается к интерфейсу маршрутизатора, который создает таблицу маршрутизации для пересылки кадров из одной подсети (VLAN) в другую (IP-адреса подсетей должны быть разными).

1.2.2. Настройка VLAN на основе стандарта IEEE 802.1Q

Построение VLAN на основе портов основано только на добавлении дополнительной информации к адресным таблицам коммутатора и не использует возможности встраивания информации о принадлежности к виртуальной сети в передаваемый кадр. Виртуальные локальные сети, построенные на основе стандарта IEEE 802.1Q, используют дополнительные поля кадра для хранения информации о принадлежности к VLAN при его перемещении по сети. С точки зрения удобства и гибкости настроек VLAN стандарта IEEE 802.1Q является лучшим решением по сравнению с VLAN на основе портов.

Порты 802.1Q могут быть в одном из следующих режимов:

– Tagged port (trunk-port) – порт пропускает пакеты, маркированные указанными номерами VLAN, но при этом сам никак не маркирует пакеты;

– Untagged port (access-port) – порт прозрачно пропускает немаркированный трафик для указанных VLAN, если трафик уходит в другие порты коммутатора за пределы указанной VLAN, то там он уже виден как маркированный номером этой VLAN;

– порт не принадлежит никаким VLAN и не учавствует в работе коммутатора.

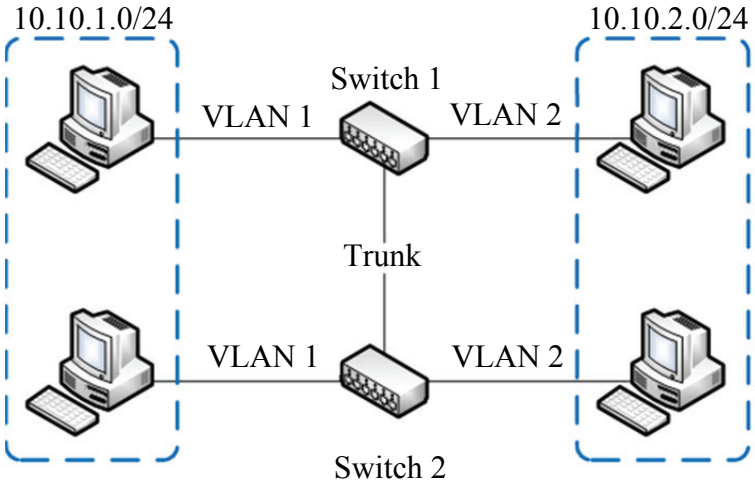


Рис. 1.6. Объединение в единую сеть компьютеров, подключенных к различным коммутаторам

Объединение в единую сеть компьютеров, подключенных к разным коммутаторам, показано на рис. 1.6 (см. на с. 8). Допустим, у вас есть компьютеры, которые подключены к разным свитчам, но их нужно объединить в одну сеть. Одни компьютеры мы объединим в виртуальную локальную сеть VLAN 1, а другие – в сеть VLAN 2.

Благодаря функции VLAN компьютеры в каждой виртуальной сети будут работать, словно подключены к одному и тому же свитчу. Компьютеры из разных виртуальных сетей VLAN 1 и VLAN 2 будут невидимы друг для друга.

Лабораторная работа № 1

***Цель*:** изучение технологии VLAN и настройка ее на коммутаторах марки D-Link.

***Задание*:** для выполнения данной лабораторной работы необходимо произвести подключение к коммутаторам второго уровня и использовать технологию VLAN для организации сегментации сети и уменьшения широковещательного трафика.

Для коммутаторов, приведенных на схеме (рис. 1.7), следует разработать план настройки портов.

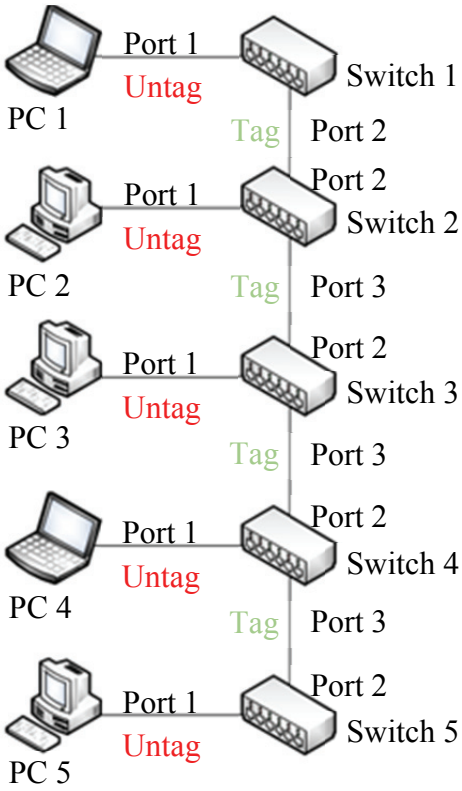


Рис. 1.7. Схема лабораторной работы

Необходимо разделить их так, чтобы PC 1, PC 3, PC 5 находились в одной VLAN, а PC 2, PC 4 – в другой VLAN. Для этого следует создать две VLAN: VLAN = 10 с именем net1 (PC 1, PC 3, PC 5) и VLAN = 20 с именем net2 (PC 2, PC 4).

При этом важно учесть, что для компьютеров необходимо использовать нетегированные порты, а для связей между коммутаторами – тегированные порты.

При настройке коммутаторов тегированные порты должны размещаться в VLAN = 10 и VLAN = 20, а нетегированные порты – только в той VLAN, к которой принадлежит компьютер.

Раздел 2

ОРГАНИЗАЦИЯ ОТКАЗОУСТОЙЧИВОГО DHCP-СЕРВЕРА

Роль сервера DHCP является стандартной для многих корпоративных сетей, так как сильно упрощает процесс настройки клиентов. Это накладывает определенные требования по обеспечению доступности серверов DHCP в локальной сети предприятия.

Windows Server 2012 предоставляет новый механизм обеспечения высокой доступности для роли DHCP. Два DHCP-сервера могут быть настроены для обеспечения высокой доступности сервиса DHCP через отказоустойчивость (failover relationship).

2.1. Организация динамической адресации в компьютерных сетях

Каждый компьютер в сетях TCP/IP имеет адреса трех уровней:
- физический (MAC-адрес);
- сетевой (IP-адрес);
- символьный (DNS-имя).

При построении информационных систем с большим числом узлов целесообразно использовать методы как статической, так и динамической адресации. Так, серверы, выполняющие все необходимые функции, связанные с управлением, используют всегда статические адреса, но при этом все клиенты, как правило, получают IP-адрес и другие параметры сети динамически. Это связано с тем, что в больших информационных системах число узлов может составлять тысячи, соответственно, ручная настройка каждого из них затруднительна. Поэтому целесообразно использовать методы автоматической настройки IP-параметров клиентских компьютеров, основанные на динамической адресации.

2.1.1. Динамическая адресация в компьютерных сетях

Как уже было сказано, IP-адреса могут назначаться администратором сети вручную. Это представляет для администратора утомительную процедуру. Ситуация усложняется еще тем, что многие пользователи

не обладают достаточными знаниями для того, чтобы конфигурировать свои компьютеры для работы в интерсети, и должны поэтому полагаться на администраторов.

Протокол Dynamic Host Configuration Protocol (DHCP) был разработан для того, чтобы освободить администратора от этих проблем. Основной функцией DHCP является динамическое назначение IP-адресов. Однако, кроме динамического, DHCP может поддерживать и более простые способы ручного и автоматического статического назначения адресов.

В ручной процедуре назначения адресов активное участие принимает администратор, который предоставляет DHCP-серверу информацию о соответствии IP-адресов физическим адресам или другим идентификаторам клиентов. Эти адреса сообщаются клиентам в ответ на их запросы к DHCP-серверу.

При автоматическом статическом способе DHCP-сервер присваивает IP-адрес (и, возможно, другие параметры конфигурации клиента) из пула наличных IP-адресов (его также называют scope, или диапазоном IP-адресов) без вмешательства оператора. Границы пула назначаемых адресов задает администратор при конфигурировании DHCP-сервера. Между идентификатором клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первичного назначения сервером DHCP IP-адреса клиенту. При всех последующих запросах сервер возвращает тот же самый IP-адрес.

При динамическом распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, что дает возможность впоследствии повторно использовать IP-адреса другими компьютерами. Служба DHCP обеспечивает надежный и простой способ конфигурации сети TCP/IP, гарантируя отсутствие конфликтов адресов за счет централизованного управления их распределением. Администратор управляет процессом назначения адресов с помощью параметра «продолжительности аренды», который определяет, как долго компьютер может использовать назначенный IP-адрес, перед тем как снова запросить его от сервера DHCP в аренду.

2.1.2. Принцип работы протокола DHCP

Выделяют три типа областей:

1) стандартные (описывают одну IP-сеть);

2) суперобласть (совокупность стандартных);

3) многоадресные (описывают IP-сети, предназначенные для многократной рассылки).

Стандартные области. Служат для объединения компьютеров в логические подсети в рамках одной физической сети. При этом администратор сначала создает область для каждой подсети, а затем использует ее для определения параметров клиентов.

Любая стандартная область характеризуется следующими свойствами:

– диапазон IP-адресов, из которых службой DHCP выбираются либо исключаются IP-адреса;

– маска подсети;

– срок аренды, назначаемый клиентам DHCP, которые динамически получают адреса.

В большинстве случаев на DHCP-серверы настраивается одна стандартная область, но если один DHCP-сервер обслуживает несколько сетей, то создается несколько стандартных областей, которые в дальнейшем объединяются в суперобласти. При этом важно следить, чтобы диапазон IP-адресов отдельных стандартных областей не пересекался.

Суперобласти. С их помощью можно получить ряд дополнительных возможностей:

1) поддержка DHCP-клиентов, расположенных на отдельном сегменте физической сети, в которой используется несколько логических IP-сетей. Если в каждой физической сети или подсети используется несколько логических сетей или подсетей, то такие конфигурации называются мультисетевыми;

2) поддержка удаленных DHCP-клиентов, расположенных на удаленной стороне агентов-ретрансляторов.

Суперобласти позволяют разрешать следующие проблемные ситуации:

− доступный диапазон в настоящее время исчерпан почти полностью, исходная область включает весь диапазон IP-сети для расширения адресного пространства для одного и того же физического сегмента сети с последующим объединением в суперобласти;

− клиенты должны перейти со временем на другую область, например, для перенумерации текущей IP-сети, в таком случае также создается новая область с последующим объединением в суперобласти;

− необходимость использования двух DHCP-серверов в физическом сегменте для управления различными логическими сетями.

Многоадресная область. В качестве диапазона адресов многоадресной групповой рассылки служит класс адресов D. Данные адреса не могут использоваться в стандартных областях.

Во всех TCP/IP-сетях каждый узел сначала должен получить индивидуальный IP (классы A, B, C). Без назначения такого адреса настройка

узла на поддержку и использование вторичных IP-адресов (адреса многоадресной рассылки) невозможна.

Членство в группе многоадресной рассылки является динамическим, что означает возможность присоединения в любое время IP-узлов или их выход.

При этом создается область многоадресной рассылки, которая будет назначать клиенту групповой адрес после получения индивидуального.

В DHCP-серверах можно резервировать за определенным MAC-адресом соответствующий IP-адрес, также можно в области добавлять исключения.

Исключения – это диапазон IP-адресов, из которого клиентам адреса не будут выдаваться. Как правило, в диапазон исключений попадают все статически заданные IP-адреса в сети.

Перечислим только основные параметры DHCP:

1) Subnet mask – маска подсети;
2) Router – список IP-адресов маршрутизаторов;
3) Domain Name Servers – список адресов DNS-серверов;
4) DNS Domain Name – DNS-суффикс клиента;
5) WINS Server Names – список адресов WINS-серверов;
6) LeaseTime – срок аренды (в секундах);
7) Renewal Time (T1) – период времени, через который клиент начинает продлевать аренду;
8) Rebinding Time (T2) – период времени, через который клиент начинает осуществлять широковещательные запросы на продление аренды.

Параметры могут применяться на следующих уровнях:

- уровень сервера;
- уровень области действия;
- уровень класса;
- уровень клиента (для зарезервированных адресов).

Параметры, определенные на нижележащем уровне, перекрывают параметры вышележащего уровня: например, параметры клиента имеют больший приоритет, чем параметры сервера. Самый высокий приоритет имеют параметры, настроенные вручную на клиентском компьютере.

Уровень класса используется для объединения клиентов в группы и применения для этой группы отдельных параметров. Отнести клиента к определенному классу можно, применив утилиту *ipconfig* с ключом /setclassid.

Процесс функционирования служб DHCP заключается в обмене сообщениями между сервером и клиентом. Список используемых сообщений представлен в табл. 2.1.

Таблица 2.1

Типы DHCP-сообщений

Тип сообщения	Направление	Значение
DHCPDISCOVER (DHCP-обнаружение)	Клиент → сервер	Широковещательный запрос для обнаружения DHCP-сервера
DHCPOFFER (DHCP-предложение)	Сервер → клиент	Ответ на DHCPDISCOVER, содержит предлагаемые сетевые параметры
DHCPREQUEST (DHCP-запрос)	Клиент → сервер	Запрос предложенных параметров
DHCPACK (DHCP-подтверждение)	Сервер → клиент	Подтверждение сетевых параметров
DHCPNAK (DHCP-несогласие)	Сервер → клиент	Отклонение запроса клиента
DHCPDECLINE (DHCP-отказ)	Клиент → сервер	Отказ клиента от предложенных параметров
DHCPRELEASE (DHCP-освобождение)	Клиент → сервер	Освобождение арендованного IP-адреса
DHCPINFORM (DHCP-информация)	Клиент → сервер	Запрос дополнительных параметров

Диаграмма переходов, иллюстрирующая принципы работы протокола DHCP, приведена на рис. 2.1. На схеме овалами обозначены состояния, в которых может находиться DHCP-клиент. Из одного состояния в другое клиент может переходить только по дугам. Каждая дуга помечена дробью, числитель которой обозначает событие (чаще всего это сообщение от DHCP-сервера), после которого клиент переходит в соответствующее состояние, а знаменатель описывает действия DHCP-клиента при переходе. Черточка в числителе означает безусловный переход.

Начальное состояние, в котором оказывается служба DHCP-клиента при запуске, – это «Инициализация». Из этого состояния происходит безусловный переход в состояние «Выбор» с рассылкой широковещательного сообщения DHCPDISCOVER. DHCP-серверы (в одной сети их может быть несколько), принимая сообщение, анализируют свою базу данных на предмет наличия свободных IP-адресов. В случае успеха серверы отправляют сообщение DHCPOFFER, которое помимо IP-адреса содержит дополнительные параметры, призванные помочь клиенту выбрать лучшее предложение. Сделав выбор, клиент посылает широковещательное сообщение DHCPREQUEST, запрашивая предложенный IP-адрес и требуемые параметры (например,

маска подсети, шлюз по умолчанию, IP-адреса DNS-серверов и др.), и переходит в состояние «Запрос». Данное сообщение требуется посылать широковещательно (т. е. оно должно доставляться всем компьютерам подсети), чтобы DHCP-серверы, предложения которых клиент отклонил, знали об отказе.

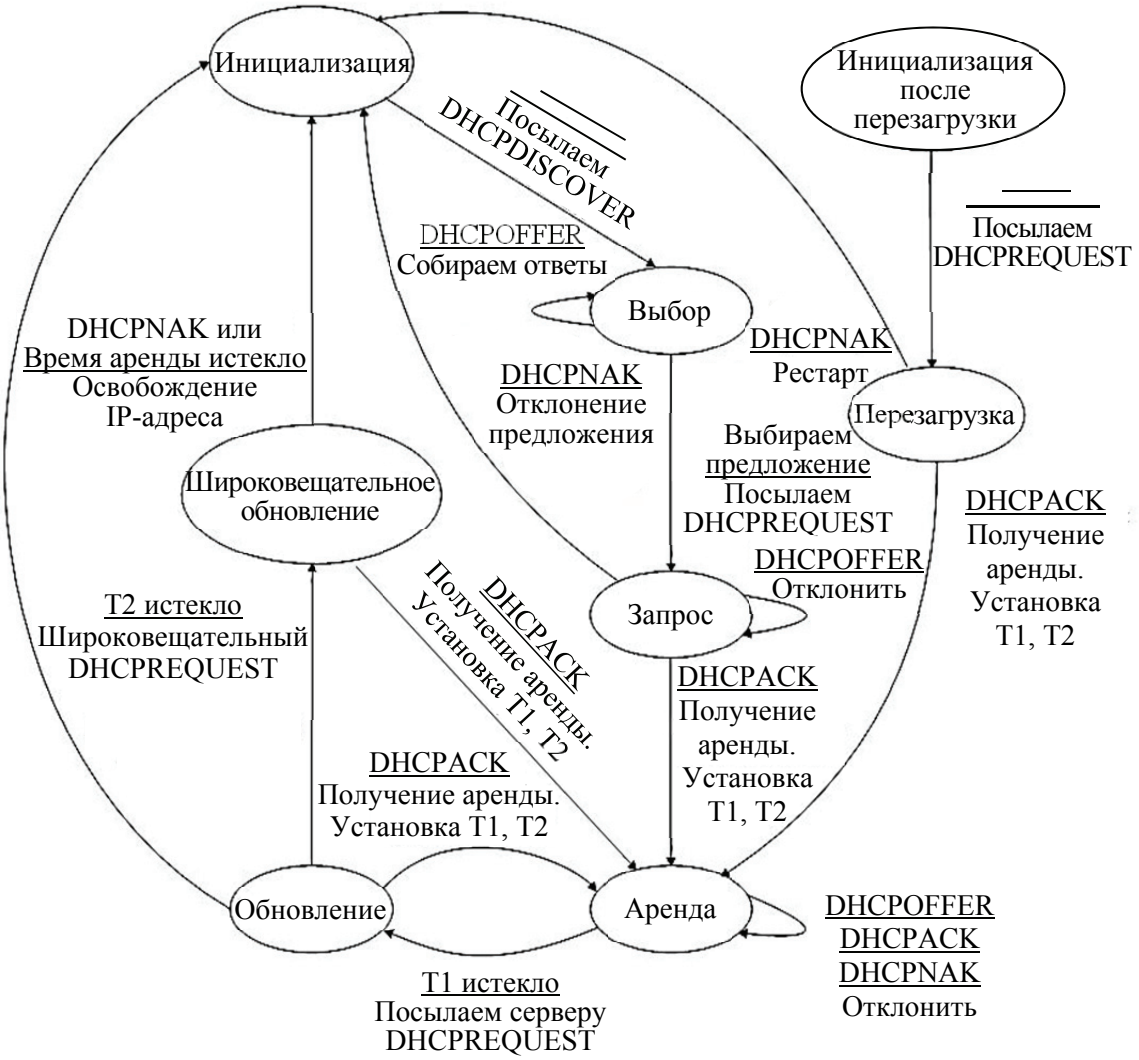


Рис. 2.1. Диаграмма переходов,
иллюстрирующая принципы работы протокола DHCP

В состоянии «Запрос» клиент ожидает подтверждение сервера о возможности использования предложенных сетевых параметров. В случае прихода такого подтверждения (сообщение DHCPACK) клиент переходит в состояние «Аренда», одновременно начиная отсчет интервалов времени T1 и T2. Если сервер по каким-либо причинам не готов предоставить клиенту предложенный IP-адрес, он посылает сообщение DHCPNAK. Клиент реагирует на это сообщение переходом

в исходное состояние «Инициализация», чтобы снова начать процесс получения IP-адреса.

Состояние «Аренда» является основным рабочим состоянием – у клиента присутствуют все необходимые сетевые параметры, и сеть может успешно функционировать.

Через временной интервал T1 от момента получения аренды (обычно T1 равно половине общего времени аренды) DHCP-клиент переходит в состояние «Обновление» и начинает процесс обновления аренды IP-адреса. Сначала клиент посылает DHCP-серверу сообщение DHCPREQUEST, включающее арендованный IP-адрес. Если DHCP-сервер готов продлить аренду этого адреса, то он отвечает сообщением DHCPACK, и клиент возвращается в состояние «Аренда» и заново начинает отсчитывать интервалы T1 и T2.

В случае если в состоянии «Обновление» по истечении интервала времени T2 (который обычно устанавливается равным 87,5% от общего времени аренды) все еще не получено подтверждение DHCPACK, клиент переходит в состояние «Широковещательное обновление» с рассылкой широковещательного сообщения DHCPREQUEST. Такая рассылка делается в предположении, что DHCP-сервер поменял свой IP-адрес (или перешел в другую подсеть) и передал свою область действия другому серверу. В этом состоянии получение DHCPACK возвращает клиента в состояние «Аренда» и аренда данного IP-адреса продлевается.

В процессе работы может оказаться, что время аренды не истекло, а служба DHCP-клиента прекратила работу (например, при перезагрузке). В этом случае DHCP-клиент начинает работу в состоянии «Инициализация после перезагрузки», рассылает широковещательное сообщение DHCPREQUEST и переходит в состояние «Перезагрузка». В случае подтверждения продления аренды (сообщение DHCPACK от DHCP-сервера) клиент переходит в состояние «Аренда». Иначе (сообщение DHCPNAK) клиент оказывается в состоянии «Инициализация».

2.2. Установка и настройка DHCP-сервера

Рассмотрим настройку DHCP-сервера на примере операционной системы Windows Server 2012.

Установка и авторизация сервера DHCP.

1. Установка службы DHCP выполняется так же, как и установка любой другой компоненты Windows Server: *Пуск* → *Панель управления* →

Установка и удаление программ → *Установка компонентов Windows* → *Сетевые службы* → кнопка *Состав* → выбрать пункт *DHCP* → кнопки *OK*, *Далее* и *Готово* (если потребуется, то следует указать путь к дистрибутиву системы).

Также можно установить DHCP-сервер, используя *Server Manager* (*Диспетчер серверов*), а именно *Start* (*Пуск*) → *Server Manager* (*Диспетчер серверов*), общий вид которого показан на рис. 2.2.

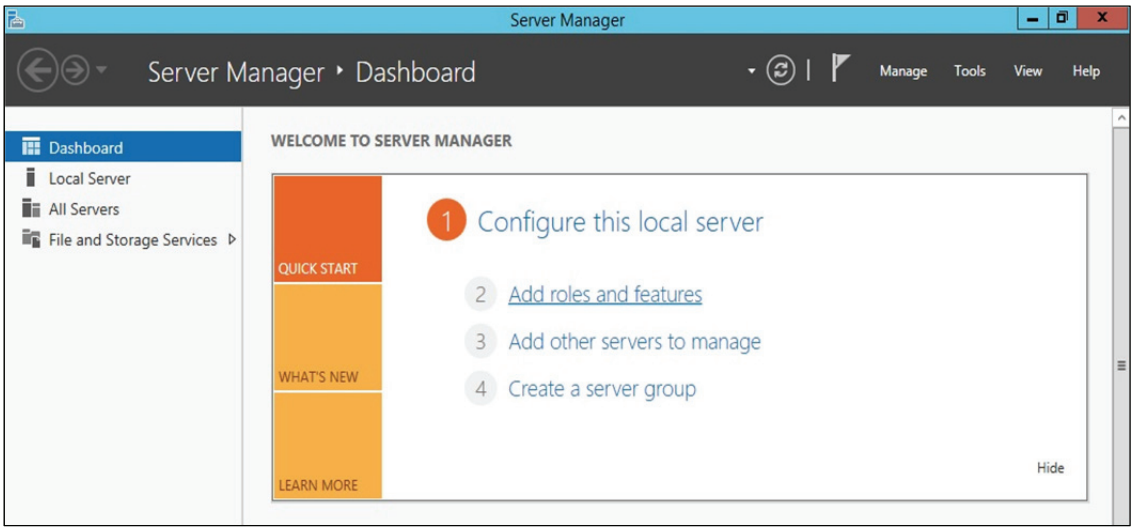


Рис. 2.2. Общий вид *Server Manager*

2. После чего нажимаем *Add Roles and Features Wizard* (*Добавить роль сервера*), что можно сделать непосредственно через быстрый запуск, а можно через меню *Управление*, и на странице приветствия щелкаем *Next* (*Далее*).

3. По умолчанию уже выбран необходимый пункт, т. е. *Role-based or feature-based installation* (*Установка ролей или компонентов*), и поэтому кликаем *Next* (*Далее*) (рис. 2.3).

4. Затем выбираем, на какой сервер или виртуальный жесткий диск будет устанавливаться DHCP-сервер (в нашем случае локально, т. е. этот же самый сервер, также необходимо заметить, что IP-адрес сервера является статическим, следовательно, создаем область в этой же подсети). Далее определяем, какую роль собираемся устанавливать, и, соответственно, указываем DHCP-сервер (рис. 2.4).

5. После нажатия откроется окно, в котором сразу предложат выбрать для установки средства администрирования DHCP-сервера. Необходимо согласиться, иначе далее все равно придется это выбирать, так как администрировать DHCP будем с данного компьютера. Затем жмем *Add Features* (*Добавить компоненты*) (см. рис. 2.5 на с. 20).

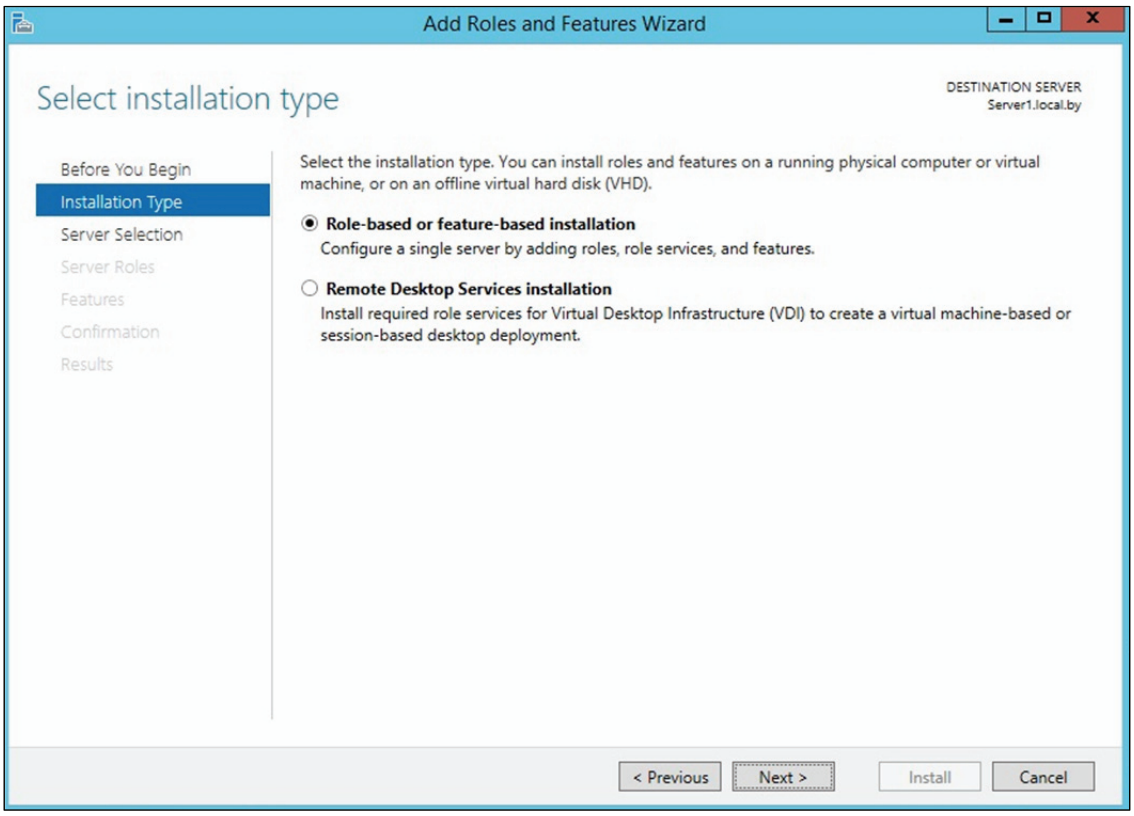


Рис. 2.3. Выбор опции установки ролей и компонент

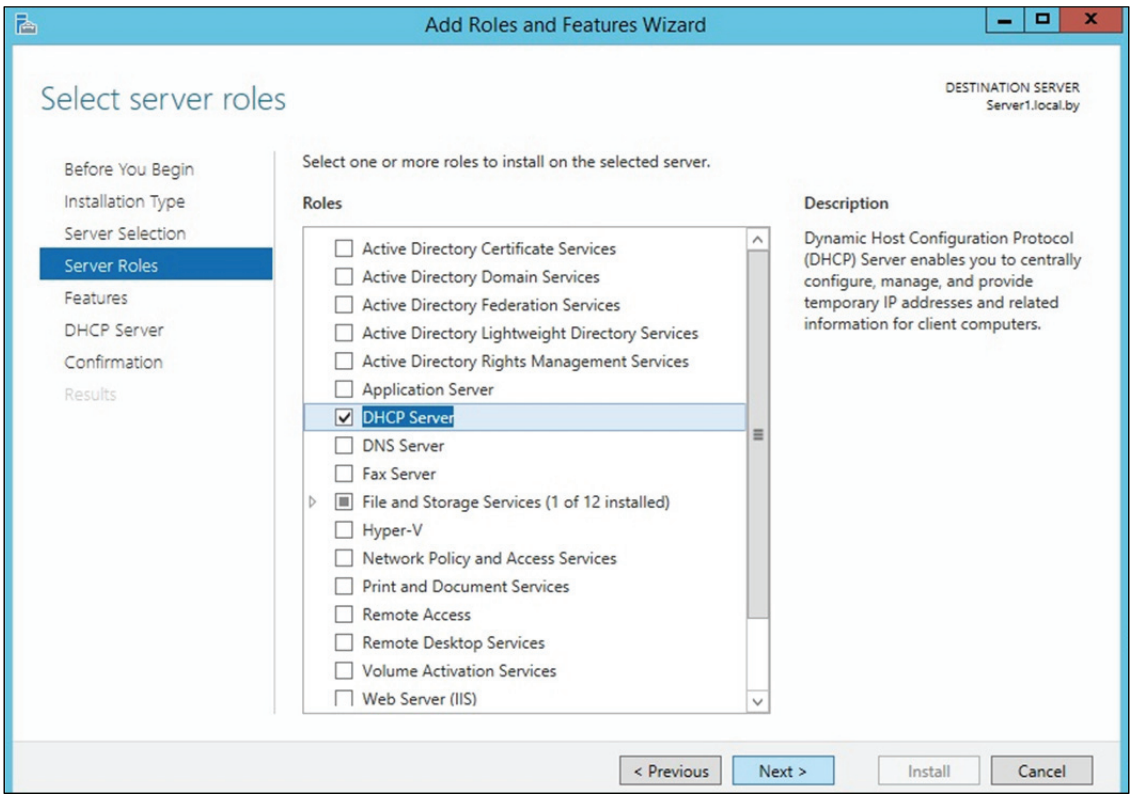


Рис. 2.4. Выбор устанавливаемой роли

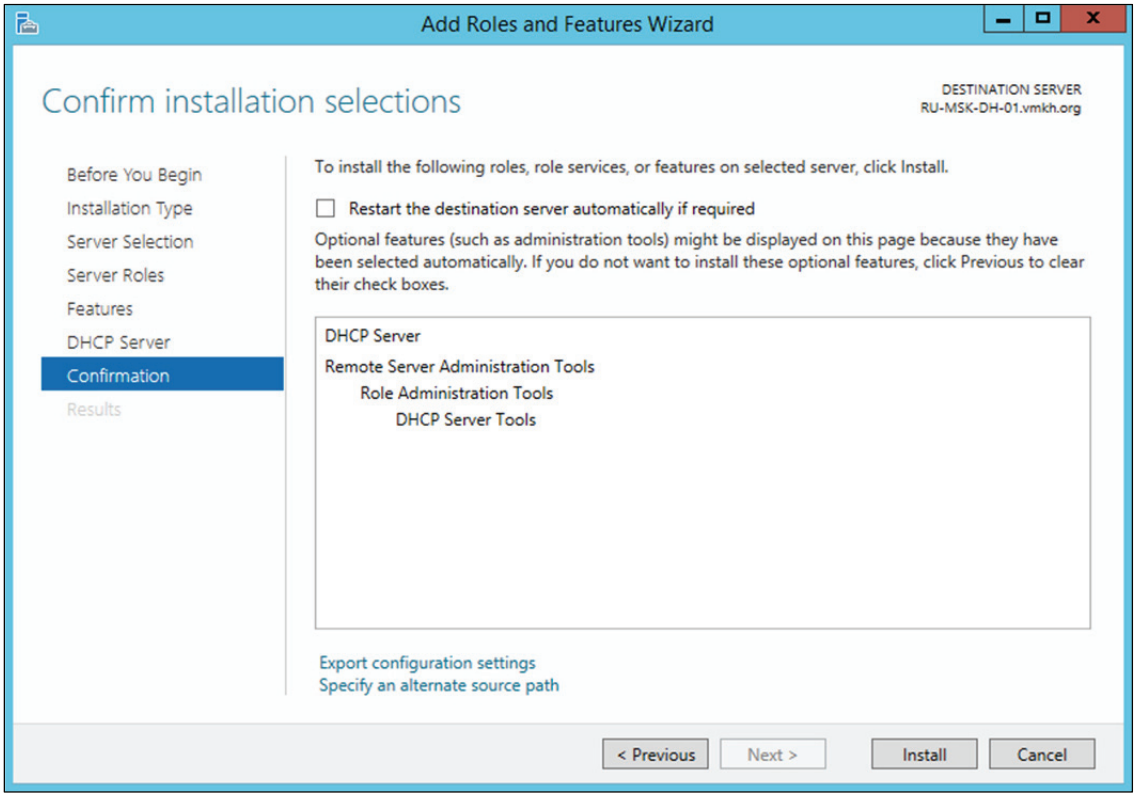


Рис. 2.5. Выбор средств администрирования компонентов

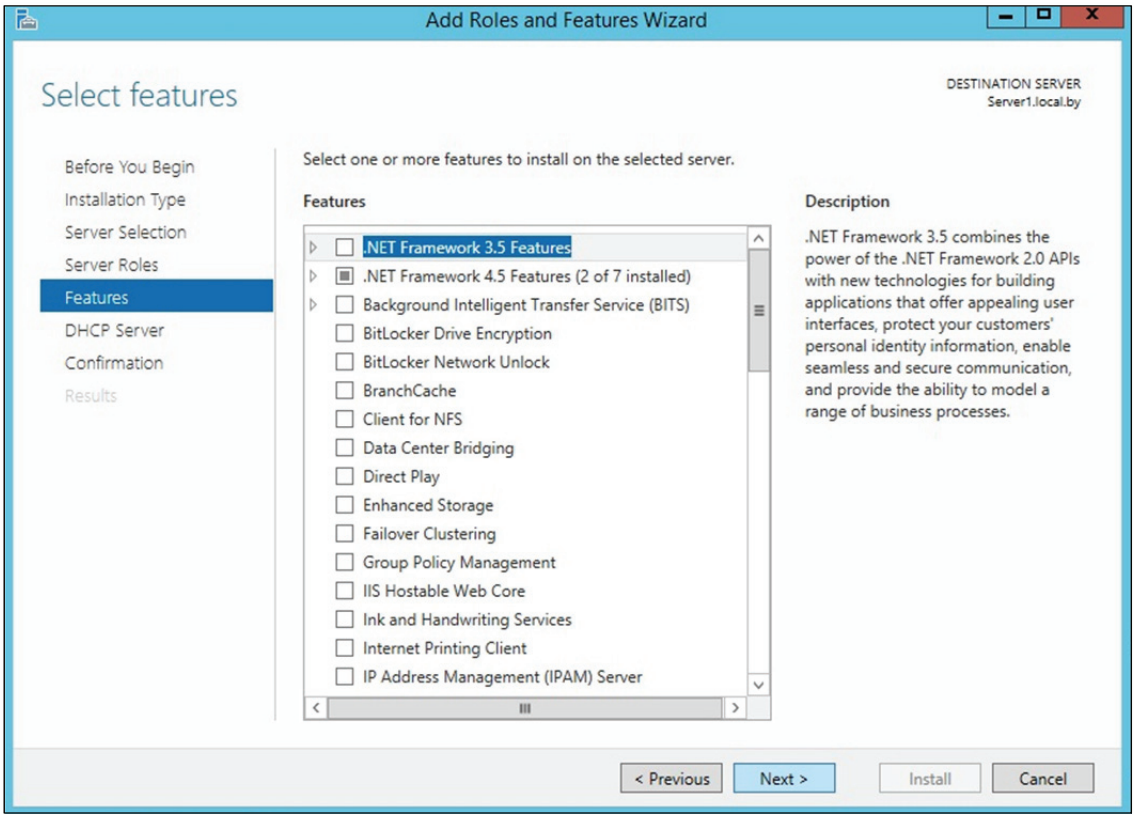


Рис. 2.6. Выбор компонент устанавливаемой роли

6. Далее будет предложено указать необходимые компоненты. Если на прошлом шаге были выбраны *Add Features* (*Добавить компоненты*), то необходимые компоненты уже будут заданы, а соответственно, кликаем *Next* (*Далее*) (см. рис. 2.6 на с. 20).

7. Еще на нескольких последующих этапах также щелкаем *Next* (*Далее*), и затем начнется установка DHCP-сервера (рис. 2.7).

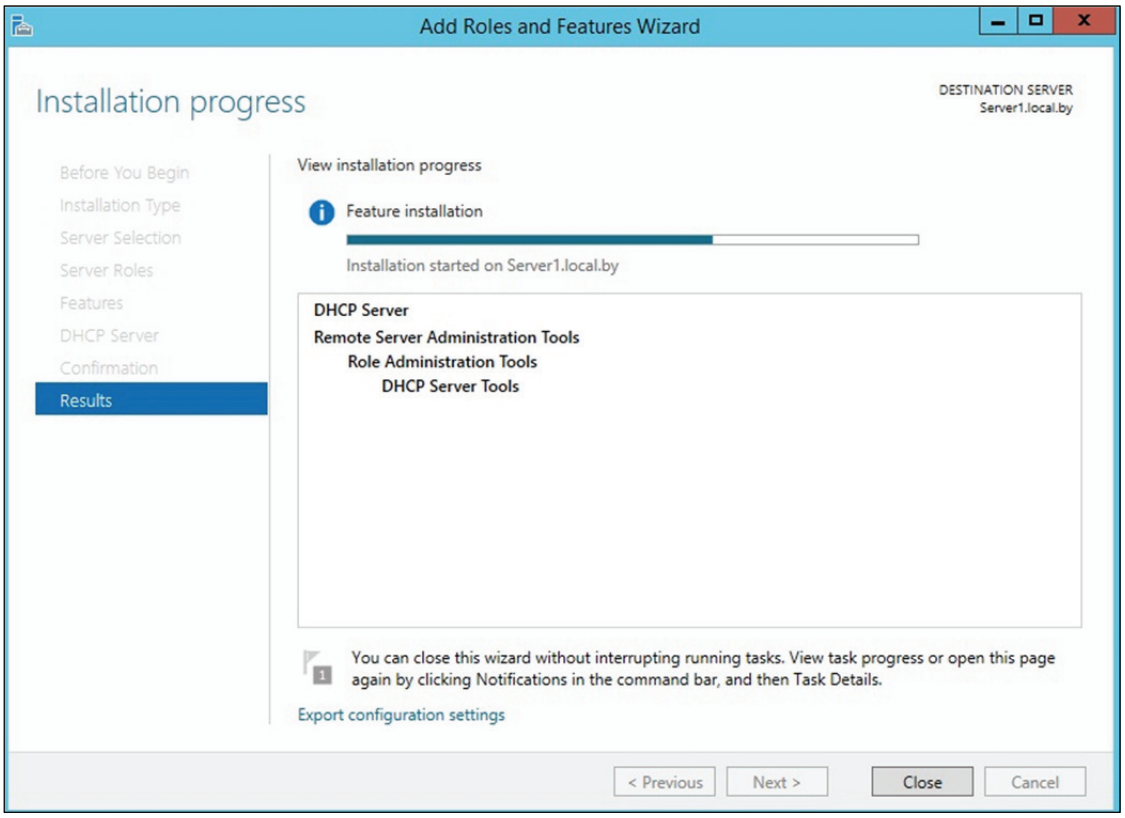


Рис. 2.7. Установка DHCP-сервера

8. После завершения установки будет предложено выполнить предварительную настройку. Рассмотрим настройку DHCP-сервера далее отдельно.

Настройка параметров DHCP-сервера. После установки DHCP-сервер и средства его администрирования необходимо настроить.

1. Для этого запускаем оснастку управления DHCP-сервером. Это можно сделать через *Server Manager* (*Диспетчер серверов*), меню *Tools* (*Средства*) (рис. 2.8).

2. Создать область можно, щелкнув правой кнопкой мыши на имени сервера и выбрав пункт меню *New Scope* (*Создать область*) (или аналогичный пункт в меню *Действие* консоли DHCP) (рис. 2.9). Консоль запустит *Мастер создания области*, который позволяет по шагам определить все необходимые параметры.

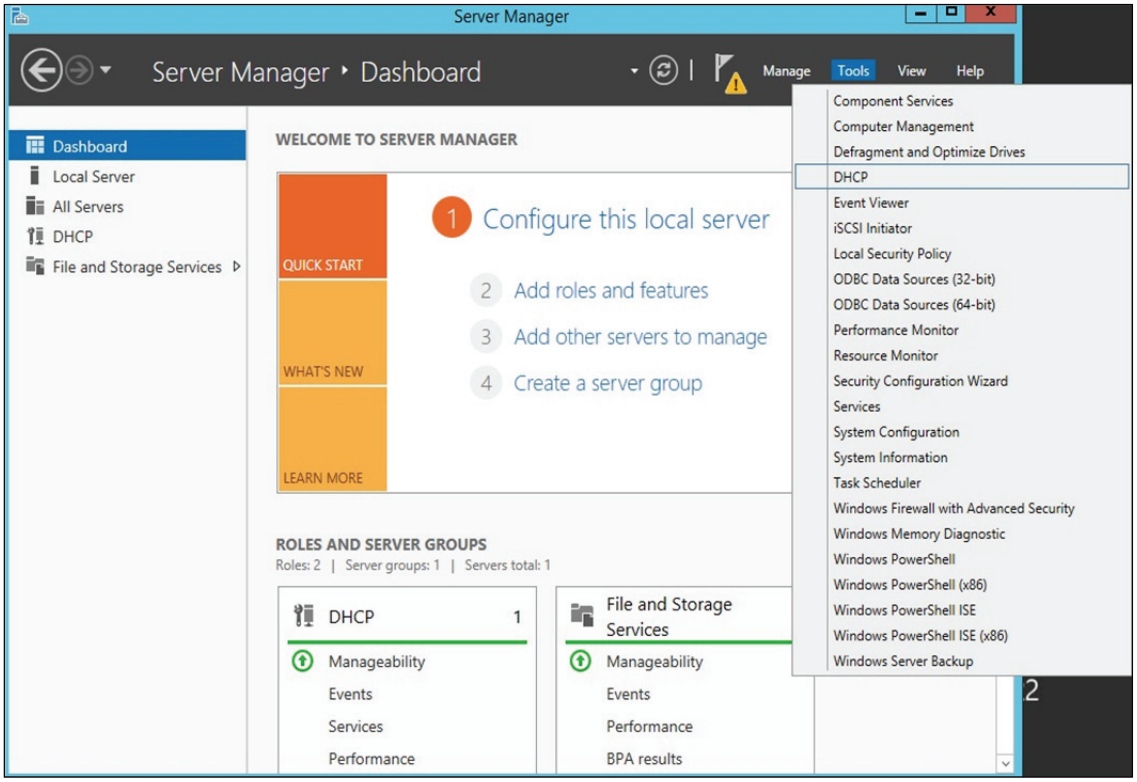


Рис. 2.8. Запуск DHCP-сервера

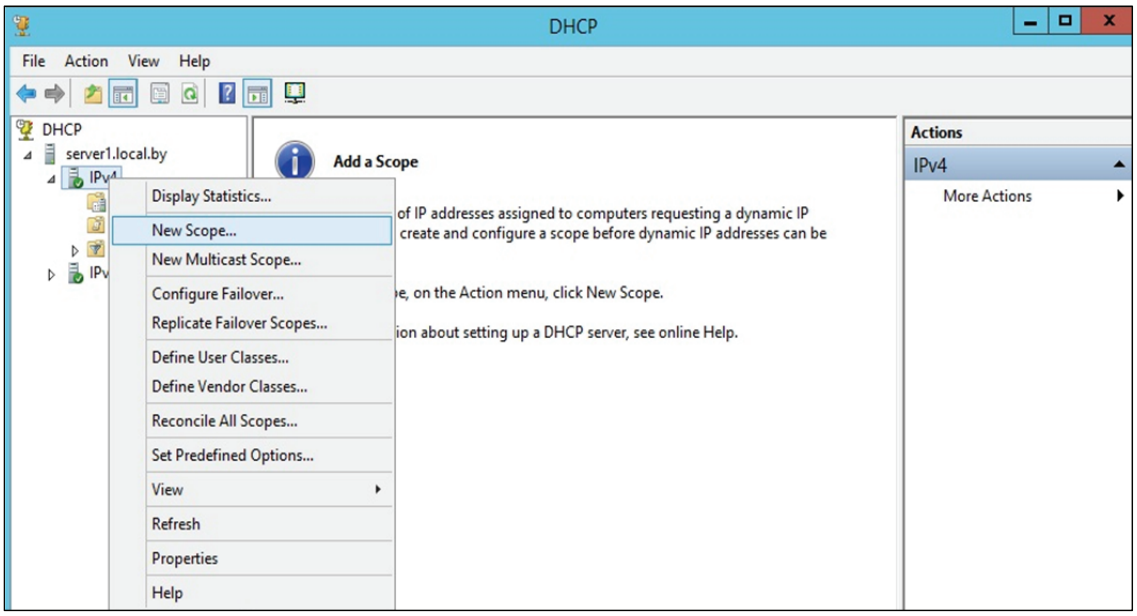


Рис. 2.9. Создание новой области DHCP-сервера

Имя и описание области. В больших сетях именование областей и задание их краткого описания облегчает работу администратора за счет более наглядного отображения в консоли всех созданных областей (рис. 2.10).

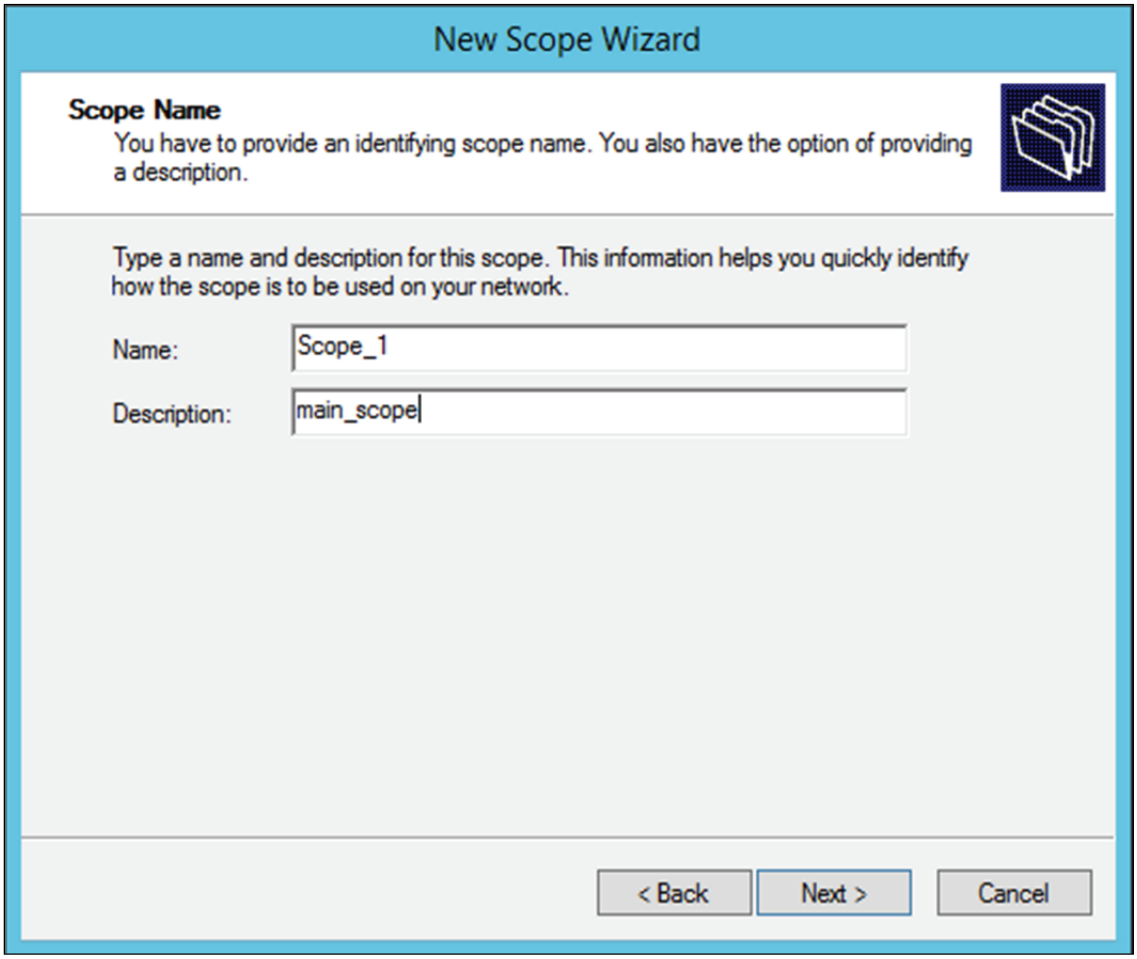


Рис. 2.10. Создание области DHCP-сервера

Дальнейший процесс создания и настройки области в Windows Server 2012 практически ничем не отличается от настройки Windows Server 2003, которая была рассмотрена и изучена в курсе «Компьютерные сети». Фактически необходимо определить диапазон IP-адресов и маски подсети (в данном примере используется подсеть с Network ID 10.90.90.0 и маской 24 бита) (рис. 2.11). Отметим, что при настройке каждый должен использовать диапазон IP-адресов и другие параметры исходя из выбранного варианта задания.

Добавление исключений. На данном шаге задаются диапазоны IP-адресов, которые будут исключены из процесса выдачи адресов клиентам (все статические IP-адреса должны быть обязательно исключены из действующего диапазона адресов).

В рассмотренном на рис. 2.12 примере исключаются адреса обоих серверов: 10.90.90.1 и 10.90.90.2.

Срок действия аренды. Стандартный срок действия – 8 дней (рис. 2.13).

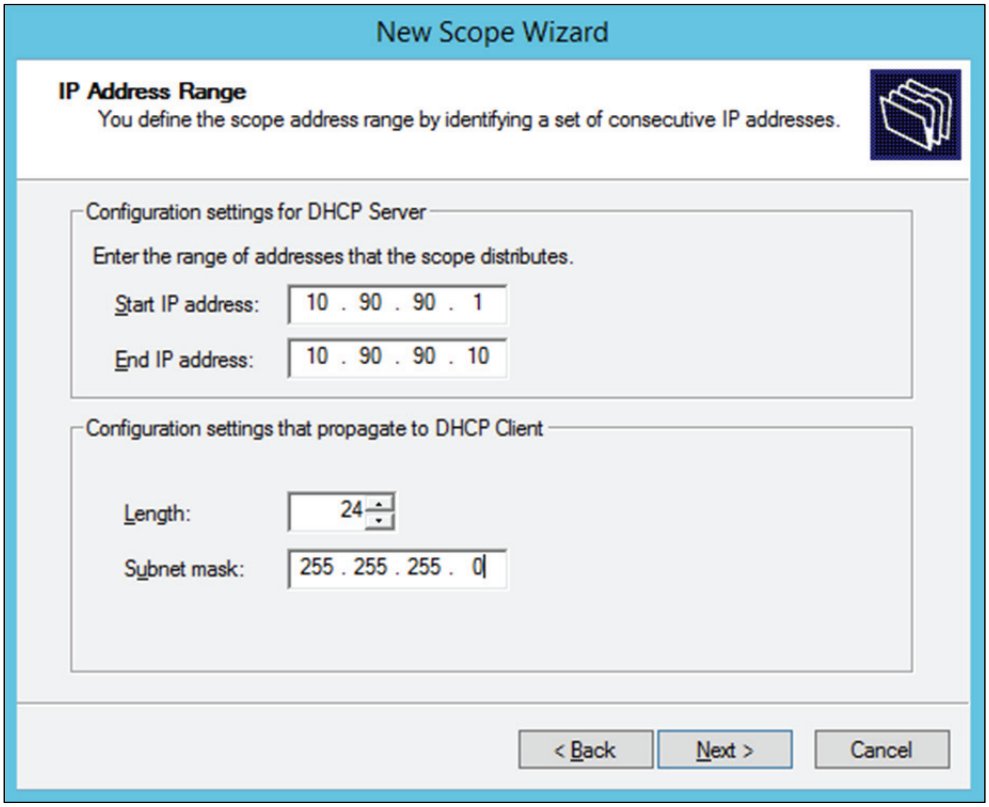


Рис. 2.11. Определение диапазона адресов области

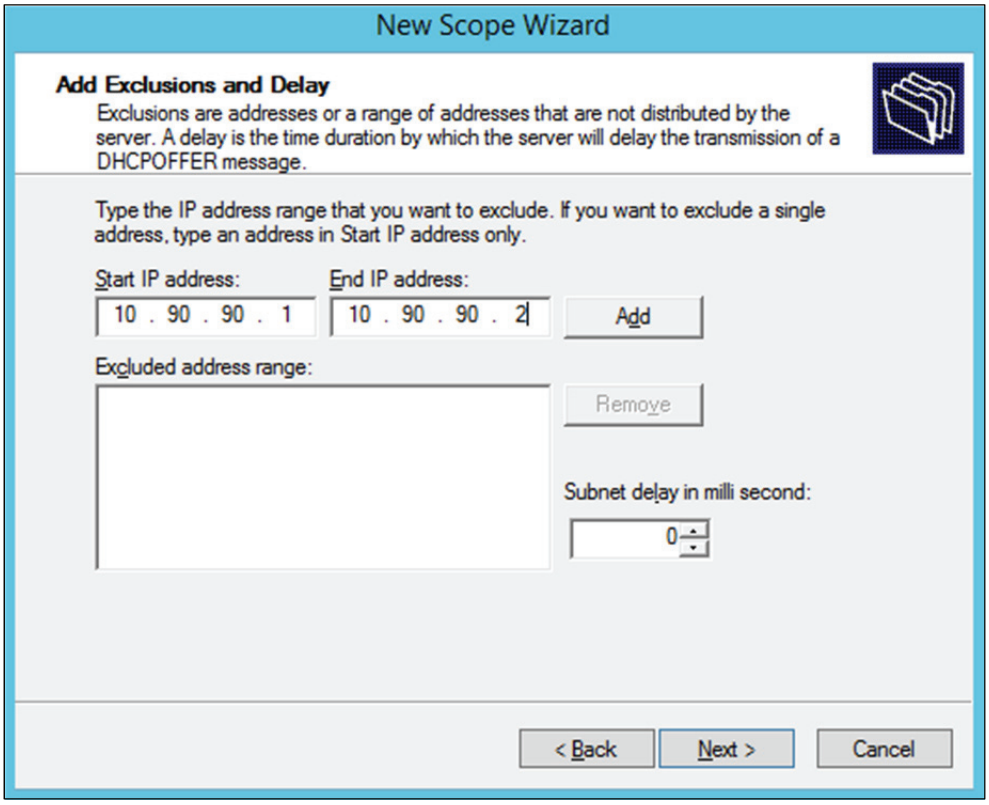


Рис. 2.12. Добавление исключающего диапазона адресов области

Рис. 2.13. Определение срока аренды клиентом адресов

Если в сети редко происходят изменения (добавление или удаление сетевых узлов, перемещение сетевых узлов из одной подсети в другую), то срок действия можно увеличить, это сократит количество запросов на обновление аренды. Если же сеть более динамичная, то срок аренды можно уменьшить, это позволит быстрее возвращать в пул IP-адреса, которые принадлежали компьютерам, уже удаленным из данной подсети.

Далее мастер предложит настроить параметры, специфичные для узлов IP-сети, относящихся к данной области, например маршрутизатор (основной шлюз), адрес DNS-сервера (можно назначить несколько адресов; рис. 2.14), адрес WINS-сервера (аналогично серверу DNS можно также назначить несколько адресов).

Запрос на активацию области. IP-адреса, заданные в созданной области, не будут выдаваться клиентам, пока область не будет активирована (рис. 2.15).

Далее завершаем работу мастера, и область готова к использованию. Если какие-либо параметры (например, адреса серверов DNS или WINS) являются общими для всех областей, управляемых данным DHCP-сервером, то такие параметры лучше определить не в разделе параметров каждой области, а в разделе параметров самого сервера.

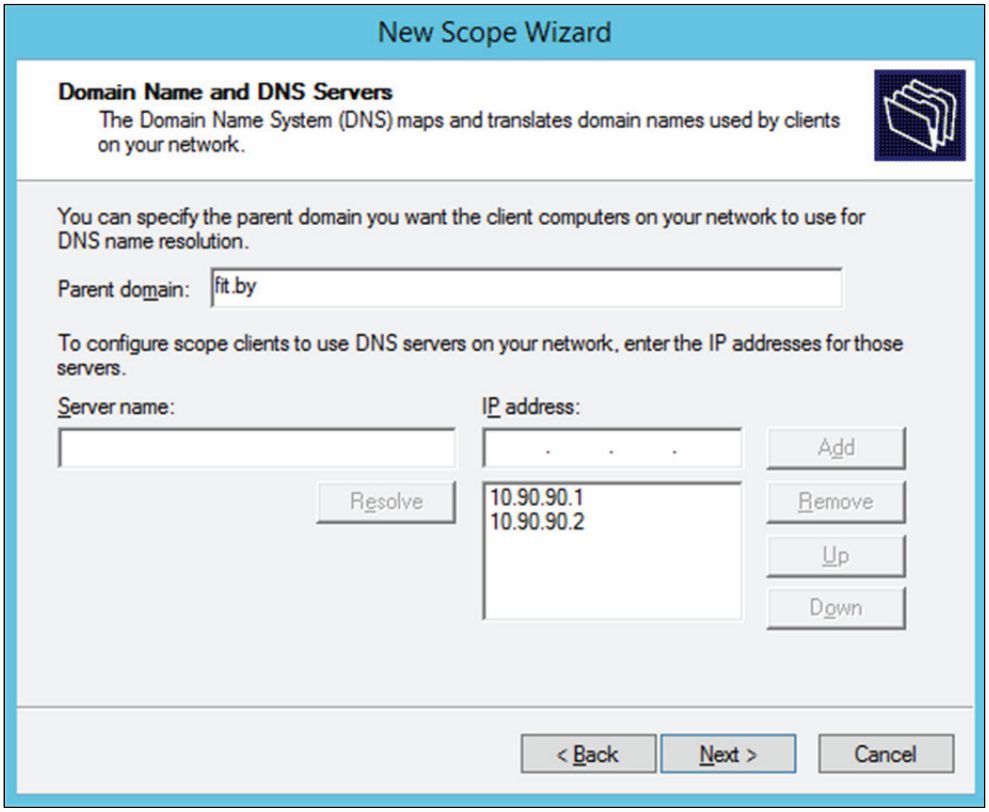


Рис. 2.14. Добавление адреса DNS-сервера, распределяемого областью

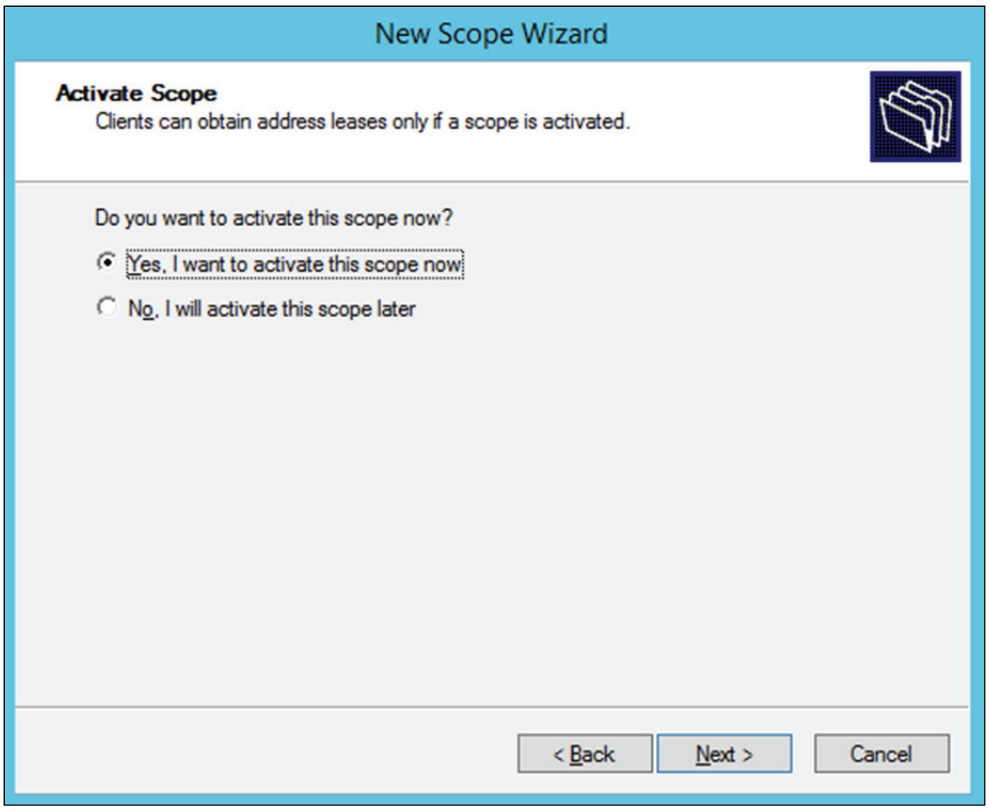


Рис. 2.15. Запрос на активацию области DHCP-сервера

Организация отказоустойчивости DHCP-сервера. Существует два режима настройки отказоустойчивости DHCP для разных топологий внедрения: балансировка нагрузки (*Load balance*) и горячее резервирование (*Hot standby*).

Балансировка нагрузки используется, когда все настроенные DHCP-серверы обрабатывают клиентские запросы, при этом процент обрабатываемых запросов конкретным сервером настраивается дополнительно (Active-Active конфигурация).

Горячее резервирование соответствует Active-Passive конфигурации. Необходимо будет указать, какой DHCP-сервер будет обрабатывать клиентские запросы, второй в это время будет находиться в резерве. Резервный сервер не занимается обслуживанием клиентских запросов, пока работает первый сервер. При этом он получает все обновления информации об аренде адресов от работающего сервера и сохраняет ее в своей базе.

Отказоустойчивые DHCP-серверы могут находиться в разных подсетях и даже в разных географических регионах.

Рассмотрим настройку отказоустойчивого DHCP-сервера.

1. Щелкаем правой клавишей по области и выбираем *Configure Failover* (*Настройка отказоустойчивости*).

2. На экране *Introduction to DHCP Failover* (*Предисловие об отказоустойчивости*) оставляем выделенным чекбокс *Select all* (*Выбрать всё*) (рис. 2.16).

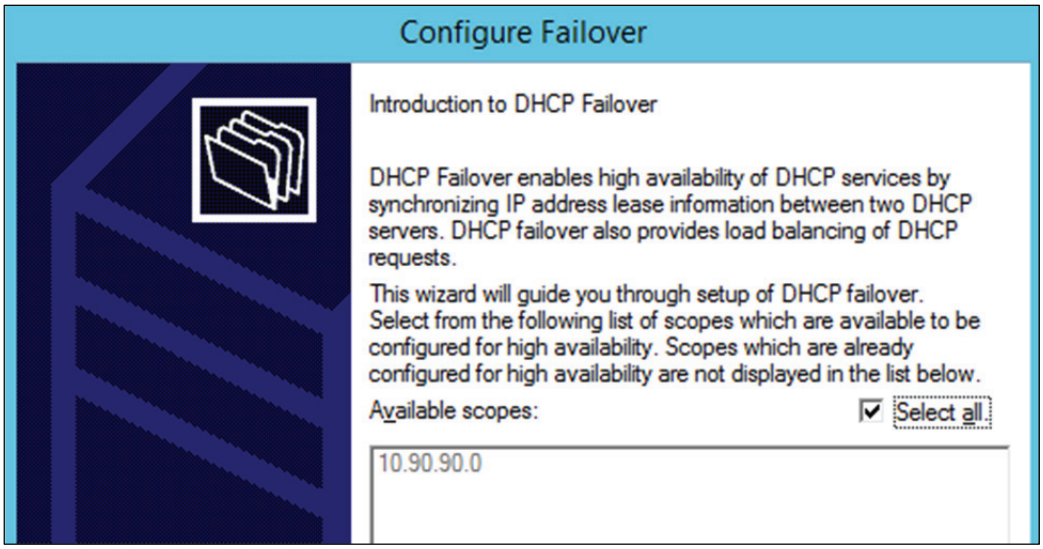


Рис. 2.16. Выбор области для отказоустойчивости

3. Кликаем *Next* (*Далее*).

4. На экране *Specify the partner server to use for failover* (*Указание партнерского сервера для отказоустойчивости*) вводим адрес *Partner*

Server (*Партнерский сервер*) для отказоустойчивости и жмем *Add Server* (*Добавить сервер*) (рис. 2.17).

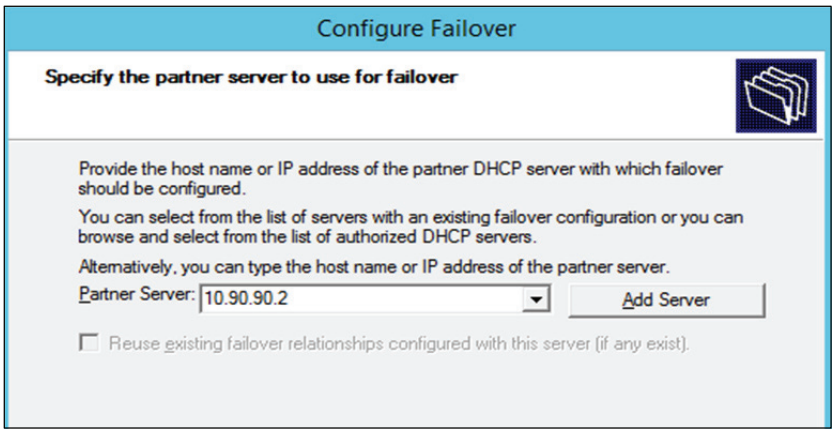


Рис. 2.17. Указание партнерского сервера
для отказоустойчивости

5. Щелкаем *Next* (*Далее*) и переходим к конфигурации отказоустойчивости DHCP-сервера (рис. 2.18).

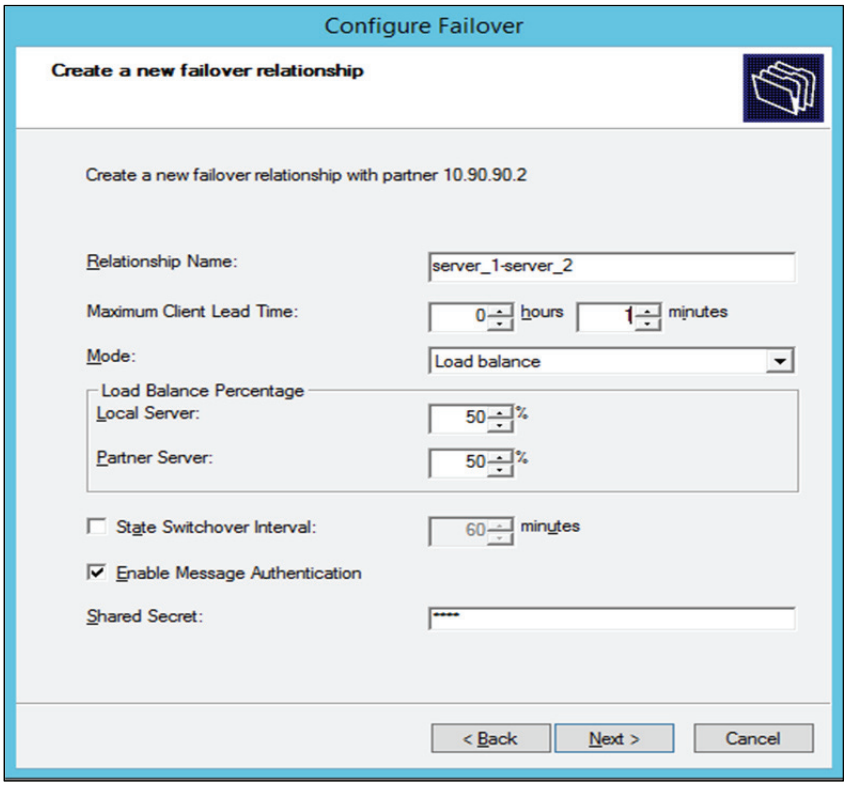


Рис. 2.18. Конфигурация отказоустойчивости DHCP-сервера

Опция *Relationship Name* (*Имя отношений*) – это уникальное имя конфигурации, которое требуется для конфигурирования отказоустойчивости

отношений между двумя серверами. Поскольку множество таких отношений может существовать на одном или более сервере, каждое именование такого схождения должно быть уникальным для сервера.

Опция *Maximum Client Lead Time* (*Максимальное время выполнения заказа клиента*) определяет временной период для аренды адреса клиента, обратившегося к отказоустойчивому серверу.

Опция *Mode* (*Режим*) определяет режим работы DHCP: *Hot standby* (*Горячее резервирование*) или *Load balance* (*Балансировка нагрузки*).

В режиме горячего резервирования два сервера работают в отказоустойчивой конфигурации, в которой активный сервер отвечает за выдачу в аренду IP-адреса и информацию о конфигурации для всех клиентов в области или подсети, тогда как вторичный сервер берет на себя его функции, если основной сервер становится недоступным. Сервер считается первичным или вторичным в контексте подсети.

В режиме балансировки нагрузки, который предлагается по умолчанию, два сервера одновременно выдают IP-адреса и опции для клиентов данной подсети. Клиентские запросы к серверам балансировки нагрузки распределяются между двумя серверами (необходимо задать желаемое процентное соотношение).

Опция *State Switchover Interval* (*Интервал переключения состояний*). Сервер, который утратил связь со своим партнером, переходит в состояние прерванного соединения. Потеря связи может означать проблемы на сетевом уровне, либо сервер-партнер просто может быть выключен. Поскольку для сервера не существует способа для выявления причин потери связи со своим партнером, сервер будет продолжать поддерживать состояние прерванного соединения, пока администратор вручную изменяет состояние партнера на «Недоступен». Альтернативным режимом будет автопереключение по таймауту, по умолчанию равному 10 мин.

Опция *Enable Message Authentication* (*Включить аутентификацию сообщений*). Для настройки проверки подлинности сообщений мастер конфигурирования DHCP failover предлагает администратору указать общий секрет на каждом из серверов.

6. Изменяем значение *Maximum Client Lead Time* (*Максимальное время выполнения заказа клиента*) до 1 мин для тестирования (рис. 2.19).

7. Оставляем для режима *Mode* (*Режим*) по умолчанию – *Load balance* (рис. 2.19).

8. Изменяем *State Switchover Interval* (*Интервал переключения состояний*) до 20 мин.

9. Предоставляем общий секрет *Shared Secret* (*Общий секрет*).

10. Кликаем *Next* (*Далее*).

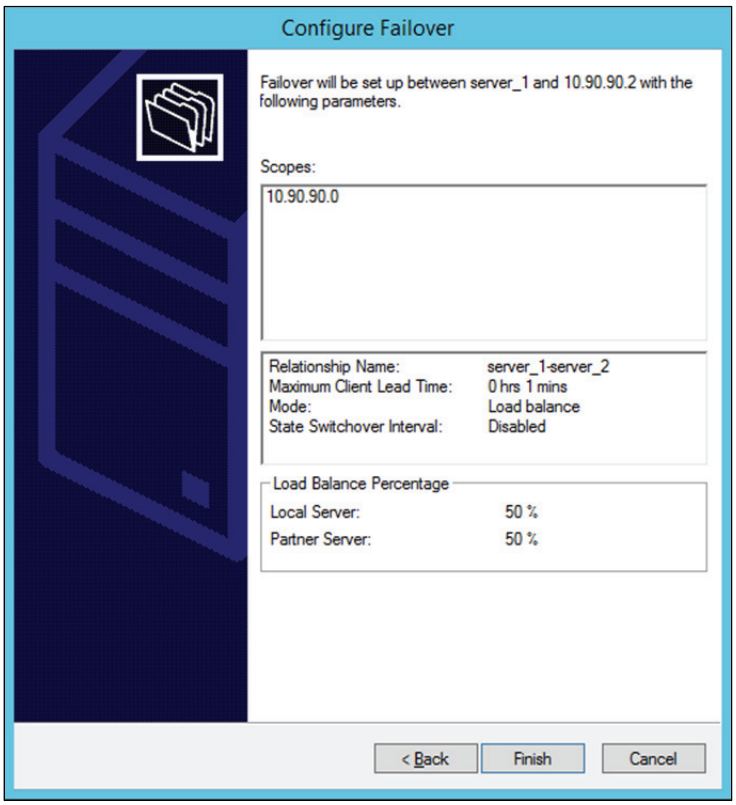


Рис. 2.19. Запрос на активацию
отказоустойчивости DHCP-сервера

11. Щелкаем *Finish* (*Завершить*) и переходим к окну завершения установки отказоустойчивого DHCP-сервера (рис. 2.20).

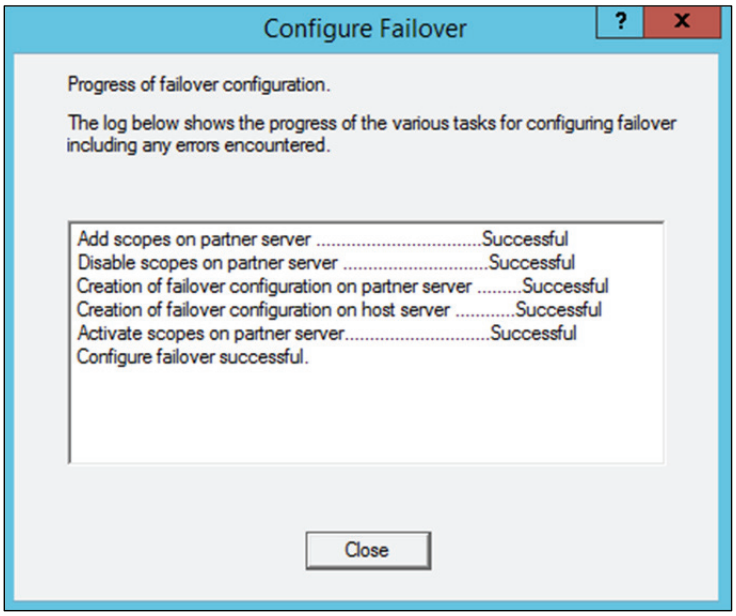


Рис. 2.20. Окно завершения установки
отказоустойчивого DHCP-сервера

Лабораторная работа № 2

***Цель*:** изучение методов установки и первичной настройки операционных систем Windows.

***Задание*:** для выполнения последующих лабораторных работ необходимо установить минимум две операционные системы (в виде виртуальных машин) типа Windows Server 2012 R2, а также две клиентские операционные системы типа Windows 7, Windows 8, Windows 10.

Лабораторная работа № 3–4

***Цель*:** изучение методов организации информационных систем с динамической адресацией на базе операционных систем Windows.

***Задание*:** лабораторная работа представляет собой организацию сети с динамической адресацией между четырьмя операционными системами. В качестве хостов должны выступать виртуальные операционные системы типа Windows со статически заданными сетевыми адресами для серверов, а также с динамически заданными адресами для клиентских машин, согласно варианту (табл. 2.2).

Таблица 2.2

Варианты заданий
для выполнения лабораторной работы № 3–4

Номер варианта	Имя хоста	Scope (диапазон IP-адресов)	IP-адрес хоста
1	Server1_1	10.90.90.3–10.90.90.9	10.90.90.1
	Server1_2		10.90.90.2
	Client1_1	Маска	Любой из scope
	Client1_2	255.255.255.0	Любой из scope
2	Server2_1	10.90.90.12–10.90.90.19	10.90.90.10
	Server2_2		10.90.90.11
	Client2_1	Маска	Любой из scope
	Client2_2	255.255.255.0	Любой из scope
3	Server3_1	10.90.90.22–10.90.90.29	10.90.90.20
	Server3_2		10.90.90.21
	Client3_1	Маска	Любой из scope
	Client3_2	255.255.255.0	Любой из scope

Окончание табл. 2.2

Номер варианта	Имя хоста	Scope (диапазон IP-адресов)	IP-адрес хоста
4	Server4_1	10.90.90.32–10.90.90.39	10.90.90.30
	Server4_2		10.90.90.31
	Client4_1	Маска 255.255.255.0	Любой из scope
	Client4_2		Любой из scope
5	Server5_1	10.90.90.42–10.90.90.49	10.90.90.40
	Server5_2		10.90.90.41
	Client5_1	Маска 255.255.255.0	Любой из scope
	Client5_2		Любой из scope
6	Server6_1	10.90.90.52–10.90.90.59	10.90.90.50
	Server6_2		10.90.90.51
	Client6_1	Маска 255.255.255.0	Любой из scope
	Client6_2		Любой из scope

Схема соединения компьютеров в сети представлена на рис. 2.21 (на примере первого варианта).

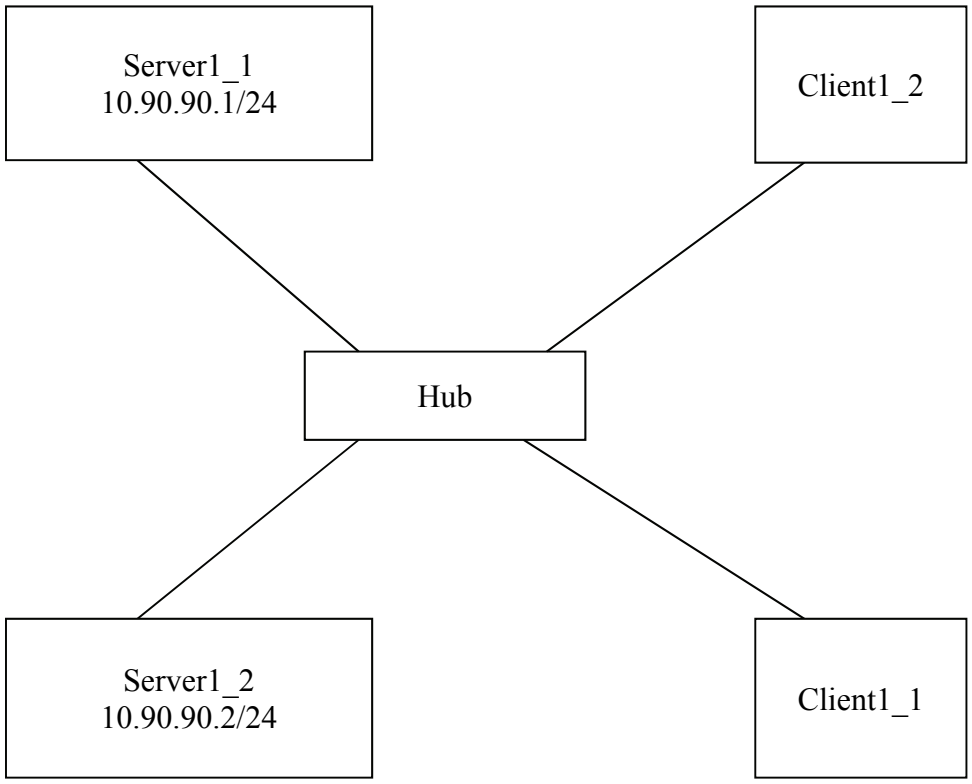


Рис. 2.21. Схема соединения компьютеров в сети

В соответствии с заданием предполагается, что серверы (Server1_1 и Server1_2) должны быть настроены с резервированием друг друга, при этом каждый из них будет обслуживать определенную часть адресного пространства (scope), т. е. Client1_2 и Client1_1 должны получить IP-адреса от разных DHCP-серверов. Таким образом, для *Server1_1* имеем следующее:

Scope: 10.90.90.3–10.90.90.5, маска 255.255.255.0.

Exclusion range: 10.90.90.1–10.90.90.2 (адреса серверов);

10.90.90.6–10.90.90.9 (данный диапазон обслуживается Server1_2, но при выходе его из строя исключение может быть снято).

При этом отметим, что это необходимо осуществлять с «жестким привязыванием» выдаваемого IP-адреса к MAC-адресу клиента (т. е. используя так называемую таблицу соответствия MAC- и IP-адресов – reservations).

Следовательно, для *Server1_2* получим:

Scope: 10.90.90.6–10.90.90.9, маска 255.255.255.0.

Exclusion range: 10.90.90.1–10.90.90.2 (адреса серверов);

10.90.90.3–10.90.90.5 (данный диапазон обслуживается Server1_1, но при выходе его из строя исключение может быть снято).

Подчеркнем, что при использовании Windows Server 2003 и Windows Server 2008 нет каких-либо средств автоматизации резервирования DHCP-серверов, т. е. администратор при возникновении неполадок с одним из серверов принимает решение о снятии исключающего диапазона на втором (рабочем) сервере, чтобы выдавать адреса из всего адресного пространства. При использовании Windows Server 2012 такая возможность уже появилась.

Результаты всей системы в целом можно продемонстрировать, используя утилиты *ping* и *ipconfig*.

Раздел 3

СИМВОЛЬНАЯ АДРЕСАЦИЯ В КОМПЬЮТЕРНЫХ СЕТЯХ

3.1. Символьный адрес DNS

В стеке протоколов TCP/IP, как уже ранее говорилось, используются три типа адресов – физические, IP-адреса и символьные доменные имена. Физические адреса служат для адресации на канальном уровне. IP-адреса применяются на сетевом уровне. Доменные имена кажутся в этом ряду необязательными, ведь сеть будет работать и без них. Однако пользователю сети неудобно запоминать числовые IP-адреса, ассоциируя их с конкретными сетевыми объектами. Все привыкли к символьным именам, и именно поэтому в стек TCP/IP была введена система доменных имен DNS (Domain Name System). Она описывается в RFC 1034 и RFC 1035. Полное название доменных имен – FQDN (Fully Qualified Domain Name – полностью определенное имя домена). Кроме DNS-имен, операционные системы Windows Server поддерживают символьные имена NetBIOS.

DNS – это распределенная база данных, поддерживающая иерархическую систему имен для идентификации узлов в сети Интернет.

Служба DNS предназначена для автоматического поиска IP-адреса по известному символьному имени узла. DNS требует статической конфигурации своих таблиц, разрешающих имена компьютеров в IP-адреса.

В ***протоколе DNS*** определены DNS-серверы и DNS-клиенты.

DNS-серверы IP-адресов – это база данных, которая распределена по административным доменам сети Интернет. *Клиенты сервера DNS* знают IP-адрес сервера DNS своего административного домена и по протоколу IP передают запрос, в котором сообщают известное символьное имя и просят вернуть соответствующий ему IP-адрес. Если данные о запрошенном соответствии хранятся в базе данного DNS-сервера, то он сразу посылает ответ клиенту, если же нет – то он посылает запрос DNS-серверу другого домена, который может сам обработать запрос либо передать его другому DNS-серверу. Все DNS-серверы соединены иерархически в соответствии с иерархией доменов сети Интернет. Клиент опрашивает эти серверы имен, пока не найдет

нужные отображения. Этот процесс ускоряется из-за того, что серверы имен постоянно кешируют информацию, предоставляемую по запросам. Клиентские компьютеры могут использовать в своей работе IP-адреса нескольких DNS-серверов для повышения надежности своей работы.

База данных DNS имеет структуру дерева, называемого ***доменным пространством имен*** поддоменов.

Имя домена идентифицирует его положение в этой базе данных по отношению к родительскому домену, причем точки в имени отделяют части, соответствующие узлам домена.

Корень базы данных DNS управляется центром Internet Network Information Center. Домены верхнего уровня назначаются для каждой страны, а также на организационной основе. Имена этих доменов должны отвечать международному стандарту ISO 3166. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, а для различных типов организаций применяются следующие аббревиатуры:

– com – коммерческие организации (например, microsoft.com);
– edu – образовательные организации (например, mit.edu);
– gov – правительственные организации (например, nsf.gov);
– org – некоммерческие организации (например, fidonet.org);
– net – организации, поддерживающие сети (например, nsf.net).

Каждый домен DNS администрируется отдельной организацией, которая обычно разбивает свой *домен* на *поддомены* и передает функции администрирования этих поддоменов другим организациям. Каждый домен имеет уникальное имя, а каждый из поддоменов имеет уникальное имя внутри своего домена. Имя домена может содержать до 63 символов. Каждый хост в сети Интернет однозначно определяется своим полным доменным именем (Fully Qualified Domain Name, FQDN), которое включает имена всех доменов по направлению от хоста к корню.

В процессе разрешения участвуют DNS-клиент и DNS-сервер. Системный компонент DNS-клиента, называемый DNS-распознавателем, отправляет запросы на DNS-серверы и бывает двух видов:

1) интерактивный – DNS-сервер обращается к DNS-серверу с просьбой разрешить имя без обращения к другим DNS-серверам;

2) рекурсивный – всю работу по разрешению имени выполняет DNS-сервер путем отправки запросов другим DNS-серверам. DNS-сервер всегда сначала ищет имя в собственной базе данных или в кеше и в случае отсутствия обращается к другим серверам.

В основном DNS-клиентами используются рекурсивные запросы. На рис. 3.1 проиллюстрирован процесс разрешения доменного имени с помощью рекурсивного запроса.

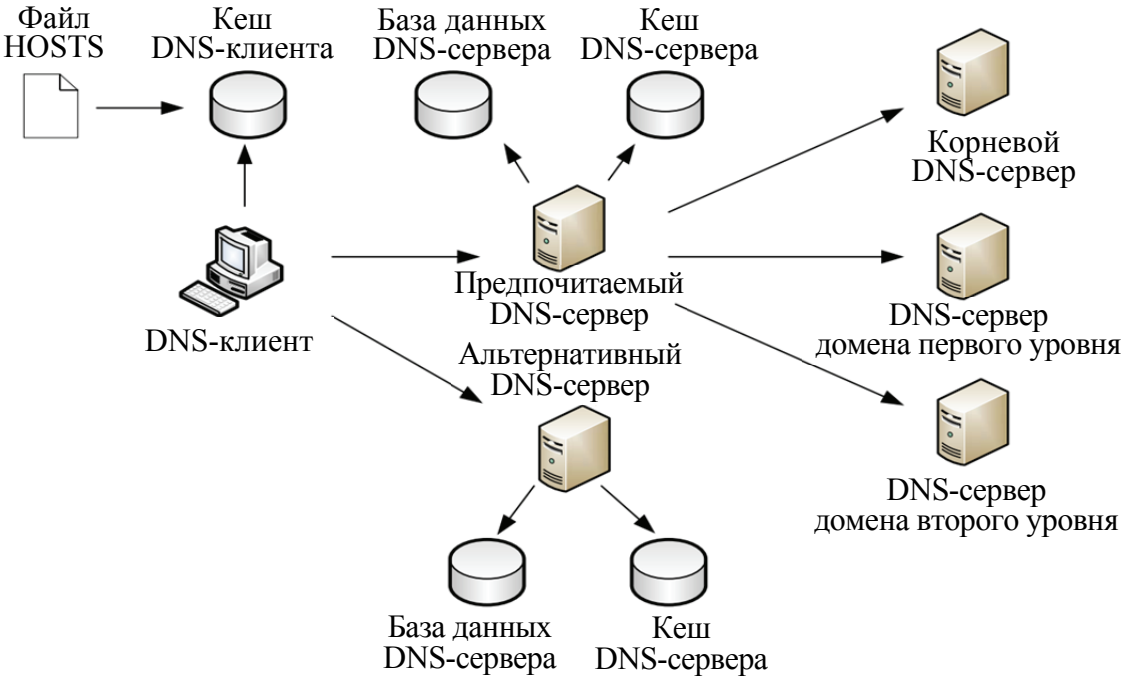


Рис. 3.1. Процесс рекурсивного разрешения имен

Сначала DNS-клиент осуществляет поиск в собственном локальном кеше DNS-имен. Это память для временного хранения ранее разрешенных запросов. В эту же память переносится содержимое файла hosts (каталог windows/system32/drivers/etc). Утилита *ipconfig* с ключом /displaydns отображает содержимое DNS-кеша. Если кеш не содержит требуемой информации, DNS-клиент обращается с рекурсивным запросом к предпочитаемому DNS-серверу (Preferred DNS server), адрес которого указывается при настройке стека TCP/IP. DNS-сервер просматривает собственную базу данных, а также кеш-память, в которой хранятся ответы на предыдущие запросы, отсутствующие в базе данных. В том случае, если запрашиваемое доменное имя не найдено, DNS-сервер осуществляет итеративные запросы к DNS-серверам верхних уровней, начиная с корневого DNS-сервера.

Рассмотрим процесс разрешения доменного имени на примере. Пусть требуется разрешить имя www.microsoft.com. Корневой домен содержит информацию о DNS-сервере, содержащем зону .com. Следующий запрос происходит к этому серверу, на котором хранятся данные обо всех поддоменах зоны .com, в том числе о домене microsoft и его DNS-сервере. Сервер зоны microsoft.com может непосредственно разрешить имя www.microsoft.com в IP-адрес. Обращение к альтернативному серверу осуществляется, только если основной сервер недоступен.

Просмотр DNS-кеша выполняется утилитой *ipconfig/displaydns*, очистка кеша – *ipconfig/flushdns*.

3.2. Символьный адрес NetBIOS

Протокол *NetBIOS* (Network Basic Input/Output System – сетевая базовая система ввода/вывода) был разработан в 1984 г. для корпорации IBM как сетевое дополнение стандартной BIOS на компьютерах IBM PC. В операционных системах Microsoft Windows NT, а также в Windows 98 протокол и имена NetBIOS являлись основными сетевыми компонентами. Начиная с Windows 2000, операционные системы Microsoft ориентируются на глобальную сеть Интернет, в этой связи фундаментом сетевых решений стали протоколы TCP/IP и доменные имена. Однако поддержка имен NetBIOS осталась и в операционной системе Windows Server 2008, а также Windows Server 2012.

В имени NetBIOS отсутствует структура, деление на уровни, как в DNS-именах. Длина имени не более 15 символов (плюс один служебный).

Для преобразования NetBIOS-имен в IP-адреса в операционной системе Windows Server используется служба *WINS* (Windows Internet Naming Service – служба имен в Интернете для Windows).

Служба WINS работает, как и служба DNS, по модели клиент – сервер. WINS-клиенты используют WINS-сервер для регистрации своего NetBIOS-имени и преобразования неизвестного NetBIOS-имени в IP-адрес. Функции сервера NetBIOS-имен описаны в RFC 1001 и RFC 1002.

Процесс разрешения имен в пространстве NetBIOS может быть выполнен одним из трех способов:

– широковещательный запрос;

– обращение к локальной базе данных NetBIOS-имен (LMhosts), хранящихся в папке, где файл hosts отображает FQDN-имена;

– обращение к централизованной базе данных имен NetBIOS, хранящихся на сервере WINS.

В зависимости от типа узла NetBIOS разрешение имен осуществляется различной комбинацией перечисленных способов. Выделяют четыре типа узла:

1) b-узел (broadcast node, широковещательный) – разрешает имена в IP-адресах посредством широковещательных сообщений broadcast node;

2) p-узел (peer node) – разрешает имена в IP-адреса с помощью WINS-сервера;

3) m-узел (mixed node, смешанный) – комбинирует запросы b- и p-узлов, первоначально узел пытается применить широковещательный запрос, а в случае неудачи обращается к WINS-серверу;

4) h-узел (hybrid node, гибридный) – комбинирует запросы b- и p-узлов, но при этом сначала обращается к WINS-серверу, а при неудаче выполняет широковещательную рассылку.

Наиболее эффективным является h-узел. Тип узла определяется следующим образом: если в свойствах протокола TCP/IP нет адреса WINS-сервера, то данный компьютер считается b-узлом, в противном случае является h-узлом. Использование других типов узлов настраивается через реестр Windows.

В больших сетях для распределения нагрузки по регистрации и разрешению NetBIOS-имен необходимо использовать несколько WINS-серверов. Считается, что один WINS-сервер должен обслуживать порядка нескольких сотен компьютеров. При использовании нескольких серверов часть клиентов настраивается на регистрацию и разрешение имен на один WINS-сервер, вторая – на другой, а между серверами, по аналогии с системой DNS, настраивается репликация.

3.3. Настройка DNS-сервера

Рассмотрим организацию DNS-адресации в локальной сети на примере Windows Server 2012 R2. Для организации DNS-адресации необходимо выполнить определенные действия на двух серверах (с именами Server1 и Server2) и клиенте.

Установка DNS-сервера. Установка службы DNS производится в целом аналогично установке DHCP-сервера, описанной ранее в подразделе 2.2, с той лишь разницей, что выбирается установка службы DNS.

Создание основной зоны прямого просмотра. На сервере DC1 создадим стандартную основную зону с именем world.ru, для этого выполним следующие операции.

1. Открываем консоль DNS (рис. 3.2).
2. Выбираем раздел *Forward Lookup Zones* (*Зоны прямого просмотра*) и запускаем *Мастер создания зоны* (тип зоны – *Primary Zone* (*Основная зона*), динамические обновления – *разрешить*, остальные параметры – по умолчанию) (рис. 3.3).
3. Вводим тип зоны и имя. В примере используется название local.by (см. рис. 3.4 и 3.5 на с. 40); имя файла, хранящего информацию о зоне, сформируется автоматически (см. рис. 3.6 на с. 41).
4. Разрешаем передачу данной зоны на любой сервер DNS (консоль DNS → зона local.by → *Properties* (*Свойства*) → закладка *Zone Transfers* (*Передачи зон*) → отметить *Allow zone transfers* (*Разрешить передачи*) и *To any server* (*На любой сервер*)) (см. рис. 3.7 на с. 41).
5. В итоге получаем зону прямого просмотра DNS-сервера, как показано на рис. 3.8 (см. на с. 42).

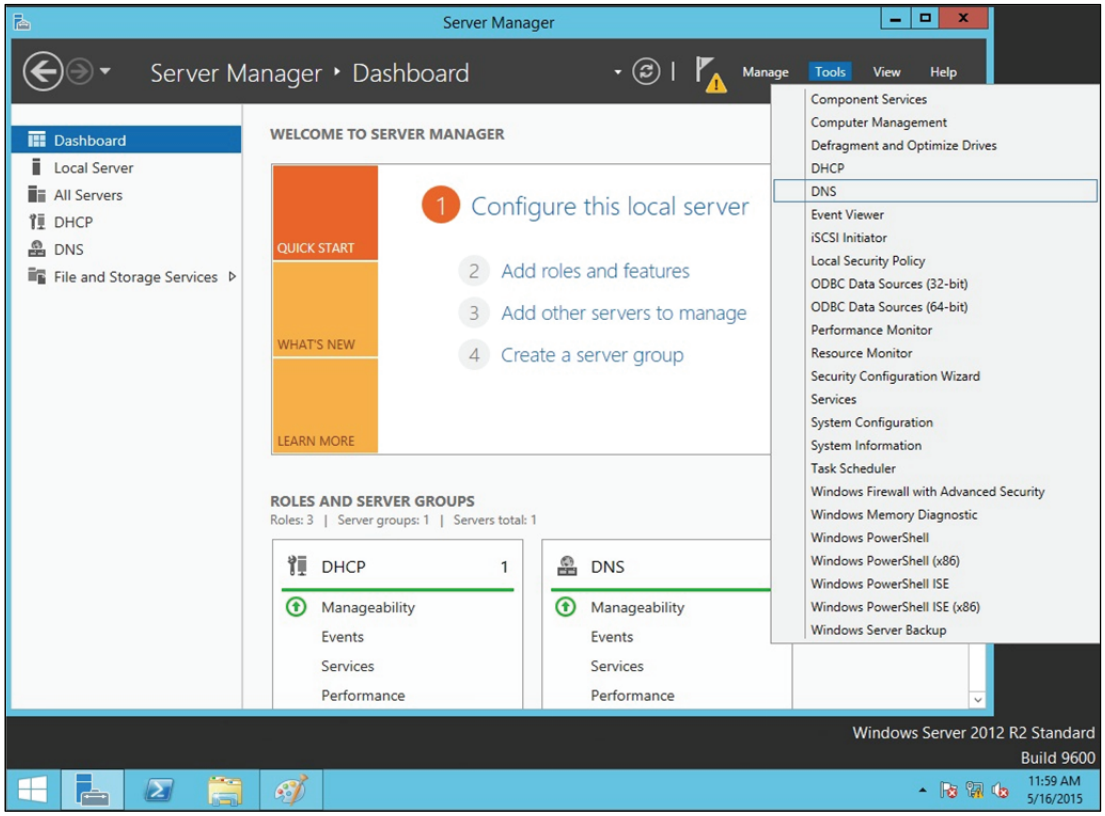


Рис. 3.2. Открытие консоли DNS-сервера

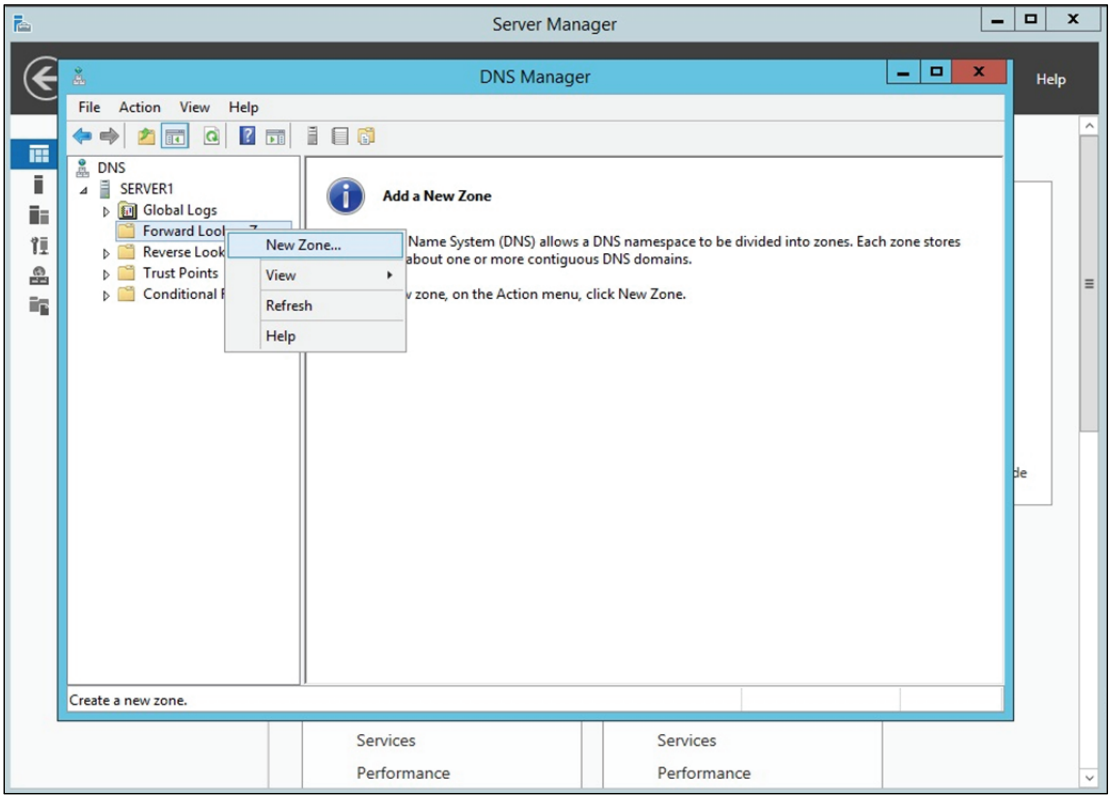


Рис. 3.3. Запуск мастера создания новой зоны DNS-сервера

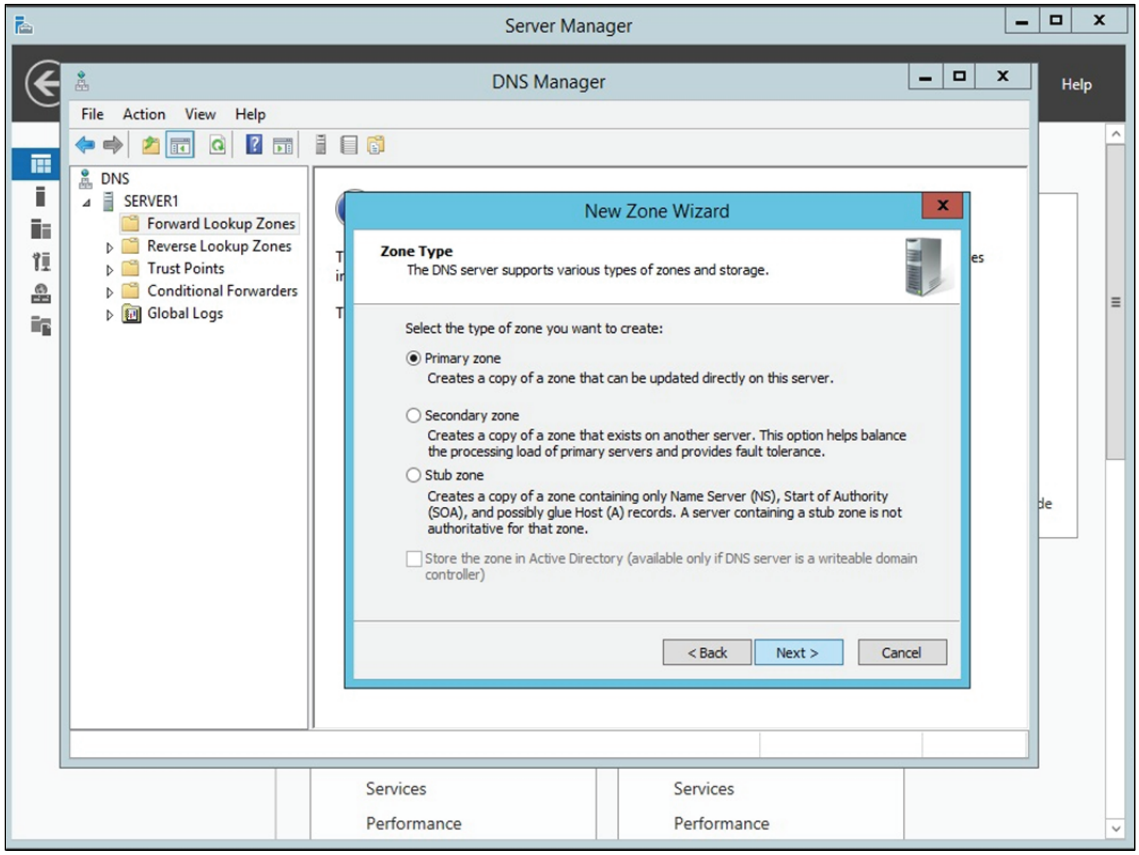


Рис. 3.4. Выбор типа новой зоны DNS-сервера

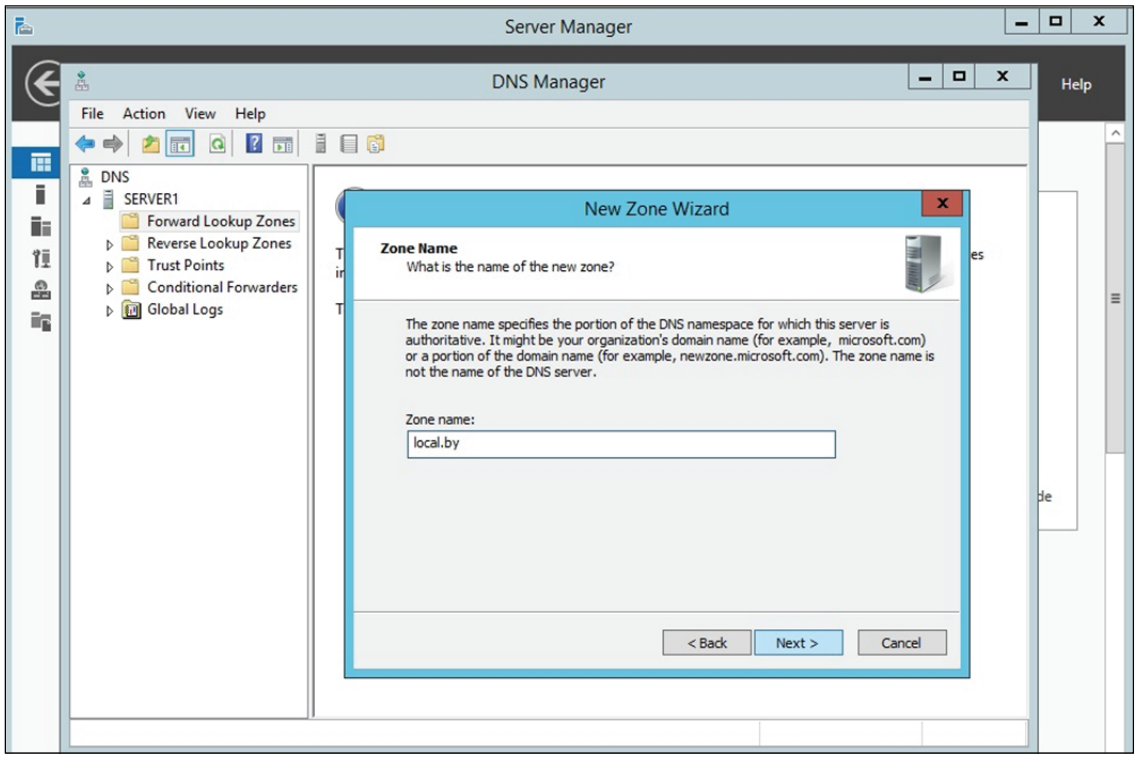


Рис. 3.5. Выбор названия новой зоны DNS-сервера

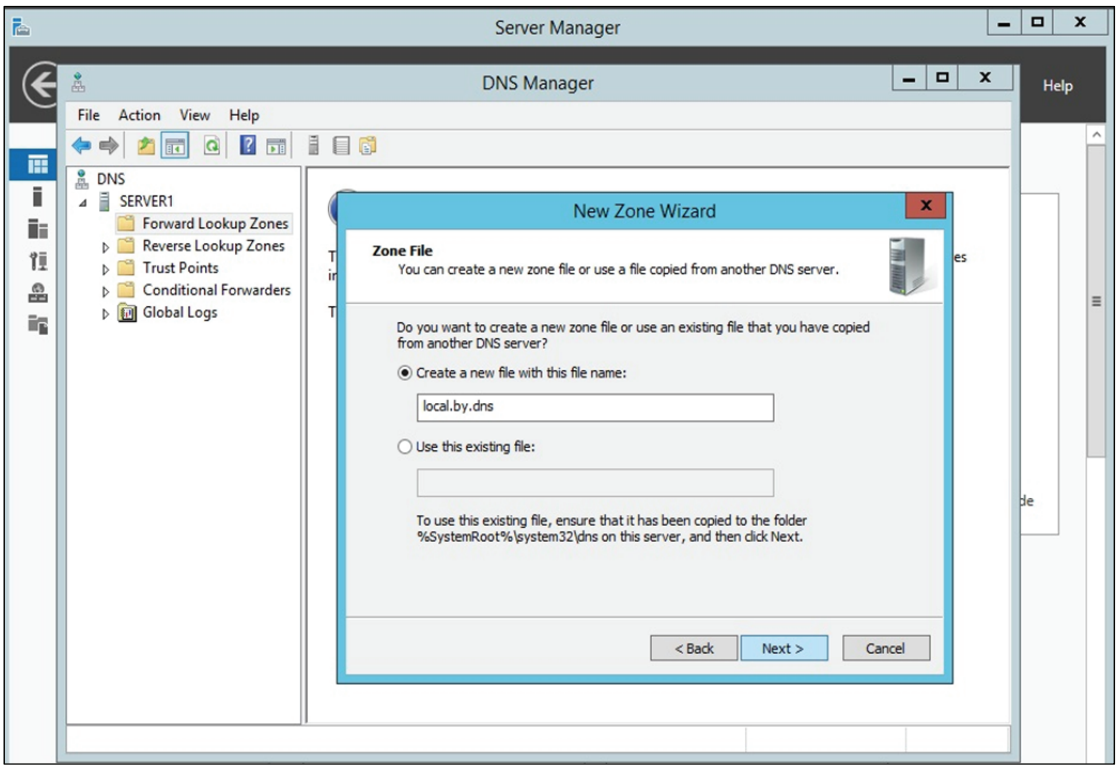


Рис. 3.6. Название файла новой зоны DNS-сервера

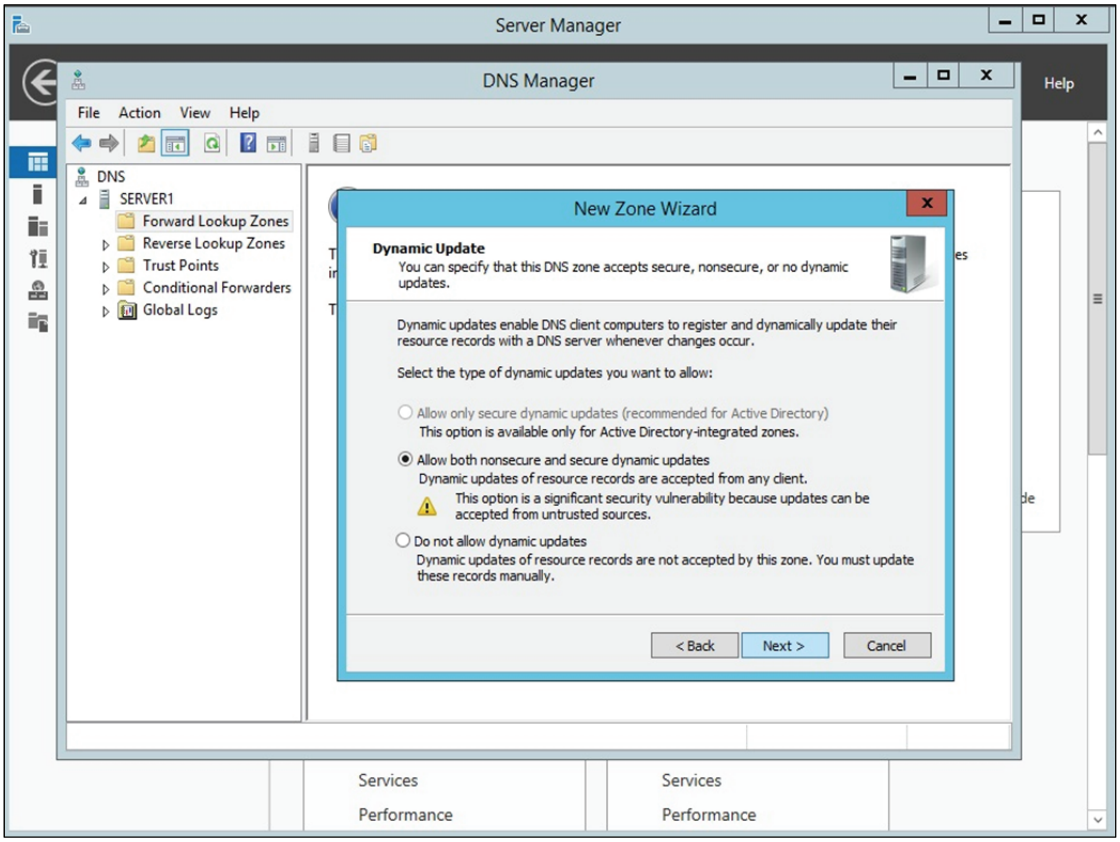


Рис. 3.7. Разрешение на передачу зоны на другой сервер

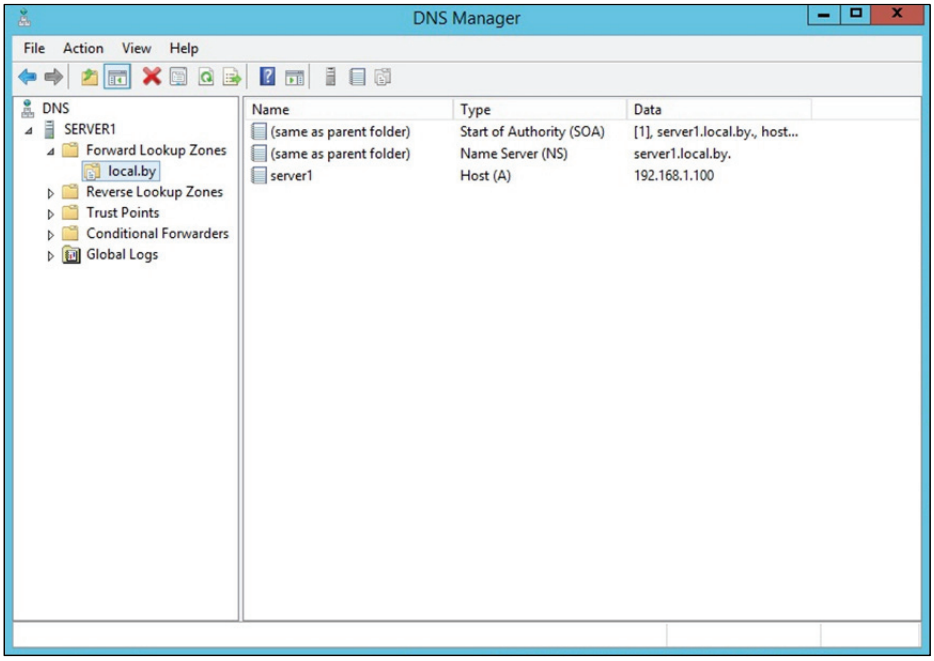


Рис. 3.8. DNS-сервер с созданной зоной прямого просмотра

Чтобы на DNS-сервере автоматически зарегистрировалось имя сервера (в нашем случае Server1), необходимо указать в свойствах компьютера DNS-суффикс (рис. 3.9), а также в IP-конфигурации должен быть приведен адрес DNS-сервера (в нашем случае это все тот же 192.168.1.100).

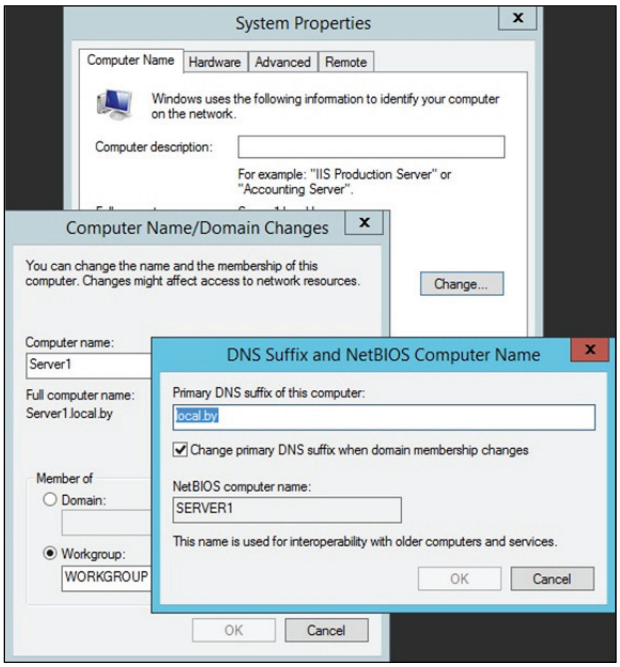


Рис. 3.9. DNS-суффикс
для символьного имени компьютера

Создание дополнительной зоны прямого просмотра. На втором сервере создадим стандартную дополнительную зону с именем local.by (все действия выполняются на втором сервере аналогично установке службы DNS на первом сервере с отличием типа зоны прямого просмотра), для этого выполним следующие операции.

1. Открываем консоль DNS, выбираем раздел *Primary Zone* (*Зоны прямого просмотра*).

2. После чего запускаем *Мастер создания зоны* (тип зоны – *Secondary Zone* (*Дополнительная зона*), IP-адрес master-сервера (с которого будет копироваться зона) – адрес сервера Server1, остальные параметры – по умолчанию) и вводим имя зоны – local.by.

В итоге получаем совместную работу DNS-серверов с реализацией функции резервирования.

Настройка узлов для выполнения динамической регистрации на сервере DNS. Для решения данной задачи нужно выполнить ряд действий как на серверах (если требуемые настройки не были установлены ранее), так и в настройках клиента DNS. Рассмотрим пример настройки клиента с его регистрацией на DNS-сервере.

На сервере DNS должна быть создана соответствующая зона, а также разрешены динамические обновления.

На клиенте DNS необходимо сделать следующее:

– задать в настройках протокола TCP/IP адрес предпочитаемого DNS-сервера – того сервера, на котором разрешены динамические обновления (в нашем примере это сервер с адресом 192.168.1.100);

– в полном имени компьютера указать соответствующий DNS-суффикс (в нашем примере это local.by). Для этого последовательно инициировать: *Мой компьютер* → *Свойства* → закладка *Имя компьютера* → кнопка *Изменить* → кнопка *Дополнительно* → в пустом текстовом поле вписать название домена local → кнопка *OK* (3 раза) (рис. 3.10).

Затем система предложит перезагрузить компьютер. После выполнения перезагрузки на сервере DNS в зоне local.by автоматически создадутся записи типа A для наших серверов (рис. 3.11). В случае несоздания записи для клиента (в нашем примере это client1) можно на стороне клиента в командной строке выполнить команду *ipconfig/registerdns*.

Аналогичные операции необходимо выполнить на всех компьютерах сети.

Если автоматически записи не создались, то их можно создать вручную (см. рис. 3.12 на с. 45), однако при этом могут возникнуть сложности с автоматическим обновлением записей при изменении IP-адресов.

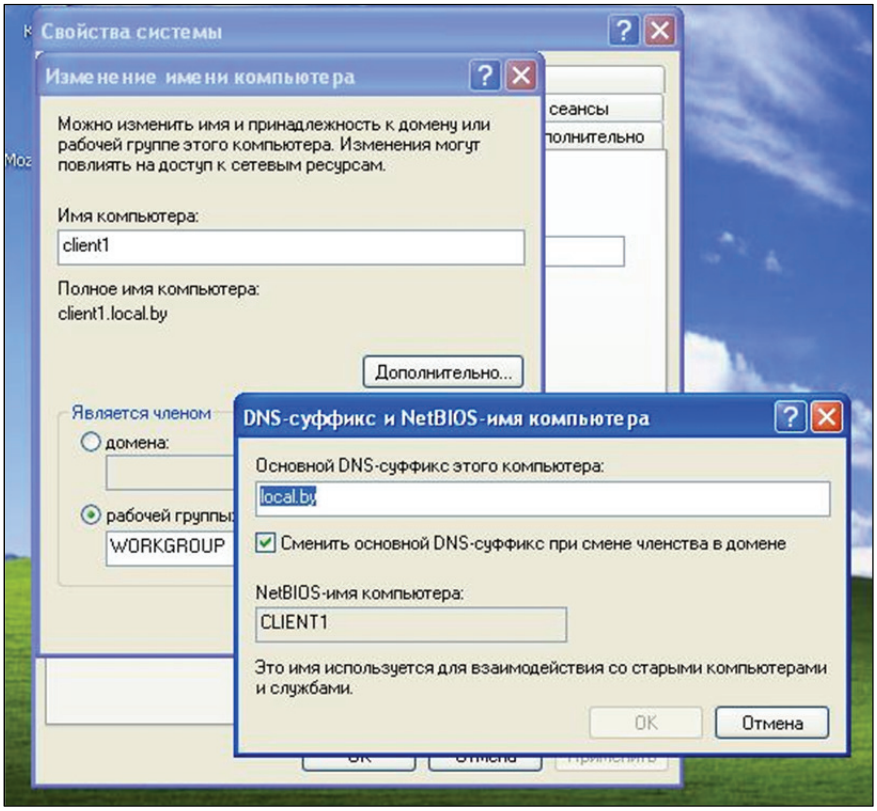


Рис. 3.10. Заполнение поля *DNS-суффикс* на клиенте DNS

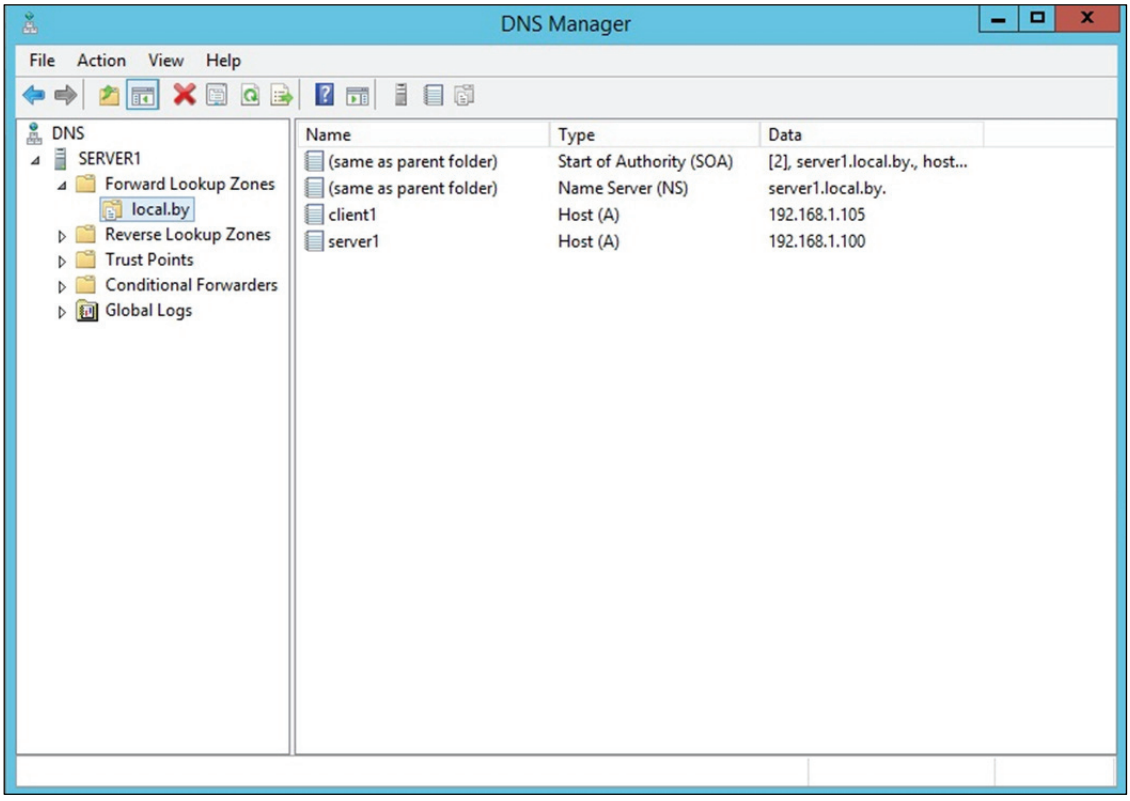


Рис. 3.11. Пример DNS-сервера с записями для клиента и сервера

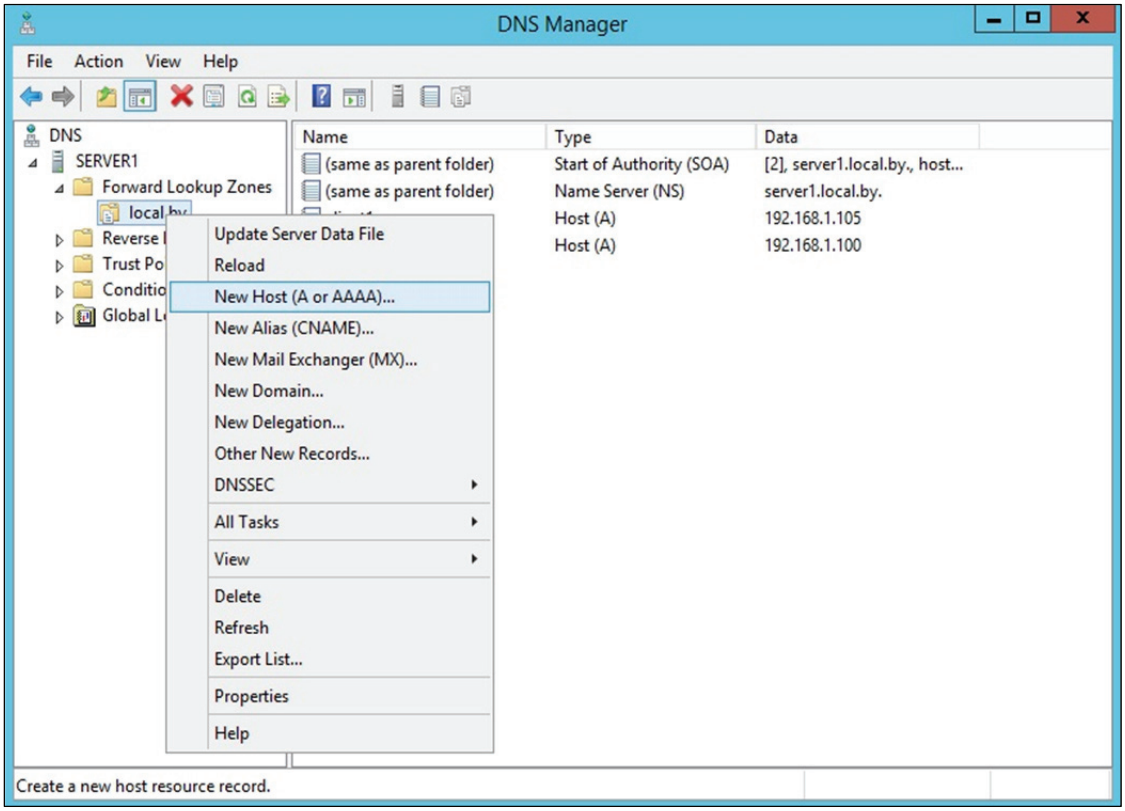


Рис. 3.12. Создание записи типа A на DNS-сервере вручную

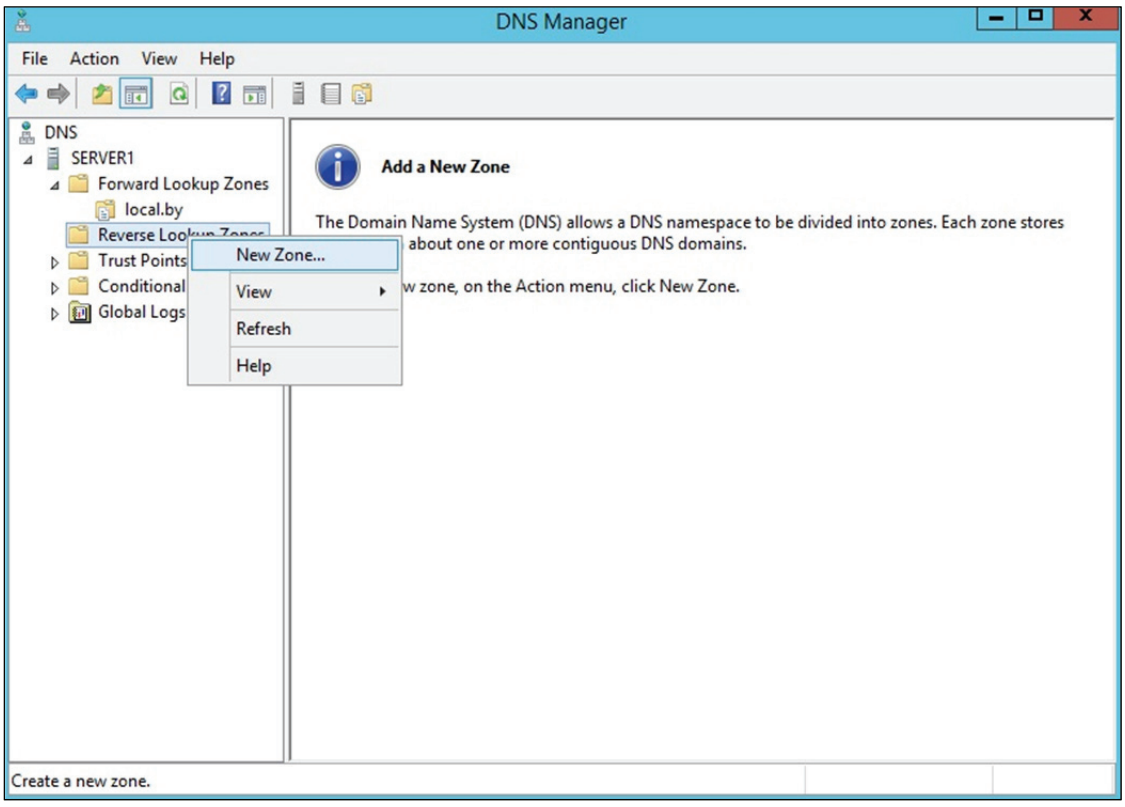


Рис. 3.13. Создание зоны обратного просмотра

Создание зоны обратного просмотра. Чтобы создать зону обратного просмотра, выполним следующие шаги.

1. Открываем консоль DNS, выбираем раздел *Reverse Lookup Zone* (*Зоны обратного просмотра*).

2. После этого запускаем *Мастер создания зоны* (тип зоны – *Primary Zone* (*Основная зона*), динамические обновления – *разрешить*, остальные параметры – по умолчанию) (см. рис. 3.13 на с. 45).

3. В поле *Код сети* (*ID*) вводим параметры идентификатора сети – 192.168.1.x, а затем выполняем команду принудительной регистрации компьютеров на сервере DNS – *ipconfig*/*registerdns*.

В итоге компьютеры зарегистрируются в обратной зоне DNS.

Лабораторная работа № 5

***Цель*:** изучение методов организации символьной адресации в информационных системах на базе клиент-серверной сети и операционных систем Windows с использованием DNS-сервера.

***Задание*:** лабораторная работа представляет собой настройку DNS-сервера в сети с клиент-серверной архитектурой, организованной при выполнении лабораторных работ № 2–4, и регистрацию DNS-клиентов. В качестве хостов должны выступать виртуальные операционные системы типа Windows с организованной динамической адресацией. DNS-сервер должен использовать статический адрес (согласно лабораторной работе № 3–4). Имена доменов следует выбрать по согласованию с преподавателем. Проверить работу DNS-сервера можно с помощью утилиты *ping*, как показано в подразделе 3.3. Также необходимо отработать использование утилиты *ipconfig* с соответствующими командами, позволяющими просмотреть DNS-кеш (displaydns) и очистить DNS-кеш (flushdns). Данные операции зачастую необходимы при изменении IP-адресов DNS-имен.

Раздел 4

ДОМЕННЫЕ СИСТЕМЫ (СЛУЖБА ACTIVE DIRECTORY)

4.1. Понятие Active Directory. Служба Active Directory

Ранее отмечалось, что в средних и крупных сетях задача настройки параметров протокола TCP/IP является очень сложной для администратора и вручную практически не выполнима. Для решения этой проблемы был разработан протокол DHCP, реализованный посредством службы DHCP.

Однако настройка сетевых параметров – лишь одна из множества задач, встающих перед системным администратором. В частности, в любой сети важнейшей является задача управления ее ресурсами (файлами и устройствами, предоставленными в общий доступ), а также компьютерами и пользователями.

Для решения задач управления ресурсами в сетях под управлением Windows Server применяется служба каталога Active Directory (активный каталог). Данная служба обеспечивает доступ к базе данных (*каталогу*), в которой хранится информация обо всех объектах сети, и позволяет управлять этими объектами.

Группа компьютеров, имеющая общий каталог и единую политику безопасности, называется ***доменом*** (domain). Под ***политикой безопасности*** понимают набор правил по применению средств обеспечения сетевой безопасности: паролей, учетных записей, протоколов аутентификации и защищенной передачи информации, шифрованной файловой системы и т. д.

Каждый домен имеет один или несколько серверов, именуемых ***контроллерами домена*** (domain controller), на которых хранятся копии каталога.

Основными преимуществами, предоставляемыми службой каталога Active Directory, являются:

1) централизованное управление – если в сети развернута служба Active Directory, системный администратор может выполнять большинство своих задач, используя единственный компьютер – *контроллер домена*;

2) простой доступ пользователей к ресурсам – пользователь, зарегистрировавшись в домене на произвольном компьютере, может

получить доступ к любому ресурсу сети при условии наличия соответствующих прав;

3) обеспечение безопасности – служба Active Directory совместно с подсистемой безопасности Windows Server предоставляет возможность гибкой настройки прав пользователей на доступ к ресурсам сети;

4) масштабируемость – это способность системы повышать свои размеры и производительность по мере увеличения требований к ним. При расширении сети организации служба каталога Active Directory способна наращивать свои возможности – увеличивать размер каталога и число контроллеров домена.

Таким образом, служба каталога Active Directory, подобно службе DHCP, существенно облегчает работу системного администратора по управлению сетевыми объектами. Кроме того, пользователи получают возможность использовать ресурсы сети, не заботясь об их месторасположении, так как все запросы обрабатываются службой Active Directory.

4.2. Объекты каталога и их именование

Объект каталога Active Directory – это элемент, содержащийся в базе данных Active Directory и имеющий набор атрибутов (характеристик). Например, объектом является пользователь, а его атрибутами – имя, фамилия и адрес электронной почты.

Некоторые объекты являются контейнерами. Это означает, что данные объекты могут содержать в своем составе другие объекты. Например, объект *домен* является контейнером и может включать пользователей, компьютеры, другие домены и т. д.

Каталог Active Directory содержит следующие основные типы объектов, не являющихся контейнерами:

- пользователь (user);
- группы пользователей (group);
- контакты (contact);
- компьютеры (computer);
- принтеры (printer);
- общедоступные папки (shared folder).

В Active Directory для именования объектов используется несколько способов.

Различающееся имя (Distinguished Name, DN) состоит из нескольких частей. Например, для пользователя Петрова, принадлежащего к организационному подразделению Teachers домена faculty.ru, различающееся имя выглядит так:

DC = ru, DC = faculty, OU = teachers, CN = users, CN = petrov,

где DC (Domain Component) – домен; OU (Organizational Unit) – организационное подразделение; CN (Common Name) – общее имя.

Различающиеся имена являются уникальными в пределах всего каталога Active Directory. В целях упрощения именования может использоваться *относительное различающееся имя* (Relative Distinguished Name, RDN). Для приведенного примера это имя CN = petrov. Имя RDN должно быть уникально в рамках объекта-контейнера, т. е. в пределах контейнера CN = users пользователь petrov должен быть единственным.

Основное имя пользователя (User Principal Name, UPN) используется для входа пользователя в систему и состоит из двух частей: имени учетной записи пользователя и имени домена, к которому принадлежит пользователь. Например: petrov@faculty.ru.

Глобальный уникальный идентификатор (Global Unique Identifier, GUID) – это 128-битовое шестнадцатеричное число, которое ассоциируется с объектом в момент его создания и никогда не меняется. В случае перемещения или переименования объекта его GUID остается прежним.

4.3. Иерархия доменов

Домен является основным элементом в логической структуре Active Directory. В рамках домена действуют единые административные полномочия и политика безопасности, применяется общее пространство доменных имен.

Каждый домен имеет, по крайней мере, один контроллер домена, на котором хранится каталог Active Directory с информацией о домене.

Для организаций со сложной структурой может создаваться иерархия доменов. Первый образованный домен называется ***корневым*** (root domain). У него могут быть дочерние домены, имеющие общее пространство доменных имен. В свою очередь, у дочерних доменов могут быть свои домены-потомки. Таким образом, создается иерархия доменов, называемая ***доменным деревом*** (domain tree).

Если требуется в рамках одной организации организовать еще одно пространство имен, то создается отдельное дерево доменов. При этом несколько деревьев, входящих в состав одного каталога Active Directory, образуют ***лес доменов*** (forest).

Для именования доменов используются правила, принятые в системе доменных имен DNS. Вследствие этого доменная структура организации может при необходимости (и соблюдении требования

уникальности имен) встраиваться в доменную структуру Интернета. Кроме того, для разрешения доменных имен становится возможным использование службы DNS.

На рис. 4.1 приведен фрагмент возможной доменной структуры университета.

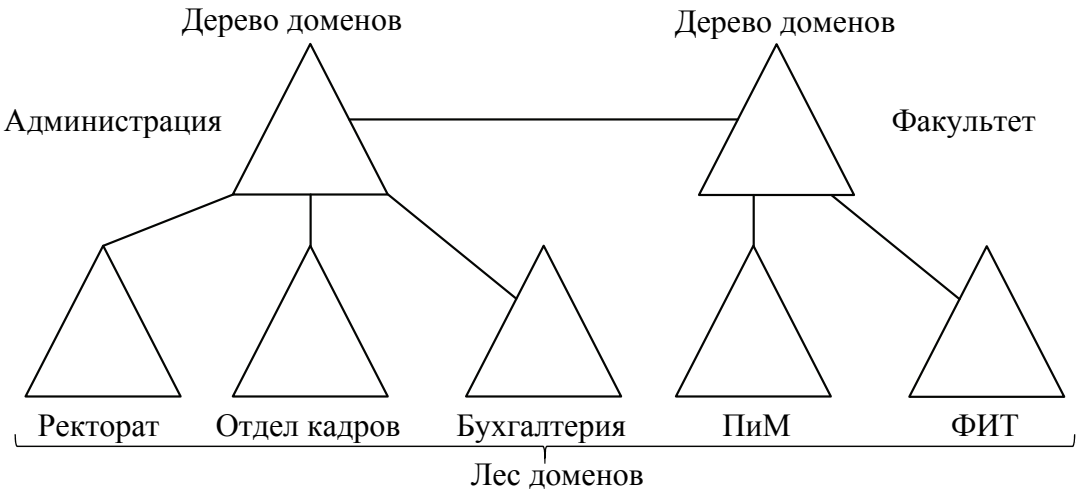


Рис. 4.1. Фрагмент возможной доменной структуры университета

В данном примере лес состоит из двух деревьев – дерева управления университета (домен администрация) и дерева факультетов (домен факультет). Корневой домен головной организации имеет три дочерних домена – ректорат, отдел кадров, бухгалтерия. Корневой домен факультет является родителем для двух доменов – ПиМ и ФИТ.

Следуя правилам DNS, полное имя (FQDN) домена ректорат будет иметь следующий вид: ректорат.администрация, а полное имя домена ПиМ – ПиМ.факультет.

4.4. Организационные подразделения

Структурирование сетевых ресурсов организации при помощи доменов не всегда бывает оправданно, так как домен подразумевает достаточно крупную часть сети. Часто для администратора возникает необходимость группировки объектов внутри одного домена. В этом случае следует использовать ***организационные подразделения*** (organizational unit).

Организационные подразделения (ОП) можно использовать в качестве контейнера для следующих объектов:

1) пользователей;
2) групп пользователей;

3) контактов;
4) компьютеров;
5) принтеров;
6) общих папок;
7) других организационных подразделений.

Объекты группируются с помощью ОП для следующих целей:

– управление несколькими объектами как одним целым – для этого используются групповые политики;

– делегирование прав администрирования – например, начальнику отдела можно делегировать административные права на его отдел при условии объединения всех объектов отдела в организационную единицу.

В качестве примера структуризации с использованием ОП можно привести возможную структуру домена ФИТ (рис. 4.2).

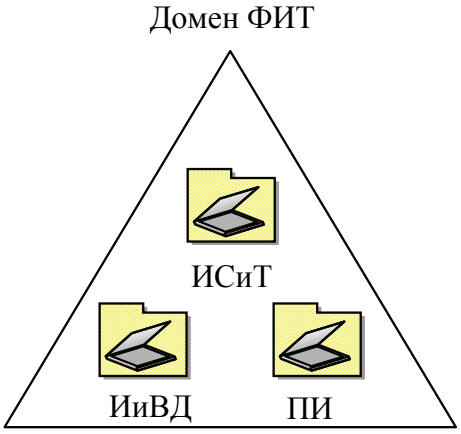


Рис. 4.2. Структура домена ФИТ

В данной ситуации выделение из домена ФИТ дочерних доменов кафедр (ИСиТ, ИиВД и ПИ) не имеет смысла, так как факультет слишком мал. В то же время требуется отразить в Active Directory внутреннюю структуру факультета. Решением является структуризация с применением организационных подразделений – в домене создаются организационные подразделения кафедр: ИСиТ, ИиВД и ПИ. При этом для каждого подразделения администратор может назначить собственный набор правил (например, общие требования к паролям).

4.5. Учетные записи пользователей

После реализации спроектированной структуры Active Directory администратор должен добавить в каталог учетные записи всех пользователей системы и назначить каждой из них определенные права.

Учетная запись пользователя – это набор атрибутов, сопоставленных с определенным пользователем. Самые важные атрибуты следующие:

1) имя учетной записи, с помощью которого пользователь осуществляет вход в систему (в пределах домена должно быть уникально);

2) полное имя пользователя;

3) пароль;

4) группы, в которые входит пользователь;

5) права пользователя.

Создав все необходимые учетные записи, администратору следует продумать, какими правами должен обладать тот или иной пользователь. ***Права пользователя*** – это список действий, которые может выполнять пользователь. Права бывают следующих видов:

– *привилегия* (privilege) – право выполнения операций по изменению состояния или параметров системы (например, выключение компьютера или изменение системного времени);

– *право на вход в систему* (logon right);

– *разрешение доступа* (access permission) – право осуществления действий с файлами, папками, принтерами, объектами Active Directory, реестром (при условии, что используется файловая система NTFS).

Если пользователей порядка десяти человек, определить необходимые права можно достаточно просто. Однако гораздо чаще на практике встречаются компьютерные системы с сотнями и тысячами учетных записей. В таких масштабах задача распределения прав отдельным пользователям становится невыполнимой. В этом случае на помощь администратору приходит механизм групп пользователей.

4.6. Группы пользователей

Группа пользователей (Security Group – группа безопасности) – это объединение учетных записей пользователей, которому можно назначать права. С использованием групп распределение прав осуществляется следующим образом. Сначала выбираются такие пользователи, список прав которых должен быть одинаковым. Затем создается группа, членами которой являются выбранные пользователи. Требуемые права назначаются уже не отдельным пользователям, а группе, и эти права автоматически распространяются на всех пользователей группы.

Следует отметить, что группы пользователей и организационные подразделения представляют собой разные механизмы, предназначенные для разных целей. Создание групп безопасности преследует цель

распределения прав доступа пользователей к ресурсам сети, в то время как основное назначение организационных подразделений – управление пользователями (а также компьютерами) (рис. 4.3).

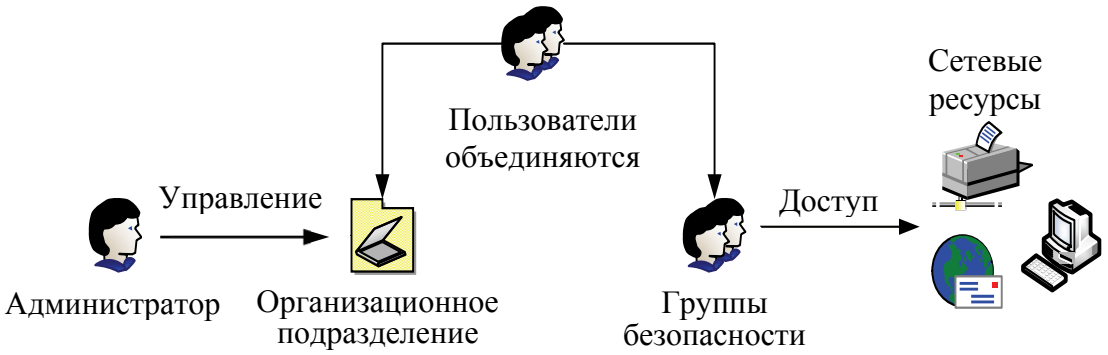


Рис. 4.3. Использование организационных подразделений и групп безопасности

Группы пользователей различаются по области действия. Выделяют три области действия:

1) доменную локальную (domain local scope);
2) глобальную (global scope);
3) универсальную (universal scope).

Доменные локальные группы действуют в рамках только своего домена. За его пределами указывать локальную доменную группу нельзя. Такие группы обычно применяются для управления доступом к файлам, общим папкам и принтерам.

Глобальные группы могут использоваться в рамках всего леса доменов. Однако глобальная группа принадлежит определенному домену, и в ее состав могут входить только объекты этого домена. Применяются глобальные группы в том случае, если пользователям одного домена нужно получить доступ к ресурсам другого домена.

Универсальные группы привязаны к корневому домену леса, но в их состав могут входить пользователи любого домена. Чаще всего универсальные группы используются для объединения глобальных групп.

4.7. Создание доменов. Создание и настройка пользователей. Распределение ресурсов

4.7.1. Создание домена. Установка роли Active Directory

После настройки сетевой и символьной адресации можно приступить к установке и настройке домена. Для этого выполним следующие операции.

1. Щелкаем *Start* → *Server Manager* (*Пуск* → *Диспетчер сервера*) и выбираем *Add Roles and Features Wizard* (*Добавить роль сервера*). Затем нажимаем *Next* (*Далее*).

2. Выбираем *Role-based or feature-based installation* (*Установка ролей и компонентов*) и щелкаем *Next* (*Далее*) (рис. 4.4).

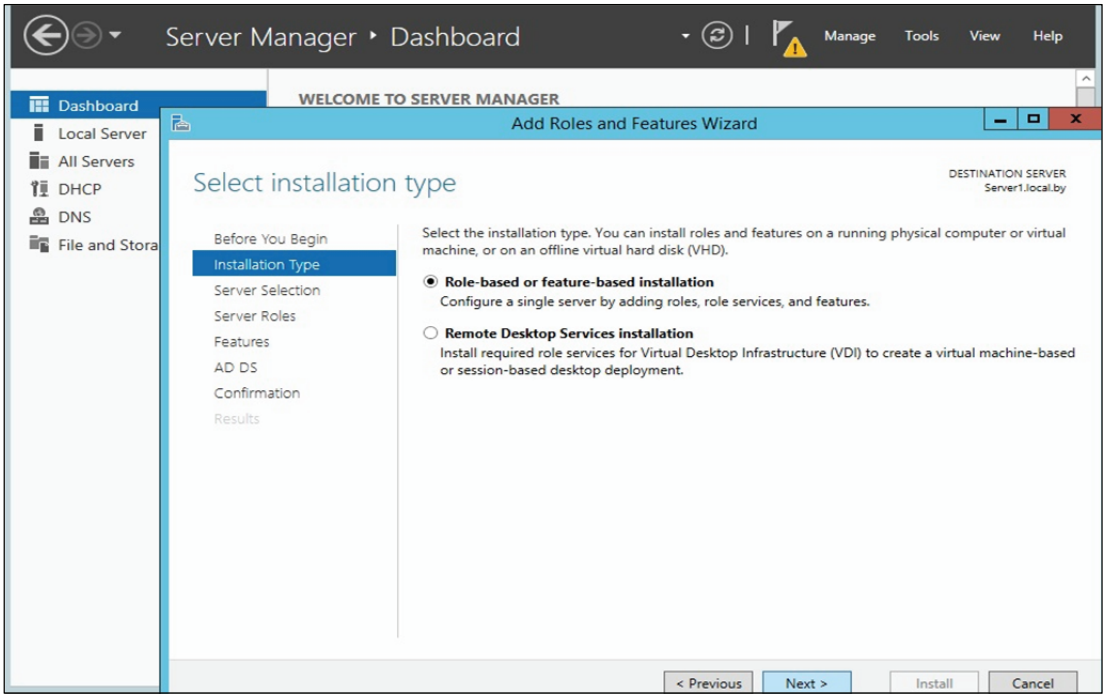


Рис. 4.4. Выбор опции установки ролей и компонент

3. Теперь определяем сервер, на который устанавливается роль AD, и кликаем *Next* (*Select a server from the server pool* → *Next*) (рис. 4.5).

4. Выбираем роль *Active Directory Domain Services* (*Доменные службы Active Directory*), после чего появляется окно с предложением добавить роли и компоненты, необходимые для установки роли AD. Нажимаем кнопку *Add Features* (*Добавление компонентов*) (рис. 4.6).

5. После этого кликаем каждый раз кнопку *Next* (*Далее*) и устанавливаем роль.

6. После задания роли закрываем окно, нажав *Close* (*Закрыть*). Затем переходим к настройке роли AD. В окне *Server Manager* (*Диспетчер сервера*) нажимаем пиктограмму флага с уведомлением и выбираем *Promote this server to a domain controller* (*Повысить роль этого сервера до уровня контроллера домена*) на плашке *Post-deployment Configuration* (см. рис. 4.7 на с. 56).

7. На следующем этапе выбираем *Add a new forest* (*Добавить новый лес*) и вводим название домена, затем щелкаем *Next* (*Далее*) (см. рис. 4.8 на с. 56).

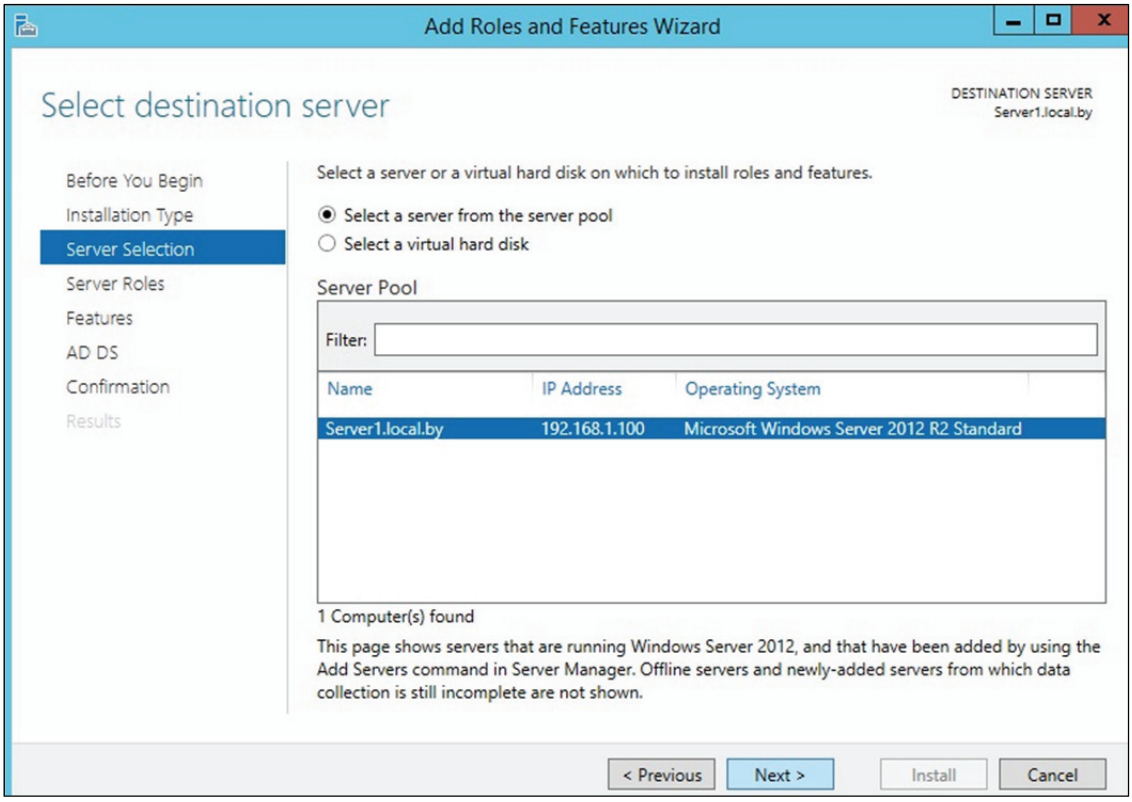


Рис. 4.5. Выбор сервера для установки AD

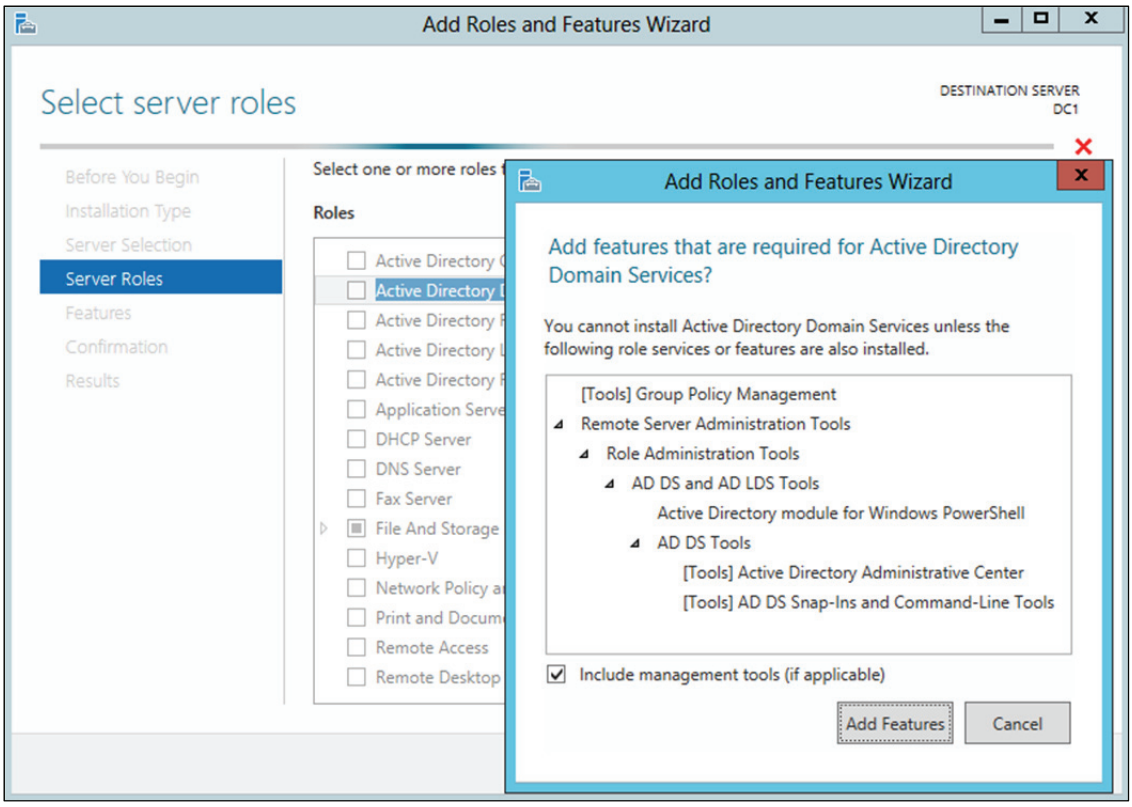


Рис. 4.6. Выбор роли и компонент AD

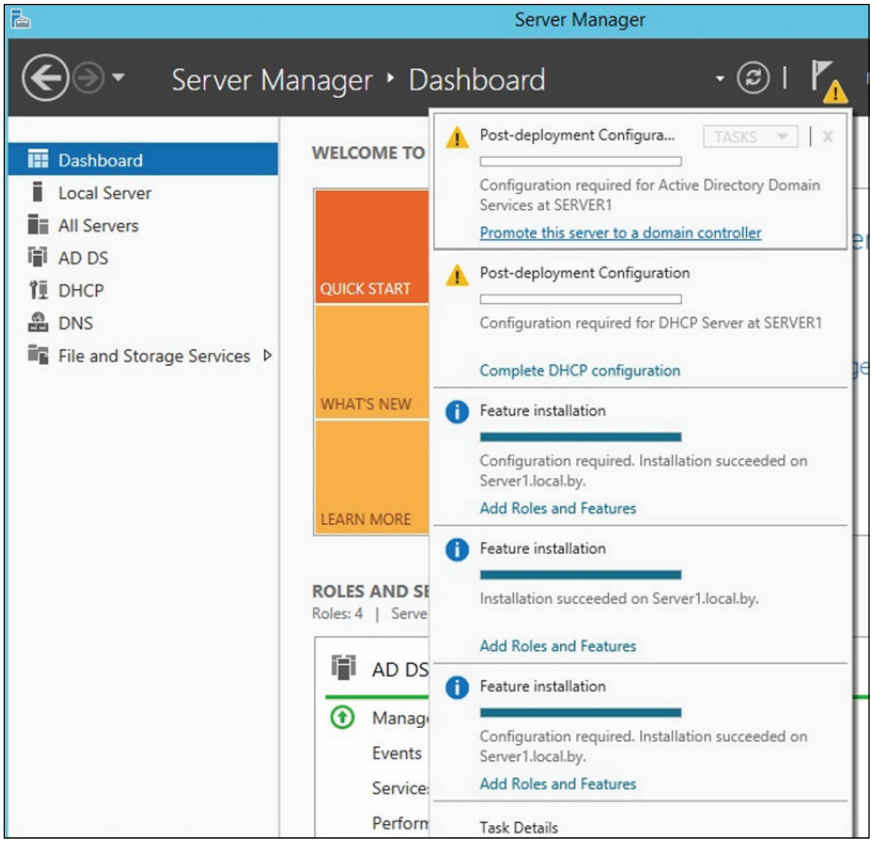


Рис. 4.7. Повышение роли сервера до уровня контроллера домена

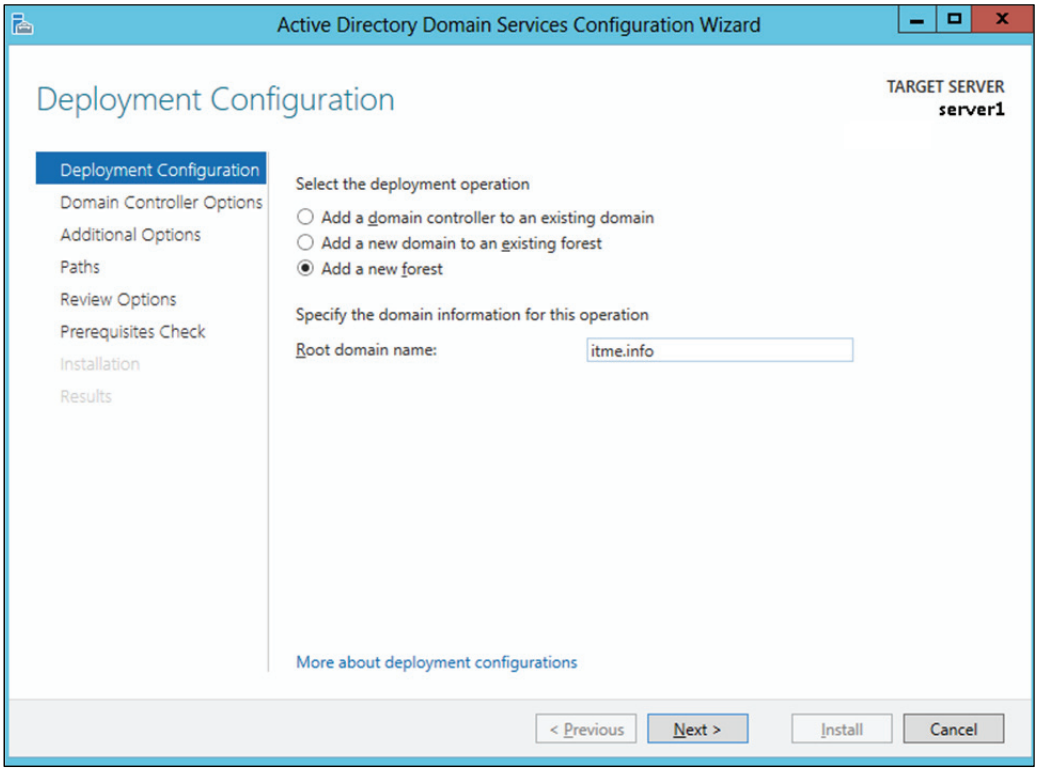


Рис. 4.8. Выбор структуры нового домена

8. На следующей вкладке задаем совместимость режима работы леса и корневого домена. По умолчанию устанавливается Windows Server 2012 R2 (рис. 4.9). Так же можно будет отключить *роль DNS Server*, но в нашем случае это нецелесообразно, а поэтому галочку оставляем и вводим пароль для DSRM (Directory Services Restore Mode – режим восстановления службы каталога). Когда все сделано, кликаем *Next* (*Далее*).

9. Затем назначаем делегирование полномочий для DNS-сервера (должно быть выбрано по умолчанию) (рис. 4.10).

10. Теперь можно изменить NetBIOS-имя, которое было присвоено домену (см. рис. 4.11 на с. 59). Для этого следим за тем, чтобы имя было в соответствии с планом, а именно *Local*, и нажимаем *Next* (*Далее*).

11. На следующем шаге можно изменить пути к каталогам базы данных AD DS (Active Directory Domain Services – доменная служба Active Directory), файлам журнала, а также папке Sysvol (см. рис. 4.12 на с. 59). Отметим, что менять что-либо нецелесообразно, а поэтому щелкаем *Next* (*Далее*).

12. На данном этапе отображается сводная информация по настройке (см. рис. 4.13 на с. 60). Нажав кнопку *View script*, можно посмотреть PowerShell скрипт, который произведет настройку доменных служб Active Directory. Убедившись, что все указано верно, нажимаем *Next* (*Далее*).

13. Теперь проверяем, все ли предварительные требования соблюдены (см. рис. 4.14 на с. 60). После чего появится отчет. Одно из обязательных требований – это установленный пароль локального администратора. В самом низу можно прочитать предупреждение о том, что после того, как будет нажата кнопка *Install* (*Установить*), уровень сервера будет повышен до контроллера домена и будет произведена автоматическая перезагрузка. После перезагрузки должна появиться надпись *All prerequisite checks passed successfully. Click “install” to begin installation*.

14. После завершения всех настроек сервер перезагрузится, и мы совершим первый ввод компьютера в домен. Для этого необходимо ввести логин и пароль администратора домена.

На этом базовая настройка служб каталога Active Directory завершена, поэтому нажимаем *Install* (*Установить*). Конечно же, еще предстоит проделать огромный объем работы по созданию подразделений, новых пользователей, по настройке групповых политик безопасности.

После завершения операций по установке контроллера домена следует выполнить повторную авторизацию DHCP-сервера. Для этого в окне *Server Manager* (*Диспетчер сервера*) нажимаем пиктограмму флага с уведомлением и выбираем *Complete DHCP configuration* (*Завершить настройку DHCP*) на плашке *Post-deployment Configuration* (см. рис. 4.15 на с. 61).

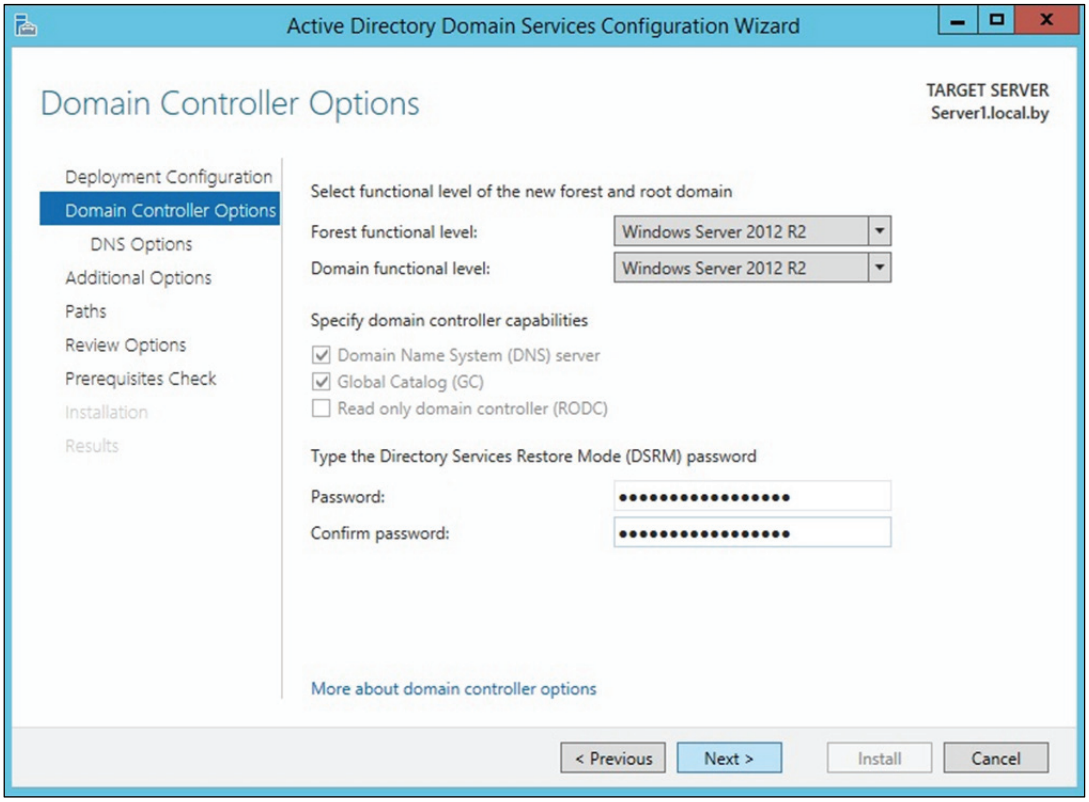


Рис. 4.9. Страница настроек контроллера нового домена

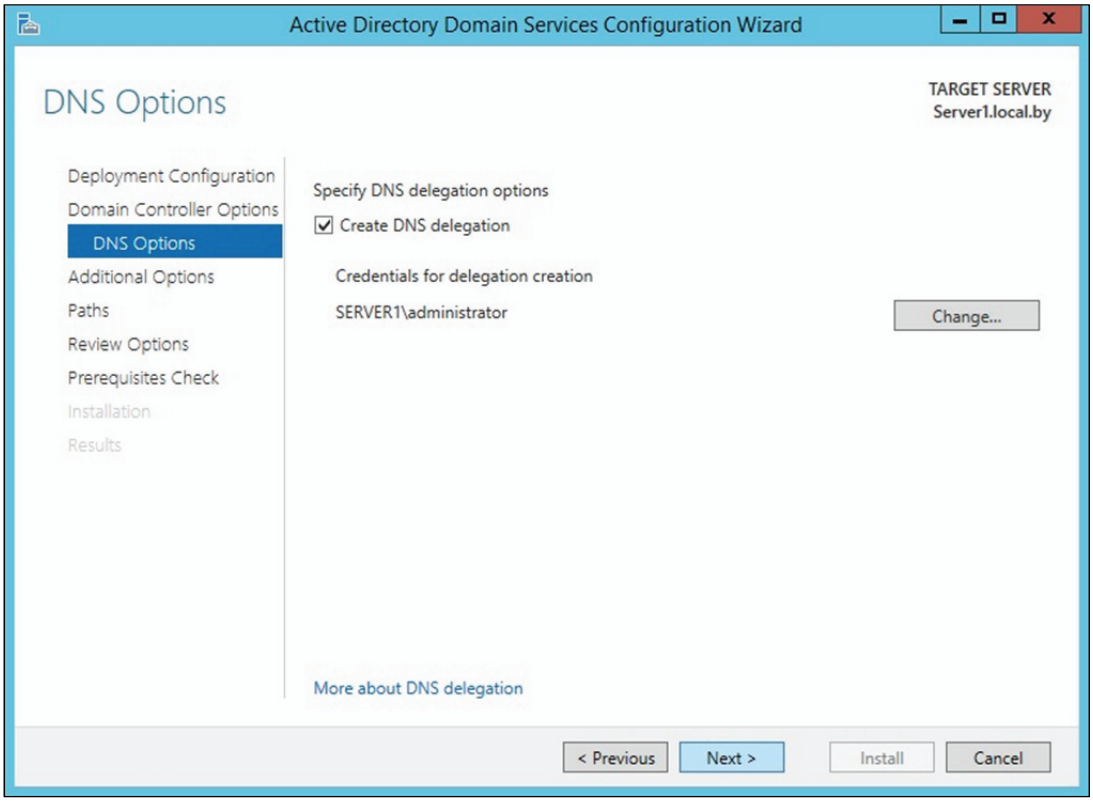


Рис. 4.10. Делегирование полномочий DNS-серверу

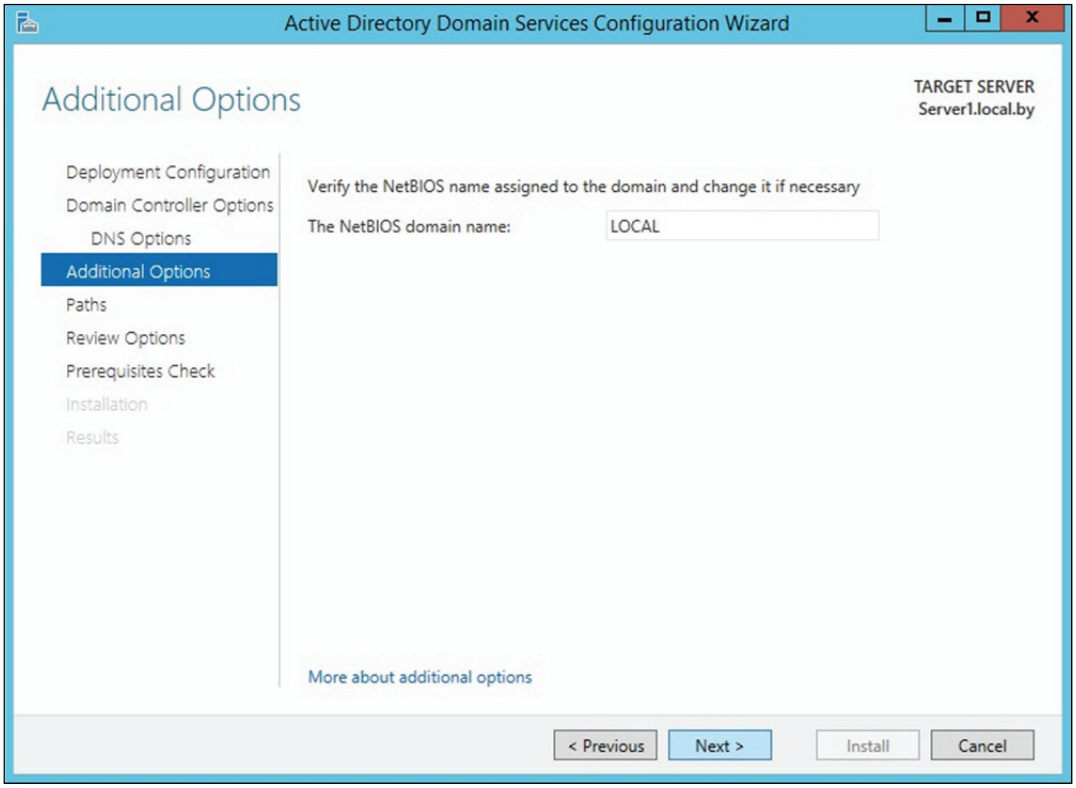


Рис. 4.11. Выбор NetBIOS-имени контроллера домена

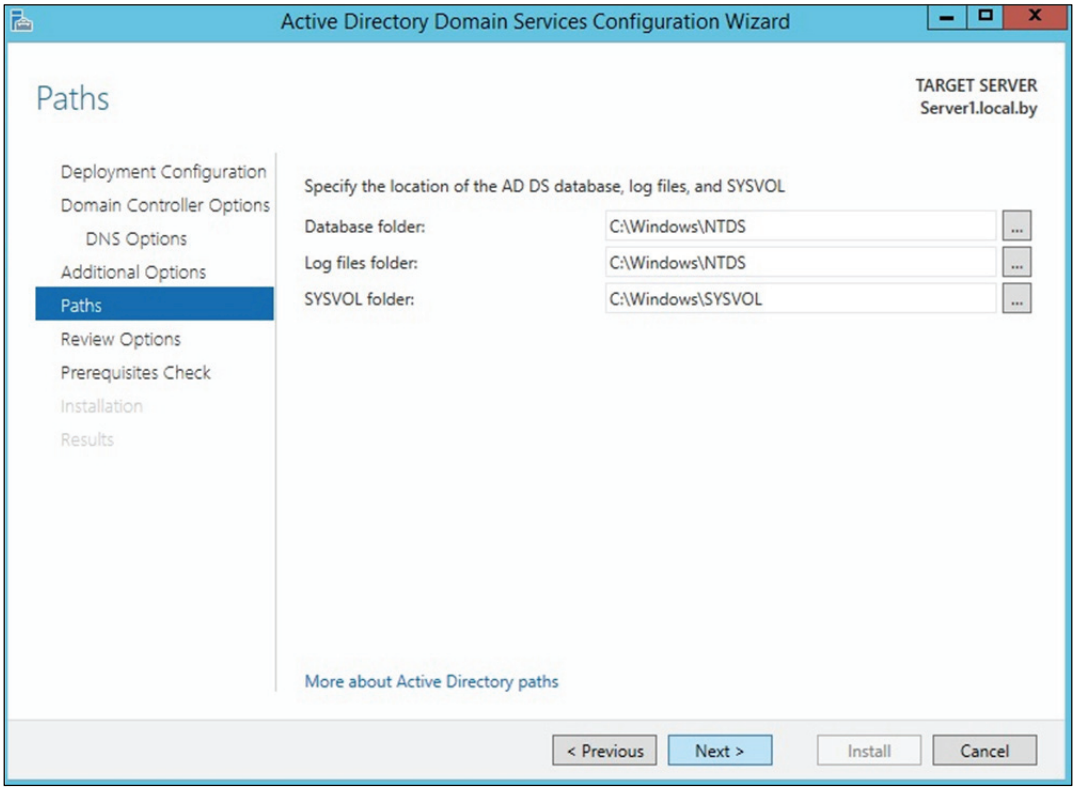


Рис. 4.12. Определение путей к каталогам базы данных,
журналам и папке Sysvol создаваемого домена

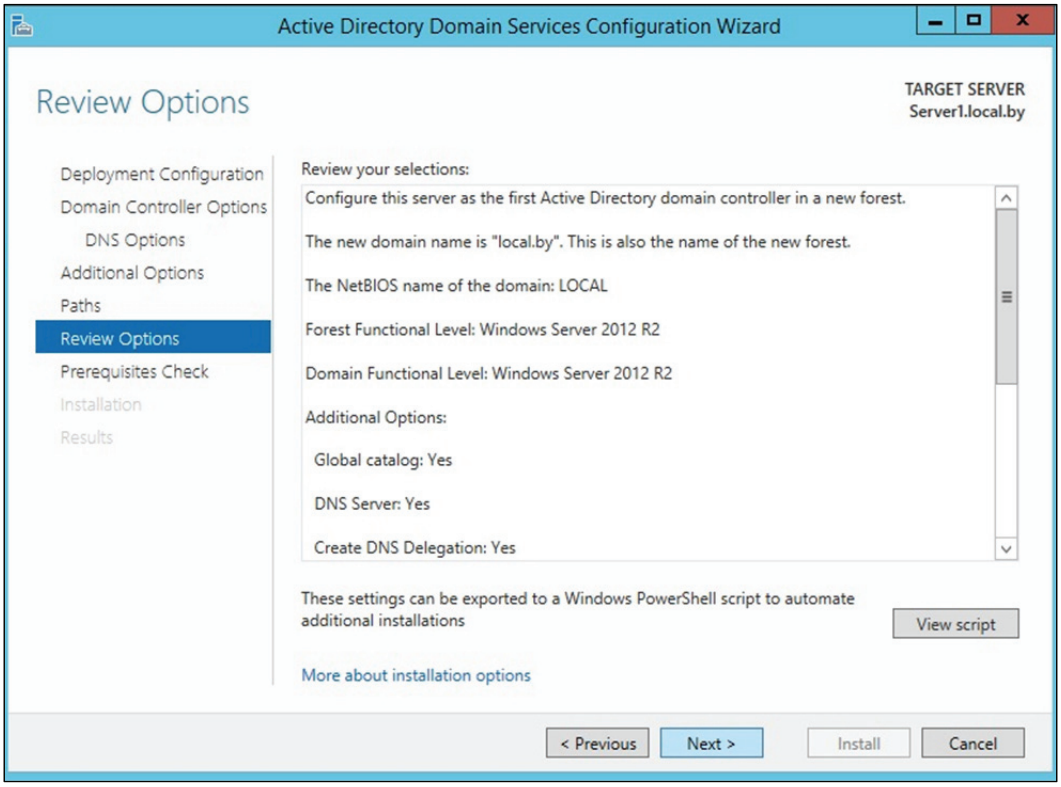


Рис. 4.13. Просмотр сводной информации по настройке домена

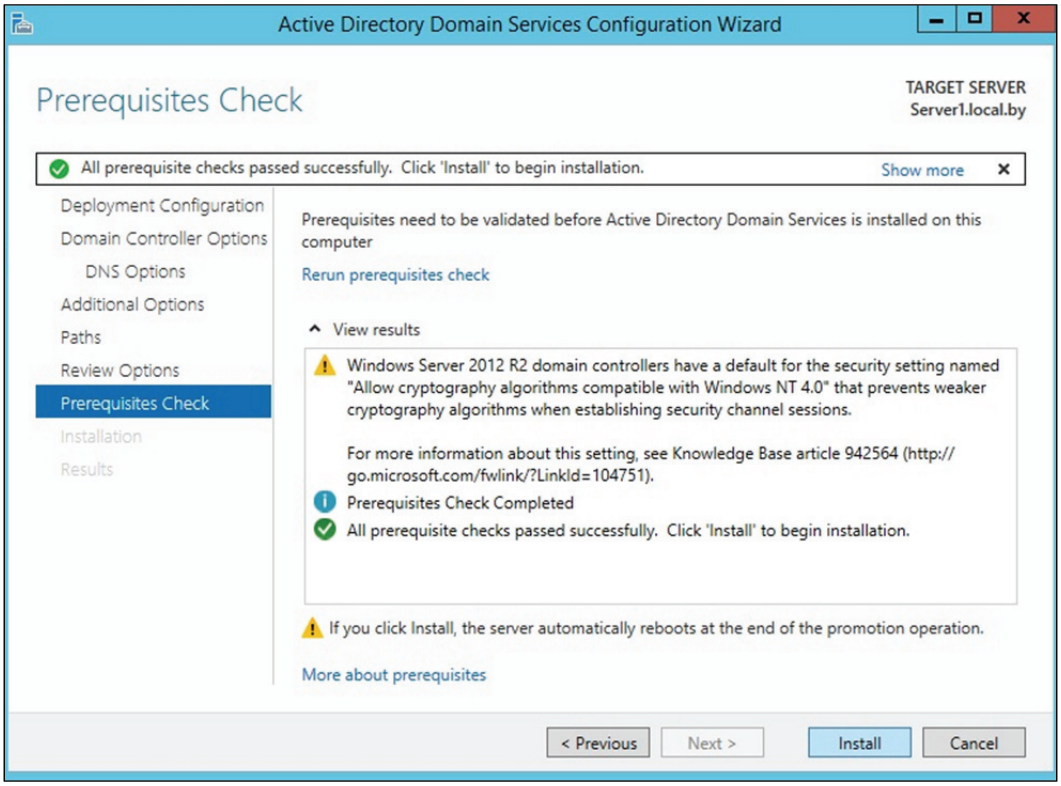


Рис. 4.14. Просмотр отчета о выполнении всех требований при создании домена

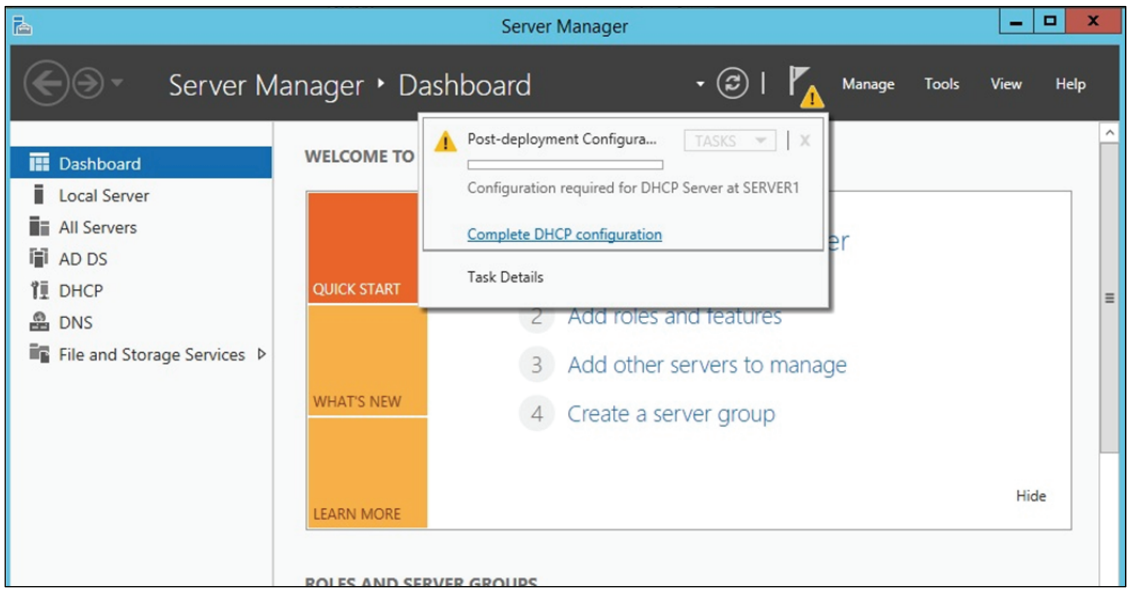


Рис. 4.15. Завершение настройки DHCP
(для повторной авторизации)

На следующих нескольких шагах фактически необходимо нажимать *Next* (*Далее*), тем самым соглашаясь с предложенным вариантом авторизации DHCP-сервера, как показано на рис. 4.16–4.18.

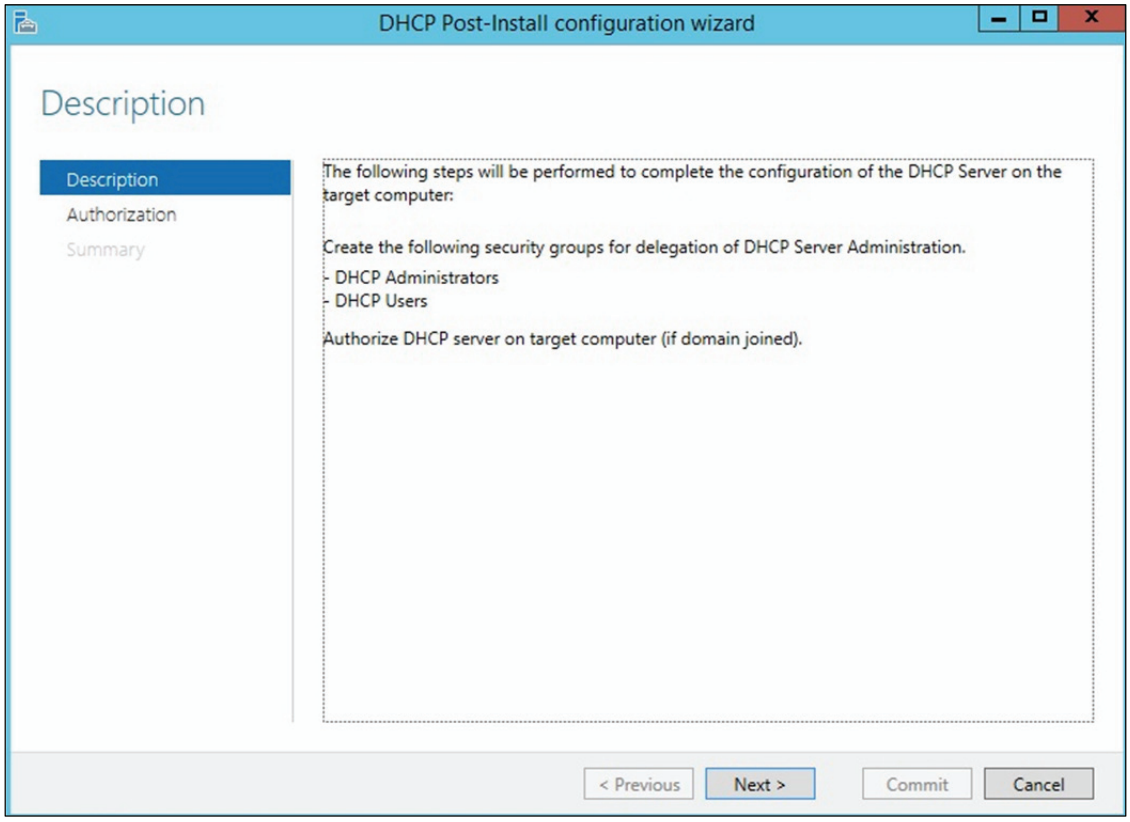


Рис. 4.16. Описание этапов установки DHCP-сервера

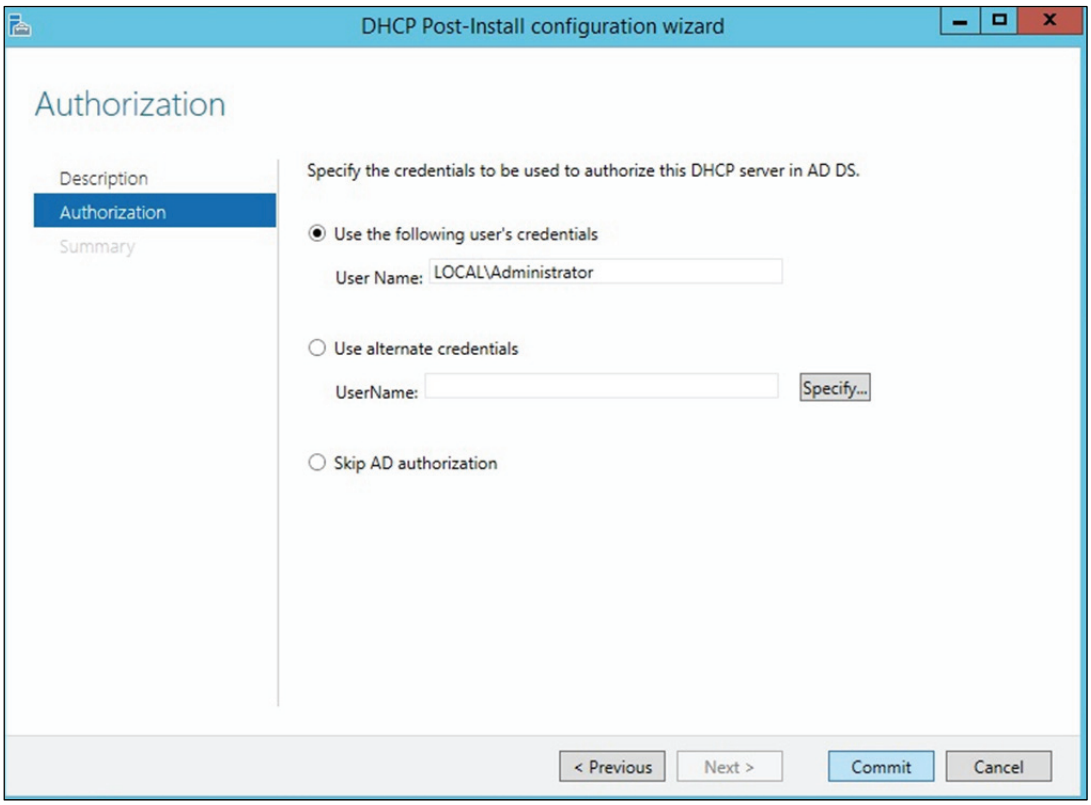


Рис. 4.17. Этап авторизации DHCP-сервера

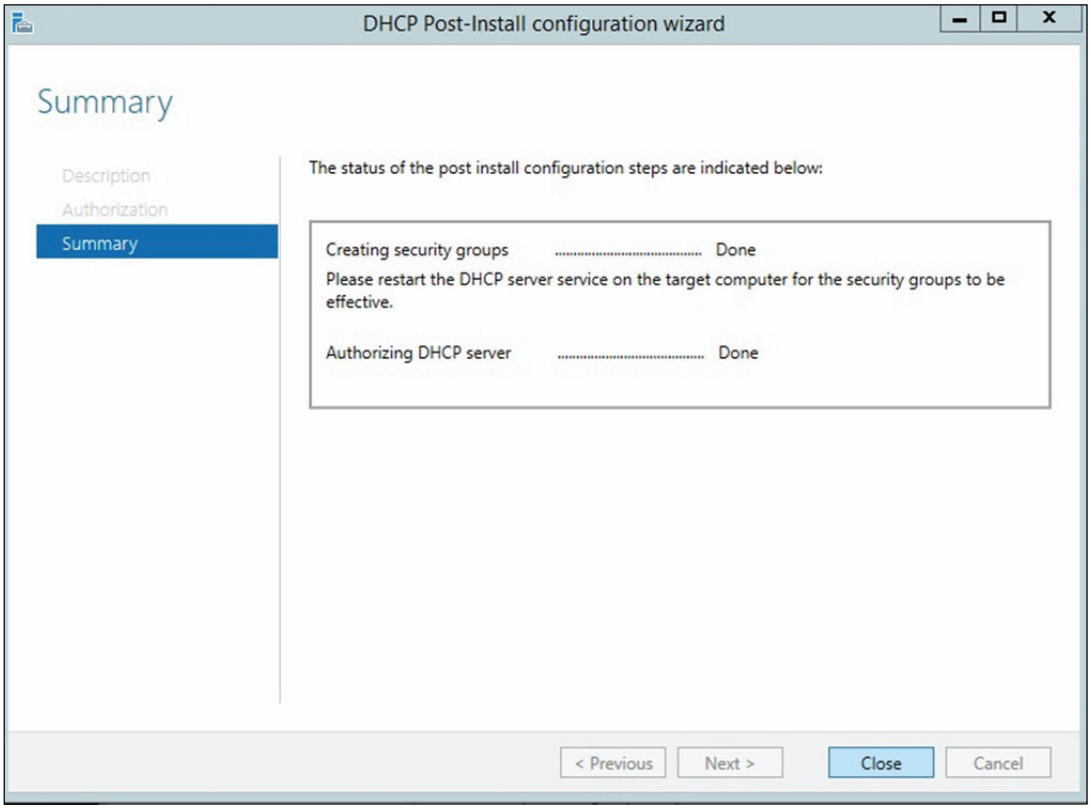


Рис. 4.18. Завершение процесса установки DHCP-сервера

4.7.2. Присоединение компьютера к домену

Для этого необходим компьютер клиента, имеющий связь с сервером (например, стабильно проходит ping-запрос, и что важно, как по IP-адресу, так и по DNS-имени). Дополнительно рекомендуется проверить правильность конфигурации DNS-сервера, чтобы на нем была создана запись ресурса службы (SRV). На втором компьютере DNS должна быть сконфигурирована так, чтобы он мог находить сервер как контроллер домена, с именем, выбранным нами, например, local.by.

1. Входим в систему на клиентском компьютере. Чтобы изменять членство этого компьютера в доменах, нужно войти в систему под учетной записью локальной группы *Администраторы* (*Administrators*).

2. Открываем вкладку *Имя компьютера* (*Computer Name*). Для этого дважды щелкаем *Система* (*System*) в панели управления, в боковом меню выбираем *Дополнительные параметры системы* (*Advanced*) и далее вкладку *Имя компьютера* (*Computer name*). На открывшейся вкладке кликаем *Изменить* (*Change*) (рис. 4.19).

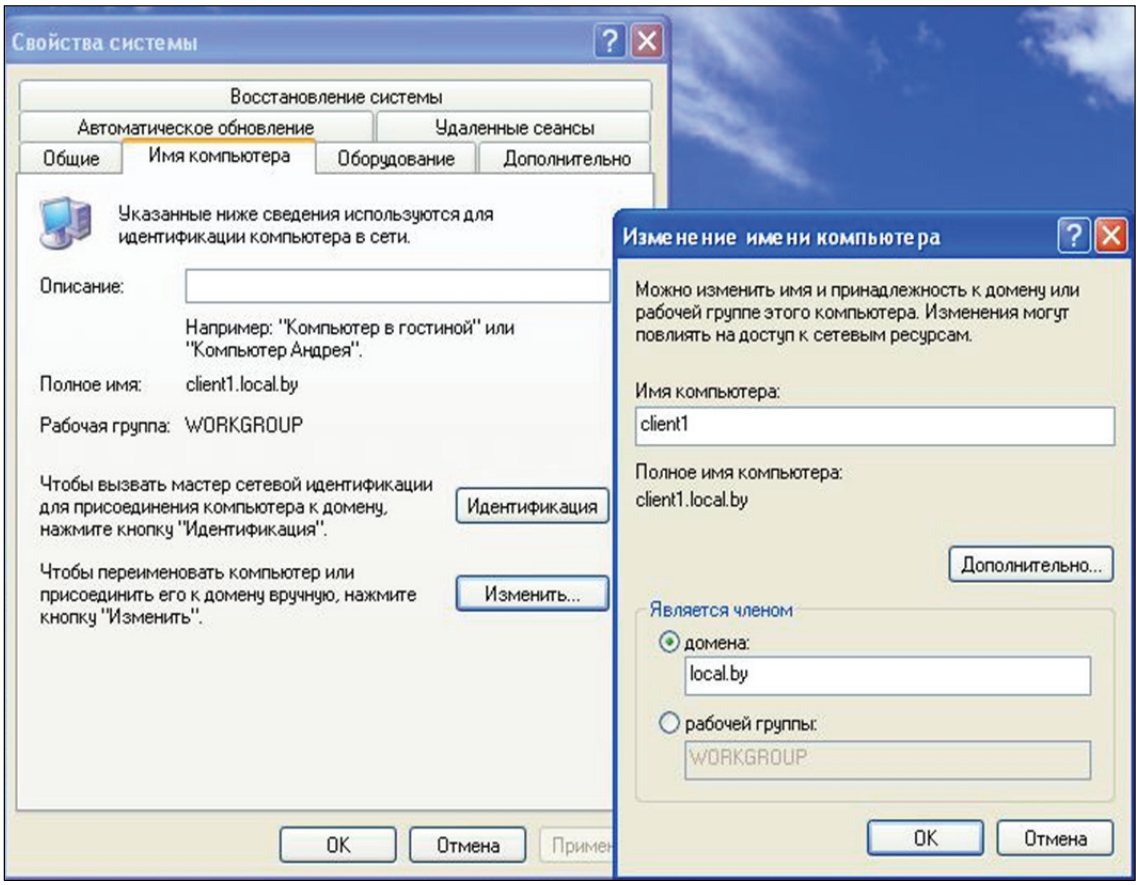


Рис. 4.19. Введение клиента в домен

3. Устанавливаем переключатель в положение домена (Domain) и вводим DNS-имя домена: в нашем примере это local.by. Далее нажимаем *OK* (рис. 4.19).

4. По запросу вводим имя и пароль учетной записи администратора домена local.by (рис. 4.20) и щелкаем *OK*.

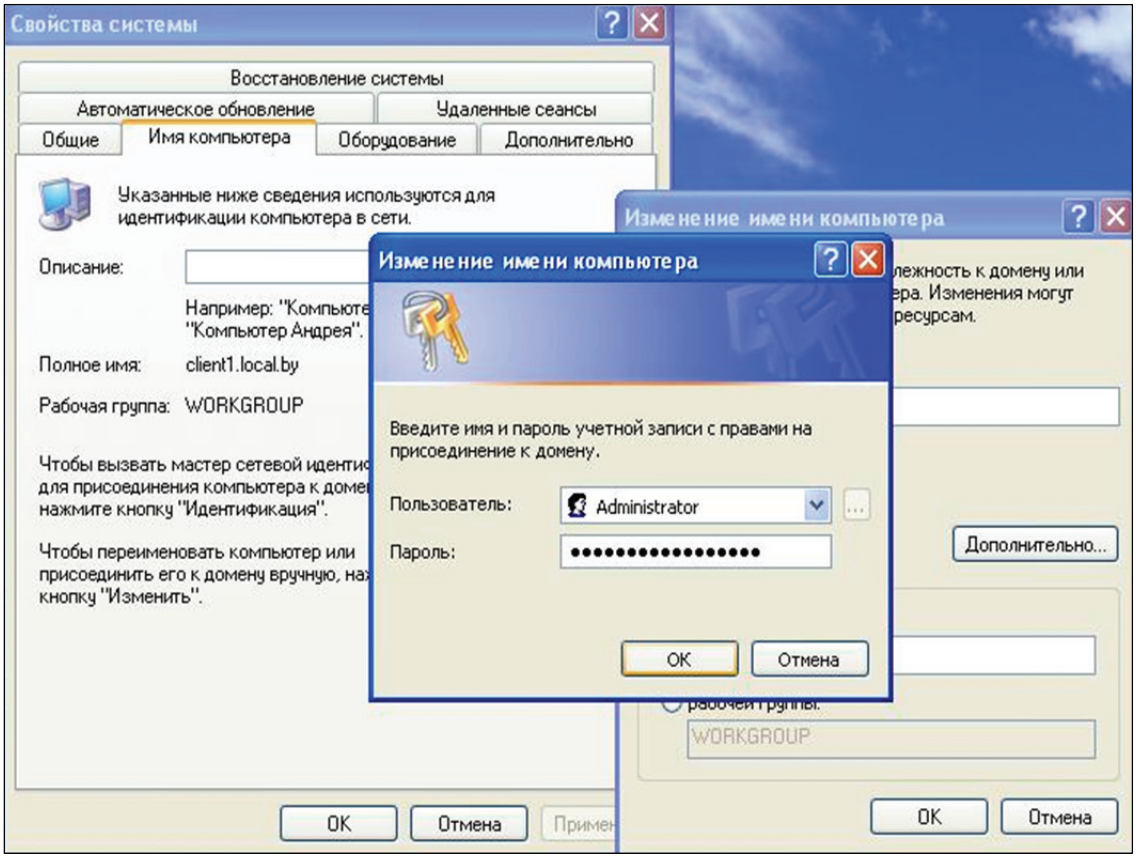


Рис. 4.20. Ввод логина и пароля администратора домена
(для получения права на присоединение к домену)

5. Далее будет получено сообщение о присоединении к домену (рис. 4.21) и предложено перезагрузить систему. Нажимаем *OK* в ответ на все сообщения и закрываем все диалоговые окна. Перезагружаем систему. В дальнейшем мы сможем входить в операционную систему под пользователями домена.

Отметим, что при вводе компьютера в домен для него должна автоматически создаться учетная запись для компьютера с соответствующим именем. Если таковое не будет выполнено, то учетную запись компьютера нужно создать самостоятельно. Для этого открываем консоль *Active Directory Users and Computers* (*Active Directory пользователи и компьютеры*) и, используя контекстное меню, выбираем создание учетной записи компьютера (рис. 4.22).

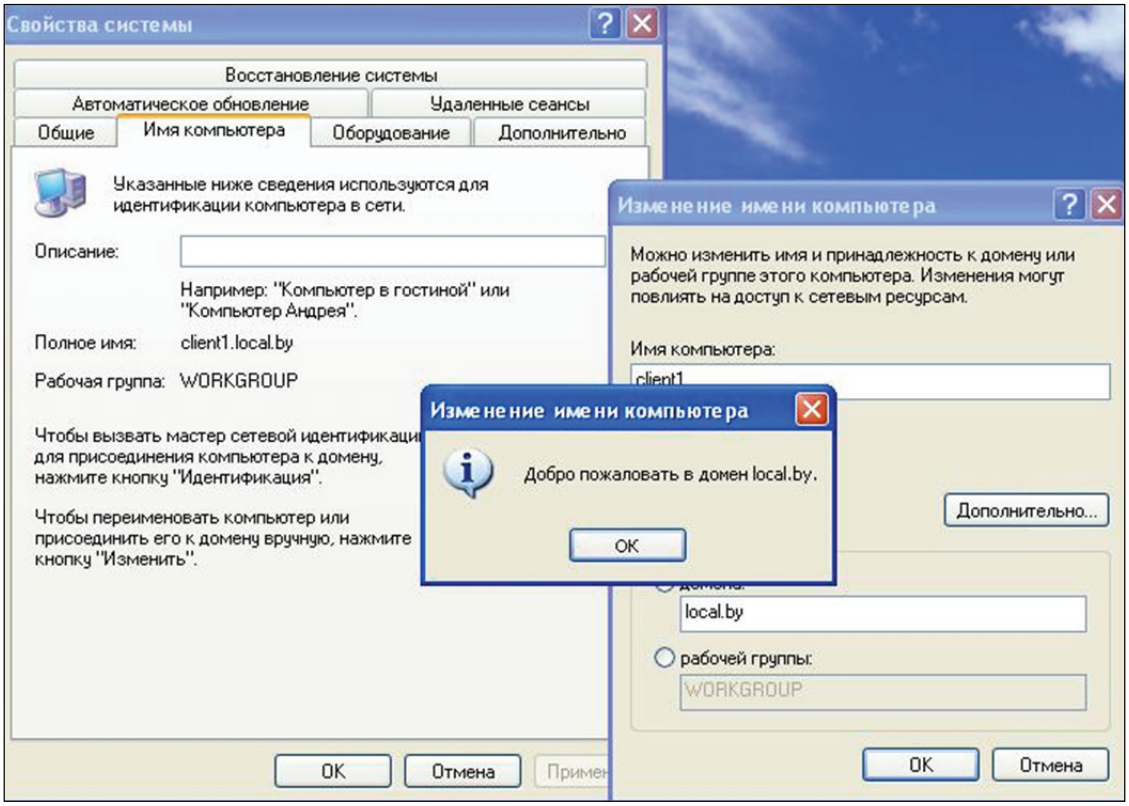


Рис. 4.21. Сообщение о введении компьютера в домен

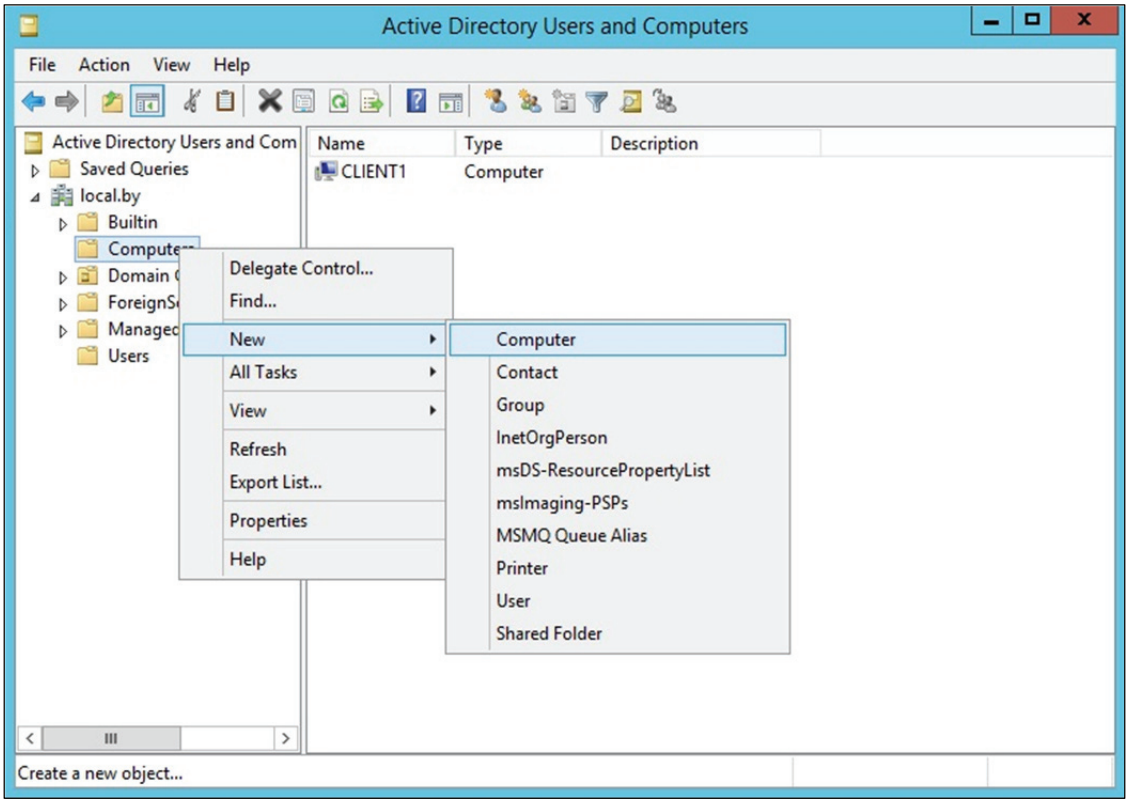


Рис. 4.22. Создание учетной записи компьютера на контроллере домена

4.7.3. Создание учетных записей пользователей. Распределение ресурсов

Для создания объектов пользователей выполним следующие действия.

1. Входим на Serverl как *Администратор* (*Administrator*).

2. Открываем консоль *Active Directory Users and Computers* (*Active Directory пользователи и компьютеры*) (рис. 4.23).

3. Выбираем группу *Users* и вызываем контекстное меню для создания пользователя (можно также создавать в организационном подразделении – это будет важно при удаленном администрировании с использованием групповых политик) (рис. 4.24).

4. Создаем учетную запись пользователя, причем задаем надежный пароль, так как при созданном домене обязательным является использование сложных паролей, например, содержащих две раскладки клавиатуры либо два разных языка, а также цифры и знаки (рис. 4.25 и 4.26).

5. Завершаем создание пользователя, нажав *Next* (*Далее*), а после ввода пароля – *OK*. Также рекомендуется задать подходящие свойства объекта пользователя на вкладках *Общие* (*General*), *Адрес* (*Address*), *Профиль* (*Profile*), *Телефоны* (*Telephones*) и *Организация* (*Organization*). Необходимо отметить, что заполнение данных полей не является обязательным, однако при построении реальной системы с большим числом пользователей рекомендуется.

Организационные подразделения (organizational unit) создаются аналогично другим объектам домена (пользователи, группы пользователей) (см. рис. 4.27 на с. 68). Они являются объектами контейнерного типа, а значит, в них, равно как и в группах пользователей, можно создавать учетные записи пользователей.

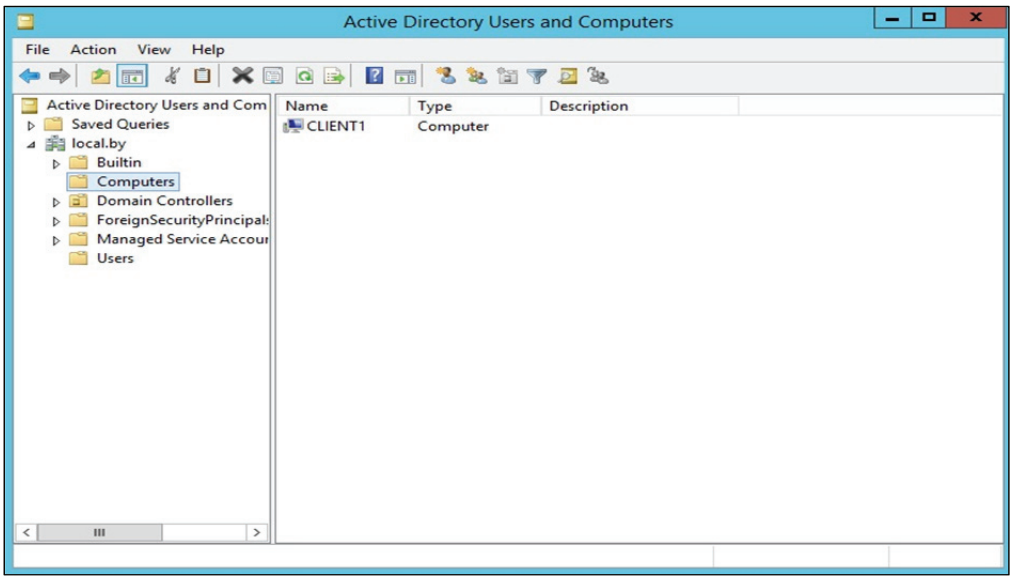


Рис. 4.23. Консоль *Active Directory Users and Computers*

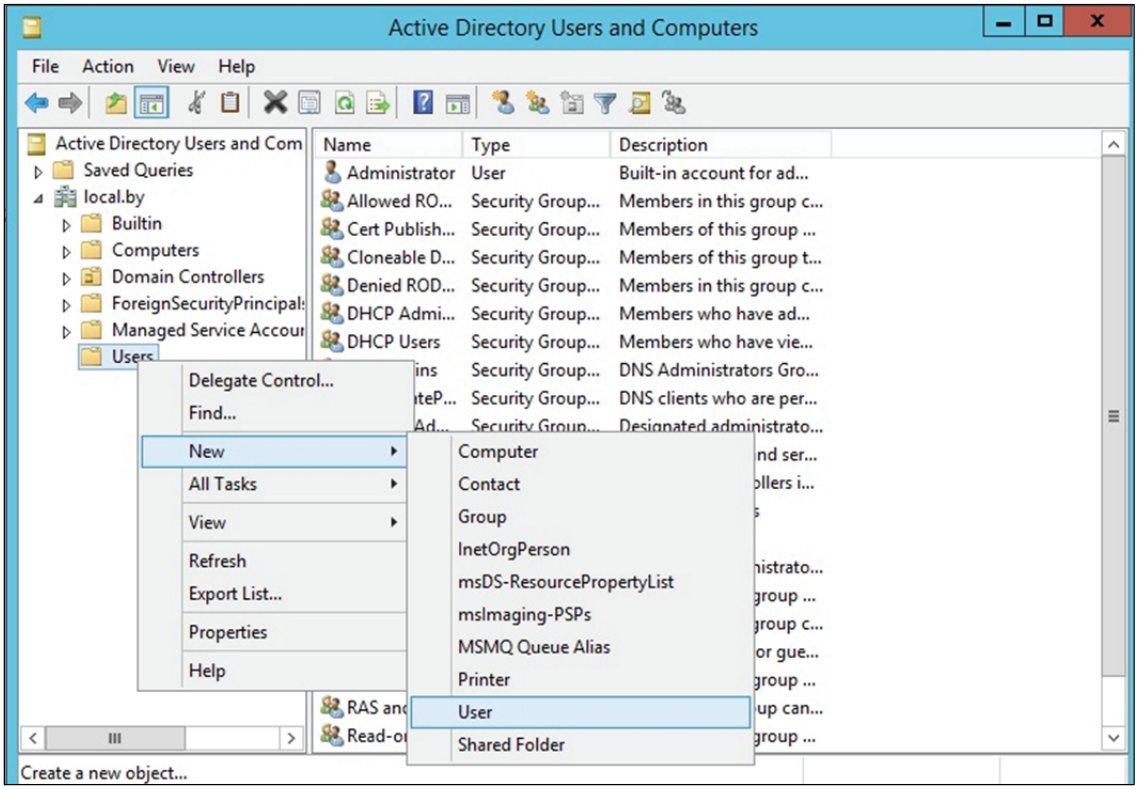


Рис. 4.24. Создание пользователя в консоли
Active Directory Users and Computers

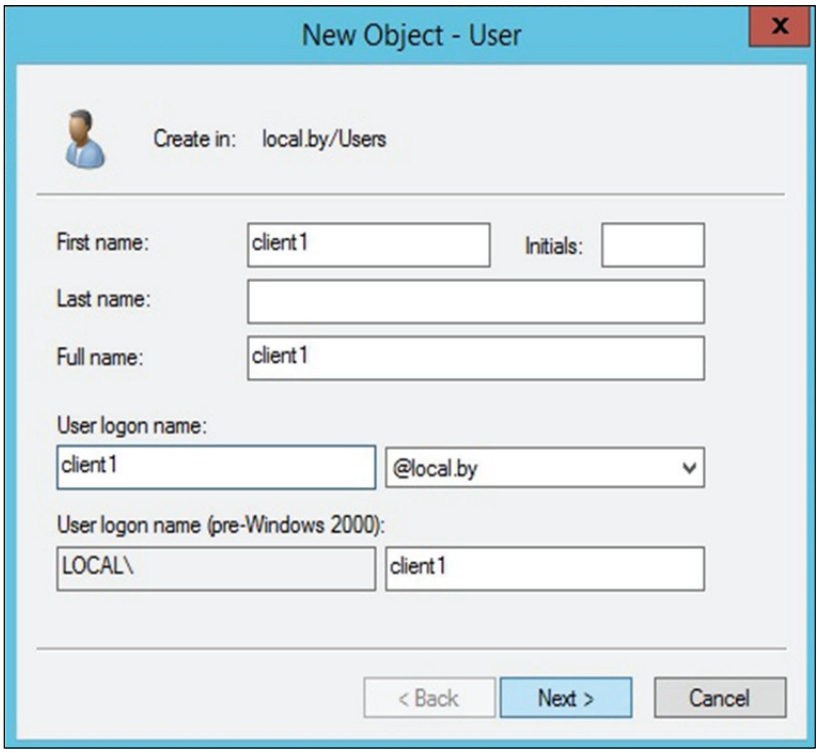


Рис. 4.25. Создание пользователя
с заданными параметрами

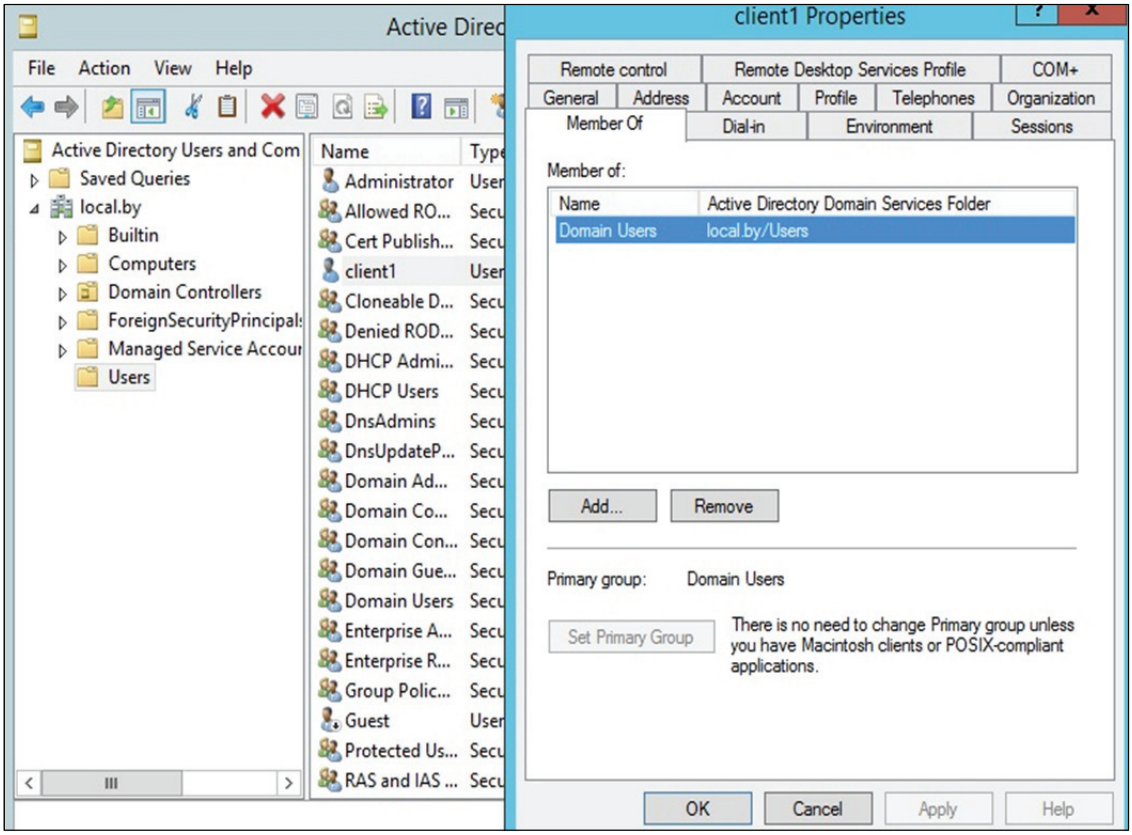


Рис. 4.26. Проверка принадлежности пользователя к группе

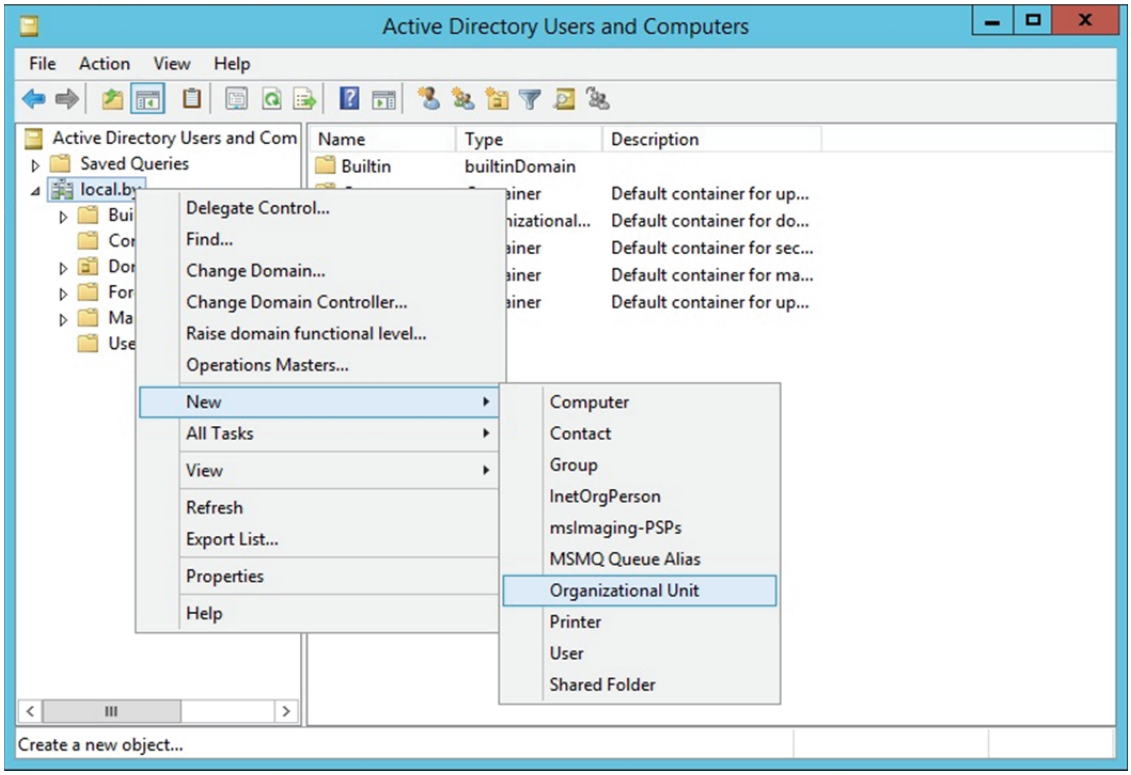


Рис. 4.27. Создание организационного подразделения

Чтобы создать домашний каталог пользователя, выполним следующие операции.

1. Открываем свойства соответствующего пользователя.
2. Переходим на вкладку *Профиль* (*Profile*) (рис. 4.28).

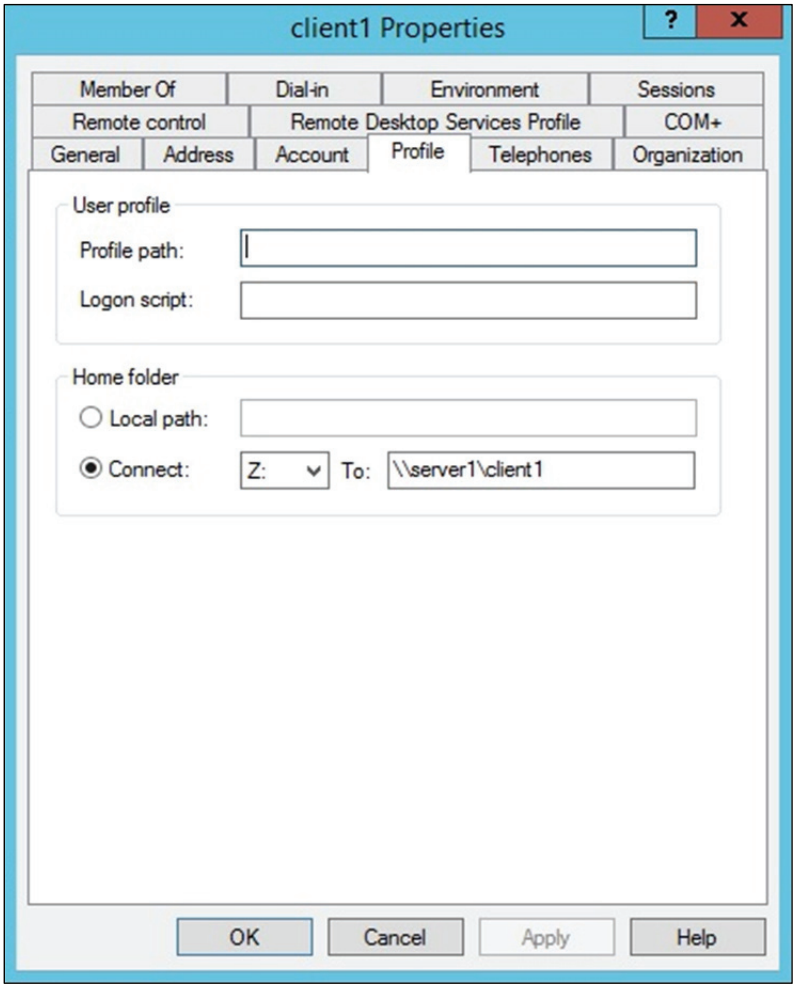


Рис. 4.28. Создание домашней папки пользователя

3. В поле *Profile path* (*Путь к профилю*) вводим сетевой путь к подготовленной домашней папке пользователя (ее целесообразно предварительно открыть в сеть с установлением всех необходимых прав через свойства файловой системы NTFS), например, \\server1\profiles\%username%.

4. Щелкаем *Apply* (*Применить*) и убеждаемся, что вместо переменной %username% было подставлено имя сервера. Важно, чтобы путь к профилю соответствовал фактическому сетевому пути к папке профиля.

5. Нажимаем *OK*.

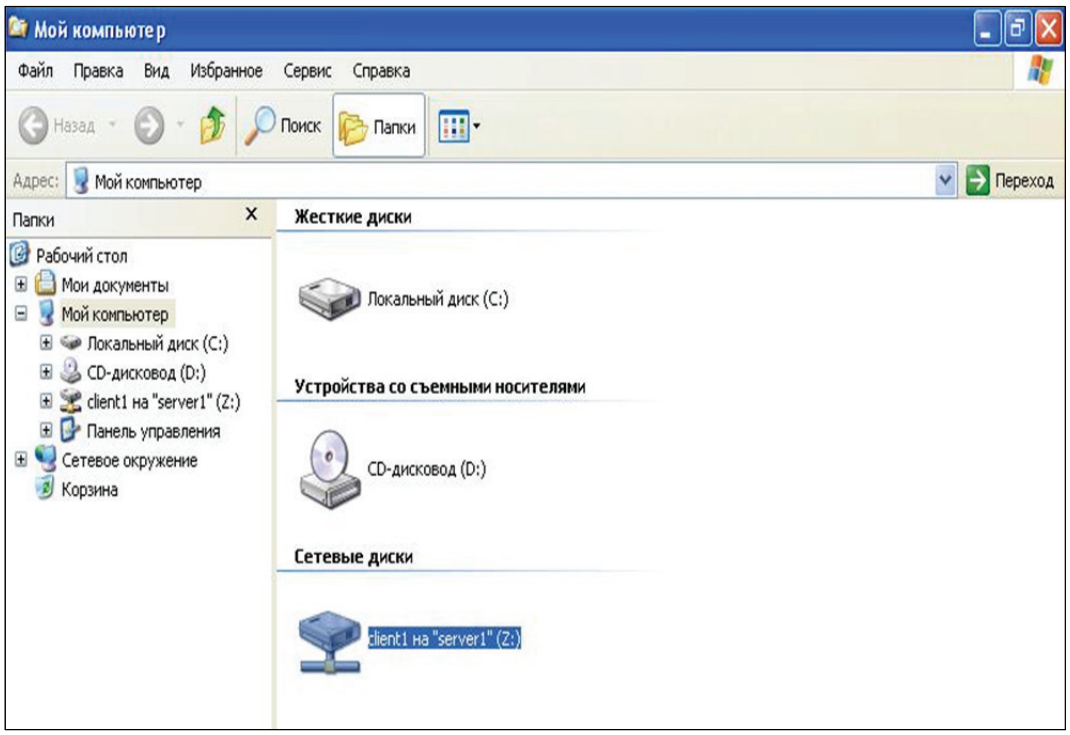


Рис. 4.29. Пример подключенной домашней папки пользователя в виде сетевого диска

После входа пользователя на клиентской машине в данном случае ему будет автоматически подключен домашний каталог в виде сетевого диска (рис. 4.29).

Лабораторная работа № 6–7

***Цель*:** создание и настройка домена, создание пользователей, подключение сетевых ресурсов.

***Задание*:** лабораторная работа состоит из двух частей.

1. Необходимо установить на один из серверов службу каталога Active Directory (имя домена целесообразно применять то же, что и DNS-суффикс, используемый в лабораторной работе № 5). Следует ввести две клиентские машины в домен. Отметим, что в данной лабораторной работе второй сервер использоваться не будет.

2. Необходимо создать двух пользователей (можно сделать так, чтобы они принадлежали разным организационным подразделениям, что будет полезным для следующих работ), а также настроить для каждого из них индивидуальные сетевые ресурсы (отметим, что сетевые папки для каждого из пользователей должны подключаться автоматически при входе пользователя в систему и быть доступны только пользователю).

Раздел 5
НАДЕЖНОСТЬ ДОМЕННЫХ СИСТЕМ

5.1. Структура каталога Active Directory

Вся информация об объектах сети содержится в каталоге Active Directory. Физически эта база данных представляет собой файл Ntds.dit, который хранится на контроллере домена.

Каталог Active Directory может рассматриваться с двух позиций: с точки зрения логической и физической структуры.

Логическая структура каталога Active Directory представлена на рис. 5.1. Цель такой структуризации – облегчение процесса администрирования.

Как говорилось выше, все сетевые объекты (пользователи, группы пользователей, компьютеры, принтеры) объединяются в домен, который является основной структурной единицей каталога. Для удобства управления объекты также могут быть сгруппированы при помощи *организационных подразделений.* Несколько иерархически связанных доменов образуют *дерево доменов.* Совокупность деревьев, имеющих общие части каталога Active Directory и общих администраторов, называется *лесом доменов.*

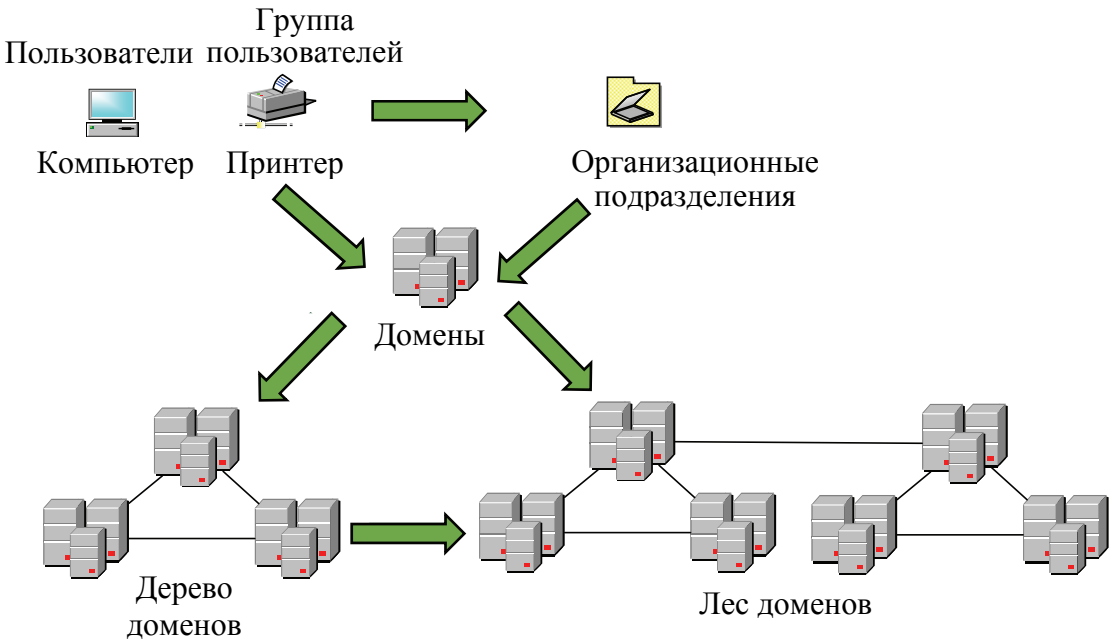


Рис. 5.1. Пример логической структуризации домена

Имея возможность такой логической структуризации, администратор может планировать и выбирать конфигурацию сети в зависимости от своих задач и масштабов организации.

Основной целью ***физической структуризации*** каталога Active Directory является оптимизация процесса копирования изменений, произведенных на одном из контроллеров домена, на все остальные контроллеры. Этот процесс называется *репликацией* (replication).

Основой физической структуры является *сайт* (site) – это часть сети, все контроллеры домена которой связаны высокоскоростным соединением. Между сайтами, наоборот, установлены более медленные линии связи (рис. 5.2).

Подобная структура позволяет планировать процесс репликации следующим образом: внутри сайта репликация осуществляется часто, и могут передаваться большие объемы информации без сжатия; между сайтами изменения реплицируются редко, и данные требуется сжимать.

Логическая и физическая структуры предназначены для решения разных задач и поэтому между собой практически не связаны: в одном домене может быть несколько сайтов, так же как один сайт может содержать несколько доменов. Общим объектом для той и другой структуры является контроллер домена с хранящимся на нем файлом каталога Ntds.dit.

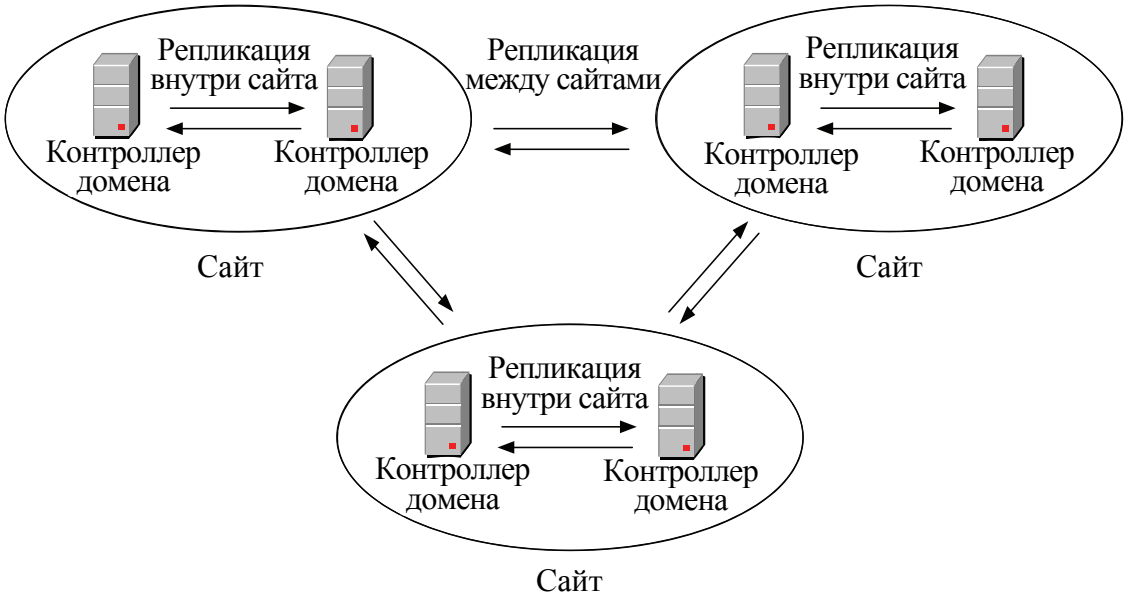


Рис. 5.2. Пример физической структуризации домена

В файле каталога Active Directory содержится информация как о логической, так и о физической структурах. Этот файл состоит из нескольких разделов:

– раздел домена (domain partition) – содержатся данные обо всех объектах домена (пользователях, компьютерах, принтерах и т. д.);

– раздел схемы (schema partition) – хранится информация о типах всех объектов, которые могут быть созданы в данном лесе доменов;

– раздел конфигурации (configuration partition) – описывается конфигурация леса доменов – информация о сайтах, соединениях между сайтами и направлениях репликации;

– раздел приложений (application partition) – специальный раздел для хранения данных приложений, не относящихся к службе Active Directory. По умолчанию здесь создается подраздел для службы DNS;

– раздел глобального каталога (global catalog partition). *Глобальный каталог* – это база данных, в которой содержится список всех объектов леса доменов без информации об атрибутах этих объектов. Глобальный каталог необходим для поиска ресурсов леса из любого принадлежащего ему домена.

В зависимости от принадлежности к разделу информация реплицируется между контроллерами доменов следующим образом:

1) раздел домена реплицируется между контроллерами одного домена;

2) разделы схемы, конфигурации и глобального каталога реплицируются на все контроллеры леса;

3) репликацией раздела приложений можно управлять – указывать, какие контроллеры будут получать реплику данного раздела.

5.2. Планирование Active Directory

Успешная работа пользователей сетевых ресурсов, а также служб, реализующих протоколы TCP/IP, зависит от правильного функционирования Active Directory. Поэтому крайне важной становится задача планирования структуры каталога Active Directory. Удачно спроектированный каталог позволит сделать работу сети более эффективной и стабильной, а также намного облегчит труд администратора.

В процессе планирования Active Directory можно выделить два основных этапа (рис. 5.3):

– планирование логической структуры, включающее проектирование доменов и организационных подразделений, а также проблему именования;

– планирование физической структуры, состоящее из разделения сети на сайты и размещения контроллеров домена.

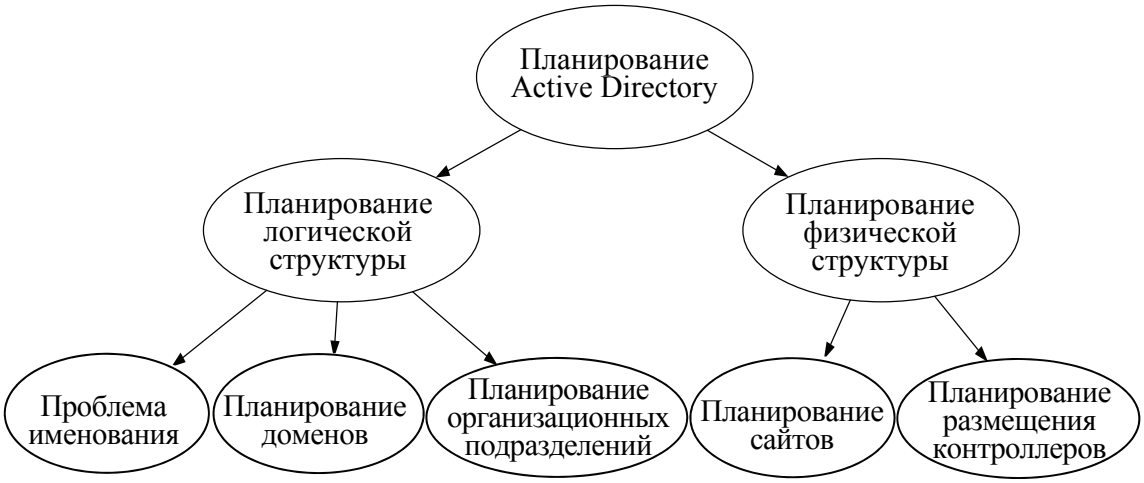


Рис. 5.3. Планирование Active Directory

5.2.1. Планирование логической структуры

При ***планировании доменной структуры*** нужно определить количество и способ организации доменов. Возможны три варианта: единственный домен, дерево доменов или лес. Критерии выбора следующие:

1) размер организации – один домен может содержать до сотен тысяч пользователей, но не рекомендуется допускать превышение данного условного предела;

2) географическое расположение – имеются ли у организации филиалы или отделы, находящиеся на большом расстоянии и связанные с центральным звеном низкоскоростными каналами связи. Наличие таких филиалов при единственном в организации домене, скорее всего, вызовет перегрузку линий связи из-за трафика репликации;

3) стабильность организации – насколько высока подвижность кадрового состава, не планируется ли в ближайшее время структурное преобразование организации;

4) потребности в разных доменных именах – в некоторых случаях в рамках одной организации требуются разные доменные имена. Например, в случае создания единой компьютерной системы двух университетов каждый из них, вероятно, захочет иметь свое собственное доменное имя;

5) способ управления сетью – может быть централизованным и децентрализованным. Централизованный способ предполагает сосредоточение всей административной власти у единого коллектива администраторов и наличие однодоменной модели. При децентрализованном способе полномочия делегируются нескольким слабосвязанным удаленным группам администраторов, управляющих доменами дерева или леса;

6) единство политики безопасности. Чаще всего политика безопасности в одной организации едина для всех отделов и сотрудников, однако бывают исключения.

Исходя из перечисленных критериев, можно выделить те признаки, по которым выбирается вариант с одним доменом:

– в организации менее сотни тысяч пользователей;
– отсутствие удаленных филиалов;
– относительная стабильность структуры организации;
– отсутствие потребности в разных доменных именах;
– централизованный способ администрирования;
– единая политика безопасности.

Отсутствие первых четырех признаков существенно склоняет выбор в пользу многодоменной модели. Последние два признака в меньшей степени должны влиять на выбор, так как задачи делегирования администрирования и разделения политик безопасности можно решить средствами организационных подразделений в рамках одного домена.

При выборе модели с несколькими доменами в большинстве ситуаций нужно использовать дерево доменов. Лес доменов приемлем в том случае, когда две независимые организации хотят иметь общие сетевые ресурсы.

После выбора доменной структуры следует продумать ***имена для создаваемых доменов***. Особенно важно имя корневого домена. Правил для выбора доменного имени немного: во-первых, оно должно отражать специфику организации, во-вторых, быть понятным всем пользователям ресурсов домена, а не только администратору и, в-третьих, не должно быть слишком сложным. Для имени очень часто используют аббревиатуры, например, belstu и т. д.

Планирование структуры организационных подразделений в каждом домене является важным шагом.

Как отмечалось в предыдущей теме, организационные подразделения применяются в том случае, если для задач управления группой объектов или делегирования административных прав образование новых доменов нецелесообразно.

В связи с тем, что организационные подразделения можно использовать в качестве контейнеров, допускается строить иерархию организационных подразделений с несколькими уровнями вложений.

Иерархию можно строить с помощью двух основных подходов: либо следуя организационной структуре предприятия (*организационный подход*), либо исходя из задач управления сетевыми объектами (*административный подход*). Оба способа используются на практике,

и задача администратора состоит в том, чтобы выяснить, какой из подходов (или их комбинация) применим в данной ситуации.

5.2.2. Планирование физической структуры

Основная цель планирования физической структуры – оптимизация трафика репликации. Цель достигается путем продуманного расположения сайтов и контроллеров домена.

Основной объем данных репликации присутствует в рамках одного домена, междоменный же трафик репликации существенно ниже внутридоменного. Для оптимизации процесса репликации рекомендуется использовать механизм сайтов.

На начальном этапе следует проанализировать существующую сеть – ее структуру, количество пользователей и компьютеров, пропускную способность, колебания трафика. Все эти данные нужно учитывать при планировании. Чем больше пользователей и компьютеров в сети, тем больше объем передаваемой информации при репликации. Линии с большой пропускной способностью могут быть сильно загружены, и большой трафик репликации внесет существенные проблемы, в то время как низкоскоростные каналы, возможно, практически свободны и выдержат дополнительный объем данных репликации.

Во время анализа следует учитывать возможность расширения сети и увеличения числа пользователей. Считается достаточным принимать коэффициент расширения в пределах 30–50%.

Основной критерий при выделении сайтов – пропускная способность линий связи. Части домена, связанные высокоскоростными линиями, помещаются в один сайт. Если между частями домена имеются каналы с низкой скоростью передачи данных, их следует разместить в разных сайтах. При этом трафик межсайтовой репликации сжимается, и его передача происходит во время наименьшей загрузки низкоскоростных линий.

Вопрос о необходимом количестве и размещении контроллеров домена решается тогда, когда известна доменная структура и расположение сайтов. Общее правило таково, что для каждого домена необходимо не менее двух контроллеров (при этом в случае отказа одного из контроллеров второй обеспечит работу сети). Количество контроллеров зависит от числа пользователей (а следовательно, числа обращений на контроллеры домена), принадлежащих данному домену или сайту. Например, если домен включает два сайта, связанных модемной линией, и к одному из сайтов принадлежит всего несколько пользователей, то

совсем не обязательно в этом сайте располагать отдельный контроллер домена (при условии, что загрузка модемной линии невысока).

5.3. Настройка репликации

Если ранее не был установлен домен (служба Active Directory) на ваш второй сервер, то пункт 5.3.1 можно пропустить.

5.3.1. Удаление Active Directory и установка второго контроллера домена

1. Если на втором сервере уже установлена служба Active Directory, то ее надо удалить. На соответствующем компьютере запускаем программу *Server Manager* (*Диспетчер сервера*) и щелкаем по *Manage* → *Remove role and features*. Далее начнется процесс удаления выбранной роли сервера, по шагам аналогичный процессу установки, поэтому подробно рассматривать его не будем. Запускаем удаление домена.

2. После окончания удаления Active Directory отказываемся от немедленной перезагрузки, открываем свойства своего сетевого соединения и указываем в свойствах TCP/IP в поле *Primary DNS Server* IP-адрес первого сервера (отметим, что данная операция, возможно, была уже сделана при настройке вторичного DNS-сервера). Производим перезагрузку компьютера.

3. После окончания перезагрузки еще раз запускаем *Server Manager* (*Диспетчер сервера*) и приступаем к установке роли Active Directory, как рассматривалось в подразделе 4.7.

4. На экране *Deployment Configuration* устанавливаем переключатель в положение *Add a domain controller to an existing domain* (*Вторичный контроллер домена в существующем лесе*) и нажимаем *Next* (*Далее*) (рис. 5.4).

5. На последующих шагах набираем имя пользователя local\Administrator и пароль в соответствующее поле. Также вводим DNS-имя домена (в рассматриваемом примере это local.by). Остальные предлагаемые параметры оставляем по умолчанию.

6. На соответствующем шаге (*Directory Services Restore Mode Administrator Password*) набираем два раза пароль для режима восстановления Active Directory. Щелкаем *Next* (*Далее*) на этом и следующих экранах и производим установку Active Directory. По окончании установки перезагружаем компьютер.

Таким образом, был создан второй контроллер домена, который будет вторичным к основному. Между ними будут реплицироваться

любые изменения, происходящие на любом из контроллеров, например, создание либо удаление пользователей и т. д.

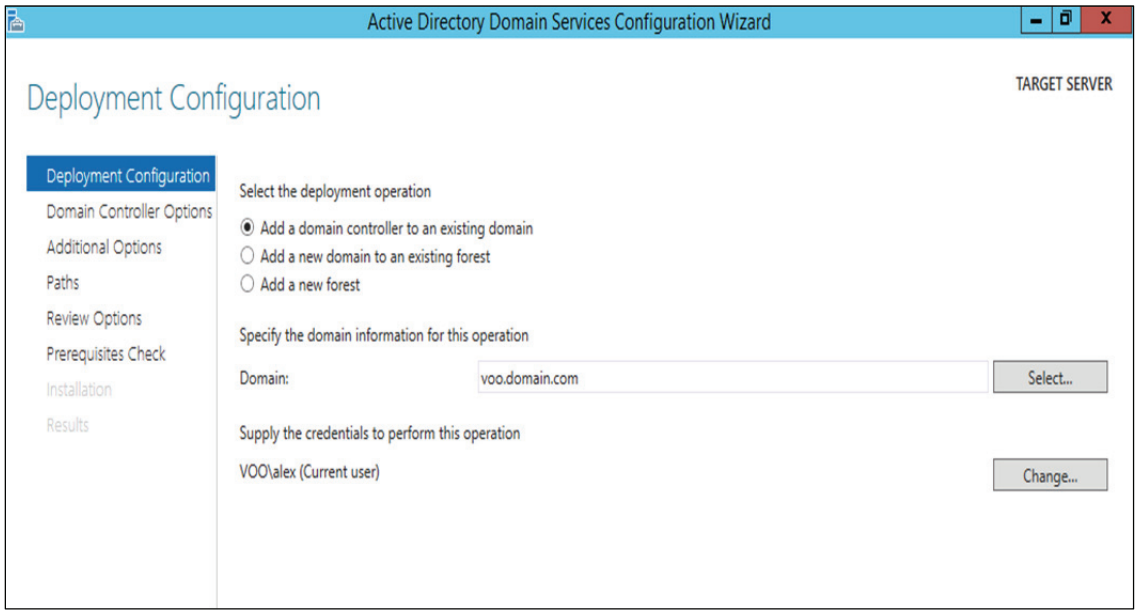


Рис. 5.4. Создание вторичного контроллера домена

5.3.2. Создание системы сайтов Active Directory и настройка расписания репликации

Пусть необходимо создать в нашем лесу второй сайт, который будет называться Site2, и поместить в него второй контроллер домена. Настроим расписание репликации между сайтами таким образом, чтобы оно выполнялось только в определенный (удобный для нас) промежуток времени.

1. Открываем консоль *Active Directory Sites and Services*. Щелкаем правой кнопкой мыши по контейнеру *Sites* и в контекстном меню выбираем *New Site*. Вводим имя создаваемого сайта Site2, в списке соединений *Site Link* выбираем единственный, имеющий соединение – DEFAULTSITELINK, и нажимаем *OK*.

2. Раскрываем узел *Sites* → *Default First Site Name* → *Servers*, щелкаем правой кнопкой мыши по объекту второго контроллера домена, в контекстном меню выбираем *Move*, в списке *Site Name* выбираем Site2 и кликаем *OK*.

3. Раскрываем узел *Inter* → *Site Transports*, затем узел IP, щелкаем правой кнопкой мыши по объекту DEFAULTIPSITELINK в правой части экрана и в контекстном меню выбираем Properties.

4. На вкладке *General* свойств DEFAULTIPSITELINK нажимаем на кнопку *Change Schedule*, выделяем весь прямоугольник и устанав-

ливаем переключатель в положение *Replication Not Available*. Затем выделяем столбец, соответствующий времени репликации, и устанавливаем для него переключатель в положение *Replication Available*. Нажимаем на кнопку *OK* два раза и закрываем консоль *Active Directory Sites and Services*.

Лабораторная работа № 8

***Цель*:** изучение методов обеспечения надежного функционирования доменной системы (путем настройки репликации контроллеров доменов).

***Задание*:** необходимо выполнить настройку репликации контроллеров доменов двумя способами (с использованием вторичных контроллеров, т. е. внутридоменная репликация и репликация между сайтами).

Раздел 6
УДАЛЕННОЕ АДМИНИСТРИРОВАНИЕ

6.1. Групповые политики

С увеличением парка компьютеров в сети все более остро встает вопрос о стоимости его управления и содержания. Ручная настройка компьютеров отнимает немало времени у администраторов и персонала и заставляет с увеличением количества компьютеров расширять штат обслуживающего их персонала. К тому же при большом количестве машин следить за соблюдением принятых на предприятии стандартов настройки становится все труднее. Групповые политики (group policy, GP) являются комплексным инструментом централизованного управления компьютерами с ОС Windows Server в домене Active Directory. К компьютерам под управлением устаревших ОС Windows типа 95, 98, ME групповые политики не применяются: они управляются системными политиками (system policy), которые в рамках данного раздела рассматриваться не будут.

6.1.1. Объекты групповых политик

Групповые политики (group policy) – это способ автоматизации работы по настройке рабочих столов пользователей и параметров компьютеров. Групповые политики представляют собой наборы правил конфигурирования, применяемых к компьютеру или пользователю. Каждый такой набор правил называется ***объектом групповой политики*** (Group Policy Object, GPO).

Один или несколько объектов групповой политики могут применяться к трем видам объединений:

– сайтам;

– доменам;

– организационным подразделениям.

Кроме того, для каждого компьютера может быть определен ***объект локальной групповой политики*** (Local Group Policy Object, LGPO).

Объекты групповых политик являются наследуемыми. Это означает, например, что GPO, применяемый к домену, наследуется всеми его организационными подразделениями. В том случае если правила одного объекта групповой политики конфликтуют с правилами другого, наибольший приоритет имеет GPO организационного подразделения,

ниже по уровню GPO домена, затем следует GPO сайта, наименьший приоритет у LGPO.

Приведем краткий обзор возможностей, предоставляемых групповыми политиками (рис. 6.1).

Объект групповой политики содержит две основные части:

1) конфигурация компьютера (Computer Configuration);
2) конфигурация пользователя (User Configuration).

Каждая из частей включает три раздела:

– настройки приложений (Software Settings);
– настройки Windows (Windows Settings);
– административные шаблоны (Administrative Templates).

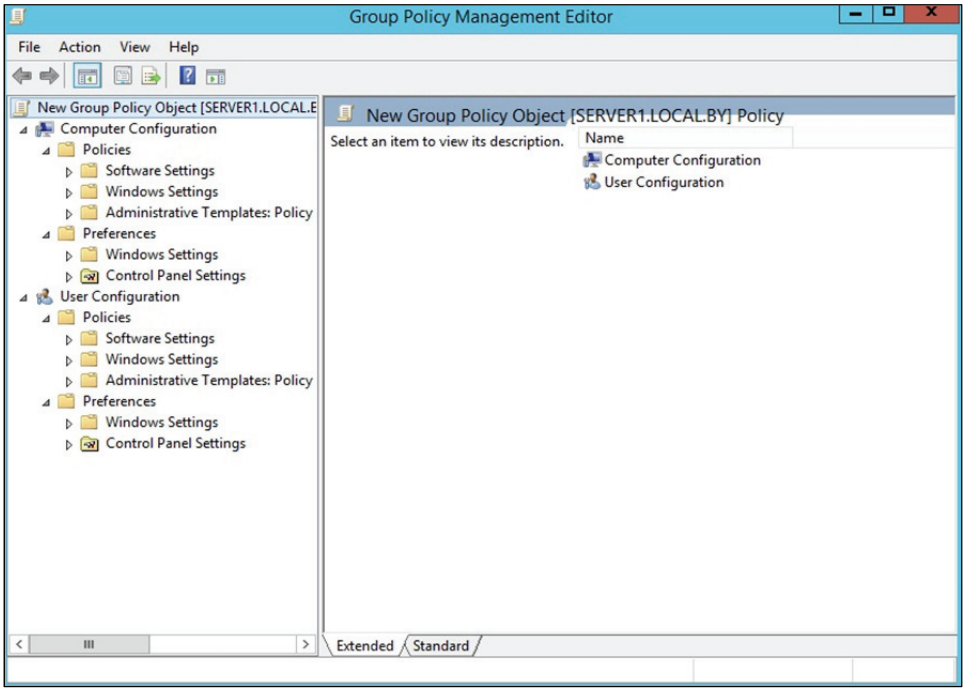


Рис. 6.1. Пример консоли для настройки объекта групповой политики

В разделе *Настройки приложений* находится подраздел *Установка приложений* (*Software Installation*), позволяющий автоматически устанавливать выбранные программы на компьютеры пользователей.

Правила, создаваемые в разделе *Настройки Windows*, позволяют:

1) выполнять задаваемые сценарии (Scripts) при включении-выключении компьютера, при входе пользователя в систему и выходе из нее;

2) настраивать параметры безопасности (Security Settings) компьютера и пользователя (требования к паролям, доступ к реестру, политику аудита событий);

3) конфигурировать Internet Explorer (Internet Explorer Maintenance);

4) изменять места расположения папок пользователей (Folder Redirection).

Раздел *Административные шаблоны* предназначен для настройки рабочего стола пользователя, ограничения доступа к системным компонентам и компонентам приложений.

Таким образом, Windows Server предоставляет мощный набор инструментов администрирования, способствующий эффективному управлению сети любой организации.

6.1.2. Создание объекта групповой политики

Для того чтобы создать политику (т. е. фактически создать новый объект групповой политики), открываем соответствующую консоль управления (в командной строке набираем gpmc.msc (рис. 6.2)) и выбираем созданное ранее организационное подразделение (в нашем примере это user_group1), либо создаем новое с добавлением туда нужных пользователей, для которого создаем новый объект GPO (Group Policy Object) (рис. 6.3), а также задаем ему название (можно оставить дефолтное) (рис. 6.4). Создавать и привязывать объект групповой политики можно только к объекту сайта, домена или организационному подразделению.

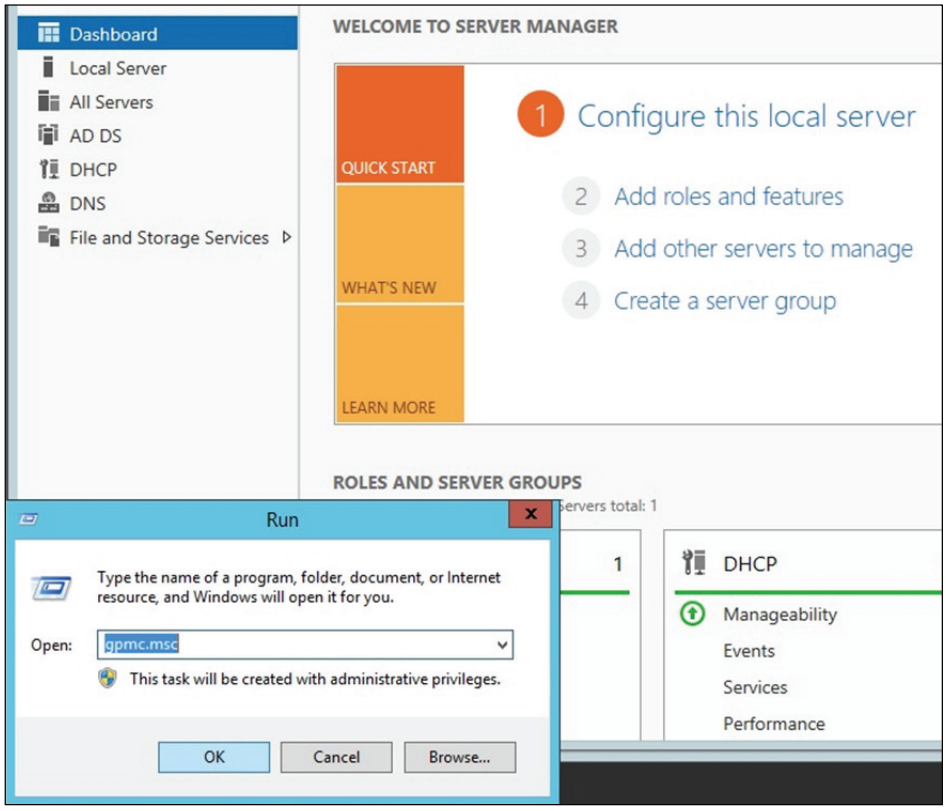


Рис. 6.2. Вызов консоли настройки групповой политики

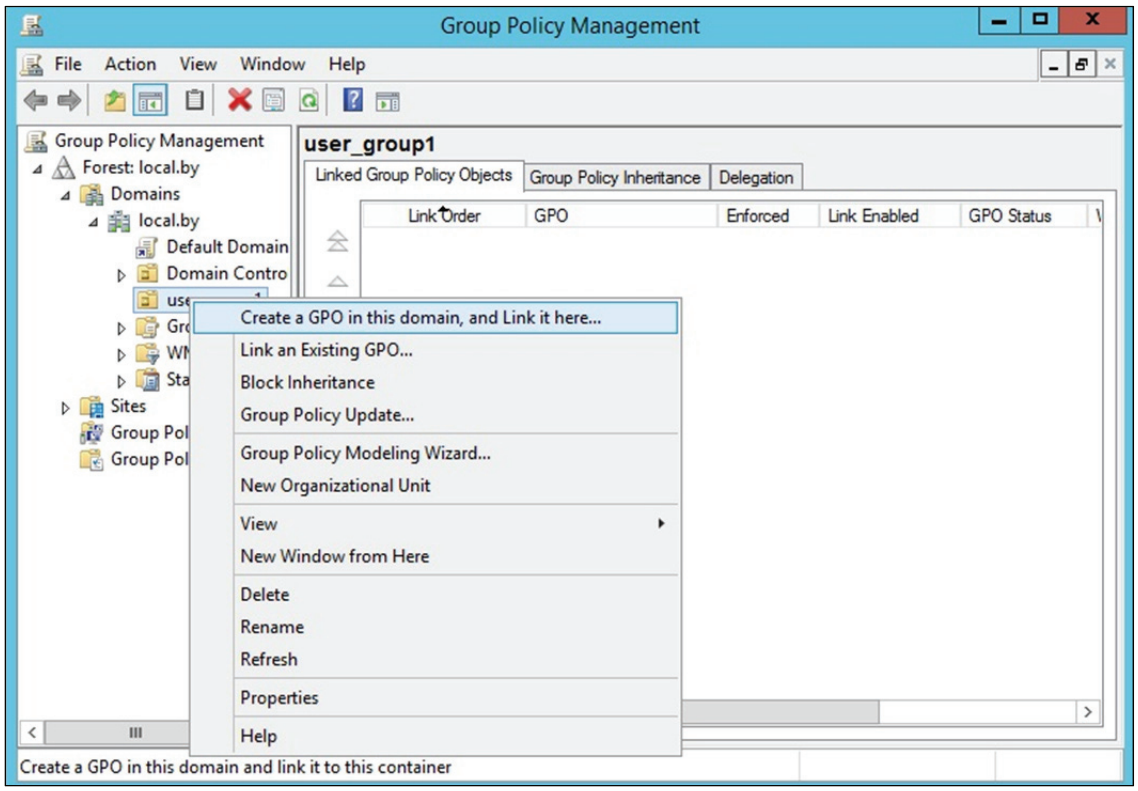


Рис. 6.3. Создание объекта групповой политики
для организационного подразделения

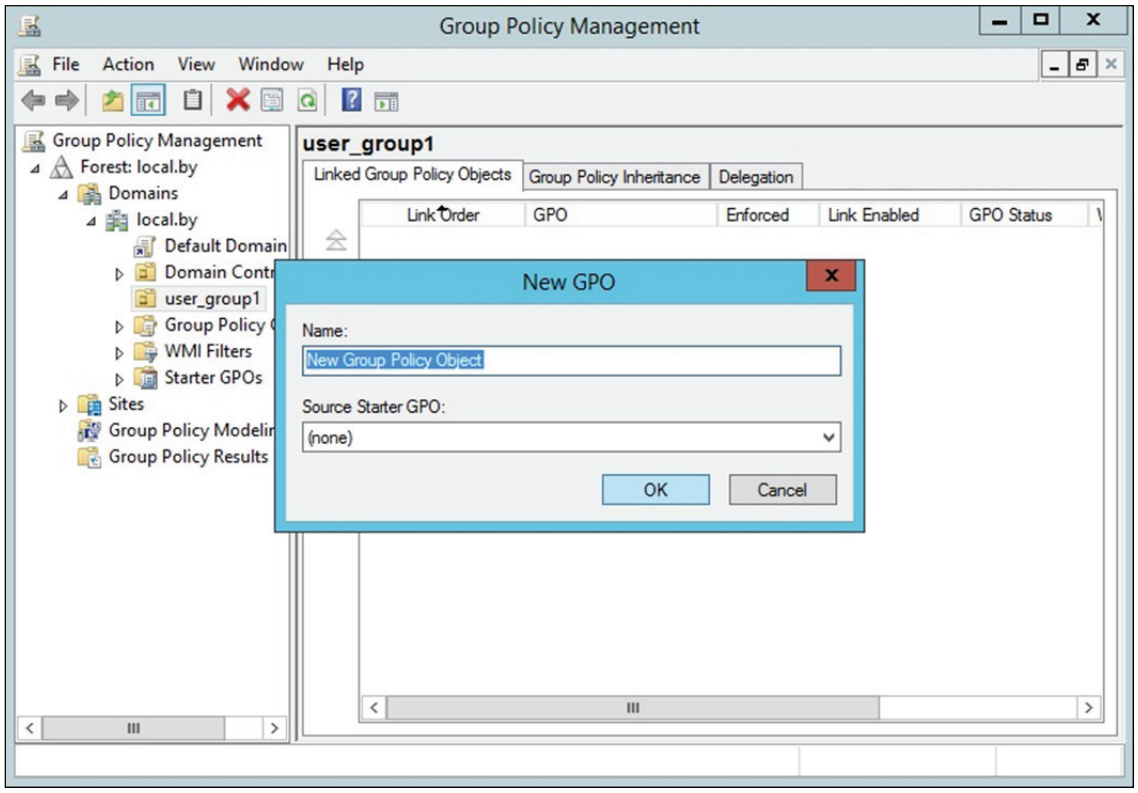


Рис. 6.4. Название объекта групповой политики

Далее фактически необходимо отредактировать (настроить) созданный объект групповой политики (рис. 6.5).

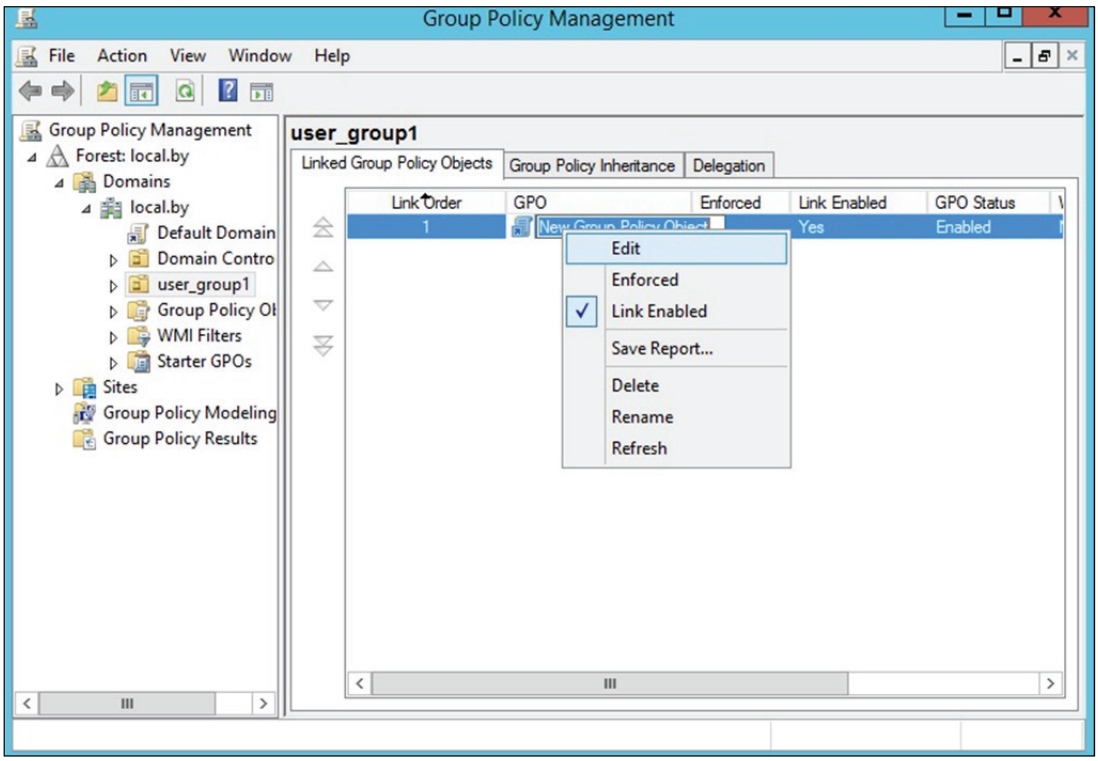


Рис. 6.5. Редактирование (настройка) объекта групповой политики

При выборе команды *Edit* из контекстного меню откроется окно редактора GPO, где мы можем настроить конкретные параметры объекта (рис. 6.6). Отметим, что целесообразно выполнять настройки конфигурации пользователя (user configuration). В таком случае пользователь независимо от места входа в домен (независимо от компьютера) будет получать всегда одни и те же настройки, права, ограничения.

Большинство основных настроек интуитивно понятны (к тому же имеют описание, если открыть соответствующую вкладку), и поэтому не будем подробно останавливаться на каждой. Приведем лишь один пример. Пусть необходимо воспользоваться «белым списком» для запрета запуска всех приложений, кроме тех, что находятся в списке разрешенных. Для этого воспользуемся политикой, показанной на рис. 6.7. При этом данную политику нужно включить и настроить соответствующим образом. На рис. 6.8 (см. на с. 86) приведена настройка запрета запуска всех приложений, кроме mspaint.exe и iexplorer.exe. Корректность настроек можно проверить, войдя под соответствующим пользователем в домен на клиентской машине и попробовав запустить различные приложения.

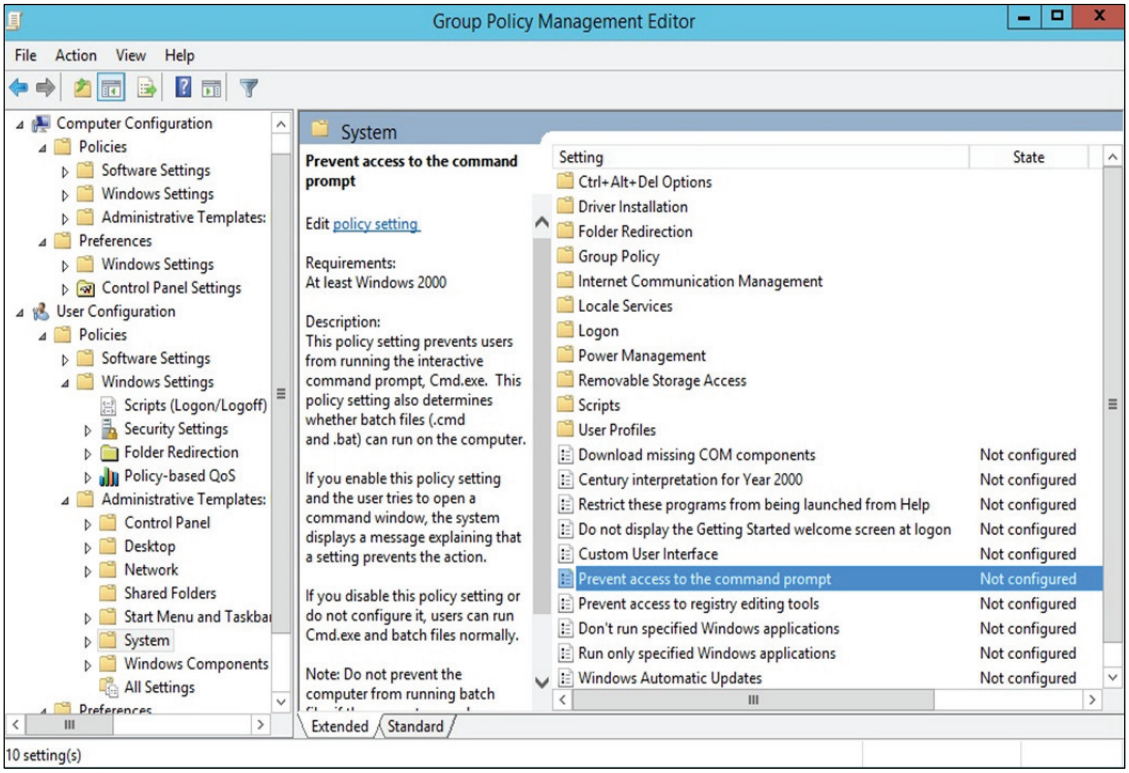


Рис. 6.6. Настройка объекта групповой политики

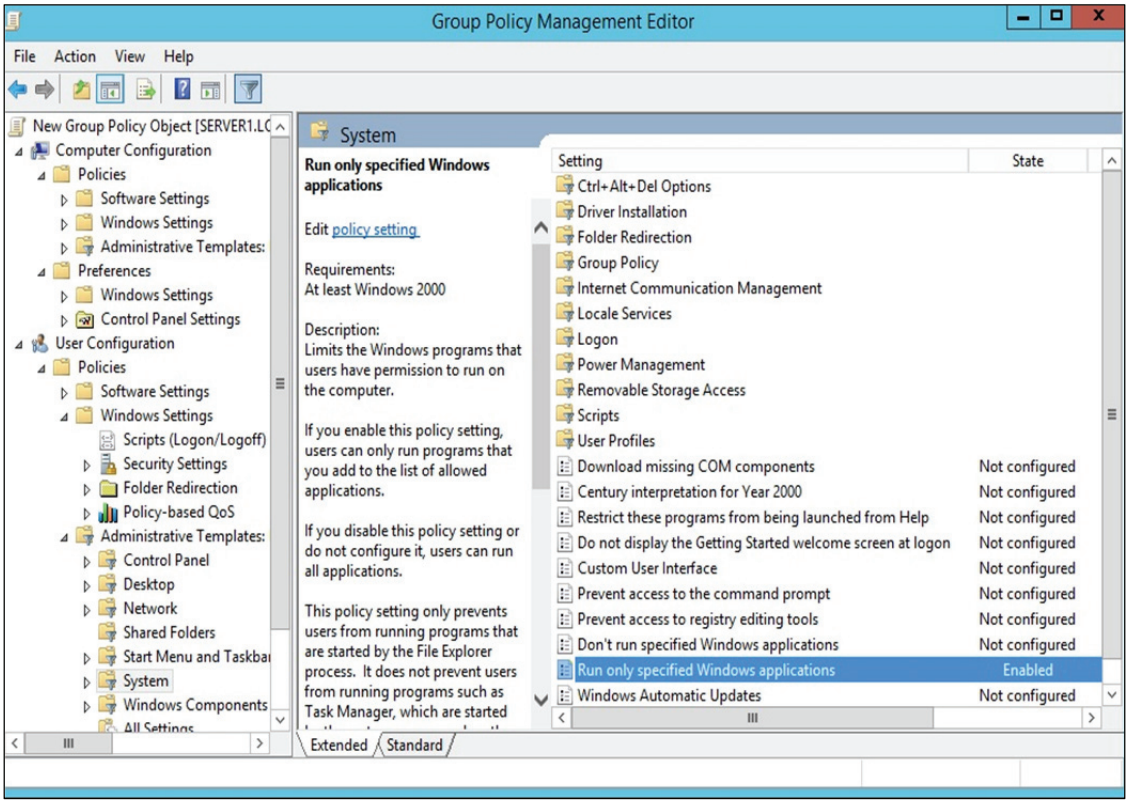


Рис. 6.7. Запрет запуска приложений
с помощью GPO

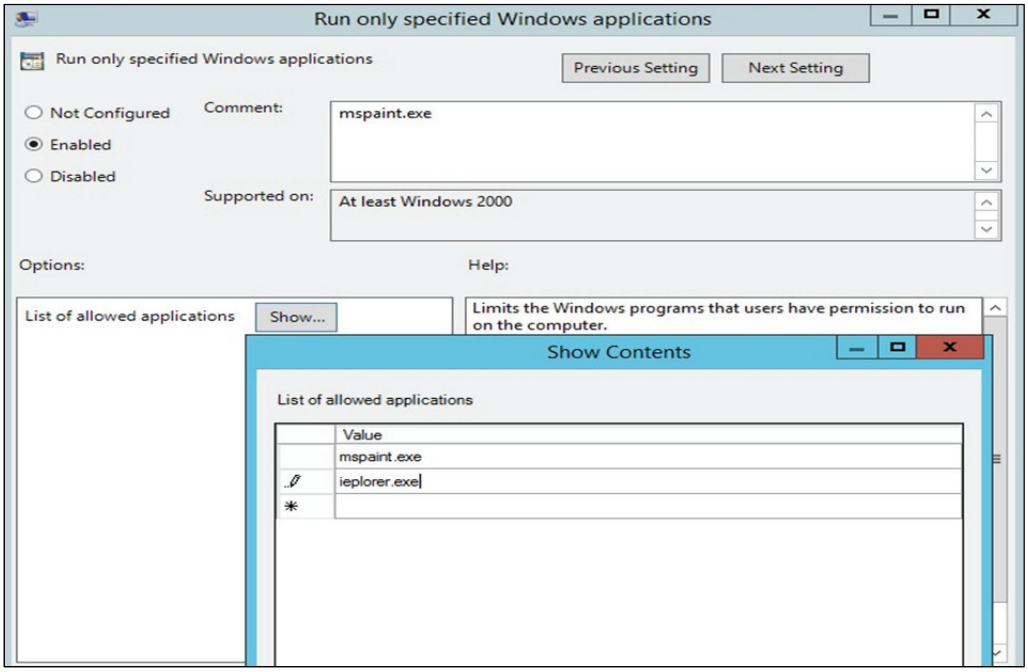


Рис. 6.8. Настройка политики запуска приложений

6.1.3. Порядок применения объектов групповой политики

Когда компьютер запускается, происходят следующие действия:

– читается реестр и определяется, к какому сайту принадлежит компьютер. Делается запрос серверу DNS с целью получения IP-адресов контроллеров домена, расположенных в этом сайте;

– получив адреса, компьютер соединяется с контроллером домена;

– клиент запрашивает список объектов GP у контроллера домена и применяет их. Последний присылает список объектов GP в том порядке, в котором они должны применяться;

– когда пользователь входит в систему, компьютер снова запрашивает список объектов GP, которые необходимо применить к пользователю, извлекает и применяет их.

Групповые политики применяются при загрузке ОС и при входе пользователя в систему. Затем они применяются каждые 90 мин, с вариацией в 30 мин для исключения перегрузки контроллера домена в случае одновременного запроса большого количества клиентов. Для контроллеров домена интервал обновления составляет 15 мин. Изменить это поведение можно в разделе *Computer Configuration Administrative Templates System Group Policy*. Объект групповой политики может действовать только на объекты «компьютер» и «пользователь». Политика действует только на объекты, находящиеся в объекте каталога (сайт, домен, организационное подразделение), с которым связан GPO, и ниже по «дереву» (если не запрещено наследование).

GPO применяются в следующем порядке: локальные политики, политики уровня сайта, политики уровня домена, политики уровня OU.

Групповые политики применяются с некоторыми ОС Windows асинхронно, а с некоторыми синхронно, т. е. пользовательский экран входа появляется только после применения всех политик компьютера, а политики пользователя применяются до того, как появился рабочий стол. Асинхронное применение политик означает, что пользовательский экран входа появляется раньше, чем успевают примениться все политики компьютера, а рабочий стол – раньше, чем применятся все пользовательские политики, что приводит к ускорению загрузки и входа пользователя.

Описанное выше поведение изменяется в двух случаях. Первый – компьютер клиента обнаружил медленное сетевое подключение. Тогда по умолчанию применяются только параметры настройки защиты и административные шаблоны. Медленным считается подключение с пропускной способностью менее 500 Кб/с. Изменить это значение можно в *Computer Configuration → Administrative Templates → System Group Policy → Group Policy slow link detection*. Также в разделе *Computer Configuration → Administrative Templates → System Group Policy* можно настроить некоторые другие параметры политик так, чтобы и они обрабатывались по медленному соединению. Второй способ изменения порядка применения политик основан на опции *User Group policy loopback processing*. Эта опция изменяет порядок применения политик по умолчанию, при котором пользовательские политики применяются после компьютерных и перезаписывают последние. Вы можете установить опцию loopback, чтобы политики компьютера применялись после пользовательских политик и перезаписывали все пользовательские политики, противоречащие политикам компьютера.

У параметра loopback есть два режима:

1) Merge (соединить) – сначала применяется компьютерная политика, затем пользовательская и снова компьютерная. При этом компьютерная политика заменяет противоречащие ей параметры пользовательской политики своими;

2) Replace (заменить) – пользовательская политика не обрабатывается.

Пояснить применение параметра *User Group policy loopback processing* можно, например, на общедоступном компьютере, на котором необходимо иметь одни и те же ограниченные настройки независимо от того, какой пользователь им пользуется.

6.1.4. Приоритетность, наследование и разрешение конфликтов

Как было уже отмечено, на всех уровнях объекты групповой политики содержат одинаковые параметры настройки, и один и тот же параметр может быть определен на нескольких уровнях по-разному. В таком случае действующим значением будет применившееся последним (о порядке применения объектов групповой политики говорилось выше). Это правило распространяется на все параметры, кроме определенных как *not configured*. Для этих параметров Windows не предпринимает никаких действий. Но есть одно исключение: все параметры настройки учетных записей и паролей могут быть определены только на уровне домена, на остальных уровнях эти настройки будут проигнорированы.

Если на одном уровне расположены несколько GPO, то они применяются «снизу вверх» по списку. Изменяя положение объекта политик в списке (стрелочками Up и Down), можно выбрать необходимый порядок применения (рис. 6.9).

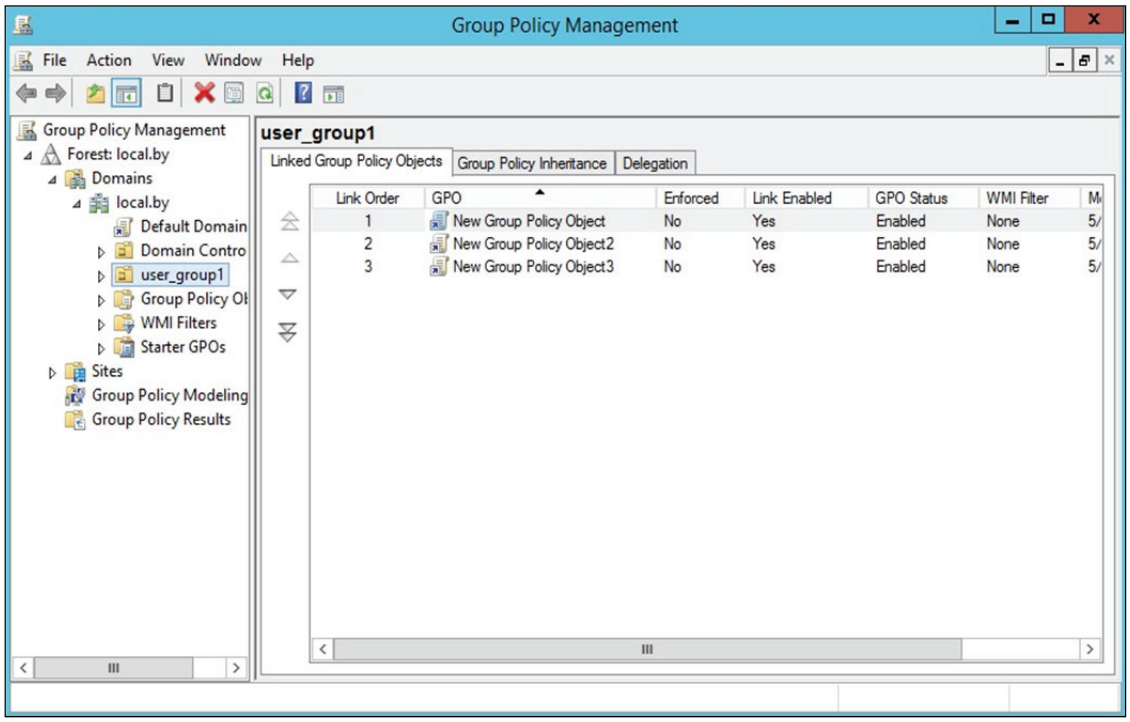


Рис. 6.9. Порядок применения политик

Отметим, что при помощи дополнительных параметров GPO можно сделать так, чтобы определенное OU не получало параметры политик от GPO, связанных с вышестоящими контейнерами. Фактически необходимо заблокировать наследование политик. При этом будут блокироваться все наследуемые параметры политик, и нет способа

блокировать отдельные параметры. Параметры настройки уровня домена, определяющие политику паролей и политику учетных записей, не могут быть заблокированы.

Лабораторная работа № 9

***Цель*:** изучение методов удаленного администрирования с использованием групповых политик.

***Задание*:** используя возможности Active Directory и групповых политик, необходимо создать два организационных подразделения в рамках домена.

1. Первому организационному подразделению (с названием, например, power_users) запрещено:

– изменять конфигурацию IP-протокола;

– создавать, удалять и изменять настройки пользователей (например, пароль);

– устанавливать/удалять приложения;

– редактировать реестр.

2. Второму организационному подразделению (с названием, например, limited_users) запрещено:

– изменять конфигурацию IP-протокола;

– запускать диспетчер задач;

– запускать управление компьютером (computer managment);

– запускать апплеты панели управления;

– изменять настройки Internet Explorer;

– изменять настройки рабочего стола;

– использовать командную строку;

– создавать, удалять и изменять настройки пользователей (например, пароль);

– устанавливать/удалять приложения;

– редактировать реестр;

– запускать какие-либо приложения, кроме тех, что в списке (список придумать самостоятельно и согласовать с преподавателем, например, проводник, Internet Explorer), т. е. запрет в данном случае необходимо организовать по принципу «белого списка».

По желанию можно выполнить настройку и иных запретов/разрешений, например, скрыть системные (локальные) диски клиентской ОС и т. д.

Отметим, что ограничения могут быть выполнены как за счет применения групповых политик, так и за счет принадлежности пользователя к определенной группе пользователей.

6.2. Удаленный рабочий стол

Подключение к удаленному рабочему столу. Клиентское приложение, используемое для подключения к серверу в контексте режима *Дистанционное управление рабочим столом* (*Remote Desktop*) или *Сервер терминалов* (*Terminal Server*), называется *Подключение к удаленному рабочему столу* (*Remote Desktop Connection*). Для клиента нет функциональных различий между этими двумя конфигурациями сервера.

На компьютерах с Windows XP и старше, а также Windows Server программа *Подключение к удаленному рабочему столу* установлена по умолчанию, но «спрятана»: *Пуск* (*Start*) → *Все программы* (*All Programms*) → *Стандартные* (*Accessories*) → *Связь* (*Communications*) → *Подключение к удаленному рабочему столу* (*Remote Desktop Connection*).

На других платформах программу *Подключение к удаленному рабочему столу* можно установить с компакт-диска Windows Server либо из установочной папки клиента (%Systemroot%\System32\Clients\Tsclient\Win32) на любом из компьютеров под управлением Windows Server. Установочный пакет msi можно распространять на системы Windows с помощью групповой политики (будет рассмотрена в подразделе 6.3).

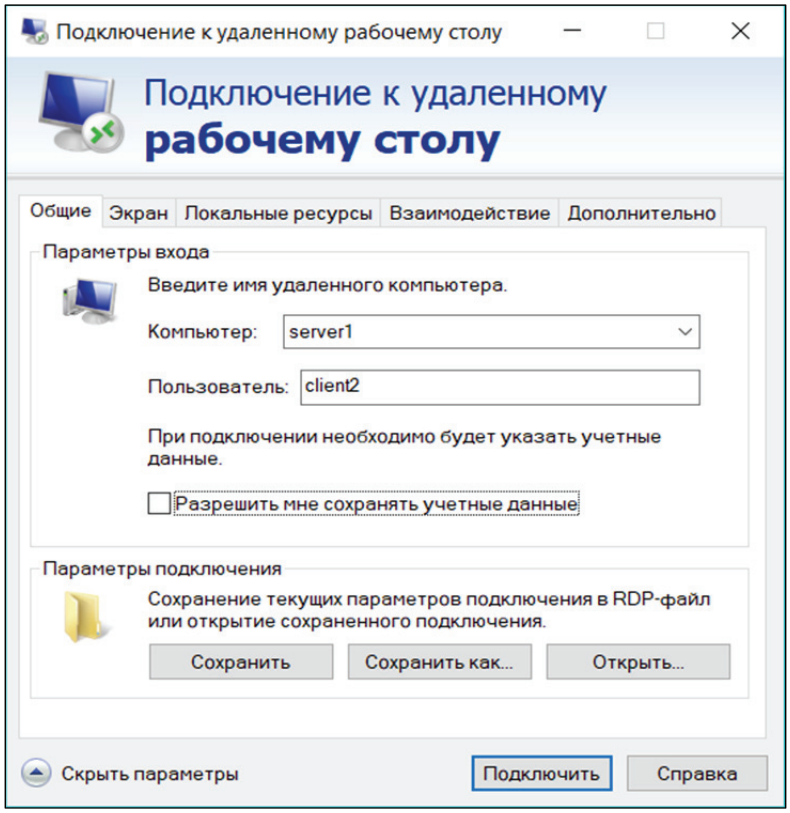


Рис. 6.10. Подключение к удаленному рабочему столу

На рис. 6.10 показан клиент программы *Дистанционное подключение к рабочему столу* (*Remote Desktop*), настроенный для подключения под пользователем client2 к серверу с именем server1 (имя компьютера может быть другим) в домене local.by (имя домена также может быть другим).

Отметим, что до подключения на стороне сервера должны быть разрешены подобные операции, а также определены пользователи, кому можно выполнять удаленное подключение (рис. 6.11).

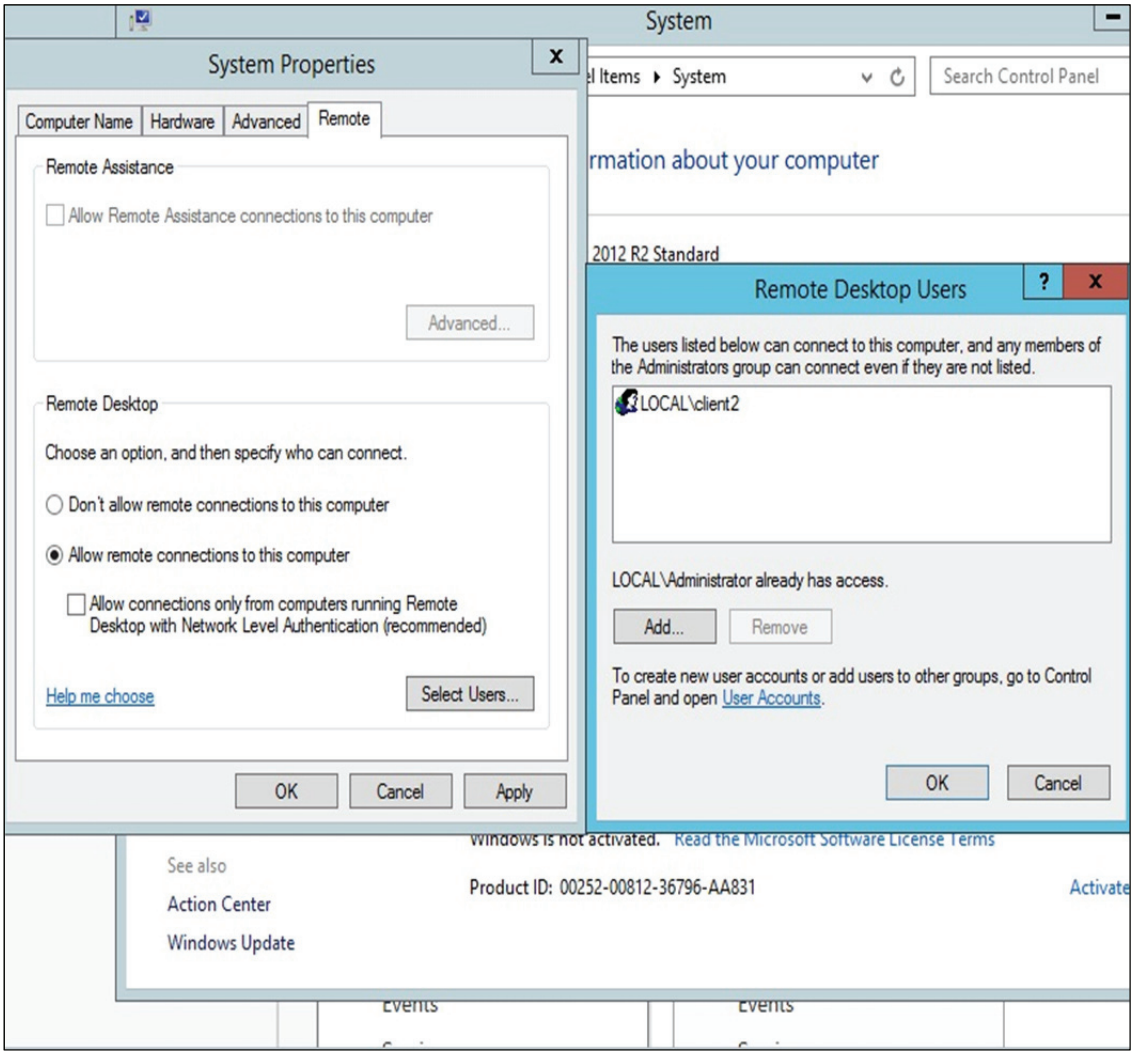


Рис. 6.11. Настройки, разрешающие подключение
по удаленному рабочему столу

Настроив клиента удаленного подключения к рабочему столу и сервер, вы сможете управлять множеством аспектов дистанционного подключения как со стороны клиента, так и со стороны сервера. В следующей таблице перечислены конфигурационные параметры и их назначение.

Параметры программы *Удаленное подключение к рабочему столу*

Параметры	Назначение
Параметры клиента	
Общие (General)	Параметры выбора компьютера, к которому необходимо подключаться, настройка статических реквизитов для входа в систему, а также сохранение параметров для данного подключения
Экран (Display)	Задает размер окна клиента, глубину цвета, а также доступность панели подключений при работе в полноэкранном режиме
Локальные ресурсы (Local Resources)	Параметры передачи звуковых событий на локальный компьютер, помимо стандартных выходных сигналов мыши, клавиатуры и экрана. Также параметры на этой вкладке определяют, как удаленный компьютер интерпретирует комбинации клавиш Windows (например, Alt + Tab), и доступны ли в сеансе удаленного доступа такие устройства, как локальные диски, принтеры и последовательные порты
Программы (Programs)	Задают путь и папки расположения для любых программ, которые необходимо запустить после установки соединения
Дополнительно (Experience)	Категории функций экрана можно включать или отключать в зависимости от пропускной способности канала связи между локальным и удаленными компьютерами. Предусмотрены параметры для отображения фона рабочего стола, содержимого окна при перетаскивании, визуальных эффектов при прорисовке меню и окон, тем рабочего стола; также вы можете активировать режим кеширования растровой графики, при котором после каждого интервала обновления передаются только изменения, а не весь экран целиком
Параметры сервера	
Параметры входа (Logon Settings)	Позволяют задать статические реквизиты для подключения вместо реквизитов, предоставляемых клиентом
Сеансы (Sessions)	Чтобы перекрыть настройки клиента, необходимо задать здесь параметры завершения прерванного сеанса, ограничения длительности сеанса и времени его простоя, а также допустимость повторного подключения
Среда (Environment)	Перекрывает настройки из профиля пользователя для данного подключения в отношении запуска программы: здесь вы можете переопределить запускаемую при подключении программу. Заданный здесь путь и папка запуска приоритетнее настроек, сделанных программой *Подключение к удаленному рабочему столу*
Разрешения (Permissions)	Позволяют задавать дополнительные разрешения для данного подключения

Окончание таблицы

Параметры	Назначение
Параметры сервера	
Удаленное управление (Remote Control)	Указывает, можно ли удаленно управлять сеансом *Подключение к удаленному рабочему столу*, и если так, то должен ли пользователь выдавать разрешение на инициализацию сеанса удаленного управления. Дополнительные параметры позволяют ограничить сеанс удаленного управления только функцией просмотра либо разрешить полную интерактивность с сеансом клиента *Дистанционное управление рабочим столом*
Параметры клиента (Client Settings)	Позволяют перекрыть параметры из конфигурации клиента, изменить глубину цвета и отключить различные коммуникационные порты (порты ввода-вывода)
Сетевой адаптер (Network Adapter)	Указывает, какие сетевые платы на сервере будут принимать удаленные подключения для администрирования
Общие (General)	Задают уровень шифрования и механизм проверки подлинности для подключений к этому серверу

Устранение неполадок при работе со службами терминалов. При использовании программы *Удаленный рабочий стол для администрирования* (*Remote Desktop for Administration*) создается подключение к консоли сервера. Есть несколько потенциальных причин неудачных подключений или сеансов с ошибками:

– сбои сети. Ошибки в работе стандартной TCP/IP-сети могут вызывать сбои или разрывы подключений *Дистанционное подключение к рабочему столу* (*Remote Desktop*). Если не функционирует служба DNS, клиент не сможет найти сервер по имени. Если не функционирует маршрутизация либо неверно настроен порт *Служб терминалов* (*Terminal Services*) (по умолчанию это порт 3389) на клиенте или сервере, соединение установить не удастся;

– реквизиты входа. Для успешного подключения к серверу средствами программы *Удаленный рабочий стол для администрирования* (*Remote Desktop for Administration*) пользователи должны быть включены в группу *Администраторы* (*Administrators*) или *Пользователи удаленного рабочего стола* (*Remote Desktop Users*);

– политика. Только администраторам разрешено подключаться средствами программы *Дистанционное подключение к рабочему столу* (*Remote Desktop*) к контроллерам доменов. Чтобы разрешить подключаться остальным пользователям, нужно настроить политику безопасности на контроллере домена;

– слишком много параллельных подключений. Если сеансы прерывались без выхода из системы, сервер может посчитать, что достигнут

предел одновременно обрабатываемых подключений, даже если в данный момент к серверу не подключены два пользователя. Например, администратор может завершить сеанс без выхода из системы. Если еще два администратора попытаются подключиться к серверу, это удастся только одному из них.

Управление настройками удаленного подключения осуществляется через консоль службы терминалов, которая устанавливается как роль сервера (*Remote Desktop Services* – компонент *Remote Desktop Licensing*) (рис. 6.12).

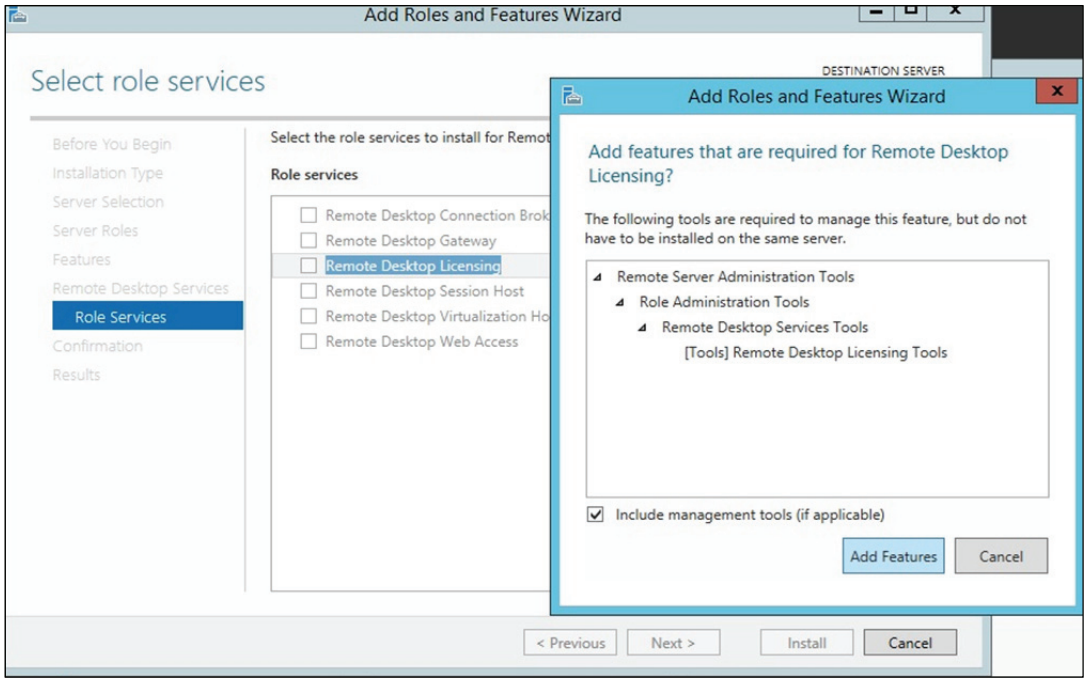


Рис. 6.12. Установка службы терминалов

Наиболее интересным является настройка следующих параметров подключения к рабочему столу.

1. На вкладке *Сетевой адаптер* (*Network Adapter*) выбираем значение параметра *Максимальное число подключений* (*Maximum Connections*) равным 1.

2. На вкладке *Сеансы* (*Sessions*) устанавливаем оба флажка *Заменить параметры пользователя* (*Override User Settings*) и изменяем настройки следующим образом: все прерванные любыми способами (или по любой причине) сеансы пользователей закрываются через 15 мин, активный сеанс не ограничивается по времени, сеансы завершаются после 15 мин бездействия:

− завершение отключенного сеанса (*End a disconnected session*) – 15 мин;

− ограничение активного сеанса (*Active session limit*) – никогда (never);

− ограничение пассивного сеанса (*Passive session limit*) – 15 мин;

− при превышении ограничений или разрыве подключения (*When session limit is reached or connection is broken*) – отключить сеанс (*Disconnect from session*).

Такая конфигурация обеспечивает следующее: только один пользователь одновременно подключен к серверу терминалов, любой прерванный сеанс закроется через 15 мин и неактивный сеанс прервется через 15 мин. Эти параметры позволяют избежать ситуации, когда прерванный или бездействующий сеанс мешает подключаться средствами программы *Удаленный рабочий стол для администрирования* (*Remote Desktop for Administration*).

Для подключения к серверу с помощью клиента удаленного подключения к рабочему столу выполним следующие операции.

1. На другом удаленном компьютере (виртуальной машине), в группе *Стандартные* → *Связь* (*Accessories* → *Communications*) щелкаем *Подключение к удаленному рабочему столу* (*Remote Desktop Connection*), подключаемся к Server1 и входим в его систему.

2. На сервере Server1 открываем консоль tscc.msc: *Администрирование* (*Administrative tools*) → *Настройка служб терминалов* (*Terminal Services Configuration*). В открывшейся консоли выбираем *Подключения* (*Connections*). Вы должны увидеть сведения о сеансе удаленного подключения к Server01.

3. Не выполняем никаких действий в этом сеансе 15 мин либо закрываем клиент программы *Удаленное подключение к рабочему столу* (*Remote Desktop*), не завершив сеанс *Сервера терминалов* (*Terminal Server*) явно: сеанс должен будет завершиться автоматически через 15 мин.

В данный момент вы подключены к Server1 удаленно и можете выполнять на нем любые задачи, допустимые в интерактивном режиме на консоли.

Лабораторная работа № 10

***Цель*:** изучение методов удаленного администрирования с помощью подключения к удаленному рабочему столу.

***Задание*:** необходимо настроить удаленное подключение к рабочему столу и выполнить подключение к серверу с помощью клиентской виртуальной машины либо другого сервера. Следует изучить

параметры подключения (число подключений, время отключения при бездействии пользователя). Необходимо выполнить подключение к компьютеру с правами администратора (т. е. с полным доступом) и с правами пользователя (т. е. с ограниченным доступом). Ограничения следует определить самостоятельно.

6.3. Удаленная установка программного обеспечения

В Active Directory групповые политики позволяют распространять программное обеспечение пользователям и компьютерам, используя переупаковывающий файловый формат .msi. Когда приложение распространяется через групповую политику, пользователю не требуется специальных прав, так как приложение устанавливается при повышенных привилегиях самой политики. Если производитель не предоставляет файл .msi для своего приложения, то можно использовать специальную переупаковывающую программу для его создания. Второй важный момент при распространении программ через групповые политики – это то, как мы их распространяем. Есть две возможности – либо Assign (назначить), либо Publish (опубликовать) их. Программы могут быть как опубликованы, так и назначены пользователям. В случае их назначения приложение начинает «следовать» за пользователем независимо от того, на каком компьютере он входит в сеть. Иконка программы появляется в стартовом меню, но программа не устанавливается до тех пор, пока пользователь не «кликнет» по иконке. Когда программа назначается компьютеру, она устанавливается на компьютер при его следующей перезагрузке и становится доступной всем пользователям этого компьютера. Когда программа публикуется (что может быть сделано только для пользователей, но не для компьютеров), она становится доступной для установки при помощи программы *Add/Remove Programs* или при обращении к соответствующему документу (когда пользователь «кликнет» по документу, формат которого ассоциируется с этой программой). Опубликование программы делает ее доступной для пользователей, но у вас не должно создаться иллюзии, что она уже является установленной.

Приложение может быть также опубликовано с использованием файла с расширением .zap, если нет файла .msi или его невозможно создать. Отметим, однако, что при использовании файла .zap у пользователя должен быть соответствующий уровень привилегий, достаточный для установки приложения. Также подчеркнем, что внедрение

программного обеспечения через групповые политики доступно только для систем с OS Windows XP и старше.

Раздел *Software Setting* (*Установка программного обеспечения*) групповой политики, где и производится назначение или опубликование программ, показан на рис. 6.13.

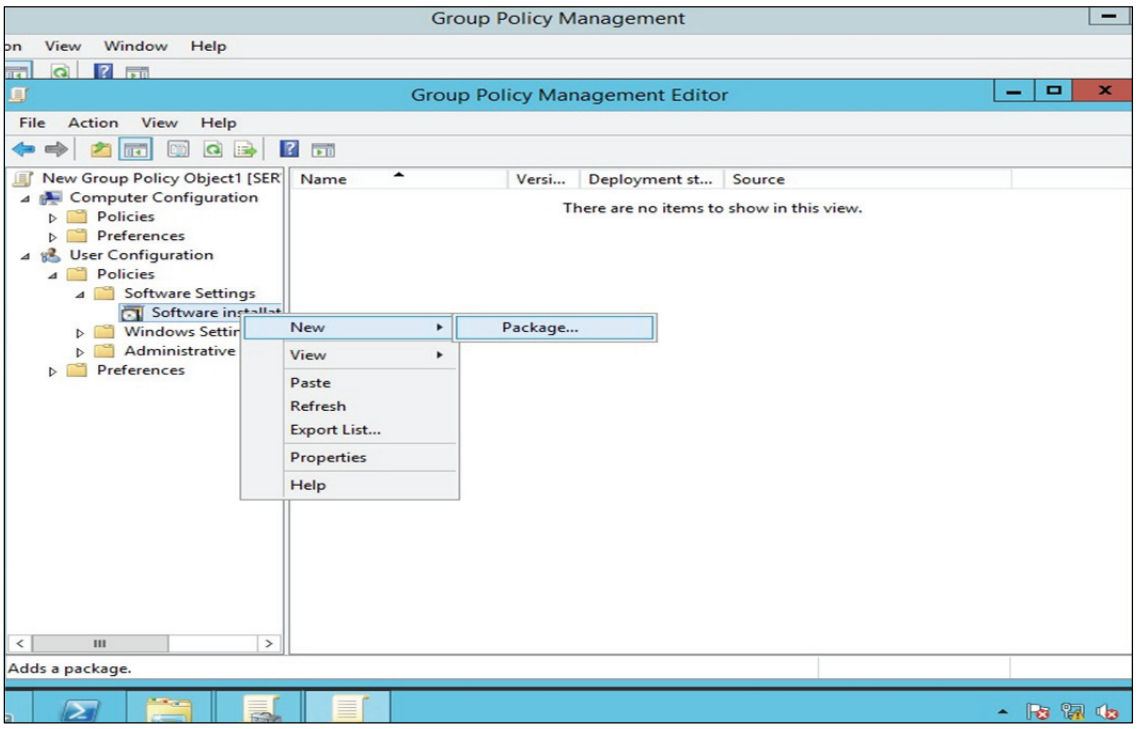


Рис. 6.13. Интерфейс консоли для организации удаленной установки ПО

Когда приложение внедряется через групповую политику, важно сам дистрибутив расположить в сетевой папке и в процессе настройки указать именно сетевой путь (см. рис. 6.14 на с. 99). Также должна быть выбрана следующая опция – будет ли ваше приложение опубликовано (Published) или назначено (Assigned) (см. рис. 6.15 на с. 99).

Дополнительные свойства для внедрения программного обеспечения могут быть настроены сразу, если вы выбираете опцию *Advanced published or assigned* (*Дополнительные настройки опубликования или назначения*), или позднее путем изменения свойств внедряемого пакета. Дополнительные свойства позволяют вам контролировать многие параметры, имеющие отношение к распространяемому приложению, включая такие, как добавление обновлений и патчей, модификация, а также удаление пакетов.

Есть шесть вкладок дополнительных свойств внедряемого приложения, и вы должны быть хорошо знакомы с ними. Вкладка *General* (*Общая*) содержит основную информацию об объекте (такую как

номер версии и т. д.) (см. рис. 6.16 на с. 100), в то время как вкладка *Security* (*Безопасность*) содержит ACL объекта (см. рис. 6.17 на с. 100). Вкладка *Deployment* (*Внедрение*) (см. рис. 6.18 на с. 101) контролирует, было ли приложение опубликовано или назначено (эта настройка может быть изменена). Если опубликовано, вы можете контролировать, будет или нет приложение устанавливаться при обращении к файлам, ассоциирующимся с данным приложением (эта опция будет «залита» серым, если вы выбрали *Назначить приложение*).

Заметьте, что существует опция *Uninstall the application when it falls out of the scope of management* (*Удалять приложение, если оно выходит из сферы управления*). Если она выбрана и групповая политика, которая установила это приложение, больше не применяется (например, если объекты «пользователь» или «компьютер» были перемещены), тогда приложение будет автоматически удалено. Опция *Installation user interface options* (*Установка опций пользовательского интерфейса*) позволяет вам контролировать, как много взаимодействий пользователь будет иметь в процессе установки.

Вкладка *Upgrades* (*Обновления*), изображенная на рис. 6.19 (см. на с. 101), позволяет автоматизировать установку патчей и обновлений (таких как новейшие версии) в приложения, которые уже внедрены через групповую политику. Если обновление должно выполняться в обязательном порядке, выбирается опция *Required* (*Обязательный*), и тогда обновление внедряется сразу и пользователь сможет использовать только новую версию приложения. Если это не обязательное требование, тогда пользователь может использовать как старую, так и новую версию. Это может быть потенциально полезным, если новое приложение не имеет обратной совместимости (не работает с документами, созданными в старой версии программы).

Вкладка *Categories* (*Категории*) позволяет контролировать то, каким образом приложение будет представлено в программе *Add*/*Remove Programs* (см. рис. 6.20 на с. 102). Например, вы можете создать категории для каждого типа приложений, таких как графические приложения, программы для работы с текстом и т. д. Эта вкладка позволит вам группировать вновь публикуемые приложения в эти категории для того, чтобы упростить пользователю процесс выбора необходимых ему программ.

И, наконец, вкладка *Modifications* (*Изменения*) позволяет выполнять дальнейшую настройку пакета для пользователей со специфическими потребностями (см. рис. 6.21 на с. 102). Например, вы хотите внедрить различающиеся по языку словари для пользователей в разных офисах и применяете модифицирование пакета. Модифицирование выполняется

в виде файла с расширением .mst (также известном, как файл трансформации). Есть специальная утилита для создания файлов .mst, которая содержится в наборе инструментов (kit resource) Microsoft Office.

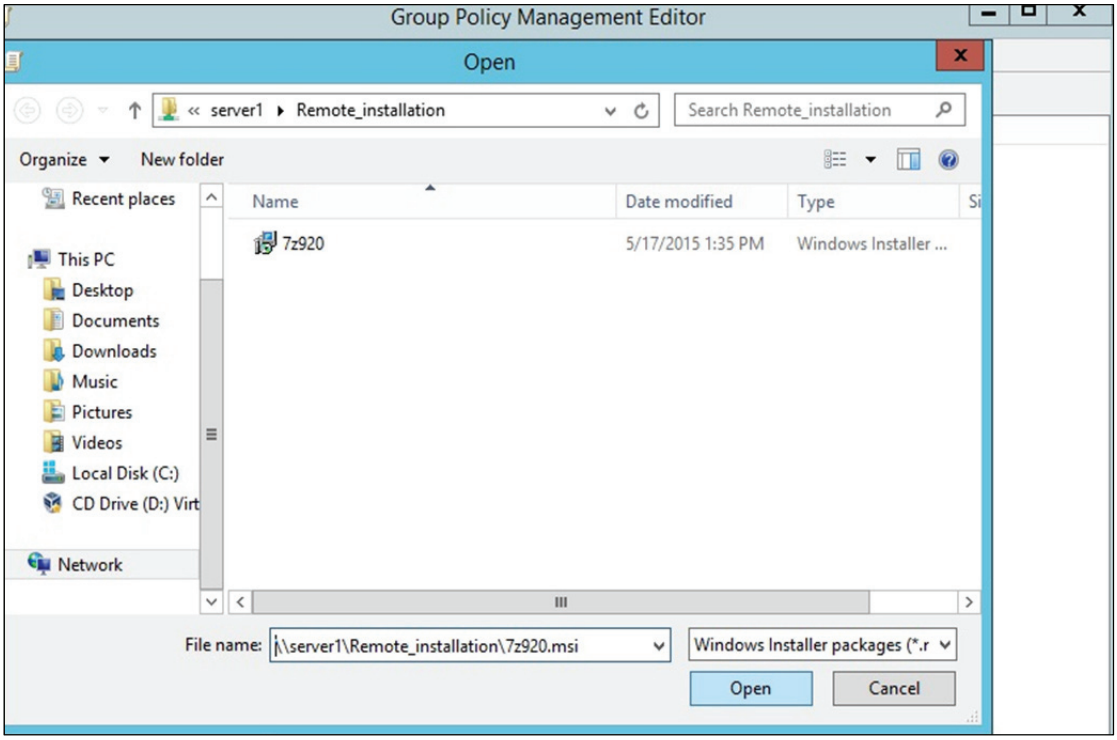


Рис. 6.14. Определение сетевого пути
к папке с дистрибутивом устанавливаемого приложения

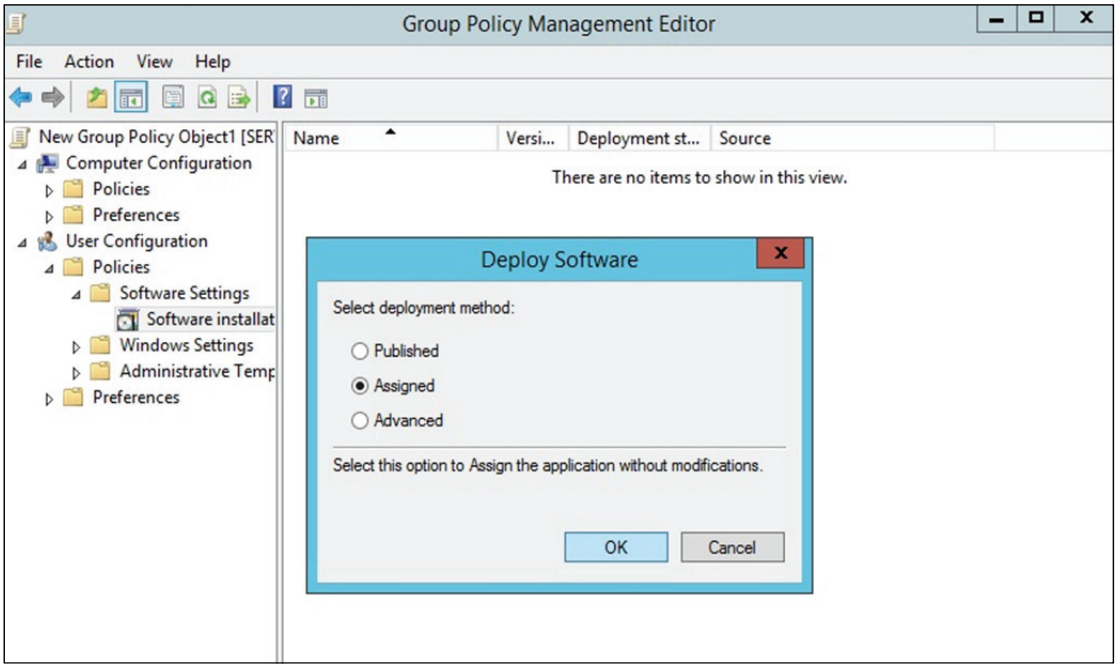


Рис. 6.15. Выбор типа публикации приложения

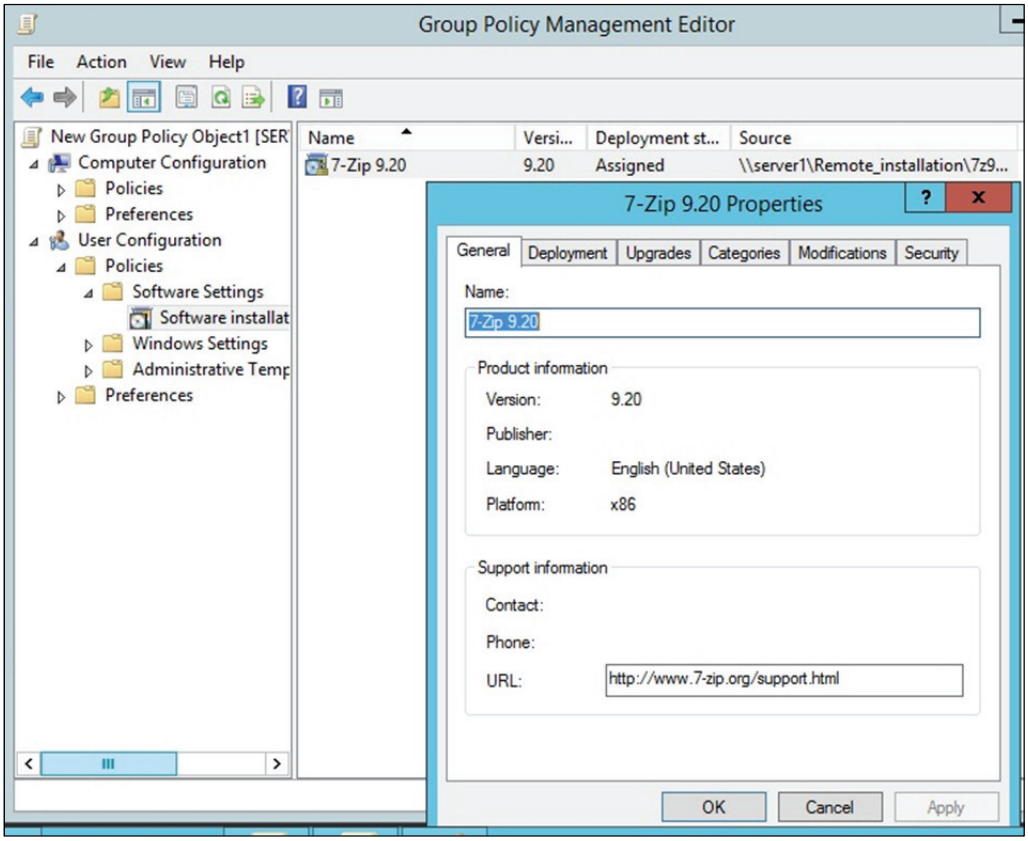


Рис. 6.16. Вкладка *General* в опциях удаленной установки ПО

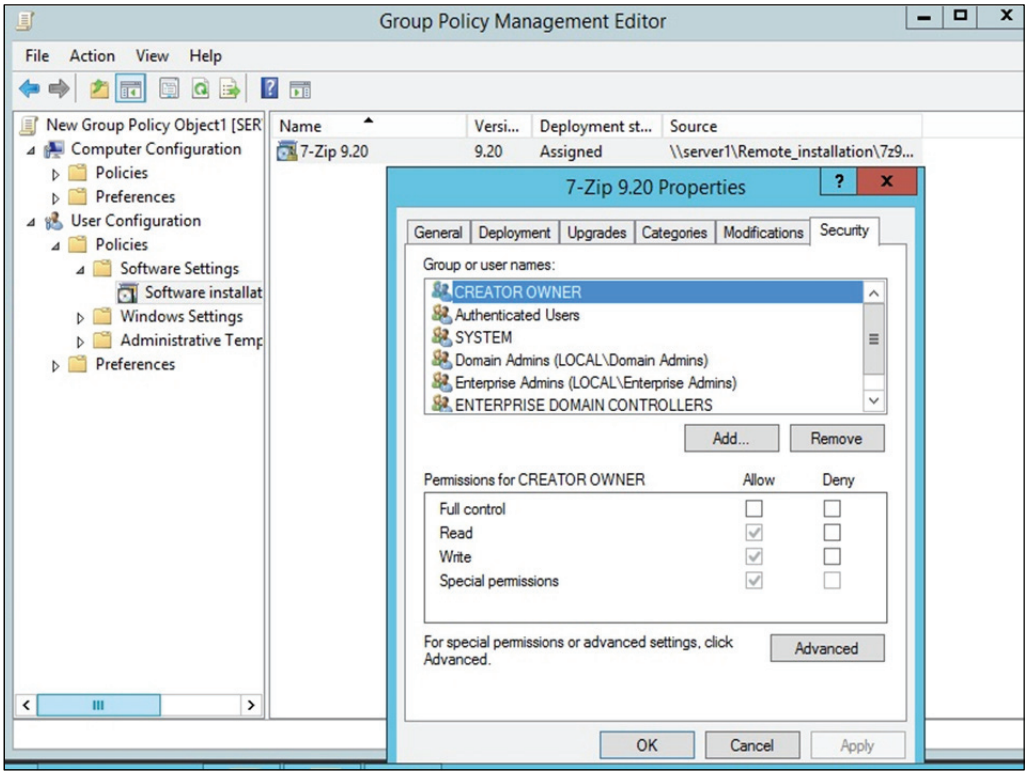


Рис. 6.17. Вкладка *Security* в опциях удаленной установки ПО

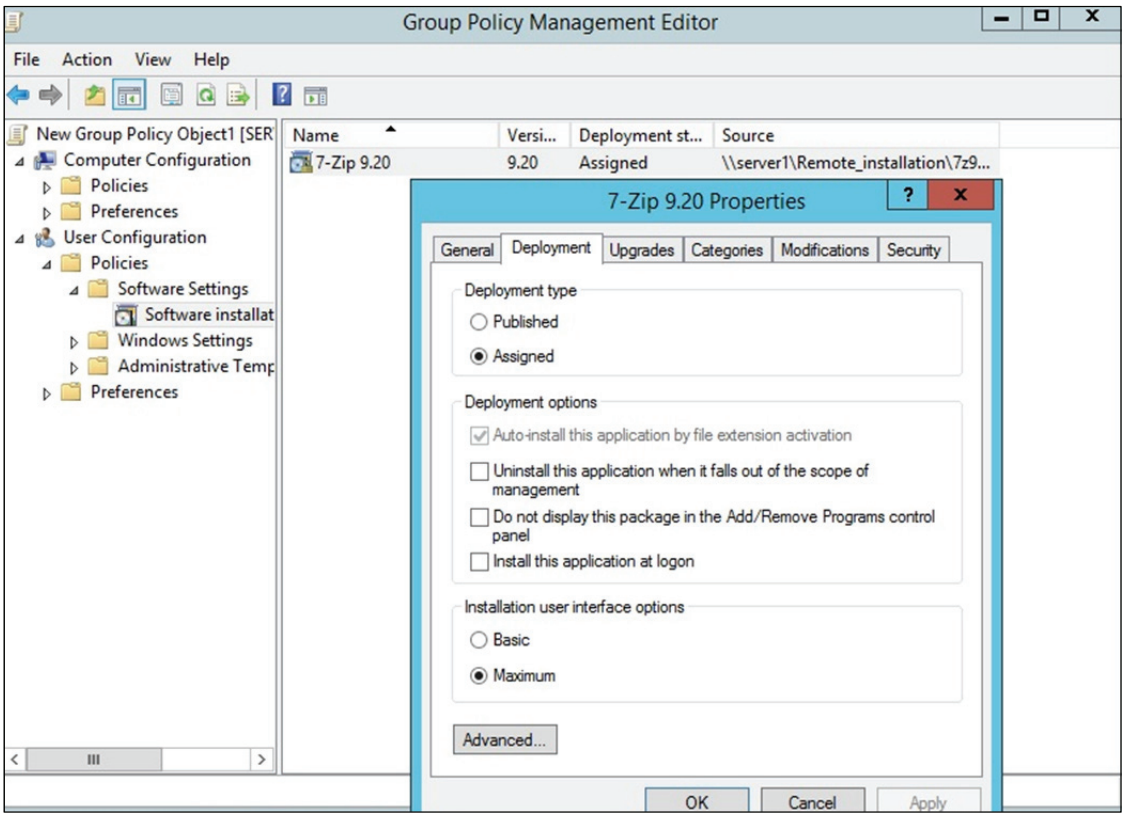


Рис. 6.18. Вкладка *Deployment* в опциях удаленной установки ПО

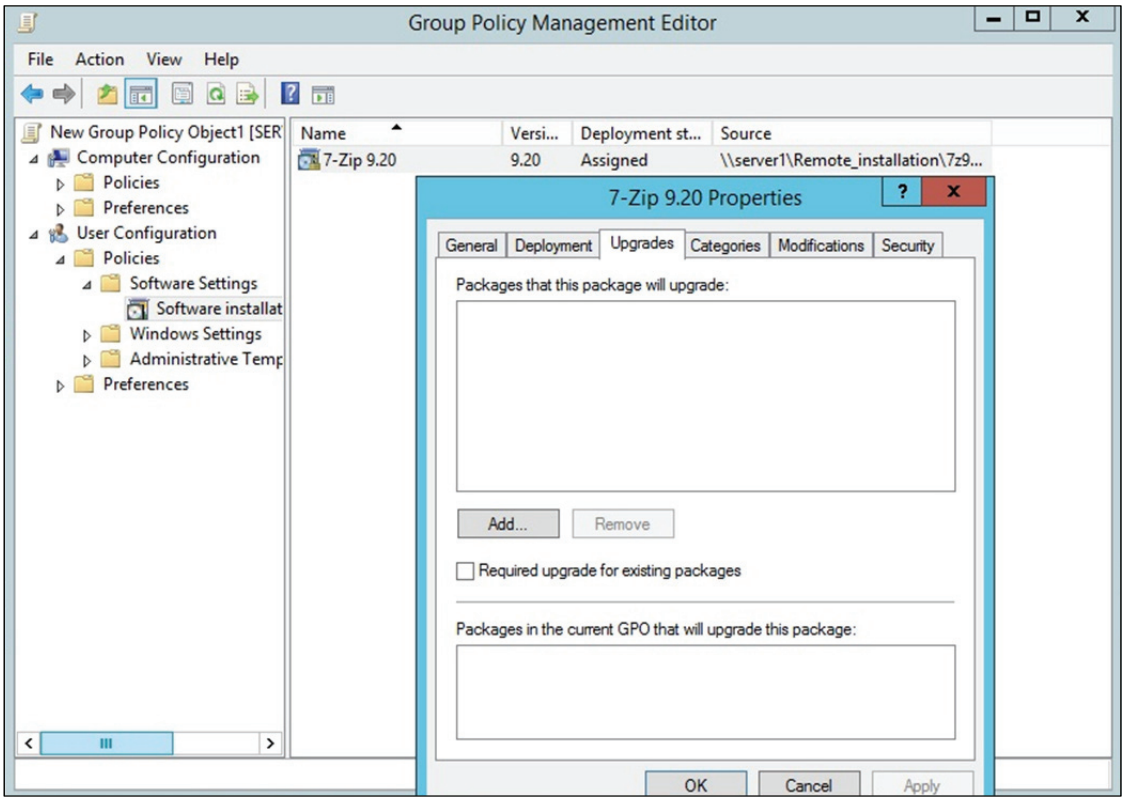


Рис. 6.19. Вкладка *Upgrades* в опциях удаленной установки ПО

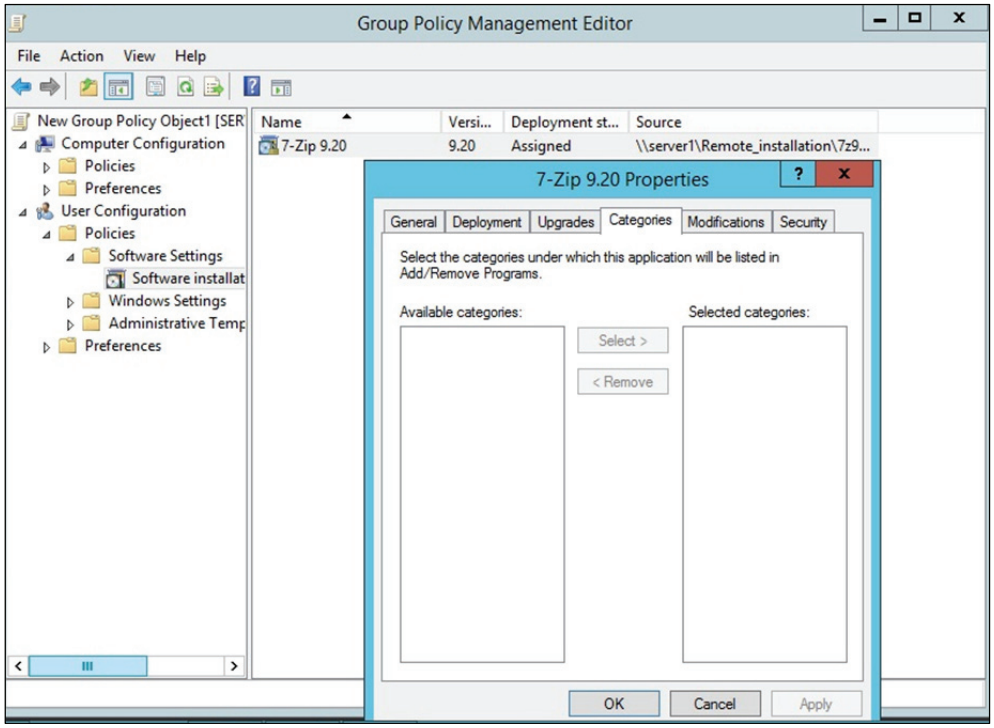


Рис. 6.20. Вкладка *Categories* в опциях удаленной установки ПО

Необходимо отметить, что дистрибутив устанавливаемого приложения должен находиться в папке, открытой для доступа по сети для соответствующего пользователя.

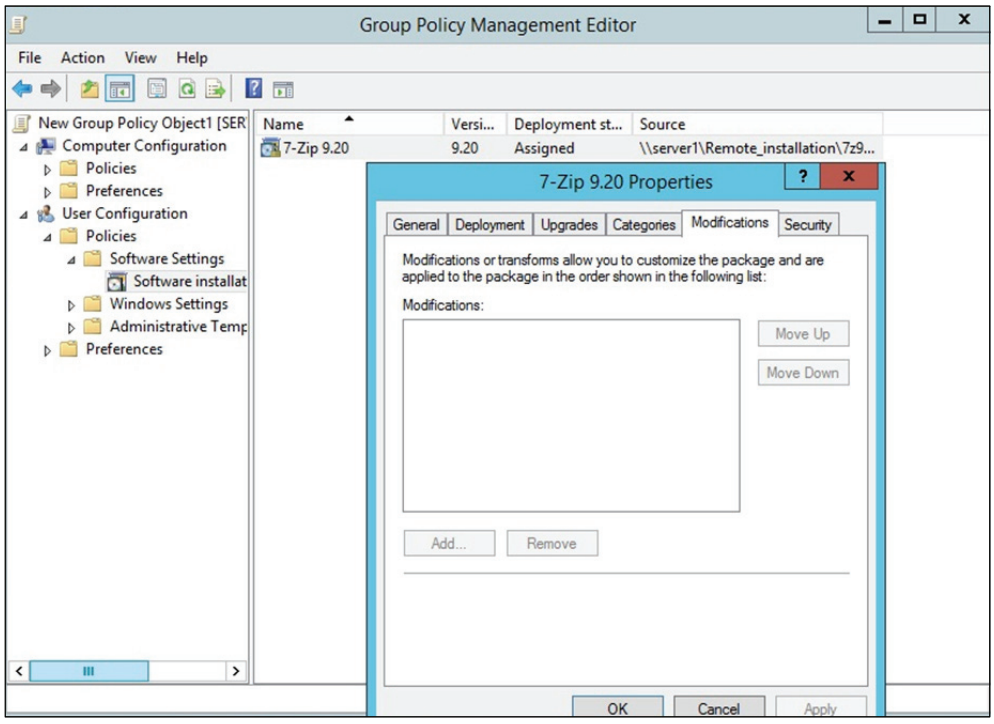


Рис. 6.21. Вкладка *Modifications* в опциях удаленной установки ПО

Лабораторная работа № 11

***Цель*:** изучение методов удаленного администрирования (удаленная установка программного обеспечения).

***Задание*:** необходимо удаленно (с помощью групповых политик) установить клиенту программное обеспечение (например, Skype, The Bat! и т. д.). Операции следует выполнить таким образом, что при первой загрузке пользователем приложение было автоматически доустановлено и, соответственно, пользователь смог им воспользоваться. Установленное приложение необходимо занести в список разрешенных приложений, сделанный в лабораторной работе № 9 (удаленное администрирование с помощью групповых политик).

Раздел 7
БЕЗОПАСНОСТЬ ДОМЕННЫХ СИСТЕМ

7.1. Мониторинг и устранение неполадок подключений TCP/IP. Прослушивание сетевого трафика

В данном подразделе рассказывается о популярных инструментах устранения неполадок протокола IP. Вы узнаете о *Сетевом мониторе* (*Network Monitor*), анализаторе протоколов, служащим для покадрового анализа сетевого трафика. Сетевые администраторы применяют анализаторы протоколов для выяснения, почему не работает механизм разрешения имен или почему сбоят подключения к сетевым ресурсам. Иначе говоря, без такого анализатора протокола, как сетевой монитор, очень сложно узнать, что в действительности происходит с сетью.

Также здесь рассматриваются инструментальные средства, чаще всего используемые для устранения неполадок связи в сети. Некоторыми из этих средств (такими как *Ipconfig* и *Ping*) администраторы пользуются ежедневно, если не ежечасно. Другие инструментальные средства, например *Диагностика сети* (*Network Diagnostics*), предоставляют больше информации и позволяют более основательно подходить к устранению неполадок связи в сети.

7.1.1. Анализ сетевого трафика средствами *Сетевого монитора*

Для наблюдения за сетевым трафиком используется анализатор протоколов, например, *Сетевой монитор* (*Network Monitor*). В версиях Windows 2003 и 2008 он устанавливается с помощью *Мастера компонентов Windows* (*Windows Components Wizard*), который запускают из окна приветствия Microsoft Windows Server 2003 или из утилиты *Установка и удаление программ* (*Add/Remove Programs*) в панели управления. Для Windows Server 2012 его необходимо скачивать с сайта www.microsoft.com. В Windows Server 2012 используется версия 3.4.

Сетевой монитор (*Network Monitor*) – это программный анализатор трафика, позволяющий:

− перехватывать кадры прямо из сети;

− отображать и фильтровать перехваченные кадры как во время сбора данных, так и после;

− редактировать перехваченные кадры и пересылать их по сети (только в полной версии);

− перехватывать кадры с удаленного компьютера (только в полной версии).

В частности, *Сетевой монитор* применяют для диагностики неполадок оборудования и ПО, когда сервер не в состоянии подключиться к другим компьютерам. Перехваченные кадры можно сохранять в файле или просматривать и анализировать непосредственно в окне *Сетевого монитора*. Разработчики сетевого ПО также применяют *Сетевой монитор* для мониторинга и отладки разрабатываемых сетевых прикладных программ.

Кадр (*frame*) – это инкапсулированный пакет данных сетевого уровня. Говоря, что *Сетевой монитор* перехватывает кадры, мы подразумеваем, что он считывает и отображает информацию об инкапсуляции, которая включает как данные сетевого (типа данных Ethernet), так более высоких уровней – таких протоколов, как ARP (Address Resolution Protocol), IP (Internet Protocol), TCP (Transmission Control Protocol) и DNS (Domain Name System). С технической точки зрения кадр отличается от пакета (packet) уровнем инкапсуляции: подразумевается, что последний относится к межсетевому уровню. Тем не менее под этими терминами часто подразумевают одно и то же.

Есть две версии *Сетевого монитора*. В составе Windows Server (бесплатно скачивается с сайта) поставляется базовая версия, а полная входит в Microsoft Systems Management Server.

Существует огромное различие между версиями *Сетевого монитора*: базовая версия собирает лишь информацию о трафике на локальном компьютере, а полная в состоянии перехватывать трафик любых компьютеров сетевого сегмента. К сожалению, это верно только в сетях, где нет коммутаторов, а только концентраторы. Но в действительности в большинстве современных сетей используются коммутаторы, которые пересылают кадры прямо на компьютер-адресат. Они сильно ограничивают возможности анализаторов протоколов (в том числе *Сетевого монитора*), скрывая весь трафик, который не создается или не предназначен компьютеру, на котором работает анализатор. Поэтому если связь узлов в сети обеспечивают коммутаторы, вы не сможете воспользоваться преимуществами полной версии.

7.1.2. Компоненты *Сетевого монитора*. Порядок работы *Сетевого монитора*

Сетевой монитор состоит из инструмента администрирования *Сетевой монитор* (*Network Monitor*) и агента *Драйвер сетевого монитора* (*Network Monitor Driver*). Оба необходимы для перехвата, отображения и анализа сетевых кадров.

Сетевой монитор отслеживает сетевой поток данных, который состоит из всей информации, пересылаемой по сети на данный момент времени. Перед пересылкой сетевое ПО разбивает данные на небольшие порции, или кадры, каждая из которых содержит следующую информацию:

1) адрес компьютера – отправителя сообщения;
2) адрес компьютера-адресата (который принял кадр);
3) заголовочная информация всех протоколов, использованных при пересылке кадра;
4) данные (или их часть), посылаемые на компьютер-адресат.

Сетевой монитор из состава Windows Server копирует в буфер кадры, исходящие или входящие на локальный компьютер, этот процесс называется записью данных (data capture). Объем информации, собираемой *Сетевым монитором*, ограничен лишь объемом памяти, однако обычно нужно собирать только небольшую часть всего потока кадров. Подмножество собираемых кадров задается фильтрами, работа которых напоминает запрос базы данных, – они выделяют из общего потока лишь нужную информацию. Фильтрованные кадры можно отсортировать на основе адресов источника и целевого узла, уровня протоколов: сетевого интерфейса, межсетевого и транспортного, а также на основе свойств протокола и при отклонении структуры кадров от заданного шаблона.

Установка и настройка *Сетевого монитора* будет рассмотрена далее.

Анализ записанных данных. При включении просмотра собранных данных открывается окно просмотра кадров со сводной информацией о кадрах в порядке их поступления (рис. 7.1).

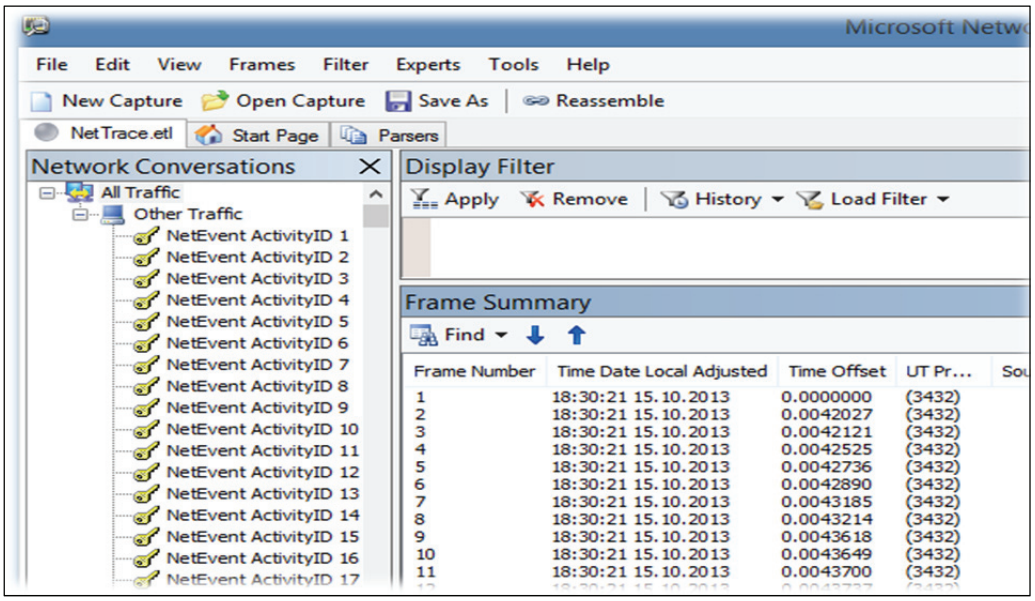


Рис. 7.1. Интерфейс *Сетевого монитора*

Двойной щелчок переключает режим отображения между исходным представлением со сводкой и представлением с тремя панелями: *Сводка* (*Summary*), *Сведения* (*Details*) и *Шестнадцатеричный* (*Hexadecimal*).

Панель *Сводка* содержит перечень всех кадров, отображаемых в текущем представлении. При выборе кадра информация о нем отображается в панелях *Сведения* и *Шестнадцатеричный*.

Панель *Сведения* содержит информацию о протоколе кадра, выбранного в панели *Сводка*. Когда кадр содержит инкапсуляцию протоколов нескольких уровней, здесь отображаются сведения о самой внешней оболочке. При выборе протокола в панели *Сводка* в панели *Шестнадцатеричный* отображаются соответствующие шестнадцатеричные строки.

На панели *Шестнадцатеричный* в шестнадцатеричном формате отображается содержимое выбранного кадра. Представленные в этой панели сведения полезны разработчикам, нуждающимся в максимально точной информации об используемых в создаваемом приложении сетевых протоколах.

Анализ кадров. В окне записи кадров в обратном порядке указаны содержащиеся в кадре протоколы: вверху – протокол самого низкого уровня (например, протокол сетевого интерфейса Ethernet), а внизу – протокол самого высокого уровня (например, прикладной протокол DNS). Именно так *Сетевой монитор* получает данные из сети.

Вот информация о кадре службы *Обозреватель компьютеров* (*Computer Browser*) в окне записи:

```
Frame: Base frame properties
ETHERNET: EType = Internet IP (IPv4)
IP: Protocol = UDP - User Datagram; Packet ID = 1576;
Total IP Length = 236; Options = No Options
UDP: Src Port: NETBIOS Datagram Service (138):
Dst Port: NETBIOS Datagram Service (138); Length = 216 (0xD8)
NBT: DS: Type = 17 (DIRECT GROUP)
SMB: C transact, File = \MAILSL0T\BR0 WSE
Browser: Workgroup Announcement [0x0c] WORKGROUP
```

Каждый протокол представлен в сводной (свернутой) форме: чтобы получить полную информацию, надо развернуть соответствующий узел. Первый уровень (Frame) добавлен *Сетевым монитором* в качестве описания кадра, которое содержит сведения об общей длине кадра и времени изменения с момента записи предыдущего кадра. Следующий уровень (ETHERNET) является самым «внешним» протоколом

кадра и соответствует уровню сетевого интерфейса в модели TCP/IP. За межсетевым уровнем следует протокол IP. В рассматриваемом наборе протоколов в качестве транспортного используется протокол UDP.

Добавление парсеров *Сетевого монитора*. Процесс чтения, анализа и описания содержимого кадров называется разбором (parsing) и выполняется специальными модулями, или парсерами (parser). В *Сетевом мониторе* это DLL-файлы, отвечающие за разбор и чтение сообщений различных протоколов. По умолчанию *Сетевой монитор* содержит более 20 парсеров, обеспечивающих разбор свыше 90 протоколов.

Функциональность *Сетевого монитора* можно расширять за счет подключения новых парсеров. Если в компании используется частный протокол, рекомендуется создать специальную DLL-библиотеку, позволяющую *Сетевому монитору* анализировать такой протокол. Файл нового парсера размещается в папке для парсеров *Сетевого монитора* – WINDOWS\System32\Netmon\Parsers. Кроме того, нужно добавить информацию о новом парсере и протоколе в файл Parser.ini. Это файл с описанием всех парсеров и протоколов, поддерживаемых *Сетевым монитором*, а размещается он в папке WINDOWS\System32\Netmon.

Добавление записей в файл Parser.ini на первый взгляд может показаться сложным, пока не узнаешь, что все записи одинаковы. Сначала в разделе parsers надо добавить следующую запись:

<имя_парсера>.dll = 0: <имя_протокола>

Затем найти разделы, соответствующие отдельным протоколам, скопировать один из них в конец файла и заменить название и описание, чтобы они соответствовали протоколу, поддерживаемому новым парсером.

Необходимо отметить, что для выполнения операций по прослушиванию сетевого трафика можно также воспользоваться альтернативными сниферами, например, Wireshark.

7.1.3. Использование *Сетевого монитора*

Запись данных средствами *Сетевого монитора*. Для записи и просмотра информации о трафике с помощью *Сетевого монитора* выполним следующие действия.

1. Входим в систему как *Администратор* (*Administrator*) и *Сетевой монитор*.

2. Далее открываем окно *Сетевой монитор* (*Network Monitor*) с сообщением о необходимости выбрать сеть (рис. 7.2). Щелкаем *OK*.

3. Разворачиваем узел *Локальный компьютер* (*Local Computer*) в левой панели окна *Выбор сети* (*Select a network*), чтобы открыть список сетевых адаптеров на локальном компьютере. Подключения по телефонной линии объединены в узел *Подключение удаленного доступа или VPN* (*Dial-up Connection or VPN*) (рис. 7.3).

4. Выбираем *Подключение к локальной сети* (*Local Area Connection*) и кликаем *OK*. Откроется окно *Сетевого монитора* (*Network Monitor*) с окном *Запись* (*Capture*) для выбранного сетевого адаптера (рис. 7.4).

5. На панели инструментов окна *Запись* (*Capture*) нажимаем кнопку *Начать запись данных* (*Start Capture*) (рис. 7.5).

6. Из командной строки выполняем команду *Ping 127.0.0.1*, чтобы проверить сетевые подключения.

7. По завершении работы команды *Ping* на панели инструментов щелкаем кнопку *Закончить запись и просмотреть данные* (*Stop and View Capture*) или нажимаем *Shift + F11*.

8. Откроется окно записи данных с заголовком *Запись данных*: 1 (*Capture*: 1). В скобках отображается слово *Сводка* (*Summary*), указывающее на то, что панель сводных данных является активной и единственной видимой панелью окна. Здесь перечисляются все записанные кадры.

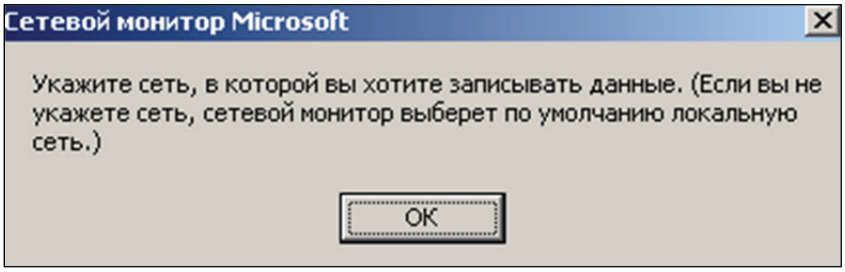


Рис. 7.2. Окно с сообщением о необходимости выбора сети

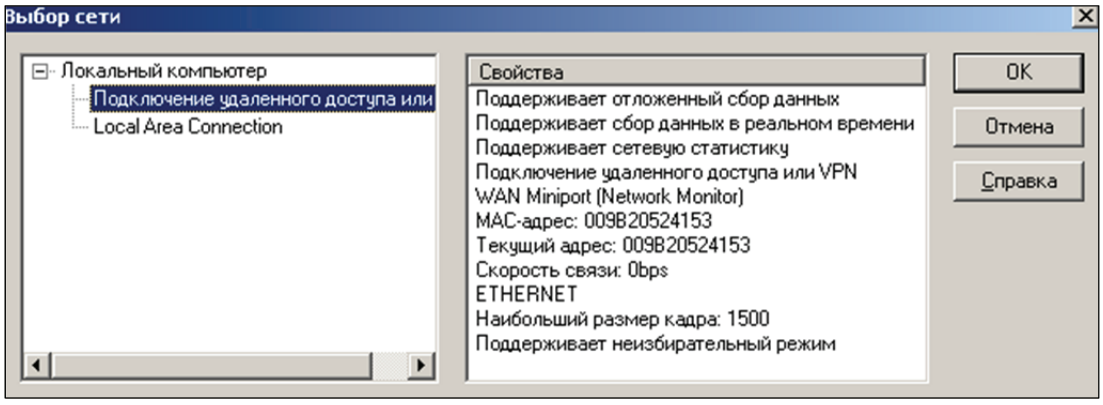


Рис. 7.3. Окно выбора подключения

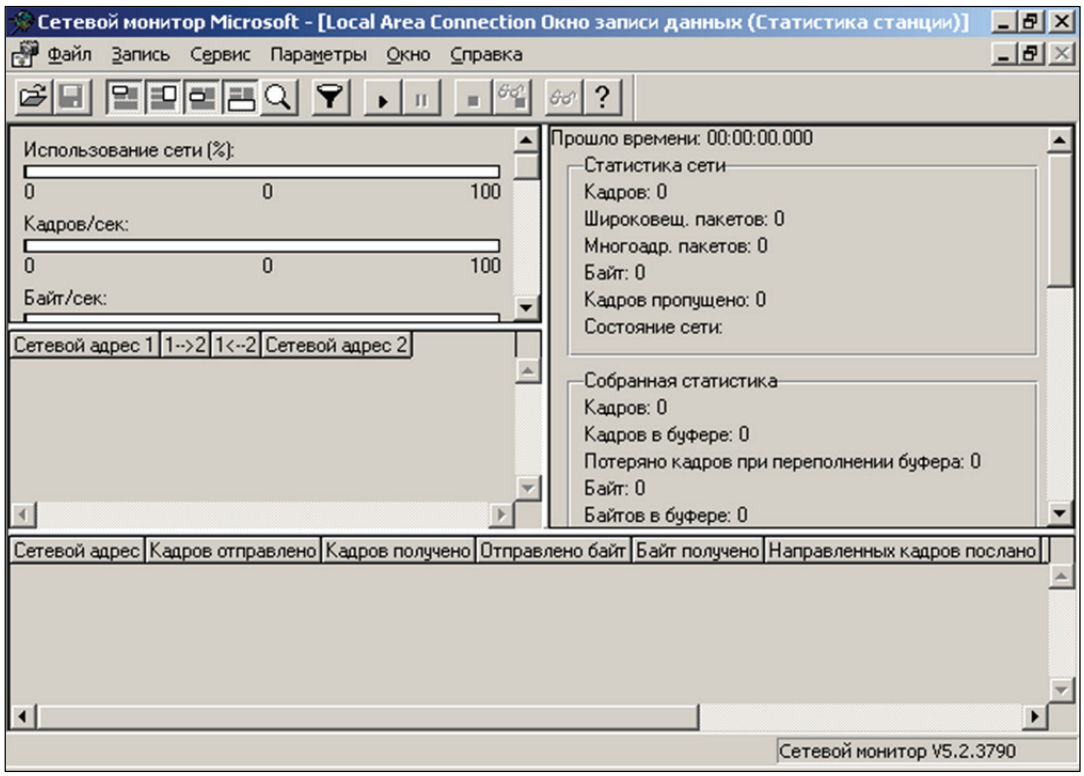


Рис. 7.4. Основное окно *Сетевого монитора*

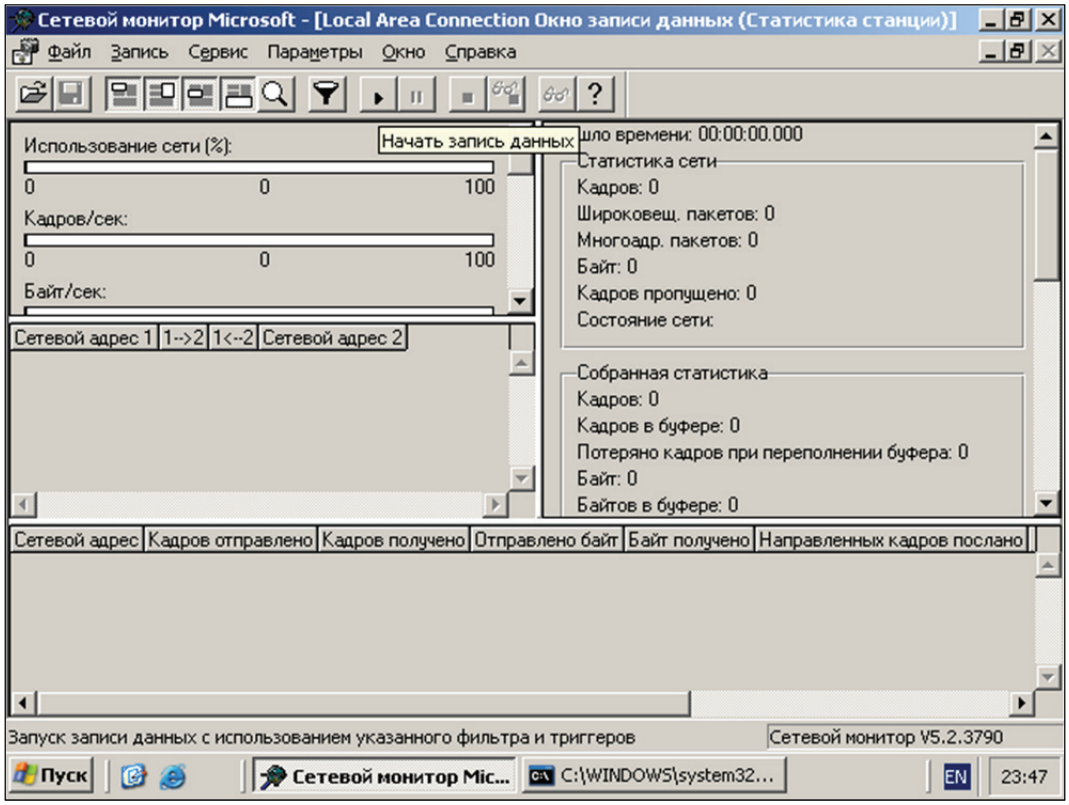


Рис. 7.5. Окно *Сетевого монитора*,
выполняющего запись передаваемой информации

9. Дважды щелкаем любой из кадров, указанных в панели *Сводка* (*Summary*). В окне записи данных откроются две дополнительные панели: *Сведения* (*Details*) и *Шестнадцатеричный* (*Hexadecimal*), содержащие подробную информацию о выбранном кадре.

10. Снова дважды щелкаем кадр в панели *Сводка* (*Summary*). Панели *Сведения* (*Details*) и *Шестнадцатеричный* (*Hexadecimal*) скроются – так переключаются между двумя представлениями окна *Запись* (*Capture*).

11. Выбираем *Файл* (*File*) → *Сохранить как* (*Save As*), чтобы открыть окно *Сохранить как* (*Save As*).

12. В поле *Имя файла* (*File Name*) вводим *Ping Capture* и щелкаем *Сохранить* (*Save*). Файл Ping Capture.cap сохранится в папке \Рабочий стол\Мои документы\Мои записи (\Desktop\My Documents\My Captures).

13. Выбираем *Файл* (*File*) → *Закрыть* (*Close*). Окно записи данных закроется, а в консоли *Сетевой монитор* (*Network Monitor*) снова появится окно *Запись* (*Capture*).

Сохранение кадров в текстовом файле. Копирование информации пакета в текстовый файл выполняется в окне *Сетевой монитор* (*Network Monitor*) под учетной записью *Администратор* (*Administrator*).

1. Выбираем *Файл* (*File*) → *Открыть* (*Open*). Откроется окно *Открыть* (*Open*) с файлом Ping Capture.cap в папке Мои записи (My Captures).

2. Выбираем файл Ping Capture.cap и щелкаем *Открыть* (*Open*), чтобы открыть его в окне записи данных.

3. В панели *Сводка* (*Summary*) находим и выбираем кадр со словом ICMP в столбце *Протокол* (*Protocol*).

4. Нажимаем *Ctrl* + *C*, чтобы скопировать кадр.

5. Открываем *Блокнот* (*Notepad*) и нажимаем *Ctrl* + *V*, чтобы вставить информацию о кадре в новый текстовый файл. В текстовый файл вставляются все данные записанного кадра. Обратите внимание, первая строка содержит все поля и в той же последовательности, что и в панели *Сводка* (*Summary*) окна сбора данных. Кроме того, большая часть данных – около 40 строк – соответствует информации, отображаемой в панели *Сведения* (*Details*). Но здесь информация представлена в развернутом виде. В конце текста размещены шестнадцатеричные значения из панели *Шестнадцатеричный* (*Hexadecimal*).

6. В *Блокноте* (*Notepad*) нажимаем *Ctrl* + *S*, чтобы сохранить файл. В открывшемся окне *Сохранить как* (*Save As*) выбираем папку \Рабочий стол\Мои документы\Мои записи (\Desktop Documents\My Captures), но пока не сохраняем файл.

7. В поле со списком *Кодировка* (*Encoding*) выбираем *Юникод* (*Unicode*).

8. В поле *Имя файла* (*File Name*) вводим *ICMP frame* и щелкаем *Сохранить* (*Save*).

9. Закрываем окно *ICMP Frame.txt – Блокнот* (*ICMI Frame.txt – Notepad*).

10. Закрываем окно *Сетевой монитор* (*Network Monitor*), выбрав *Файл* (*File*) → *Выход* (*Exit*). На предложение сохранить адрес в базе данных отвечаем *Нет* (*No*).

11. Выходим из системы.

7.2. Протокол IPsec

Протокол Kerberos применяется для аутентификации участников соединения. Но и после этапа аутентификации данные, передаваемые по сети, следует защищать. Стандартные протоколы стека TCP/IP, такие как IP, TCP, UDP, не обладают встроенными средствами защиты. На эту проблему в 1994 г. обратил внимание Совет по архитектуре Интернета (Internet Architecture Board, IAB), издав RFC 1636 (Report of IAB Workshop on Security in the Internet Architectures («Отчет семинара IAB по безопасности в архитектуре Интернета»)). Инициированная этим сообщением работа привела к появлению протокола *IPsec* (IPSecurity – безопасность IP), описанного в нескольких стандартах RFC (в частности, в RFC 2401–2412). Новая технология безопасности является необходимой частью протокола IPv6, а также применяется и в сетях IPv4.

Протокол IPsec действует на сетевом уровне модели OSI и может применяться независимо от протоколов верхнего уровня, т. е. прикладной протокол может использовать IPsec, считая, что работает с обычным протоколом IP. При этом данные протоколов верхних уровней упаковываются в пакеты IPsec, которые, в свою очередь, помещаются в пакеты протокола IP.

7.2.1. Функции протокола IPsec

Протокол IPsec обеспечивает наличие следующих функций:

– аутентификация – приемник пакетов в состоянии проверить подлинность их источника;

– целостность – осуществляется контроль того, что данные дойдут до получателя в неизменном виде;

– конфиденциальность – шифрование данных обеспечивает их недоступность для несанкционированного просмотра;

– распределение секретных ключей – для правильной работы протокола IPsec необходимо автоматически обеспечивать источник и приемник пакетов секретными ключами для шифрования и расшифрования данных.

Для реализации представленных функций используются три основных протокола:

1) AH (Authentication Header – заголовок аутентификации) обеспечивает целостность и аутентичность;

2) ESP (Encapsulating Security Payload – инкапсуляция зашифрованных данных) предоставляет функции целостности, аутентичности и конфиденциальности;

3) IKE (Internet Key Exchange – обмен ключами Интернета) генерирует и распределяет секретные ключи.

Можно заметить, что протокол ESP имеет схожие функции с протоколом AH. Пересечение функций вызвано тем, что на применение протоколов шифрования во многих странах накладываются определенные ограничения. В связи с этим оба протокола могут применяться независимо, хотя наивысший уровень защиты достигается при их совместном использовании.

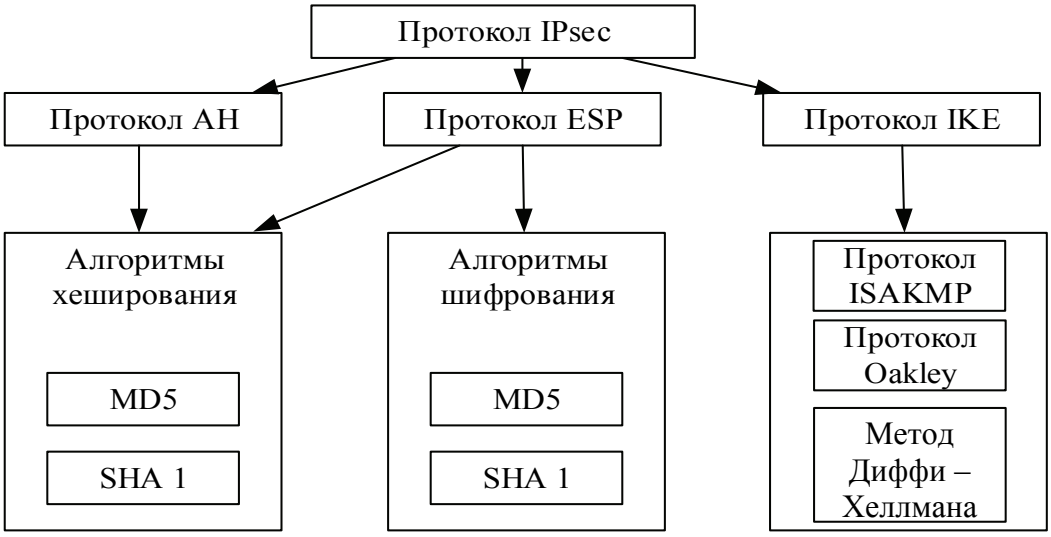


Рис. 7.6. Структура протокола IPsec

На рис. 7.6 представлена структура протокола IPsec и взаимосвязь основных протоколов, входящих в его состав.

7.2.2. Протоколы AH и ESP

Протокол AH (описан в RFC 2402) снабжает пакет IPsec своим незашифрованным заголовком, который обеспечивает:

− аутентификацию исходных данных;
− целостность данных;
− защиту от дублирования уже полученных данных.

Первые две функции протокола AH реализуются путем применения алгоритмов хеширования (MD5 (алгоритм MD5 (Message Digest – алгоритм формирования профиля сообщения) разработан Рональдом Ривестом (Ronald Rivest), описан в RFC 2403) или SHA1 (алгоритм SHA1 (Secure Hash Algorithm – алгоритм безопасного хеша) разработан Национальным институтом стандартов и технологий (National Institute of Standards and Technology, NIST), является более стойким по сравнению с MD5, описан в RFC 2404)). Процедура хеширования осуществляется источником с помощью секретного ключа, который был выдан источнику и приемнику пакета с использованием протокола IKE. Полученное значение хеша помещается в специальное поле заголовка AH. Приемник также осуществляет процедуру хеширования, применяя тот же секретный ключ. В том случае если вычисленный хеш совпадает с хешем, извлеченным из пакета, данные считаются аутентифицированными и целостными. Иначе пакет в процессе передачи подвергся каким-либо изменениям и не является правильным.

Функция защиты от дублирования уже полученных пакетов осуществляется с помощью поля номера пакета в заголовке AH. В это поле приемник заносит значение счетчика, увеличивающееся при отправке каждого пакета на единицу. Приемник отслеживает номера получаемых пакетов, и если такой номер совпадает с недавно полученным, пакет отбрасывается.

Протокол ESP (описан в RFC 2406) решает задачи, подобные протоколу AH, – обеспечение аутентификации и целостности исходных данных, а также защита от дублирования пакетов. Кроме того, протокол ESP предоставляет средства обеспечения конфиденциальности данных при помощи алгоритмов шифрования.

Задачи аутентификации, целостности и защиты от дублирования решаются теми же методами, что и в протоколе AH. Передаваемый пакет, за исключением нескольких служебных полей, шифруется с применением алгоритмов шифрования DES и 3DES (DES с тремя ключами).

7.2.3. Протокол IKE

Управление секретными ключами в протоколе IPsec осуществляется при помощи протокола IKE (описан в RFC 2409). Данный протокол основан на двух протоколах: ISAKMP (Internet Security Association and Key Management Protocol – протокол межсетевой ассоциации защиты и управления ключами) и протоколе определения ключей Оакли (Oakley Key Determination Protocol).

Протокол IKE устанавливает соединение между двумя узлами сети, называемое ***безопасной ассоциацией*** (Security Association, SA). Безопасная ассоциация обеспечивает передачу защищенных данных только в одну сторону, поэтому для установки двустороннего соединения требуется определить две безопасные ассоциации. Для аутентификации узлов безопасной ассоциации, согласования между ними методов хеширования и шифрования IKE использует протокол ISAKMP (описан в RFC 2408).

Для генерации и обмена секретными ключами IKE использует протокол определения ключей Оакли (описан в RFC 2412), разработанный на основе метода обмена ключами Диффи – Хеллмана (Diffie – Hellman). В этом методе секретный ключ генерируется на двух узлах путем обмена двумя числами через открытую сеть. При этом перехват чисел не даст информации о ключах.

7.3. Настройка протокола IPSecurity

Для шифрования сетевого трафика необходимо выполнить настройку политики безопасности (оснастку) на обоих компьютерах с операционной системой Windows Server. Далее будет рассмотрен пример настройки одной из ОС.

1. В командной строке выполняем команду MMC. Откроется оболочка *Microsot Management Console*. В меню *File* выбираем *Add*/*Remove Snap-In* и добавляем две консоли: *IP Security Policy Managment* (для локального компьютера) и *IP Security Monitor*. Нажимаем на кнопку *Add*, а затем *OK*, чтобы вернуться в основное окно консоли. Для удобства созданную нами консоль можно сохранить, например, на рабочем столе под именем IPSec.msc.

2. В созданной консоли раскрываем узел *IP Security Policies on Local Computer*, щелкаем по этому узлу правой кнопкой мыши и в контекстном меню выбираем *Create IP Security Policy*. Запустится мастер создания политики IPSecurity.

3. На первом экране мастера вводим имя политики (например, TestPolicy) и нажимаем *Next*.

4. На втором экране (*Requests for Secure Communication*) снимаем флажок *Activate the default response rule* и щелкаем *Next*.

5. На последнем экране мастера убеждаемся, что флажок *Edit Properties* установлен, и кликаем *Finish*. Откроется экран свойств нашей политики. Нажимаем в нем на кнопку *Add*, чтобы добавить новое правило для нашей политики. На первом экране мастера создания правил щелкаем *Next*.

6. На втором экране мастера (*Tunnel Endpoint*) проверяем, стоит ли переключатель в положении *This rule does not specify a tunnel*, и нажимаем *Next*.

7. На экране *Network Type* оставляем переключатель в положении *All network connections* и щелкаем *Next*.

8. На экране *IP Filter List* нажимаем на кнопку *Add*. Откроется окно создания нового фильтра. В этом окне вводим название фильтра (например, имя компьютера партнера_filter) и щелкаем *Add*. Откроется еще один мастер создания фильтров. На его первых двух экранах нажимаем *Next*.

9. На экране *IP Traffic Source* оставляем в качестве адреса источника *My IP Address* и щелкаем *Next*.

10. На экране *IP Traffic Destination* выбираем в списке адресов назначения *A specific IP address* и указываем IP-адрес нашего партнера. На остальных экранах этого мастера оставляем значения по умолчанию. Опять возвращаемся в окно *IP Filter List*, в котором будет присутствовать созданный нами фильтр. Нажимаем в нем *OK* и в окне *Security Rule Wizard* на экране *IP Filter List* устанавливаем переключатель напротив созданного нами фильтра. Щелкаем *Next*.

11. На следующем экране (*Filter Action*) устанавливаем переключатель в положение *Require Security* и нажимаем *Next*.

12. На экране *Authentification Method* устанавливаем переключатель в положение *Use this string to protect the key exchange* (*preshared key*) и в поле внизу вводим текстовое значение, например TEST. Это значение должно совпадать с тем значением, которое ввел у себя партнер. Щелкаем *Next*, на последнем экране снимаем флажок *Edit Properties* и нажимаем *Finish*. Затем в окне консоли *MMC* кликаем правой кнопкой мыши по созданной политике и в контекстном меню выбираем *Assign*. Дожидаемся, пока партнер завершит выполнение аналогичных действий на своем компьютере.

13. Раскрываем узел *IP Security Monitor* → *Имя вашего компьютера* → *Active Policy* и просматриваем информацию о назначенной нами политике и статистике взаимодействия по IPSec (под Main Mode).

14. Запускаем *Network Monitor* (либо другой снифер) и настраиваем в нем фильтр для перехвата трафика между операционными системами Windows Server. В качестве сетевого трафика может выступать отправленный ping-запрос или подключение по ftp, http, telnet и т. д. *Network Monitor* покажет служебную информацию протокола ESP (а не какого-либо другого в зависимости от типа сетевого трафика). Необходимо обратить внимание, что для успешного шифрования сетевого трафика настроить оснастки следует на обеих машинах с операционными системами Windows Server.

Лабораторная работа № 12–13

***Цель*:** изучение методов прослушивания и шифрования сетевого трафика между операционными системами типа Windows.

***Задание*:** необходимо выполнить настройку политик безопасности (оснастки) для шифрования сетевого трафика с помощью протокола IPSecurity на обеих виртуальных машинах с ОС Windows Server. Следует проверить обеспечение безопасности (шифрования данных) путем прослушивания сетевого трафика при помощи программы-снифера, например *Network Monitor*.

Раздел 8
НАСТРОЙКА ВЕБ-СЕРВЕРА APACHE

8.1. Введение в веб-сервер Apache

Веб-сервер Apache в настоящее время является одним из самых популярных веб-серверов в мире. Он хорошо документирован и используется с момента создания сети Интернет, что позволяет его применять для хостинга веб-сайта. Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS. Основными достоинствами Apache считаются надежность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать системы управления базами данных (СУБД) для аутентификации пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает IPv6.

Существует множество модулей, добавляющих к Apache поддержку различных языков программирования и систем разработки. К ним относятся: PHP, Python, Ruby, Perl, ASP, TCL.

Кроме того, Apache поддерживает механизмы CGI и FastCGI, что позволяет исполнять программы практически на всех языках программирования, в том числе C, C++, Lua, sh, Java.

Apache имеет различные механизмы обеспечения безопасности и разграничения доступа к данным, основными из которых являются:

– ограничение доступа к определенным директориям или файлам;

– механизм авторизации пользователей для доступа к директории на основе HTTP- и digest-аутентификации;

– ограничение доступа к определенным директориям или всему серверу, основанное на IP-адресах пользователей;

– запрет доступа к определенным типам файлов для всех или части пользователей, например запрет доступа к конфигурационным файлам и файлам баз данных;

– наличие модулей, реализующих авторизацию через СУБД или PAM.

Также существует механизм suexec, используемый для запуска скриптов и CGI-приложений с правами и идентификационными данными пользователя.

Для шифрования данных, передающихся между клиентом и сервером, используется механизм SSL, реализованный через библиотеку

OpenSSL. Для удостоверения подлинности веб-сервера используются сертификаты X.509.

Администраторы часто выбирают Apache из-за его гибкости, мощности и широкой распространенности. Он может быть расширен с помощью системы динамически загружаемых модулей и исполнять программы на большом количестве интерпретируемых языков программирования без использования внешнего программного обеспечения.

Apache предоставляет несколько модулей мультипроцессинга (multi-processing modules, MPM), которые отвечают за то, как запрос клиента будет обработан. Это позволяет администраторам определять политику обработки соединений. Ниже представлен список MPM-модулей Apache:

1) mpm_prefork – этот модуль создает по одному процессу с одним потоком на каждый запрос. Каждый процесс может обрабатывать только одно соединение в один момент времени. Пока число запросов меньше числа процессов, данный MPM работает очень быстро. Однако производительность быстро падает, когда число запросов начинает превосходить число процессов, поэтому в большинстве случаев это не самый лучший выбор. Каждый процесс потребляет значительный объем RAM, поэтому этот MPM сложно поддается масштабированию. Но он может быть использован вместе с компонентами, которые не созданы для работы в многопоточной среде. Например, PHP не является потокобезопасным, поэтому данный MPM рекомендуется использовать как безопасный метод работы с mod_php;

2) mpm_worker – этот модуль создает процессы, каждый из которых может управлять несколькими потоками. Каждый поток может обрабатывать одно соединение. Потоки значительно более эффективны, чем процессы, это означает, что mpm_worker масштабируется значительно лучше, чем mpm_prefork. Так как потоков больше, чем процессов, это свидетельствует о том, что новое соединение может быть сразу обработано свободным потоком и нет необходимости ждать, пока освободится процесс;

3) mpm_event – этот модуль похож на mpm_worker, но оптимизирован под работу с keep-alive соединениями. Когда используется mpm_worker, соединение будет удерживать поток вне зависимости от того, активное это соединение или keep-alive. Mpm_event выделяет отдельные потоки для keep-alive соединений и отдельные потоки для активных соединений. Это позволяет модулю не погрязнуть в keep-alive соединениях, что необходимо для быстрой работы. Указанный модуль был отмечен как стабильный в Apache версии 2.4.

Среди наиболее распространенных модулей можно назвать следующие:

– mod_perl – модуль, интегрирующий Perl-интерпретатор в Apache httpd, что позволяет писать высокопроизводительные CGI-скрипты на языке Perl;

– FastCGI – программный интерфейс (C, Perl, Java, TCL, Python), позволяющий увеличить в несколько раз производительность CGI-скриптов путем единичной загрузки скрипта в память, буферизации ввода-вывода и циклической обработки запросов с помощью FastCGI API;

– mod_backhand – модуль, который позволяет с равномерным распределением нагрузки объединить несколько веб-серверов в кластер, обслуживающий один высокозагруженный ресурс.

Apache может раздавать статический контент, используя стандартные file-based методы. Производительность таких операций зависит от выбранного MPM.

Apache также может раздавать динамический контент, встраивая интерпретатор нужного языка в каждый воркер. Это позволяет обрабатывать запросы к динамическому содержимому средствами самого веб-сервера и не полагаться на внешние компоненты. Интерпретаторы языков могут быть подключены к Apache с помощью динамически загружаемых модулей.

Возможность обрабатывать динамический контент средствами самого Apache упрощает конфигурирование. Нет необходимости настраивать взаимодействие с дополнительным софтом, динамический модуль может быть легко отключен в случае изменившихся требований.

Система модулей Apache позволяет динамически загружать и выгружать модули, чтобы удовлетворить ваши потребности, в то время как ваш сервер запущен. Ядро Apache всегда доступно, в то время как модули можно включать и выключать, чтобы добавить или удалить функциональность из основного сервера.

Apache использует эту функциональность для решения широкого круга задач. Благодаря зрелости платформы существует огромное множество модулей, которые могут изменять ключевые особенности сервера, например модуль mod_php позволяет включать PHP-интерпретатор в каждый воркер.

Использование модулей не ограничивается лишь обработкой динамических запросов. К возможностям модулей относятся: изменение URL (URL rewrite), аутентификация клиентов, защита сервера, логирование, кеширование, сжатие, проксирование, ограничение частоты запросов, шифрование. Динамические модули могут значительно расширить функциональность ядра.

8.2. Установка и развертывание программного обеспечения

8.2.1. Разбор конфигурационного файла

По умолчанию все конфигурационные файлы хранятся в папке etc/apache2 (рис. 8.1).

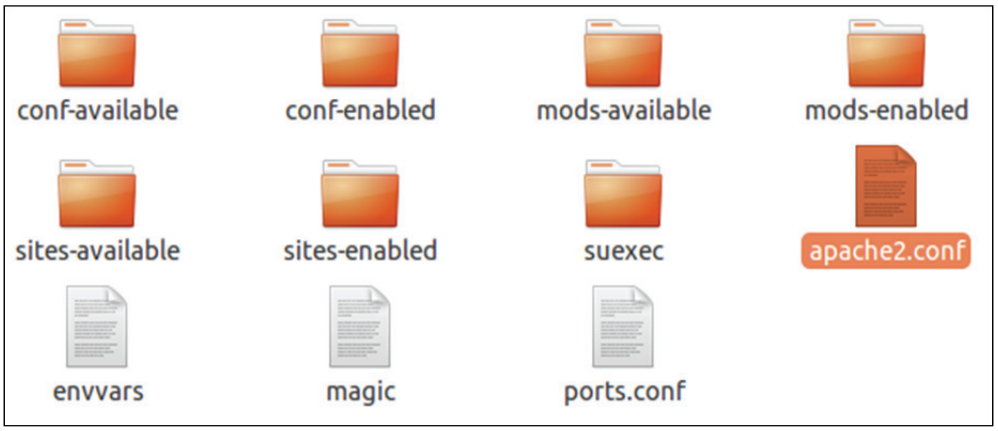


Рис. 8.1. Список конфигурационных файлов

Файл apache2.conf – это основной конфигурационный файл сервера Apache2, который содержит глобальные настройки для всего Apache2. Для вступления в силу изменений, внесенных в этот файл, требуется перезапуск сервера Apache.

Файл envvars – это файл, где устанавливаются переменные окружения Apache2.

Файл ports.conf – это файл, содержащий инструкции, которые определяют, какие TCP-порты прослушивает Apache2.

Каталог mods-available содержит конфигурационные файлы как для загрузки модулей, так и для их настройки. Тем не менее не все модули имеют отдельные файлы настройки.

Каталог mods-enabled содержит символьные ссылки на файлы в /etc/apache2/mods-available. Когда создается символьная ссылка на файл настроек модуля, он включается при следующем рестарте Apache2.

Каталог sites-available содержит файлы настроек для виртуальных сетевых узлов (Virtual Hosts) Apache2. Виртуальные сетевые узлы позволяют настраивать Apache2 на множество сайтов с отдельными конфигурациями.

Каталог sites-enabled подобно mods-enabled содержит символьные ссылки на каталог /etc/apache2/sites-available. Аналогично, когда файл настроек из sites-available получает здесь символьную ссылку, соответствующий ему сайт будет активен при следующем перезапуске Apache2.

Синтаксис конфигурационных файлов должен выглядеть следующим образом. На одной строке должна быть расположена только одна директива. Символ \ (обратный слэш) может быть использован в качестве последнего символа строки, чтобы указать, что директива продолжается на следующей строке. После символа \ не должно быть никаких других пробельных символов, кроме символа конца строки.

Директива и ее аргументы разделяются пробелом. Сами аргументы также разделяются пробелами. Если аргумент содержит пробелы, его необходимо заключить в кавычки.

Директивы не чувствительны к регистру символов, а вот аргументы наоборот – чувствительны.

Строка, которая начинается с символа # (решетка), считается комментарием и игнорируется. Комментарий не может быть расположен на одной строке с директивой.

Пробельные символы, вставленные до директивы, игнорируются, поэтому их можно использовать для вставки отступов, чтобы было удобнее читать код. Пустые строки также игнорируются.

Модуль – это просто подключаемый файл, который позволяет добавить дополнительный функционал. Все подключаемые модули по умолчанию располагаются в папке modules.

Модули позволяют использовать директивы, которые не входят в состав ядра веб-сервера Apache.

.htaccess – это файл дополнительной конфигурации веб-сервера Apache, а также подобных ему серверов. Он позволяет настраивать функционал для отдельных каталогов, без изменения основного конфигурационного файла, так как доступ к нему чаще всего запрещен.

Файл .htaccess действует только на каталог, в котором располагается, и на его дочерние каталоги. Файл .htaccess может быть размещен в любом каталоге. Директивы этого файла действуют на все файлы в текущем каталоге и во всех его подкаталогах (если эти директивы не переопределены директивами нижележащих файлов .htaccess). Изменения, вносимые в файл .htaccess, не требуют перезапуска сервера.

Если в файле .htaccess была допущена какая-либо ошибка, например, неправильно написана директива или значение директивы, то сервер вернет ошибку: Error 500.

Файл .htaccess использует тот же синтаксис, что и конфигурационные файлы Apache.

Для того, чтобы включить .htaccess в Apache2, надо отредактировать всего лишь один файл. Этот файл лежит в каталоге /etc/apache2/sites-available и называется 000-default.conf (рис. 8.2).

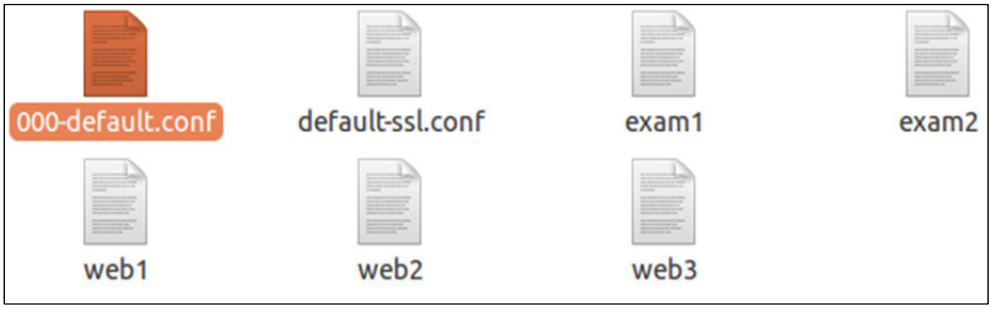


Рис. 8.2. Конфигурационный файл

Отредактировать файл можно любым текстовым редактором так, как это показано на рис. 8.3. После чего следует перезагрузить Apache.

```
<Virtualhost *:80>
        ServerAdmin admin@site.com

        DocumentRoot /var/www

        <Directory /var/www>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride All # - это значение было None
                Order allow,deny
                allow from all
        </Directory>
</Virtualhost>
```

Рис. 8.3. Отредактированный конфигурационный файл 000-default.conf

8.2.2. Виртуальные хосты

Apache разделяет свои функциональные возможности и компоненты на отдельные части, которые могут быть настроены и сконфигурированы независимо друг от друга. Базовая часть, которая отвечает за отдельный сайт или домен, называется ***виртуальным хостом*** (virtual host).

Эта система позволяет администратору использовать один сервер, чтобы раздавать несколько сайтов, используя один интерфейс или IP.

Каждый настроенный соответствующим образом домен будет направлять пользователя к определенной директории сервера, содержащей информацию этого сайта, соответствующего домену. При этом посетитель сайта не узнает, что данный сервер хранит и другие сайты.

Для создания виртуальных хостов в первую очередь необходимо создать структуру директорий, содержащую данные сайта, которые будут отображаться посетителям (рис. 8.4).

```
root@polina-VirtualBox:/# mkdir -p /var/www/test1.com/public_html
root@polina-VirtualBox:/# mkdir -p /var/www/test2.com/public_html
```

Рис. 8.4. Создание директорий

Создадим контент для отображения наших сайтов (рис. 8.5 и 8.6).

```
root@polina-VirtualBox:/# nano  /var/www/test1.com/public_html/index.html
root@polina-VirtualBox:/# nano  /var/www/test2.com/public_html/index.html
```

Рис. 8.5. Создание файлов

```
<html>
<head>
<title>Welcome to Test.com!</title>
</head>
<body>
<h1>Success! The test2.com virtual host is working!</h1>
</body>
</html>
```

Рис. 8.6. Код страницы

Файлы виртуальных хостов задают их конфигурацию и определяют, как именно веб-сервер Apache будет отвечать на запросы к разным доменам. По умолчанию, Apache имеет файл виртуального хоста 000-default.conf, который можно использовать в качестве отправной точки. Скопируем его, чтобы создать файлы виртуального хоста для каждого из наших доменов (рис. 8.7).

```
root@polina-VirtualBox:/# cp /etc/apache2/sites-available/000-default.conf  /etc/apache2/sites-av
ailable/test1.com.conf
```

Рис. 8.7. Копирование файлов

После этого откроем файл и изменим в нем настройки. Прежде всего, мы должны изменить директиву ServerAdmin на адрес электронной почты, на который администратор сайта будет получать электронные письма. Затем мы должны добавить две новые директивы. Первая, ServerName, устанавливает основной домен, который должен соответствовать названию виртуального хоста. Вторая, ServerAlias, определяет другие имена, которые должны интерпретироваться так, как будто это основной домен. Далее необходимо изменить это расположение корневого каталога данного домена (рис. 8.8).

```
ServerAdmin linagurinovich@mail.ru
ServerName test1.com
ServerAlias www.test1.com
DocumentRoot /var/www/test1.com/public_html
```

Рис. 8.8. Все настройки

Аналогично настраиваем второй домен.

После этого необходимо включить каждый сайт и перезагрузить сервер (рис. 8.9). Проверяем (рис. 8.10).

```
root@polina-VirtualBox:/# a2ensite test1.com.conf
Enabling site test1.com.
To activate the new configuration, you need to run:
  service apache2 reload
root@polina-VirtualBox:/# a2ensite test2.com.conf
Enabling site test2.com.
To activate the new configuration, you need to run:
  service apache2 reload
```

Рис. 8.9. Включение сайтов

Рис. 8.10. Виртуальный хост

8.2.3. Установка LAMP (Linux, Apache, MySQL, PHP)

Перед началом установки обновим свою систему и убедимся, что установлены самые свежие пакеты. После этого можно начинать установку Apache2 (рис. 8.11 и 8.12).

```
polina@polina-VirtualBox:~$ sudo apt-get update
```

Рис. 8.11. Установка обновлений

```
polina@polina-VirtualBox:~$ sudo apt-get install apache2
```

Рис. 8.12. Установка Apache2

После установки следует проверить работу Apache2, для этого в браузере необходимо ввести localhost. Должна появиться страница по умолчанию Apache2 (рис. 8.13).

Если этого не случилось, возможно включен файрволл. Следует разрешить Apache2 выполнять запрос на порт 80 и порт 443.

Для этого необходимо установить UFW (рис. 8.14). Затем разрешить трафику с HTTP и HTTPS проходить через файрволл (рис. 8.15).

Следующий компонент – MySQL. Это система управления базами данных приложения (см. рис. 8.16 на с. 127).

Во время установки будет запрошен пароль пользователя root (см. рис. 8.17 на с. 127). Нельзя оставлять поле пустым.

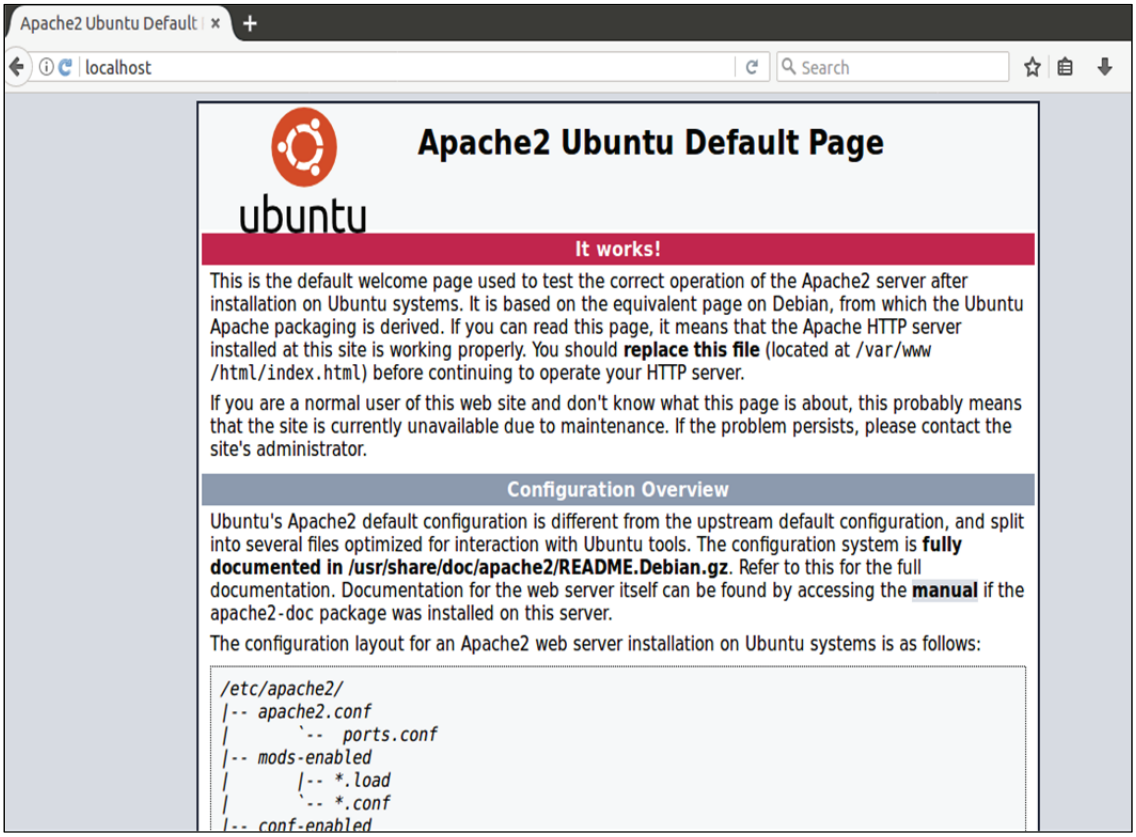


Рис. 8.13. Страница по умолчанию Apache2

```
polina@polina-VirtualBox:~$ sudo apt-get install ufw
Чтение списков пакетов… Готово
Построение дерева зависимостей
Чтение информации о состоянии… Готово
```

Рис. 8.14. Установка UFW

```
polina@polina-VirtualBox:~$ sudo ufw allow https
Правила обновлены
Правила обновлены (v6)
polina@polina-VirtualBox:~$ sudo ufw allow http
Правила обновлены
Правила обновлены (v6)
```

Рис. 8.15. Разрешения

После установки можно проверить статус MySQL (рис. 8.18).

После установки компонентов, описанных выше, можно приступать к установке главного компонента – PHP (рис. 8.19).

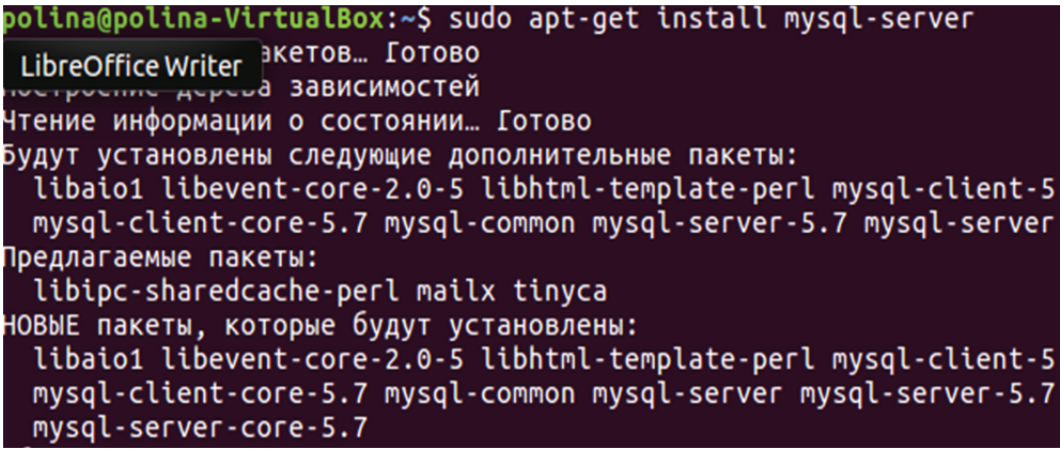


Рис. 8.16. Установка MySQL

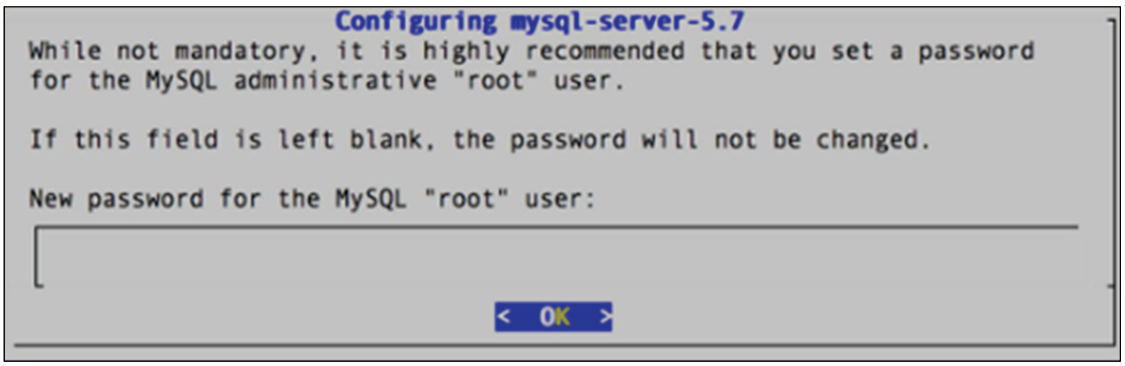


Рис. 8.17. Пароль пользователя root

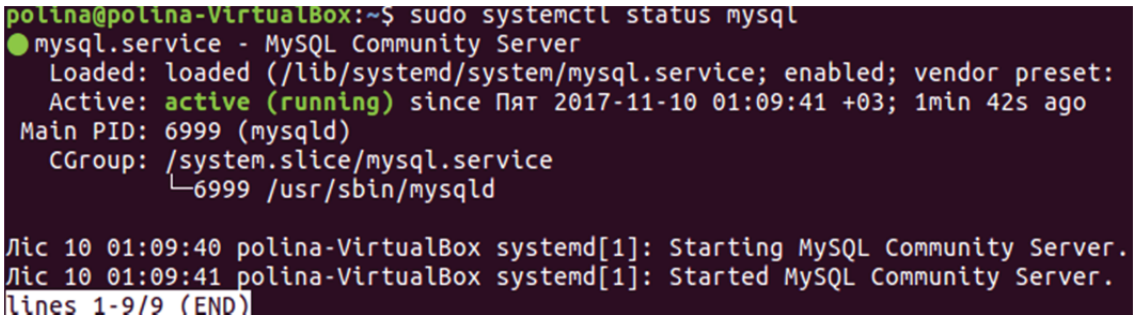


Рис. 8.18. Статус MySQL

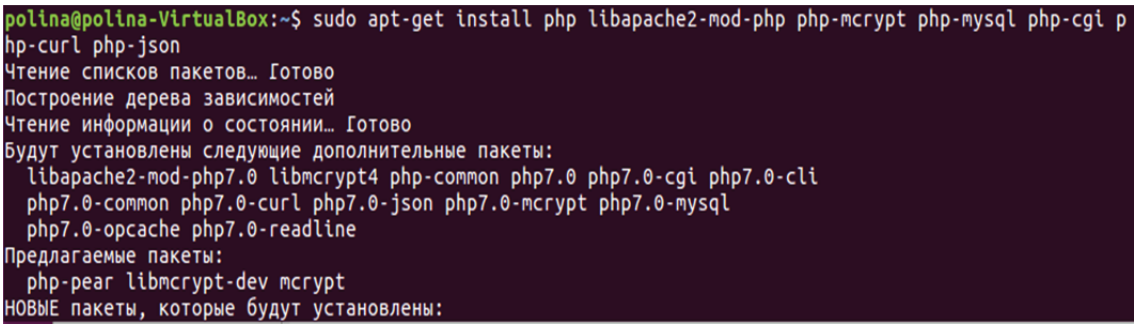


Рис. 8.19. Установка PHP

Для проверки установки PHP создадим файл с расширением .php и выведем информацию о PHP (рис. 8.20). Чтобы проверить работу PHP, необходимо набрать в браузере после localhost название своей странички с расширением (рис. 8.21).

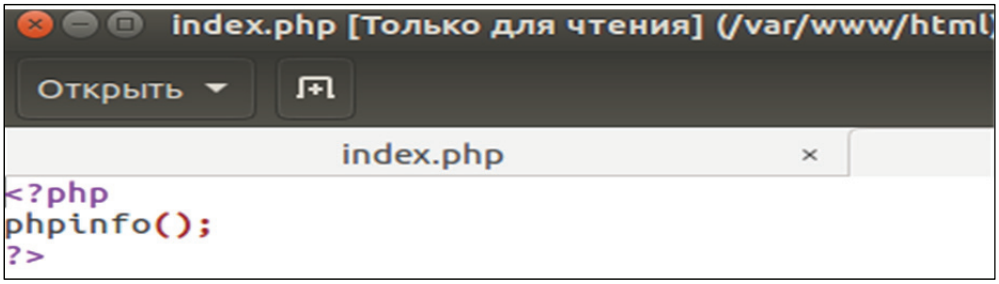
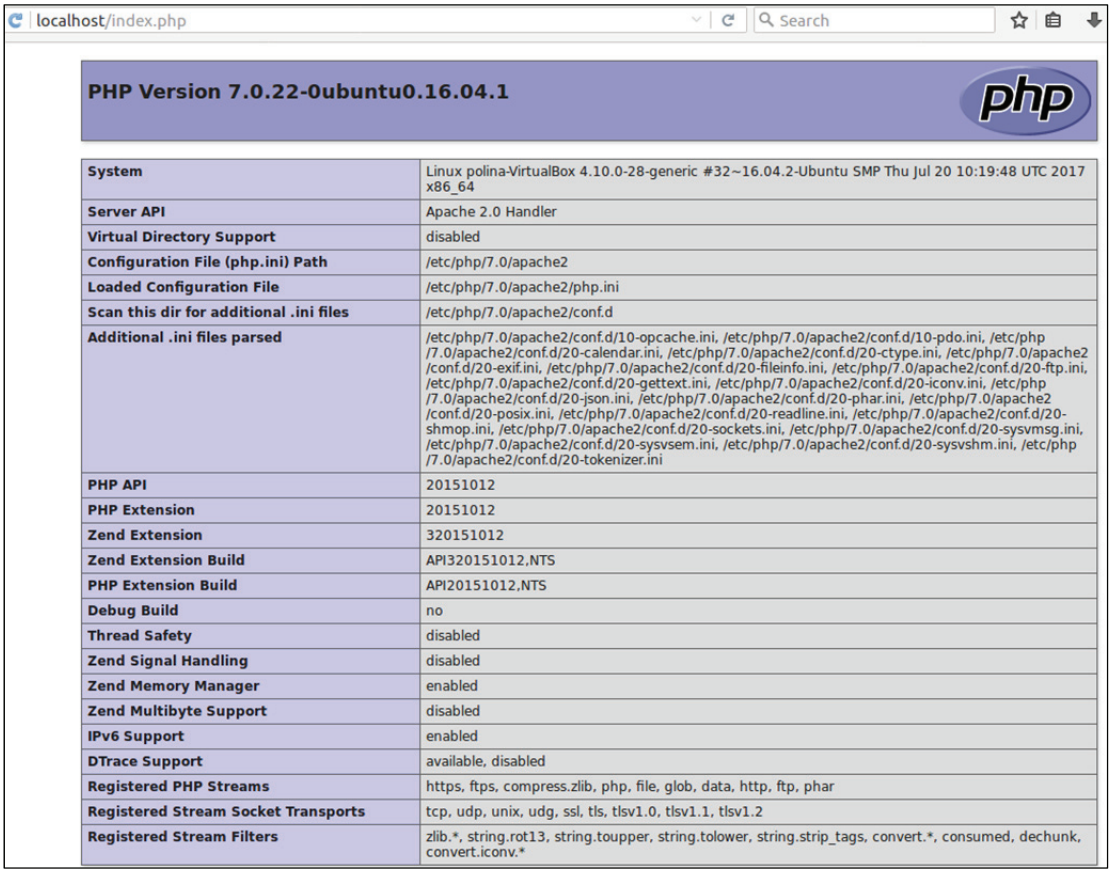


Рис. 8.20. Код страницы



System	Linux polina-VirtualBox 4.10.0-28-generic #32~16.04.2-Ubuntu SMP Thu Jul 20 10:19:48 UTC 2017 x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS
PHP Extension Build	API20151012,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

Рис. 8.21. Страницы пользователя

8.2.4. Модули mod_perl, mod_fastcgi, SuExec, PHP-FPM

Mod_perl – дополнительный модуль для веб-сервера Apache, внедряющий интерпретатор языка Perl в Apache и позволяющий избежать значительных накладных расходов на запуск Perl для обработки каждого запроса.

Для установки данного модуля необходимо поставить пакет libapache2-mod-perl (рис. 8.22).

```
polina@polina-VirtualBox:~$ sudo apt-get install libapache2-mod-perl2
[sudo] пароль для polina:
Чтение списков пакетов… Готово
Построение дерева зависимостей
Чтение информации о состоянии… Готово
Будут установлены следующие дополнительные пакеты:
  libapache2-reload-perl libbsd-resource-perl libdevel-symdump-perl
```

Рис. 8.22. Установка libapache2-mod-perl

После установки проверим работоспособность: напишем скрипт (рис. 8.23) и запустим в консоли (рис. 8.24).

test.cgi

```
#!/usr/bin/perl
use strict;
use warnings;

print qq(Content-type: text/plain\n\n);

print "Hello from perl mod\n";
```

Рис. 8.23. Скрипты проверки работоспособности Perl

```
root@polina-VirtualBox:/#  /var/www/html/perl/test.pl
Content-type: text/plain

Hello from perl mod
root@polina-VirtualBox:/#
```

Рис. 8.24. Результат выполнения скрипта

CGI (Common Gateway Interface – общий интерфейс шлюза) – это стандарт, который описывает, как веб-сервер должен запускать прикладные программы (скрипты), как должен передавать им параметры HTTP-запроса, как программы должны передавать результаты своей работы веб-серверу. Прикладную программу, взаимодействующую с веб-сервером по протоколу CGI, принято называть шлюзом, хотя более распространено название CGI-скрипт или CGI-программа.

В качестве CGI-программ могут использоваться программы/скрипты, написанные на любых языках программирования, как на компилируемых, так и на скриптовых, и даже на shell.

Основной момент: CGI это не язык программирования и не отдельная программа! Это просто протокол (стандарт, спецификация, соглашение, набор правил).

Для установки данного модуля необходимо установить пакет libapache2-mod-fastcgi (рис. 8.25). В файл конфигурации следует записать строки, приведенные на рис. 8.26.

```
polina@polina-VirtualBox:~$ sudo apt-get install libapache2-mod-fastcgi
Чтение списков пакетов… Готово
Построение дерева зависимостей
Чтение информации о состоянии… Готово
НОВЫЕ пакеты, которые будут установлены:
  libapache2-mod-fastcgi
обновлено 0, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 38 пакетов н
Необходимо скачать 48,9 kБ архивов.
После данной операции, объём занятого дискового пространства возрастёт на 209 kB.
Пол:1 http://by.archive.ubuntu.com/ubuntu xenial/multiverse amd64 libapache2-mod-fastcgi
910052141-1.2 [48,9 kB]
```

Рис. 8.25. Установка libapache2-mod-fastcgi

```
ScriptAlias /perl/ /var/www/html/perl/
<Directory "/var/www/html/perl">
AllowOverride None
Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
Require all granted
</Directory>
```

Рис. 8.26. Файл конфигурации

В результате мы можем запуспить наш скрипт в браузере (рис. 8.27).

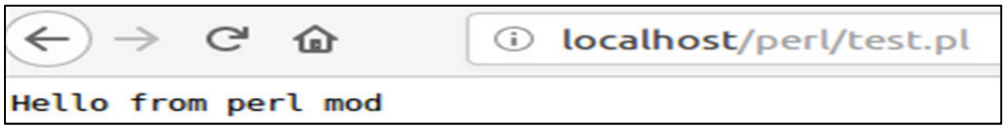


Рис. 8.27. Итог выполнения test.pl в браузере

SuExec – это модуль веб-сервера Apache, который позволяет запускать CGI и аналогичные собственные или сторонних разработчиков скрипты/программы внутри веб-папки домена от имени вполне конкретного пользователя (которому данная папка/домен принадлежат), а не от пользователя/группы, от имени которого работает непосредственно сам Apache веб-сервер.

Для начала установим нужные пакеты (рис. 8.28) и активируем необходимые модули (рис. 8.29). После этого перезапустим Apache.

```
polina@polina-VirtualBox:~$ sudo apt-get install libapache2-mod-fcgid apache2-suexec-custom -y
Чтение списков пакетов… Готово
Построение дерева зависимостей
Чтение информации о состоянии… Готово
НОВЫЕ пакеты, которые будут установлены:
  apache2-suexec-custom libapache2-mod-fcgid
обновлено 0, установлено 2 новых пакетов, для удаления отмечено 0 пакетов, и 70 пакетов не обновлено.
Необходимо скачать 85,6 kБ архивов.
После данной операции, объём занятого дискового пространства возрастёт на 403 kB.
Пол:1 http://by.archive.ubuntu.com/ubuntu xenial/universe amd64 libapache2-mod-fcgid amd64 1:2.3.9-1 [
70,5 kB]
Пол:2 http://by.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 apache2-suexec-custom amd64 2.
```

Рис. 8.28. Установка пакетов

```
a2enmod rewrite
a2enmod suexec
a2enmod include
a2enmod fcgid
```

Рис. 8.29. Активация модулей

Теперь создадим виртуальные хосты для примера www.test1.com (с корнем документа /var/www/test1/web) и www.test2.com (с корнем документа /var/www/test2/web). Для начала создадим пользователей и группы (рис. 8.30 и 8.31).

```
root@polina-VirtualBox:/# useradd -s /bin/false -d /var/www/exam1 -m -g exam1 exam1
root@polina-VirtualBox:/# useradd -s /bin/false -d /var/www/exam2 -m -g exam2 exam2
```

Рис. 8.30. Создание пользователей

```
root@polina-VirtualBox:/# groupadd exam1
root@polina-VirtualBox:/# groupadd exam2
```

Рис. 8.31. Создание групп

Теперь создадим корневые каталоги и назначим им соответствующих владельцев (рис. 8.32).

```
root@polina-VirtualBox:/# mkdir -p /var/www/exam1/web
root@polina-VirtualBox:/# mkdir -p /var/www/exam2/web
root@polina-VirtualBox:/# chown exam1:exam1 /var/www/exam1/web
root@polina-VirtualBox:/# chown exam2:exam2 /var/www/exam2/web
```

Рис. 8.32. Создание корневых каталогов

Поскольку интерпретатор PHP размещается за пределами корневого каталога suExeci и не позволяет использовать символьные ссылки, то единственным способом решить эту проблему будет создание скрипта-обертки для каждого веб-сайта в подкаталогах каталога /var/www. Важнейшей задачей скрипта-обертки будет запуск двоичного файла /usr/lib/cgi-bin/php. Этот скрипт должен принадлежать тем пользователю и группе, которые являются владельцами соответствующего веб-сайта, поэтому нам понадобятся отдельные скрипты для каждого сайта. Будем размещать скрипты-обертки в подкаталогах каталога /var/www/php-fcgi-scripts, т. е. в /var/www/php-fcgi-scripts/test1 и /var/www/php-fcgi-scripts/test2 (рис. 8.33 и 8.34).

```
root@polina-VirtualBox:/# mkdir -p /var/www/test1.com/public_html
root@polina-VirtualBox:/# mkdir -p /var/www/test2.com/public_html
```

Рис. 8.33. Создание каталогов

```
root@polina-VirtualBox:/# nano /var/www/php-fcgi-scripts/exam1/php-fcgi-starter
root@polina-VirtualBox:/# nano /var/www/php-fcgi-scripts/exam2/php-fcgi-starter
```

Рис. 8.34. Создание файлов-оберток

В данные файлы запишем скрипт, в котором строка PHPRC указывает на каталог, где размещен файл php.ini (т. е. /etc/php5/cgi транслируется в /etc/php5/cgi/php.ini). PHP_FCGI_MAX_REQUESTS задает максимальное число запросов, после обработки которых процесс fcgid будет остановлен и запущен заново. PHP_FCGI_CHILDREN определяет число дочерних процессов PHP, которые будут запущены (рис. 8.35).

```
#!/bin/sh
PHPRC=/etc/php/7.0/cgi/
export PHPRC
export PHP_FCGI_MAX_REQUESTS=5000
export PHP_FCGI_CHILDREN=8
exec /usr/lib/cgi-bin/php
```

Рис. 8.35. Скрипт для запуска процесса fcgid

Скрипты php-fcgi-starter должны быть исполнимыми; кроме того, они (и каталоги, в которых скрипты размещаются) должны принадлежать тем же пользователю и группе, которым принадлежит веб-сайт (рис. 8.36).

```
root@polina-VirtualBox:/# chmod 755 /var/www/php-fcgi-scripts/exam1/php-fcgi-starter
root@polina-VirtualBox:/# chmod 755 /var/www/php-fcgi-scripts/exam2/php-fcgi-starter
root@polina-VirtualBox:/# chown -R  exam1:exam1 /var/www/php-fcgi-scripts/exam1
root@polina-VirtualBox:/# chown -R  exam2:exam2 /var/www/php-fcgi-scripts/exam2
```

Рис. 8.36. Права пользователя и группы

Теперь настроим виртуальные хосты www.test1.com и www.test2.com в конфигурации Apache (рис. 8.37 и 8.38). Перезагрузим сервер.

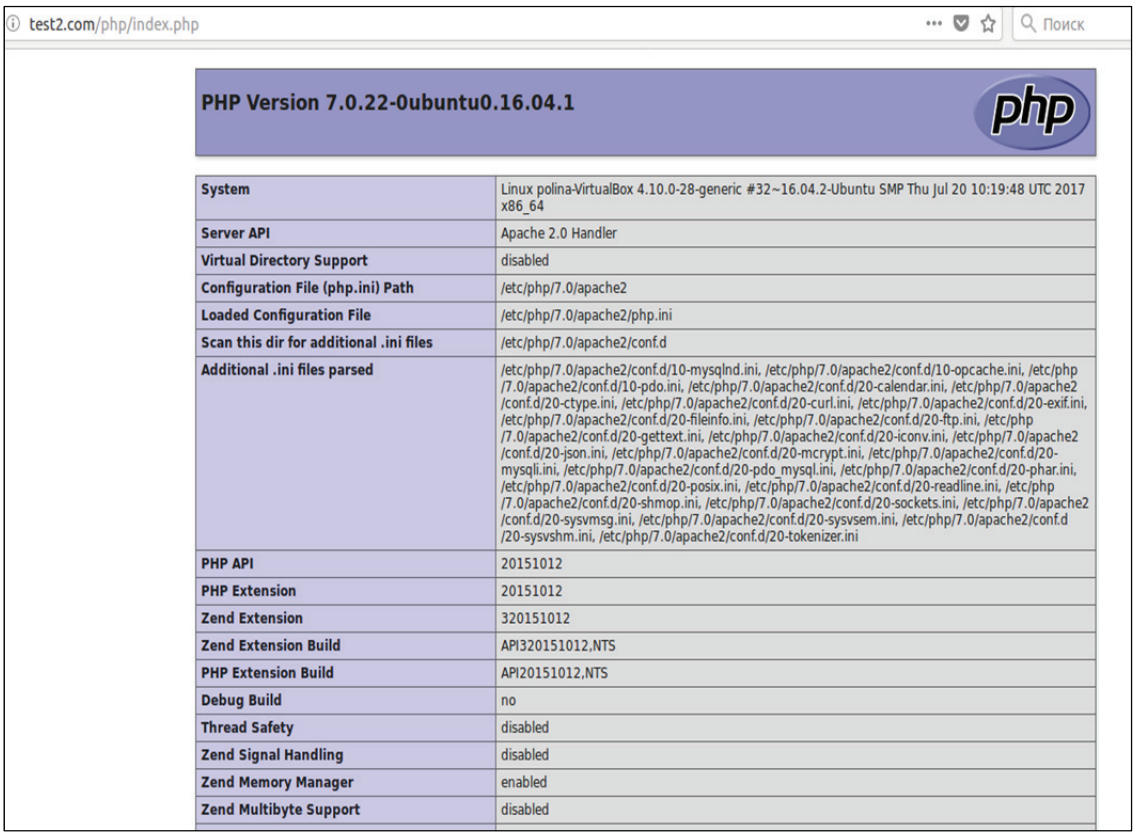
```
root@polina-VirtualBox:/# nano /etc/apache2/sites-available/exam1
root@polina-VirtualBox:/# nano /etc/apache2/sites-available/exam2
```

Рис. 8.37. Каталоги виртуальных хостов

```
<IfModule mod_fastcgi.c>
AddHandler php5-fcgi .php
Action php /php5-fcgi
Alias /php5-fcgi var/www/test2/php
FastCgiExternalServer /usr/lib/cgi-bin/php5-fcgi -host 127.0.0.1:9000 -pass-header Authorization
</IfModule>
```

Рис. 8.38. Скрипт php-fcgi-starter

На данном этапе можем проверить, как работает сервер. Для этого создадим небольшой файл, например, для сайта www.test2.com и вызовем из браузера http://www.test2.com/php/info.php (рис. 8.39).



System	Linux polina-VirtualBox 4.10.0-28-generic #32~16.04.2-Ubuntu SMP Thu Jul 20 10:19:48 UTC 2017 x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-curl.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-mcrypt.ini, /etc/php/7.0/apache2/conf.d/20-mysqli.ini, /etc/php/7.0/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS
PHP Extension Build	API20151012,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled

Рис. 8.39. Результат работы сервера

PHP-FPM (FastCGI Process Manager) – это альтернативная реализация PHP с некоторыми дополнительными функциями.

Для того чтобы Apache работал с PHP-FPM, нам понадобятся следующие строки (рис. 8.40) и перезагрузка сервера.

```
<IfModule mod_fastcgi.c>
AddHandler php5-fcgi .php
Action php5-fcgi /php5-fcgi
Alias /php5-fcgi /usr/lib/cgi-bin/php5-fcgi
FastCgiExternalServer /usr/lib/cgi-bin/php5-fcgi -host 127.0.0.1:9000 -pass-header Authorization
</IfModule>
```

Рис. 8.40. Строки для работы PHP-FRM

Чтобы проверить работу, можно в каталоге /var/www создать файл info.php (рис. 8.41 и 8.42)

```
<?php
phpinfo();
?>
```

Рис. 8.41. Файл info.php

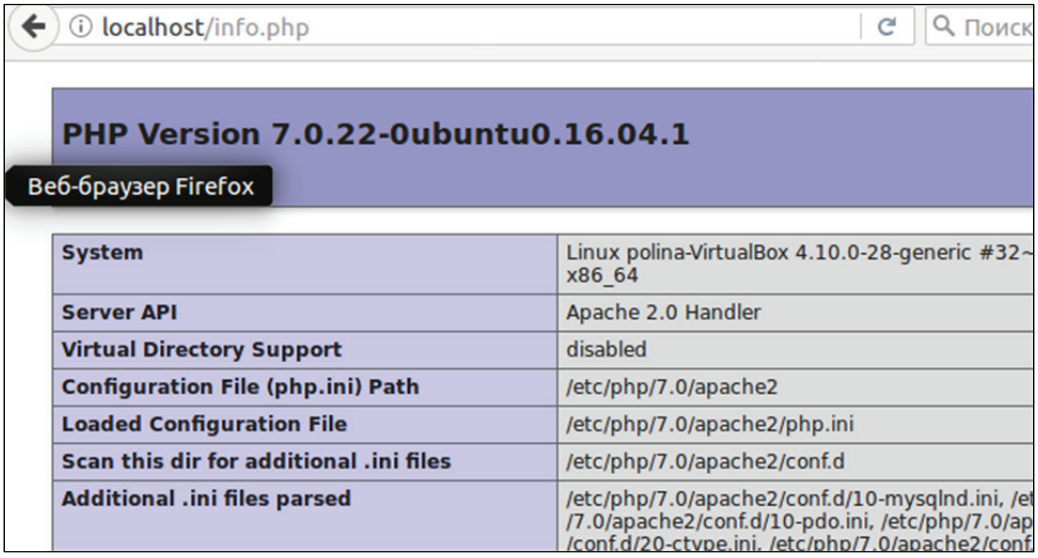


Рис. 8.42. Страница info.php

Лабораторная работа № 14–15

***Цель*:** получение практических навыков по развертыванию веб-сервера под управлением Apache (версия 2.x), включая установку httpd, основные настройки и конфигурирование виртуальных хостов.

***Задание*:** необходимо установить веб-сервер Apache в основной конфигурации в качестве системного сервиса (runlevel 3 и 5), проверить правильность установки, выполнить настройку веб-сервера, протестировать работу веб-сервера. Следует настроить два-три именованных виртуальных хоста, доступных с любого компьютера в пределах дисплейного класса, где проводится лабораторная работа.

Необходимо произвести установку надстроек для сервера и установку PHP интерпретатора для обработки php-скриптов.

Следует установить сервер баз данных MySQL и административную панель phpMyAdmin, а также создать на сервере баз данных три учетных записи с одноименными базами данных. Необходимо установить на именованные виртуальные хосты сайты трех видов: систему управления контентом, форум и облачное хранилище. Следует наполнить сайты содержанием на тему администрирования информационных систем (минимум 5 иллюстрированных статей в системе управления контентом, 5 тем на форуме и 5 презентаций в облачном хранилище). Необходимо найти и разместить на установленных сайтах адаптивные шаблоны оформления, добавить модули интерации сайтов (например, вывод в системе управления контентом последних тем с форума). Следует продемонстрировать работу сайтов на веб-сервере Apache.

Раздел 9
ИСПОЛЬЗОВАНИЕ ВЕБ-СЕРВЕРА NGINX

9.1. Введение в веб-сервер Nginx

Nginx – веб-сервер и почтовый прокси-сервер, работающий на Unix-подобных операционных системах (тестировалась сборка и работа на FreeBSD, OpenBSD, Linux, Solaris, Mac OS X, AIX и HP-UX). Начиная с версии 0.7.52 появилась экспериментальная бинарная сборка под Microsoft Windows. Nginx позиционируется производителем как простой, быстрый и надежный сервер, не перегруженный функциями.

Применение Nginx целесообразно прежде всего для статических веб-сайтов и как прокси-сервера перед динамическими сайтами.

Nginx появился на сцене позднее Apache, по этой причине его разработчик был лучше осведомлен о проблемах конкурентности, с которыми сталкиваются сайты при масштабировании. Благодаря этим знаниям Nginx изначально был спроектирован на базе асинхронных неблокирующих event-driven алгоритмов.

Nginx создает процессы-воркеры, каждый из которых может обслуживать тысячи соединений. Воркеры достигают такого результата благодаря механизму, основанному на быстром цикле, в котором проверяются и обрабатываются события. Отделение основной работы от обработки соединений позволяет каждому воркеру заниматься своей работой и отвлекаться на обработку соединений только тогда, когда произошло новое событие.

Каждое соединение, обрабатываемое воркером, помещается в event loop вместе с другими соединениями. В этом цикле события обрабатываются асинхронно, позволяя обрабатывать задачи в неблокирующей манере. Когда соединение закрывается, оно удаляется из цикла.

Этот подход к обработке соединений позволяет Nginx невероятно масштабироваться при ограниченных ресурсах. Поскольку сервер однопоточный и он не создает процессы под каждое соединение, использование памяти и CPU относительно равномерне, даже при высоких нагрузках.

Nginx не имеет возможности самостоятельно обрабатывать запросы к динамическому контенту. Для обработки запросов к PHP или другому динамическому контенту Nginx должен передать запрос внешнему процессору для исполнения, подождать, пока ответ будет сгенерирован, и получить его. Затем результат может быть отправлен клиенту.

Для администраторов это означает, что нужно настроить взаимодействие Nginx с таким процессором, используя один из протоколов, который известен Nginx (http, FastCGI, SCGI, uWSGI, memcache). Это может немного усложнить процесс настройки, в особенности когда вы будете пытаться предугадать, какое число соединений разрешить, так как будет использоваться дополнительное соединение с процессором на каждый пользовательский запрос.

Однако этот метод имеет и свои преимущества. Так как интерпретатор не встроен в каждый воркер, то оверхед, связанный с этим, будет иметь место только при запросах к динамическому контенту. Статический контент будет возвращен клиенту простым способом, и запросы к интерпретатору будут выполняться только тогда, когда они нужны. Apache тоже может работать в такой манере, но тогда это лишит его всех преимуществ, описанных в предыдущем разделе.

9.2. Установка и развертывание программного обеспечения

9.2.1. Установка Nginx

Установка сервера Nginx на ОС Linux производится так же, как и большинство других программных продуктов, с помощью пакетного менеджера (apt, apt-get, aptitude). В нашем случае мы использовали пакетный менеджер apt.

Nginx есть в официальных репозиториях Ubuntu, но для установки самой новой версии необходимо добавить PPA:

```
$ sudo apt-add-repository ppa:nginx/stable
```

Дальше следует обновить списки пакетов из репозиториев:

```
$ sudo apt update
```

После чего необходимо установить Nginx:

```
$ sudo apt install nginx
```

Затем следует добавить программу в автозагрузку, чтобы она запускалась автоматически:

```
$ sudo systemctl enable nginx
```

Все, теперь Nginx установлен, и уже в браузере при заходе на localhost можно заметить рабочий сервер. На рис. 9.1 показана страничка, которая запустится в браузере.

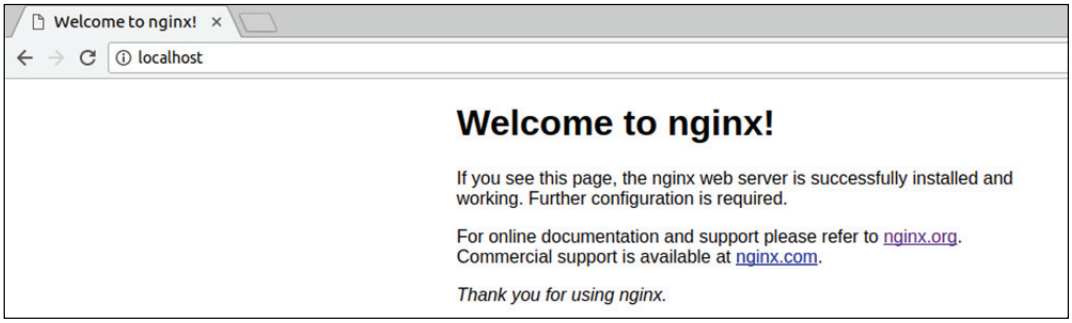


Рис. 9.1. Запущенное веб-приложение

Для проверки запуска Nginx можно использовать команду:

$ systemctl

На рис. 9.2 можно увидеть список запущенных приложений, в котором можно найти nginx.service со статусом loaded active running.

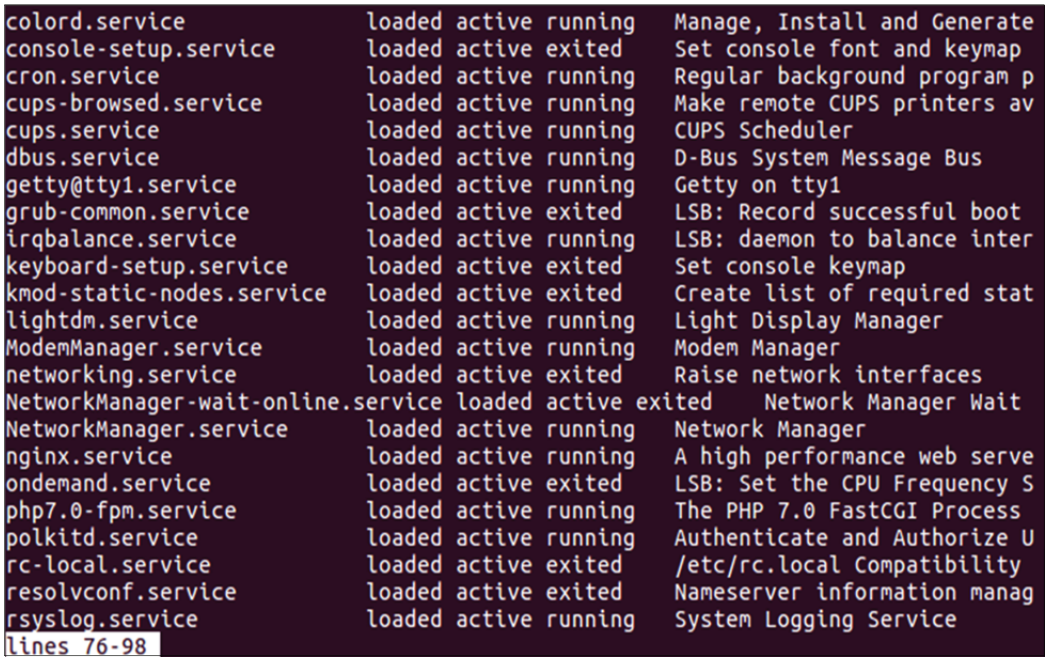


Рис. 9.2. Список запущенных приложений

9.2.2. Разбор конфигурационного файла

Для настройки Nginx предусмотрен файл конфигурации, который находится в директории /etc/nginx/nginx.conf. Также для настройки виртуальных хостов имеются следующие директории:

– /etc/nginx/sites-available/* – файлы конфигурации для виртуальных хостов;

– /etc/nginx/sites-enable/* – файлы конфигурации активированных хостов.

Обычно в директории sites-enable содержатся только ссылки на sites-available. Это было придумано для того, чтобы отключать хост, при этом конфиг не обязательно было удалять.

Рассмотрим главный файл конфигурации nginx.conf, который разбит на следующие секции:

```
глобальные опции
events{}
http{
server{
location{}
}
server{}
}
mail{}
```

Секция «глобальные опции» отвечает за работу всей программы. Секция events содержит настройки для работы с сетью. Секция http содержит настройки веб-сервера, а также должна содержать секцию server для настройки каждого хоста либо включать в себя файлы конфигурации из директории sites-enable. Секция location может находиться только внутри секции server и содержит настройки только для определенного запроса. Секция mail предназначена для настройки почтового прокси. На рис. 9.3 показан пример файла конфигурации.

Перед тем как перейти к опциям, нужно сказать еще пару слов о синтаксисе строки в конфигурационном файле. Он выглядит вот так:

```
параметр значение дополнительное_значение…;
```

Строка должна обязательно заканчиваться «;», а все открытые скобки { должны быть закрыты.

Теперь, когда мы немного изучили глобальную структуру, можно переходить к рассмотрению самих параметров. Глобальных опций не так уж много:

1) user – пользователь, от имени которого будет работать программа;

2) worker_processes – устанавливает, сколько процессов нужно запускать для параллелизации работы программы. Следует запускать не больше процессов, чем у вас есть ядер. Можно установить параметр auto, и тогда программа определит это число сама;

3) pid – файл pid-программы;

4) worker_rlimit_nofile – указывает максимальное количество файлов, которые может открыть программа. Рассчитывается как worker_processes * worker_connections * 2.

```
dmitry@superman:/etc/nginx$ cat nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
        worker_connections 768;
        # multi_accept on;
}

http {

        ##
        # Basic Settings
        ##

        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 65;
        types_hash_max_size 2048;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;

        ##
        # SSL Settings
        ##

        ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # Dropping SSLv3, ref: POODLE
        ssl_prefer_server_ciphers on;

        ##
        # Logging Settings
        ##

        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;

        ##
        # Gzip Settings
        ##

        gzip on;
        gzip_disable "msie6";
```

Рис. 9.3. Файл конфигурации nginx.conf

С глобальными опциями закончили, их было не так много, и они не такие интересные. Куда важнее в плане оптимизации опции с секции events:

– worker_connections – количество соединений, которые программа может обрабатывать одновременно на одном процессе. Если умножить worker_process на этот параметр, то мы получим максимальное

количество пользователей, которые могут подключиться к серверу одновременно. Рекомендуется устанавливать значение от 1024 до 4048;

– multi_accept – разрешает принимать много подключений одновременно. Можно установить параметр on или off;

– use – способ работы с сетевым стеком. По умолчанию используется poll, но для Linux эффективнее использовать epoll.

Дальше переходим к самой главной секции http. Здесь опций намного больше:

1) sendfile – использует метод отправки данных. Значение on;

2) tcp_nodelay, tcp_nopush – отправляет заголовки и начало файла одним пакетом. Значение on;

3) keepalive_timeout – таймаут ожидания, перед тем как keepalive-соединение будет разорвано. По умолчанию значение 65 с, но можно уменьшить до 10 с;

4) keepalive_requests – максимальное количество keepalive-соединений от одного клиента. Рекомендовано 100;

5) reset_timedout_connection – разрывает соединения после таймаута. Значение on;

6) open_file_cache – кеширует информацию об открытых файлах. Строчка настройки выглядит вот так: open_file_cache max=200000 inactive=20s, где max – максимальное количество файлов в кеше, время кеширования;

7) open_file_cache_valid – указывает, по истечении какого времени нужно удалить информацию из кеша. Например: open_file_cache_valid 30s;

8) open_file_cache_min_uses – кеширует информацию о файлах, которые были открыты как минимум указанное количество раз;

9) open_file_cache_errors – кеширует информацию об отсутствующих файлах. Значение on.

В секции server основными параметрами являются:

– listen 80 – указывает, что нужно ожидать подключения на порту 80, может также содержать опцию default-server, которая означает, что этот домен будет открываться, если домен не был задан в запросе;

– root /var/www/html – директория, в которой находятся файлы сайта;

– index index.html – страница, которая будет открываться по умолчанию;

– server_name – доменное имя сайта;

– access_log – файл для записи лога запросов к серверу. Может использоваться как глобально в секции http, так и для определенного типа файлов в location;

– error_log – лог ошибок веб-сервера. Может принимать дополнительный параметр, указывающий подробность лога: warn – максимум, crit – только критические ошибки.

Пример файла конфигурации виртуального хоста можно увидеть на рис. 9.4.

```
dmitry@superman:/etc/nginx/sites-available$ cat default
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
        listen 80;
        listen [::]:80;

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html;

        # Add index.php to the list if you are using PHP
        index index.html index.htm index.nginx-debian.html;

        server_name default.local;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
```

Рис. 9.4. Файл конфигурации виртуального хоста

Секции http и location не применяются в nginx.conf, а используются при написании конфигурационного файла для виртуальных хостов, поскольку они затем включаются в главный конфигурационный файл.

Секция location описывает поведение сервера для определенных директорий и файлов. Ее синтаксис:

location адрес

В качестве адреса может использоваться как прямой запрос относительно корня сервера, так и регулярные выражения. В случае использования регулярных выражений перед ним ставится символ «~». Ниже представлен список возможных параметров:

1) allow – разрешает доступ к местоположению для пользователей. Значение all – для всех, также можно указать ip или подсеть;

2) deny – запрещает доступ к местоположению. Значение all – для всех;

3) try-files – пытается открыть файлы в определенном порядке, открывает первый обнаруженный файл. Например, такая конструкция: $uri $uri/index.html $uri.html = 404; сначала пытается открыть $uri, затем index.html, если не найден $uri.html, и лишь потом, если ни одного из предложенных файлов не существует, выдает ошибку 404;

4) expires – задает время кеширования браузером отданного элемента. Например, 1d – один день, 2h – 2 ч, 30s – 30 с.

9.2.3. Настройка Nginx в качестве front-end к Apache

Поскольку Nginx является сервером, который может работать только с статическими данными, он обычно используется в связке с каким-то модулем (Apache, PHP-FPM). В данном пункте рассмотрим настройку Nginx в качестве front-end к Apache.

Установку Nginx и Apache производим с помощью команды, представленной на рис. 9.5.

```
dmitry@superserver:/etc/apache2$ sudo apt install apache2 nginx
Чтение списков пакетов… Готово
Построение дерева зависимостей
Чтение информации о состоянии… Готово
```

Рис. 9.5. Установка Apache

Можно отметить, что при установке может возникнуть ошибка с установкой ПО, тогда нужно будет устанавливать по очереди, предварительно изменив порт прослушивания, чтобы можно было автоматически запустить другую службу. Как изменить порт прослушивания, будет описано ниже.

После установки Apache необходимо, чтобы порт прослушивания был отличный от 80 (обычно 8080). Конфигурационный файл должен находиться в директории /etc/apache2/ports.conf. Меняем значение Listen на другое. Также дописываем значение NameVirtualHost *:8080. Измененный файл конфигурации представлен на рис. 9.6.

В настройки виртуального хоста, который будет являться прокси, добавим следующие строки (рис. 9.7).

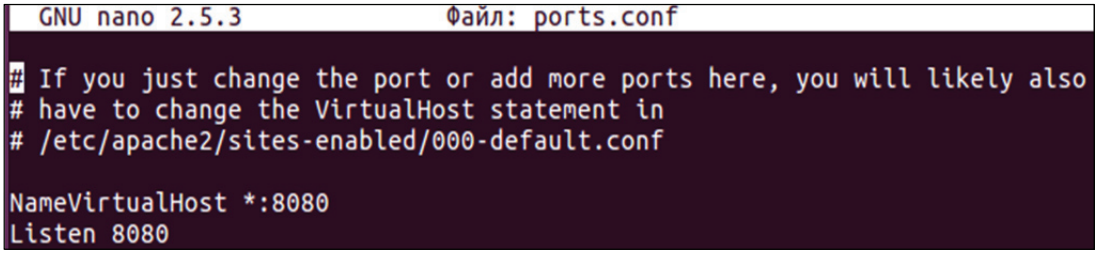


Рис. 9.6. Измененный файл конфигурации для Apache

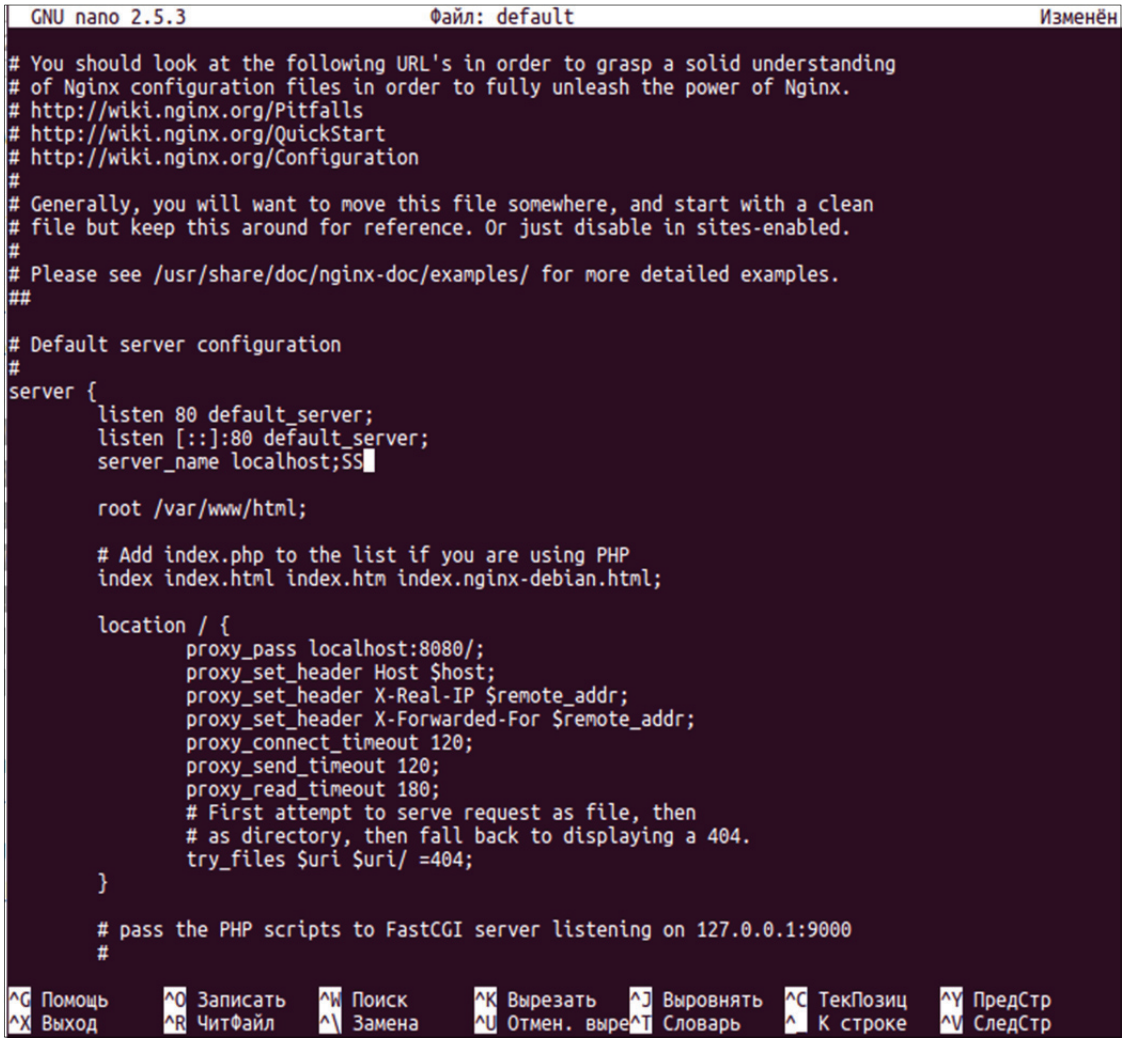


Рис. 9.7. Файл конфигурации для виртуального хоста-прокси

Заметим, что в параметрах server_name и proxy_pass необходимо писать имя вашего домена. В нашем случае это localhost.

После изменения всех конфигов следует перезапустить программы командами (рис. 9.8):

```
/etc/init.d/apache2 restart
/etc/init.d/nginx restart
```

```
dmitry@superserver:/etc/nginx/sites-available$ /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
dmitry@superserver:/etc/nginx/sites-available$ /etc/init.d/nginx restart
[ ok ] Restarting nginx (via systemctl): nginx.service.
dmitry@superserver:/etc/nginx/sites-available$ /etc/init.d/apache2 stop
```

Рис. 9.8. Перезапуск программ

Теперь при заходе на localhost нас перенаправляют на сайт сервера Apache. Запущенный сайт можно увидеть на рис. 9.9.

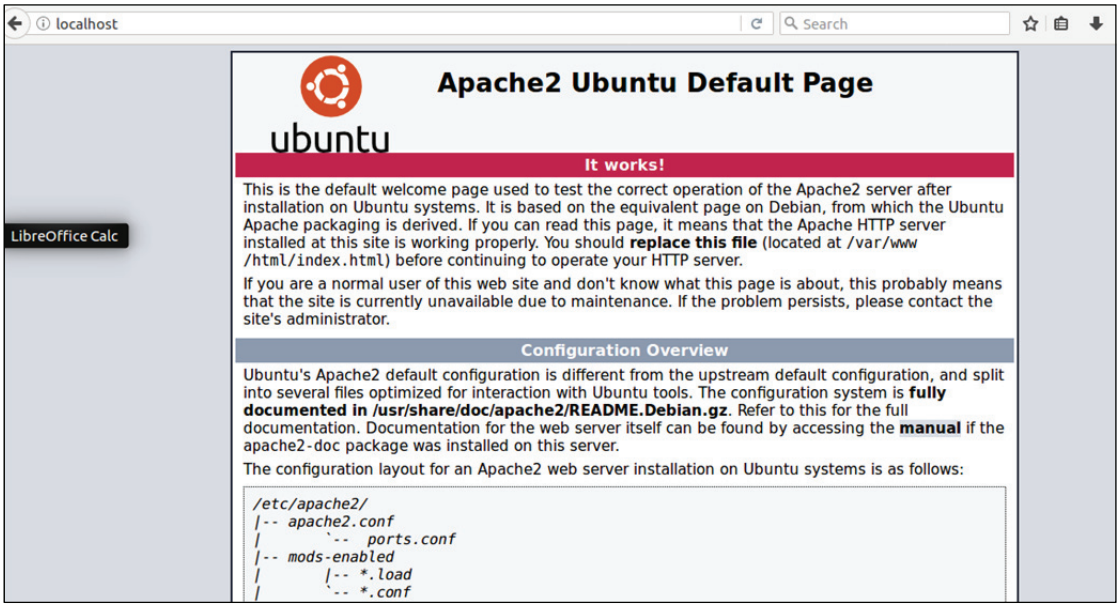


Рис. 9.9. Запущенный сайт

Если выключить сервер Nginx, то доступа к сайту не будет.

9.2.4. Настройка связи Nginx+PHP-FPM

После установки Nginx необходимо установить PHP-FPM. Чтобы проверить версию Nginx, можно воспользоваться командой, приведенной на рис. 9.10.

```
dmitry@superman:~$ nginx -v
nginx version: nginx/1.12.1
dmitry@superman:~$
```

Рис. 9.10. Проверка версии Nginx

Установка PHP-FPM производится при помощи команды:

```
$sudo apt install php-fpm
```

После установки можно проверить версию установленной программы командой, показанной на рис. 9.11.

```
dmitry@superman:~$ php-fpm7.0 -v
PHP 7.0.22-0ubuntu0.16.04.1 (fpm-fcgi)
Copyright (c) 1997-2017 The PHP Group
Zend Engine v3.0.0, Copyright (c) 1998-2017 Zend Technologies
    with Zend OPcache v7.0.22-0ubuntu0.16.04.1, Copyright (c) 1999-2017, by Zend
 Technologies
dmitry@superman:~$
```

Рис. 9.11. Проверка версии PHP-FPM

При необходимости можно запустить PHP-FPM командой:

```
$ sudo service php7.0-fpm start
```

Итак, базовая настройка PHP-FPM для работы нам подходит. Следует настроить виртуальный хост Nginx на то, чтобы он работал с PHP-FPM. Для этого нужно добавить в конфигурацию виртуального хоста следующие строки, приведенные на рис. 9.12.

```
server {
        listen 80; # порт, прослушивающий nginx
        server_name    example.local; # доменное имя, относящиеся к текущему вир
туальному хосту
        root  /var/www/example.local; # каталог в котором лежит проект, путь к т
очке входа


        index index.php;
        # add_header Access-Control-Allow-Origin *;


        # serve static files directly
        location ~* \.(jpg|jpeg|gif|css|png|js|ico|html)$ {
                access_log off;
                expires max;
                log_not_found off;
        }


        location / {
                # add_header Access-Control-Allow-Origin *;
                try_files $uri $uri/ /index.php?$query_string;
        }

        location ~* \.php$ {
        try_files $uri = 404;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/var/run/php/php7.0-fpm.sock; # подключаем сокет php-f
pm
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\.ht {
                deny all;
        }
}
```

Рис. 9.12. Файл конфигурации виртуального хоста
для работы с PHP-FPM

Нам необходимо добавить в тег location по адресу *.php, чтобы сервер понимал, что делать с файлами формата .php. И внутри подключить сокет PHP-FPM, который расположен в директории /var/run/php/php7.0-fpm.sock для обработки PHP-файлов.

В нашем виртуальном хосте настроен главный файл index.php в директории /var/www/example.local. Зайдем в директорию и создадим простой файл с PHP-кодом. Пример файла можно увидеть на рис. 9.13.

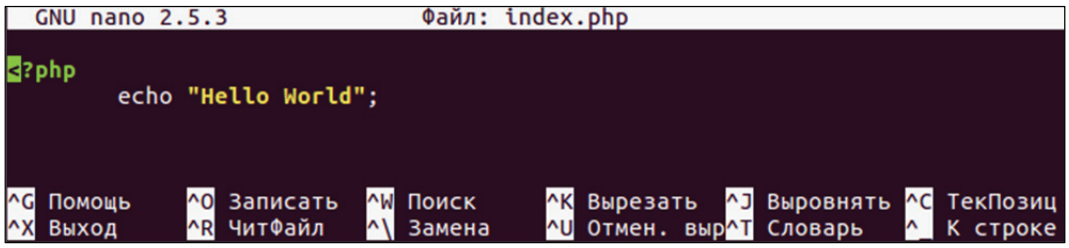


Рис. 9.13. Файл с PHP-кодом

Для того чтобы не было проблем с доступом к каталогу, следует дать ему полные права. Пример команды представлен на рис. 9.14.

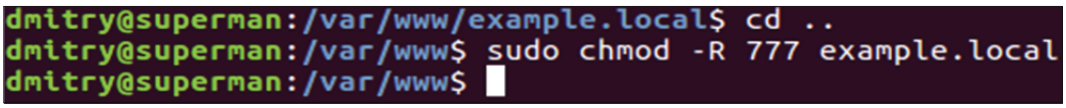


Рис. 9.14. Установка прав доступа к каталогу с содержимым сайта

После этого настройка завершена. Теперь можно проверить работоспособность, зайдя на сайт под вашим доменом. Пример работы приведен на рис. 9.15. На этом настройка PHP-FPM для сервера Nginx завершена.

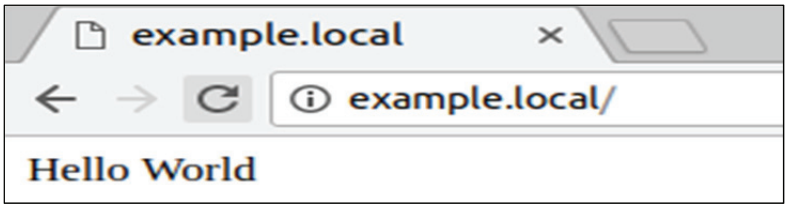


Рис. 9.15. Работа сайта, написанного с помощью PHP

9.2.5. Установка модуля ngx_pagespeed

Есть два варианта установки модуля ngx_pagespeed для сервера: с файлов из репозиториев ubuntu и исходных файлов.

При установке из репозиториев ubuntu первоначально необходимо скачать исходные файлы сервера с помощью команды:

```
$ sudo apt-get source nginx
```

Затем следует скачать исходные файлы ngx_pagespeed с помощью команд, представленных на рис. 9.16 и 9.17.

Далее переходим в каталог с загруженными исходными файлами Nginx и там в каталог /debian. В файле rules в секциях: light_configure_flage, full_configure_flage, extras_configure_flage добавляем --add module=path_to_module. Здесь path_to_module – путь к исходникам модуля. Пример файла показан на рис. 9.18 (см. на с. 148).

```
dmitry@superman:~$ #[check the release notes for the latest version]
dmitry@superman:~$ NPS_VERSION=1.12.34.2-stable
dmitry@superman:~$ cd
dmitry@superman:~$ wget https://github.com/pagespeed/ngx_pagespeed/archive/v${N
S_VERSION}.zip
--2017-12-12 22:27:08--  https://github.com/pagespeed/ngx_pagespeed/archive/v1.
2.34.2-stable.zip
Распознаётся github.com (github.com)... 192.30.253.113, 192.30.253.112
Подключение к github.com (github.com)|192.30.253.113|:443... соединение установ
ено.
HTTP-запрос отправлен. Ожидание ответа... 302 Found
Адрес: https://codeload.github.com/pagespeed/ngx_pagespeed/zip/v1.12.34.2-stabl
 [переход]
--2017-12-12 22:27:09--  https://codeload.github.com/pagespeed/ngx_pagespeed/zi
/v1.12.34.2-stable
Распознаётся codeload.github.com (codeload.github.com)... 192.30.253.120, 192.3
.253.121
Подключение к codeload.github.com (codeload.github.com)|192.30.253.120|:443...
оединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: нет данных [application/zip]
Сохранение в каталог: ««v1.12.34.2-stable.zip»».

v1.12.34.2-stable.z     [          <=>       ] 143,95K  63,8KB/s    in 2,3s

2017-12-12 22:27:13 (63,8 KB/s) - «v1.12.34.2-stable.zip» сохранён [147404]

dmitry@superman:~$ unzip v${NPS_VERSION}.zip
Archive:  v1.12.34.2-stable.zip
```

Рис. 9.16. Подключение для загрузки исходных данных ngx_pagespeed

```
dmitry@superman:~/ngx_pagespeed-1.12.34.2-stable$ NPS_RELEASE_NUMBER=${NPS_VERSI
ON/stable/}
dmitry@superman:~/ngx_pagespeed-1.12.34.2-stable$ psol_url=https://dl.google.com
/dl/page-speed/psol/${NPS_RELEASE_NUMBER}.tar.gz
dmitry@superman:~/ngx_pagespeed-1.12.34.2-stable$ [ -e scripts/format_binary_url
.sh ] && psol_url=$(scripts/format_binary_url.sh PSOL_BINARY_URL)
dmitry@superman:~/ngx_pagespeed-1.12.34.2-stable$ wget ${psol_url}
--2017-12-12 22:27:13--  https://dl.google.com/dl/page-speed/psol/1.12.34.2-x64.
tar.gz
Распознаётся dl.google.com (dl.google.com)... 216.58.205.238, 2a00:1450:4001:820
::200e
Подключение к dl.google.com (dl.google.com)|216.58.205.238|:443... соединение ус
тановлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 16968086 (16M) [application/x-tar]
Сохранение в каталог: ««1.12.34.2-x64.tar.gz»».

1.12.34.2-x64.tar.g 100%[===================>]  16,18M   594KB/s    in 33s

2017-12-12 22:27:47 (502 KB/s) - «1.12.34.2-x64.tar.gz» сохранён [16968086/16968
086]

dmitry@superman:~/ngx_pagespeed-1.12.34.2-stable$ tar -xzvf $(basename ${psol_ur
l})  # extracts to psol/
psol/
psol/include_history.txt
```

Рис. 9.17. Загрузка исходных данных ngx_pagespeed

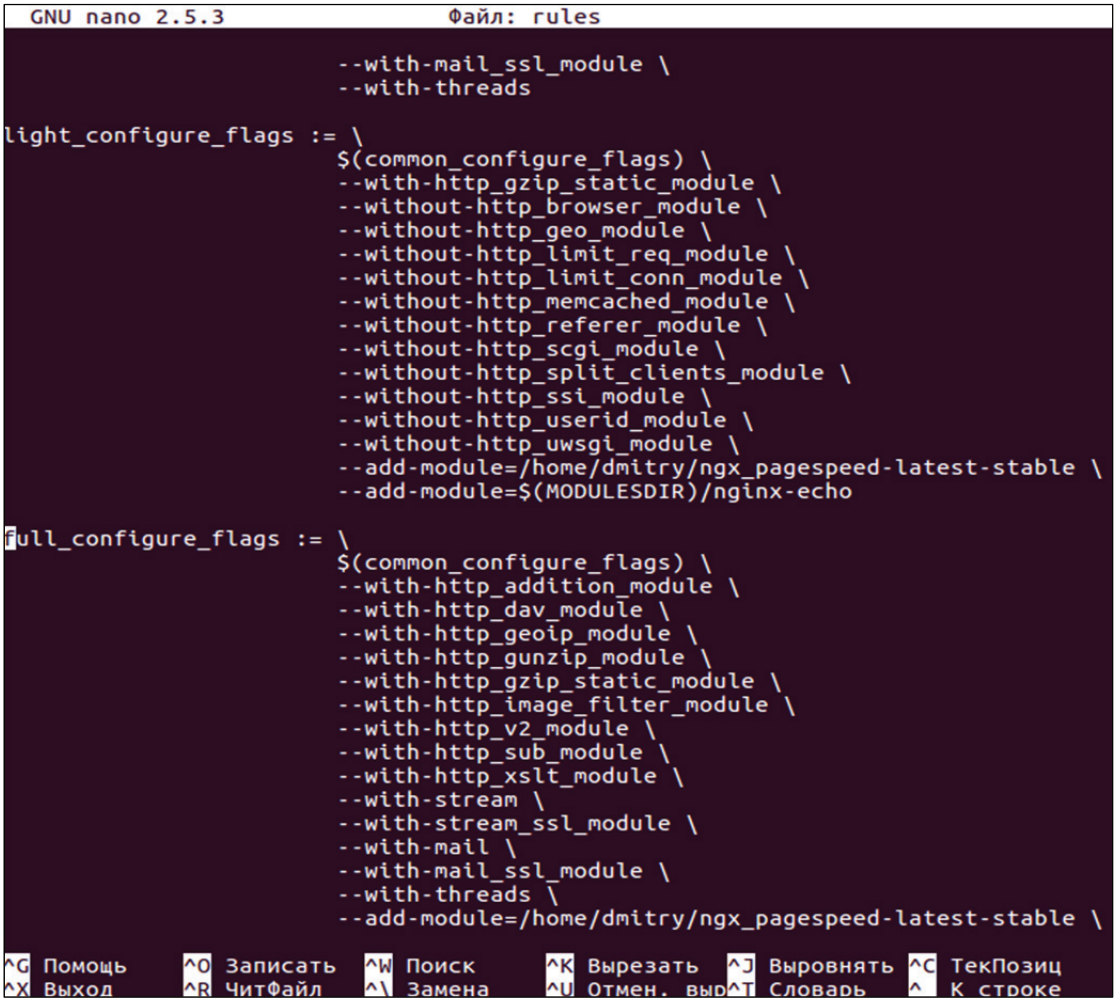


Рис. 9.18. Файл rules

Затем можем собрать пакет, для этого из каталога исходных файлов необходимо запустить команду, приведенную на рис. 9.19.

```
dmitry@superman:~/nginx-1.10.3/debian$ cd ..
dmitry@superman:~/nginx-1.10.3$ dpkg-buildpackage -b
```

Рис. 9.19. Сборка пакетов

В итоге в каталоге выше мы получаем deb-пакеты, которые затем можем установить при помощи команды:

```
dpkg -i
```

По окончании установки можем проверить, все ли установилось, введя команду nginx -V, и найти наш модуль ngx_pagespeed (рис. 9.20). На этом установка ngx_pagespeed завершена.

Второй вариант установки можно найти в документации на сайте: https://www.modpagespeed.com/doc/build_ngx_pagespeed_from_source.

Рис. 9.20. Проверка версии и установленных модулей Nginx

После установки нам необходимо настроить модуль, который настраивается как для всего сервера, так и для виртуального хостинга (зависит от того, в каком конфиге добавить включение модуля). Добавим поддержку модуля в виртуальном хосте. Для этого следует добавить строки в файл конфигурации виртуального хостинга в секцию server, который представлен на рис. 9.21.

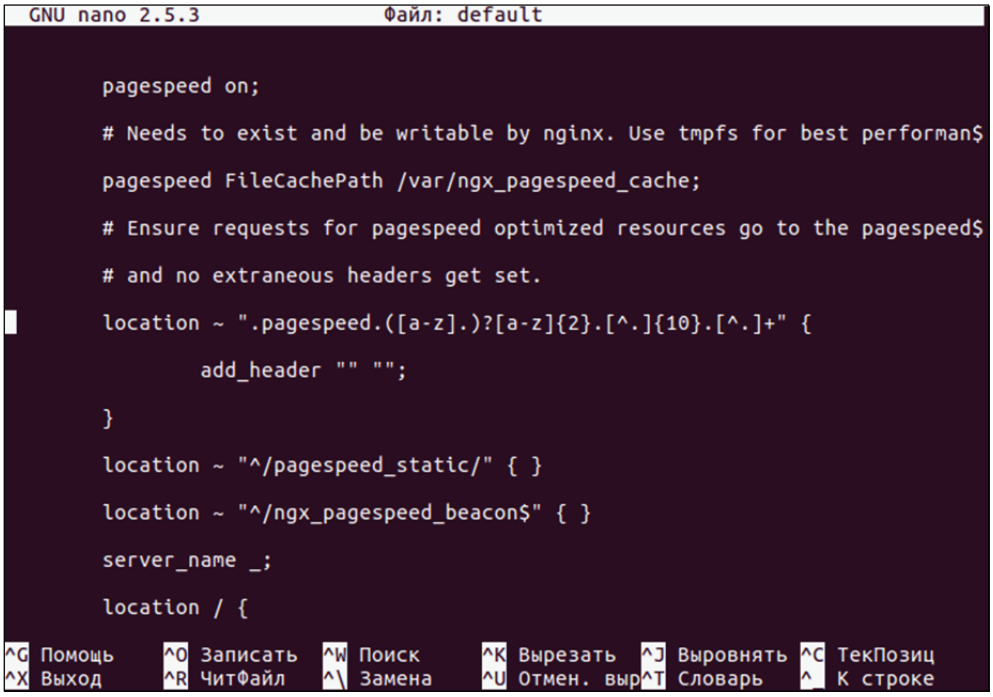


Рис. 9.21. Настройка модуля ngx_pagespeed для хоста

Далее необходимо перезапустить сервер Nginx командой, указанной на рис. 9.22.

```
dmitry@superman:~$ sudo /etc/init.d/nginx restart
[ ok ] Restarting nginx (via systemctl): nginx.service.
dmitry@superman:~$
```

Рис. 9.22. Перезапуск Nginx

Проверить работу модуля можно по заголовку ответа сервера X-Page-Speed (рис. 9.23).

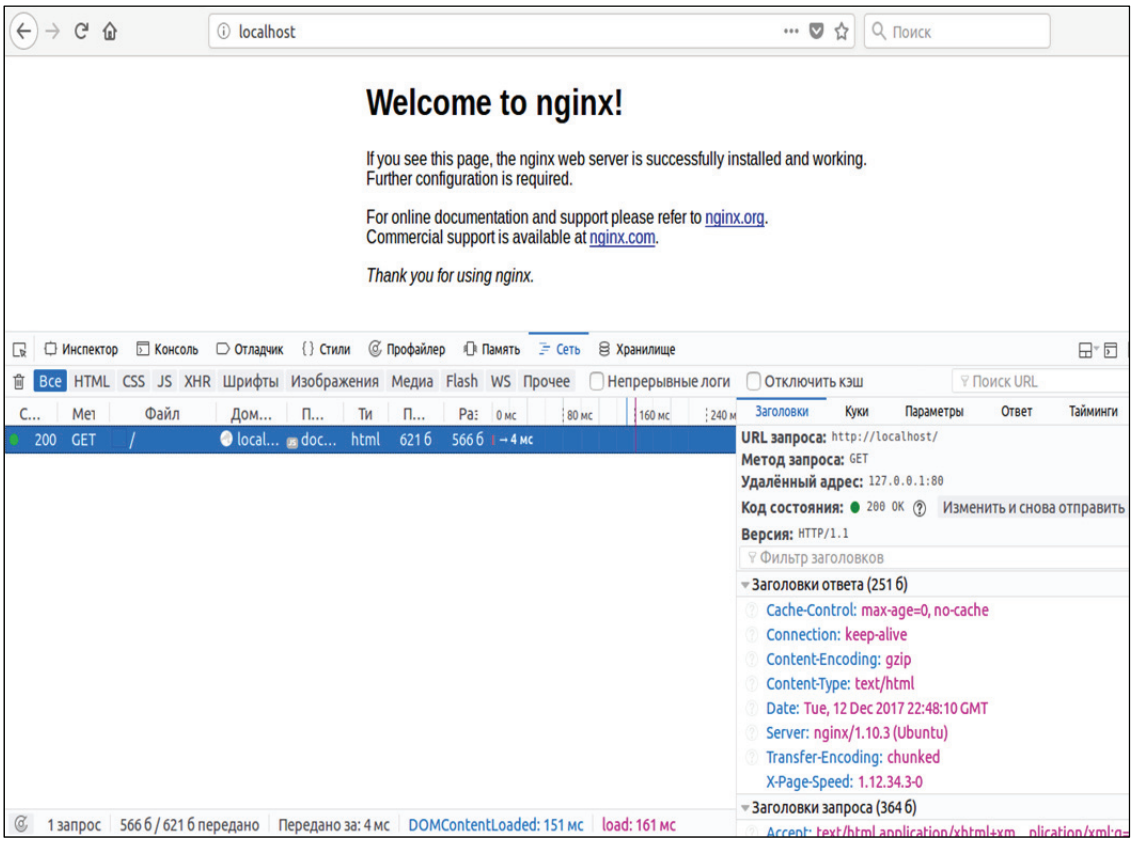


Рис. 9.23. Пример работы сайта

На этом базовая настройка модуля ngx_pagespeed завершена.

9.2.6. Балансировка нагрузки

Сервер Nginx можно настроить как балансировщик нагрузки с целью распределения нагрузки на несколько серверов в случае большой нагрузки.

Обычно при балансировке используются несколько серверов backend, на которых расположен наш сайт, и frontend сервер, к кото-

рому идет подключение клиенов. Пример связанных серверов представлен на рис. 9.24.

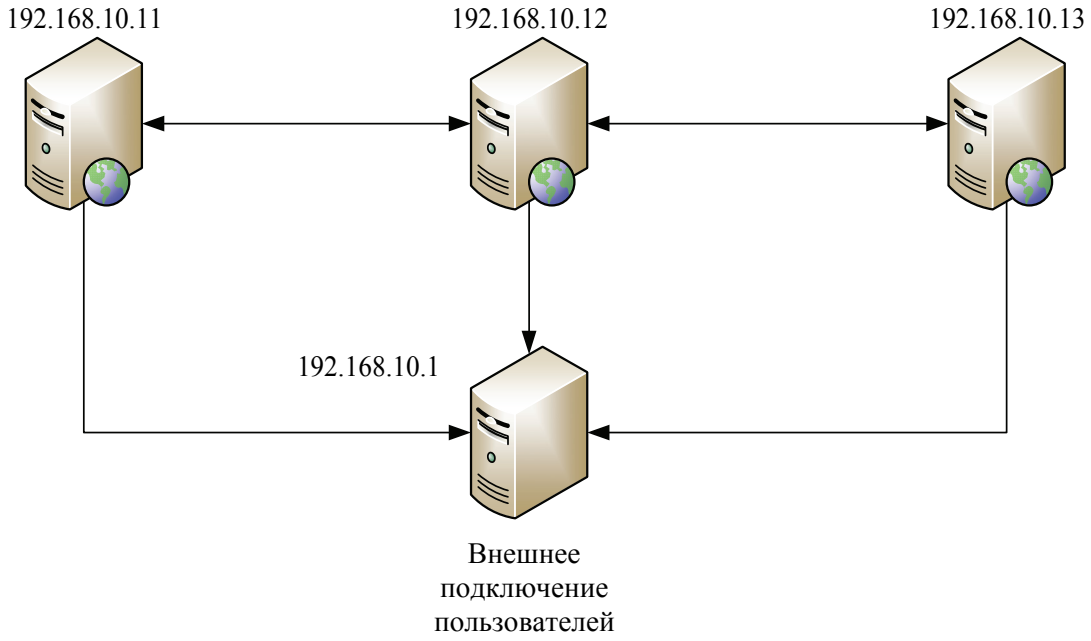


Рис. 9.24. Связка серверов для балансировки

В нашем примере будет три сервера backend, на которых может быть установлено абсолютно любое программное обеспечение (Apache, Nginx), и один сервер frontend для балансировки, на котором мы устанавливаем Nginx.

Настройки backend-серверов являются стандартными и не нуждаются в дополнительной настройке, кроме установки порта. Нужно установить порт, отличный от порта frontend-сервера (обычно от 80) и статический IP.

Далее следует настроить frontend-сервер. Процесс установки опустим, поскольку он описан выше. Нам необходимо настроить виртуальный хостинг, который будет переопределять наши запросы на определенный backend-сервер.

Пример конфигурационного файла для хостинга представлен на рис. 9.25.

В нашем случае виртуальный хостинг называется expamle.org, и чтобы при перезапуске сервера не было ошибок, следует создать директорию /var/www/root с правами доступа для пользователя www-data (стандартный пользователь для работы с сервером).

Чтобы подключиться к серверу и не поднимать DNS-сервер, добавим в файл hosts соответствующую запись типа: ip example.org, где ip – IP-адрес сервера.

```
upstream backend {
    server 192.168.10.11:8080;
    server 192.168.10.12:8080;
    server 192.168.10.13:8080;
}

server {
    listen    80;
    server_name  example.org;
    location ~* \.()$ {
    root   /var/www/example.org;  }
    location / {
    client_max_body_size    10m;
    client_body_buffer_size 128k;
    proxy_send_timeout   90;
    proxy_read_timeout   90;
    proxy_buffer_size    4k;
    proxy_buffers     16 32k;
    proxy_busy_buffers_size 64k;
    proxy_temp_file_write_size 64k;
    proxy_connect_timeout 30s;
    proxy_pass   http://backend;
    proxy_set_header   Host   $host;
    proxy_set_header   X-Real-IP  $remote_addr;
    proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
       }
location ~* /.(jpg|jpeg|gif|png|css|mp3|avi|mpg|txt|js|jar|rar|zip|tar|wav|wmv)
root    /var/www/example.org;}
```

Рис. 9.25. Конфигурационный файл для frontend-сервера

Главной настройкой для балансировки является модуль upstream. Рассмотрим различные параметры для управления методами балансировки:

– round-robin – используется по умолчанию (нет директивы, которая бы его включала). Запросы распределяются между серверами группы равномерно. Учитывается вес сервера (по умолчанию вес равен 1). Пример настройки представлен на рис. 9.26;

```
upstream backend {
 server backend1.example.com;
 server backend2.example.com;
}
```

Рис. 9.26. Настройка round-robin

– least_conn – запросы уходят на сервер с минимальным количеством активных соединений. Учитывается вес сервера. Пример настройки приведен на рис. 9.27;

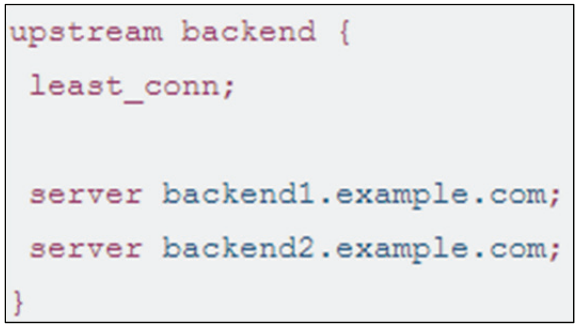


Рис. 9.27. Настройка least_conn

– ip_hash – сервер, к которому отправятся запросы, определяется на основании IP-адреса клиента. Для вычисления хеш-функции используются первые три октета IPv4-адреса либо весь IPv6-адрес. Этот метод гарантирует то, что запросы конкретного клиента попадут на конкретный сервер. Пример настройки представлен на рис. 9.28.

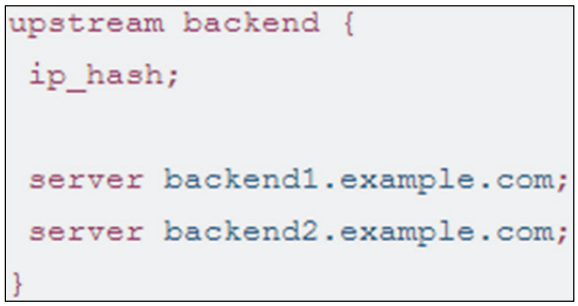


Рис. 9.28. Настройка ip_hash

Если же один из серверов группы нужно временно вывести из эксплуатации, то его можно пометить параметром down (рис. 9.29). В таком случае запрос клиента автоматически пойдет на следующий сервер группы;

Рис. 9.29. Настройка с выключенным сервером

– hash – сервер, на который пойдет запрос, определяется ключом клиента, который может быть текстом, переменной или их комбинацией.

Например, ключ может быть IP-адресом и портом клиента или URI. Пример настройки приведен на рис. 9.30;

```
upstream backend {
 hash $request_uri consistent;

 server backend1.example.com;
 server backend2.example.com;
}
```

Рис. 9.30. Настройка параметра hash

– consistent – включает метод консистентного кеширования ketama. Это значит, что при добавлении или удалении сервера из группы на другие серверы будет перераспределено минимальное число ключей;

– weight – по умолчанию вес каждого сервера в группе одинаков и равен 1. Также по умолчанию Nginx использует метод балансировки round-robin. Изменим вес одного из серверов группы (определяется параметром weight). Пример настройки представлен на рис. 9.31.

```
upstream backend {
 server backend1.example.com weight=5;
 server backend2.example.com;
 server 192.0.0.1 backup;
}
```

Рис. 9.31. Настройка весов серверов

В данном случае backend1 получил вес 5. Другие два сервера имеют вес, по умолчанию равный 1. При такой конфигурации пять запросов уходят на backend1 и один запрос на backend2 и т. д.

9.2.7. Использование контейнера docker

Сейчас мы попробуем запустить сервер Nginx в контейнере docker, чтобы затем его можно было легко разворачивать на других машинах, независимо от операционной системы.

Для начала необходимо установить docker командой:

```
$ sudo apt install docker-ce
```

После установки можно проверить его версию с помощью команды:

```
$ sudo docker -v
```

Итак, для того, чтобы запустить сервер в контейнере, необходимо создать dockerfile, с помощью которого мы будет собирать наш сервер. Пример dockerfile представлен на рис. 9.32.

NginxDockerfile

```
FROM ubuntu
MAINTAINER Dmitry Kravchuk <dimysa30@outlook.com>

RUN apt-get update
RUN apt-get install -y nginx php php-fpm php-intl php-mysql php-xdebug
RUN echo "\ndaemon off;" >> /etc/nginx/nginx.conf
RUN sed -i -e "s/;\?daemonize\s*=\s*yes/daemonize = no/g" /etc/php/7.0/fpm/php-fpm.conf

# Nginx config
RUN rm /etc/nginx/sites-enabled/default
RUN ln -s /etc/nginx/sites-available/default /etc/nginx/sites-enabled/default

# PHP config
RUN sed -i -e "s/;\?date.timezone\s*=\s*.*/date.timezone = Europe\/Kiev/g" /etc/php/7.0/fpm/php.ini

# Define default command.
CMD service php7.0-fpm start && nginx

# Expose ports.
EXPOSE 80
```

Рис. 9.32. Пример dockerfile

Можно заметить, что это просто набор инструкций, который будет впоследствии выполняться. Рассмотрим следующие команды:

1) FROM – это первая запись в вашем dockerfile. Эта строчка сообщает docker, на базе какого дистрибутива будет создан новый образ;

2) MAINTAINER – если вы публикуете свои образы, то стоит прописать данную строку. Она даст возможность видеть, кто автор данного образа или контейнера;

3) ADD – позволяет извлекать файлы и копировать их куда-нибудь в вашем контейнере;

4) RUN – выполняет команду, которую вы напишете в консоли;

5) EXPOSE – пробрасывает порты наружу.

Примечание. Является внутренним портом. При этом данный порт будет отличаться тем, что будет использоваться снаружи. Порт назначается динамическим docker.

6) CMD – запускает задачи, которые вы укажите.

Затем нам необходимо собрать наш контейнер. Это можно сделать с помощью команды, приведенной на рис. 9.33.

```
dmitry@superman:~/docker-nginx$ sudo docker build -f UbuntuDockerfile -t course-
nginx /home/dmitry/docker-nginx/
Sending build context to Docker daemon 17.41 kB
Step 1/11 : FROM ubuntu
 ---> 20c44cd7596f
Step 2/11 : MAINTAINER Dmitry Kravchuk <dimysa30@outlook.com>
 ---> Using cache
 ---> 349413a1a20d
Step 3/11 : RUN apt-get update
 ---> Using cache
 ---> 1b944a75c858
Step 4/11 : RUN apt-get install -y nginx php php-fpm php-intl php-mysql php-xdeb
ug
 ---> Using cache
 ---> 6f287b3081db
Step 5/11 : RUN echo "\ndaemon off;" >> /etc/nginx/nginx.conf
 ---> Using cache
 ---> 4b8384b8158a
Step 6/11 : RUN sed -i -e "s/;\?daemonize\s*=\s*yes/daemonize = no/g" /etc/php/7
.0/fpm/php-fpm.conf
 ---> Using cache
 ---> 72a8a9e3384b
Step 7/11 : RUN rm /etc/nginx/sites-enabled/default
 ---> Using cache
 ---> 8a1bbc77bce6
```

Рис. 9.33. Сборка контейнера

Параметр -f указывает на название dockerfile. Если он стандартный (dockerfile), то этот параметр не обязателен. Параметр -t дает ему тег – его название (рис. 9.33).

После того, как сборка пройдет успешно, мы можем запустить контейнер. Запуск выполняется командой, представленной на рис. 9.34.

```
dmitry@superman:~/docker-nginx$ sudo docker run --name course -p 8080:80 -v /hom
e/dmitry/docker-nginx/html/:/var/www/html course-nginx
```

Рис. 9.34. Запуск контейнера

Здесь все просто, за исключением параметра -v. Он предназначен для связывания папки вашего компьютера с папками контейнера. В нашем случае связывается директория /home/Dmitry/docker-nginx/html с /var/www/html, т. е. заменяются файлы хостинга. Таким образом, можно поменять и конфигурационные файлы сервера. Параметр -p указывает, на каком порту запускаем контейнер.

После запуска мы можем зайти на сайт, указав IP-адрес docker. Чтобы его посмотреть, используем команду (рис. 9.35).

```
dmitry@superman:~$ ifconfig
docker0   Link encap:Ethernet  HWaddr 02:42:79:44:bf:3b
          inet addr:172.17.0.1  Bcast:0.0.0.0  Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Рис. 9.35. Проверка IP-адреса

Работающий сайт представлен на рис. 9.36.

Рис. 9.36. Работа хостинга

На этом работа с контейнером docker завершена. Можно отметить, что в docker репозиториях лежит большое количество готовых сборок, поэтому проще брать уже готовые и разворачивать в контейнере, чем настраивать самому.

Лабораторная работа № 16–17

***Цель*:** получение практических навыков по развертыванию веб-сервера Nginx.

***Задание*:** следует произвести установку веб-сервера Nginx. Необходимо использовать Nginx для проксирования запросов к Apache 2.x, настроить проксирование статических и динамических запросов. Следует использовать PHP-FPM с Nginx без Apache 2.x.

Необходимо выполнить настройку модуля для ngx_pagespeed, который представляет собой набор фильтров и позволяет значительно повысить производительность сайта.

Следует выявить, от какого пользователя работает Nginx, а также где сохраняются логи Nginx.

Необходимо выбрать и установить оптимальное значение worker_connections.

Следует сконфигурировать Nginx со следующими настройками (прокомментировать каждую строчку конфигурационного файла):

– выявить, от какого пользователя работает Nginx;

– настроить Nginx на прослушивание 80 порта по default-серверу;

– указать имя сервера, чтобы оно соответствовало вашему домену или IP;

– параметр server_name_in_redirect установить в режим off;

– задать корректные настройки proxy_set_header так, чтобы сервер открывал запросы с основного хоста по 80 порту;

– задать типы/список индексных файлов;

– установить ограничения по максимальному размеру принимаемых файлов;

– установить оптимальный таймаут соединения.

Необходимо настроить Nginx для отдачи всего статичного контента.

В результате работы следует получить сервер, который принимает все запросы через Nginx, кеширует и отдает статику из отдельной папки.

Раздел 10
ПРИМЕНЕНИЕ ZABBIX ДЛЯ КОМПЛЕКСНОГО МОНИТОРИНГА ЛОКАЛЬНОЙ СЕТИ

Мониторинг – методика и система наблюдений за состоянием определенного объекта или процесса, дающая возможность наблюдать их в развитии и работе, оценивать, оперативно выявлять результаты воздействия различных внешних и внутренних факторов. Результаты мониторинга дают возможность вносить корректировки по управлению объектом или процессом.

10.1. Система мониторинга Zabbix

Zabbix – это программное обеспечение мониторинга многочисленных параметров сети, а также состояния и работоспособности серверов. Zabbix использует гибкий механизм уведомлений, что позволяет пользователям настраивать оповещения по почте практически для любого события. Это дает возможность быстро среагировать на проблемы с сервером.

Zabbix поддерживает опрос данных (пуллер) и получение данных (траппер). Все отчеты и статистика Zabbix так же, как и параметры настроек, доступны через веб-интерфейс. Веб-интерфейс обеспечивает такую возможность, чтобы состояние вашей сети и жизнедеятельность ваших серверов можно было оценить из любого места. Это делает Zabbix идеальным инструментом для планирования и масштабирования.

Определения в Zabbix. ***Узел сети*** – сетевое устройство, мониторинг которого вы хотите производить, имеющее IP/DNS.

Группа узлов сети – логическая группировка узлов сети; она может содержать узлы сети и шаблоны. Узлы сети и шаблоны в группе узлов сети никаким образом не связаны с друг другом. Группы узлов сети используются при назначении прав доступа к узлам сети различным группам пользователей.

Элемент данных – конкретный фрагмент данных, который вы хотите получать от узла сети, метрика.

Триггер – логическое выражение, которое определяет порог проблемы и используется для «оценки» данных, полученных элементами данных.

Если полученные данные превышают порог, триггер переходит из состояния «Ок» в состояние «Проблема». Если полученные данные ниже порога, триггер остается/возвращается в состояние «Ок».

Событие – одиночное возникновение того, что заслуживает внимания, например изменение состояния триггера или обнаружение/авторегистрация агента.

Действие – предопределенные средства реагирования на события. Действия состоят из операций (например, отправка оповещений) и условий (когда осуществляется операция).

Эскалация – пользовательский сценарий для выполнения операций в рамках действия; последовательность отправки оповещений/выполнений удаленных команд.

Способ оповещения – способ доставки оповещений; канал доставки.

Оповещение – сообщение о некотором событии, отправленное пользователю через выбранный канал доставки.

Удаленная команда – предопределенная команда, которая будет автоматически выполнена на наблюдаемом узле сети при некоторых условиях.

Шаблон – набор сущностей (элементы данных, триггеры, графики, комплексные экраны, группы элементов данных, правила низкоуровневого обнаружения, веб-сценарии), которые готовы к присоединению к одному или нескольким узлам сети.

Задача шаблонов – повысить скорость развертывания задач мониторинга узлов сети; кроме того, упростить применение массовых изменений к задачам наблюдения. Шаблоны соединяются напрямую с отдельными узлами сети.

Группа элементов данных – группировка элементов данных в некую логическую группу.

Веб-сценарий – один или несколько HTTP-запросов для проверки доступности веб-сайта.

Веб-интерфейс – веб-интерфейс, поставляемый с Zabbix.

Zabbix API позволяет вам использовать протокол JSON RPC для создания, обновления и получения объектов Zabbix (например, узлов сети, элементов данных, графиков и пр.) или для выполнения любых других пользовательских задач.

Zabbix-сервер – главный процесс программного обеспечения Zabbix, который выполняет мониторинг, взаимодействует с прокси и агентами Zabbix, вычисляет триггеры, отправляет оповещения; центральное хранилище данных.

Zabbix-агент – процесс, разворачиваемый на наблюдаемых целях для активного мониторинга локальных ресурсов и приложений.

Zabbix-прокси – процесс, который может собирать данные от имени Zabbix-сервера, перенимая часть нагрузки сервера.

10.1.1. Zabbix-сервер

Zabbix-сервер является главным компонентом, которому агенты сообщают информацию и статистику о доступности и целостности.

Сервер – главное хранилище, в котором хранятся все конфигурационные, статистические и оперативные данные, также он рассылает уведомления администраторам в случае возникновения проблем с любой из наблюдаемых систем.

Сервер выполняет опрос и захват данных, он вычисляет триггеры, отправляет оповещения пользователям. Это главный компонент, которому Zabbix-агенты и прокси отправляют данные доступности и целостности системы. Сервер может самостоятельно удаленно проверять сетевые устройства (так же, как и веб-серверы, и почтовые серверы), используя простые проверки сервиса.

Функционал базового Zabbix-сервера разделен на три отдельных компонента: Zabbix-сервер, веб-интерфейс и хранилище в базе данных.

Все данные о конфигурации Zabbix хранятся в базе данных, с которой взаимодействует и сервер, и веб-интерфейс. Например, когда вы создаете новый элемент данных, используя веб-интерфейс (или API), запись об этом добавляется в таблицу элементов данных в базе данных. Затем раз в минуту Zabbix-сервер опрашивает таблицу элементов данных для получения списка активных элементов данных и сохраняет этот список в кеш Zabbix-сервера. Именно поэтому любые изменения в веб-интерфейсе Zabbix будут отображены в разделе последних данных с задержкой до 2 мин.

10.1.2. Zabbix-агент

Zabbix-агенты разворачиваются на наблюдаемых целях для активного мониторинга за локальными ресурсами и приложениями (статистика жестких дисков, памяти, процессоров и т. д.).

Агент собирает локальную оперативную информацию и отправляет данные Zabbix-серверу для дальнейшей обработки. В случае проблем (таких как рабочий жесткий диск заполнен или упал процесс сервиса) Zabbix-сервер может быстро уведомить администраторов конкретного сервера, который сообщил об ошибке.

Zabbix-агенты чрезвычайно эффективны, потому что используют нативные системные вызовы для сбора информации статистики.

Zabbix-агенты могут выполнять пассивные и активные проверки. В случае *пассивной проверки* агент отвечает на запрос данных. Zabbix-

сервер (или прокси) запрашивает данные, например загрузку центрального процессорного устройства, и Zabbix-агент возвращает результат. *Активные проверки* требуют более сложной обработки. Агент сначала получает список элементов данных для независимой обработки от Zabbix-сервера. Далее он будет периодически отправлять новые значения серверу.

10.2. Развертывание системы мониторинга Zabbix

10.2.1. Установка Zabbix 3.0

Прежде чем начать саму установку на Linux CentOS7, проводится предварительная настройка сервера.

1. Обновляем систему с помощью команды yum -y update (как показано на рис. 10.1).

```
[root@localhost ~]# yum -y update
```

Рис. 10.1. Команда обновления

2. Отключаем SElinux, чтобы не было проблем с настройкой в дальнейшем. Открываем редактор: *mcedit /etc/sysconfig/selinux*. Вместо *mcedit* можно использовать встроенный редактор *vim* (рис. 10.2). Редактируем строку: *SELINUX = DISABLED*.

```
[root@localhost ~]# vim /etc/sysconfig/selinux
```

Рис. 10.2. Команда открытия редактора

3. Устанавливаем *mariadb*, который является ответвлением *mysql*. Они полностью совместимы, возможен в любой момент переход с одной СУБД на другую и обратно. Вся информация о конфигурации, а также данные, собранные Zabbix, хранятся в базе данных. Используем команду yum install -y mariadb mariadb-server.

4. Запускаем *mariadb* и добавляем в автозагрузку: systemctl start mariadb и systemctl enable mariadb.service (рис. 10.3).

5. Отрабатываем скрипт первоначальной настройки *mysql*: /usr/bin/mysql_secure_installation (рис. 10.4).

Теперь переходим к самой установке Zabbix 3.0.

1. Подключаем репозиторий 3.0: rpm -ivh http://repo.zabbix.com/zabbix/3.0/rhel/7/x86_64/zabbix-release-3.0-1.el7.noarch.rpm (рис. 10.5).

2. Устанавливаем Zabbix-сервер с веб-интерфейсом и базой данных MySQL: yum install -y zabbix-server-mysql zabbix-web-mysql (рис. 10.6).

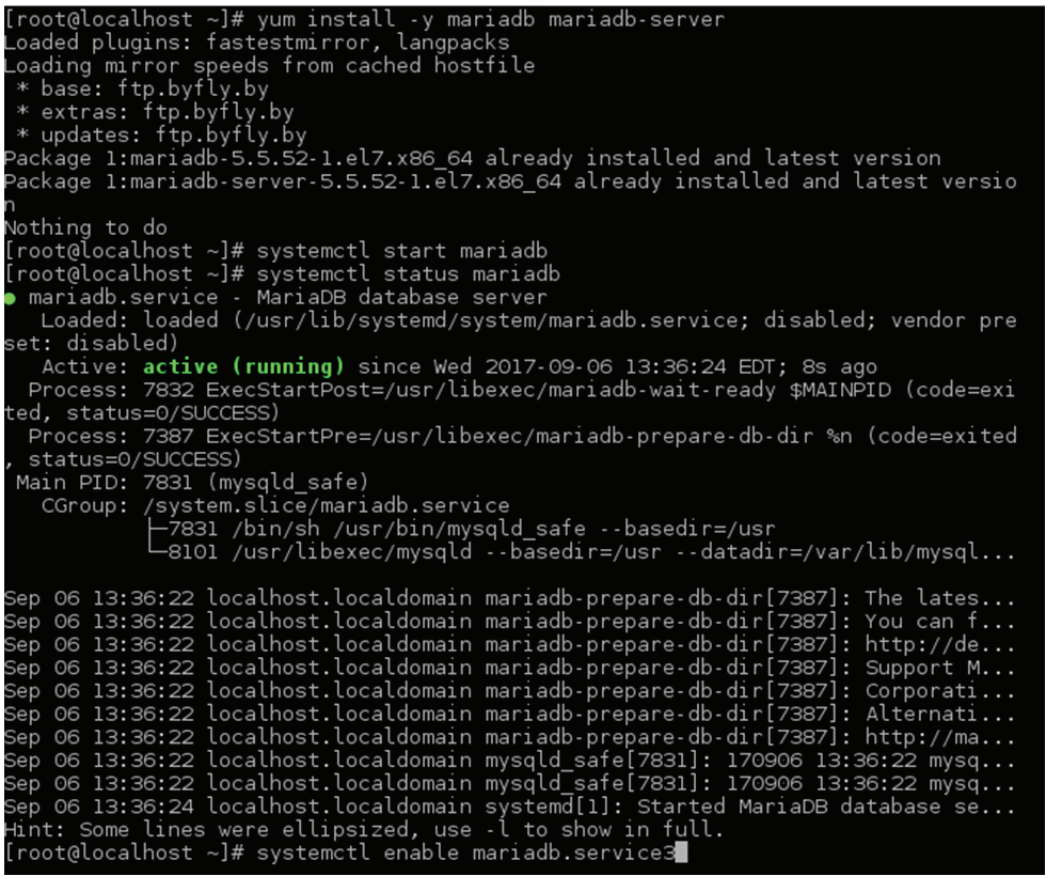


Рис. 10.3. Команды запуска и проверки статуса *mariadb*

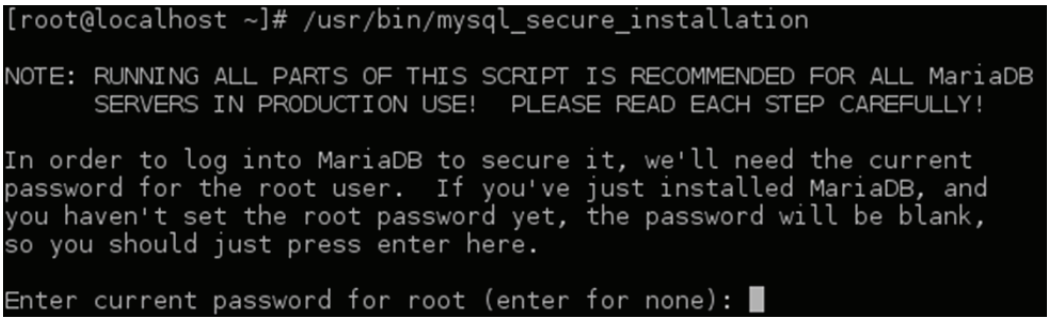


Рис. 10.4. Запуск первоначальной настройки

Рис. 10.5. Подключение репозитория 3.0

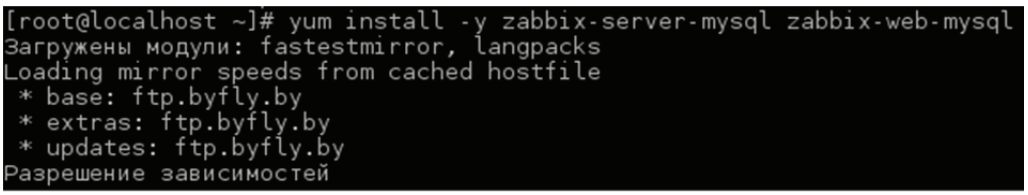


Рис. 10.6. Установка Zabbix-сервера

3. Создаем пользователя и базу данных для мониторинга: mysql -uroot -p12042016t (рис. 10.7).

Теперь вводим строки: create database zabbix character set utf8 collate utf8_bin; grant all privileges on zabbix.* to zabbix@localhost identified by '12042016', где topsecret – пароль пользователя root mysql, это не системный root; 12042016 – пароль пользователя Zabbix, у которого полный доступ к базе mysql zabbix.

Для выхода из базы набираем *exit*.

4. Теперь импортируем схему БД и начальные данные (рис. 10.8) с помощью команды zcat /usr/share/doc/zabbix-server-mysql-3.0.1/create.sql.gz | mysql -uroot -ptopsecret zabbix.

5. На данном шаге редактируем файл конфигурации сервера Zabbix. Прописываем данные для подключения к БД: mcedit/etc/zabbix/zabbix_server.conf (как было уже сказано, можно использовать *vim*) (рис. 10.9).

Изменяем строки:

DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=12042016

6. Запускаем Zabbix-сервер и добавляем в автозагрузку (рис. 10.10): systemctl start zabbix-server и systemctl enable zabbix-server.

7. Проверяем лог файла на наличие ошибок (рис. 10.11): cat /var/log/zabbix/zabbix_server.log.

8. Для продолжения настройки сервера нам нужно зайти в веб-интерфейс. Перед этим отредактируем файл конфигурации веб-сервера. Откроем конфиг и раскомментируем одну строку, изменив ее под свой часовой пояс (рис. 10.12): mcedit /etc/httpd/conf.d/zabbix.conf (и тут используем *vim*). Изменяем строку (рис. 10.13): php_value date.timezone Europe/Moscow.

9. Сохраняем файл. Теперь запускаем *httpd* и добавляем его в автозагрузку (рис. 10.14): systemctl start httpd и systemctl enable httpd.

10. Можно заходить на веб-интерфейс по адресу http://192.168.0.110/zabbix, где 192.168.0.110 – IP-адрес сервера, где устанавливаем и настраиваем мониторинг.

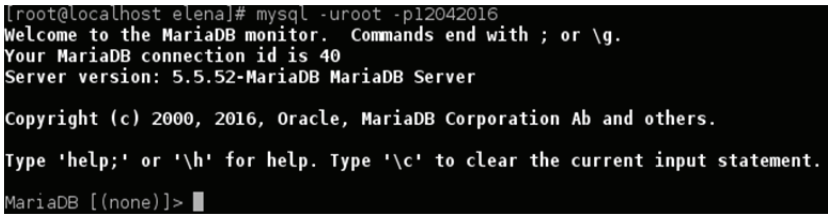


Рис. 10.7. Вход в *mariadb*

```
[root@localhost ~]# zcat /usr/share/doc/zabbix-server-mysql-3.0.1/create.sql.gz
| mysql -uroot -p zabbix
```

Рис. 10.8. Команда импорта

```
[root@localhost ~]# vim /etc/zabbix/zabbix_server.conf
```

Рис. 10.9. Открытие редактором
конфигурационного файла

```
[root@localhost ~]# systemctl start zabbix-server
[root@localhost ~]# systemctl enable zabbix-server
```

Рис. 10.10. Команды запуска Zabbix-сервера

```
[root@localhost ~]# systemctl start zabbix-server
[root@localhost ~]# systemctl enable zabbix-server
[root@localhost ~]# cat /var/log/zabbix/zabbix_server.log
 22343:20170906:141129.493 Starting Zabbix Server. Zabbix 3.0.10 (revision 70208
).
 22343:20170906:141129.493 ****** Enabled features ******
 22343:20170906:141129.493 SNMP monitoring:           YES
 22343:20170906:141129.493 IPMI monitoring:           YES
 22343:20170906:141129.493 Web monitoring:            YES
 22343:20170906:141129.493 VMware monitoring:         YES
 22343:20170906:141129.493 SMTP authentication:       YES
 22343:20170906:141129.493 Jabber notifications:      YES
 22343:20170906:141129.493 Ez Texting notifications:  YES
 22343:20170906:141129.493 ODBC:                      YES
 22343:20170906:141129.493 SSH2 support:              YES
 22343:20170906:141129.493 IPv6 support:              YES
 22343:20170906:141129.493 TLS support:               YES
 22343:20170906:141129.493 ******************************
 22343:20170906:141129.493 using configuration file: /etc/zabbix/zabbix_server.c
onf
 22343:20170906:141129.564 current database version (mandatory/optional): 030000
00/03000000
 22343:20170906:141129.564 required mandatory version: 03000000
 22343:20170906:141129.570 server #0 started [main process]
 22359:20170906:141129.585 server #12 started [trapper #4]
 22360:20170906:141129.588 server #13 started [trapper #5]
 22361:20170906:141129.590 server #14 started [icmp pinger #1]
 22363:20170906:141129.590 server #16 started [housekeeper #1]
 22366:20170906:141129.591 server #19 started [discoverer #1]
```

Рис. 10.11. Проверка лог файла

```
[root@localhost ~]# vim /etc/httpd/conf.d/zabbix.conf
```

Рис. 10.12. Запуск через *vim*
в конфигурационный файл

```
<Directory "/usr/share/zabbix">
    Options FollowSymLinks
    AllowOverride None
    Require all granted

    <IfModule mod_php5.c>
        php_value max_execution_time 300
        php_value memory_limit 128M
        php_value post_max_size 16M
        php_value upload_max_filesize 2M
        php_value max_input_time 300
        php_value always_populate_raw_post_data -1
        php_value date.timezone Europe/Moscow
    </IfModule>
</Directory>
```

Рис. 10.13. Конфигурационный файл

```
[root@localhost ~]# systemctl start httpd
[root@localhost ~]# systemctl enable httpd
```

Рис. 10.14. Команды запуска *httpd*

10.2.2. Настройка Zabbix 3.0 через веб-интерфейс

1. Открываем в браузере веб-интерфейс Zabbix (через адрес http://192.168.0.110/zabbix). Нас встречает страница приветствия (рис. 10.15).

Рис. 10.15. Первая страница веб-интерфейса

2. Жмем *Next step* и попадаем на страницу проверок. Если все делать точно по инструкции, то все проверки будут пройдены (рис. 10.16).

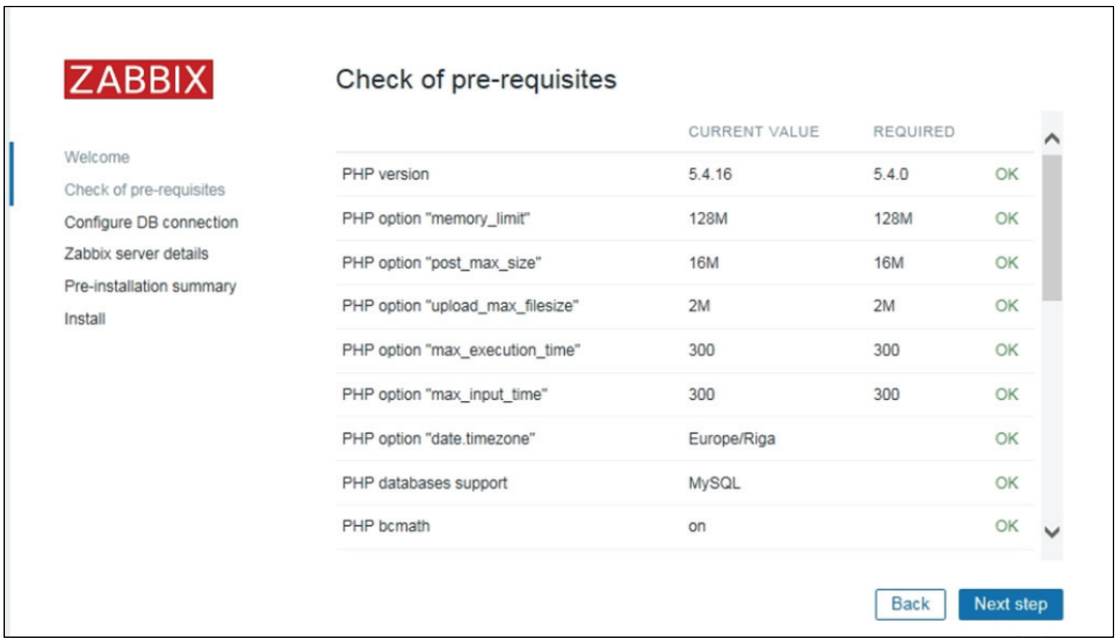


Рис. 10.16. Проверка на необходимые компоненты

3. Двигаемся дальше и указываем параметры для подключения к MySQL. Данные те же, что указывались ранее при создании БД и пользователя (рис. 10.17).

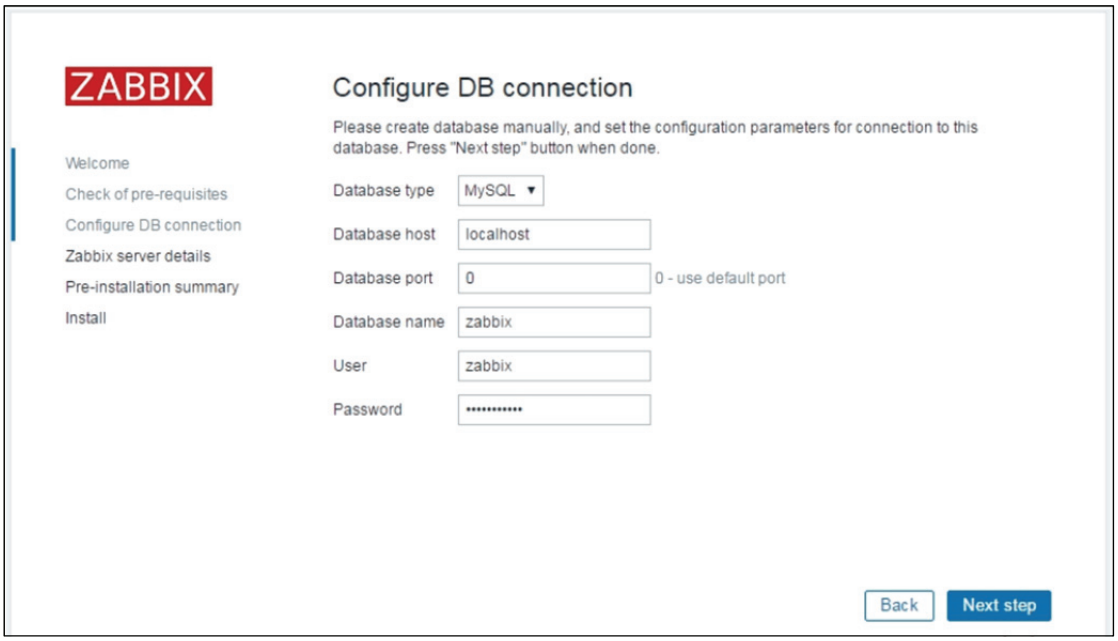


Рис. 10.17. Подключение к базе данных

4. На следующем этапе указываем адрес сервера и порт, на котором он будет работать. Оставляем значения по умолчанию (рис. 10.18).

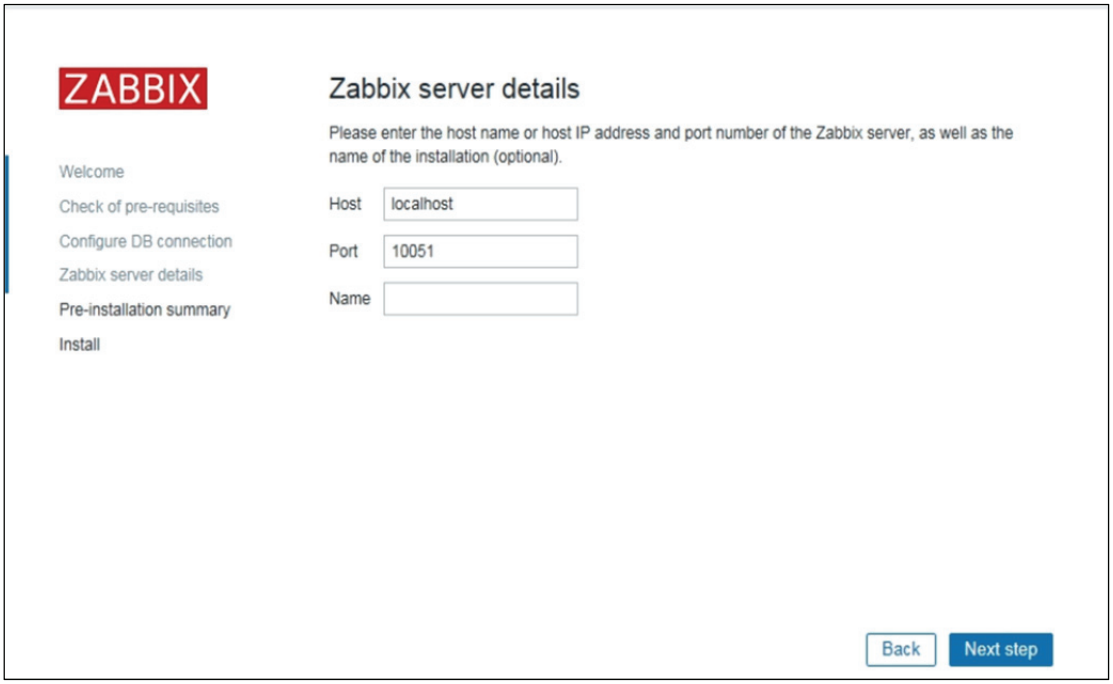


Рис. 10.18. Настройка расположения сервера

5. Еще раз проверяем все настройки. Если все в порядке, двигаемся дальше на заключительный этап установки (рис. 10.19).

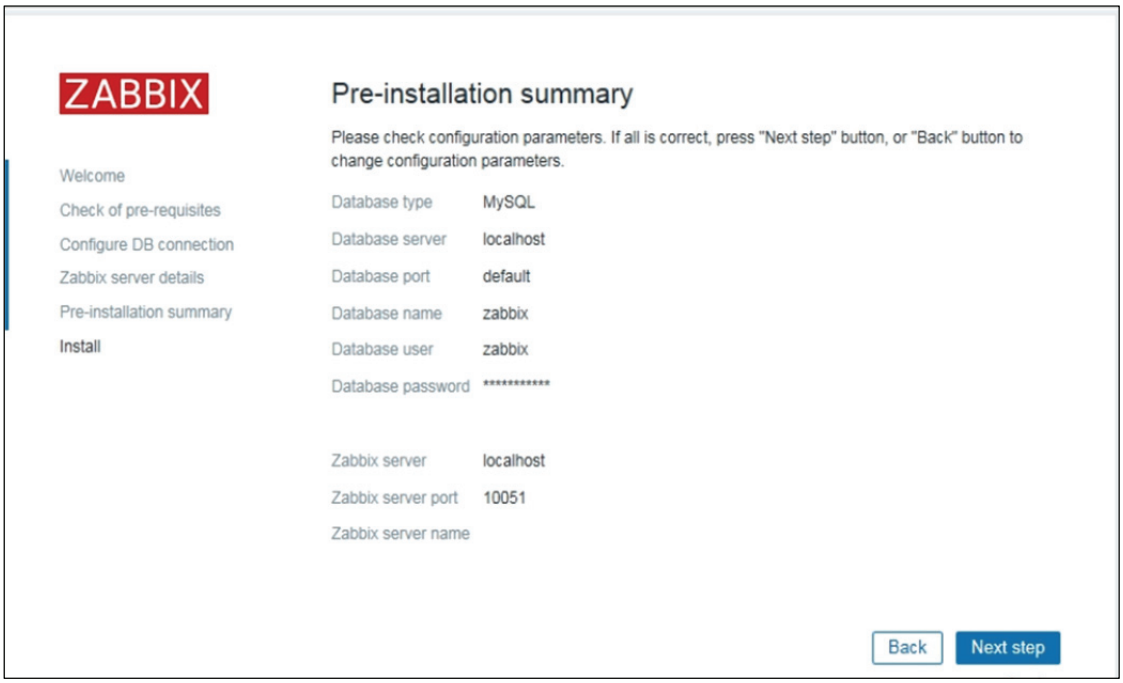


Рис. 10.19. Проверка настройки сервера

6. Если получили это сообщение, то установка сервера мониторинга Zabbix 3.0 закончена (рис. 10.20).

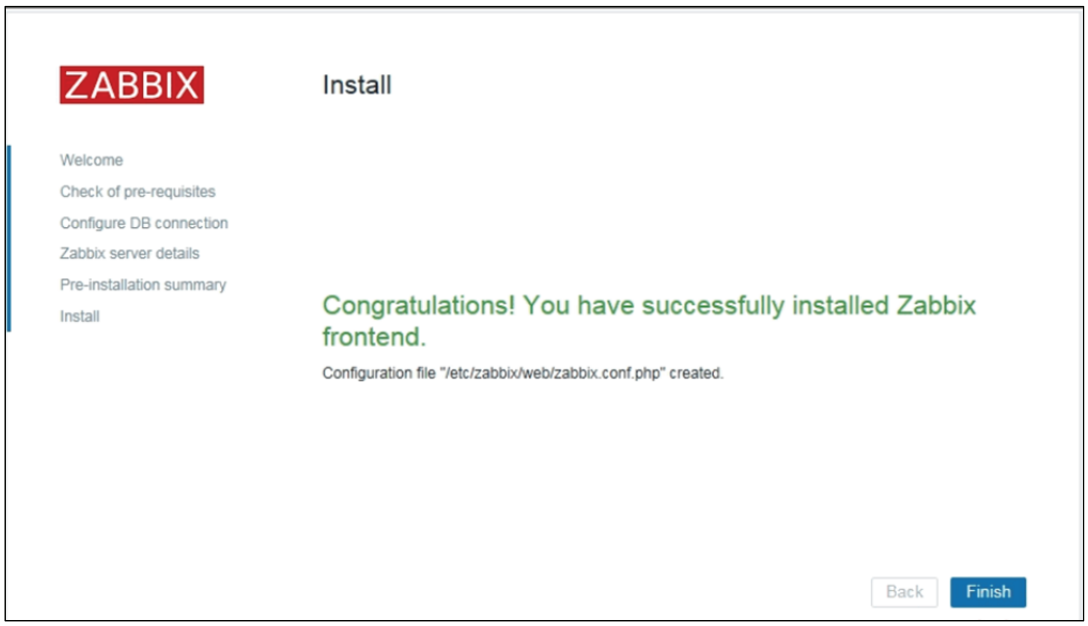


Рис. 10.20. Сообщение о завершении установки

Нажимаем *Finish* и попадаем на страницу логина в систему. Данные по умолчанию для захода на сайт Zabbix: пользователь – Admin; пароль – zabbix.

При заходе на сервер встречается голая панель управления (рис. 10.21), так как никаких параметров мы не наблюдаем и не имеем ни одного объекта сбора данных.

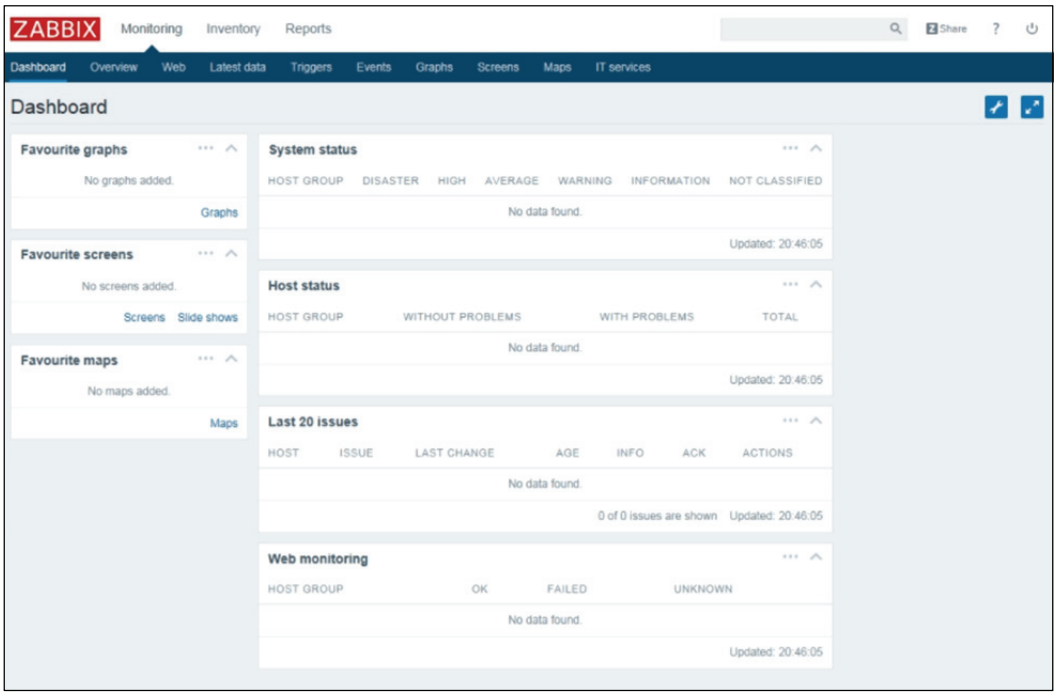


Рис. 10.21. Панель управления Zabbix-сервера

Для настройки агента на Zabbix выполним следующие операции.

1. Чтобы работал мониторинг удаленных машин, нужно добавить на сервер мониторинга *host* с таким же *hostname*, что мы указали в конфиге агента. Переходим по вкладкам *Configuration* → *Host* и нажимаем *Create host* (рис. 10.22).

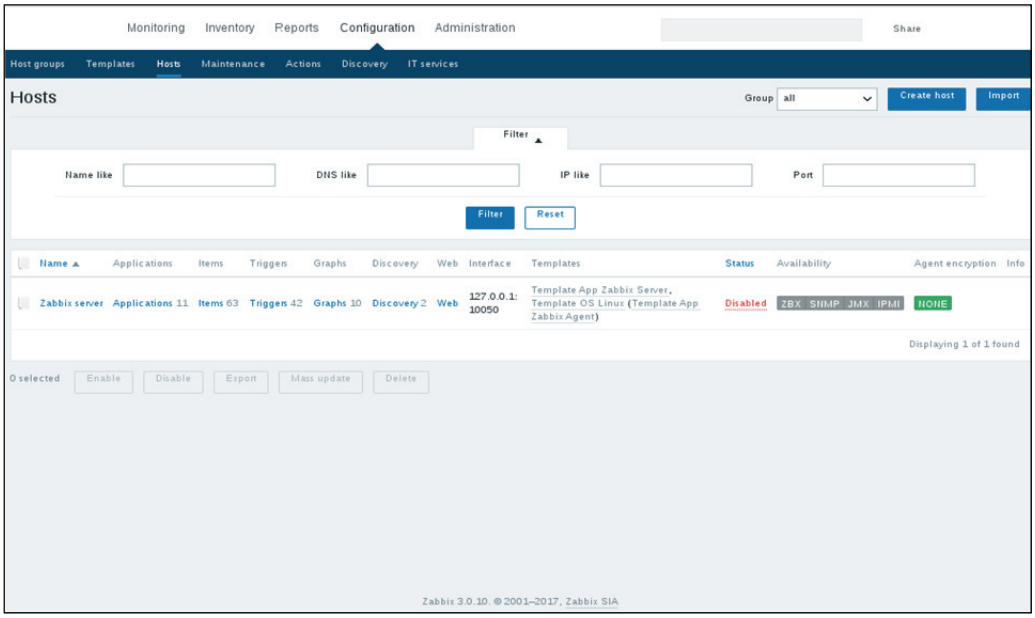


Рис. 10.22. Добавление хоста

2. Заполняем поля для ввода: указываем имя хоста, такое же, как на клиенте, добавляем в любую существующую группу и задаем IP-адрес компьютера, как показано на рис. 10.23.

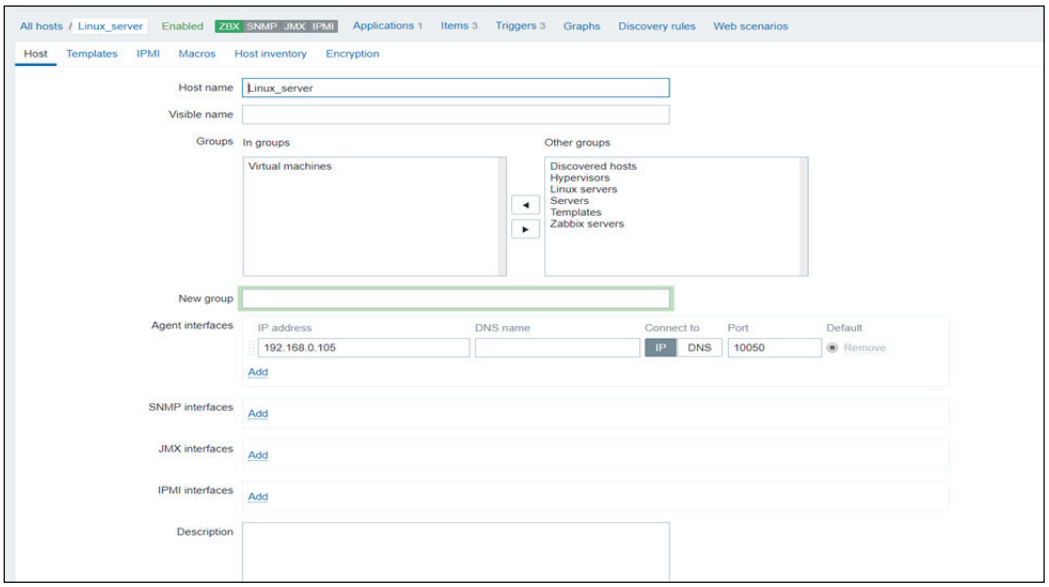


Рис. 10.23. Регистрация хоста

3. На вкладке *Templates* нажимаем *Select*, из списка выбираем, например, *Template OS Windows*, после чего нажимаем на ссылку *Add* (маленькая кнопочка под номером 1, приведенная на рис. 10.24). Шаблон добавлен.

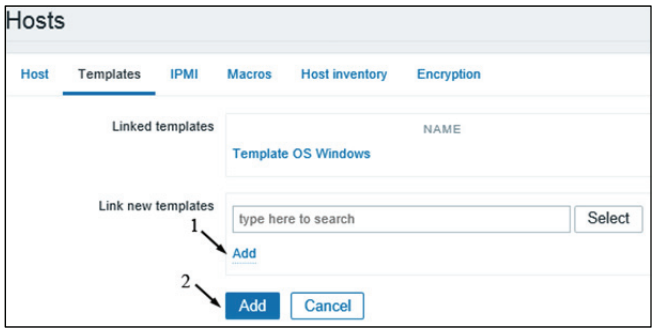


Рис. 10.24. Добавление шаблона

4. Вновь нажимаем кнопку *Add*, как показано на рис. 10.25.

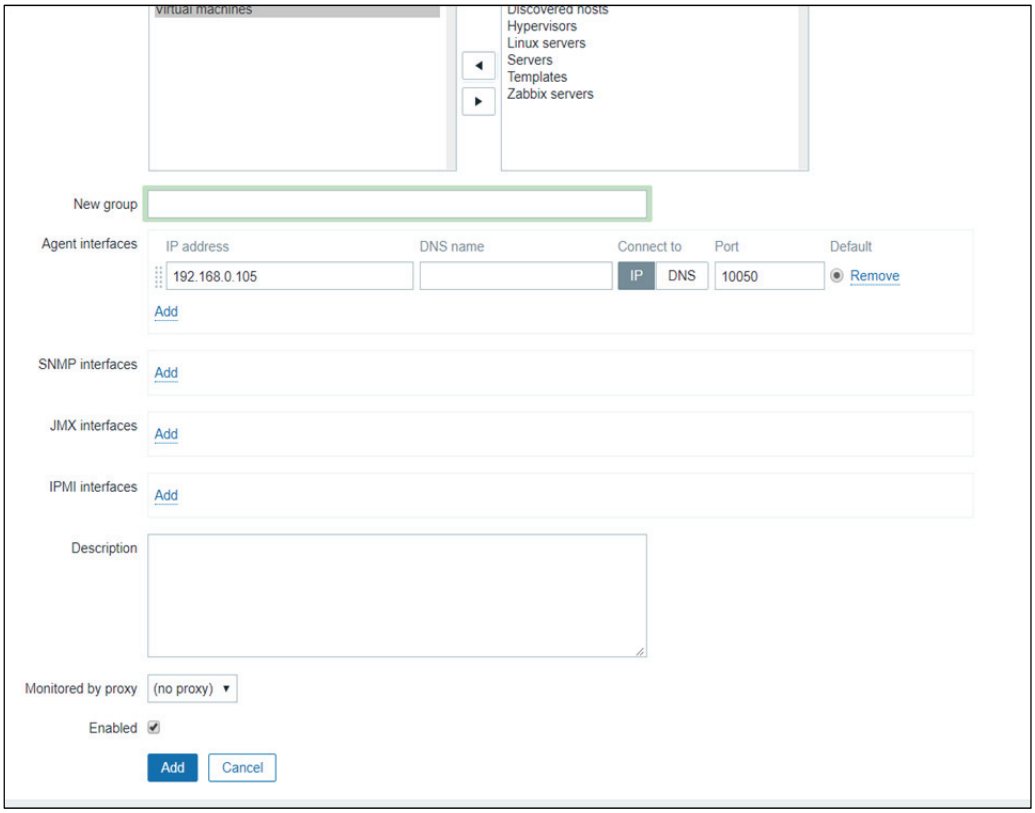


Рис. 10.25. Последние добавления

Таким образом, заканчивается первоначальная настройка Zabbix-сервера, проверка на подключение к нему через веб-интерфейс и добавление проверяемых агентов.

Лабораторная работа № 18–19

***Цель*:** рассмотрение основных принципов развертывания системы мониторинга Zabbix.

***Задание*:** необходимо развернуть систему мониторинга на виртуальной машине с операционной системой Linux. Следует произвести установку Zabbix-сервера, веб-интерфейса и базы данных на одном хосте. Необходимо установить Zabbix-агентов на трех хостах и выполнить их дальнейшую настройку. Кроме того, настроить внешние проверки и настройки триггеров/графиков/шаблонов и т. п.

10.3. Мониторинг сервисов и построение карт сети

10.3.1. Мониторинг Windows-серверов

IP данного сервера в работе =192.168.0.145. На сервере ничего установлено не будет, но будет проведена дальнейшая проверка на установленные программы.

С целью настройки агента выполним следующие действия.

1. Скачиваем самый последний агент для Windows (http://www.zabbix.com/download.php). Распаковываем архив. Создаем на диске C папку zabbix и копируем туда следующие файлы: zabbix_agentd.exe, zabbix_get.exe, zabbix_sender.exe, zabbix_agentd.win.conf.

Исполняемые файлы берем той разрядности, какая в системе. В исходном архиве есть как x32, так и x64.

2. Далее открываем командную строку с правами администратора (рис. 10.26) и выполняем следующую команду для установки Zabbix-агента на Windows (рис. 10.27): c:/zabbix/zabbix_agentd.exe --config c:/zabbix/zabbix_agentd.win.conf --install.

3. Открываем файл zabbix_agentd.win.conf любым текстовым редактором и изменяем следующие параметры (рис. 10.28 и 10.29):

```
Server=192.168.0.110
ServerActive=192.168.0.110
Hostname=win7-01
```

4. Теперь открываем оснастку со службами, ищем службу с именем Zabbix Agent (см. рис. 10.30 на с. 174) и запускаем ее.

Вот и настроили наши серверы и агенты на них.

Часто возникает небольшая проблема: Zabbix-сервер видит наш Zabbix-агент, но не может подключиться к нему (см. рис. 10.31 на с. 174). Тогда следует отключить firewall: systemctl stop firawalld, systemctl stop firawalld.

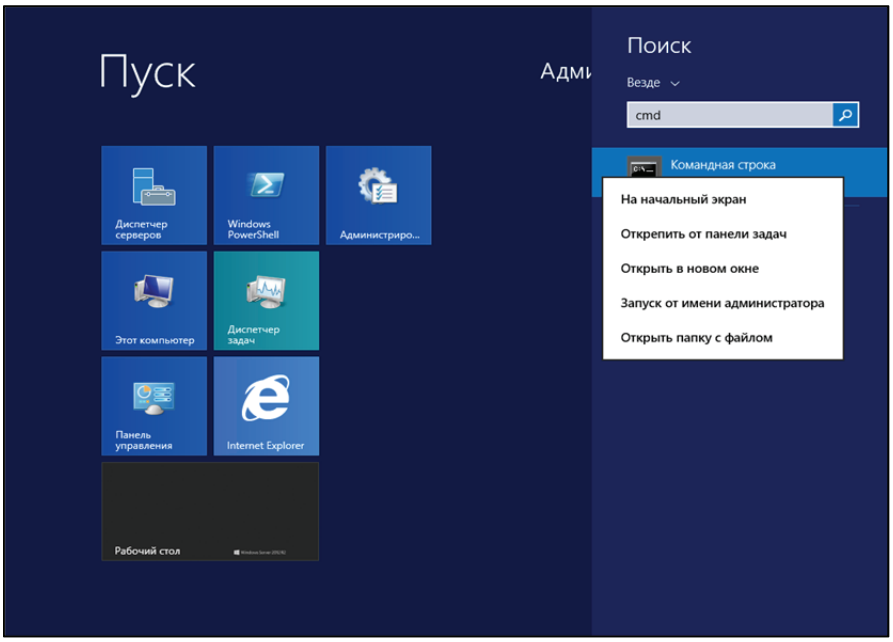


Рис. 10.26. Запуск консоли с правами администратора

```
C:\Windows\system32>c:/zabbix/zabbix_agentd.exe --config c:/zabbix/zabbix_agentd
.win.conf --install
zabbix_agentd.exe [2780]: service [Zabbix Agent] installed successfully
zabbix_agentd.exe [2780]: event source [Zabbix Agent] installed successfully
```

Рис. 10.27. Установка Zabbix-агента

```
65  ### Option: Server
66  #   List of comma delimited IP addresses (or hostnames) of Zabbix servers.
67  #   Incoming connections will be accepted only from the hosts listed here.
68  #   If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are
69  #
70  # Mandatory: no
71  # Default:
72  # Server=
73
74  Server=192.168.0.110
```

Рис. 10.28. Изменение конфигурационного файла (Server)

```
110 #   Example: ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
111 #
112 # Mandatory: no
113 # Default:
114 # ServerActive=
115
116 ServerActive=192.168.0.110
117
118 ### Option: Hostname
119 #   Unique, case sensitive hostname.
120 #   Required for active checks and must match hostname as configured on the server.
121 #   Value is acquired from HostnameItem if undefined.
122 #
123 # Mandatory: no
124 # Default:
125 # Hostname=
126
127 Hostname=Windows_server
```

Рис. 10.29. Изменение конфигурационного файла (ServerActive и Hostname)

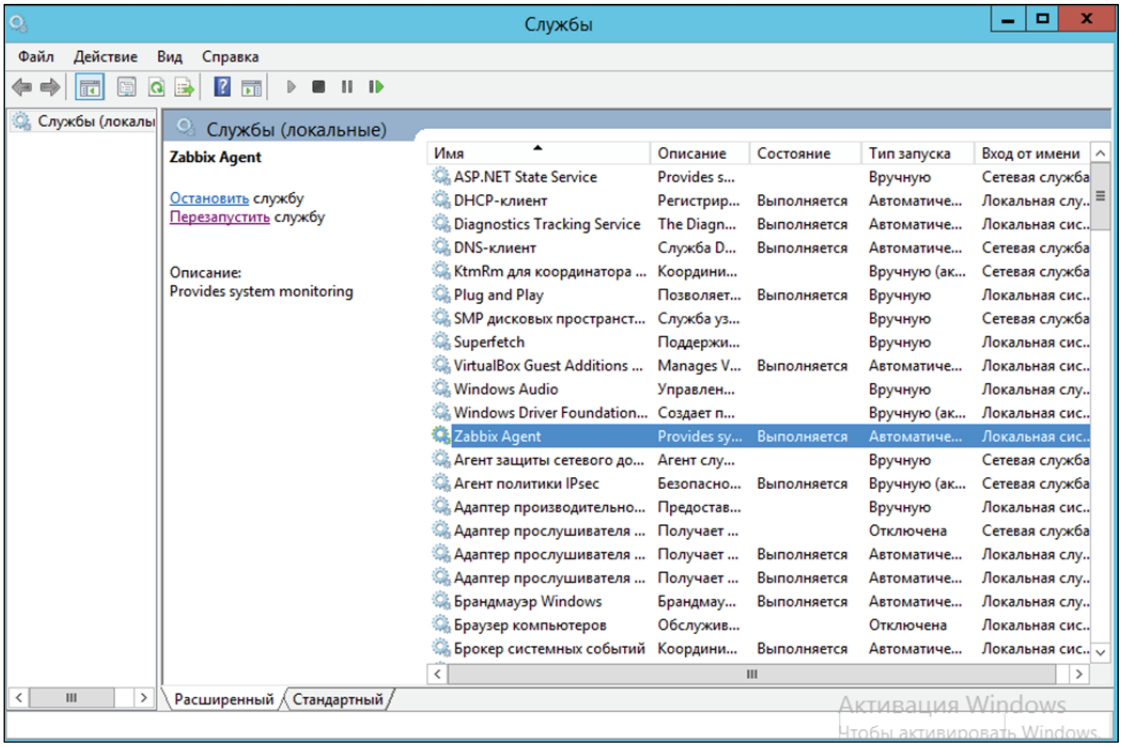


Рис. 10.30. Запуск службы Zabbix Agent

Рис. 10.31. Ошибка на сервере

По итогу всех настроек должно быть, как показано на рис. 10.32.

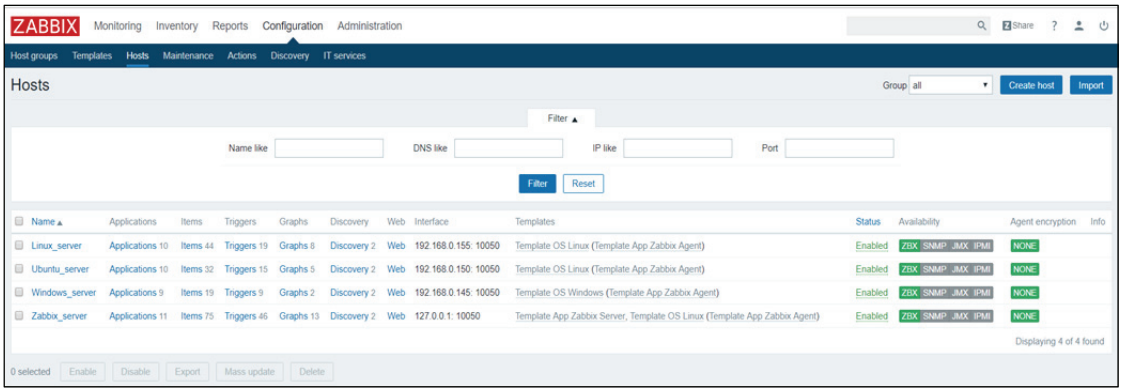


Рис. 10.32. Список подключенных серверов

После проверки корректности работы серверов и их агентов можно добавлять различные дополнительные настройки для мониторинга конкретных служб, установки и настройки дополнительных утилит и их просмотра на веб-интерфейсе Zabbix-сервера.

10.3.2. Мониторинг MySQL

Данный мониторинг проводится на Ubuntu-сервере. Для начала заходим в MySQL-сервер с помощью команды mysql -u root -p. После выполнения команды просят ввести пароль для входа в базу данных и после успешного входа можно вводить команды в базу данных (рис. 10.33).

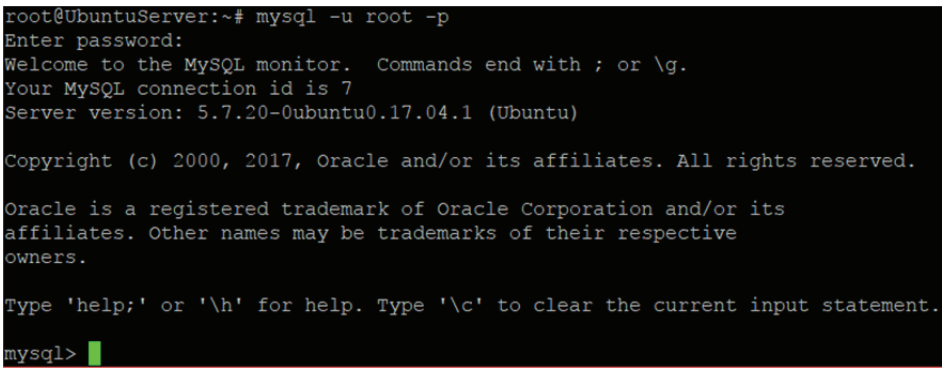


Рис. 10.33. Вход в базу данных

1. Создаем в MySQL пользователя, у которого будет доступ к информации о репликации, и сразу выдаем необходимые привилегии:

GRANT USAGE ON *.* TO 'zabbix'@'%' IDENTIFIED BY 'superpassword'; FLUSH PRIVILEGES.

Результат действий представлен на рис. 10.34.

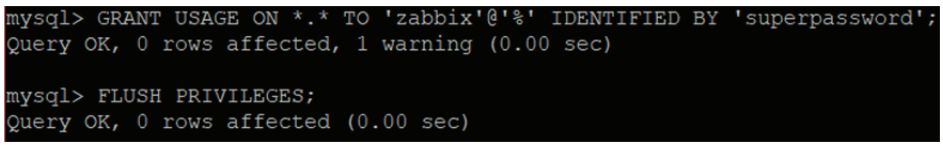


Рис. 10.34. Создание пользователя и выдача привилегий

2. Создаем файл с настройками подключения к MySQL-серверу: /etc/zabbix/.my.cnf (рис. 10.35).

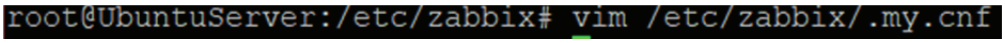


Рисунок 10.35. Создание файла с настройками подключения

3. Добавляем в него содержимое, как показано на рис. 10.36.

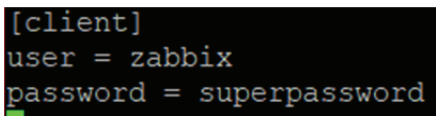


Рис. 10.36. Файл для конфигурации
к MySQL-серверу

4. Следует также изменить файл userparameter_mysql.conf. Поэтому после команды vim /etc/zabbix/zabbix_agentd.d/userparameter_mysql.conf добавляем необходимые строки (рис. 10.37).

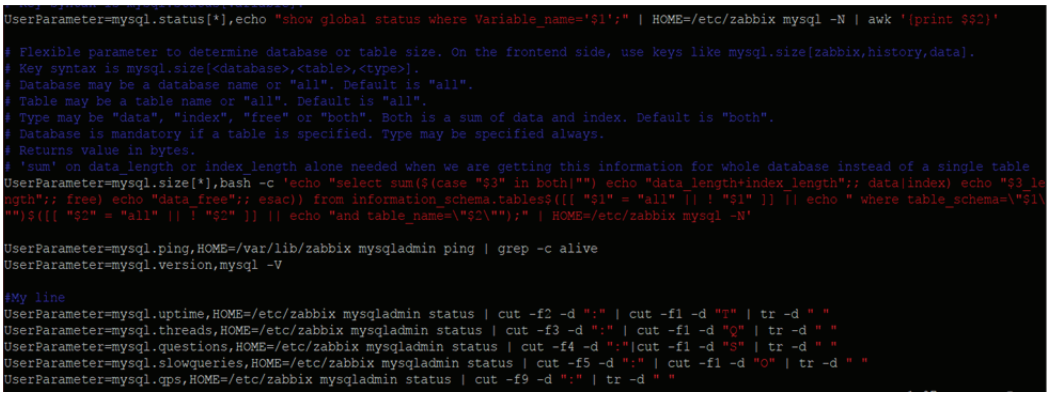


Рис. 10.37. Данные, которые будут передаваться агентом серверу Zabbix

Для того чтобы изменения заработали, следует перезагрузить агента (рис. 10.38).

Рис. 10.38. Команда перезагрузки агента

5. Для мониторинга данных через веб-интерфейс Zabbix необходимо добавить шаблон агенту. Заходим в *Configuration* и затем *Hosts*. Там нажимаем на ссылку самого сервера (рис. 10.39) и переходим к настройке мониторинга.

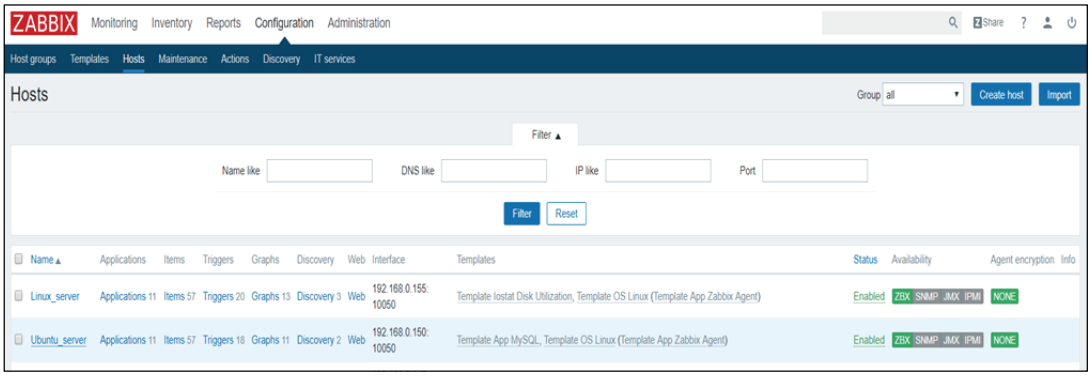


Рис. 10.39. Изменение агента

6. Переходим в *Templates* и через *Select* находим необходимый шаблон. Данный шаблон уже есть в самом Zabbix, так что просто кликаем на шаблон *Template App MySQL* (рис. 10.40), затем на маленькую ссылку *Add*, после чего на синюю кнопку *Update* (рис. 10.41). Важно не забыть нажать *Update*, иначе изменения не вступят в силу.

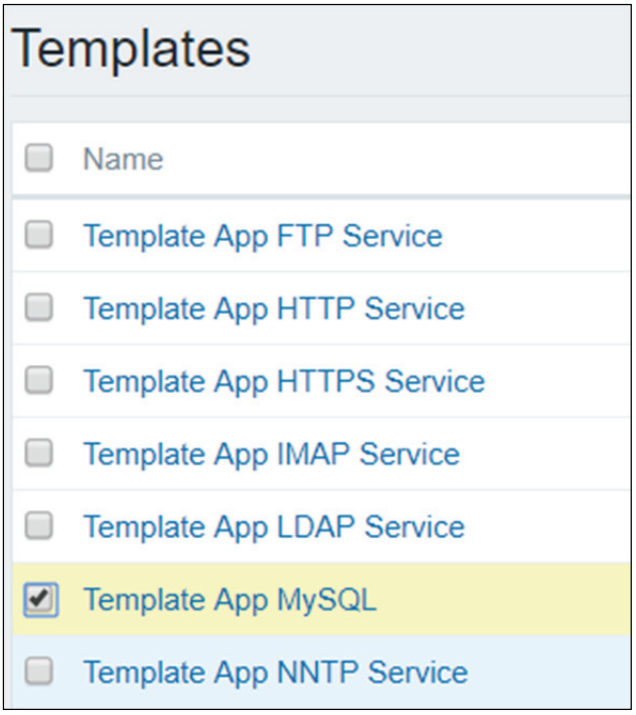


Рис. 10.40. Выбор необходимого шаблона

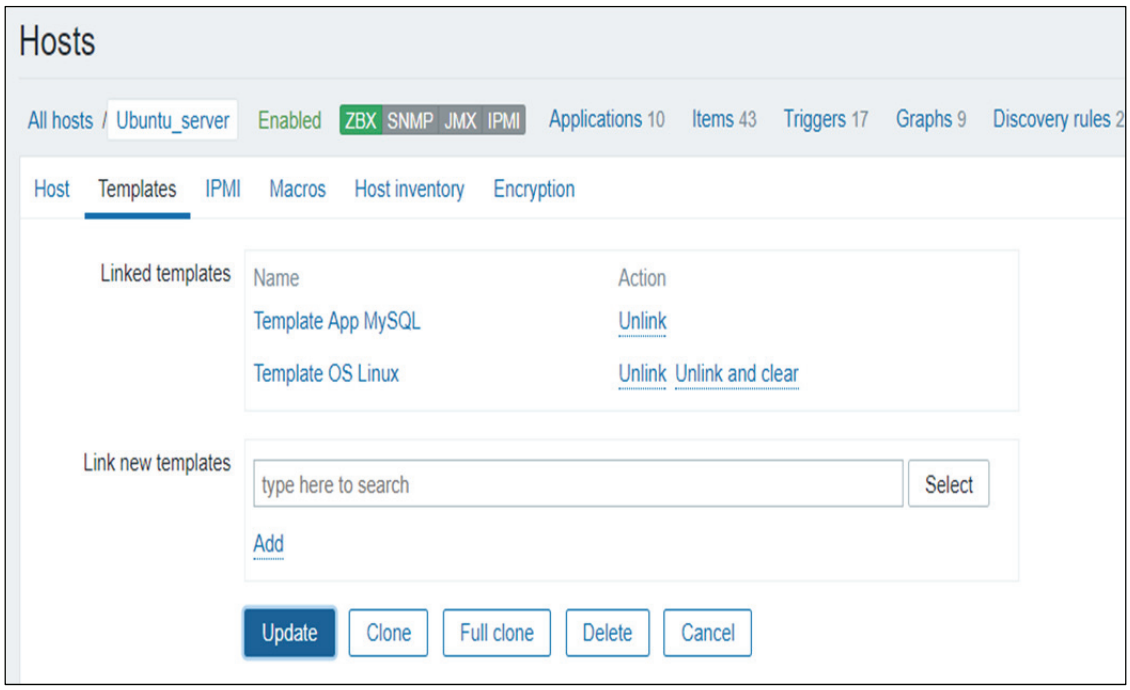
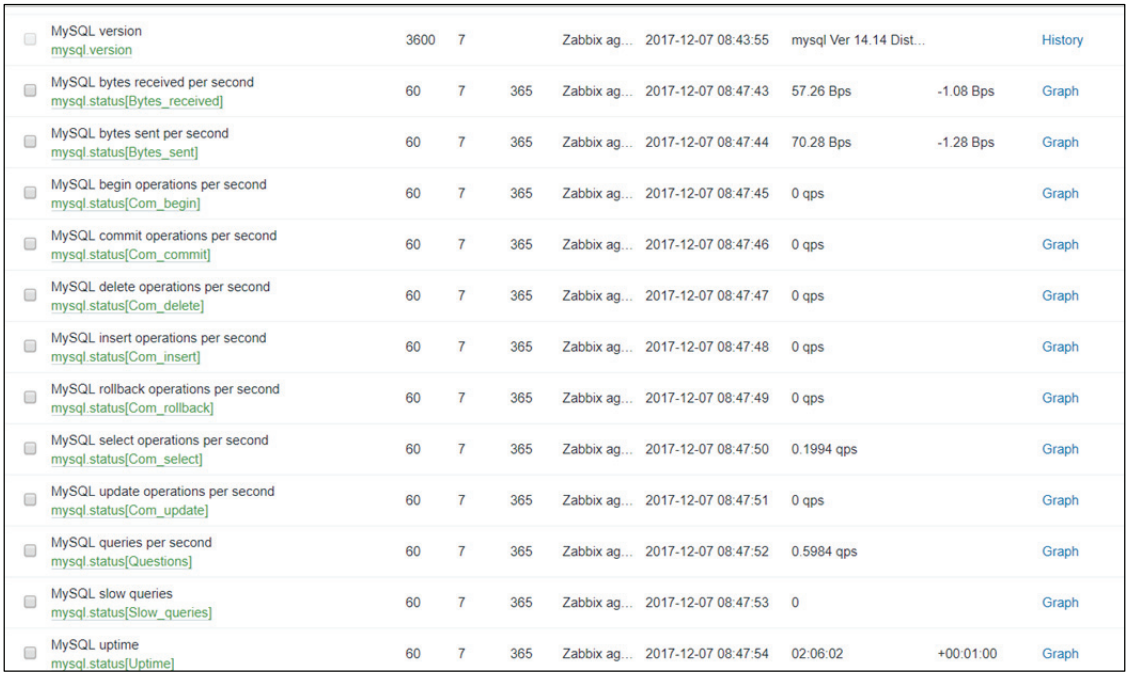


Рис. 10.41. Подтверждение изменений

После успешного добавления и корректной настройки можно увидеть, что Zabbix-сервер получает данные с Zabbix-агента, как показано на рис. 10.42.



MySQL version mysql.version	3600	7		Zabbix ag...	2017-12-07 08:43:55	mysql Ver 14.14 Dist...		History
MySQL bytes received per second mysql.status[Bytes_received]	60	7	365	Zabbix ag...	2017-12-07 08:47:43	57.26 Bps	-1.08 Bps	Graph
MySQL bytes sent per second mysql.status[Bytes_sent]	60	7	365	Zabbix ag...	2017-12-07 08:47:44	70.28 Bps	-1.28 Bps	Graph
MySQL begin operations per second mysql.status[Com_begin]	60	7	365	Zabbix ag...	2017-12-07 08:47:45	0 qps		Graph
MySQL commit operations per second mysql.status[Com_commit]	60	7	365	Zabbix ag...	2017-12-07 08:47:46	0 qps		Graph
MySQL delete operations per second mysql.status[Com_delete]	60	7	365	Zabbix ag...	2017-12-07 08:47:47	0 qps		Graph
MySQL insert operations per second mysql.status[Com_insert]	60	7	365	Zabbix ag...	2017-12-07 08:47:48	0 qps		Graph
MySQL rollback operations per second mysql.status[Com_rollback]	60	7	365	Zabbix ag...	2017-12-07 08:47:49	0 qps		Graph
MySQL select operations per second mysql.status[Com_select]	60	7	365	Zabbix ag...	2017-12-07 08:47:50	0.1994 qps		Graph
MySQL update operations per second mysql.status[Com_update]	60	7	365	Zabbix ag...	2017-12-07 08:47:51	0 qps		Graph
MySQL queries per second mysql.status[Questions]	60	7	365	Zabbix ag...	2017-12-07 08:47:52	0.5984 qps		Graph
MySQL slow queries mysql.status[Slow_queries]	60	7	365	Zabbix ag...	2017-12-07 08:47:53	0		Graph
MySQL uptime mysql.status[Uptime]	60	7	365	Zabbix ag...	2017-12-07 08:47:54	02:06:02	+00:01:00	Graph

Рис. 10.42. Данные, получаемые сервером Zabbix

10.3.3. Построение карт сети

Для удобства и наглядности создается карта сети. Настройка карты сети в Zabbix требует сначала создать карту, определив ее общие параметры, и затем заполнить карту элементами и связями между этими элементами.

Можно заполнять карту элементами: узлами сети, группами узлов сети, триггерами, изображениями или другими картами сети.

Для отображения элементов карты используются иконки. В них указывается информация, которая будет отображена с иконками, и устанавливается, какие недавние проблемы будут отображаться особым образом. Можно связать иконки и задать информацию, которая будет отображаться у связей.

Можно добавить пользовательские URL, которые будут доступны при нажатии на иконках. Таким образом, можно связать иконку узла сети со свойствами узла сети или иконку карты сети с другой картой.

Карты сети, которые уже готовы, можно просмотреть в *Monitoring → Maps*. На странице мониторинга можно нажать на иконки и воспользоваться ссылками на какие-нибудь скрипты или URL.

Все пользователи Zabbix (включая пользователей не администраторов) могут создавать карты сети. Карты сети имеют владельца – пользователя, который создал их.

Карты сети экранов можно сделать публичными или приватными. Публичные карты сети видимы всем пользователям, однако они

должны иметь права на чтение всех элементов карты сети, чтобы ее увидеть. Для добавления элемента на карту сети пользователь также должен иметь права как минимум на чтение этого элемента.

Карты сети видимы их владельцам. Владелец может давать общий доступ к приватным картам сети другим пользователям и группам пользователей. Обычные (не *Суперадминистраторы*) пользователи могут назначать общий доступ только тем группам и пользователям, в которые они входят сами. Приватные карты сети будут видны своим владельцам и пользователям с общим доступом к этой карте сети так долго, пока они имеют права на чтение всех элементов карты сети. Пользователи уровня *Администратора*, пока они имеют права на чтение всех элементов карты сети, могут просматривать и редактировать приватные карты сети независимо от того, являются ли они владельцами и входят ли в список пользователей общего доступа.

Для создания карты выполним следующие операции.

1. Переходим в *Monitoring → Maps*.
2. Просматриваем все карты, как показано на рис. 10.43.

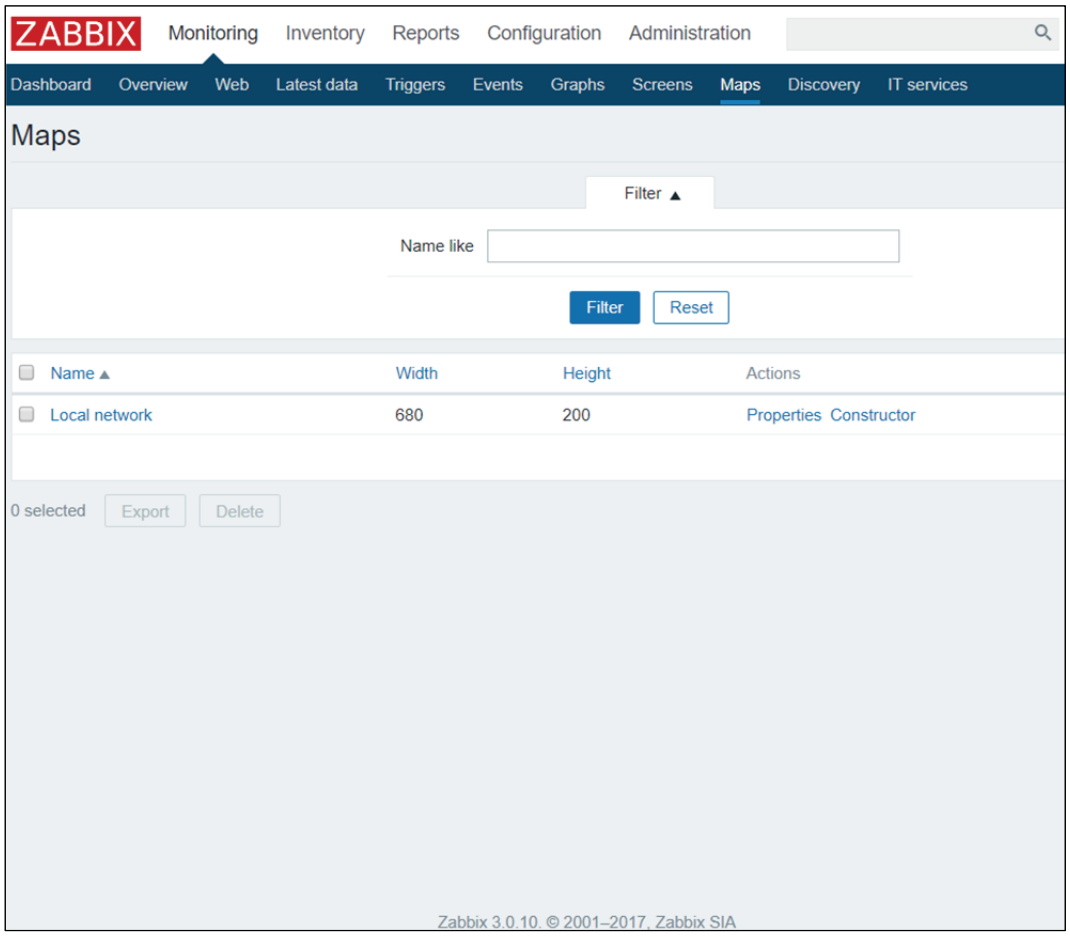


Рис. 10.43. Карты сетей

3. Нажимаем *Create map*. Здесь мы задаем имя нашей карты MY_NETWORK, выбираем размеры карты и информацию, которую она будет предоставлять (рис. 10.44).

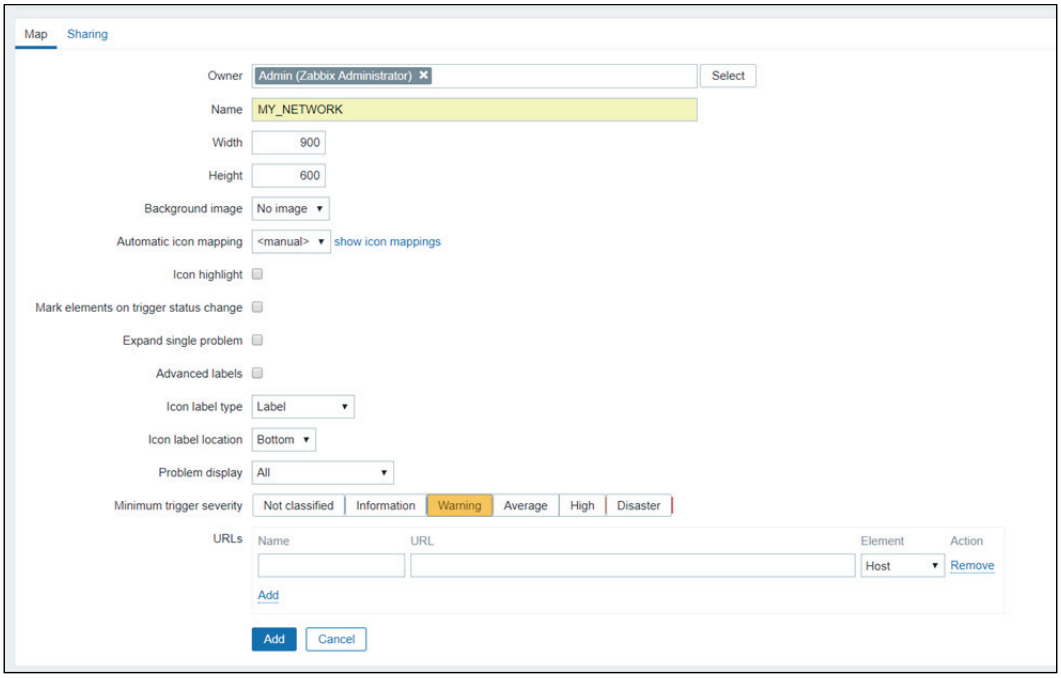


Рис. 10.44. Создание карты сети

При успешном создании карты получим следующий результат (рис. 10.45).

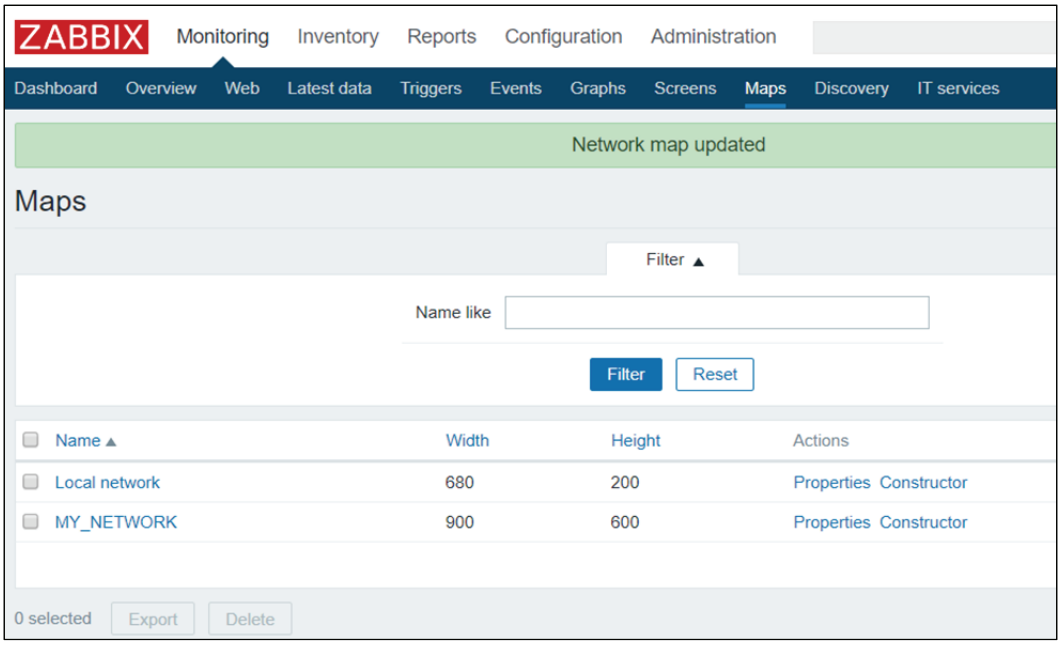


Рис. 10.45. Созданная карта сети в списке

4. Для дальнейшей настройки и добавления элементов нажимаем ссылку *Constructor*.

Чтобы добавить новый элемент кликаем на ссылку *Add* возле *Icon*. Новый элемент появится в левом верхнем углу карты. Его можно переместить, куда мы хотим.

Добавляем хост, который будет представлять собой наш Zabbix-сервер. Чтобы он таковым себя ассоциировал, кликаем на него и настраиваем (рис. 10.46): *Type – Host, Label – Zabbix.*

В *Host* нажимаем *Select* → *Zabbix Server* (имя хоста – Zabbix-server), для *Icons* можно выбрать любую иконку.

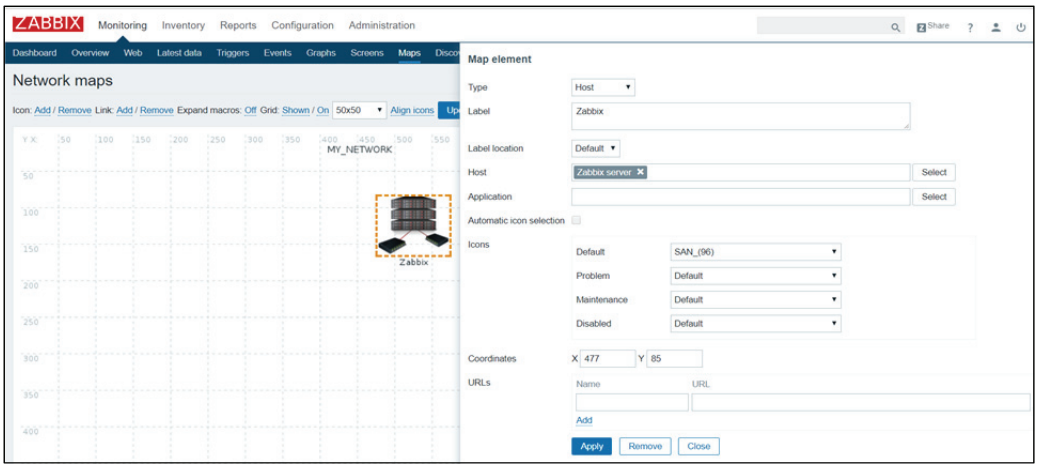


Рис. 10.46. Создание элемента и его настройка

Добавляем таким же образом другие серверы (рис. 10.47): *Type – Host, Label – Linux_server* (*Ubuntu_server, Windows_server*).

В *Host* нажимаем *Select* → *Linux_server* (имя сервера, который будем мониторить).

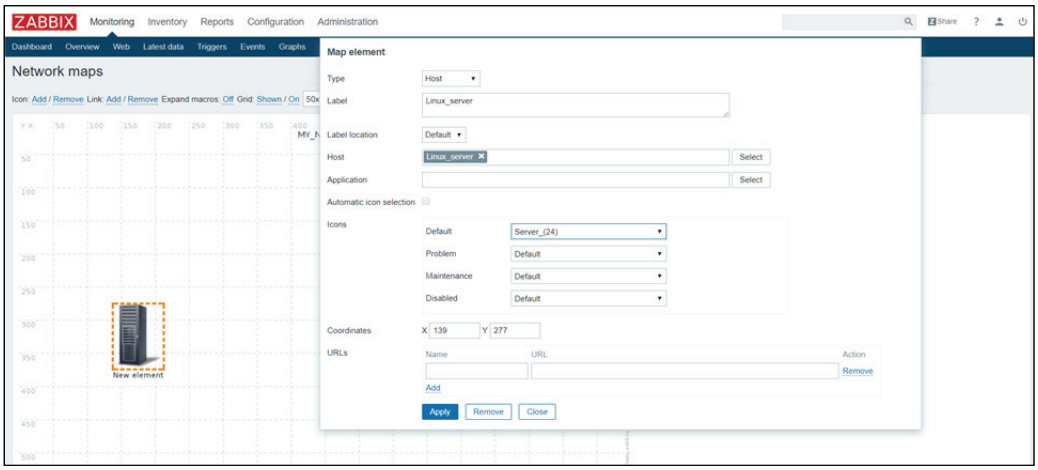


Рис. 10.47. Созданный сервер, который мониторим, и его настройка

5. После того, как мы расставили элементы на карте, самое время соединить их. Для соединения двух элементов мы должны сначала выбрать их.

Когда элементы будут выбраны, нажимаем в верхнем правом углу *Link*, кликаем ссылку *Add*. При наличии созданной связи диалог одного элемента теперь содержит дополнительный раздел *Link* (рис. 10.48).

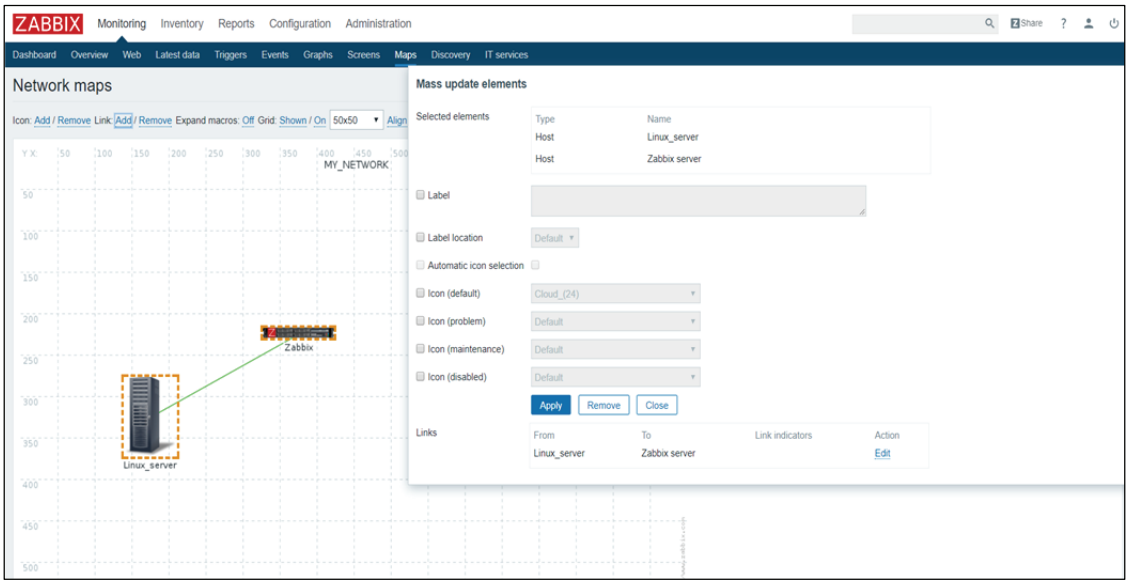


Рис. 10.48. Созданная связь (поле *Link*)

Щелкаем на один элемент и видим нашу связь. Можно ее настройки изменить, кликнув *Edit* (рис. 10.49).

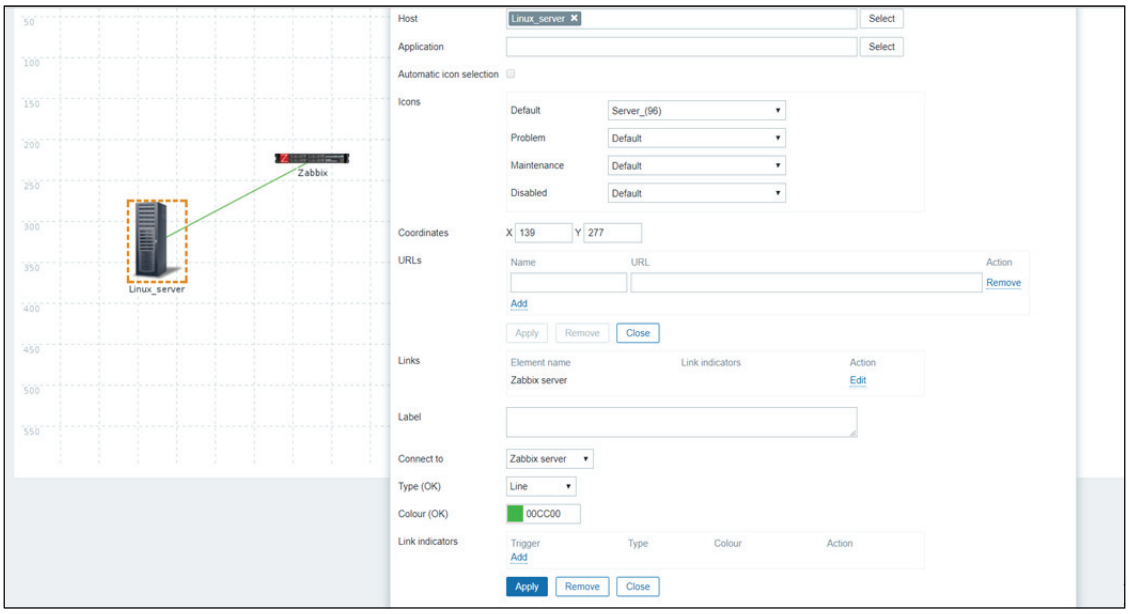


Рис. 10.49. Изменения связи

6. В карте добавляем надпись на связи, изменяем обычную линию на жирную и добавляем триггер, который следит за работой агента на серверах (для каждого сервера он свой).

Триггер добавляется ссылкой *Add*. Выбор триггера зависит от настройки *Host*. В настройках хоста был выбран шаблон *Template OS X* и вместо *X* операционная система, которую мониторим (Windows и Linux, как это показано на рис. 10.50). Таким образом, благодаря шаблону будут и триггеры, которые мониторят наш сервер. Выбранный в карте сети триггер следит за работой Zabbix-агента и показывает, если произошла ошибка.

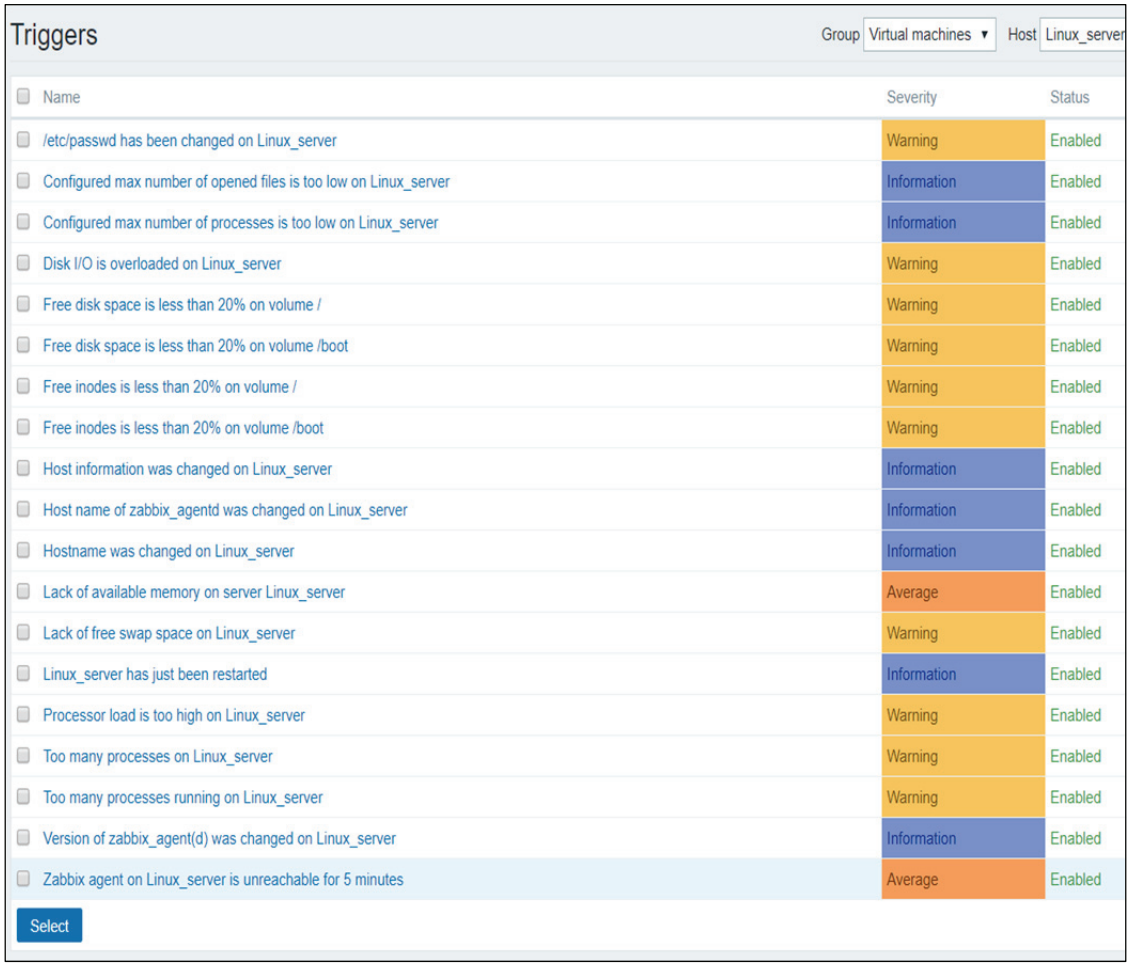


Рис. 10.50. Выбор триггера

После завершения получим результат, представленный на рис. 10.51. Видно, что все агенты на серверах работают корректно.

Продемонстрируем отключение агента на сервере. Наша связь поменяла цвет с зеленого на красный, и под иконкой Windows_server появилось объявление об ошибке (рис. 10.52).

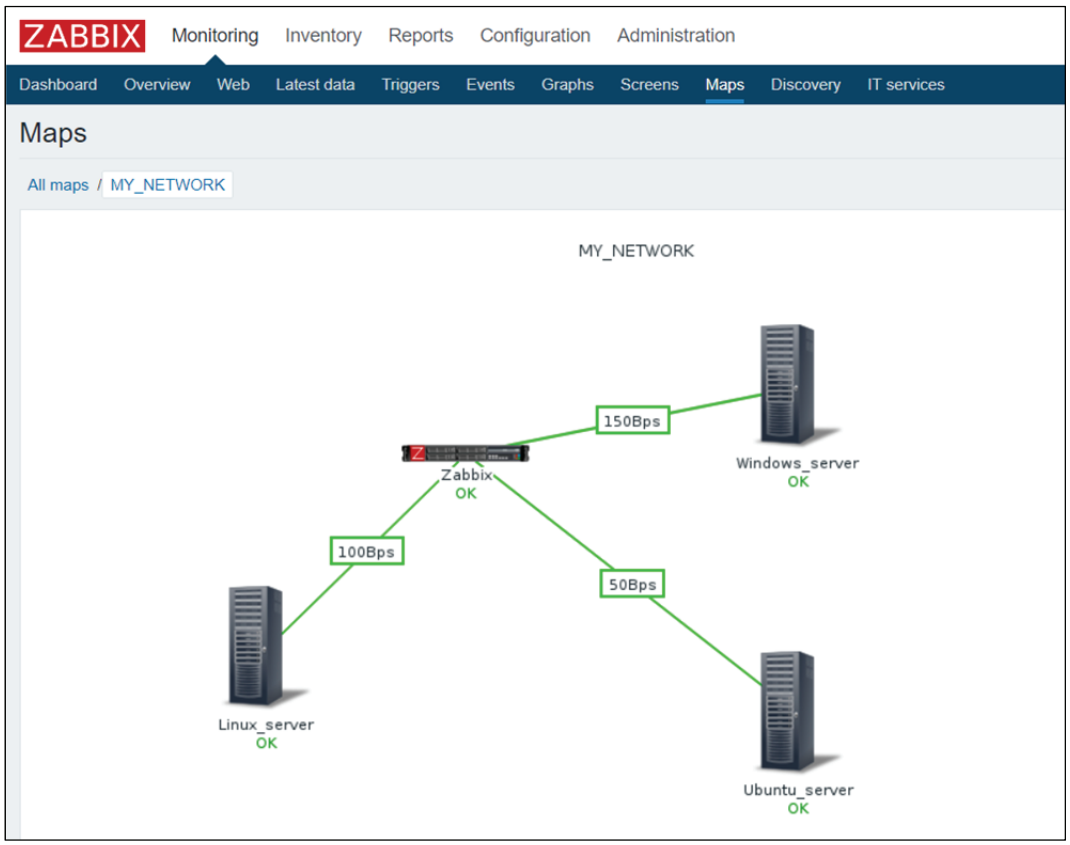


Рис. 10.51. Созданная карта сети

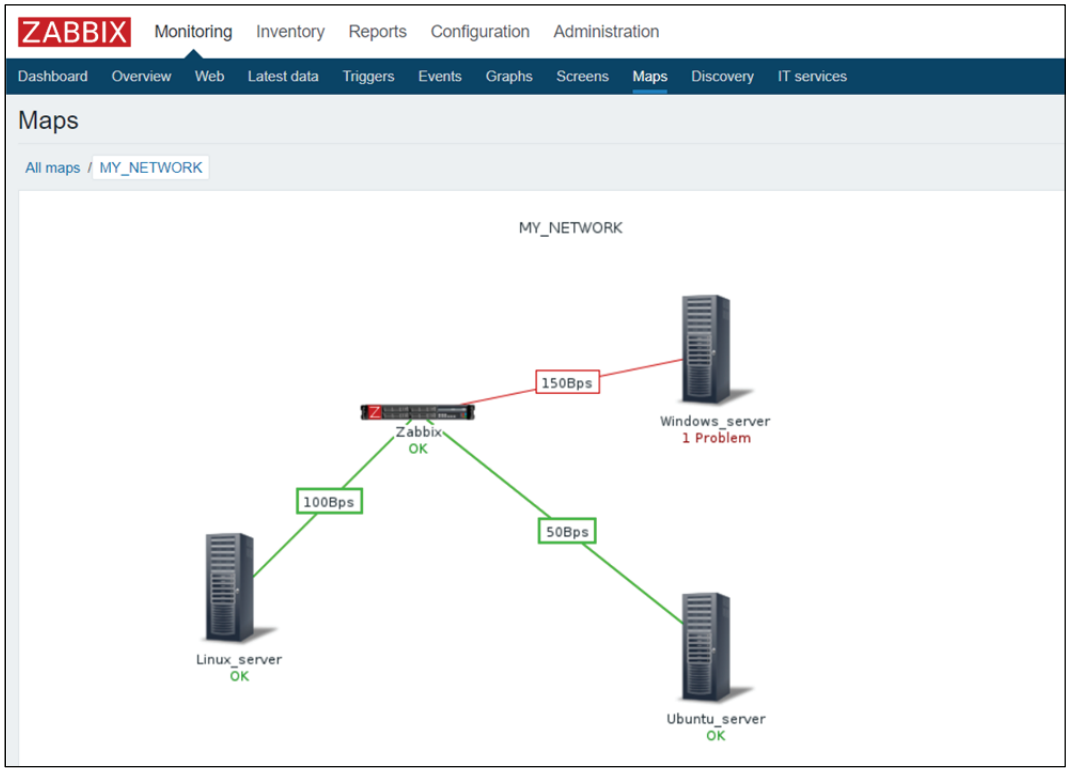


Рис. 10.52. Ошибка, отображенная на карте сети

Лабораторная работа № 20–22

***Цель*:** получение навыков по настройке интерактивных карт сети для мониторинга.

***Задание*:** необходимо произвести настройку карты сети в Zabbix. Вначале следует создать карту, определив ее общие параметры, и затем заполнить карту элементами и связями между этими элементами.

Карту необходимо заполнить узлами сети, группами узлов сети, триггерами, изображениями или другими картами сети.

Следует создать новый триггер для получения информации от коммутатора по протоколу SNMP, а также взаимодействовать с интерфейсом системы мониторинга, анализировать полученную информацию при изменении состояния операционной системы.

ЛИТЕРАТУРА

1. Урбанович, П. П. Компьютерные сети: учеб. пособие / П. П. Урбанович, Д. М. Романенко, Е. В. Кабак. – Минск: БГТУ, 2011. – 400 с.

2. Романенко, Д. М. Компьютерные сети. Лабораторный практикум / Д. М. Романенко, Н. В. Пацей. – Минск: БГТУ, 2011. – 133 с.

3. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – 5-е изд. – СПб.: Питер, 2015. – 944 с.

4. Линн, С. Администрирование Windows Server 2012 / С. Линн. – СПб.: Питер, 2013. – 304 с.

5. Романенко, Д. М. Компьютерные сети: учеб.-метод. пособие / Д. М. Романенко, Н. В. Пацей, М. Ф. Кудлацкая. – Минск: БГТУ, 2016. – 163 с.

Учебное издание

Романенко Дмитрий Михайлович
Миронов Игорь Александрович

АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ И ВЕБ-ПОРТАЛОВ

Лабораторный практикум

Редактор *Е. С. Ватеичкина*
Компьютерная верстка *А. А. Селиванова*
Корректор *Е. С. Ватеичкина*

Издатель:
УО «Белорусский государственный технологический университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/227 от 20.03.2014.
Ул. Свердлова, 13а, 220006, г. Минск.