Бегущий админ, – это очень плохая примета.

Настоящий айтишник с трудом может прочитать правильно (и с первого раза) слово север даже в фантастической литературе.

Посвящается, само собой, родителям, жене, знакомым, друзьям и моим постоянным читателям.

Да пребудет с Вами… BSOD в его призрачном, но не физическом виде.

idkfa

iddqd

Антипиратская позиция

Проект придерживается определенных правил и стандартов, принятых на наших сайтах. Они же приняты и в этой книги. Я бы хотел их прокомментировать. На форуме у нас есть правило, которое гласит:

> 1.1 Здесь категорически запрещено обсуждение вареза (как на самом форуме, так и в личной переписке), т.е. взломанных (пиратских) программ, ОС (сборки в частности) и игр, а также кряков/взломщиков к оным, а также, само собой, распространение оных в любом виде - ссылки, файлы и тому подобное.

Я хочу дать раз и навсегда разъяснения почему мы очень жестко придерживаемся подобной позиции и принимаем крайне суровые меры в случае нарушения вышеозначенного пункта.

Как Вы надеюсь знаете, администрация проекта на всех его сайтах (в том числе **sonikelf.ru**) придерживается концепции взаимного уважения, лояльности и дружелюбия пользователей к пользователям, а также пытается создать комфортную обстановку для всех посетителей путем интеграции норм и концепций общения в качестве части правил ресурсов.

Как это связано с антипиратской политикой? Очень просто: любой труд, а особенно труд в области разработки программного обеспечения, игр и любых других интеллектуальных продуктов, должен уважаться и это уважение вполне можно описать в старой-доброй заповеди: "не укради". Это уж если не взывать к не менее (а местами и более) правильному высказыванию: "любой труд должен быть оплачен". Пользуясь же взломанными версиями программ, игр, операционных систем и тп, Вы идете вопреки вышенаписанным высказываниям, а значит проявляете неуважение к ресурсам других людей (причем, зачастую, очень многих, ибо в создании какой-нибудь игры зачастую участвуют сотни человек), а именно к затраченному на разработку времени, силам (моральным и физическим) и деньгам. Учитывая, что, как Вы помните, уважение — это основа политики, на которой построен этот проект, - Вы нарушаете не только человеческие нормы и принципы морали, но и наши правила.

Вдвойне грустно это выглядит в свете того, что у каждой платной программы есть множество бесплатных аналогов, которые можно использовать, тем самым, не воруя у других разработчиков средства, силы, время и прочие ресурсы. Тем более, почти все современные программы дают время (как правило это 30 дней) на бесплатное тестирование возможностей и принятия решения о покупке (либо же выбор пути удаления программы, что, кстати нормально, ибо за 30 дней часто можно решить свои проблемы и с

чистой совестью пойти дальше). Не все понимают, что стоимость программы — это не просто деньги в чей-то карман, а вполне реальная благодарность за проделанную работу вполне живым людям, у которых тоже есть семья, проблемы и желание кушать. В качестве примера могу привести вполне простую аналогию: представьте, что вместо того, чтобы выдать зарплату, Ваш начальник нашел какую-нибудь хитрость, позволяющую, во-первых, эту зарплату не платить, а, во-вторых, пользоваться Вашими услугами бесплатно и дальше (ну или передать хитрость другим работодателем, чтобы те кидали Вас точно так же, сразу после первого месяца или недели Вашего честного труда). Если Вас это не волнует и деньги Вам не важны, то всё это понятно, но нормальный человек проникнется, подумает и поймет, о чем я.

Ну и наконец... Антипиратская политика близка мне, автору, именно потому, что я и команда этого проекта знаем, какой огромной ценой (деньги, силы, человекочасы и тп) и трудами часто достается создание чего-либо, помогающего людям решать их проблемы. Откуда знаем? Дык мы сами вот уже несколько лет занимаемся оным на всех сайтах линейки [Sonikelf's Project's], будь то статейная, форум или новостной портал (и, уж поверьте, это дается не просто), а посему мы уважаем и ценим труды других разработчиков ПО, игр, операционных систем или интернет-проектов, что выражается как в том, что мы рекомендуем исключительно бесплатный софт, так и в жесткой антипиратской

политике по удалению ссылок на варезные программы, сайты, программы-взломщики (Crack, keygen), отказываемся консультировать по вопросам обхода проверок подлинности Windows и прочее прочее прочее.

Как-то так. Это, повторюсь, лежит и в основе всех книг и проекта, так что имейте ввиду.

Глава 0. Вводная

Несколько слов лирики. О книге и авторе

Рад приветствовать всех тех, кому довелось читать эту книгу. Я объективно считаю, что эти строки заслуживают только хорошие и адекватные люди, а раз Вы это читаете, то определенно так оно и есть. За что Вам огромное спасибо.

А, и конечно же, доброго времени суток. Да-да, снова эта ненавистная фраза, с которой я начинаю огромное количество своих статей и теперь еще и книг. Подошло время публикации второй моей крупной работы, которую многие из Вас ждали и/или не ждали, а просто злорадствовали на эту тему, но серверная книга, так или иначе, явилась на свет.

В рамках **ПУВС** (Персональный Угол В Серверной) мы поговорим о том, о чем говорить якобы “вредно”, но чего Вы ждали и хотели достаточно давно, а именно: о серверах, сетях и всём таком прочем. Под “говорить вредно” я имею ввиду то, что информация, которая представлена в этой книге, во многом “скрывается” от простых пользователей и **IT**-шников, либо её попросту нет в том виде, в каком она должна быть.

Конечно это не вселенский заговор, просто многим профессионалам из этой сферы не свойственно уметь (и хотеть) доступно, подробно и

полезно писать на эти темы, делиться знаниями и всё такое прочее. В отличии от Вашего покорного слуги ;)

Начиная книгу, я хочу высказать слова благодарности тем, кто заказывал первую и вторую книгу (а также будет заказывать последующие), кто всегда с проектом и кому было интересно то, что я делаю и/или кто читает это по факту заказа и/или предзаказа (а не скачав где-нибудь из вредности, жадности и иных побуждений).

Традиционно я заранее предупреждаю всех, что книга написана с налётом лирики, т.е личных мыслей автора, а посему много здесь будет, как полезного с практической точки зрения, так и просто различных размышлений в формате прозы. Это не сухой библиотечный учебник, а авторский материал, вышедший из-под руки профессионального **IT**-шника.

Об авторе: кто, зачем, почему и всё такое прочее

Само собой, что не написать этот кусок было бы странно, а то вдруг Вы читаете этот материал спонтанно, случайно, впервые, не будучи знакомы с моими статьями, да еще и где-то добыв эту книгу вне рамок сайта **sonikelf.ru**, входящего в проекты **[Sonikelf's Projects']**.

Собственно, что можно сказать обо мне и моём опыте? На данный момент я молодой (хотя что считать молодостью?), но опытный специалист в своей сфере - **IT**. Да, многие представляют Соника, то

бишь меня, то бишь автора проекта, как сурового бородатого и умудренного опытом дядьку, но на самом деле это частично далеко от реальности. Во всяком случае я не столь суров, хоть и бородат, а уж про опыт...

Если говорить про опыт - то перед Вами человек, стоявший, если не у истоков, то основ индустрии, а если откинуть бравады, то речь идет о временах, когда первый пентиум с частотой **133 МГЦ** был в квартире обычного человека страшно мощной машиной, докинутые несколько мегабайт оперативной памяти в многоизвестной игрушке **DOOM** позволяли обгонять ракету, а установка **95**-ой **Windows** не была чем-то из ряда вон выходящим, равно как и компьютер без **Windows** вообще. Конечно, когда компьютеры были большими и ламповыми я не застал, но всё вышенаписанное - вполне. Это если говорить о временных промежутках.

А так, если брать в цифрах, за моими плечами примерно 15 лет работы самой разной направленности в **IT**-сфере, причем от типичного мастера-на-все-компьютерные-руки (эникей) и системного администратора (в мощном, так сказать, его понимании), до **IT**-руководителя команд разного уровня, целей и направлений. В мой опыт входит довольно большой пласт знаний из самых разных сфер, хотя часть из них (вроде программирования) мною не очень любимы, а потому набросаны по верхам. Тем не

менее, это позволяет мне успешно справляться практически с любыми целями и задачами или искать профессионалов для их решения.

В настоящий же момент я веду известный сайт **sonikelf.ru**, представляющий проект **[Sonikelf's Project's]**, администрирую (в том числе удаленно) сервера, занимаюсь созданием-продвижением-монетизацией сайтов, обучением специалистов в IT-индустрии и много чем еще. В недавнее время понемногу начал заниматься разработкой мобильных приложений, в связи с чем, возможно, напишу еще одну книгу и/или пакет обучения.

В двух словах как-то так. Чуть больше Вы можете узнать на вышеупомянутом сайте или у меня лично... через форму обратной связи на вышеупомянутом сайте :)

Информация для критиков

Заранее про критику. Всё, что Вам надлежит прочитать, есть мой личный опыт, основанный на многолетней практике, повсеместно применяемой в разных местах, причем не только мной, но и моими коллегами, учениками, а также пользователями, читавшими и читающими мой сайт **sonikelf.ru** из линейки сайтов проекта **[Sonikelf's Project's]**. Всё это проверено не только огромным количеством людей огромное количество раз, но и временем. Посему, имейте в виду несколько нюансов:

- Критикуя мой опыт, т.е. описанное в книгах и статьях, будьте готовы аргументированно объяснить собственную позицию, а именно что, когда, почему и в результате каких действий (и в каком количестве), Вас не устроило, не устраивает и не будет устраивать, а также почему так, а не иначе;
- Критикуя мой опыт, рекомендую иметь за плечами собственное предложение того, как на самом деле правильно. Если Вы заявляете, что предложенное мною решение плохое, то будьте готовы объяснить, чем оно плохое, предоставить аргументы по этому вопросу, альтернативное решение, а также объяснить, чем альтернативное решение лучше;
- Критикуя мой опыт, будьте готовы рассказать кто Вы и что из себя представляете, какими знаниями владеете, сколько раз проверили свои и мои методики и тп и тд. Профессионал ли Вы с многолетним стажем или просто пользователь?

Я всегда рад объективной и адекватной критике (особенно от коллег), ибо для меня важно перенимать опыт и новые решения, совершенствовать себя как профессионала и расширять сферу своих знаний.

Но имейте в виду, что принятые в русском сообществе не аргументированные выкрики в духе “Эта программа ****, автор ****!” здесь не работают и восприниматься всерьез не будут. Более того, подобные люди будут идти... лесом. Остальные - добро пожаловать, мои контакты найти легко на моём же сайте.

Общая информация о книге

Очень большое количество достаточно опытных пользователей, эникеев и начинающих сисадминов не развиваются дальше и боятся серверов как огня, ибо считают, что это безумно сложная тема, требующая миллионы часов на изучения и **IQ** выше **100 000**.

Примерно то же самое касается локальных сетей, которыми все повсеместно пользуются, но совершенно не понимают, как они работают и не умеют (или не хотят уметь) их строить, распознавать, настраивать и оптимизировать, боятся виртуализации и прочих админских тем.

Цель книги, — показать, что сервера и сети, да и прочие админские вещи, вовсе не такая страшная штука, как кажется на первый взгляд, предоставить базовые знания по теме и дать достаточно информации, чтобы развиваться в этом направлении.

К сожалению, вместить всё необходимое в одну книгу попросту невозможно, поэтому, скорее всего, если Вам понравится, то будет еще несколько частей-продолжений **ПУВС**, т.е то, что Вы держите сейчас в руках или читаете на экране монитора, собственно, только начало.

Послесловие

Как и в первой книге, здесь собран мой опыт, который изложен достаточно простым языком и затрагивает самые базовые и необходимые знания из области администрирования, а именно серверов и сетей.

Спасибо Вам за помощь и поддержку, если эту книгу Вы приобрели на сайте **sonikelf.ru** и да настигнет Вас святой **BSOD**, если книга получена Вами не самыми добрыми методами.

Пора начинать.

Глава 1. Системное администрирование

Предисловие к первой главе

Начнем с основ, которые так часто не понимают или с чем-то путают даже профессионалы, не то что обычные пользователи, а именно поговорим о том, что за человек такой системный администратор (или IT-шник из какой-либо сферы, не суть), какими качествами он должен обладать, чем мыслить, жить и всё такое прочее.

Во второй же главе поговорим в общих чертах о том, что есть сервера, зачем они нужны, какие бывают и какие роли выполняют.

Всё нижесказанное, как и сама книга, является мнением, подтвержденным личным опытом, опытом коллег, а также техниками моделирования и анализа.

Концепция: человек и технологии

Что за “зверь” такой и каков он есть/должен быть

Настоящий системный администратор, да и просто **IT**-шник из любой сферы, - это прежде всего человек, не только и не столько

обладающий определенными навыками, сколько имеющий определенный склад ума, направление мышления и любовь, собственно, к этой сфере, т.е к управлению компьютерами, серверами, сетями и прочей айтишной (и не очень) техникой.

Вспомните своих учителей в школе/университете, и Вы поймете, о чем я, ибо те из них, кто пришел по собственному призванию, вели уроки действительно интересно, их было здорово слушать, ведь они горели своей работой, им нравилось то, что они делали и соответственно болели за качество, а значит старались и у них получалось. И наоборот, были унылые, скучные люди, находящиеся не на своём месте, не в своё время и явно не увлеченные своей работой.

Так и в компьютерном мире, настоящим специалистом, у которого многое получается и который реально является профессионалом, является именно тот, кому близка эта сфера, кто находится, как любят говорить, на гребне волны, т.е постоянно постигает новые технологии, решает появляющиеся проблемы и ищет новые способы их решения/предотвращения и всё такое прочее. Говоря простым языком, системный администратор, во всяком случае хороший - это человек прежде всего по призванию, потом по навыкам, и далее всё остальное. Причем первое (призвание) тянет за собой второе (знание и навыки) и наоборот. Это... Сложно объяснить словами.

Наверное, это Настоящий врач, но просто в **IT**. Ему нравится эффективно создавать, настраивать и использовать своё окружение и делать, чинить, налаживать, исправлять, улучшать систему до той степени, когда она работает идеально, стабильно и без сбоев. Причем не просто старается, но и хочет это сделать. И делает.

Дело в том, что современный айтишник это, как не крути, не просто.. специалист узкого профиля (что бы там не вещали пафосные личности кричащие, что все, кто знает больше одной-двух тем, собственно, нелепые эникейщики), а, скажем так, вполне себе мастер на все руки, который и в смартфоне может разобраться/починить/настроить и видеонаблюдение собрать при необходимости, и **SIP**-повесить, и сервер поднять-настроить-починить, и сайт сделать. Причем всё это вполне качественно и, что называется, с душой.

Т.е ему (айтишнику) нужно нужно быть готовым ко всему, стремиться охватывать все сферы своей области и стараться, помимо чтения развлекательных порталов (литературы), изучать как можно больше разного рода тематических книг, статей, мануалов, новостей и всего такого разного, т.е, как я уже говорил выше, собственно, быть в теме и жить на “гребне волны” индустрии.

Но ведь IT-сфера очень огромна. Что же делать?

Да, невозможно знать всё к сожалению, сфера слишком масштабна, чтобы даже прочитав все книги по теме, понять и запомнить

решения всех проблем, конфигураций и тд и тп. Тем не менее, обладая определенным образом мышления и рядом основополагающих вещей, можно уметь буквально всё, не имея фактически этого в голове и не умея решать проблему еще минуту назад, до её возникновения.

Как ни крути, самое важное это базис знаний (и практика на основе этого базиса) за плечами и мышление, т.е понимание сферы и того как в ней всё работает, плюс желание и умение искать, находить и применять на практике.

Плюс, опять же, одно из главных качеств профессионального айтишника... Это интуиция. Я серьезно, имея за спиной даже опыт обычного домашнего пользователя, Вы должны уметь чувствовать и понимать, как именно всё это работает, повторюсь, не бояться и уметь искать решения, причем искать как самостоятельно, так и в интернете, в советах коллег или где-то еще. А уж отсутствие страха гуглить, читать и делать - это вообще что-то из области золотого ключика.

По сути, нет такой проблемы, с которой не может справится нормальный системный администратор, который просто способен думать головой, анализировать и использовать различные подходы в сложившейся ситуации. Современный интернет, а он есть везде, предоставляет Вам доступ к бесконечному количеству информации,

используя которую Вы можете справиться почти с любой задачей и обучиться почти любым навыкам. Было бы желание.

Ну и само собой... Надо практиковать, практиковать, практиковать, в поисках действительно рабочих вариантов или устраивающих по скорости или эффективности.

Такие дела.

Общее послесловие по теме первой главы

Я знаю, что в этой главе написал много раз одно и тоже, но разными (а где-то буквально одними и теми же) словами, но мне важно, чтобы Вы поняли - было бы желание и руки, всё остальное, как говорится, дело техники. Тем более в современном-то мире.

И да, не слушайте тех, кто говорит Вам, что много знать и уметь плохо, Вы эникей и прочее - это слова как минимум лентяев. Если не сказать больше.

Что касается выбора направления администрирования (это один из самых частых вопросов) в плане операционных систем - я говорил это в первой книге и повторюсь сейчас.

Полезно и правильно уметь администрировать (или хотя бы иметь обширное представление) и в **Linux**, и в **Windows**, но куда важнее просто стать хорошим, годным, адекватным специалистом хотя бы в

одном направлении. Да и поменять его (или подтянуть знания), собственно, никогда не поздно.

Да пребудет с Вами опыт.

Глава 2. Сервера и операционные системы

Предисловие к второй главе

Давайте начнем с основ, а именно поговорим о том, что такое вообще сервер, зачем он нужен, какие роли выполняет и так далее, и тому подобное.

А также затронем тему операционных систем в контексте их применения в рамках серверных ролей и других нюансов. Поехали.

Что есть сервер и зачем он нужен

Сервер, по сути это устройство, выполняющее определенные серверные/сервисные (узконаправленные) функции. Сервером, как ни странно, может являться обычный компьютер и/или вообще любое устройство, хоть мобильный телефон, хоть роутер. Дабы не изобретать велосипед, процитирую Википедию.

> Сервером называется компьютер, выделенный из группы персональных компьютеров (или рабочих станций) для выполнения какой-либо сервисной задачи без непосредственного участия человека.
>
> Сервер и рабочая станция могут иметь одинаковую аппаратную конфигурацию, так как различаются лишь по участию в своей работе человека за консолью.

> Некоторые сервисные задачи могут выполняться на рабочей станции параллельно с работой пользователя. Такую рабочую станцию условно называют невыделенным сервером.

Что касается консоли, то все Вы её наверняка видели в **Windows** и **Linux**, но в этом случае речь не совсем о командной строке. Консоль в данном случае - это физический элемент управления, т.е обычно это сочетание монитор+клавиатура+мышь, которое необходимо только на стадии первичной настройки, и\или при аппаратно-техническом обслуживании и управлении в нештатных ситуациях (штатно же большинство серверов управляются удаленно). Для нештатных ситуаций сервера обычно обеспечиваются одним консольным комплектом на группу.

Что касается мощностей, то любая компания или организация имеет вполне определенные цели и важные бизнес-задачи (даже если они явно не прописаны и Вам о них не рассказали, т.е компания маленькая). Роль этих задач и масштаб предприятия часто являются определяющими факторами для установки выделенного сервера (т.е отдельной физической машинки), обеспечивающего выполнение этих жизненно важных для компании задач и приложений. Исходя из этих же целей и задач и подбирается конфигурация сервера с точки зрения аппаратных и программных нюансов.

В зависимости от задач и условий использования, сервер имеет следующие основные параметры:

- Производительность;
- Надежность;
- Масштабируемость;
- Управляемость.

Давайте кратко рассмотрим каждое из этих направлений.

Производительность сервера

Как ни “странно”, производительность сервера зависит от его комплектующих, т.е. всё примерно то же самое, что и у обычных рабочих станций, домашних компьютеров и прочей техники. Логично, что чем производительнее его комплектующие, тем он работает шустрее.

Правда в отличии от обычных домашних компьютеров, в серверах как правило, применяются более надежные и производительные решения, вроде двух и более процессорных плат, огромного количества ядер, оперативной памяти (с коррекцией ошибок) и дисковых массивов, RAID. Всякие видеокарты и прочие разности здесь вторичны, но тем не менее сервер на класс превосходит мощности обычных ПК, правда в своей нише (обсчет поставленных задач).

В принципе это всё, что стоит тут говорить, и ничего очень уж необычного нет. Детальнее поговорим в следующих главах.

Надежность сервера

Надежность - это отдельный показатель, который отличает, опять же, сервера от обычных компьютеров (но всё еще не забывайте, что обычный компьютер вполне может выполнять серверные функции и быть им, но тогда показатель надежности, скорее всего, будет крайне низким).

Чем отличает?

Как минимум тем, что серверное оборудование и комплектующие на порядок или несколько порядков более безопасны, надежны и отказоустойчивы, т.е. сделаны из дорогих и качественных компонентов, рассчитанных на длительную работу, т.е. показатель наработки до отказа куда выше (при отсутствии внешних факторов), чем у обычных решений.

Изначально в серверную платформу (материнская плата + процессор + память + дисковый массив) закладываются разработки, увеличивающие долговечность и надежность данной машины. Надежность не только в отношении того, что он будет работать и не зависать, глючить и тормозить, что часто бывает с домашним компьютером. В него заложено два вида надежности, это:

- **Физическая надежность** – стабильность работы, достигаемая надежными комплектующими и качественной сборкой в целом;

- **Аппаратная надежность** – отсутствие программных сбоев за счет стабильности работы аппаратной части.

Например, есть такая штука, как технология коррекции ошибок памяти – **ECC**. Эта вещь позволяет предотвращать появление сбоев в работе памяти (при записи и чтении внутри чипов памяти) и избегать перебоев работы, которые, например, в обычном случае приводят к синим экранам смерти.

Или же **RAID**-массивы, которые построены зачастую по принципу зеркалирования (рейды 1, 0+1, 1+0, 5 и тд и тп) и используют несколько дисков для дублирования друг друга, с целью повышения не только и не столько скорости работы, сколько надежности, ибо при выходе из строя одного из накопителей, другой дублирует его, что называется, на лету, позволяя продолжить работу без прерывания процесса, перезагрузок и тп.

Есть ли такое в домашнем случае? Есть, но редко, т.е. только если пользователь с этим специально заморачивается. В серверной среде это является стандартом, ибо минута простоя работы компании может измеряться миллионами, в то время как дома можно перезагрузить компьютер или забросить его на недельку до покупки новой запчасти.

Масштабируемость серверов

Что есть масштабируемость? Ну, говоря просто - это возможность увеличить вычислительную мощность сервера или операционной

системы (в частности, их способности выполнять больше операций или транзакций за определенный период времени, либо запускать больше различных служб) за счет установки большего числа процессоров, памяти, дисков и прочего, т.е. комплектующих.

Надо сказать, что сервера куда сильнее поддерживают эту возможность, нежели домашние решения, т.к. масштабируемость играет огромную роль в работе и позволяет на лету увеличивать вычислительные мощности, а значит объем решаемых задач.

Т.е. изначально сервера в продаже идут в базовой комплектации, но с заложенным потенциалом к «апгрейду» за счет добавления каких-либо ресурсов (**2**-го процессора, например). В обычных компьютерах это заложено на достаточно низком уровне, т.е. нет возможности увеличить производительность на порядок или несколько.

Управляемость сервера

Управляемость - это не что иное, как возможность удаленного мониторинга состояния всех систем и параметров, удаленное включение, выключение и перезагрузка сервера, а также возможность удаленной диагностики сервера даже в выключенном состоянии и использования системы прогнозирования неполадок.

В серверах уже давно внедряются модули, позволяющие делать с ним всю диагностическую работу удаленно, в любом его состоянии,

единственным из условий которых является наличие питания и линии сети (интернета или локалки).

Кратенько, пожалуй, как-то так. Идем дальше.

Классификация серверов в “природе”. Серверные роли

Сервера можно делить по назначению, выполняемым функциям или ролям. Или по всем этим направлениям одновременно. Сразу оговорюсь, что один сервер может замещать собой (совмещать в себе) все виды серверов описанных ниже.

Давайте разберемся чуть детальнее во всём этом.

Сервер рабочей группы и/или файл-сервер

Представляет собой простенький сервер начального уровня, зачастую он является однопроцессорным, не сильно мощным и вообще представляет из себя обычный компьютер, выполняющий функции сервера.

В этой стране такой подход использует большая часть небольших, средних и даже, что до сих пор поражает автора, крупных компаний, причем размещая машину прямо в офисе и вдобавок - да простит меня святой BSOD - порой не выделяя под него даже отдельного помещения.

По функциям обычно служит файлопомойкой (извините, хранилищем важных данных и файлов компании), а также выполняет функции разграничения прав доступа сотрудников к этим данным (бухгалтерам одно, менеджерам второе, гендирам вообще всё).

Контроллер домена

Необходим в организации с количеством сотрудников в несколько десятков рабочих мест и более.

Как правило позволяет централизованно управлять сетевыми и файловыми ресурсами компании, в частности исполняет роль принт-сервера (о нем ниже), сетевого шлюза и другие сопутствующие роли (в том числе выполняет функционал сервера рабочей группы, но в более широком понимании).

Логично, что **Domain Controller**, т.е контроллер домена, должен быть на порядок качественнее и надежнее сервера рабочей группы, иметь возможность масштабирования при росте количества пользователей локальной сети и тд и тп. Адекватные люди здесь уже не используют обычные десктопные машины, но... Такое всё еще встречается.

Как правило развертывается на **Windows Server** с **Active Directory**.

Прокси сервер (+-VPN) (+-шлюз) и тп

В этой роли серверная машина обеспечивает общий доступ в интернет всем (или определенным) компьютерам офиса, т.е функции шлюза, а также безопасную работу сотрудников в Интернете, а в случае с **VPN** еще и доступ к сети компании извне.

К тому же, если бизнес компании жестко связан с работой сотрудников во внешней сети, такой сервер (шлюз) должен быть не только отказоустойчивым, но и достаточно производительным: работа специального софта (антивирусных программ, анализ и учет трафика, анализаторы атак и тд и тп) может требовать большого количества системных ресурсов и высокоскоростных мощных железяк.

Сервер электронной почты

Здесь ничего сложного - это сервер, крутящий **E-mail**, т.е принимающий их из локальной сети компании и отправляющий наружу, равно как и наоборот, принимающий к себе и рассылающий локально внешнюю корреспонденцию. Зачастую занимается фильтрацией спама, обеспечением антивирусной безопасности почты и прочими плюшками.

Плюс ко всему позволяет управлять не только внешней, но и внутренней перепиской и документооборотом, а также хранить всё это дело локально тысячелетиями без ограничения (условно) лимитов места (при наличии масштабируемости сервера).

Веб-сервер или сервер приложений

Необходимость иметь свой сайт уже не столько необходимость, сколько стандарт для многих компаний, не говоря уже о том, что на этом самом сайте может быть завязан весь бизнес вообще.

Многие современные компании и организации имеют свой виртуальный офис (такое себе приложение, где ставят задачи, следят за их исполнением, хранят документы, внутреннюю переписку, систему тикетов и тп), простой сайт визитку, интернет-магазин и что-либо еще, будь то портал, тикет-система, форум и тд и тп.

Вообще говоря, бизнес-процесс практически всех фирм в современном мире полностью зависим от работы в интернете, а, следовательно, от веб-сервера.

Его (сервера) доступность и отказоустойчивость, безопасность и защищенность, производительность и прочее, являются ключевыми факторами для обеспечения необходимого функционала (примеры приведены выше). С ростом и развитием систем виртуализации, а также постройкой новых ДЦ (Дата Центров), зачастую веб-сервер не держат локально, а арендуют в виде **VPS** (виртуальный выделенный сервер) у какой-либо хостинг-компании, что дешевле и частенько наиболее приемлемо с точки зрения других нюансов.

Терминальный сервер

Работу удаленных офисов, мобильных пользователей и сотрудников, часто работающих из дома или в командировке, и организацию привычного доступа к рабочим ресурсам посредством сети Интернет или выделенных каналов связи обеспечивает терминальный сервер, он же сервер удаленного доступа.

Как правило он отвечает за создание, обработку, разграничение прав и другие функции по работе с удаленными рабочими столами. Часто работает в связке с **VPN** и/или прокси сервером.

В частности, он же занимается шифрованием передаваемых данных и обеспечивает безопасность такого вида связи.

Как работает? Да очень просто - пользователь соединяется через канал связи с сервером, вводит свои учетные данные и попадает на свой виртуальный рабочий стол или в хранилище своих файлов/документов, что часто одно и то же. Это удобно тем, что важные данные хранятся (и обсчитываются) непосредственно на сервере, и доступ к ним можно получить из любой точки мира, был бы там лишь доступ в интернет. Т.е. формально можно работать даже с ресурсоёмкими приложениями, файлами и данными, с маломощных и мобильных машин.

Сервер баз данных

Следующая роль, которая следует из названия, это обработка данных, которые организованы и структурированы, согласно определенным правилам и хранятся совместно.

Наиболее часто используемыми средствами управления данными являются Oracle, **MySQL** и иже с ними. В случае наличия у бизнес-процессов компании потребности в подготовке и обработке данных, необходим выделенный вычислительный ресурс, т.е. сервер баз данных.

Этот узел обработки данных играет важную роль в жизни компании и определяет необходимый уровень доступности данных, т.е. отказоустойчивости и надежности всей системы. Как правило параметры такого узла напрямую зависят от масштаба базы данных, количества пользователей, динамики и характера обращений к данным.

Как и веб-сервер, часто арендуется у стороннего хостера, либо хранится локально. Это зависит от целей, задач и других нюансов.

Сервера приложений

Для сервера приложений характерны расширенные возможности обработки информации, а взаимодействие с клиентом становится подобным работе приложения.

В маркетинге термином «сервер приложений» обычно обозначают предлагаемое продавцами комплексное решение, которое содержит все требуемые для работы компоненты технологий. Для некоторых организаций такой комплексный подход к построению сервера приложений облегчает разработку благодаря унификации разрабатываемых моделей и централизации поддержки.

Достаточно специфичная вещь в плане понимания, но часто удобная в плане использования.

Сервера-брандмауеры (фаерволлы)

Сервера можно сконфигурировать так, что они будут принимать или отвергать определенные типы сетевых запросов, поступающие как из локальной сети, так и из интернета.

В такой конфигурации сервер (зачастую прокси/vpn/шлюз-сервер, упомянутый выше) становится межсетевым экраном — брандмауэром (фаерволлом), т.е. представляет собой средство обеспечения безопасности.

Принципы работы фаерволлов Вам прекрасно известны на примере разного рода программ такого рода. Сервер-фаерволл, т.е, условно говоря, аппаратное решение, отличается от них качеством и скоростью работы, за счет имеющихся в расположении мощностей.

Впрочем, частенько эту роль совмещают с другими.

Сервер-DHCP

Занимается раздачей **IP**-адресов машинам в сети или подсети. Часто может быть развернут на маршрутизаторе (зависит от необходимых мощностей), т.е. роутере.

Основное преимущество сервера **DHCP** — свобода изменения конфигурации локальной сети при её расширении, добавлении или удалении машин (например, портативных ПК), т.к. всё происходит автоматически и адрес выдается среди свободных в заданном диапазоне.

Часто встречается в крупных сетях, карта **IP**-адресов которых может динамически изменяться, что не очень удобно с точки зрения удаленного управления и идентификации конкретной машины в сети. Решается настройкой **DHCP**-сервера с привязкой по **MAC** или программами-сканерами и другими простыми вариантами.

Сервера FTP

Подобные сервера, работающие на основе протокола **File Transfer Protocol**, уже много десятилетий назад стали стандартом при перемещении файлов в Интернете. Занимаются тем, что хранят файлы и предоставляют возможность управления (закачки, перемещения, удаления и тп) ими через **FTP**-клиенты по данному протоколу.

Сложные **FTP**-сервера обеспечивают администратору большие возможности управления в том, что касается прав на подключение и совместного использования файлов, типов разделяемых файлов и их размещения.

Зачастую представляют куда большую скорость и простоту при передаче и манипулировании большими объемами файлов в интернете (т.е простая **HTTP**-передача может идти на порядок дольше, чем загрузка с **FTP**).

Принт-сервера

Позволяют всем подключенным к сети компьютерам распечатывать документы на одном или нескольких общих принтерах. В этом случае отпадает необходимость комплектовать каждый компьютер собственным... эээ... печатающим устройством.

Кроме того, принимая на себя все заботы о выводе документов на печать, принт-сервер освобождает компьютеры для другой работы. Например, такой сервер хранит посланные на печать документы на своем жестком диске, выстраивает их в очередь и выводит на принтер в порядке очередности.

В общем, удобно и не требует каких-то особых ресурсов. Последнее время часто встречается на хороших маршрутизаторах.

Прочие серверные роли

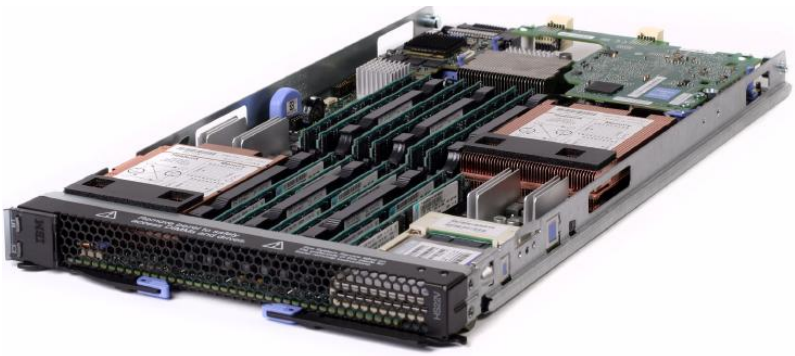
Можно выделить еще множество мелких (и не очень) серверных ролей, но я и так много их перечислил. Часто сервер совмещает в себе большую часть ролей сразу, т.е контроллер домена обычно может выступать сразу файл-сервером, веб-сервером, сервером-брандмауэром, принт-сервером и черт знает, чем еще.

Такая практика зачастую применяется в средних и малых компаниях, но... Многое зависит от системного администратора (чуете, куда я клоню? ;))

Аппаратная специализация и нюансы

Размеры и другие детали внешнего исполнения

Вообще, визуально, настоящий (так сказать, промышленный) сервер несколько отличается от привычного нам компьютера и выглядит примерно следующим образом:



Конструктивно аппаратные серверы могут исполняться в настольном, напольном и стоечном вариантах. Последний вариант обеспечивает наибольшую плотность размещения вычислительных мощностей на единицу площади, а также максимальную масштабируемость. В настоящее время используются так называемые blade-сервера (см.скриншот выше), т.к они компактны и позволяют эффективно использовать пространство:

Сервера, не требующие высокой производительности и большого количества внешних устройств, зачастую уменьшают в размерах. Часто это уменьшение сопровождается уменьшением ресурсов.

Что касается “промышленного решения”, то по-хорошему серверы размещаются в специально оборудованных помещениях, называемых нет, не серверной (это в компаниях), а ДЦ (Дата Центром).

Младшие модели серверов могут размещаться в обычных офисных помещениях (и если повезет, обитают как раз в серверной, т.е. комнате, соответствующей определенным стандартам и особенностям), и от простых десктопных компьютеров их часто отличает лишь автономная работа, мощности, возложенный функционал и подключение к блоку бесперебойного питания повышенной ёмкости.

Управление серверами осуществляют квалифицированные специалисты… Ну или не квалифицированные. В общем, системные администраторы, о которых мы с Вами говорили буквально в прошлой главе.

Альтернативные решения

Крайней степенью реализации серверов являются так называемые альтернативные решения, выполненные на подручном оборудовании (роутеры, сетевые дисковые массивы-хранилища, аппаратные терминалы, ноутбуки и прочее, то есть то, о чем я говорил в самом начале, упоминая, что сервером может быть хоть телефон, хоть рабочий компьютер. Суть в том, что сервер это по сути функционал и лишь потом железо).

Аппаратное обеспечение таких решений строится «с нуля» или перерабатывается из существующей платформы без учёта совместимости, что делает невозможным использование устройства с привычным программным обеспечением.

Программное обеспечение в таких решениях загружается в постоянную и/или энергонезависимую память производителем, примером могут служить всякие дорогостоящие роутеры-медиасервера со встроенными жесткими дисками и оболочкой (прошивкой) с поддержкой **VPN, Samba, SSH** и тд и тп.

Такие решения обычно менее гибки и универсальны, зато зачастую дешевле серверов в промышленном варианте. Да, если поискать можно найти и дороже, но кому оно нужно?

Ресурсы серверов

По ресурсам (частота и количество процессоров, количество памяти, количество и производительность жёстких дисков, производительность сетевых адаптеров) сервера специализируются в двух противоположных направлениях — наращивании ресурсов и, как бы ни казалось странно, в их уменьшении.

Наращивание ресурсов преследует целью увеличение ёмкости (например, специализация для файл-сервера) и производительности сервера. Когда же производительность достигает некоторого предела, дальнейшее наращивание продолжают другими методами, например, распараллеливанием задачи между несколькими серверами или даже виртуализацией.

Урезание ресурсов преследует цели уменьшения размеров и энергопотребления серверов, тепловыделения, шумоподавления и других свойств.

Послесловие ко второй главе

Пожалуй, этого должно хватить для общего базиса и понимания того, что такое сервер, чем он может заниматься, какими характеристиками должен отличаться и как выглядеть.

Чуть больше деталей, скорее всего, Вы увидите в следующих частях книги, ну или найдете сами, если поняли посыл первой главы.

Глава 3. Сервера и оборудование

Предисловие к третьей главе

Вопреки ожиданию многих, эта глава будет крайне короткой. Качества сервера, вроде отказоустойчивости, надежности и прочих, мы кратко рассмотрели в прошлых главах и более подробно описывать их, в рамках первой части этой книги, нет никакого смысла.

Что же касается непосредственного выбора серверных железяк, то в общем и целом в этой теме нет ничего сложного, и потому я ограничусь общими рекомендациями, перейдя к теме операционных систем. Думаю, прочитав Вы поймете почему так.

Как выбрать серверное оборудование

Какие требования предъявляются к аппаратной части сервера? Как, я надеюсь, Вы поняли из упомянутой уже главы, главное – это бесперебойная круглосуточная работа, без сбоев, зависаний и поломок, т.е. стабильно, т.е. отказоустойчиво.

При этом желательно, чтобы она была достаточно быстрой (ибо сама суть сервера в обсчете возложенного на него функционала за счет своих мощностей, заточенных специально под это), а значит,

как я говорил, еще одна основа любой серверной информационной системы – это масштабируемость и гарантированное сохранение данных.

Общие правила выбора

Как выбрать подходящее серверное оборудование? Ни в коем случае не ориентируясь в первую очередь на стоимость.

Чтобы правильно выбрать серверное оборудование, как уже говорилось, сначала стоит понять, какие процессы ему предстоит выполнять с учетом направления деятельности компании, т.е требуется изучить предстоящие бизнес-задачи, причем с учетом роста предприятия в будущем. В частности, важно учесть структуру информационной системы, спрогнозировать ее требования, постараться максимально точно определить нагрузку, что в общем-то не так сложно, как кажется.

В комплектации сервера, что логично, не должно быть ничего лишнего, но и экономить сильно тоже не стоит, так как всё это обернется в конечном итоге только очередными затратами (опять же вспоминаем про отказоустойчивость и масштабируемость) причем не только денег, но и нервов (т.к, как правило, руководство крайне неохотно выделяет средства на новые железяки, а вопрос расширения, так или иначе, обязательно встанет перед Вами со временем).

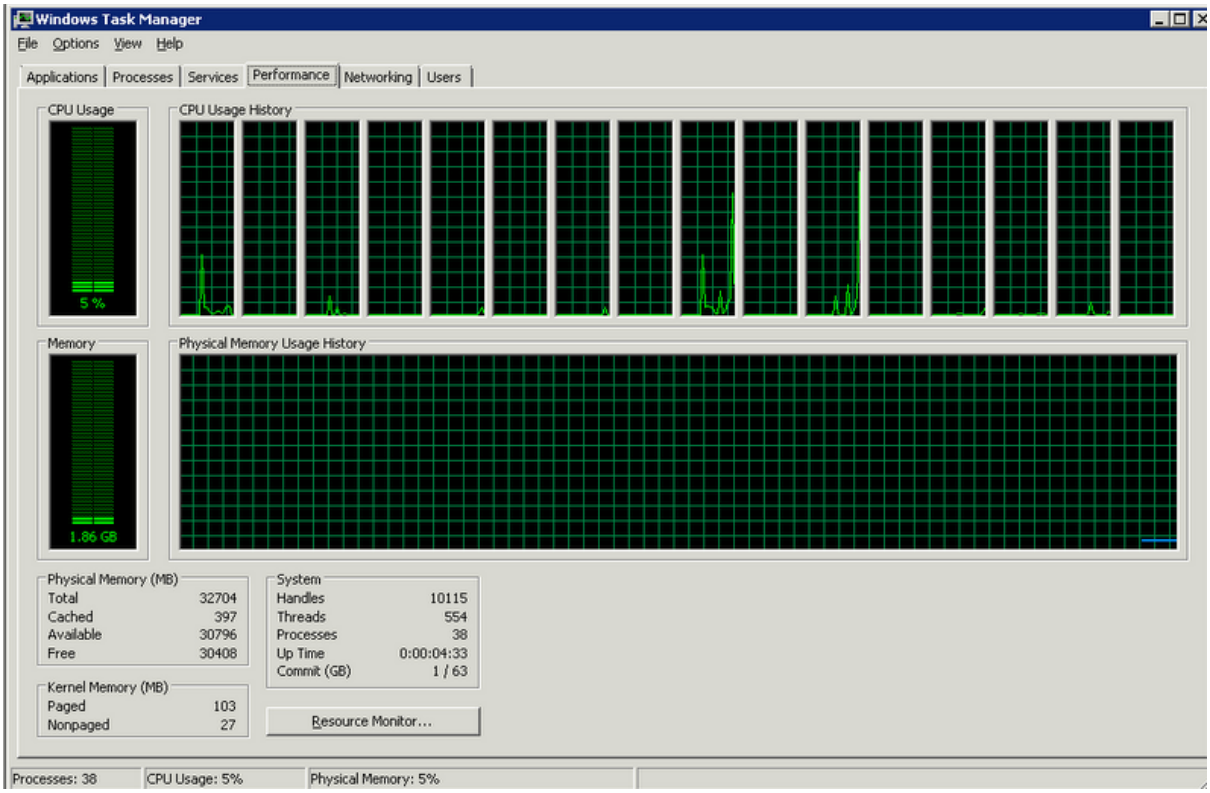
Глобально же, в выборе серверного оборудования (если конечно Вы не покупаете готовый сервер в сборки) стоит опираться на уже привычный состав характеристик, за исключением того, что требуются компоненты большей мощности, со специальными технологиями (вроде **ECC**, который был кратко описан в прошлой главе) и высокой надежности, т.е. непосредственно из серверного сегмента.

В прошлой книге, да и на сайте **sonikelf.ru**, я неоднократно описывал как, почему и исходя из каких характеристик стоит делать выбор тех или иных комплектующих, поэтому в данном случае я просто дам небольшие комментарии в ракурсе серверов, остальное же, думаю, Вы прекрасно сообразите сами.

Еще раз для тех, кто не понял - написанное ниже, это некое обобщение, конкретный выбор делается исходя из целей и задач, т.е. ролей, выполняемых сервером и размером компании в физических единицах рабочего **IT**-оборудования.

Процессор сервера

Как правило, в нормальном сервере используется минимум два процессора (не ядра, а именно процессора), и/или большое количество ядер.



Чаще всего выбор падает на **Intel Xeon**'ы. Меня часто спрашивают почему именно они, с чего они такие дорогие и с фига ли в сервер не ставят какие-нибудь простенькие дешевенькие **AMD**. Попробую кратко объяснить.

В первую очередь - поддержка упомянутого неоднократно **ECC**. Дело в том, что высокая скорость и большой объем оперативной памяти в сервере, таки повышают вероятность возникновения программных ошибок. Обычно такие ошибки оказываются незаметными, но несмотря на это, они могут привести к изменению данных или, что еще хуже, падению системы. И как я и говорил, если для настольных компьютеров подобные ошибки не страшны из-за их редкого возникновения, то в серверах, работающих круглые сутки по несколько лет, они недопустимы в принципе. Так

вот, поддержка **ECC** должна быть реализована не только на уровне памяти, но и процессора, а вот процессоров, которых поддерживают эту самую **ECC**, достаточно мало.

Далее идет кэш-память – она представляет собой такую прослойку между процессором и оперативной памятью. Обладает высокой скоростью, а потому процессор всегда смотрит в кеш на предмет данных для обсчета (из оперативки их забирать дольше). Логично, что чем больше объем кэша, тем быстрее работает процессор, так как даже очень быстрая оперативная память работает значительно медленнее кэш-памяти. Обычно процессоры **Xeon** имеют б**о**льший объем кэша и уровней кеша, чем какие-либо другие, но дело еще и в том, что кеш даёт выигрыш далеко не во всех ситуациях (например, игры и расчет числа ПИ - немного разные процессы), однако в серверах, зачастую, важен именно он.

Тепловая мощность (**TDP**) и рассеиваемое тепло. В общем и целом, определяет скорость энергопотребления и тепловыделения при максимальной (и не только) нагрузке. **Xeon**'ы, как правило, выделяют меньше тепла, кушают меньше электроэнергии, что экономит деньги, а также позволяет более эффективно охлаждать процессор, что немаловажно для серверов.

Частота процессора, архитектурность и "ядерность". Здесь можно было бы сказать что-то умное по поводу высоких частот, но по факту частота здесь является не самым главным фактором, а вот

количество ядер (например, штуки по **24** на процессор) и соответствующая архитектура, умеющая эффективно работать в многопоточном режиме, у **Xeon**'ов частенько на самом высоком уровне.

Поддержка технологий, направленных на увеличение скорости работы и/или безопасности работы, вроде **Intel TSX-NI** или **Trusted Execution**.

Первая, т.е **Intel Transactional Synchronization Extensions New Instructions** подразумевает под собой надстройку над системой работы с кэшем процессора, оптимизирующую среду исполнения многопоточных приложений, но конечно только в том случае, если эти приложения используют программные интерфейсы **TSX-NI**. Наборы инструкций **TSX-NI** позволяют более эффективно реализовать работу в случаях, когда множество потоков обращаются к одним и тем же данным и возникают ситуации блокировки потоков.

Спекулятивный доступ к данным, который реализован в **TSX**, позволяет эффективнее строить такие приложения и более динамично масштабировать производительность, при увеличении числа параллельно исполняемых потоков за счет разрешения конфликтов при доступе к общим данным.

Вторая, т.е **Intel Trusted Execution** расширяет возможности безопасного выполнения команд посредством аппаратного расширения возможностей процессоров и наборов микросхем **Intel**, например за счет функции защиты, вроде создания среды, где одни приложения выполняются изолированно от других приложений системы.

И, само собой, немаловажна (или даже очень важна) поддержка огромного количества оперативной памяти, чего не могут показать процессоры других линеек. Те же 768 Гб оперативной памяти на процессор, вполне себе реальная цифра, имеющая вполне себе разумное обоснование под собой.

Дисковый массив сервера

Одним из ключевых элементов сервера, само собой, является жесткий/твердотельный диск, ибо именно он отвечает за скорость, стабильность, качество и надежность работы сервера. Диски рекомендуются **Western Digital**, но окончательный выбор всегда остается за Вами. Предварительно следует шерстить интернет на предмет брака конкретной партии.

Но лично я (да и вообще, наверное, все на свете), не представляю себе сервера без **RAID**, т.е с одиночными дисками в принципе, ибо одной из ключевых обязанностей, возложенных на любой сервер, является сохранение всех данных любой ценой и именно на диски, точнее на массив, ложится эта задача.

RAID — это комплекс или, если хотите, связка из нескольких устройств - жестких/твердотельных дисков. Как я и говорил выше, этот массив служит для повышения надёжности хранения данных и/или для повышения скорости чтения/записи информации (или и то и другое).

Собственно, то, чем именно занимается оная связка из дисков, т.е ускорением работы или повышением безопасности данных, - зависит от Вас, а точнее, от выбора текущей конфигурации рейда(ов). Разные типы этих конфигураций как раз и отмечаются разными номерами**: 1, 2, 3, 4** и соответственно выполняют разные функции.

Рейды ощутимо удобнее и эффективнее использования одного диска в системе. Я бы даже рекомендовал их всем поголовно использовать и на клиентских машинах, но так как это повышает конечную стоимость инфраструктуры, да и в общем и целом не имеет смысла (во всяком случае я надеюсь на это, ибо в любой компании должен быть файл-сервер, который будет отвечать за целостность, централизованность и сохранность информации). Для серверов же установка **RAID**'а - это не рекомендация, а стандарт.

Физически **RAID**-массив представляет собой от двух до n-го количества жестких дисков, подключенных к мат.плате, поддерживающей возможность создания **RAID.** Или к соответствующему контроллеру, что реже, ибо они весьма дороги для рядового пользователя (контроллеры обычно используются на серверах в силу повышенной надежности и производительности). Т.е. на глаз ничего внутри системника не изменяется, никаких лишних подключений или соединений дисков между собой или с чем-то еще попросту нет.

В общем, в аппаратной части всё почти как всегда, а изменяется лишь “программный” (хотя это не совсем корректное слово) подход на уровне мат.платы или контроллера, который, собственно, и задается, путем выбора типа рейда, т.е. как именно должны работать подключенные диски.

В системе же, после создания рейда, тоже не появляется никаких особенных причуд. По сути вся разница в работе с рейдом заключается только в небольшой настройке в биосе, которая собственно организует рейд (см.ниже) и в использовании драйвера. В остальном ВСЁ совершенно то же самое – в "Мой компьютер" те же **C, D** и прочие диски, всё те же папки, файлы... В общем, на глаз, полная идентичность.

В простейшем эконом варианте, на маленький офис, ставится рейд под номером **1**, он же “зеркало”. Диски в таком рейде полностью дублируют друг друга, и он имеет высокую надежность, т.е всё работает (и все данные существуют “всегда”, а не исчезают с выходом из строя одного из устройств) до тех пор, пока функционирует хотя бы один диск, т.е. если даже если, утрируя, в ходе работы стукнуть молотком по одному из дисков и вывести его из строя, скорее всего, Вы не потеряете ни байта информации, т.к. второй (третий-четвертый-пятый-100-ый) диск является чистой копией первого и заменяет его при выходе из строя.

Дисков в таком рейде может быть сколько угодно, и чем их больше, тем, в теории, надежнее массив. Минус тут в скорости - зеркалирование отнимает ресурс чтения/записи, но надежность превыше всего. А также “потеря места”, т.е поставив 4 диска по одному терабайту в первый рейд, Вы будете видеть только один терабайт в системе, все остальные будут молча дублировать информацию и нигде, кроме биоса, толком не светиться, до, что бывает, выхода какого-либо диска из строя.

В **RAID 0** (он же страйп («**Striping**»)) — используется также несколько жестких дисков, которые совместно обрабатывают информацию, что повышает производительность. Чтобы было понятно - таскать мешки одному человеку дольше и сложнее чем вчетвером (хотя мешки остаются все теми же по своим физическим свойствам, меняются лишь мощности с ними взаимодействующие). Программно же информация на рейде такого типа разбивается на блоки данных и записывается на оба/несколько дисков поочередно.

Один блок данных на один диск, другой блок данных на другой и тд. Таким образом существенно повышается производительность (от количества дисков в массиве зависит кратность увеличения производительности, т.е 4-е диска будут бегать шустрее, чем два и тд), но ОЧЕНЬ страдает безопасность данных на всём массиве, что неприемлемо для серверов. При выходе из строя любого из входящих в такой **RAID** винчестеров (т.е. жестких дисков) практически полностью и безвозвратно пропадает вся информация,

поэтому нулевки почти не используются разумными администраторами.

RAID 2 зарезервирован для массивов, которые применяют некий код Хемминга. Принцип работы примерно такой: данные записываются на соответствующие устройства так же, как и в **RAID 0**, т.е они разбиваются на небольшие блоки по всем дискам, которые участвуют в хранении информации. Оставшиеся же (специально выделенные под это) диски хранят коды коррекции ошибок, по которым, в случае выхода какого-либо винчестера из строя, возможно восстановление информации. То бишь в массивах такого типа диски делятся на две группы — для данных и для кодов коррекции ошибок. Я в серверах их давно не встречал, да и в общем-то оно не очень нужно.

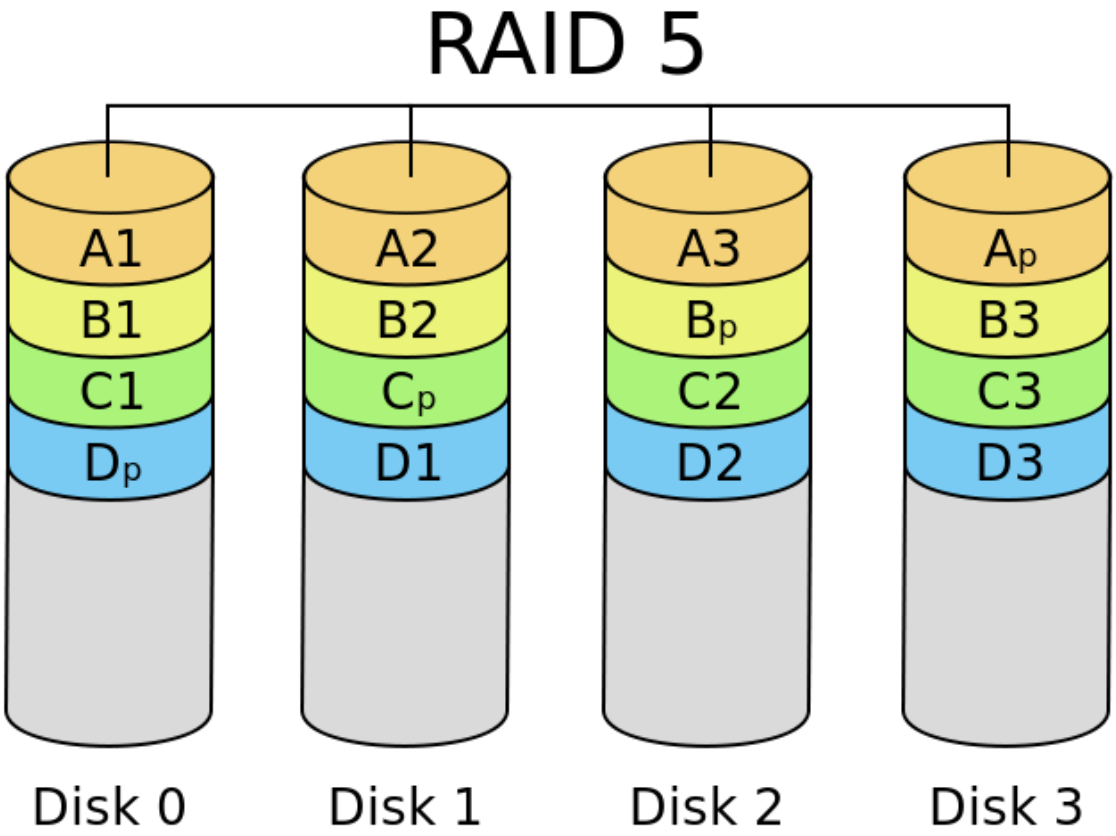
Основным недостатком уровней **RAID** от 2-го до 4-го является невозможность производить параллельные операции записи, так как для хранения информации о чётности используется отдельный контрольный диск. **RAID 5** не имеет этого недостатка.

> Блоки данных и контрольные суммы циклически записываются на все диски массива, нет асимметричности конфигурации дисков. Под контрольными суммами подразумевается результат операции XOR (исключающее или). Xor обладает особенностью, которая даёт возможность заменить любой операнд результатом, и, применив алгоритм xor, получить в результате недостающий операнд.

Например: **a xor b = c** (где **a, b, c** — три диска рейд-массива), в случае если a откажет, мы можем получить его, поставив на его место c и проведя xor между **c** и **b**: **c xor b = a**. Это применимо вне зависимости от количества операндов**: a xor b xor c xor d = e**.

Если отказывает c, тогда e встаёт на его место и, проведя xor, в результате получаем **c: a xor b xor e xor d = c**. Этот метод по сути обеспечивает отказоустойчивость 5 версии. Для хранения результата xor требуется всего **1** диск, размер которого равен размеру любого другого диска в **RAID**.

Минимальное количество используемых дисков равно трём. © Википедия



Скажу лишь, что наиболее оптимальными являются комбинированные варианты рейдов, но они и самые дорогие. В вариантах подороже и компаниях посерьезнее, могут использоваться варианты **0+1** или **1+0**, т.е пара рейдов в связке, что позволяет добится большей скорости при приемлемой надежности. **0+1** = два **RAID 0** объединяются в **RAID 1**; **1+0** = зеркалированный массив, данные в котором записываются последовательно на несколько дисков, как в **RAID 0**. Эта архитектура представляет собой массив типа **RAID 0**, сегментами которого вместо отдельных дисков являются массивы **RAID 1**. Соответственно, массив этого уровня должен содержать как минимум **4** диска (и всегда чётное количество). **RAID 10** объединяет в себе высокую отказоустойчивость и производительность.

Выбирать массив, как и всё остальное, стоит под цели и задачи, но в первую очередь стоит исходить из его надежности. Про прочие рейды можете прочитать в Википедии, откуда и взята цитата про пятый выше и картинка. Автор рекомендует комбинированные рейды или хотя бы бюджетные однерочки.

Оперативная память сервера

Всё, что требуется знать об оперативке сервера, помимо общеизвестных фактов про оперативную память вообще (см.статьи или прошлую книгу) - это то, что её должно быть много (реально

много) и она, как и процессор, должна поддерживать **ECC**, а также быть известного производителя и крайне надежной (как и в случае, с дисками, стоит выгугливать конкретное наименование на предмет брака и отзывов).

Лучше брать памяти с большим запасом, чем потом “задохнуться” и загрузить диски подкачкой. Поверьте, через месяц-два после ввода сервера в эксплуатацию, попытавшись сэкономить здесь, Вы будете крайне неприятно удивлены. Не мерьте серверную память критериями обычных компьютеров, даже игровых, особенно её объем.

Всё остальное, что нужно

Далее, думаю, что всё очевидно.

Мат.плата должна быть хорошей, поддерживать несколько процессоров (в лучшем варианте) или много памяти (в дешевом). Подробнее про мат.платы, опять же, было в книге или на сайте **sonikelf.ru**. В идеале должна быть поддержка горячей замены дисков и прочих серверных плюшек.

Блок питания, ни в коем случае, никакой не **noname**, а известного производителя. Мощности надо, как и в случае с памятью, с большим запасом и тд и тп. По секрету скажу, что БП вылетают чаще всего, в моей практике были периоды, когда сыпались они чаще чем диски, я не успевал их подвозить на площадку в ДЦ.

Если размещаете сервер в офисе, а так обычно и бывает, то потребуется хороший бесперебойник, способный выдержать хотя бы **20** минут работы (хотя, конечно, зависит от мощности сервера и нагрузки на него в момент “падения” света).

Форм-фактор корпуса и тп.. Я бы рекомендовал брать промышленные, ибо они меньше по объёму, их проще охлаждать, когда будут деньги можно прикупить действительно хороший серверный шкафчик или поставить его в стойку в ДЦ (арендовать место в стойке дешевле, чем физический сервер целиком).

Что там еще? Видеокарты обычно не нужны (ибо реализованы на уровне мат.платы или процессора с видеоядром), да и вообще сервер управляется удаленно с любой рабочей машины. У меня, как правило, они не имеют даже монитора и мышки с клавиатурой (либо в кладовке, на предмет внештатников).

В остальном... Всё как обычно. А да, по-хорошему, нужен нормальный **RAID**-контроллер, а не встроенный в мат.плату. Под хорошим я имею в виду с большим числом слотов, поддержкой нужного Вам вида рейда, в идеале с резервным питанием и большим количеством памяти для кеша на борту.

Выбор вендора (производителя) оборудования

Лично я рекомендую всем покупать оборудование среднего ценового диапазона. Топовые модели обычно обладают

функционалом, который не будет востребован во время эксплуатации. А самые дешевые бывает работают не так стабильно, как хотелось бы.

Этот принцип можно распространить и на выбор производителя (все эти **HP**, **Dell** и тп). Как правило, информацию о ранжировании производителей получить достаточно легко - погуглить. И после я рекомендую фирмы из середины списка.

Наиболее известные производители часто завышают стоимость оборудования, пользуясь известностью своей марки. Так работает рынок во всём и сервера тому не исключение.

Но здесь не следует поддаваться на рекламные обещания: крупные компании выделяют на маркетинговые цели весьма существенный процент от стоимости оборудования. Например, один из производителей коммутационного оборудования только на посреднические цели (партнерам) выделяет от **30** до **40%** от стоимости проданного оборудования. Естественно, что партнеры всячески будут способствовать продвижению именно такой линейки и убеждать в ее исключительности, да еще и завышать цены самостоятельно.

Имеет смысл комплектовать однотипное оборудование моделями одного производителя. Как правило, таковые поставляют совместно с оборудованием некоторые дополнительные опции, которые

позволяют упростить администрирование. Например, это может быть программное обеспечение централизованного администрирования серверов или проприетарные протоколы коммутационного оборудования.

Сервисные контракты

Чем дороже оборудование, тем, как правило, больше задач оно решает в информационной системе. И тем к большим потерям приведет его простой на время ремонта или обслуживания. Поэтому целесообразно заключать такую штуку, как сервисные контракты, которые гарантируют восстановление оборудования в течении оговоренного срока.

Возможность заключения сервисного контракта с заданным уровнем обслуживания (временем поставки вышедшего из строя компонента) следует учитывать при выборе оборудования. Особенно если предприятие расположено вдалеке от региональных центров.

Перед принятием решения обязательно уточните наличие региональных складов, время доставки на предприятие детали с такого склада, наличие в регионе сертифицированных вендором специалистов, которым разрешено проводить обслуживание и ремонт предполагаемого к покупке оборудования.

Запасные компоненты на любой случай

ОБЯЗАТЕЛЬНО постарайтесь приобрести запасные части к приобретаемому оборудованию. Обычно сервисные контракты после **3—4** лет эксплуатации становятся очень дорогими, а приобрести детали становится невозможным, поскольку они перестают выпускаться в связи с переходом на новые модели.

Например, для серверов необходимо обязательно приобрести запасные жесткие диски и блоки питания. Эти детали чаще всего отказывают во время эксплуатации. Причем эту мысль хорошо бы **заранее** донести до руководства максимально развернутым образом, объяснив риски и вышеупомянутый нюанс про стоимость.

На этом, пожалуй, хватит. Давайте к послесловию по главе.

Послесловие к третьей главе

Еще раз, подводя итог: выбирать надо исходя из целей и задач, возлагаемых на сервер, и делать это с запасом мощностей или с учетом возможности дальнейшей масштабируемости. Исходить надо из надежности и производительности, а не цены.

Если ни черта не понимаете и/или не хотите собирать сервер сами и прочее, то лучше купить готовый, но только у известного производителя и у нормальной компании, занимающейся именно

серверами (лучше у авторизованного продавца), а не у доморощенных сборщиков-чудотворцев.

Про цели и задачи, мол как понять, сколько потребуется мощностей под какие роли: почитайте системные требования, посмотрите готовые конфигурации, включите голову. Логично, что под файл сервер нужно много дисков и хороший массив, под сервер баз данных куча оперативной памяти и тд и тп.

Остальное написано выше, а больше подробностей в следующих частях книги.

Двигаемся дальше.

Глава 4. Windows Server. Базис

Предисловие к четвертой главе

Как я говорил в самом начале, выбор ОС зависит от Ваших умений, желаний и целей, а посему любая из них обычно хорошо подойдет для решения поставленных задач.

Так что эта глава не является призывом к повсеместной установке **Windows** или пропагандой продуктов **Microsoft**, но тем не менее её нельзя не написать, ибо надо признать, что именно **Windows Server** используется на большинстве, так сказать, офисных серверов. В отличии от **ДЦ**, специализирующихся на предоставлении хостинг-услуг, где, как правило, используется **Linux**.

Некоторые куски из этой (и последующих) глав книги, Вы можете встретить в интернете, а также разного рода книгах, блогах и заметках, других авторов, в частности, на ряде моих сайтов под другим именем. Это нормально по нескольким причинам, начиная от утечек составляющих уроков обучения **sonikelf.ru** (часть из которых использовано для написания этой книги) и заканчивая тем, что некоторые вещи просто нельзя написать как-то по-новому и приходится использовать их, что называется, “как есть”.

Приступим.

Windows Server - какой, почему и как

Windows Server 2012 - общее описание и прочее

Когда я начинал эту книгу, долго думал о какой из систем писать. Как админу старой закалки, мне всегда нравилась **Windows Server 2003**, но надо признать, что, как и в своё время пользовательская **Windows XP**, эта система устарела.

С приходом **Windows Server 2008 R2** ситуация поменялась, но прямо скажем **2008** мне не очень понравилась, тому много причин, но... Пожалуй это не первоцель и тема сей главы, а посему я просто скажу, что в рамках этой книги мы будем говорить про **Windows Server 2012,** хотя в общем-то на дворе уже **2016** и можно было бы писать про старшую версию... Но по некоторым причинам (частично описанным ниже) , я всё же хочу поговорить про **2012**.

Microsoft выпустила **Windows Server 2012** через три года после выхода на свет **Windows Server 2008 R2**. **Server 2012** ныне считается самым значительным обновлением серверной операционной системы от **Microsoft** с тех самых пор, когда **Windows Server NT 3.51** был обновлен до **NT 4.0** и к **Windows Server** был представлен современный графический интерфейс.

Server 2012 — это настолько же значимое продвижение вперед, так как, скорей всего, это первый вариант **Windows Server**, который создан с учетом не только требований организаций, но и ожиданий и интересов конечных пользователей. **2012**-ый выполнен в расчете на совместимость с тремя ключевыми тенденциями компьютерного мира и на поддержку этих тенденций. Распространение каждой из этих трех тенденций связано в основном с требованиями конечных потребителей. Это виртуализация, облачные вычисления, и тенденция ориентации информационных технологий на потребителя, а конкретнее - консьюмеризация ИТ. Характеризуется она возрастающим интересом к использованию персоналом компаний личных вычислительных устройств, в частности мобильных телефонов, в повседневной рабочей среде.

Microsoft спроектировала **Server 2012** для удовлетворения этих трех требований рынка, добавив в эту операционную систему (ОС) некоторые обновления и усовершенствования. Возможности, касающиеся виртуализации и облачных вычислений, поддерживаются благодаря новым технологиям виртуализации, реализованным в **Hyper-V 3.0**. Появились также возможности подключения центров обработки данных к общедоступным облачным сервисам и возможности, которые позволяют системным администраторам создавать гибридные и мультиарендные частные облачные сервисы. Аппаратное обеспечение серверов, хранилищ данных и компьютерных сетей может быть виртуализовано, таким

образом, уменьшается стоимость электроэнергии, потребляемой оборудованием, централизуется администрирование, появляется возможность быстрого и эффективного масштабирования при расширении инфраструктуры.

Консьюмеризация в **IT** — это уже давно тенденция, которая привлекает к себе особое внимание. Персональные высокотехнологичные устройства становятся все более сложными и распространенными, люди все чаще хотят использовать эти устройства в офисах. Информационные технологии должны позволить соблюсти тонкий баланс между сохранением контроля над бизнес-сетями, к которым подключаются подобные устройства, и предоставлением пользователям полноценной рабочей среды.

Server 2012 предлагает возможности соблюдения этого баланса благодаря улучшениям службы удаленных рабочих столов (**Remote Desktop Services, RDS**) и технологии инфраструктуры виртуальных рабочих столов (**Virtual Desktop Infrastructure, VDI**). **Microsoft** внесла в **VDI** улучшения, которые касаются работы с глобальными сетями, так что теперь надежность работы с приложениями с помощью технологии удаленного рабочего стола сопоставима с возможностями, которые ранее были характерны лишь для подключения к приложениям и сетевым ресурсам в локальных сетях.

Управление службами удаленных рабочих столов и удаленными клиентами теперь централизованно благодаря обновленному средству управления сервером (**Server Manager**).

Server 2012 не только отвечает потребностям меняющихся технологий, применяемых при организации рабочих мест пользователей, но и предлагает новые возможности и улучшения доступных ранее функций. Существует множество новых функций и улучшений, некоторые из них находятся «под капотом» системы и не бросаются в глаза.

Такие дела.

Essentials как одна из редакций Win Server

Как и линейки домашних систем **Microsoft**, **Windows Server** также представлена в нескольких редакциях: **Foundation, Essentials, Standard, Datacenter**, их отличия наглядно показаны на той же Википедии в виде таблицы или на сайте **Microsoft** в виде маркетинговой информации.

В рамках этой книги я буду говорить про **Essentials**, т.к., если обобщить, то редакция **Essentials** имеет большинство ролей **Windows Server**, которые могут быть необходимы и при этом не содержит “лишнего” функционала.

Что же умеет **Essentials** в общем и целом:

1. Авторизация и аутентификация пользователей вашей сети (домен контроллер службы каталогов **Active Directory**);
2. Файловое хранилище (роль файлового сервера);
3. Удаленный доступ к корпоративной сети (**VPN** и **DirectAccess** сервер);
4. Удаленный доступ к файловому хранилищу через Web-интерфейс (настроенный для этого **IIS**);
5. Удаленный доступ к рабочим столам клиентских машин (шлюз удаленных рабочих столов);
6. Резервное копирование клиентских машин (**Windows backup**);
7. Резервное копирование самого сервера (**Windows backup**);
8. Интеграция с облачными технологиями **Microsoft** (**Office 365, Azure backup** и тд.);
9. Консоль единой настройки **Essentials**, которая позволит настроить возможности описанные выше даже не подготовленному системному администратору.

С другой стороны, некоторые из ролей настроены, некоторые доступны в полном объеме, некоторые как например **Hyper-V** с серьезными ограничениями, что является минусом. Компенсацией за эти все ограничения является более низкая цена, а также **25** включенных клиентских лицензий, централизованная и простая настройка и много других ништяков. Такой себе крепкий

середнячок для малых компаний, в которых поначалу, наверное, многим придется работать.

Установка Windows Server 2012 Essentials

Предисловие к режимам установки

Как и ранее, существует два основных варианта установки **Server 2012**: в виде основных серверных служб (**Server Core**) и графического пользовательского интерфейса (**Server GUI**). Установка основных компонентов сервера, **Server Core**, применяется по умолчанию, благодаря этому уменьшается потребление системных ресурсов, которые необходимы для установки графического пользовательского интерфейса, что оптимизирует производительность сервера.

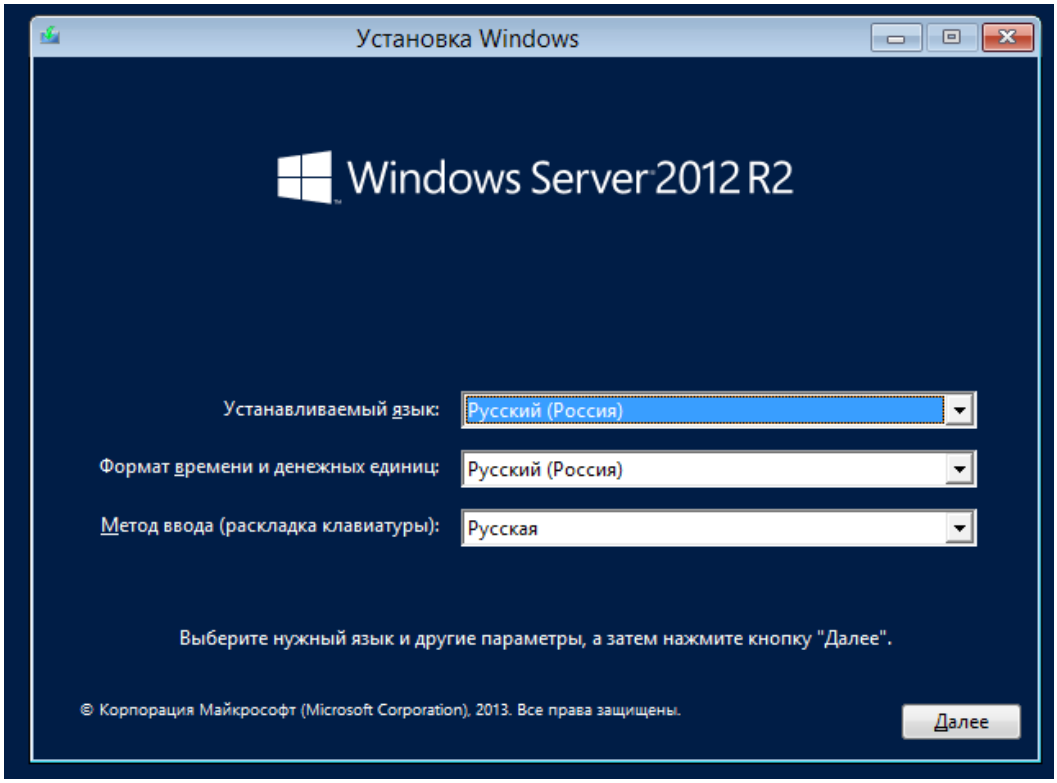
При установке в варианте **Server Core** уменьшаются объем необходимого дискового пространства, требования к обслуживанию, а также уязвимость сервера к атакам. Установка сервера с графическим пользовательским интерфейсом, **Server GUI**, — это другое название полной установки (**Full Installation**) из **Server 2008 R2**.

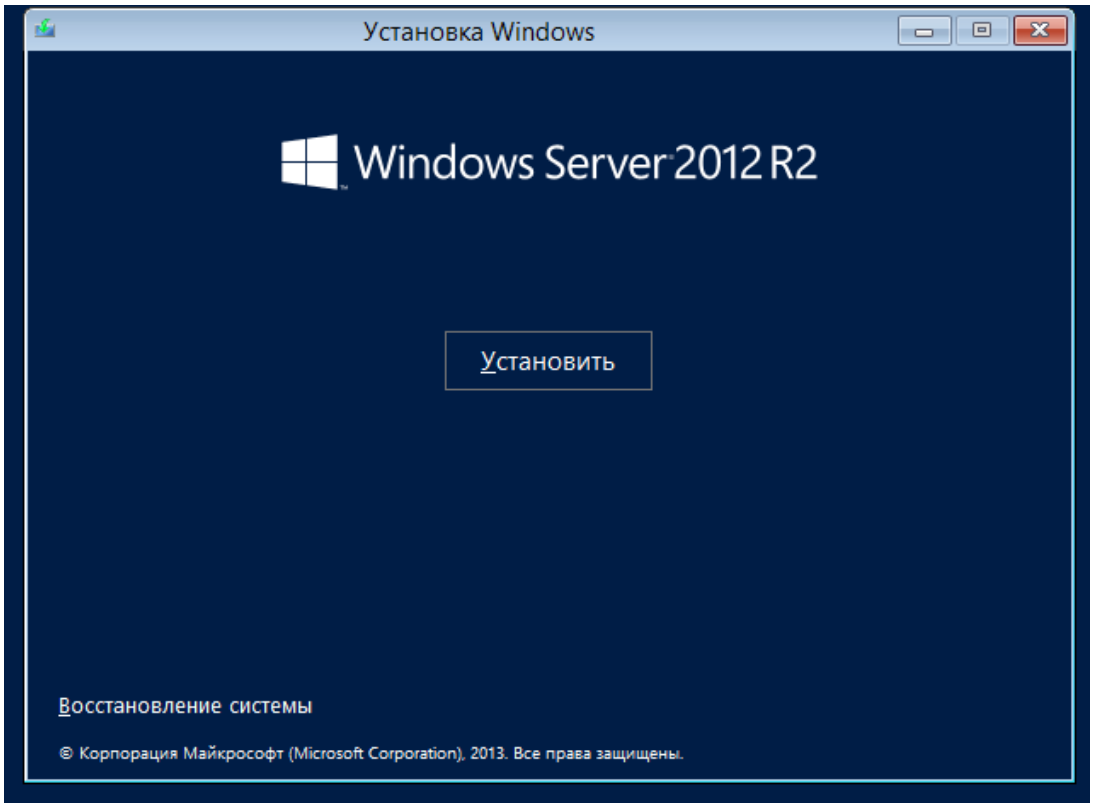
При этом варианте установки выполняется загрузка полного интерфейса **Server 2012**, в том числе интерфейса в стиле **Windows 8 Modern UI**, а также всех средств для управления сервером, имеющих графический интерфейс. Появился и новый

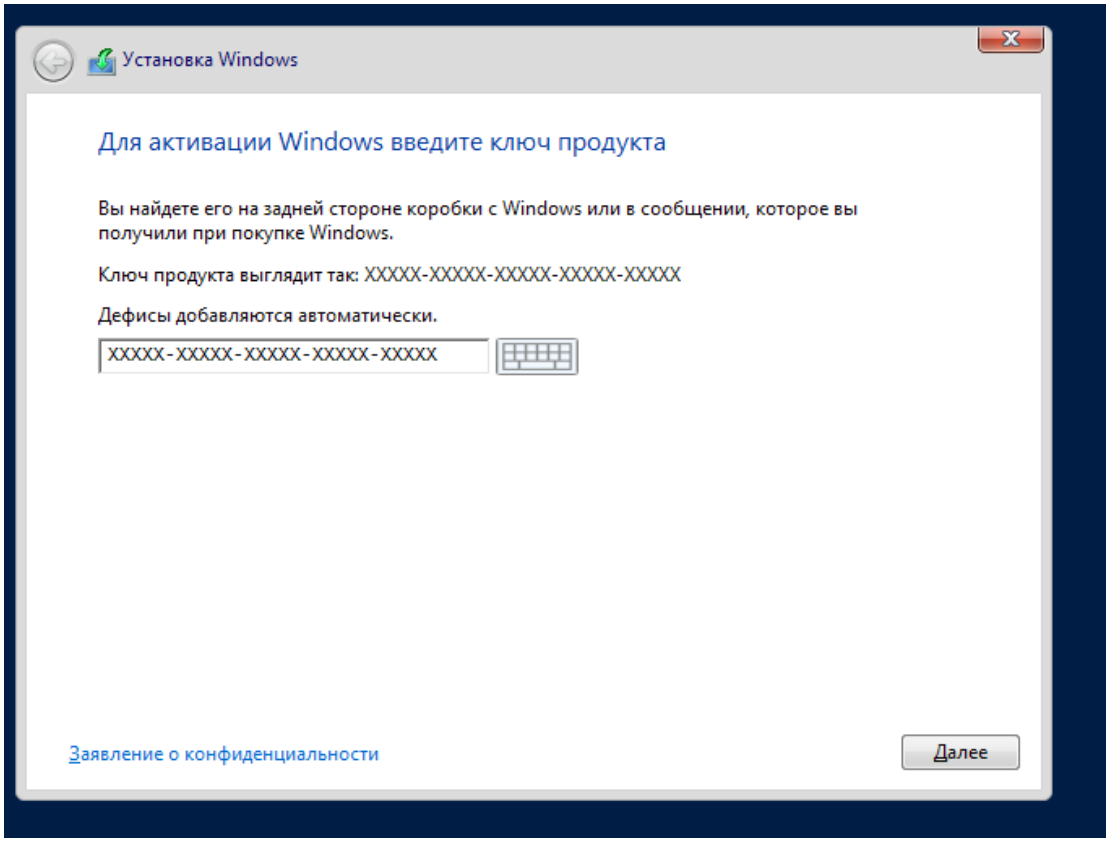
функционал — возможность переключения между вариантами установки. Например, вы можете установить сервер с графическим интерфейсом и использовать соответствующие инструменты для его настройки, а затем переключиться на вариант установки основных компонентов и получить все преимущества повышенного уровня безопасности и экономного потребления ресурсов.

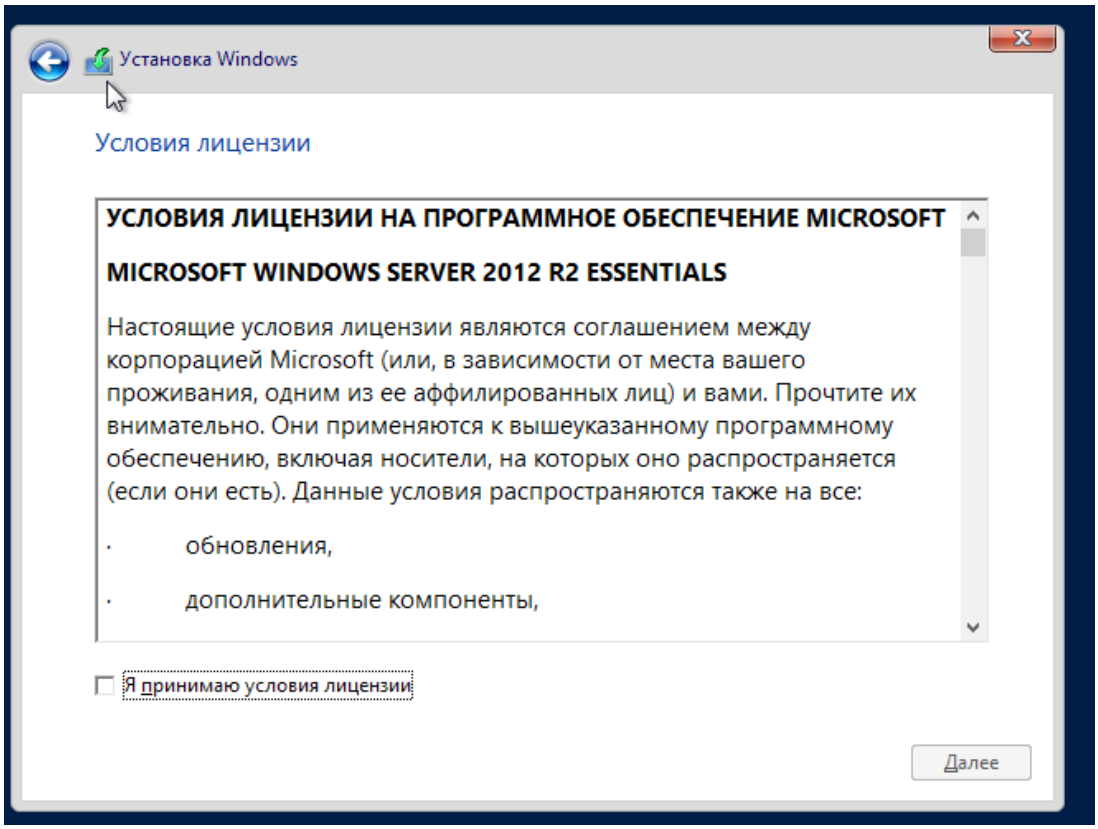
Непосредственно процесс установки (в картинках)

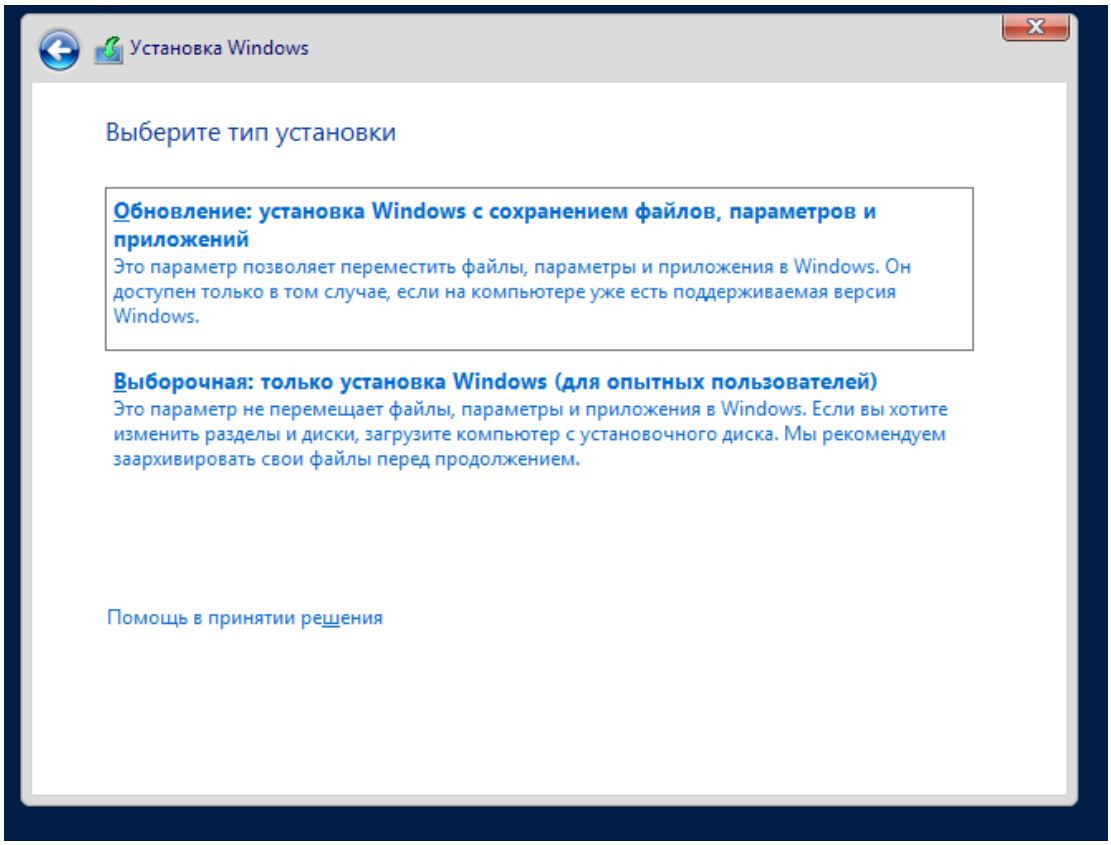
Мы рассмотрим вариант установки с графическим интерфейсом и полным функционалом. В таком варианте установка **Server 2012** не отличается от **Server 2008 R2**, да и в целом не сложнее установки **Windows 7-10**, так что будет проста и понятна даже неискушенному пользователю.

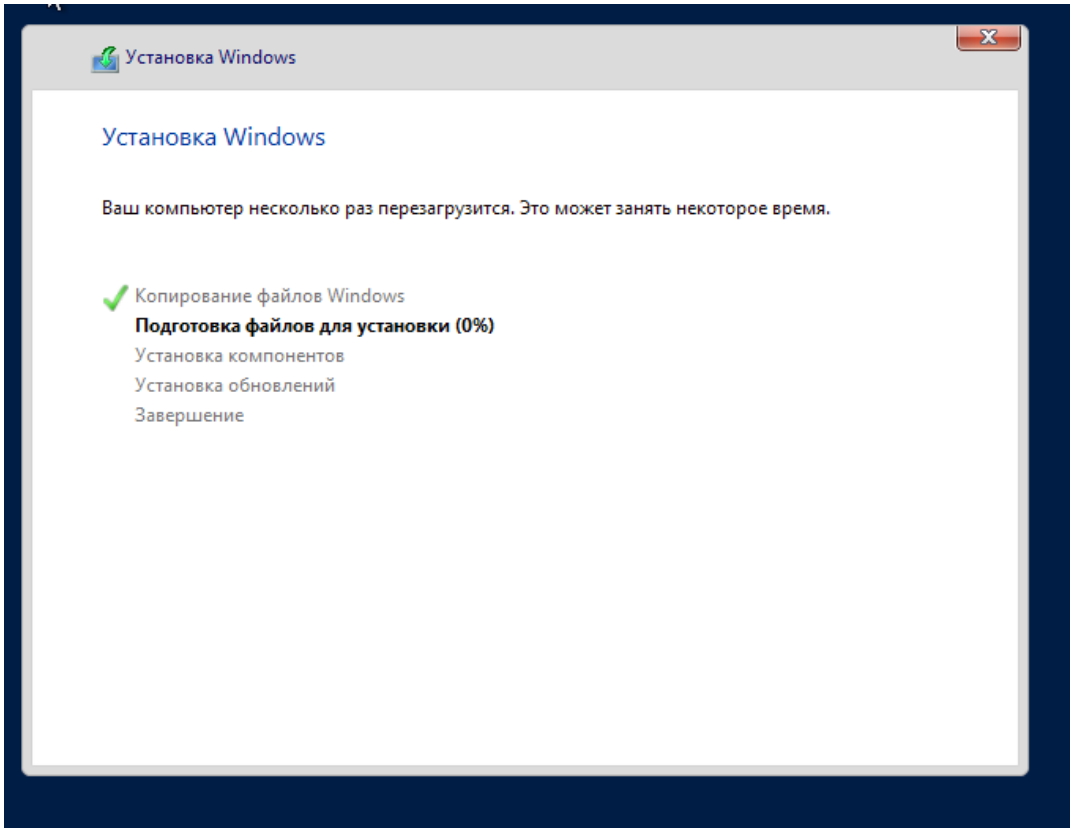

Установка Windows
Windows Server 2012 R2
Устанавливаемый язык:
Русский (Россия)
Формат времени и денежных единиц:
Русский (Россия)
Метод ввода (раскладка клавиатуры):
Русская
Выберите нужный язык и другие параметры, а затем нажмите кнопку "Далее".
© Корпорация Майкрософт (Microsoft Corporation), 2013. Все права защищены.
Далее


Установка Windows
Windows Server 2012 R2
Установить
Восстановление системы
© Корпорация Майкрософт (Microsoft Corporation), 2013. Все права защищены.

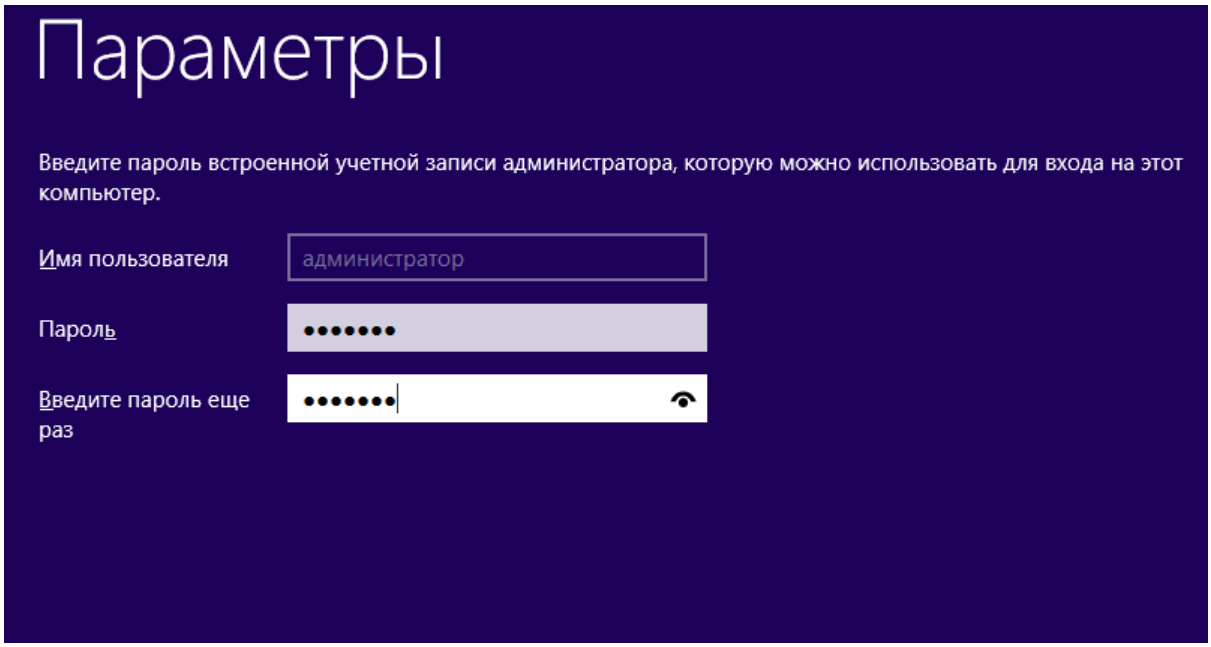

Установка Windows
Для активации Windows введите ключ продукта
Вы найдете его на задней стороне коробки с Windows или в сообщении, которое вы получили при покупке Windows.
Ключ продукта выглядит так: XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
Дефисы добавляются автоматически.
XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
Заявление о конфиденциальности
Далее


Установка Windows
Условия лицензии
УСЛОВИЯ ЛИЦЕНЗИИ НА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ MICROSOFT
MICROSOFT WINDOWS SERVER 2012 R2 ESSENTIALS
Настоящие условия лицензии являются соглашением между корпорацией Microsoft (или, в зависимости от места вашего проживания, одним из ее аффилированных лиц) и вами. Прочтите их внимательно. Они применяются к вышеуказанному программному обеспечению, включая носители, на которых оно распространяется (если они есть). Данные условия распространяются также на все:
· обновления,
· дополнительные компоненты,
Я принимаю условия лицензии
Далее

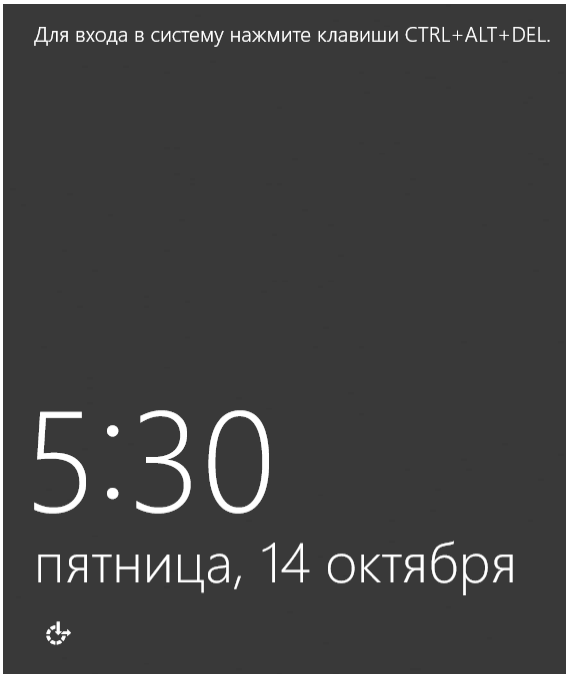

Установка Windows
Выберите тип установки
Обновление: установка Windows с сохранением файлов, параметров и приложений
Это параметр позволяет переместить файлы, параметры и приложения в Windows. Он доступен только в том случае, если на компьютере уже есть поддерживаемая версия Windows.
Выборочная: только установка Windows (для опытных пользователей)
Это параметр не перемещает файлы, параметры и приложения в Windows. Если вы хотите изменить разделы и диски, загрузите компьютер с установочного диска. Мы рекомендуем заархивировать свои файлы перед продолжением.
Помощь в принятии решения

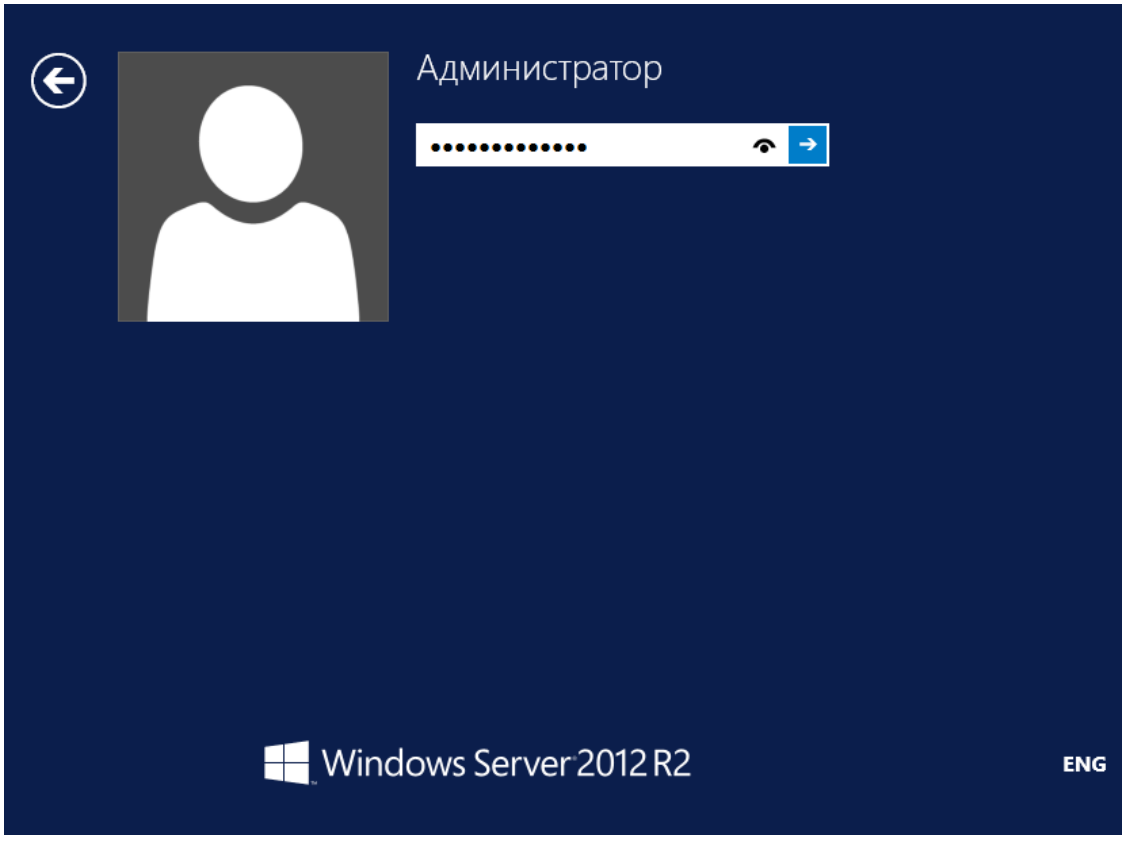

Установка Windows
Установка Windows
Ваш компьютер несколько раз перезагрузится. Это может занять некоторое время.
Копирование файлов Windows
Подготовка файлов для установки (0%)
Установка компонентов
Установка обновлений
Завершение

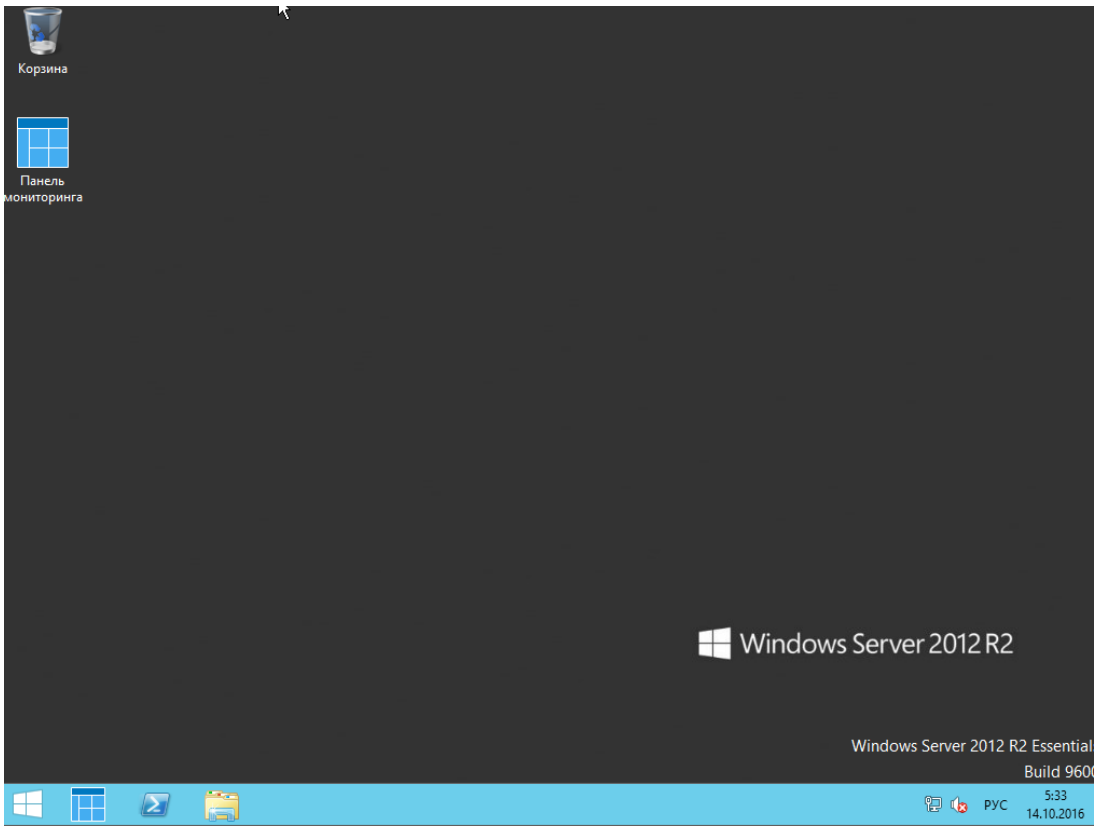
Как видите, процесс установки графической версии достаточно типичен практически во всём. На том же этапе выбора, обновляться или сделать опытную установку, просто дают возможность подробить разделы (что, само собой, стоит сделать), отформатировать их и выбрать тот, куда будет производится установка.



В завершении установки будет предложено задать пароль администратора, что стоит сделать с умом (см.прошлую книгу или статью на **sonikelf.ru** по генерации паролей).

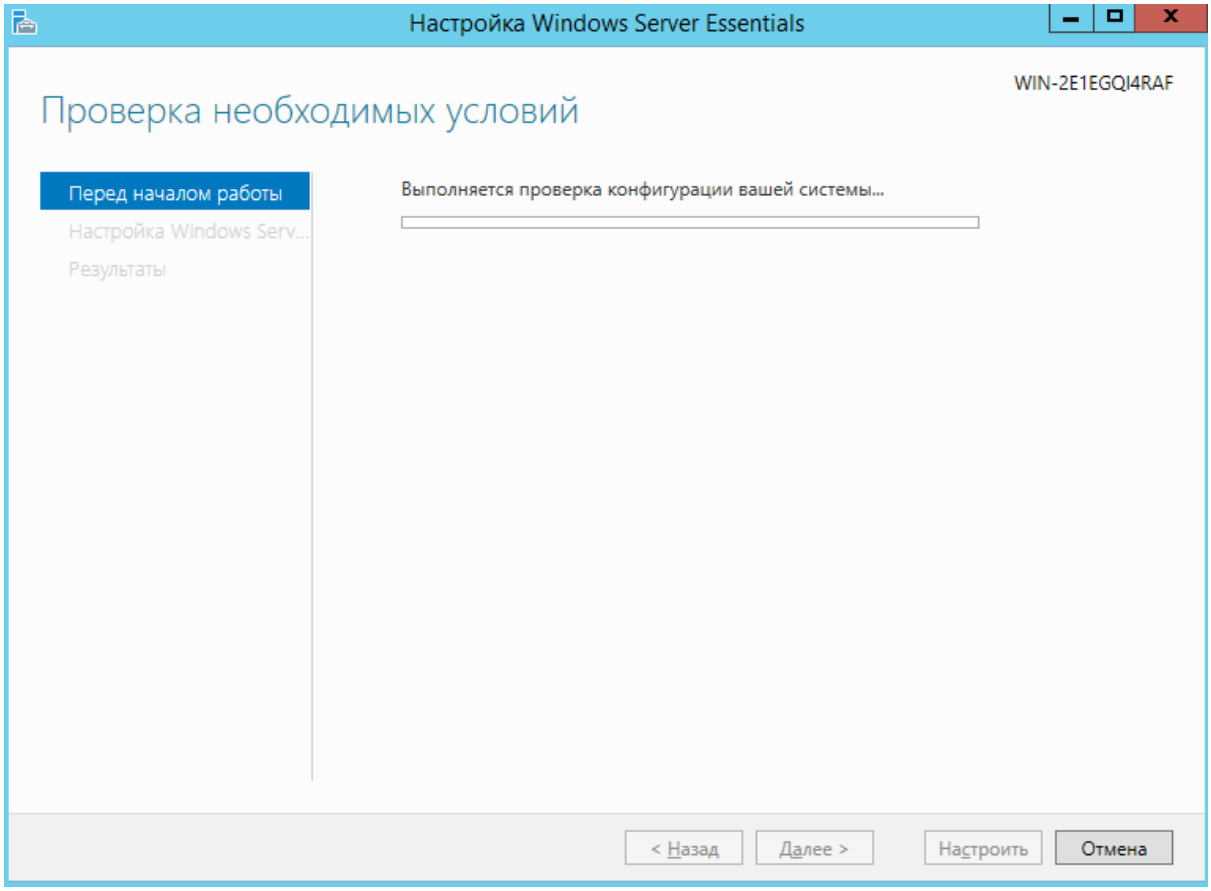

Для входа в систему нажмите клавиши CTRL+ALT+DEL.
5:30
пятница, 14 октября

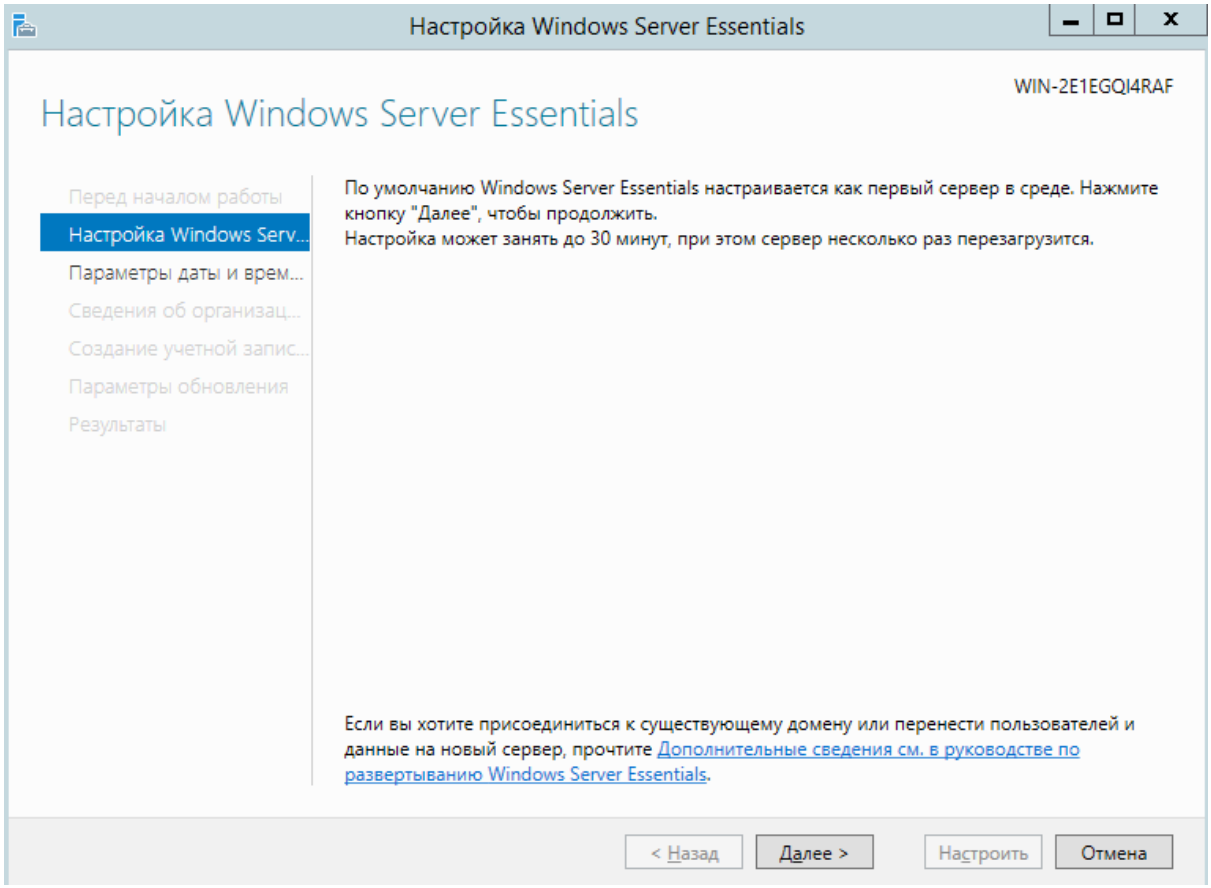

Администратор
Windows Server 2012 R2
ENG



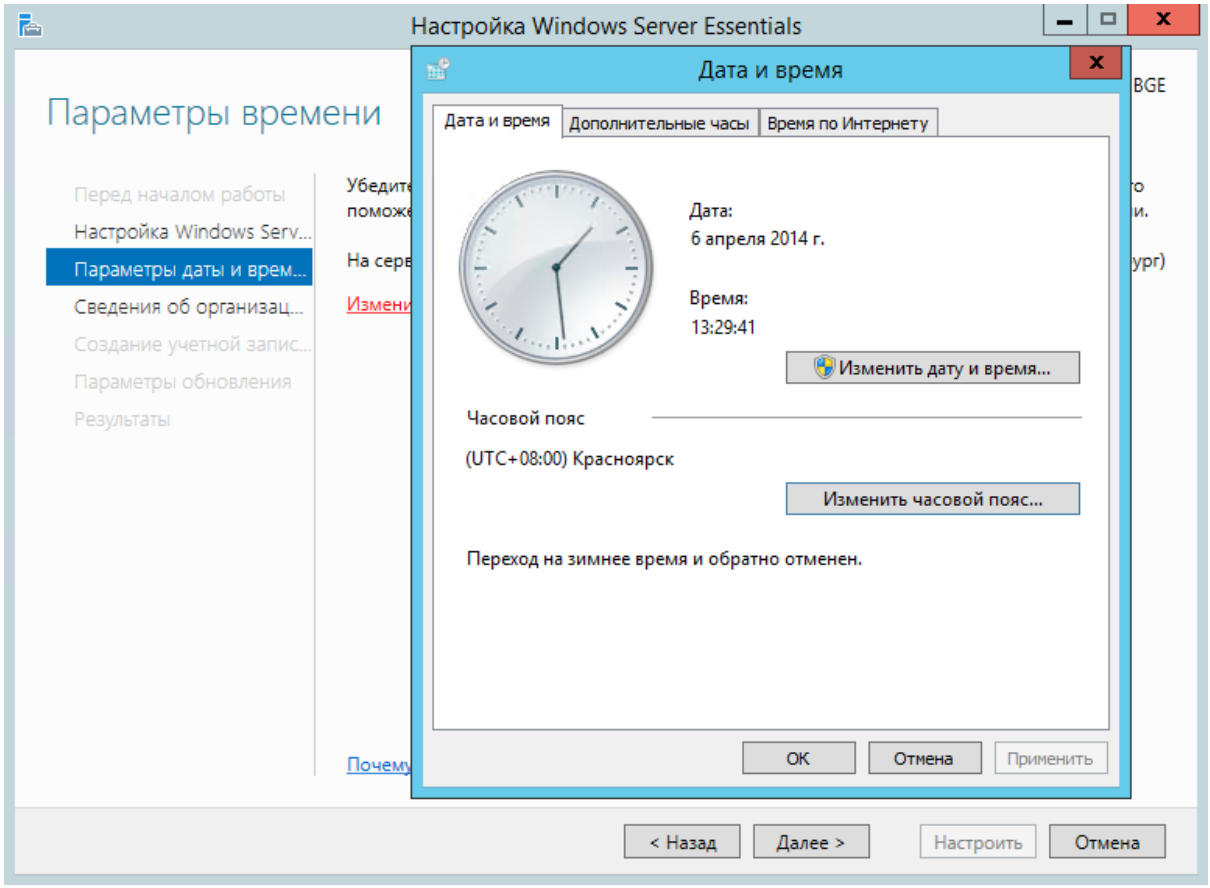
Настройка Windows Server 2012 Essentials

После первого входа в свежеустановленную **ОС** запустится мастер «Настройка **Windows Server Essentials**», который поможет произвести первоначальную настройку.

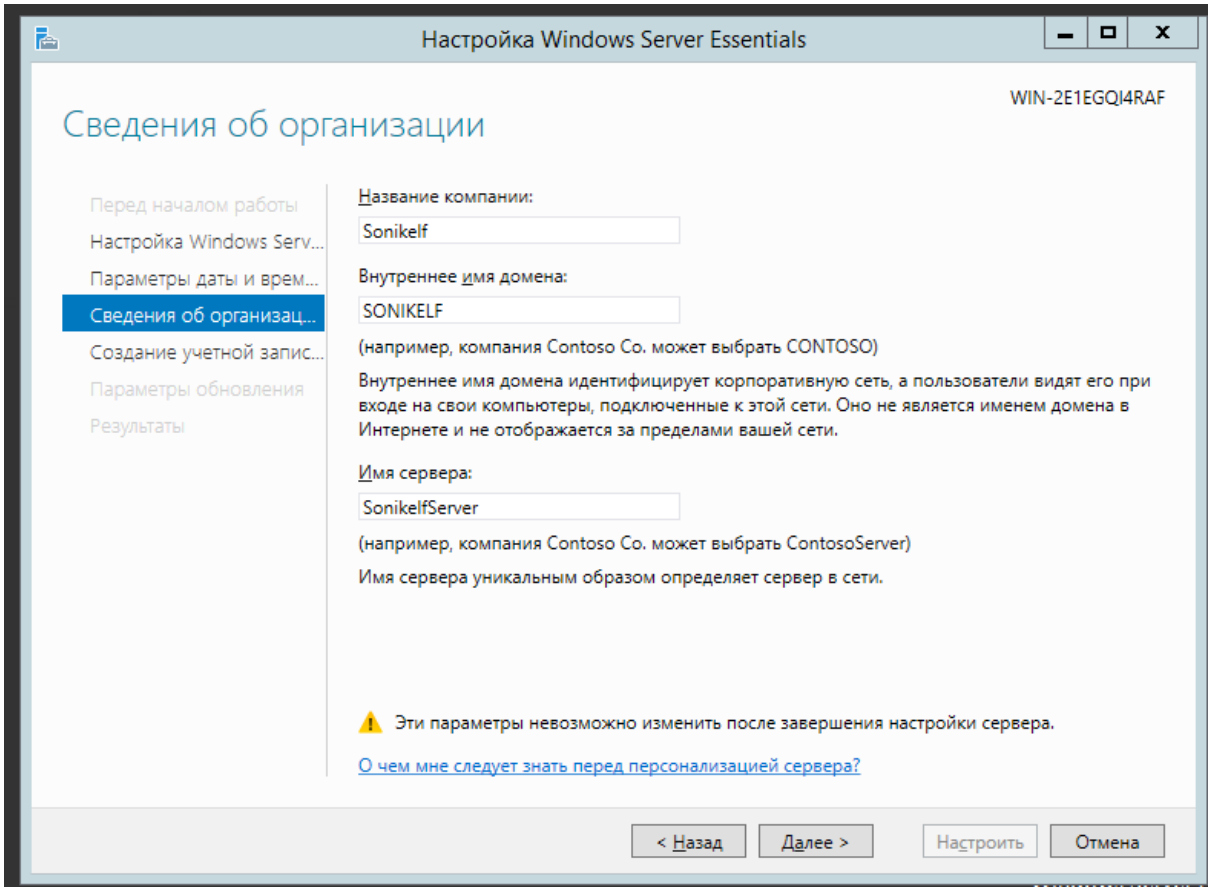

Настройка Windows Server Essentials
WIN-2E1EGQI4RAF
Проверка необходимых условий
Перед началом работы
Настройка Windows Serv...
Результаты
Выполняется проверка конфигурации вашей системы...
< Назад
Далее >
Настроить
Отмена



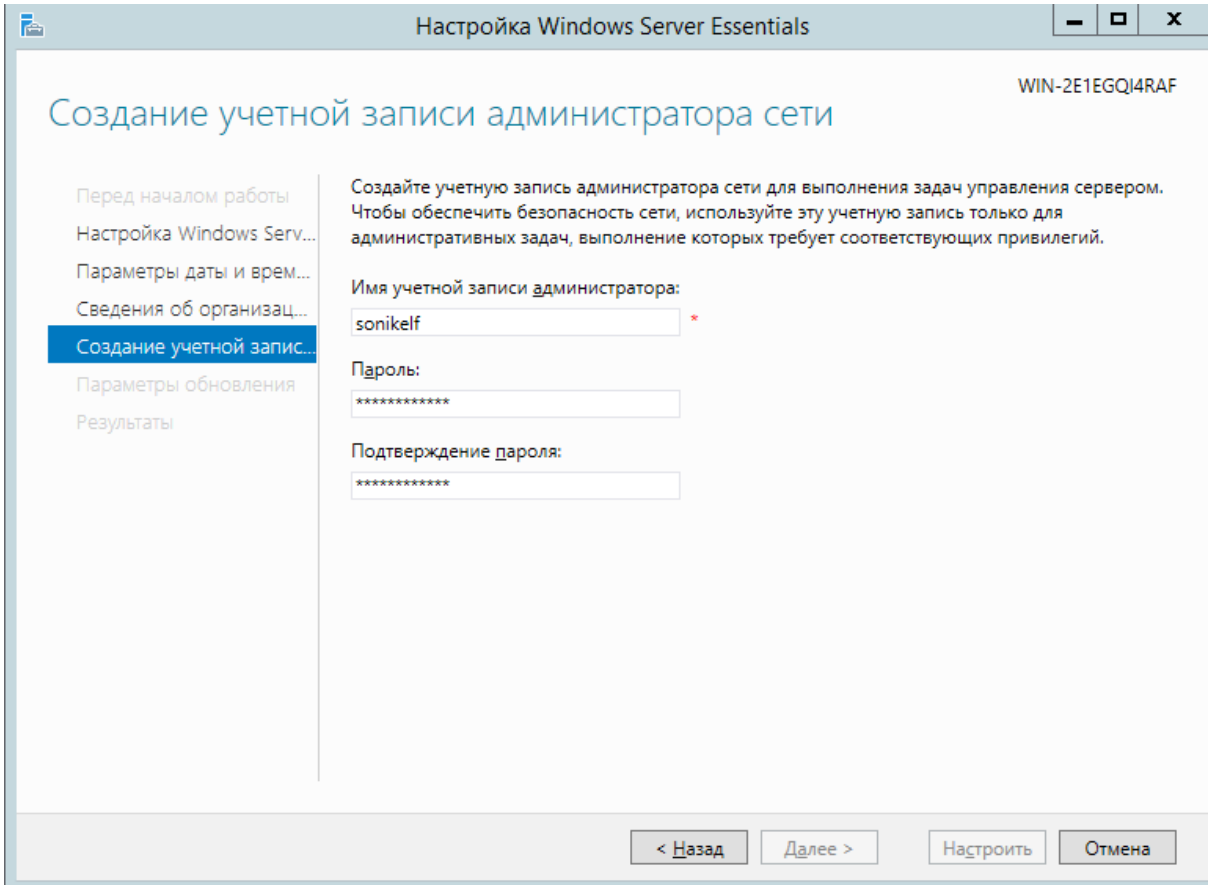
На первом шаге вам необходимо задать настройки даты и времени.



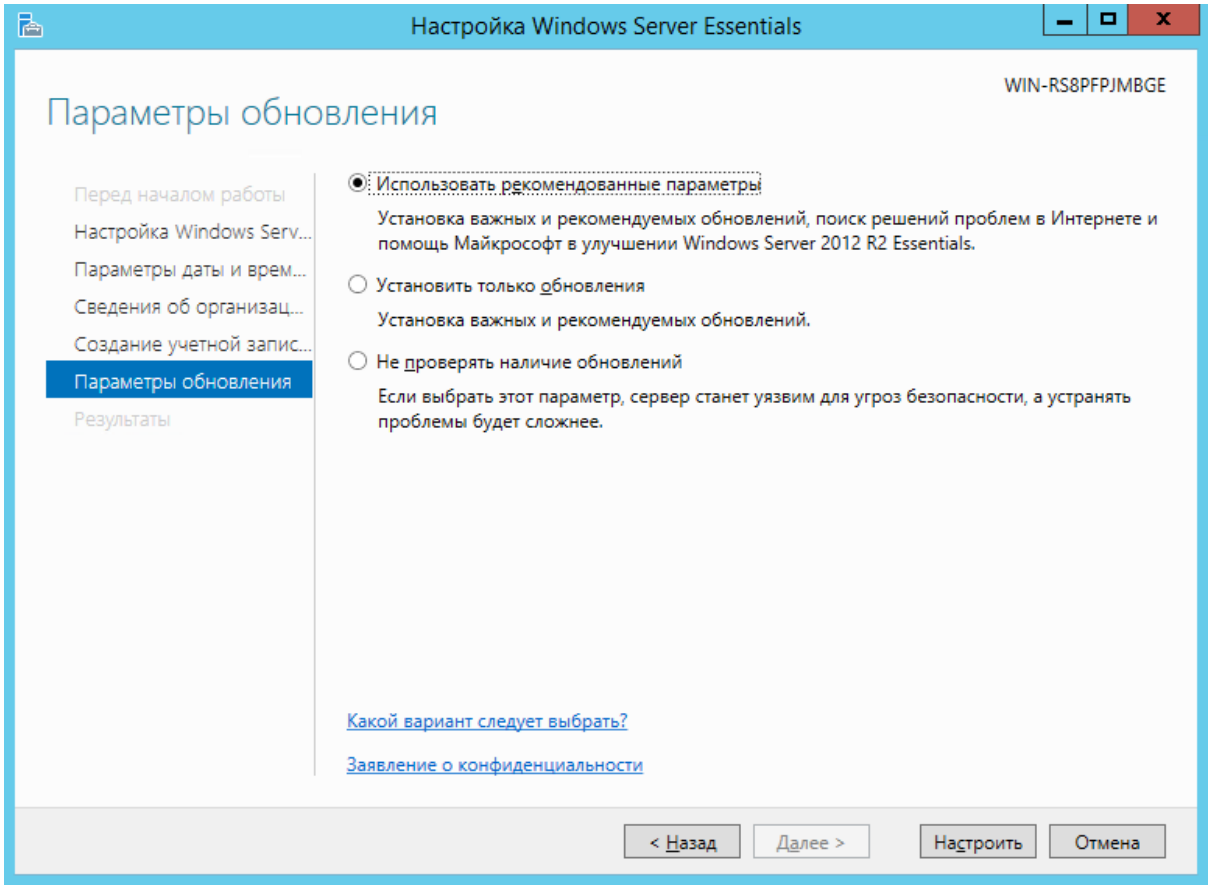
На втором шаге вам необходимо заполнить на английском языке название компании. Имя домена и имя сервера будут в таком случае сгенерированы автоматически, хотя конечно вы можете поменять их.



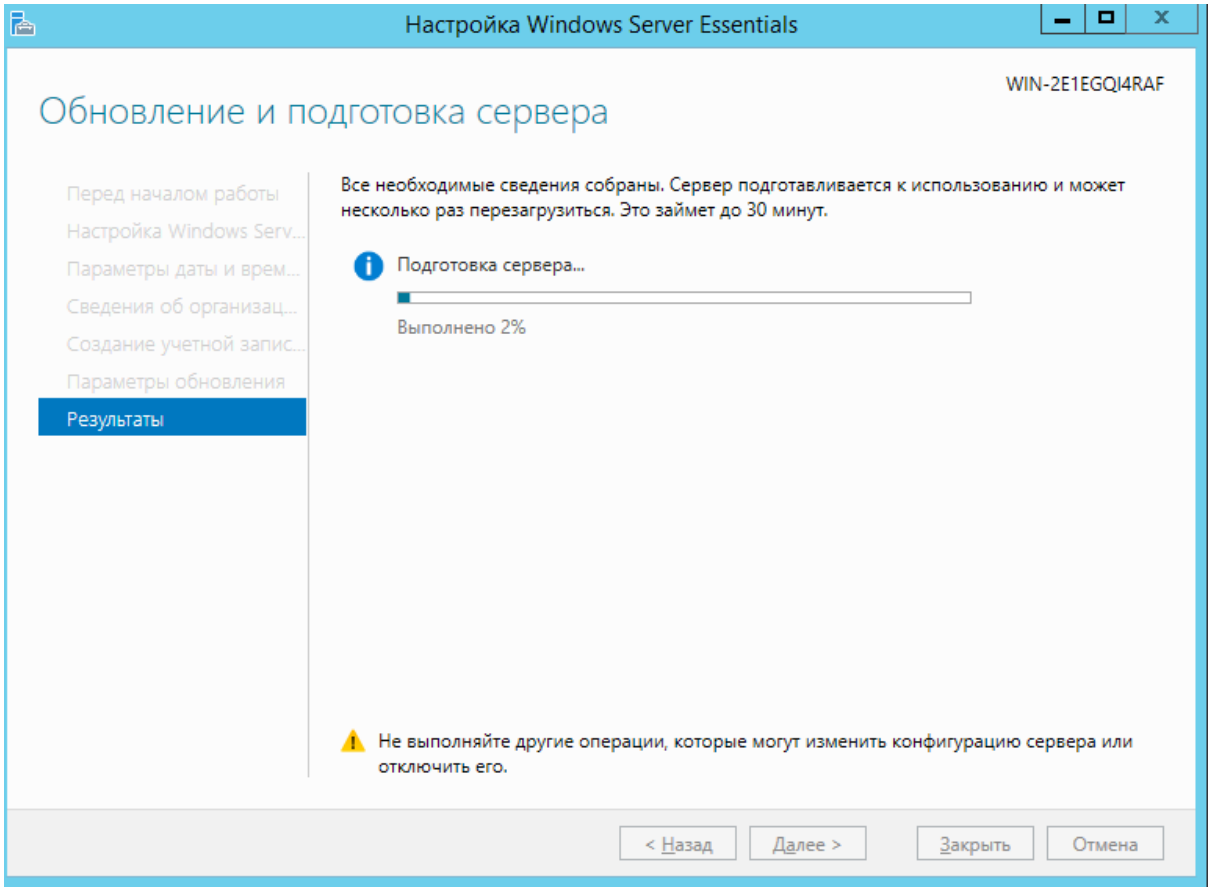
На следующем этапе потребуется заполнить имя Администратора и задать его пароль.



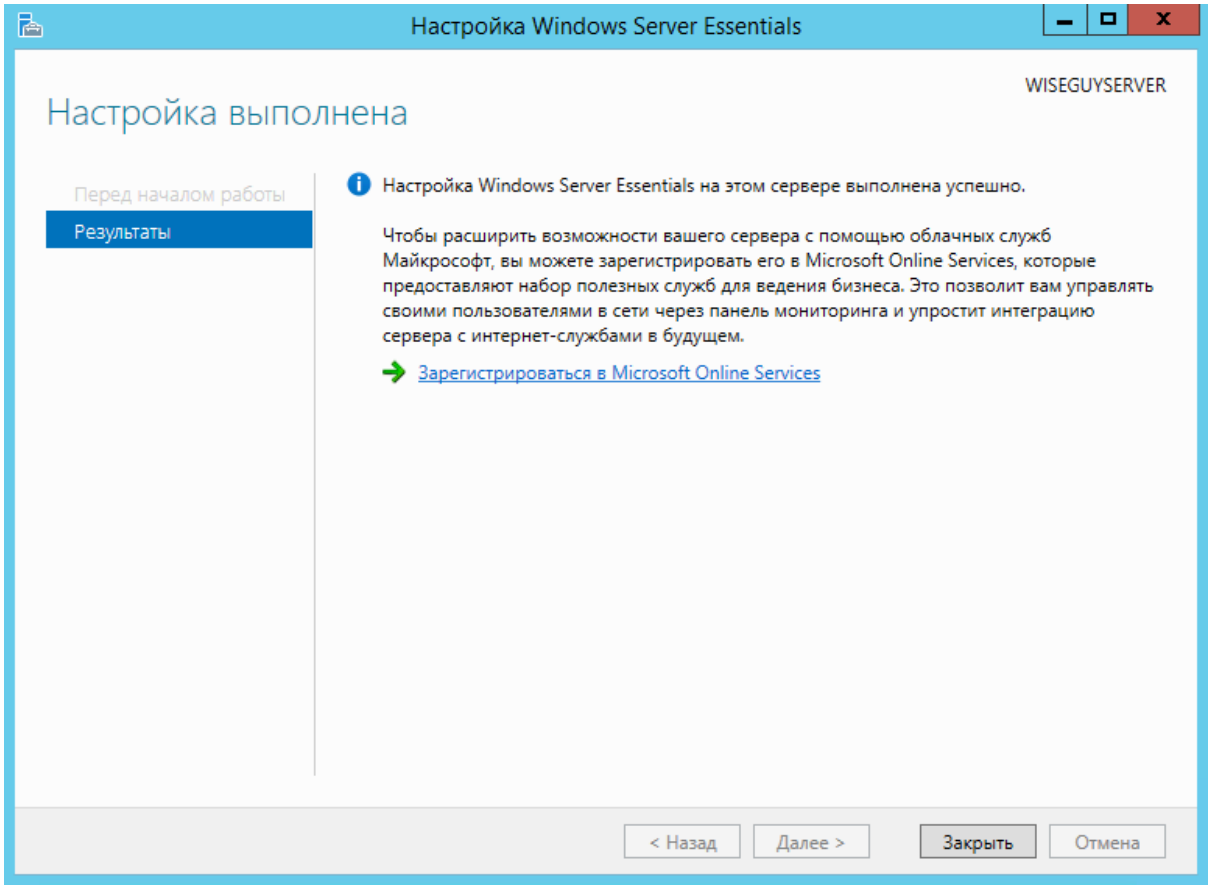
На последнем шаге необходимо указать способ обновления операционной системы и тыркнуть в кнопку “**Настроить**”:



После этого запустится процесс, который произведет все необходимые первоначальные настройки, что займет порядка **30** минут и потребует несколько перезагрузок.

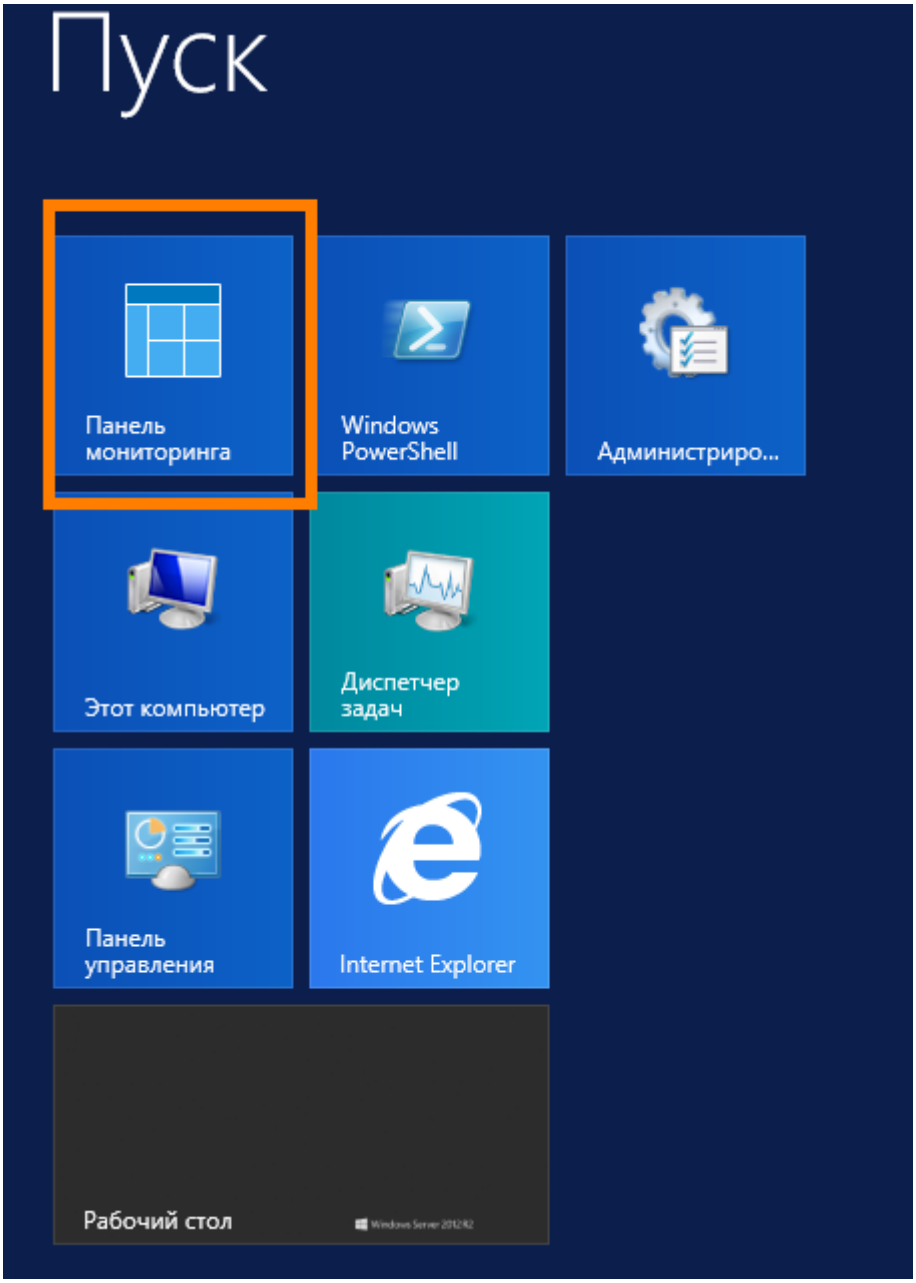


За это время **ОС** успеет в частности установить необходимые роли и настроить сервер в качестве домен-контроллера для нового домена.



Настройка в первичном виде

Продукт весьма большой и обширный, я хотел бы рассказать о самых базовых возможностях настройки, таких как создание пользователей, настройка удаленного доступа, создание папок, подключение клиентов.

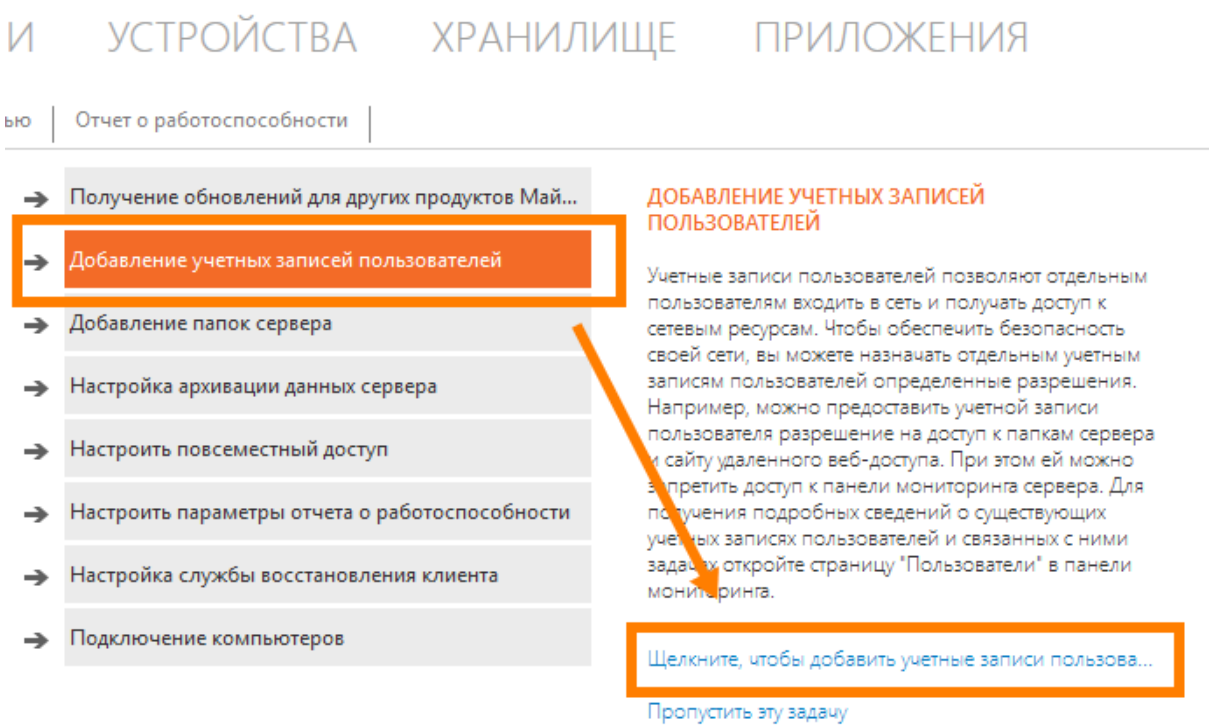


Вся настройка происходит в панели мониторинга, доступ к ней есть с рабочего стола, панели быстрого запуска и стартового экрана.

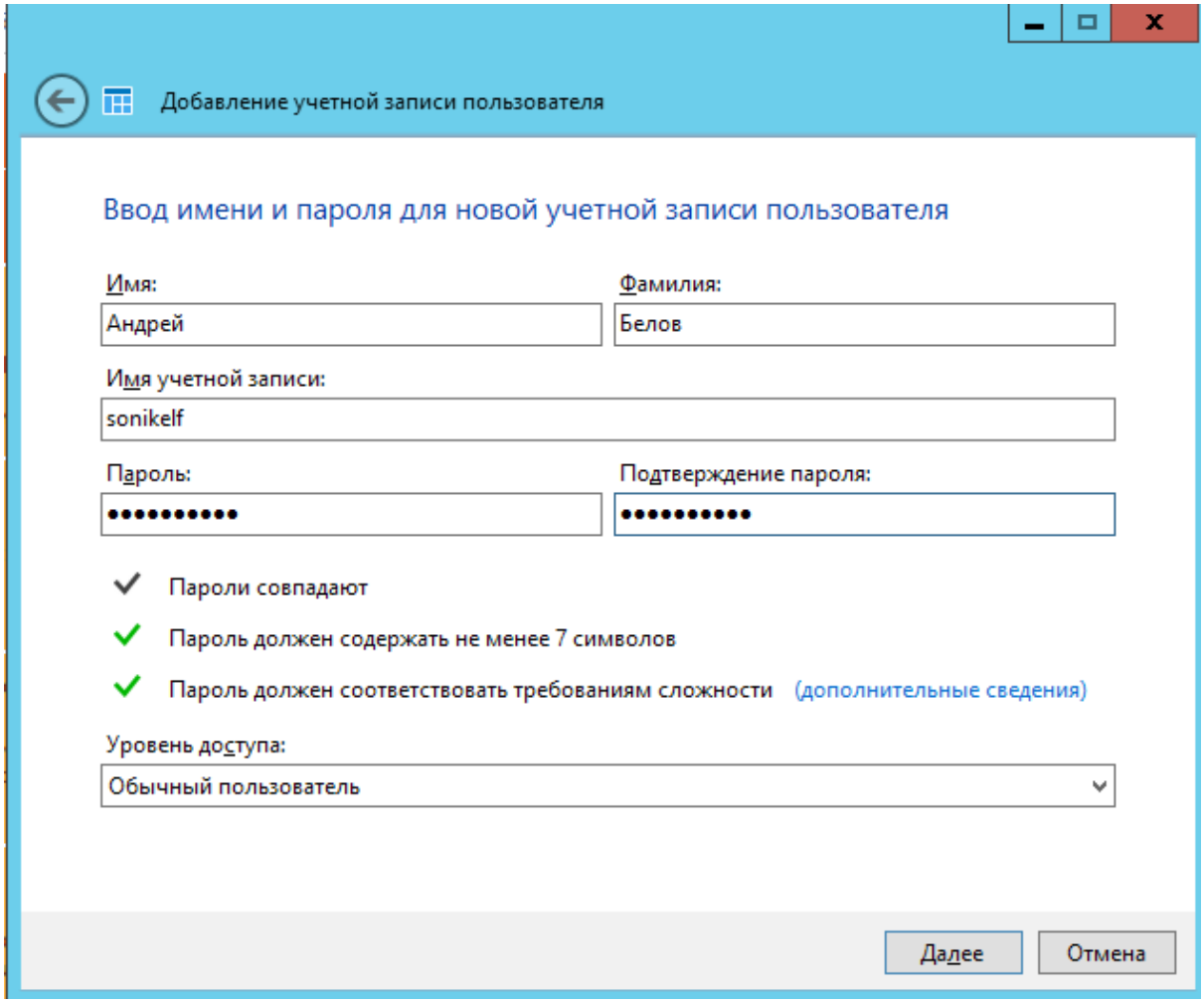
Создание пользователей

При первом запуске данной панели вам откроется вкладка “**Установка**”, на которой можно выполнить ряд задач по настройке сервера.

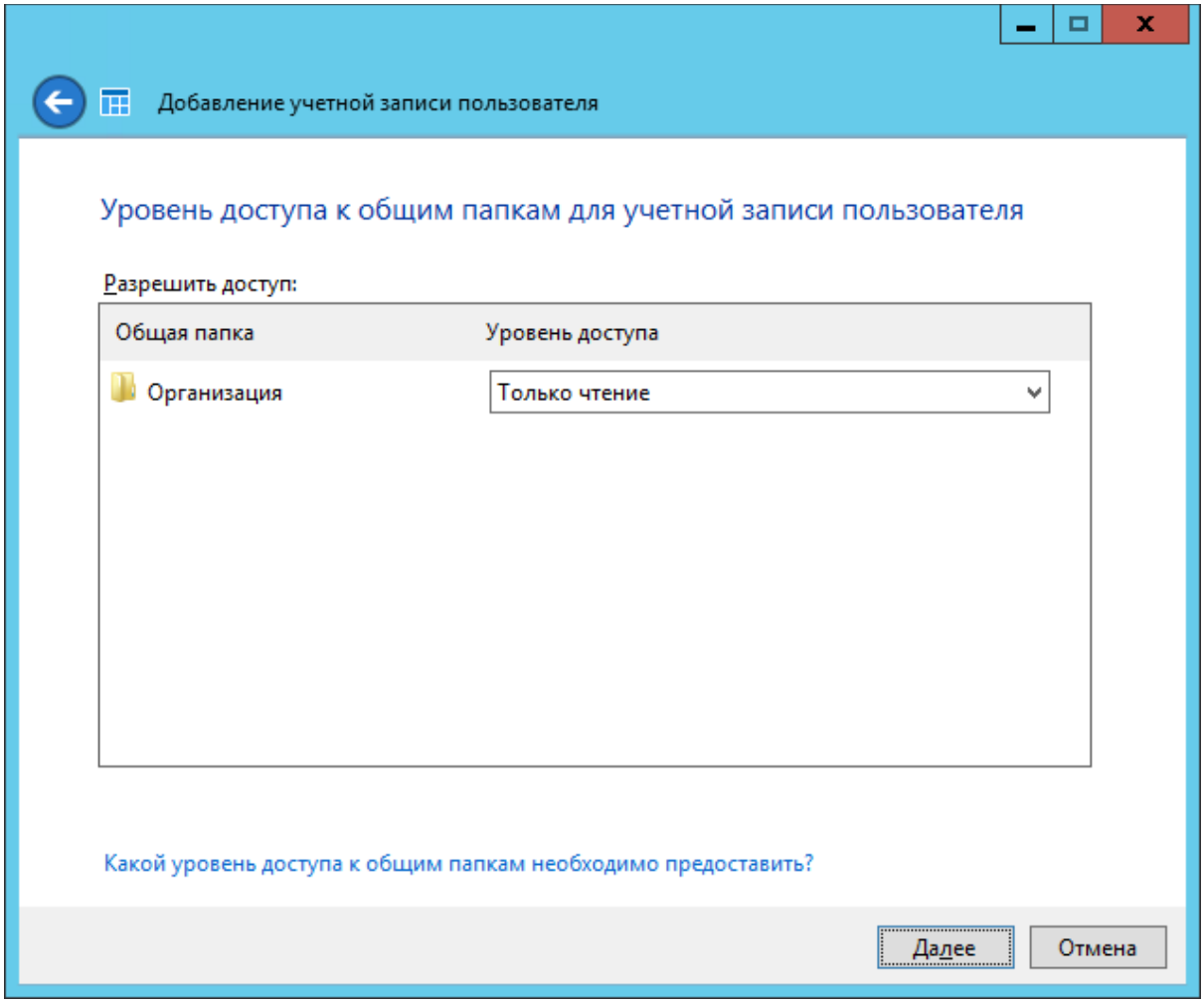
Я начну с добавления пользователей. Щелкаем ссылку для добавления учетных записей.



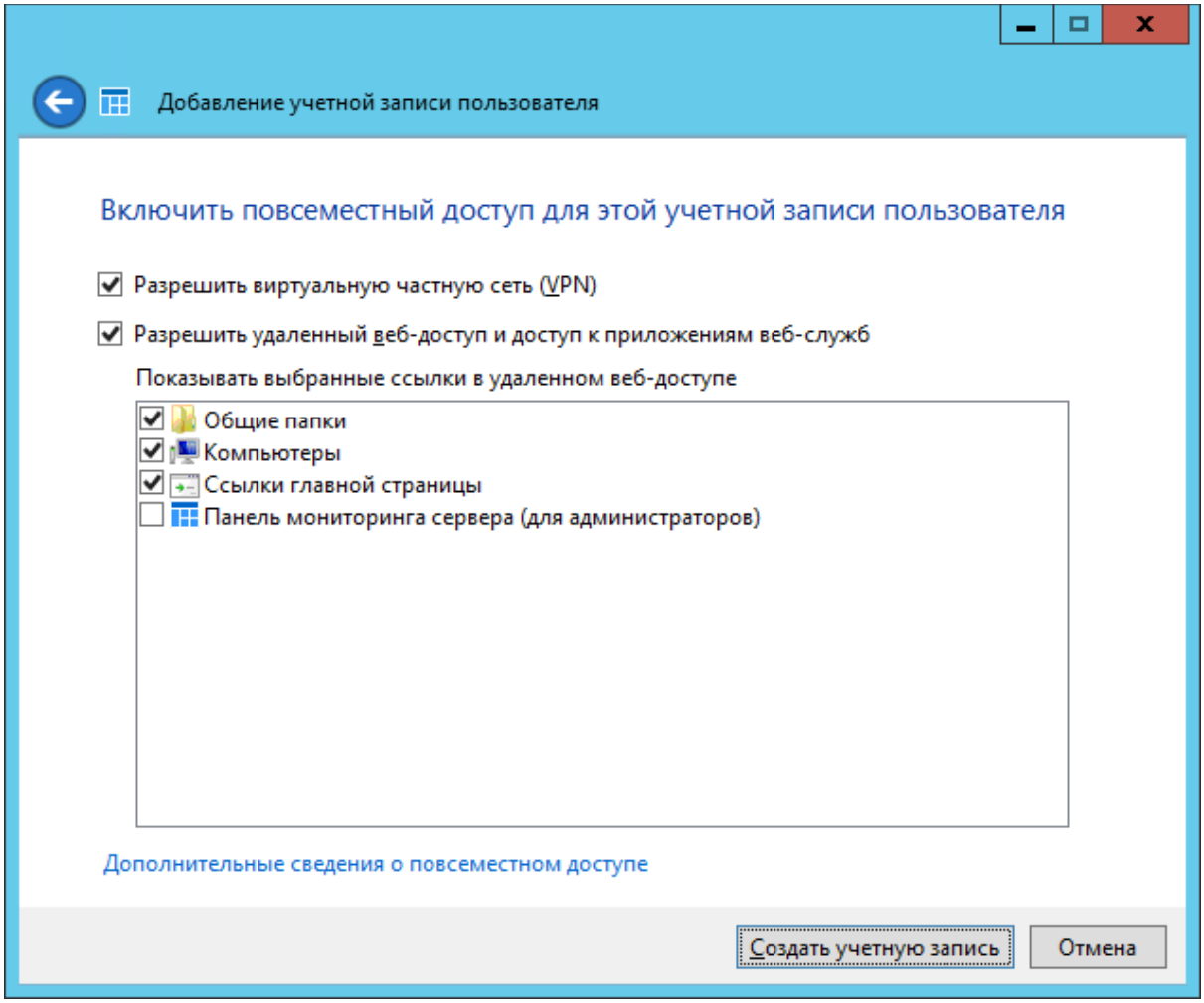
Заполняем поля формы, выставляем уровень доступа и, традиционно, нажимаем “**Далее**”:



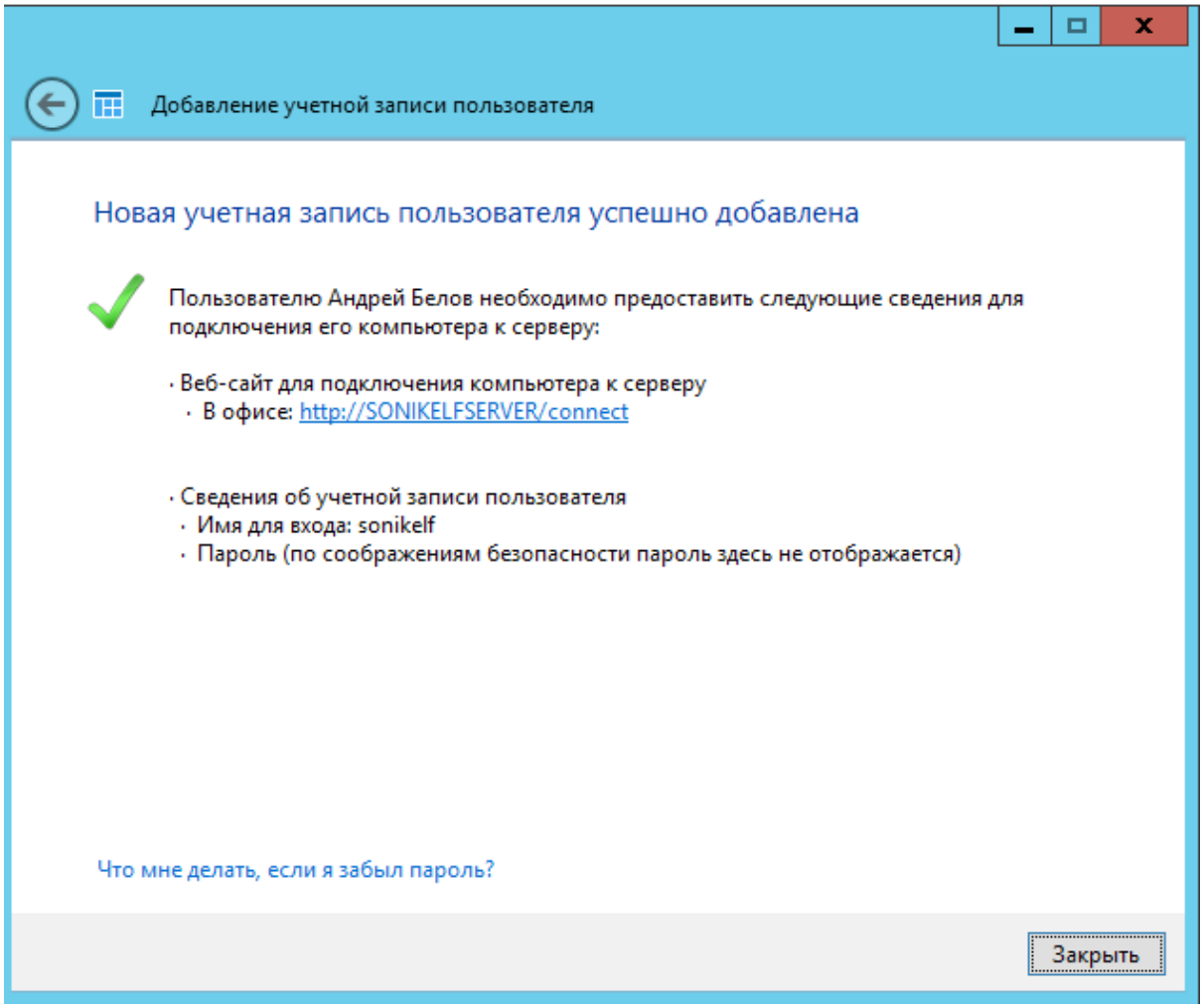
Выбираем уровень доступа к общим папкам, которые были созданы. На начальном этапе существует лишь одна – Организация. В дальнейшем вы можете менять разрешения на доступ, как из свойств пользователя, так и из свойств папки.



Далее устанавливаем, что будет доступно для пользователя удаленно. Про удаленный доступ расскажу чуть позже.



Учетная запись создана, посему жмём в “**Закрыть**”. К слову, если Вы задавали администраторский уровень прав, то выбора папок как выше не будет - по умолчанию доступно будет почти всё.

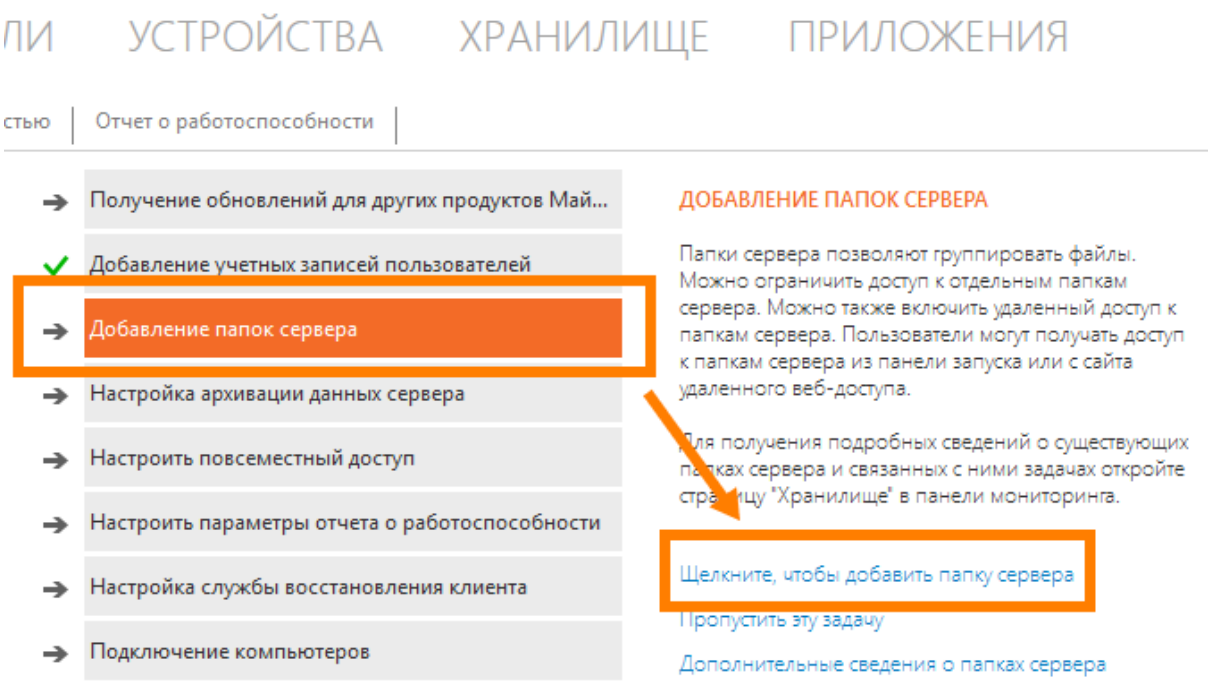


Подобным образом можно создать множество учетных записей. Безусловно, Вы можете пользоваться и привычным и знакомым для вас интерфейсом **Active Directory Users** and **Computers**, но в таком случае выдавать разрешения на доступ вам придется ручками.

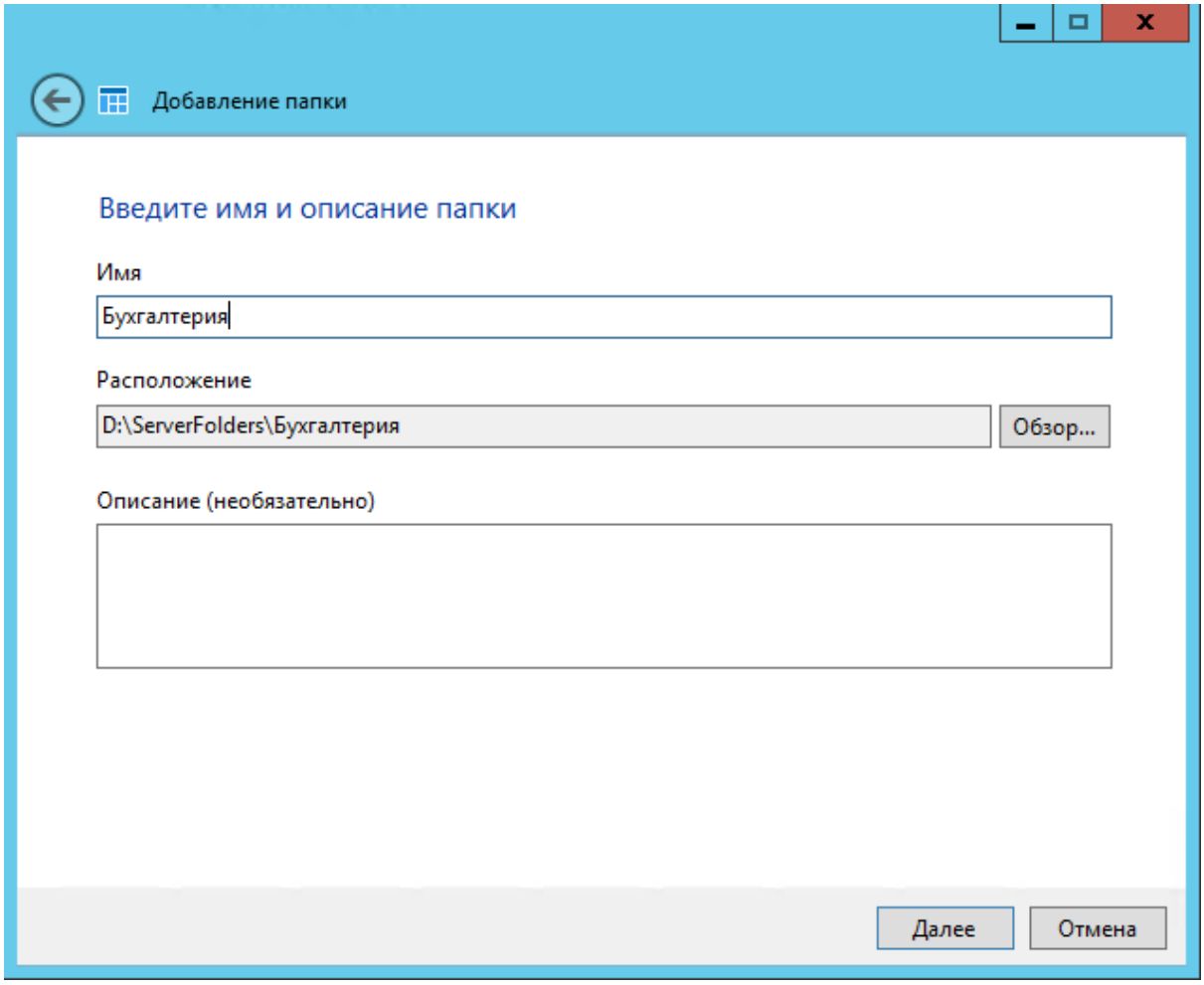
Добавление папок сервера

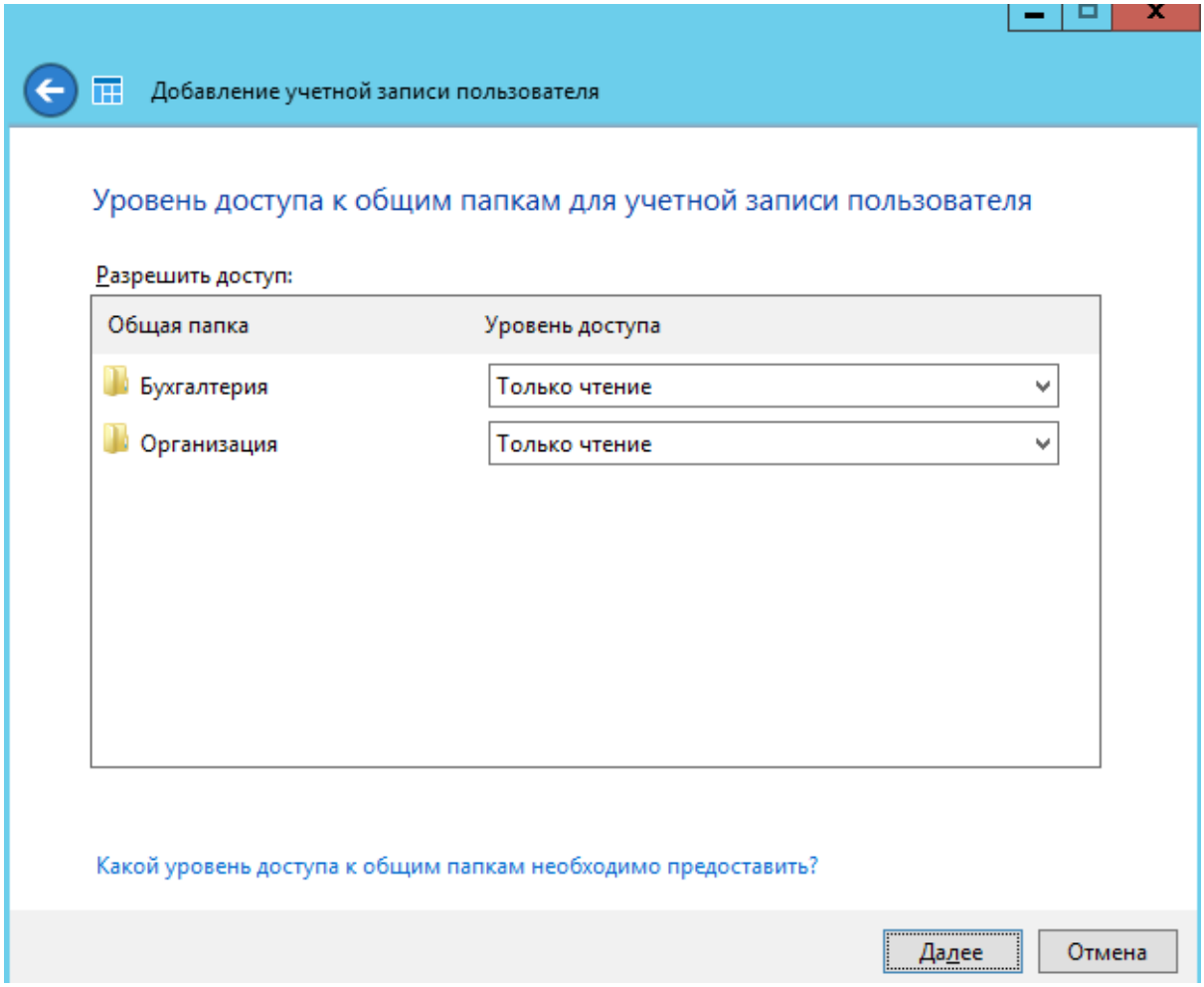
Для добавления папок существует другой мастер, который поможет и создать папку на диске, и общий доступ для нее настроить, и

разрешения выдать. Для его запуска необходимо щелкнуть соответствующую ссылку в панели мониторинга.

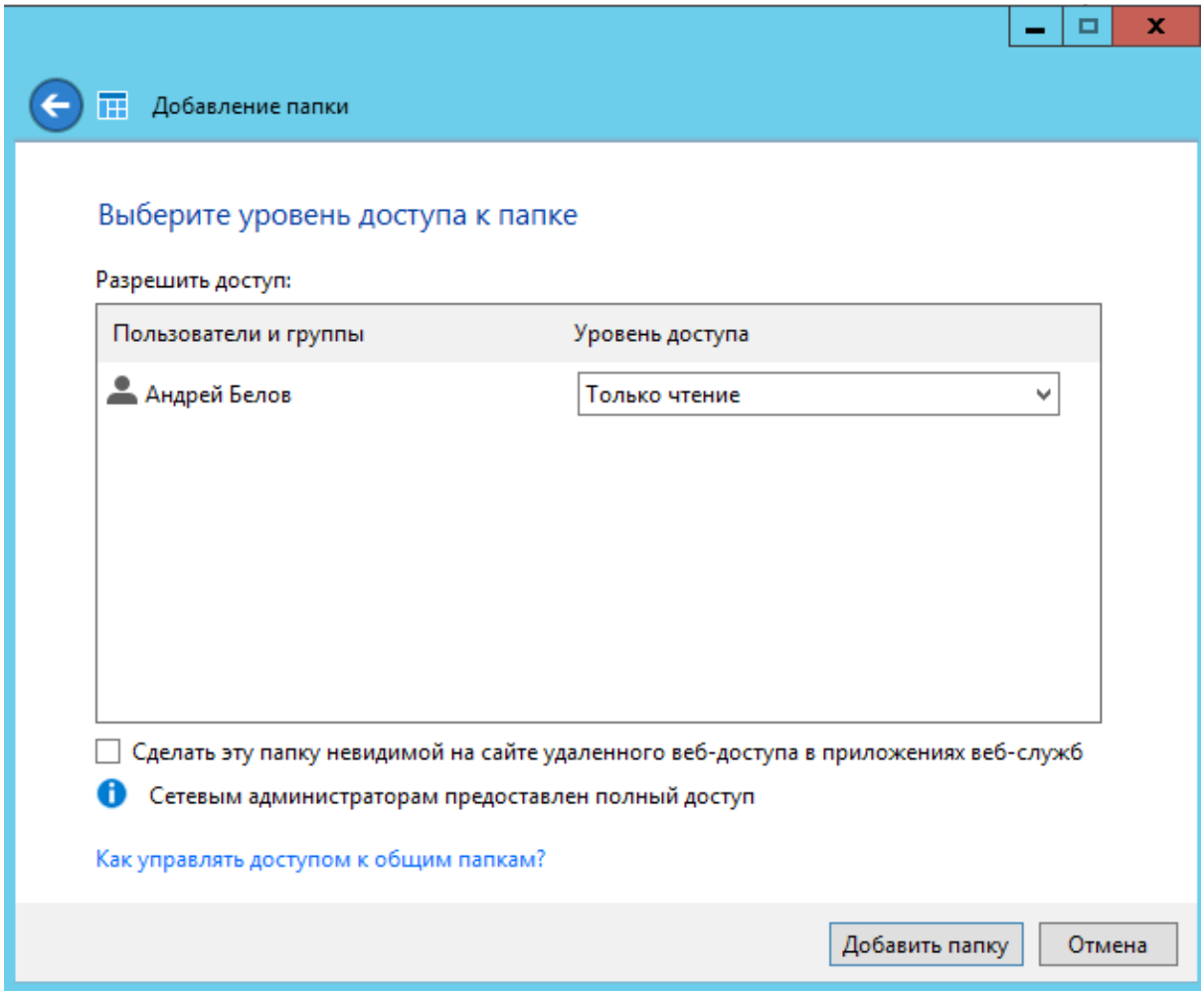


В открывшемся окне мастера вводим название. Можно изменить расположение и добавить описание. Нажимаем далее.

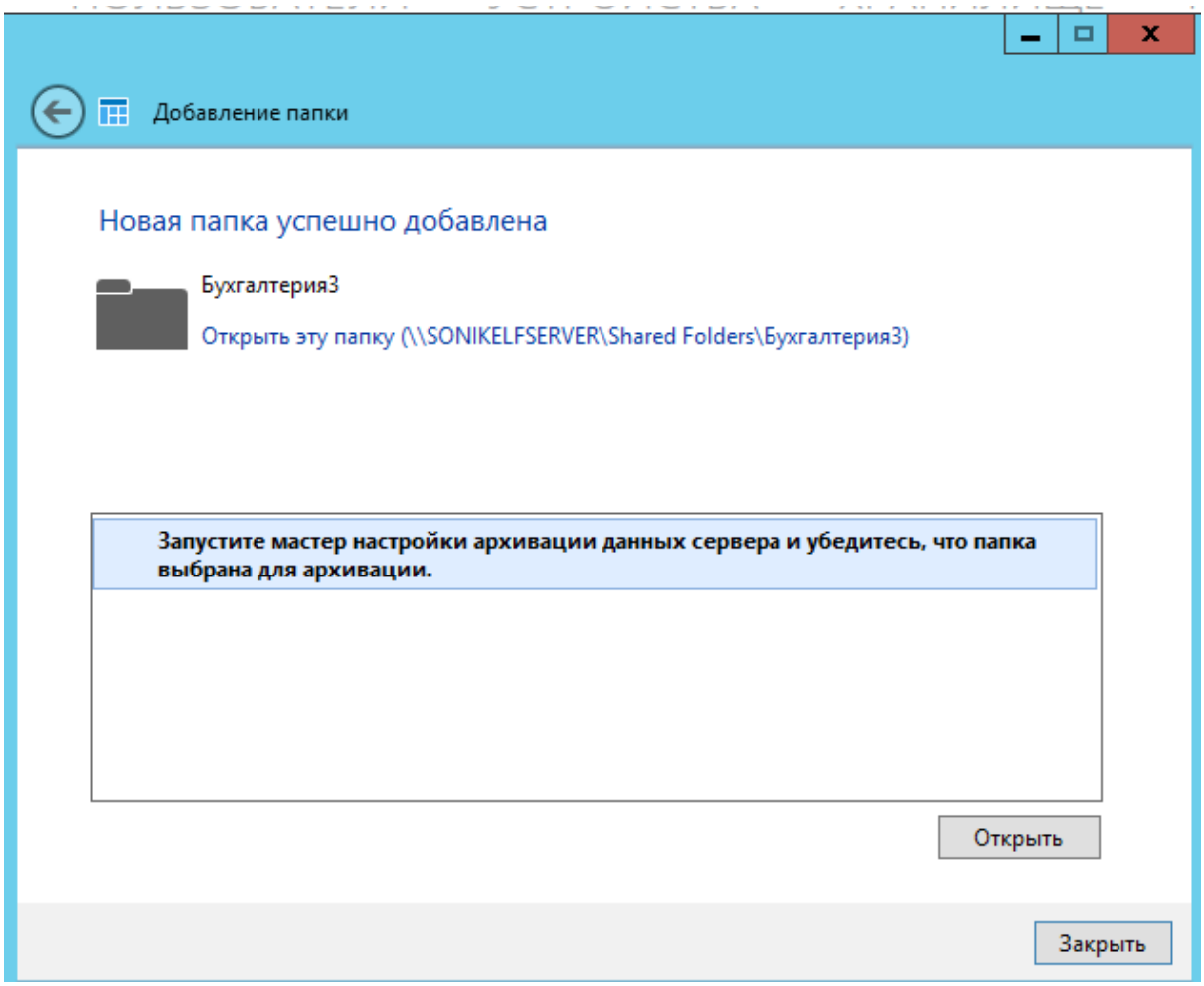

Добавление папки
Введите имя и описание папки
Имя
Бухгалтерия
Расположение
D:\ServerFolders\Бухгалтерия
Обзор...
Описание (необязательно)
Далее
Отмена



На следующей странице указываем необходимые разрешения. При желании делаем ее недоступной для удаленного доступа.

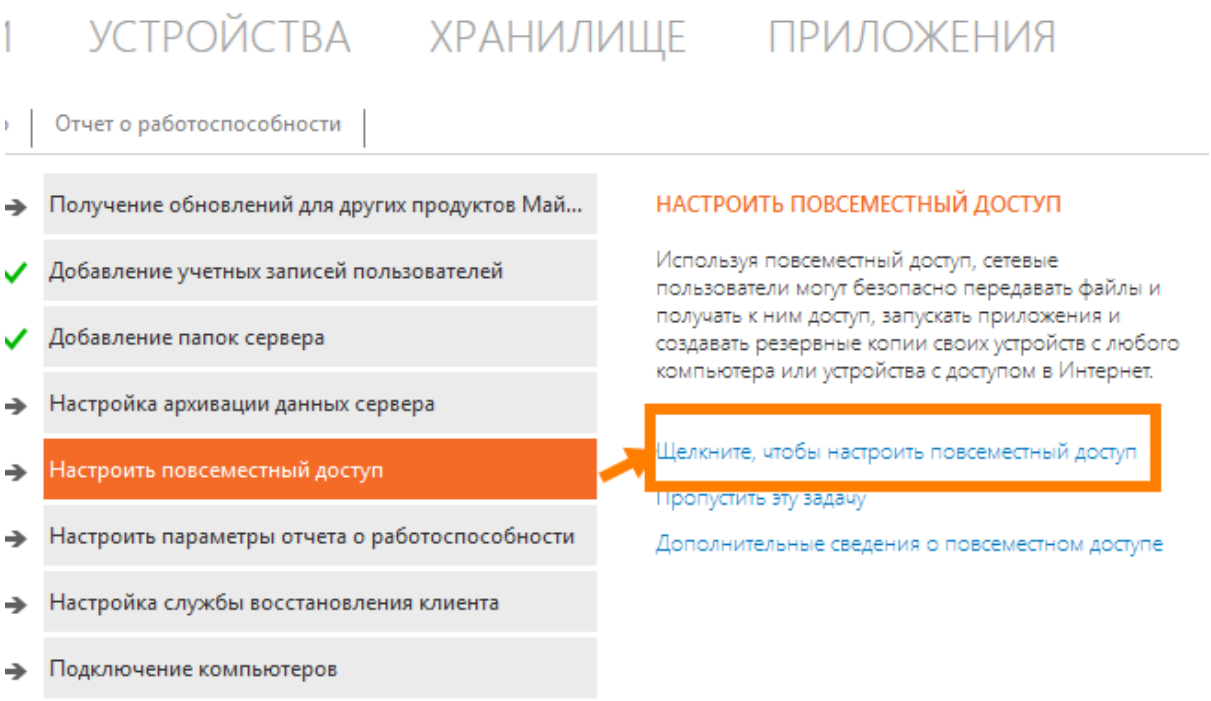


С последнего шага данного мастера можно запустить мастер настройки архивации. Нажимаем закрыть.



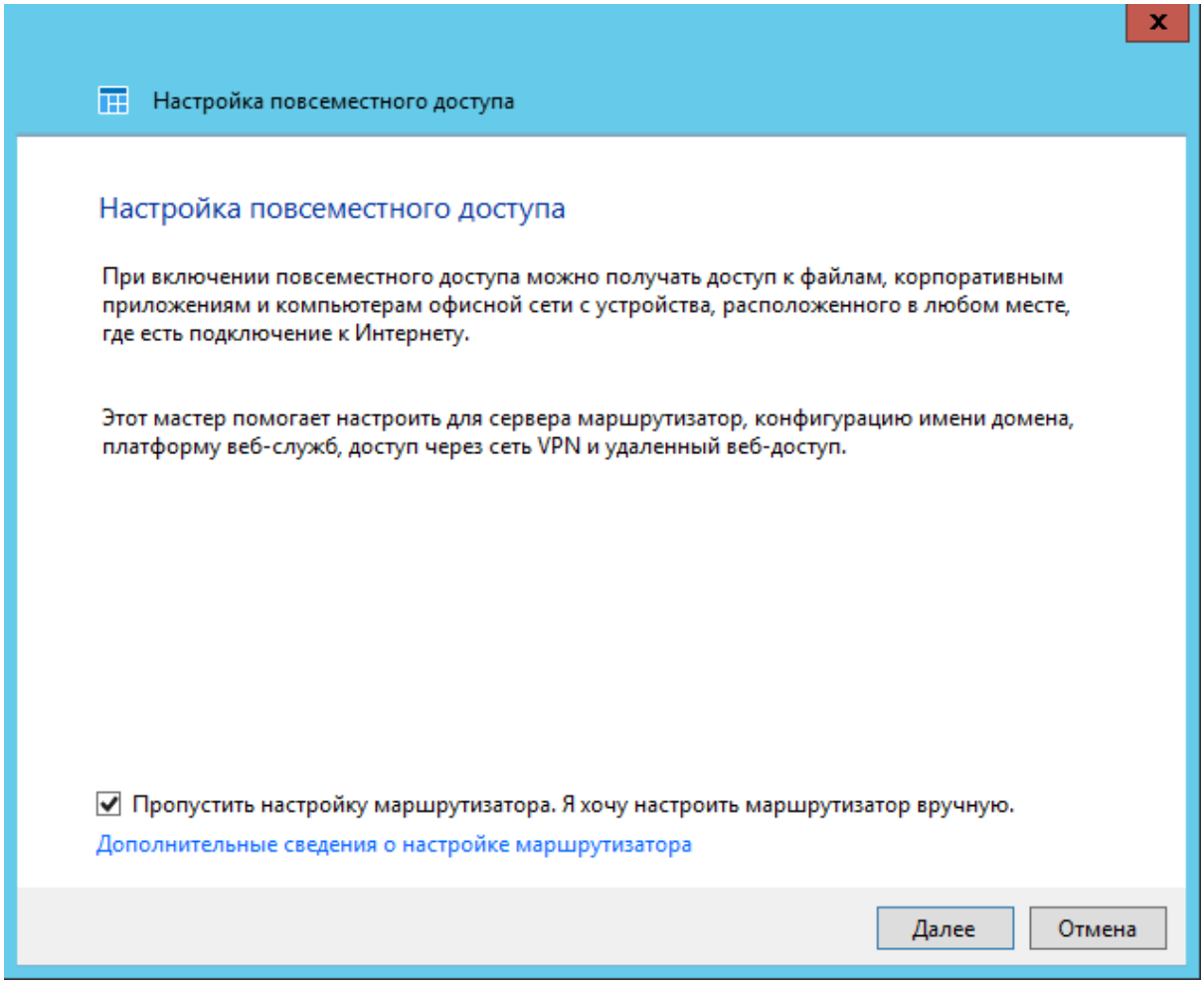
Настройка удаленного доступа

Один, наверное, из самых сложных этапов настройки **Windows Server 2012R2 Essentials**. Настройка так же происходит с помощью мастера. Мастер традиционно запускается из панели мониторинга.

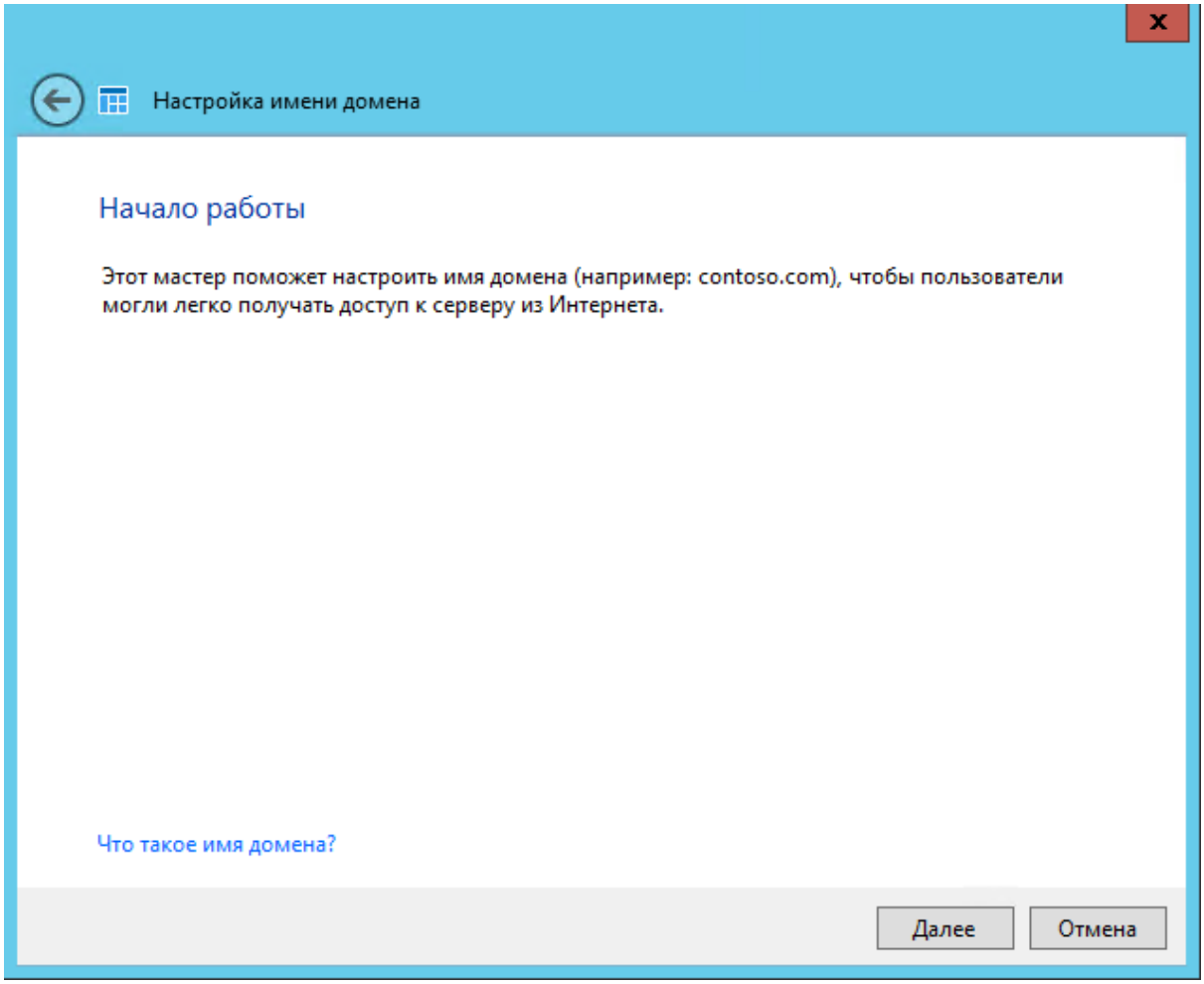


Первое, что Вам необходимо настроить, это ваш маршрутизатор – об этом Вам сообщает мастер. На самом деле Вам необходимо настроить перенаправление портов на маршрутизаторе. Для этого у маршрутизатора должен быть «белый» **IP** адрес. А на самом сервере лучше настроить статический **IP** адрес.

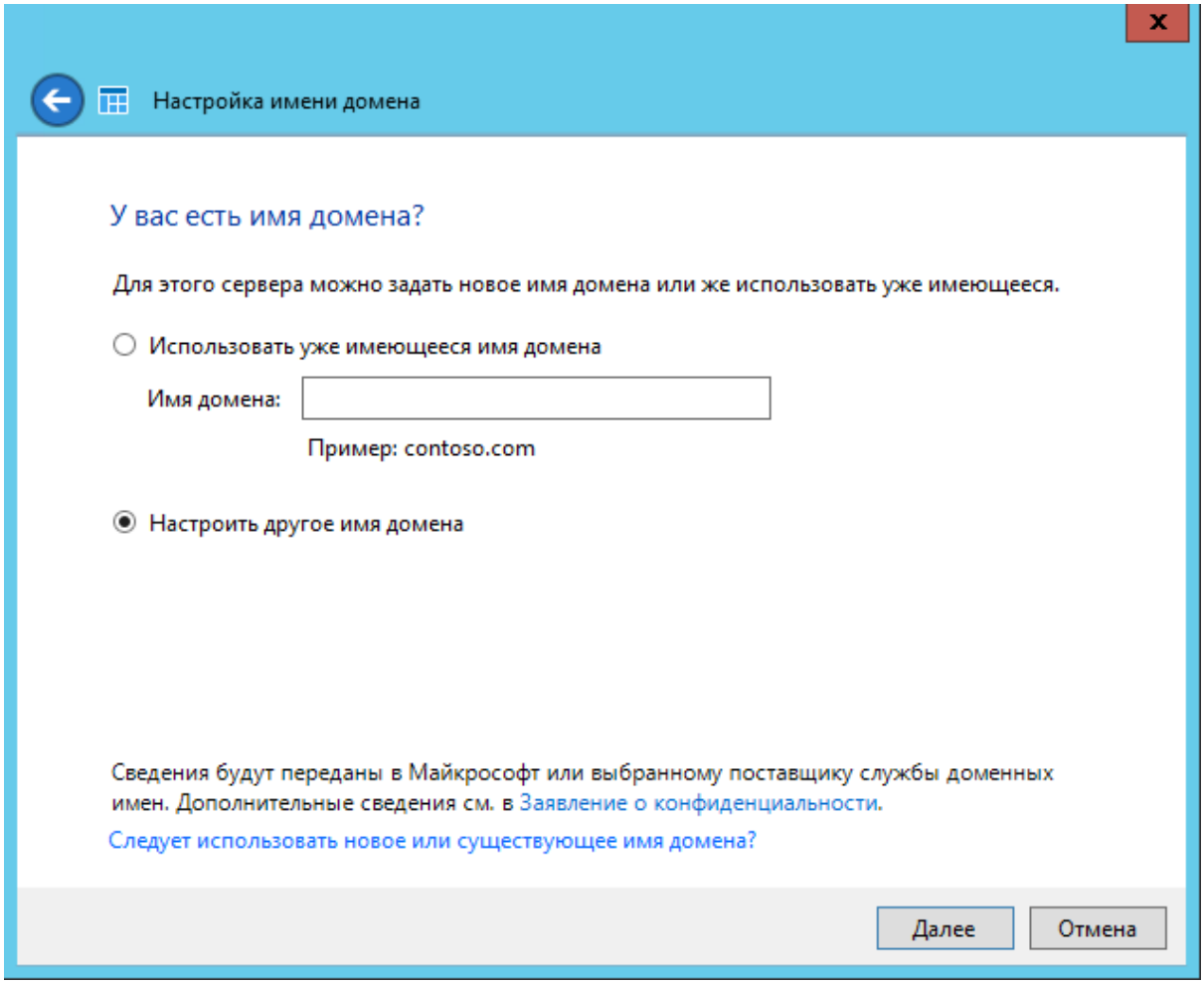
Перенаправить нужно следующие порты **80**, **443**, **1723**, **987** на **IP** адрес вашего сервера. В общем-то процедуру настройки может выполнить и сам мастер, если ваш маршрутизатор поддерживает **UPnP**. Я обычно делаю настройку ручками, поэтому пропускаю данный шаг.



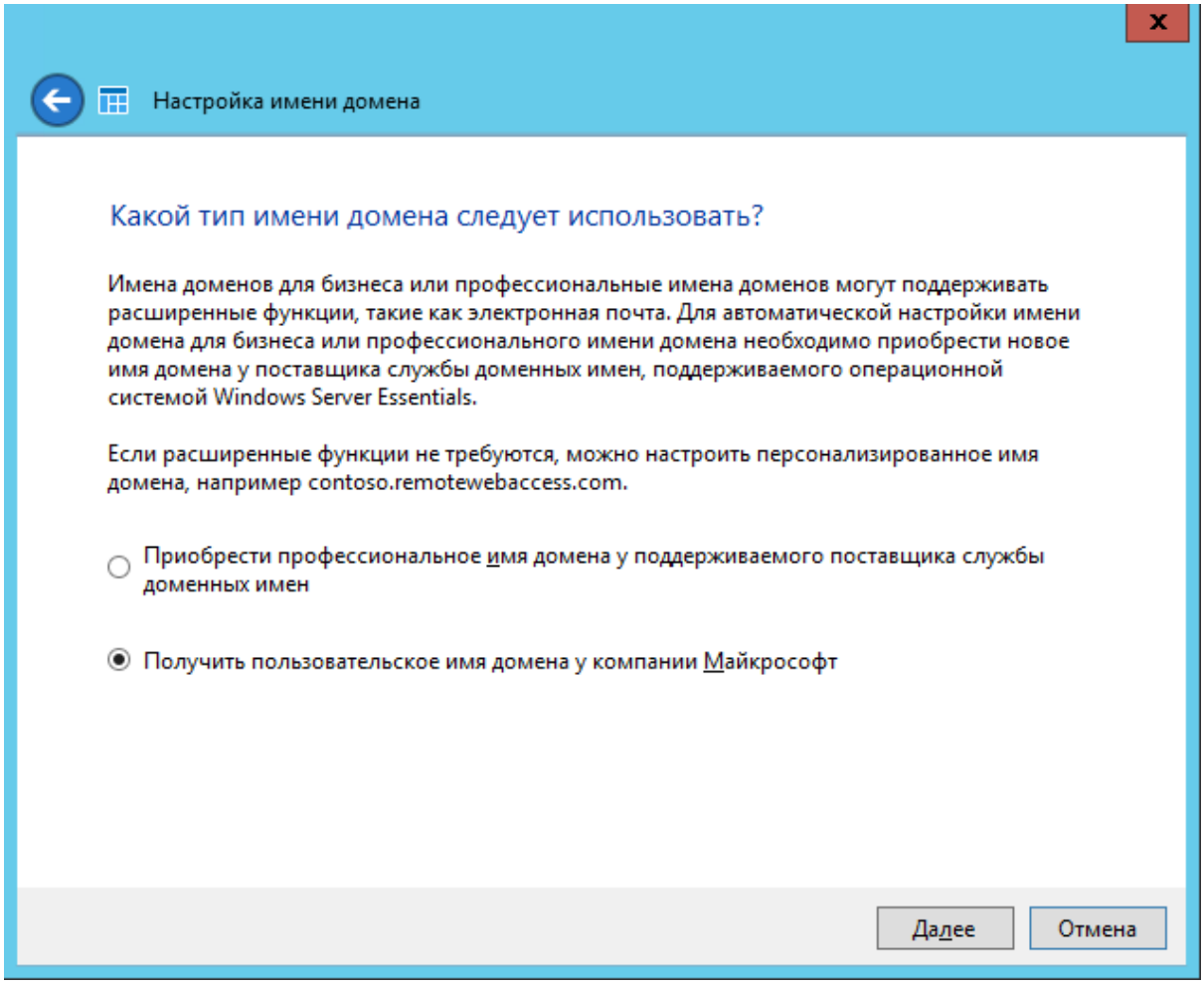
После этого открывается новый мастер настройки доменного имени. Нажимаем далее.



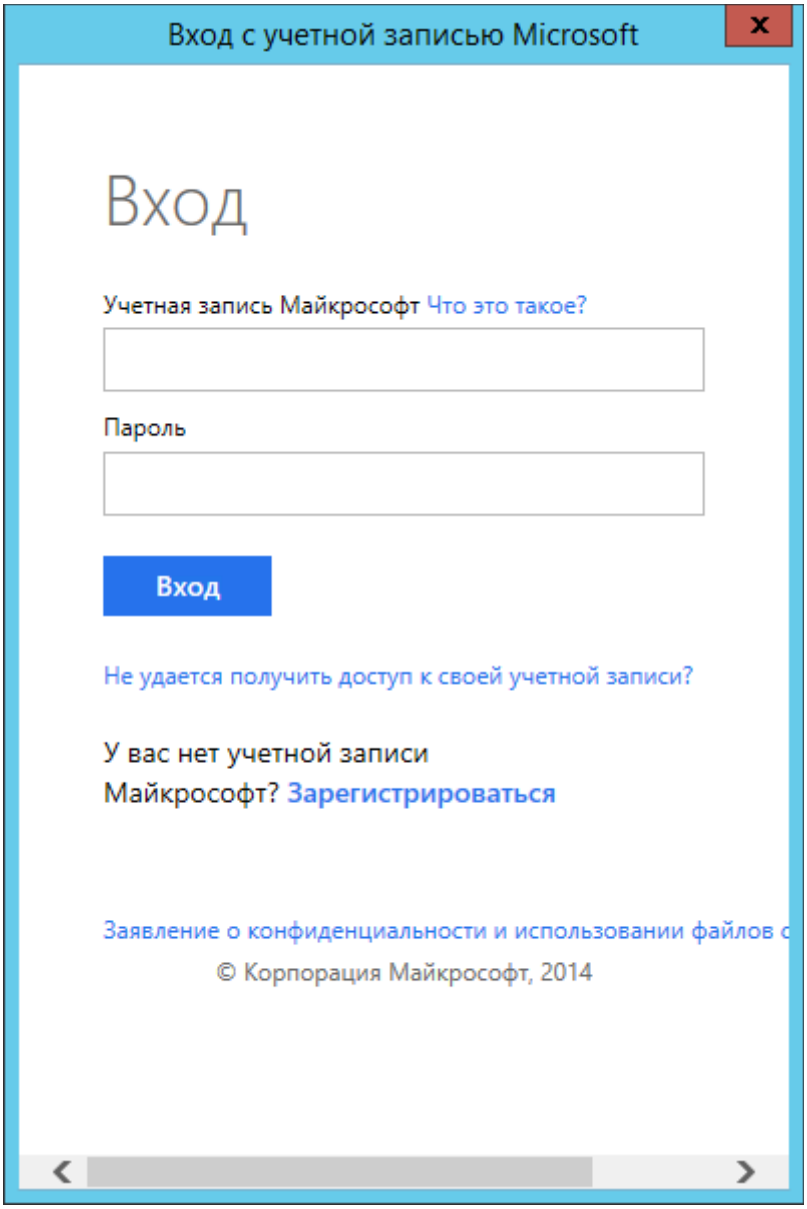
Мастер предложит ввести имя внешнего домена или создать новый. Для собственного домена Вам понадобится сертификат, поэтому рассмотрим тут вариант настройки с использованием домена **Microsoft**. Выбираем другое имя домена и щелкаем далее.



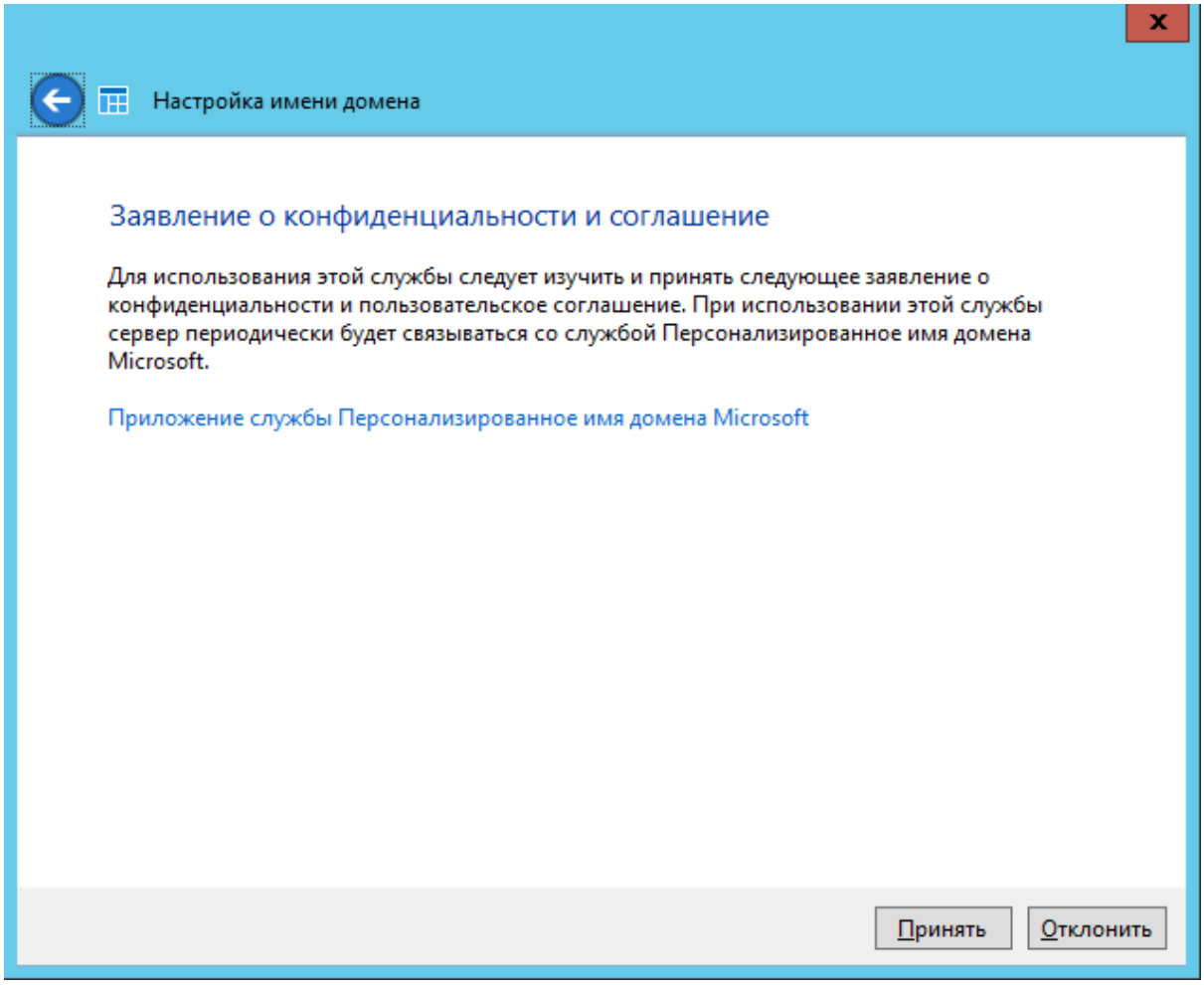
Рассмотрим вариант с доменом компании **Microsoft**.



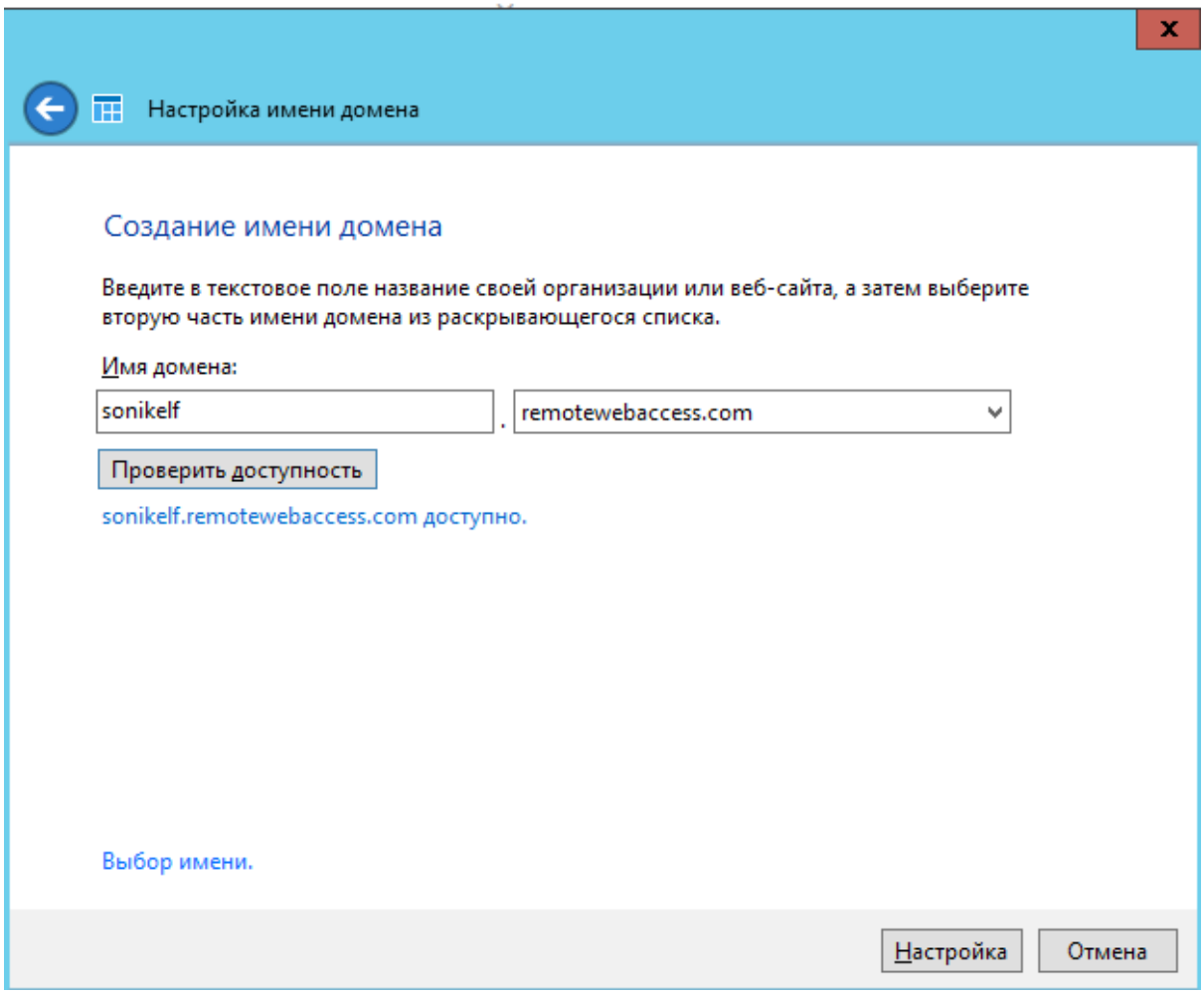
Далее Вас попросят авторизоваться в **Microsoft Account**, если у Вас его нет, то рекомендую зарегистрироваться, ибо по жизни и работе, крайне полезно, т.к майкрософт сейчас крайне мощно объединили свои учетки.



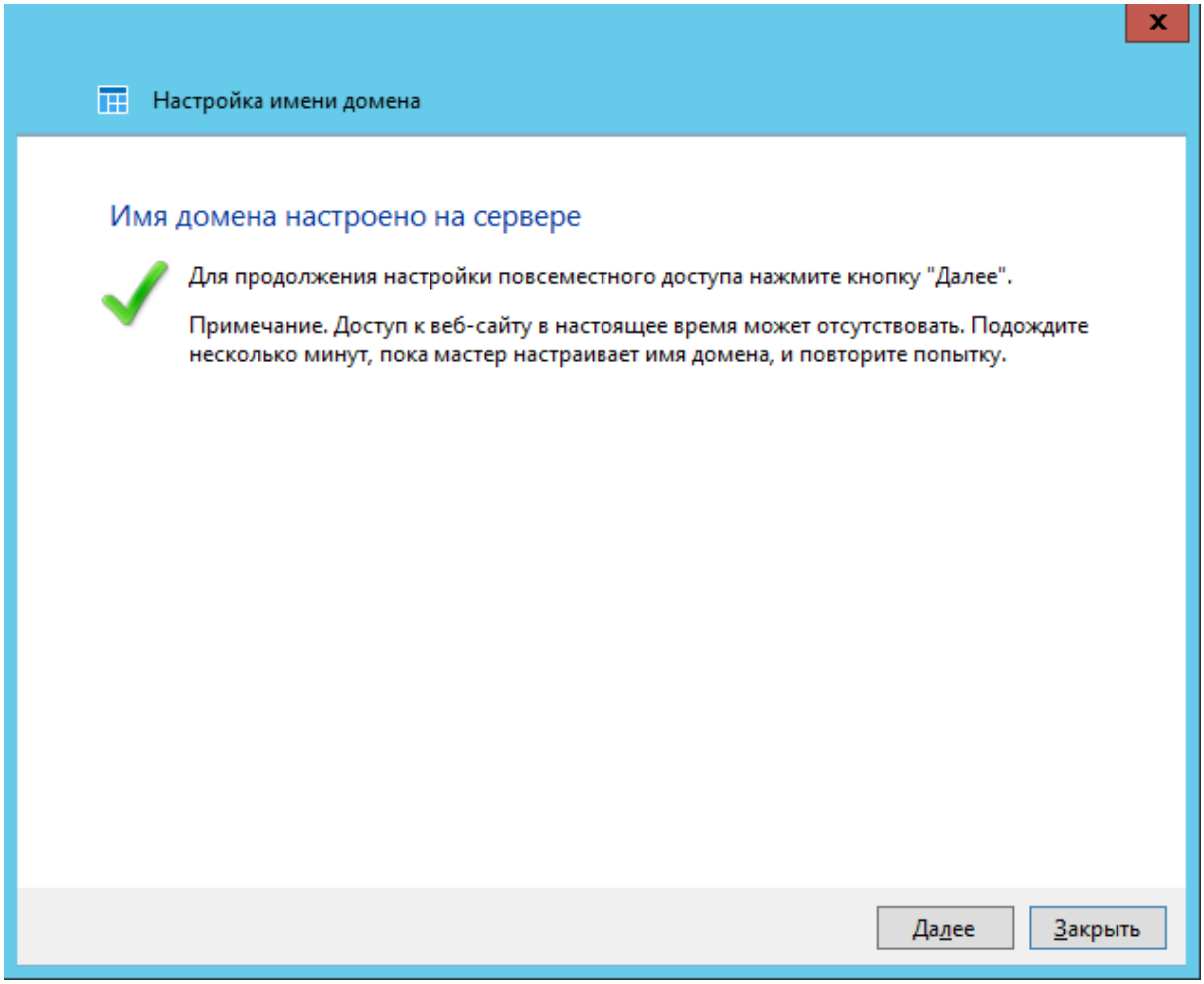
После авторизации принимаем заявление о конфиденциальности.



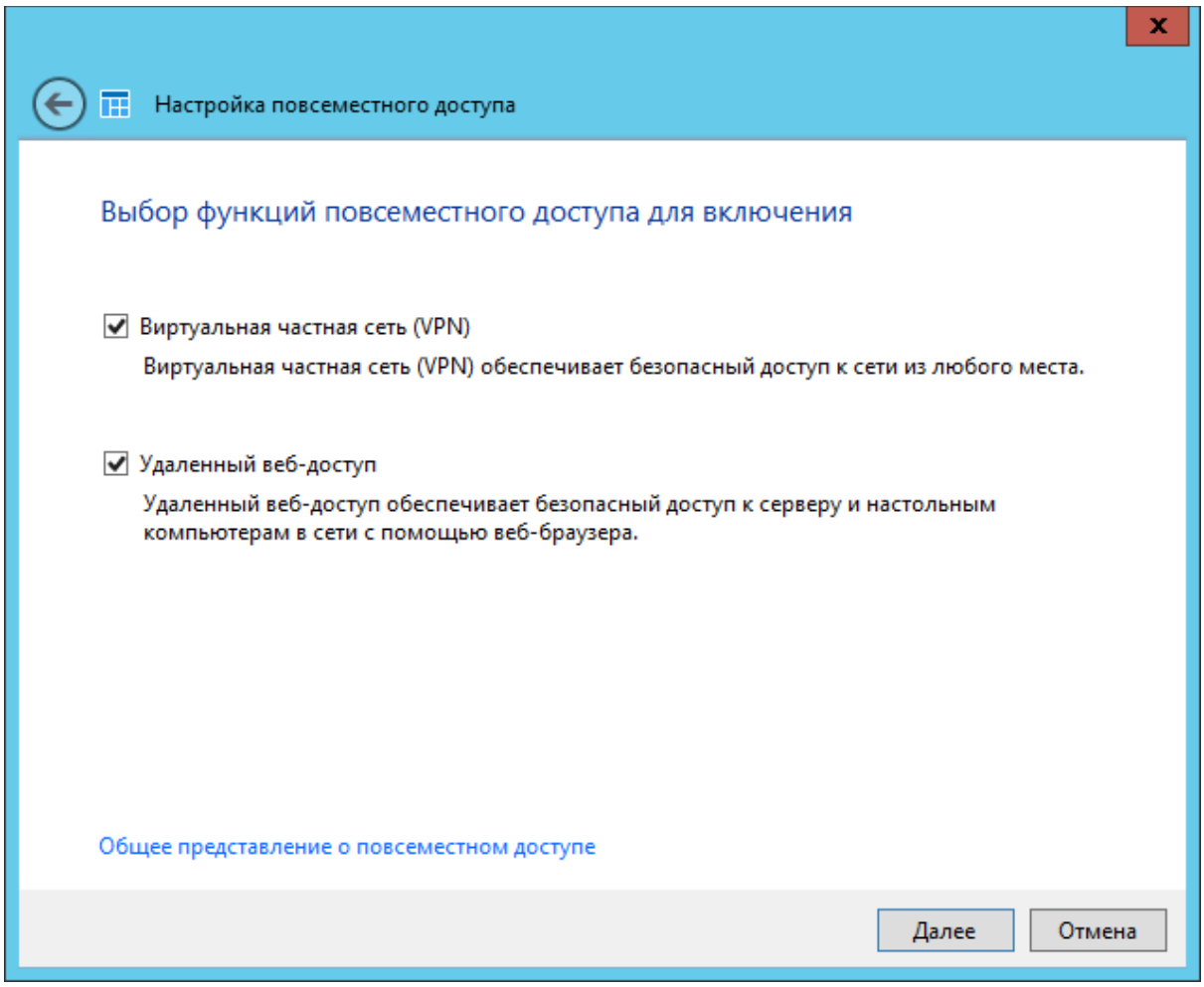
Вводим имя домена и проверяем его доступность, после чего жмем в кнопку “**Настроить**”.



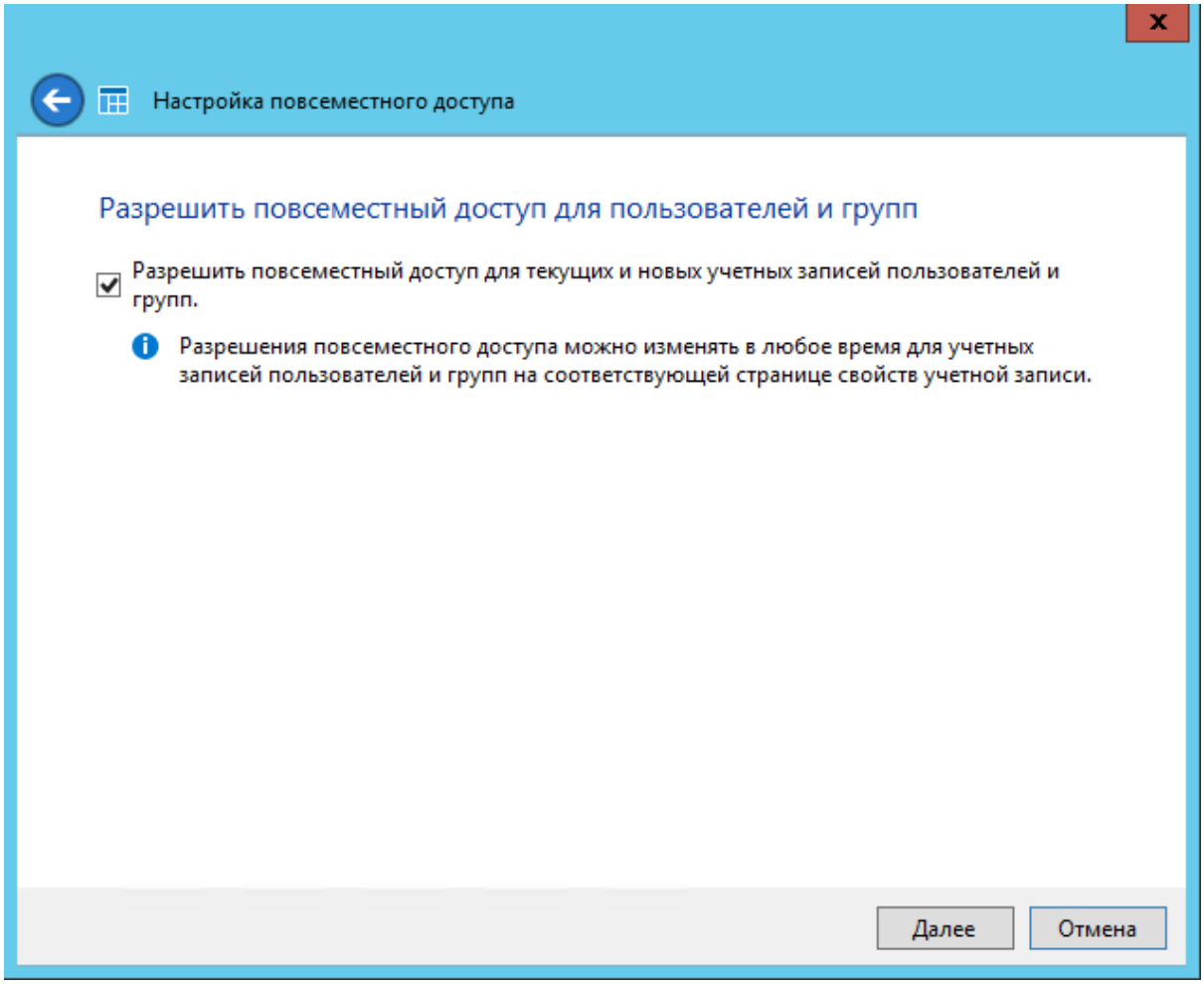
Ну что, с именем домена разобрались. Продолжаем далее.



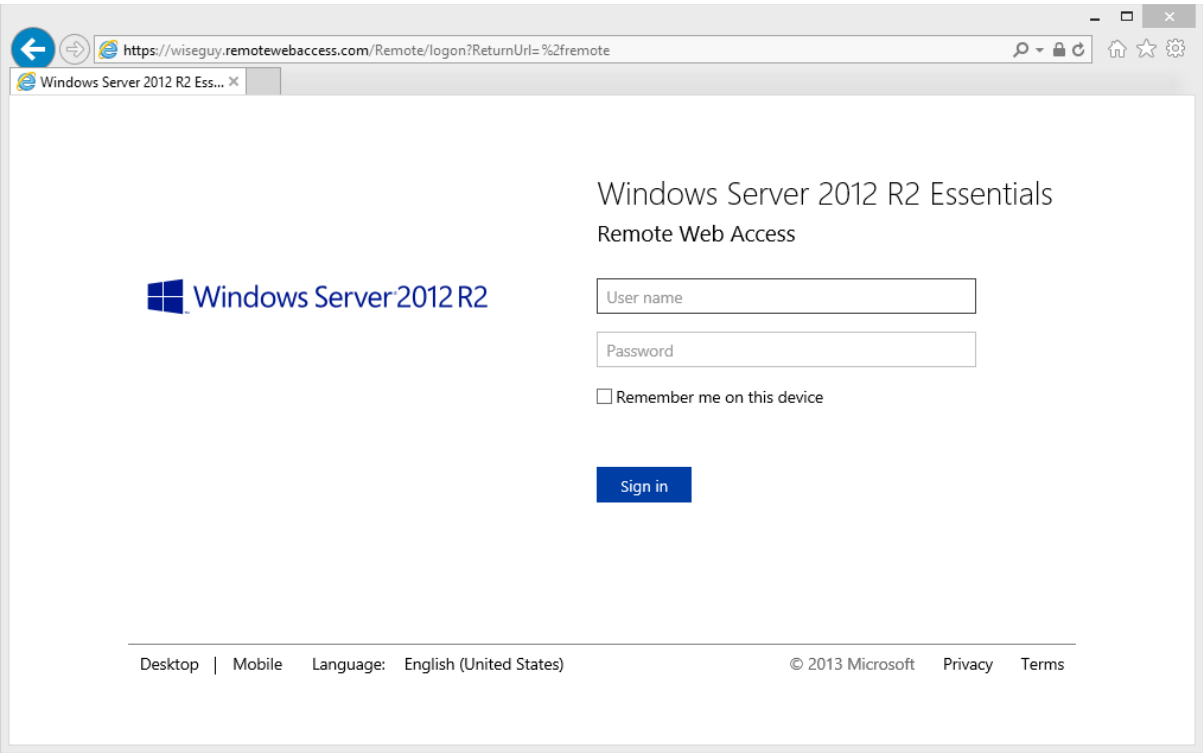
Выбираем, какие именно возможности будут доступны, а именно нужен ли нам **VPN** и **Удаленный веб-доступ**.



Ну а следующим шагом выбираем, будет ли доступен удаленный доступ для текущих пользователей.



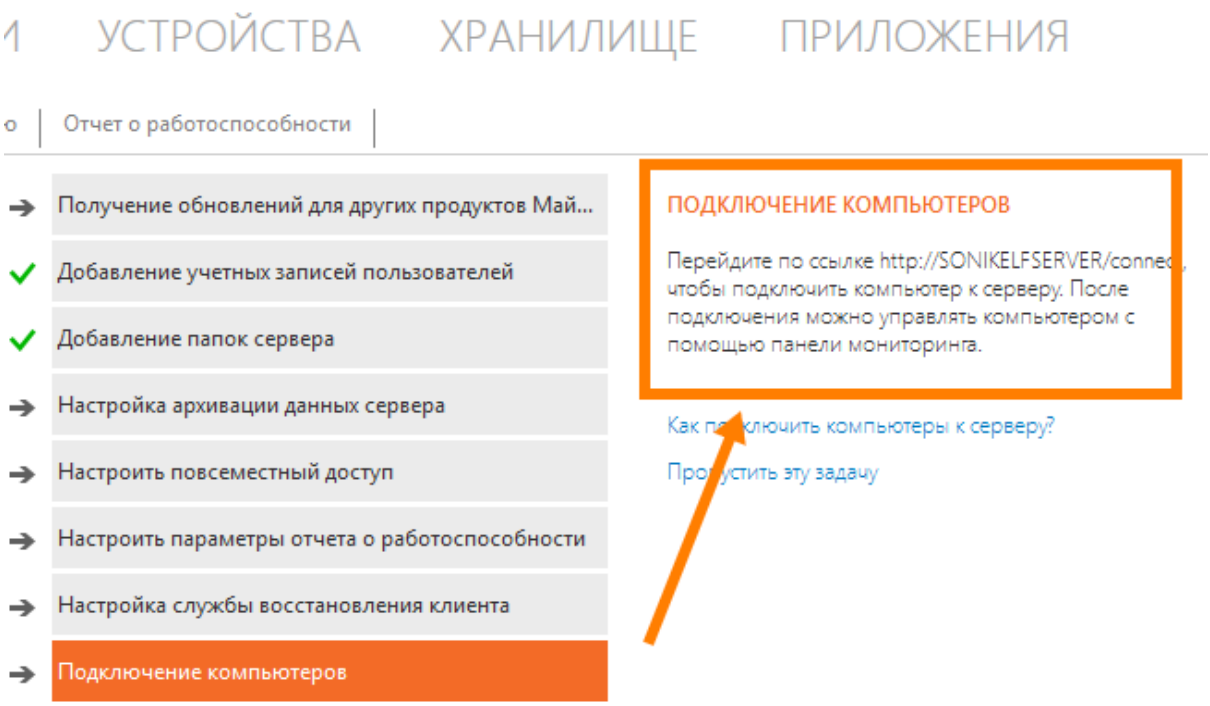
Ну вот и все, можете попробовать зайти на сайт sonikelf.remoteweaccess.com, точнее говоря на тот, который Вы, собственно, создадите.



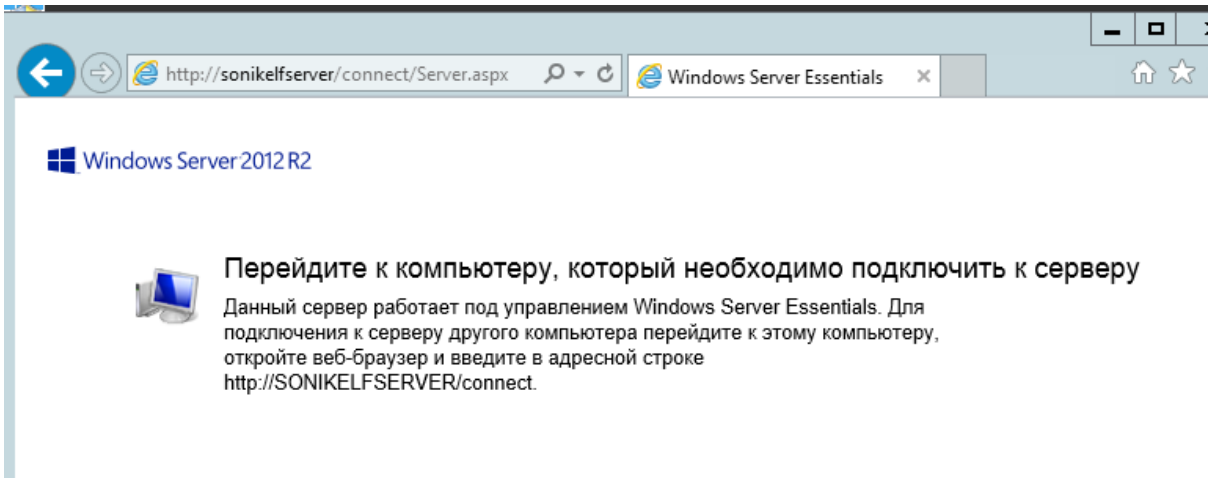
С данного веб-сайта есть возможность доступа к общим папкам и доступ к рабочим столам пользователей. Удобно, не так ли? Я бы сказал, что даже более чем.

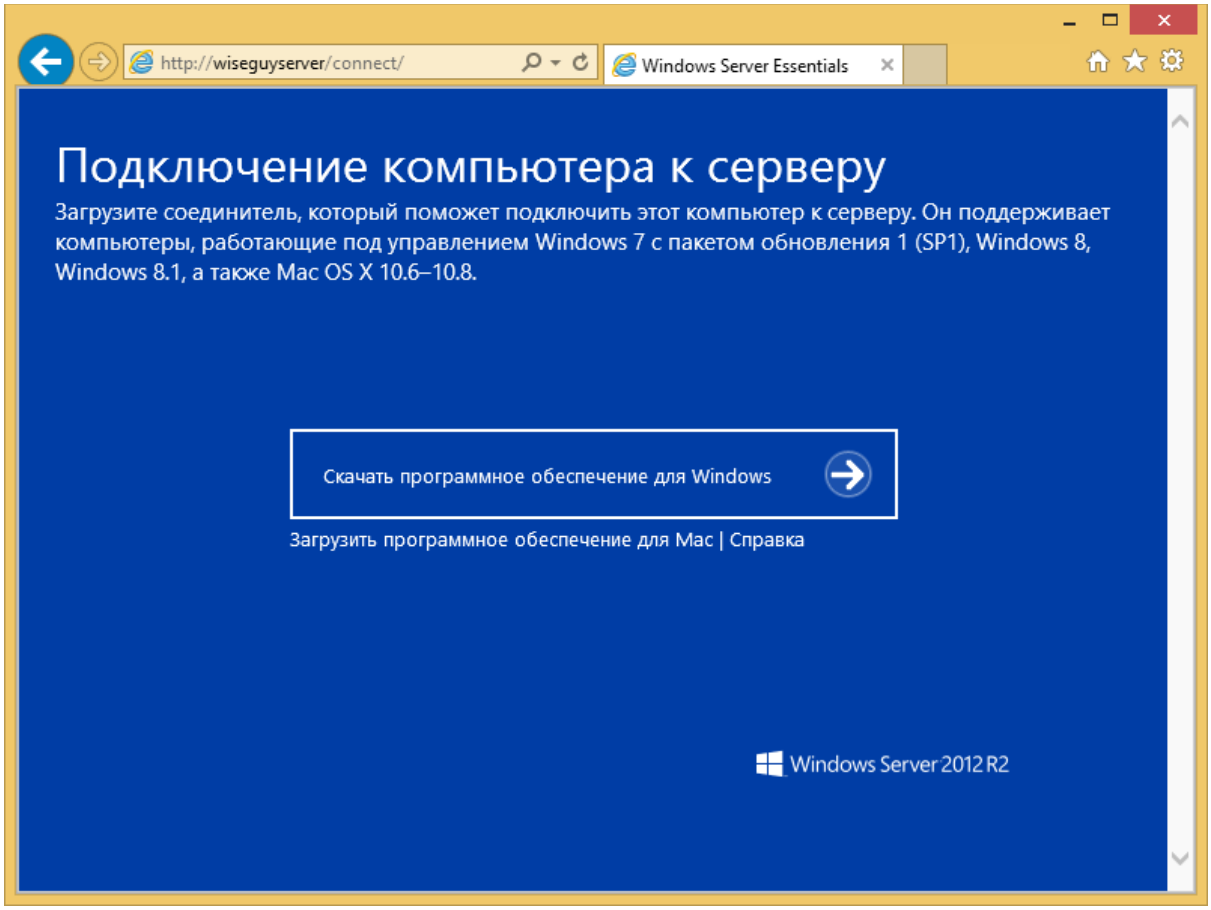
Подключение рабочих станций

Если мы и на этот раз откроем панель мониторинга и перейдем на страницу подключение компьютеров, то увидим там лишь инструкцию к действию

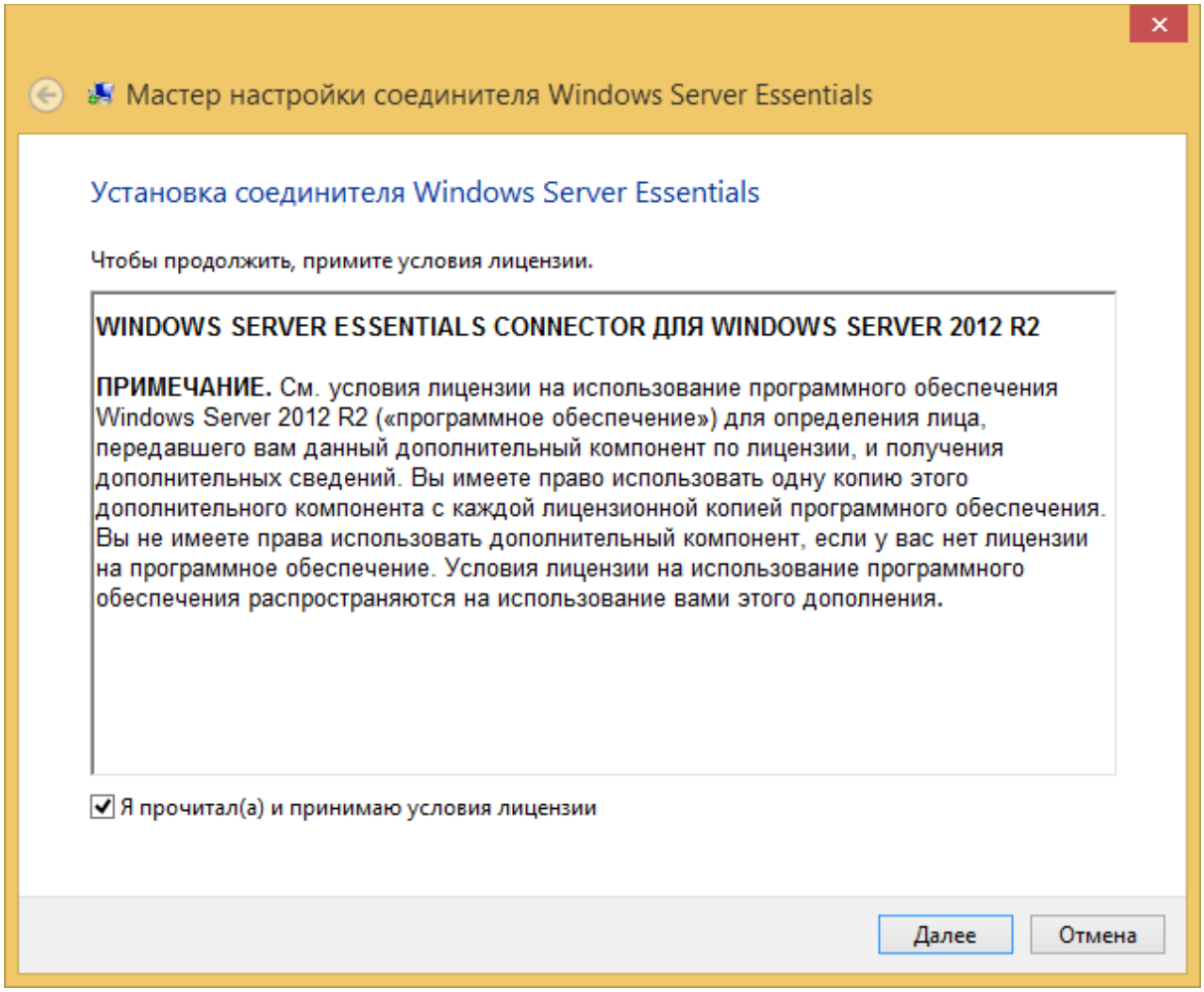


Следуя инструкции, написанной в правом верхнем углу, собственно, открываем страничку “**http://<Имя сервера>/connect**” и нажимаем ссылку для скачивания.

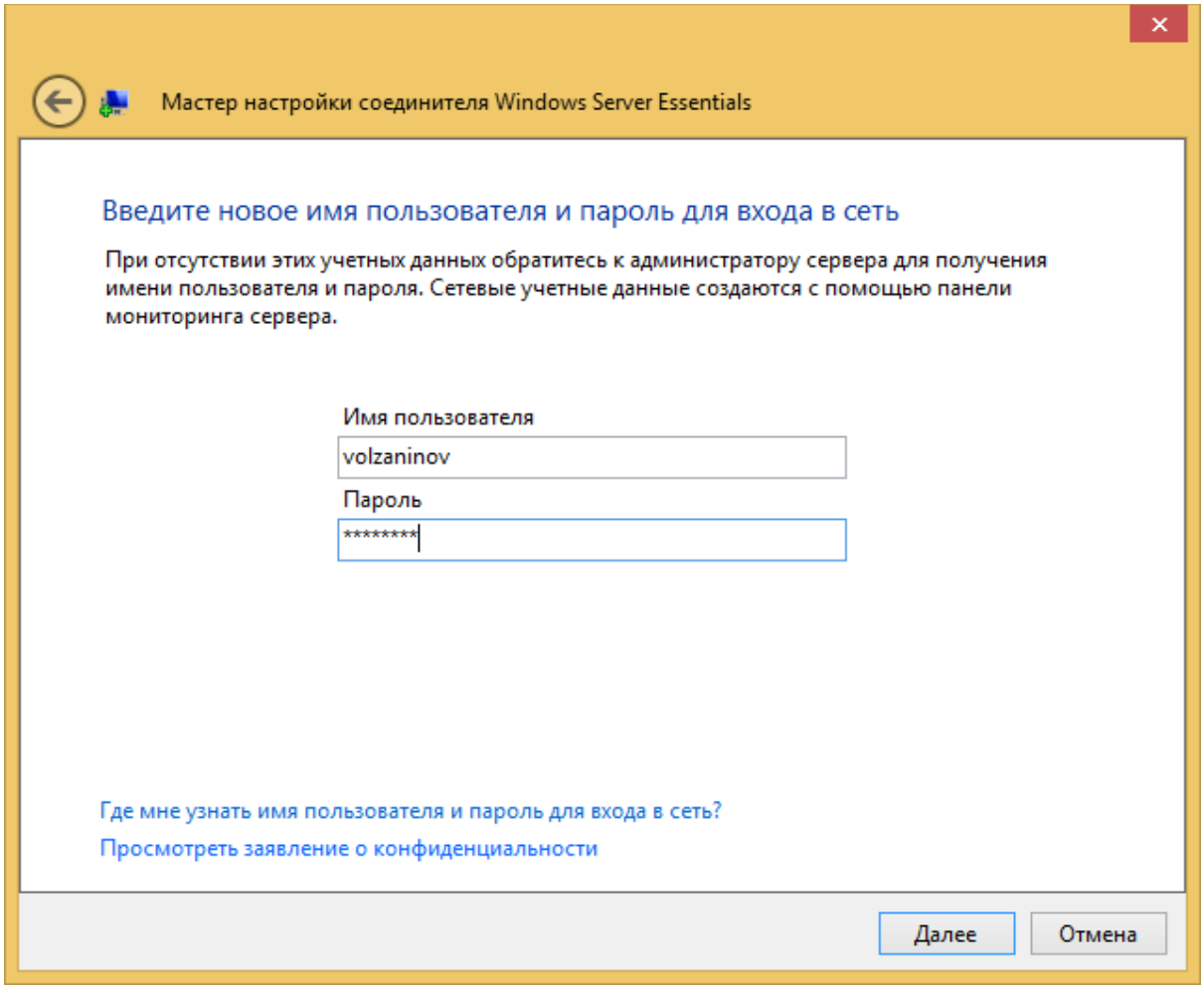




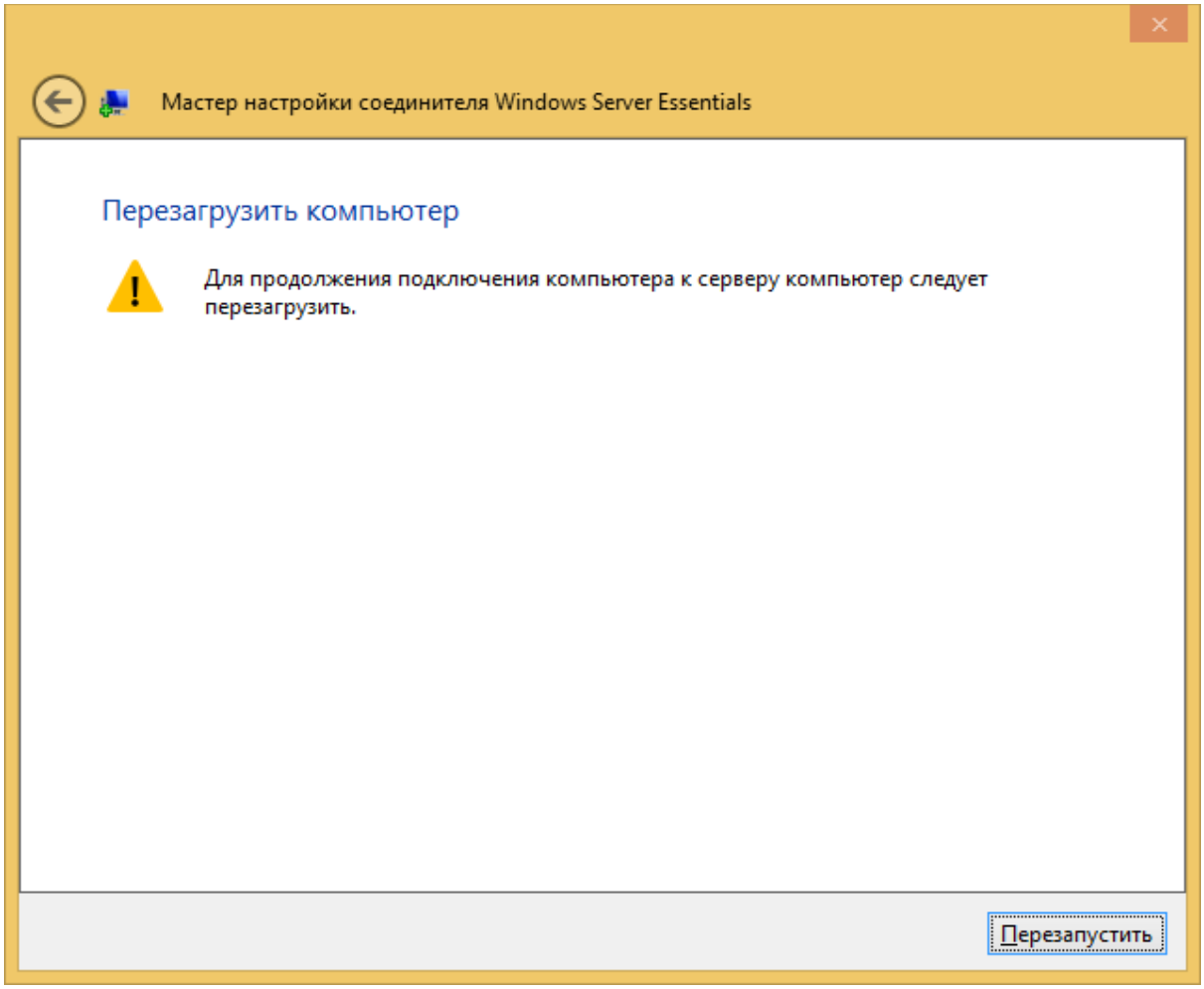
Следуя инструкциям, сохраняем и запускаем файл или просто скачиваем и запускаем, после чего принимаем лицензию и ждем.



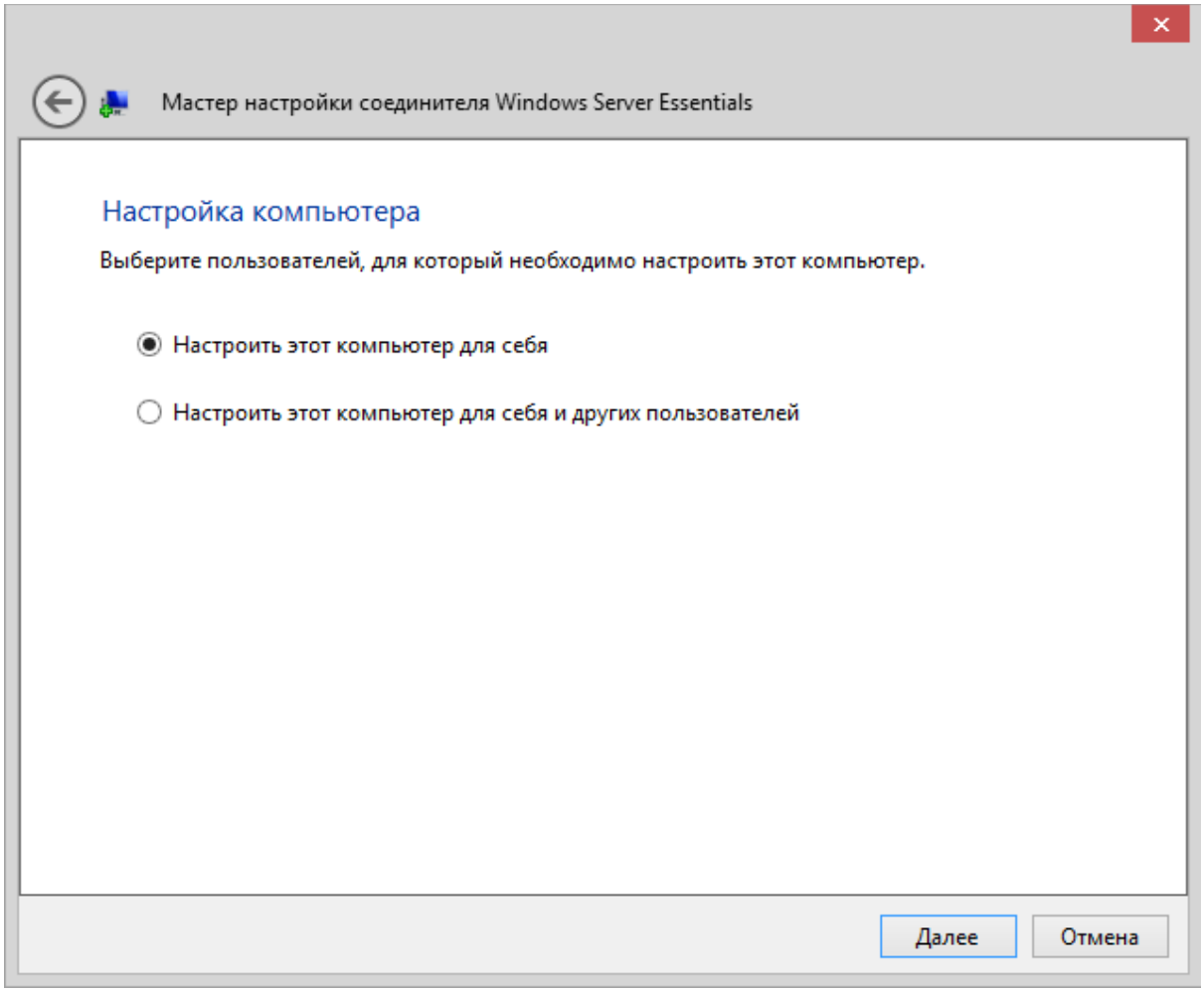
Вводим имя пользователя и пароль пользователя данного компьютера или администратора, либо учетную запись пользователя:



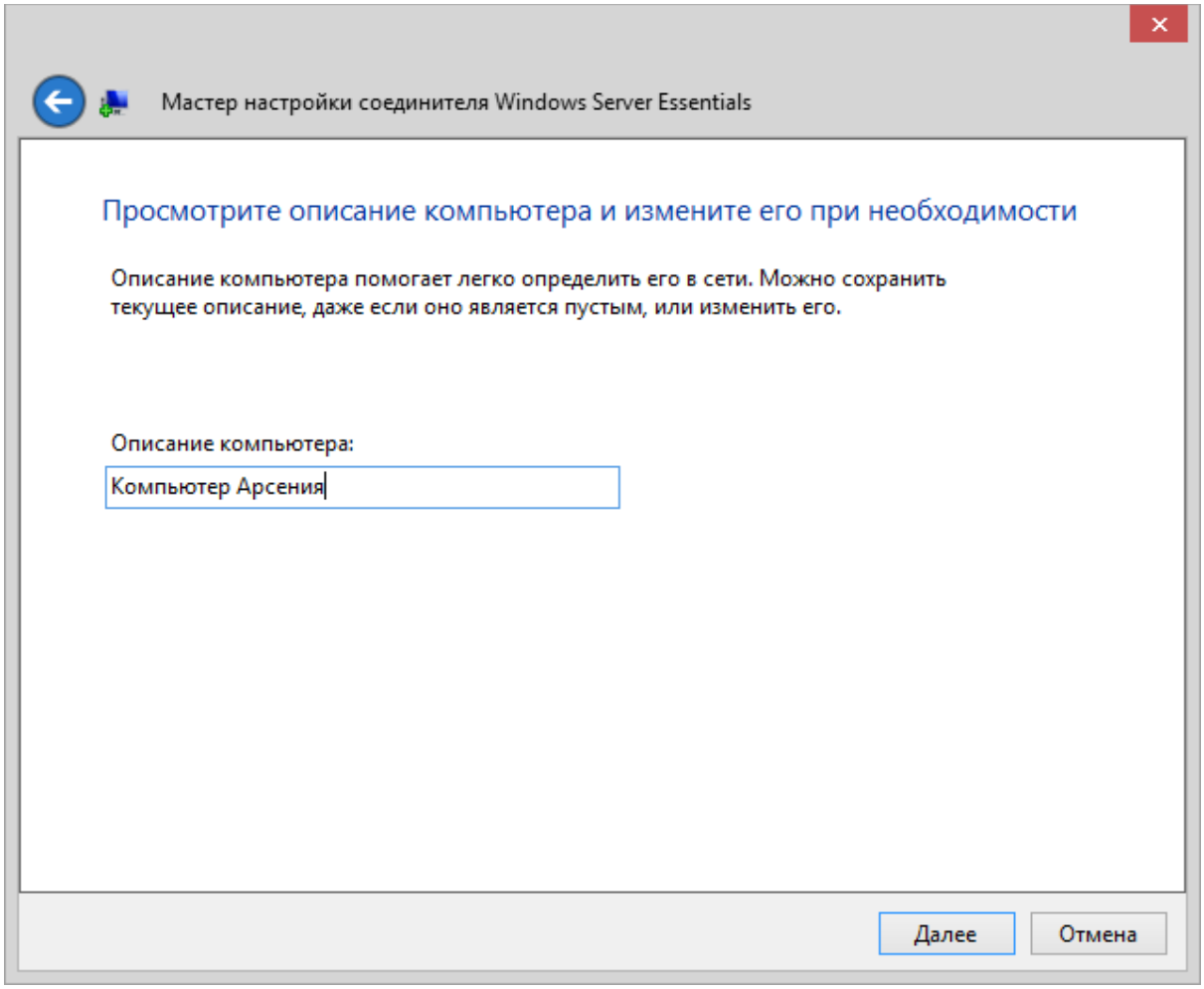
После чего решительно перезагружаем сервер:



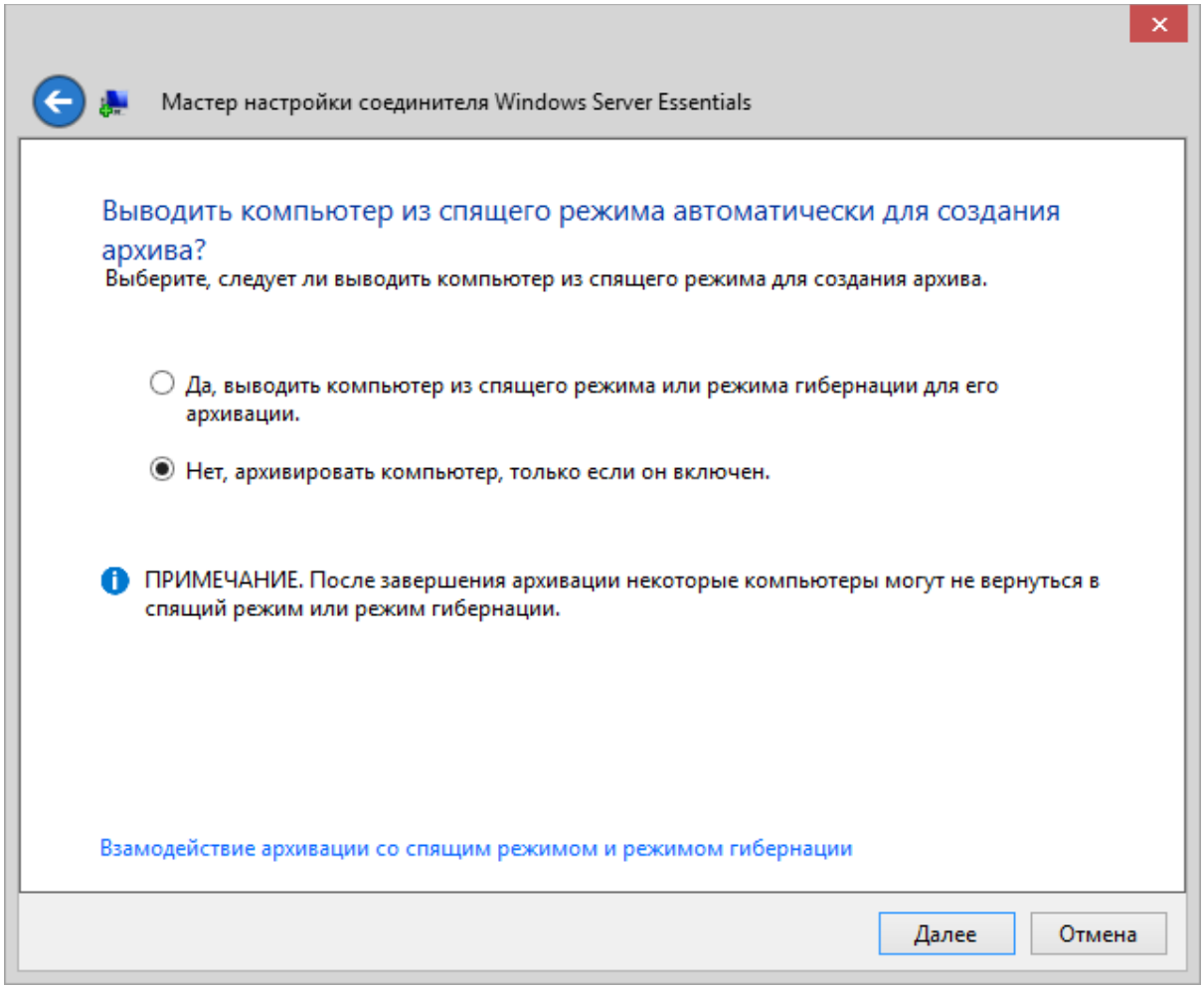
Выбираем, кто будет пользоваться компьютером.



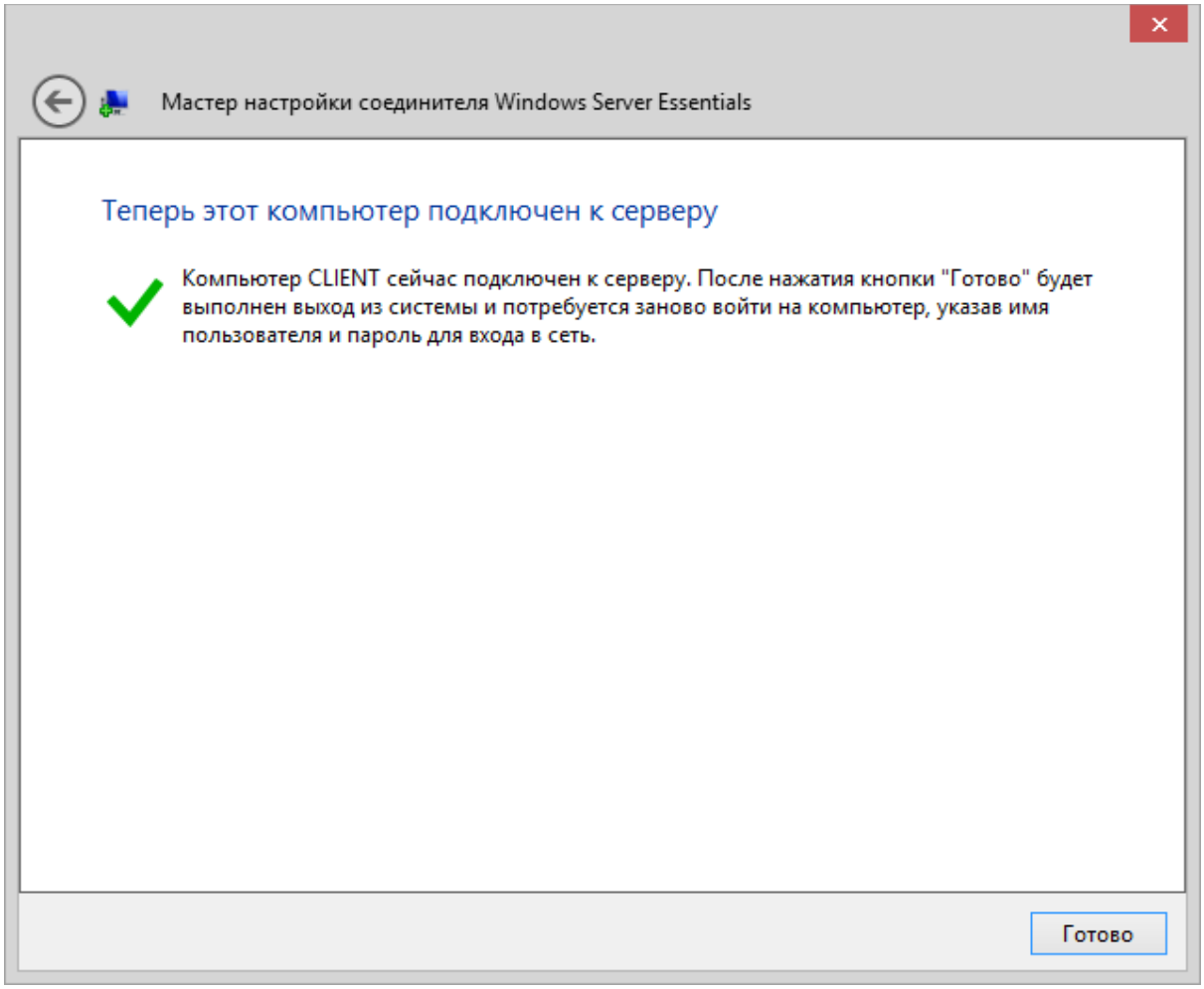
Вводим имя компьютера, желательно делать его понятным, очевидным и тп, чтобы потом Вы могли сами определить его в сети.



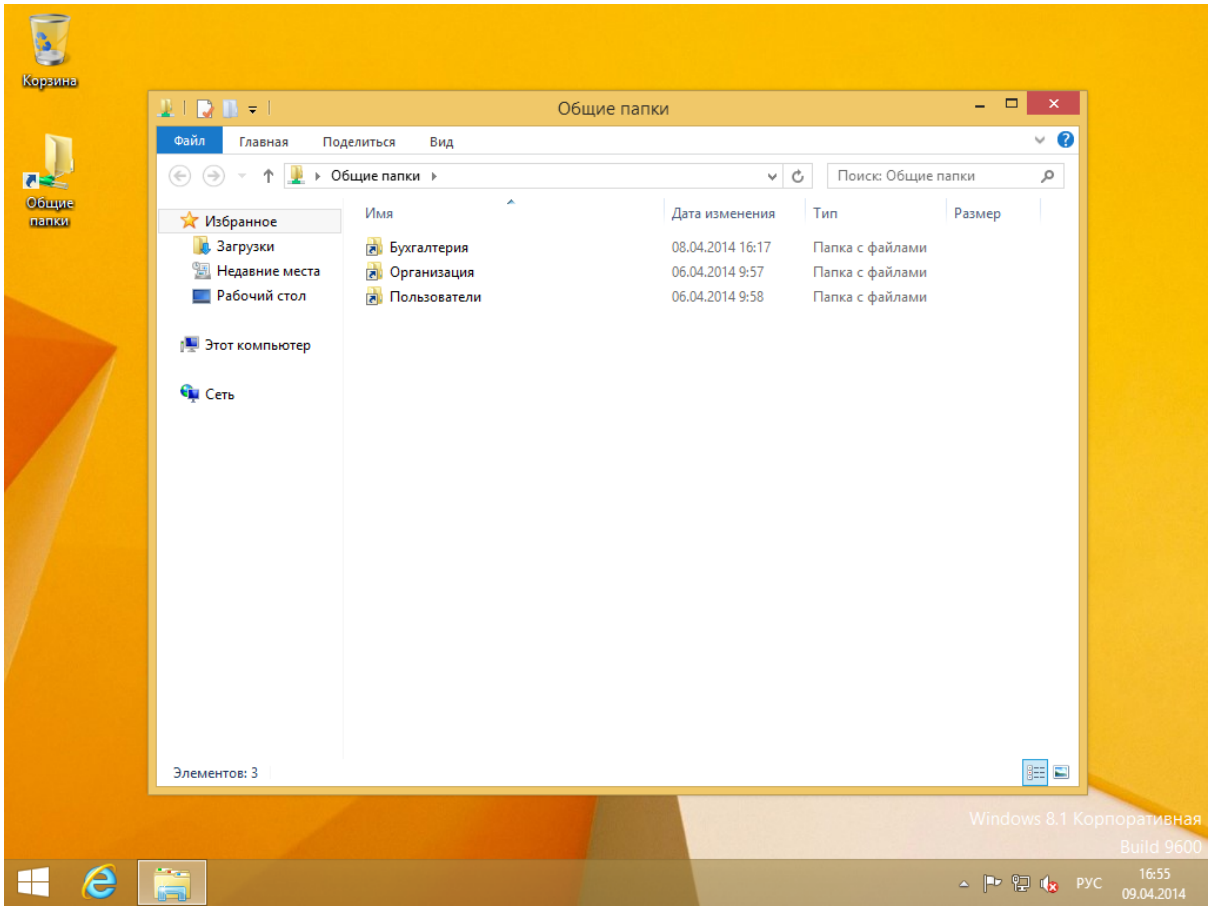
Следующим шагом задаём параметры архивации. Достаточно полезная штука.



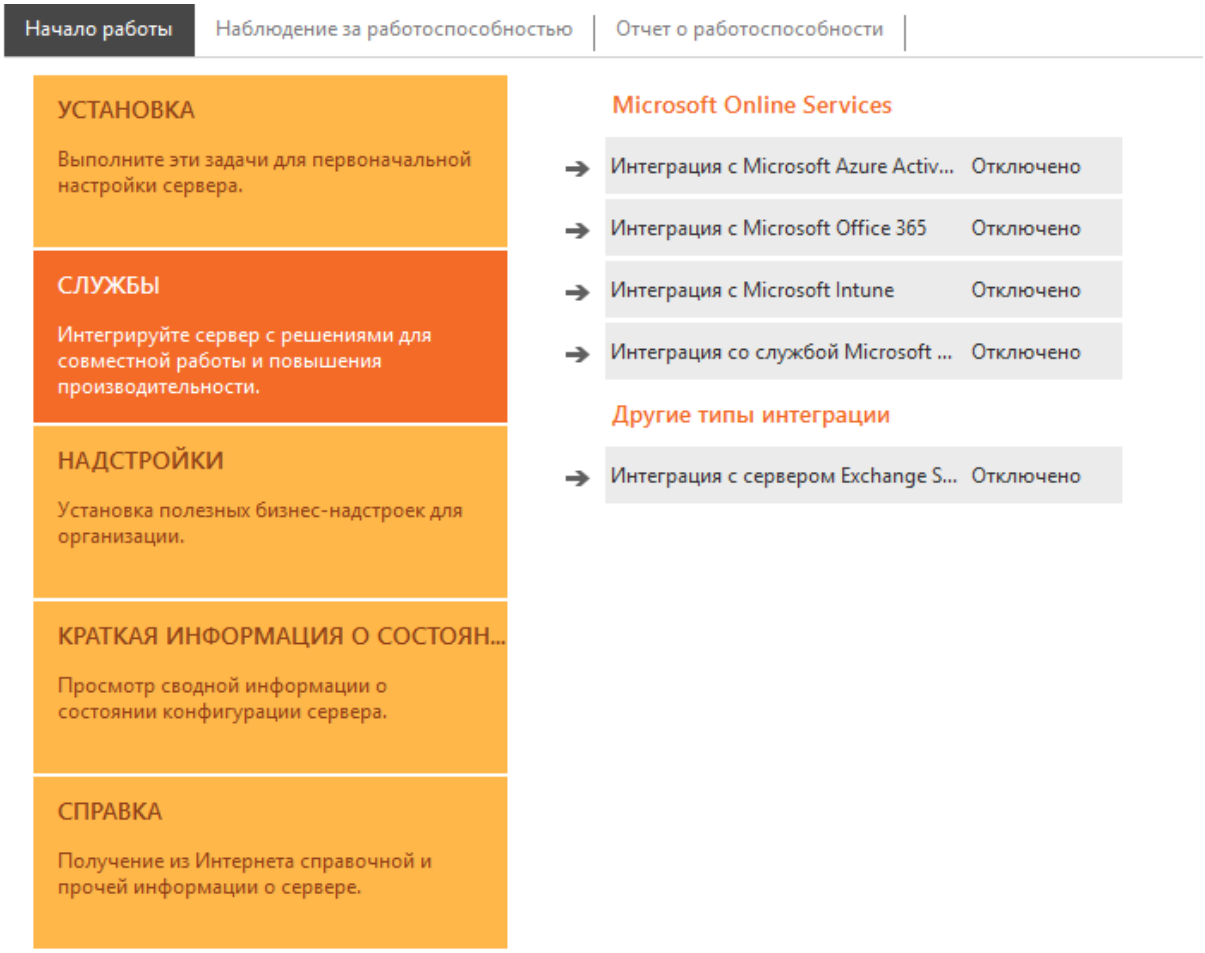
Ну и, собственно, логическое завершение:



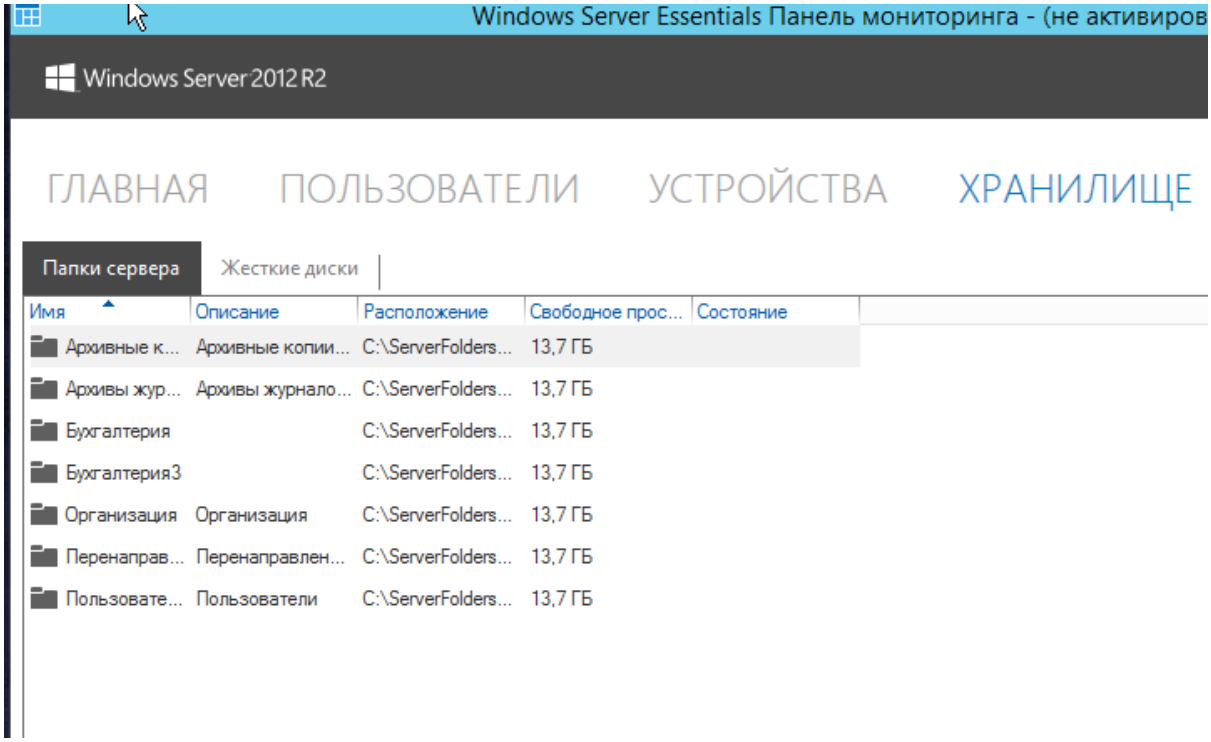
Далее просто заходим под учетной записью пользователя и можно работать. На рабочем столе уже есть все необходимые ярлыки, которые мы создавали ранее (точнее мы создавали не ярлыки, а общие папки сервера).

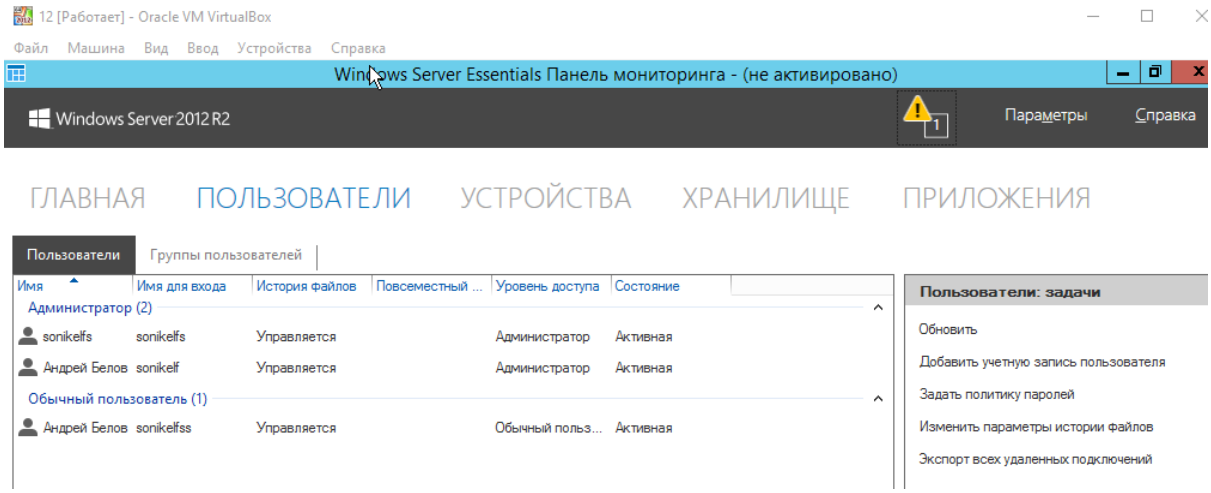


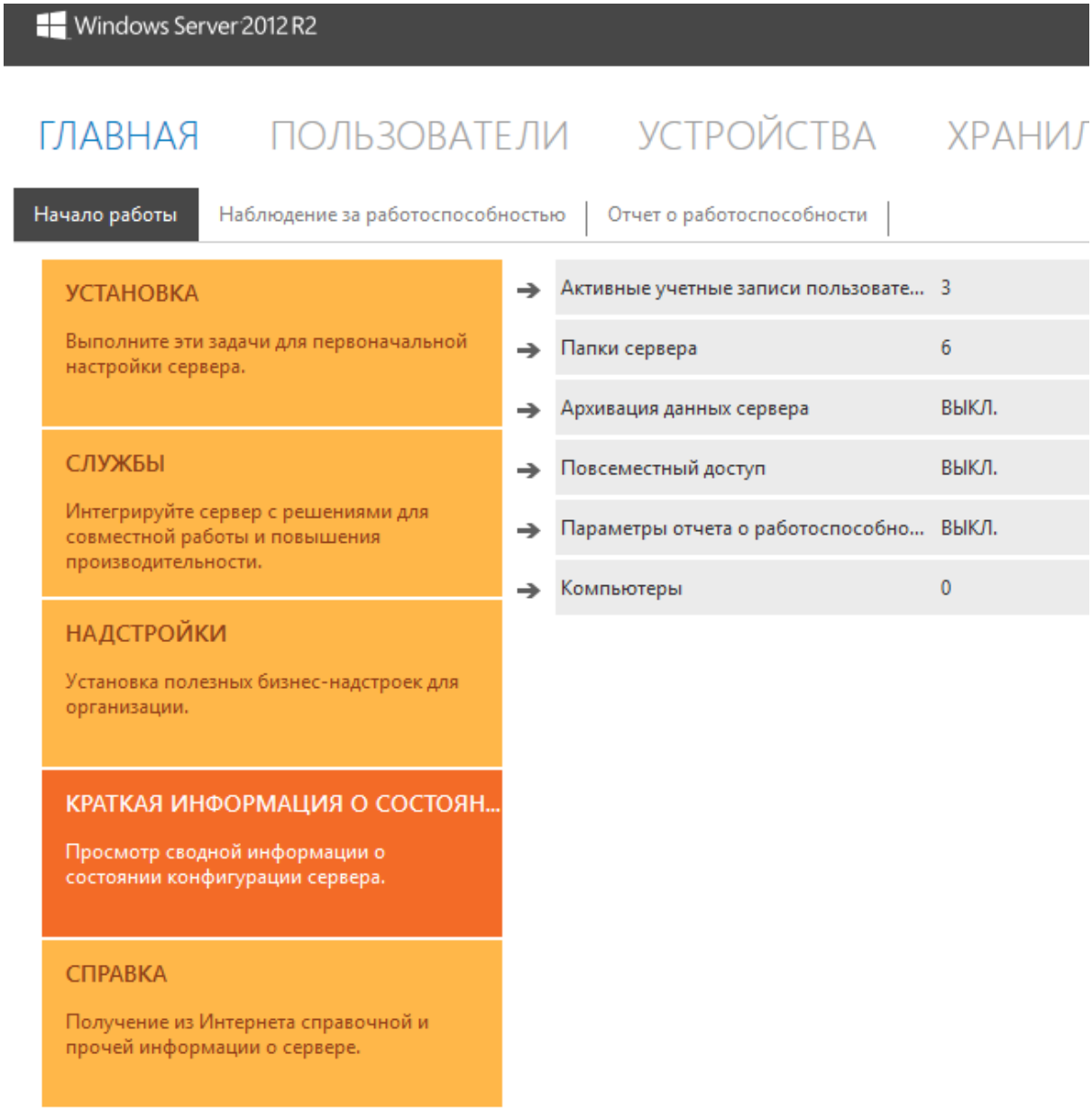
Дальнейшая настройка сервера происходит через ту же самую панель по схожему принципу.



Из этой же панели мониторинга.. Осуществляется мониторинг и управление всем на свете, будь то пользователи, устройства, хранилища, надстройки и прочее:


Windows Server Essentials Панель мониторинга - (не активиров
Windows Server 2012 R2
ГЛАВНАЯ
ПОЛЬЗОВАТЕЛИ
УСТРОЙСТВА
ХРАНИЛИЩЕ
Папки сервера
Жесткие диски
Имя
Описание
Расположение
Свободное прос...
Состояние
Архивные к...
Архивные копии...
C:\ServerFolders...
13,7 ГБ
Архивы жур...
Архивы журнало...
C:\ServerFolders...
13,7 ГБ
Бухгалтерия
C:\ServerFolders...
13,7 ГБ
Бухгалтерия3
C:\ServerFolders...
13,7 ГБ
Организация
Организация
C:\ServerFolders...
13,7 ГБ
Перенаправ...
Перенаправлен...
C:\ServerFolders...
13,7 ГБ
Пользовате...
Пользователи
C:\ServerFolders...
13,7 ГБ


12 [Работает] - Oracle VM VirtualBox
Файл
Машина
Вид
Ввод
Устройства
Справка
Windows Server Essentials Панель мониторинга - (не активировано)
Windows Server 2012 R2
Параметры
Справка
ГЛАВНАЯ
ПОЛЬЗОВАТЕЛИ
УСТРОЙСТВА
ХРАНИЛИЩЕ
ПРИЛОЖЕНИЯ
Пользователи
Группы пользователей
Имя
Имя для входа
История файлов
Повсеместный ...
Уровень доступа
Состояние
Администратор (2)
sonikelfs
sonikelfs
Управляется
Администратор
Активная
Андрей Белов
sonikelf
Управляется
Администратор
Активная
Обычный пользователь (1)
Андрей Белов
sonikelfss
Управляется
Обычный польз...
Активная
Пользователи: задачи
Обновить
Добавить учетную запись пользователя
Задать политику паролей
Изменить параметры истории файлов
Экспорт всех удаленных подключений



В рамках первой части серии книг, описывать всё это не представляется возможным, но так как всё на русском, просто и удобно, то, думаю, Вы легко разберетесь сами (или дождитесь следующих частей книги ;)).

Active Directory - что к чему и почему

Небольшая вводная

Возможно, ни один из механизмов **Windows**-окружения серверной части не вызывает столько беспокойства и разочарований, не отнимает столько времени **IT**-специалистов, как **Active Directory (AD)**.

Active Directory впервые была представлена в **Windows Server 2000** и входила во все последующие выпуски **Windows Server**. В **Windows Server 2012** включена самая совершенная на текущий момент версия **Active Directory**.

Одна из важнейших причин, по которой данная версия **AD** может считаться лучшей, — это развертывание доменных служб **Active Directory (Active Directory Domain Services, AD DS)**, содержащее теперь в едином интерфейсе новый мастер установки доменных служб **Active Directory (Active Directory Domain Services Wizard)**, все шаги которого необходимы для развертывания нового контроллера домена. Кроме того, **AD DS** в **Server 2012** легче управлять благодаря расширенному мастеру, построенному на основе **PowerShell**, который интегрирован с диспетчером серверов. Проверка перед установкой, выполняемая с помощью инструмента **Adprep.exe**, и предварительная проверка системных требований являются частью процесса установки, что

помогает уменьшить вероятность возникновения ошибок при развертывании **AD DS**.

Означают ли эти улучшения, что развертывание и управление **AD** в **Server 2012** стало совершенно безупречным? По некоторым причинам — нет. Развертывая **AD** в **Server 2012** впервые, я сталкивался с ошибками и предупреждениями, говорящими о необходимости добавления некоторых компонентов или об особой настройке тех или иных механизмов, необходимых для успешной установки **AD**. Однако эти уведомления и сообщения об ошибках отображались автоматически, мне не пришлось охотиться за ними, используя средство “Просмотр событий” (**Event Viewer**) для поиска предупреждений, имеющих отношение к **AD**, как обычно приходилось делать в **Server 2008 R2** и более старых версиях **Windows Server**.

В процессе установки в обновленном диспетчере серверов выводятся предупреждения, сообщения об ошибках и советы по подготовке сервера к запуску **AD**. Весь процесс развертывания **AD** сосредоточен в диспетчере серверов, что повышает эффективность работы и упрощает решение проблем. В этой главе вы узнаете о новом интерфейсе центра администрирования **Active Directory (Active Directory Administrative Center, ADAC)**. Кроме того, я расскажу о развертывании **AD DS** и об управлении этой службой, о том, как подключить **Server 2012** к существующему домену и,

наоборот, подключить компьютер под управлением **Server 2012** к домену **Server 2008 R2**.

Поскольку в **Server 2012** обновлен интерфейс диспетчера серверов и средств управления **Active Directory**, процесс установки окажется новым даже для тех, кто раньше уже устанавливал доменные службы **Active Directory (Active Directory Domain Services, AD DS)**.

Давайте приступим.

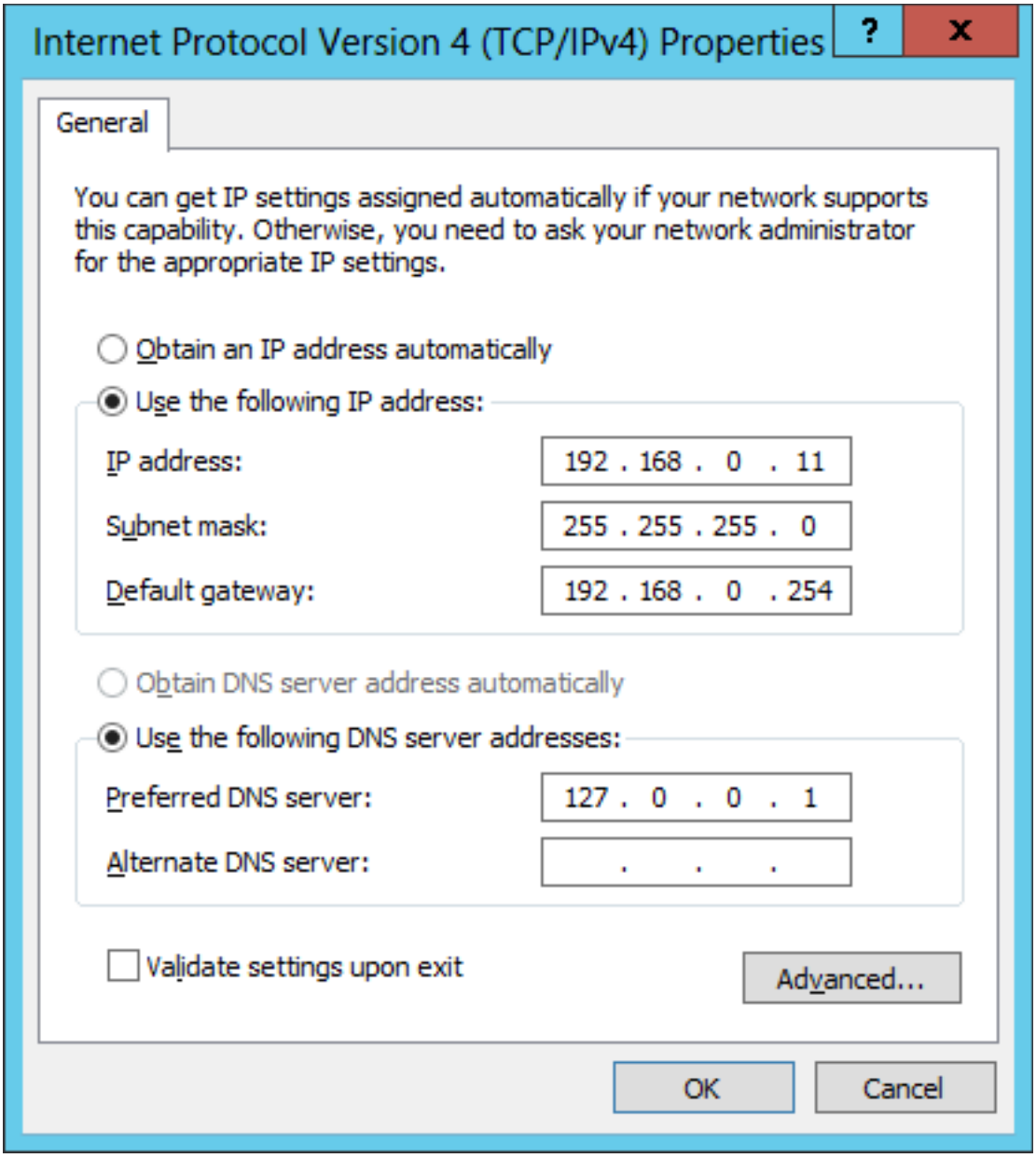
Предварительная настройка сервера для установки AD

Прежде, чем настраивать роль **Active Directory** необходимо произвести настройку **Windows Server 2012** — задать статический **IP** адрес и переименовать компьютер. Я рассмотрю здесь английскую версию, во-первых, потому, что на момент написания, не нашлось под рукой русскоязычного сервера, а, во-вторых, потому, что чаще всего многим из Вас придется работать именно с английской версией. И это хорошо. Итаааак..

Чтобы установить статический **IP** адрес, необходимо щелкнуть правой кнопкой мышки по иконке “**Network**“ в панели задач и выбрать “**Open Network ang Sharing Center** -> **Change adapter settings**“. Выбрать адаптер, который смотрит во внутреннюю сеть.

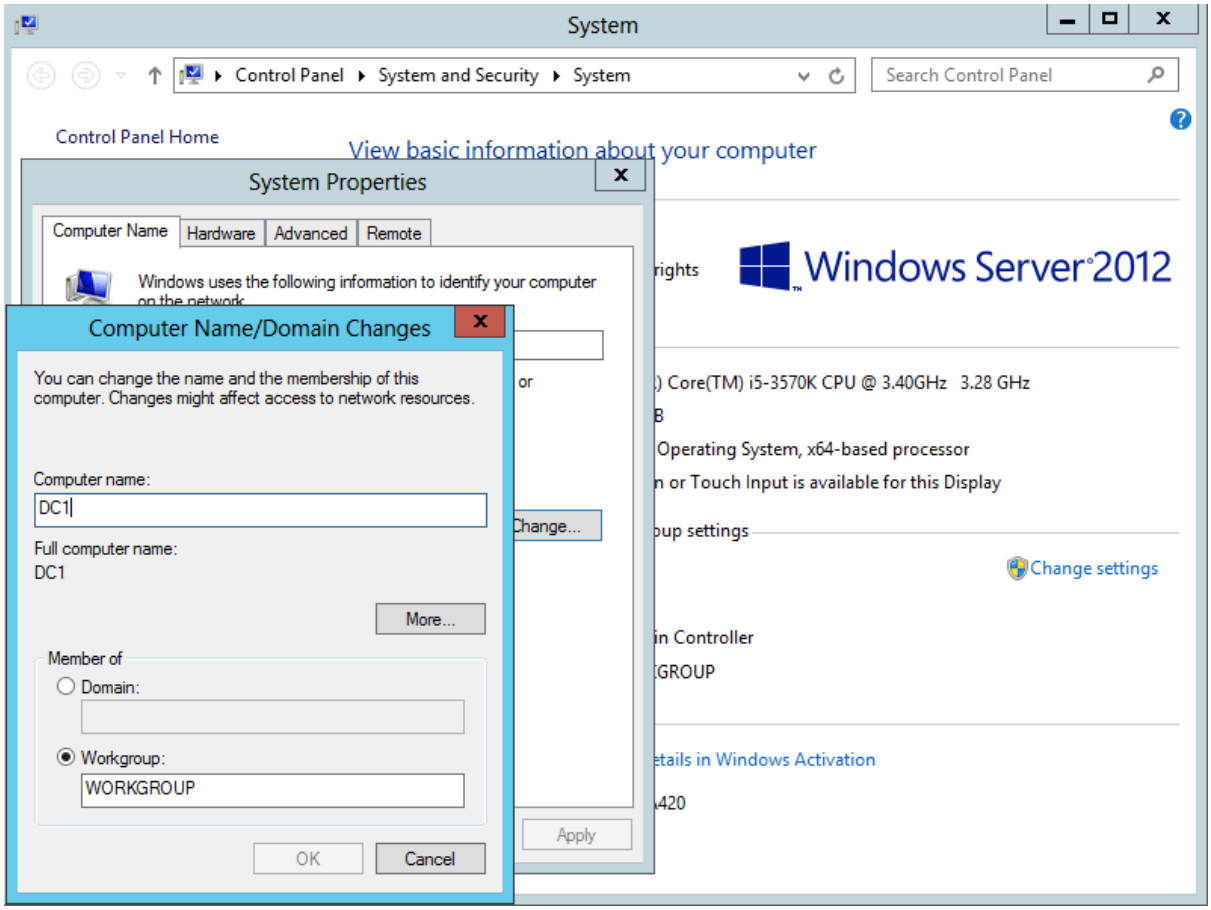
“**Properties -> Internet Protocol Version 4 (TCP/IPv4)**“ и задать **IP** адрес по подобию, как приведено на картинке.

192.168.0.11 — IP адрес текущего сервера, — первого контроллера домена. **192.168.0.254 — IP** адрес шлюза.



Теперь необходимо переименовать имя сервера и перезагрузить его. “**Start -> System -> Change Settings -> Computer Name ->**

Change“. Ввести “**Computer Name**“. В примере сервер будет называться “**DC1**“.

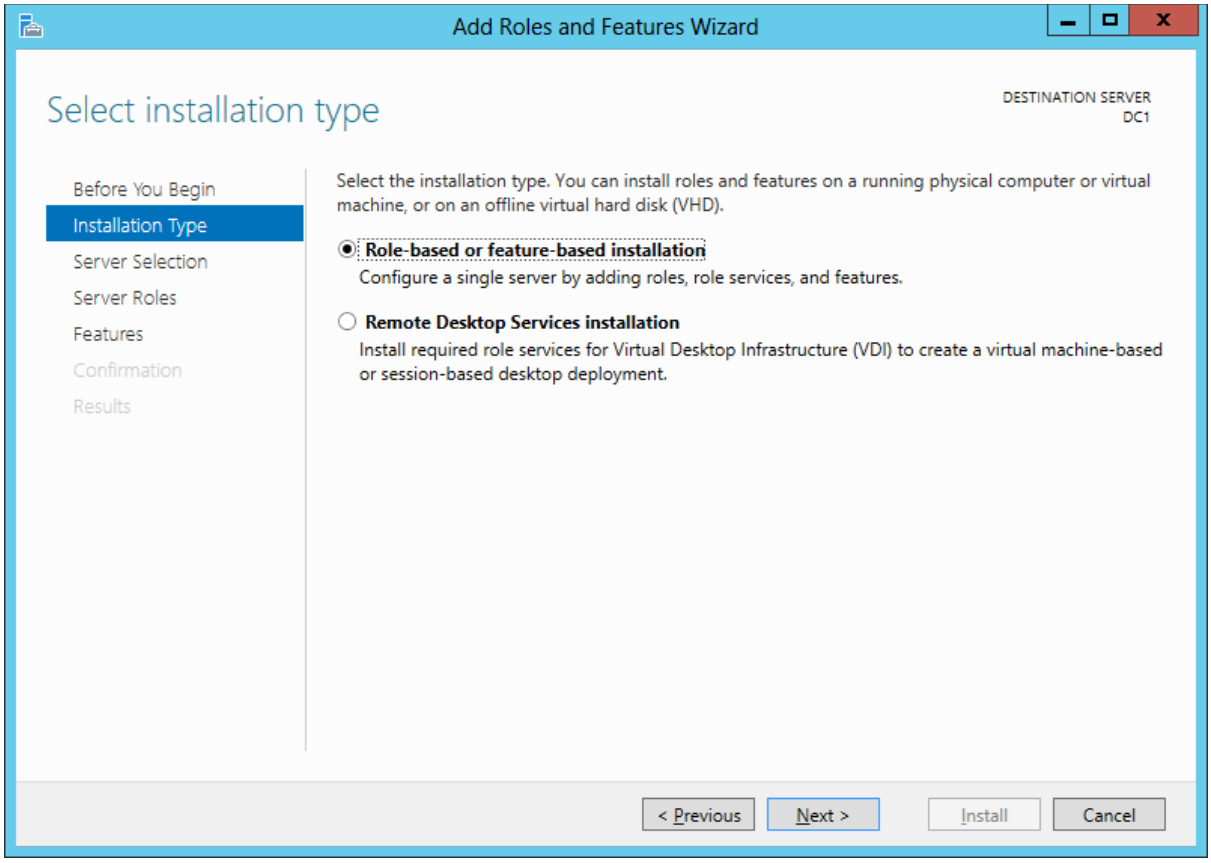


Установка роли Active Directory на Windows Server 2012

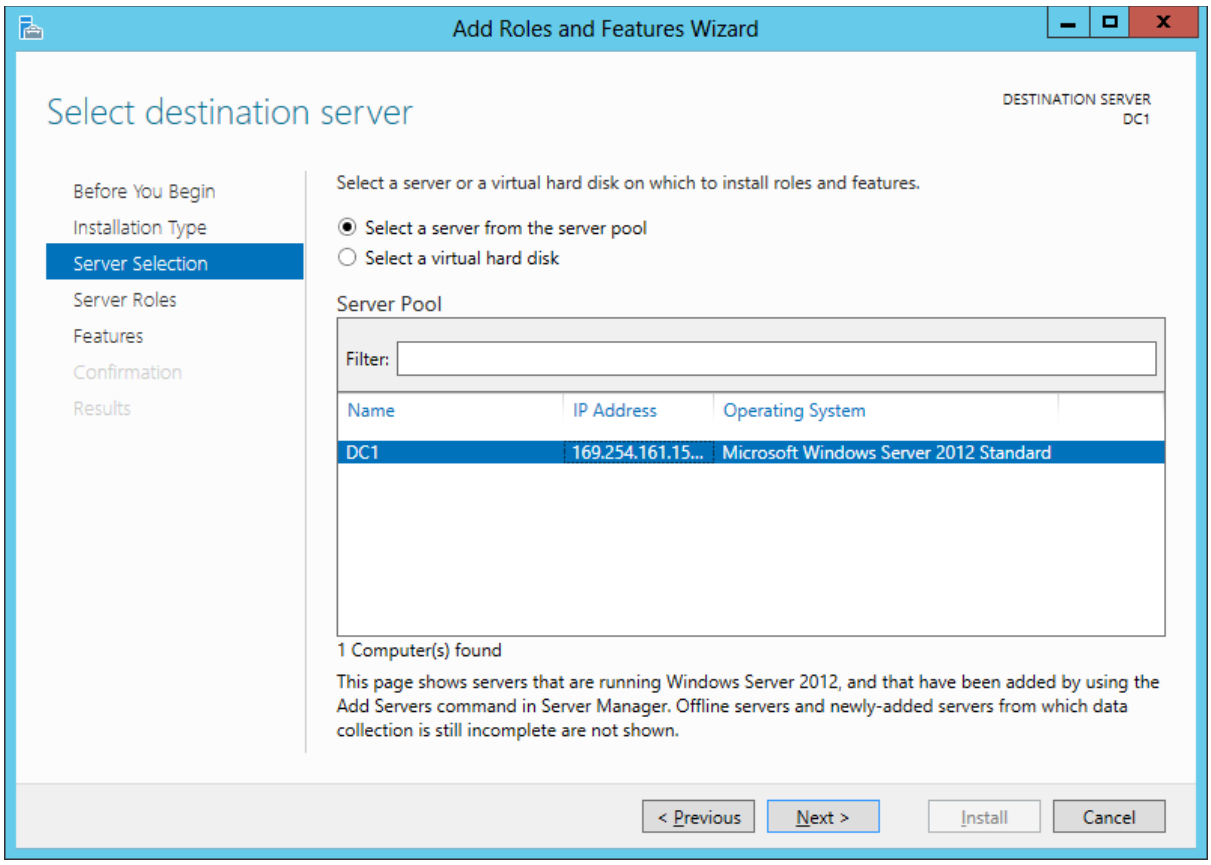
Итак, после предварительной настройки сервера, переходим к установке роли службы каталогов.

- **Start -> Server Manager (Пуск -> Диспетчер сервера).**
- **Add roles and features -> Next**

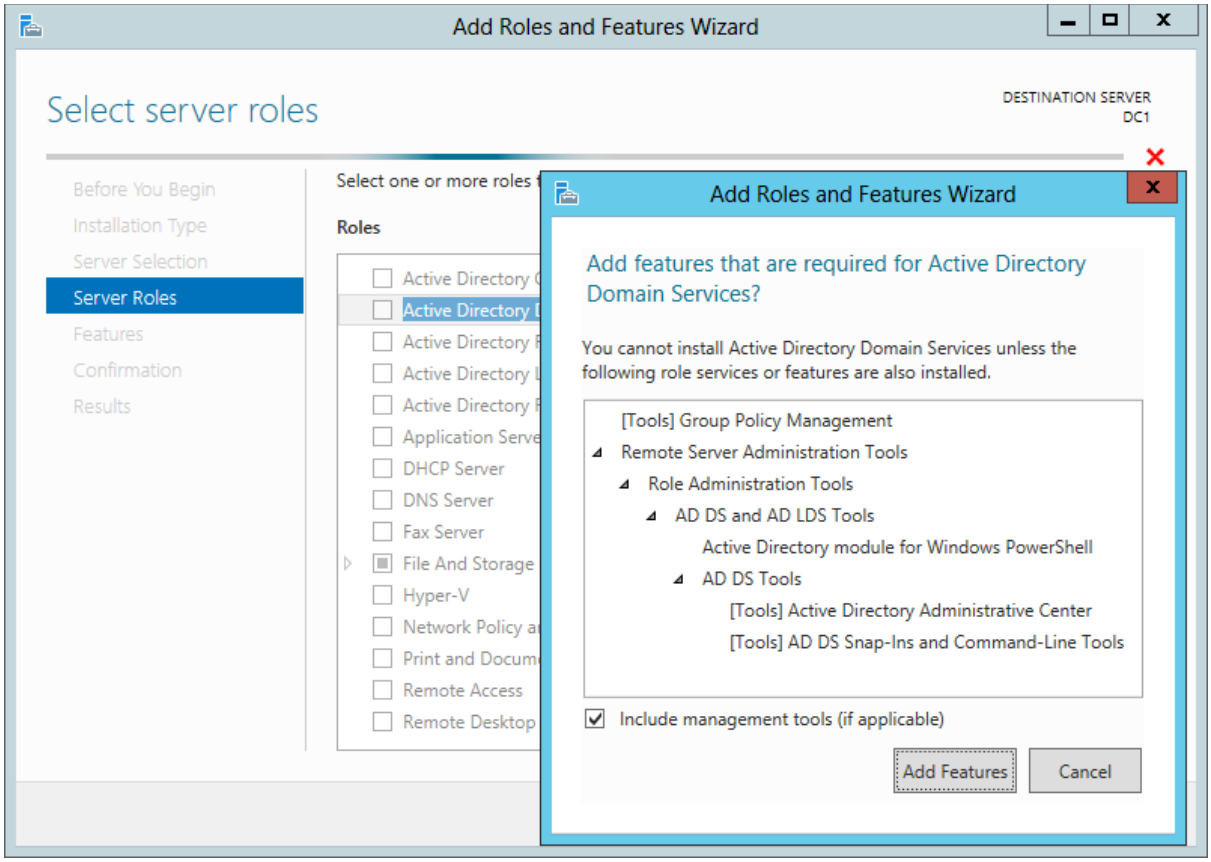
Выбрать “**Role-based or feature-based Installation** (Установка ролей и компонентов)“ -> **Next**



Выбрать сервер, на который устанавливается роль **AD** и нажать Далее. “**Select a server from the server pool -> Next**“



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**.



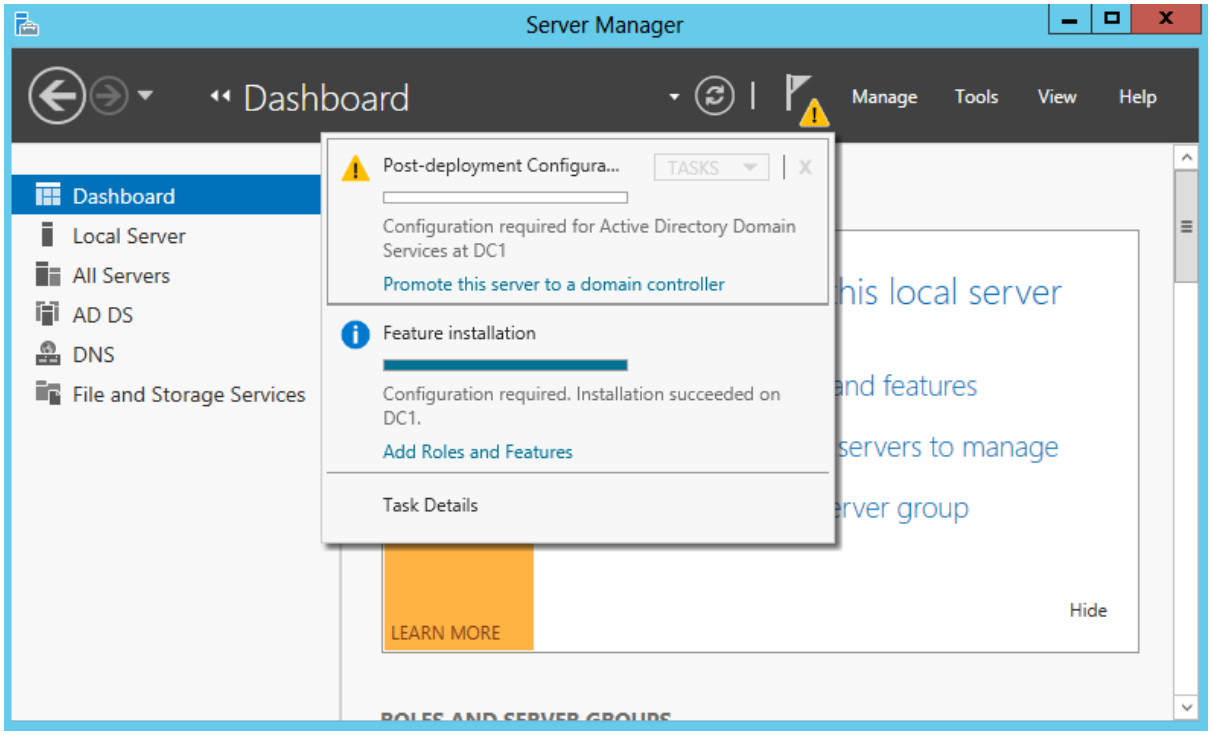
Можно также выбрать роль **DNS Server**. Если вы забудете установить галочку для добавления роли **DNS Server**, можно особо не переживать, т.к. её можно будет добавить позже, на стадии настройки роли **AD**.

После этого жмем каждый раз кнопку **Next** и устанавливаем роль.

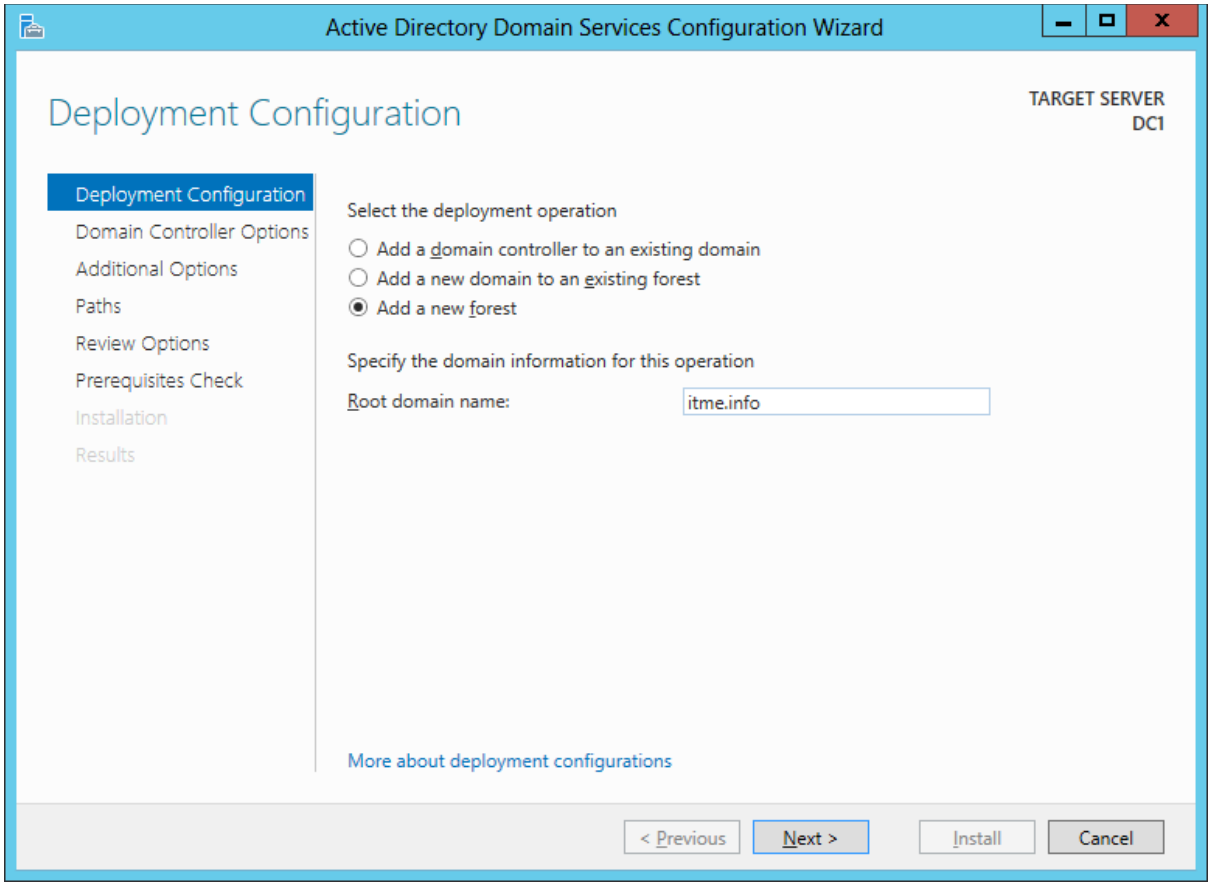
Настройка доменных служб Active Directory

После установки роли, закрыть окно — **Close**. Теперь необходимо перейти к настройке роли **AD**.

В окне **Server Manager** нажать пиктограмму флага с уведомлением и нажать “**Promote this server to a domain controller**“ (Повысить роль этого сервера до уровня контроллера домена) на плашке “**Post-deployment Configuration**“.



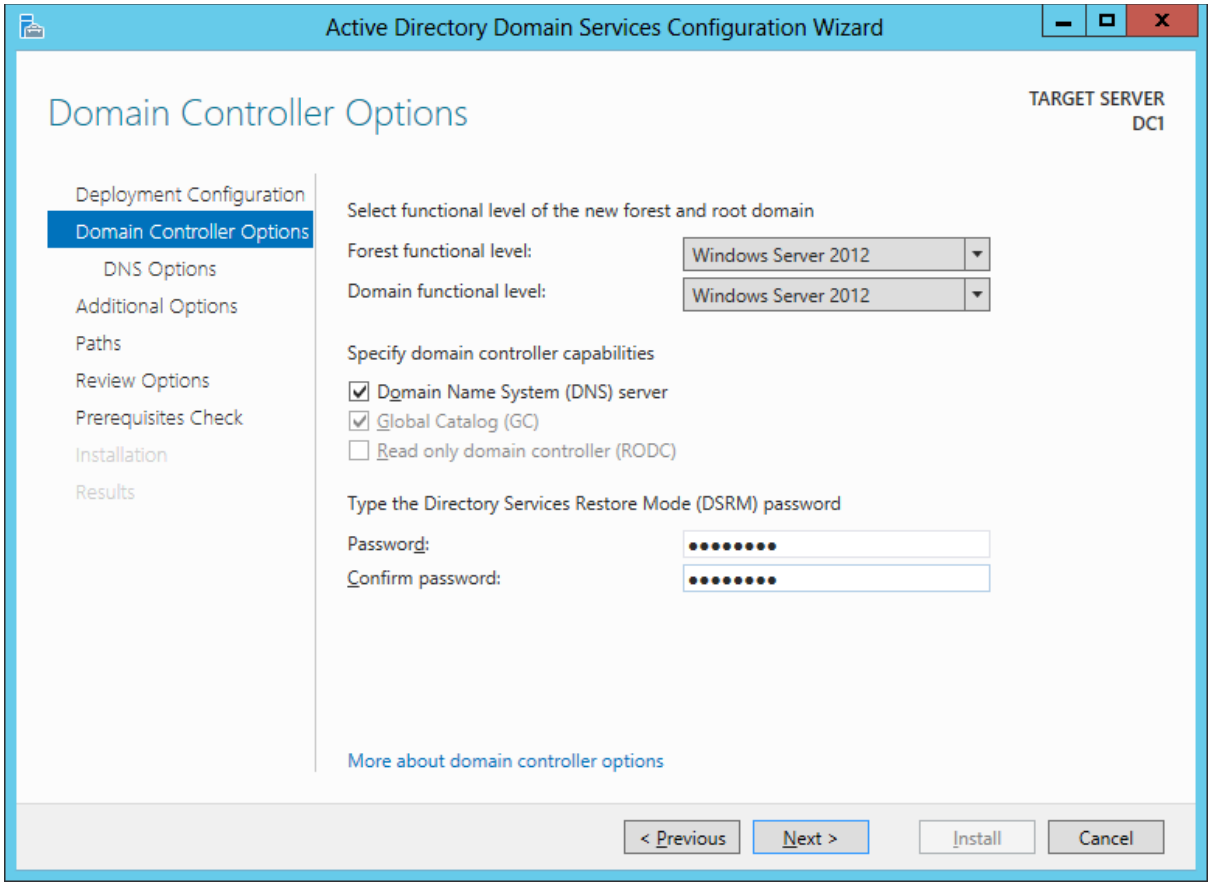
Выбрать “**Add a new forest** (Добавить новый лес)“, ввести название домена и нажать “**Далее**“.



Можете выбрать совместимость режима работы леса и корневого домена. По умолчанию устанавливается “**Windows Server 2012**“.

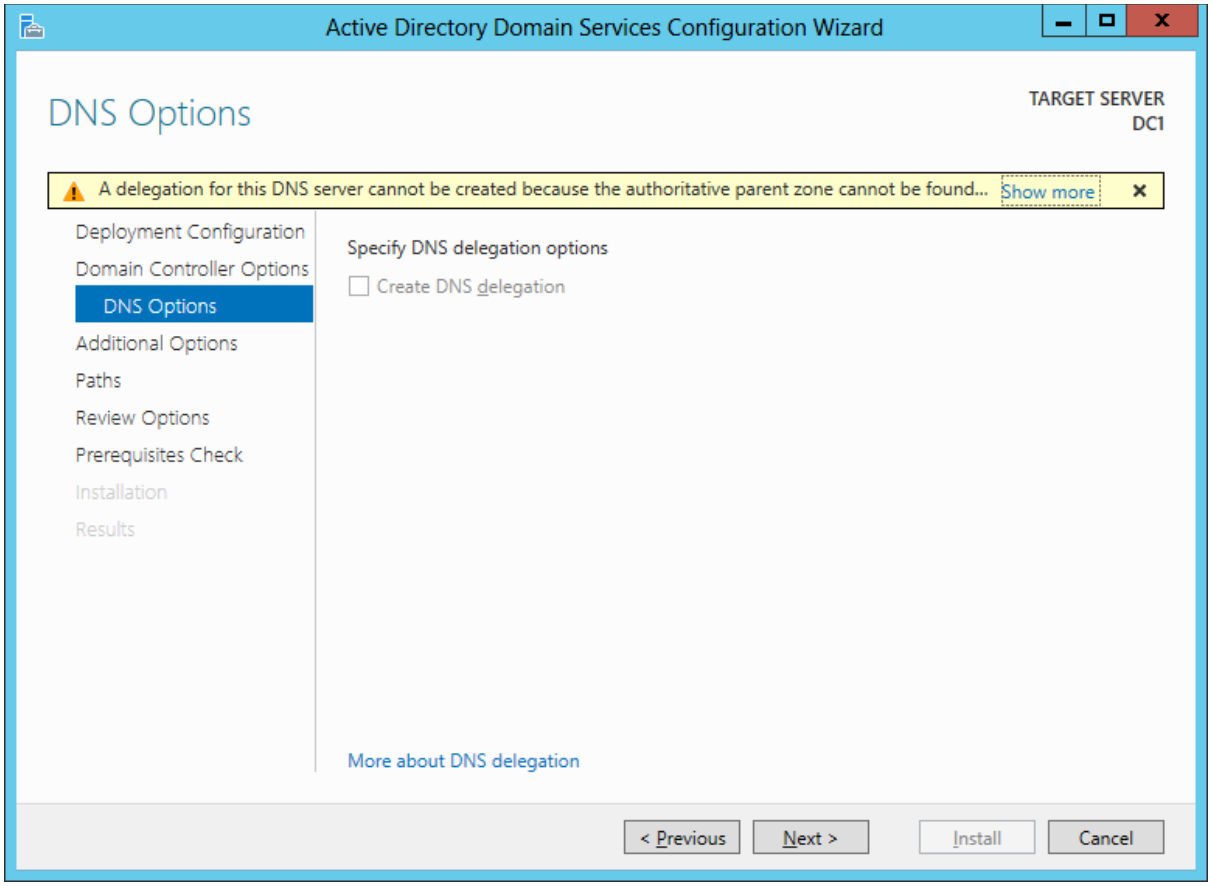
На этой вкладке можно будет отключить роль “**DNS Server**“. Но, в нашем случае, галочку оставляем.

Далее ввести пароль для “**DSRM (Directory Service Restore Mode**“ — режим восстановления службы каталога) и нажимаем “**Далее**“.

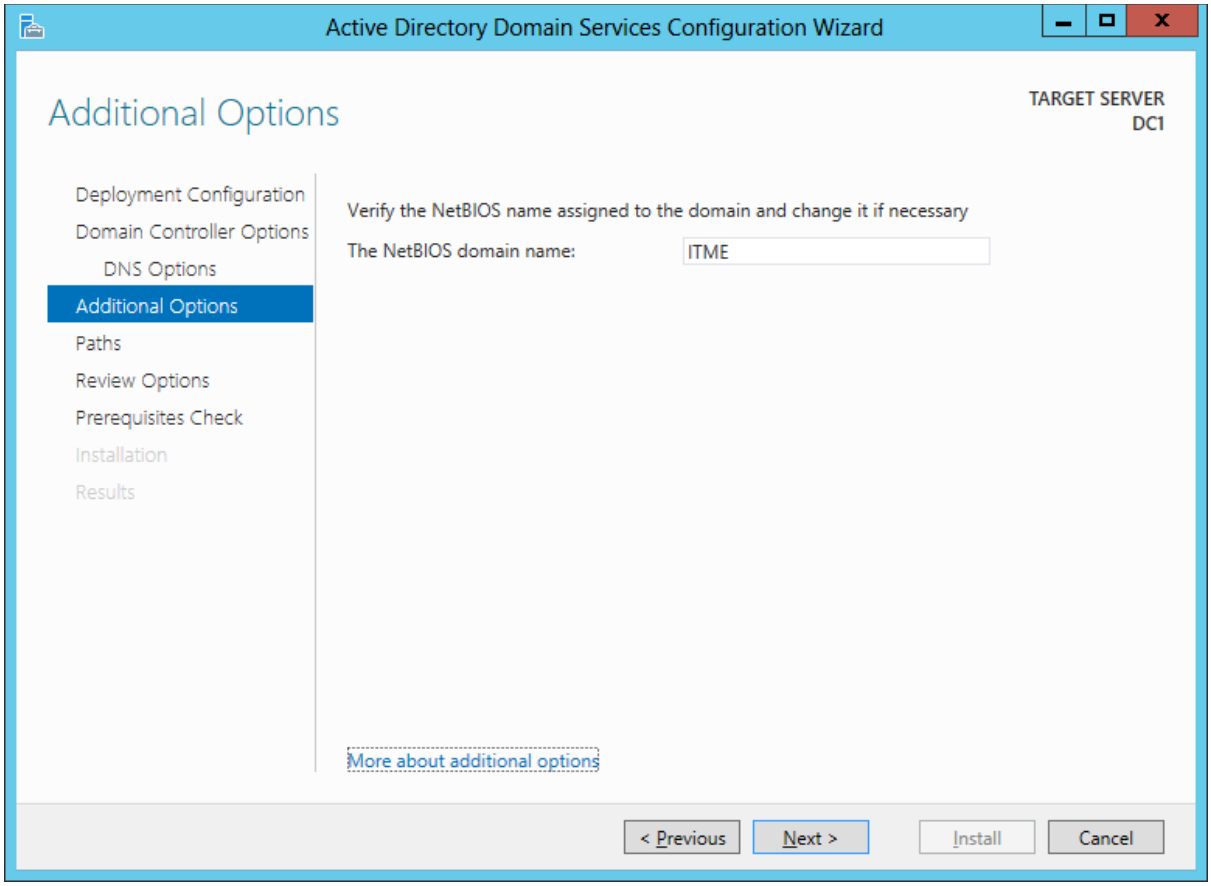


На следующем шаге мастер предупреждает о том, что делегирование для этого **DNS**-сервера создано не было (A delegation for this DNS server cannot be created because the authoritative parent zone cannot be found or it does not run Windows DNS server. If you are integrating with an existing DNS infrastructure, you should manually create a delegation to this DNS server in the parent zone to ensure reliable name resolution from outside the domain «itme.info». Otherwise, no action is required.).

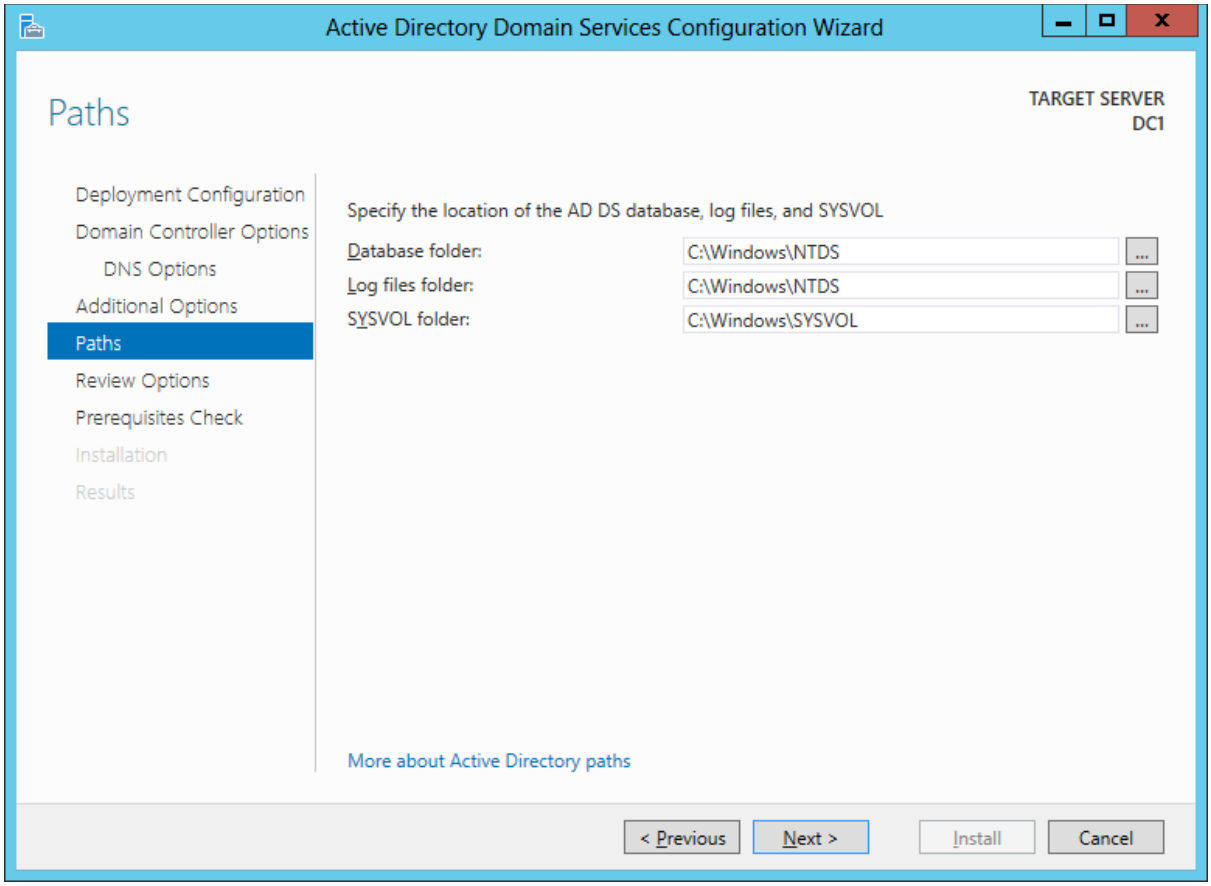
Нажимаем “**Next**“.



На следующем шаге можно изменить “**NetBIOS**“ имя, которое было присвоено домену. Мы этого делать не будем. Просто нажимаем “**Далее**“.



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**“. Мы менять ничего не будем. Нажимаем кнопку “**Далее**”.



На следующем шаге отображается сводная информация по настройке. Нажав кнопку **View Script**, можно посмотреть **Powershell** скрипт, который произведет настройку доменных служб **Active Directory**.

```
# Windows PowerShell script for AD DS Deployment

Import-Module ADDSDeployment
Install-ADDSForest `
-CreateDnsDelegation:$false `
-DatabasePath "C:\Windows\NTDS" `
-DomainMode "Win2012" `
-DomainName "itme.info" `
-DomainNetbiosName "ITME" `
-ForestMode "Win2012" `
```

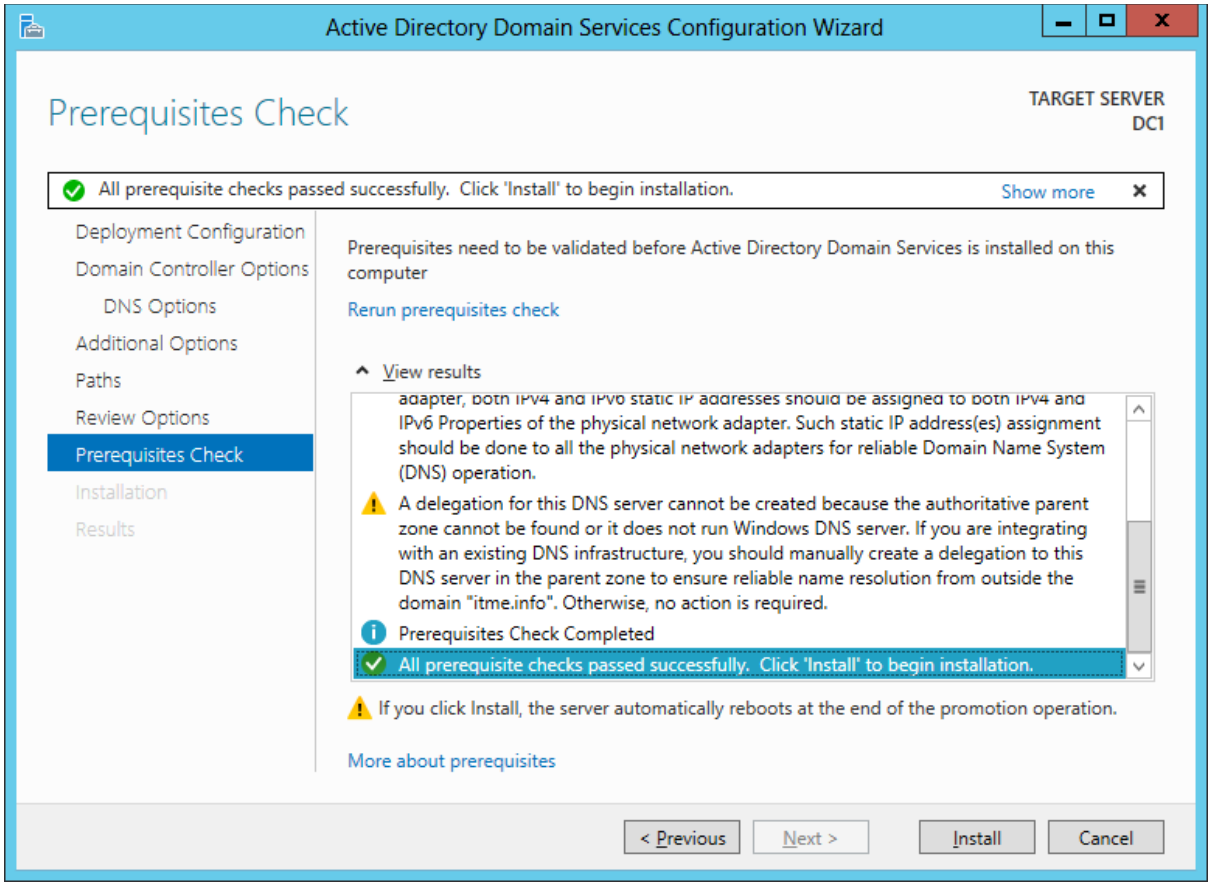
```
-InstallDns:$true                                                    `
-LogPath                    "C:\Windows\NTDS"                        `
-NoRebootOnCompletion:$false                                         `
-SysvolPath                 "C:\Windows\SYSVOL"                      `
-Force:$true
```

Убедившись, что все указано верно, нажимаем на кнопку “**Next**“.

На следующем шаге производится проверка, все ли предварительные требования соблюдены. После чего покажет нам отчёт. 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**“ уровень сервера будет повышен до контроллера домена и будет произведена автоматическая перезагрузка.

Должна появиться надпись “**All prerequisite checks are passed successfully. Click «install» to begin installation**”.

Нажимаем кнопку **Install**.



После завершения всех настроек, сервер перезагрузится, и вы совершите первый ввод компьютера в ваш домен. Для этого необходимо ввести логин и пароль администратора домена.

На этом базовая настройка служб каталога **Active Directory** завершена.

Добавление компьютеров к домену Server 2012

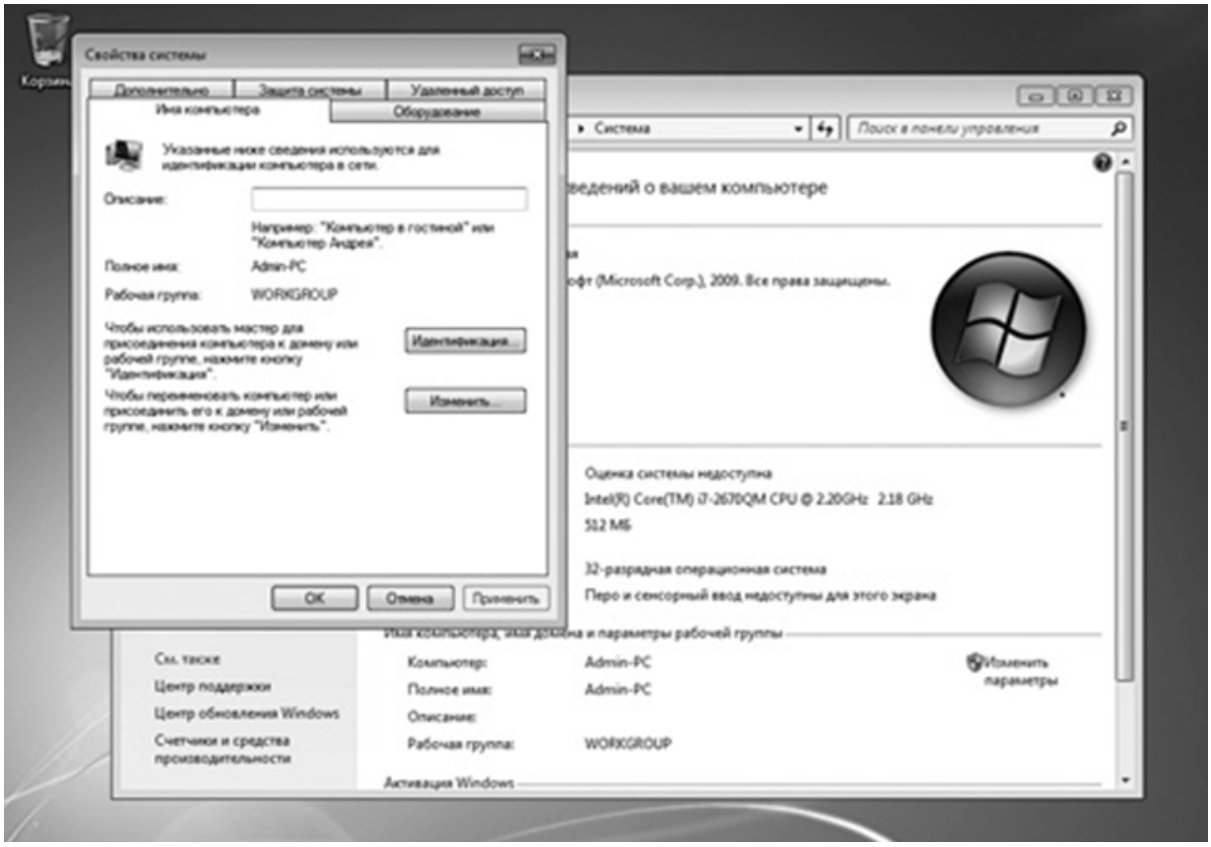
Обычно **IT**-специалисты сначала обновляют серверы, а уже потом — клиентские компьютеры или просто заменяют устаревшие машины новыми, на которых установлены последние версии

клиентских ОС **Windows**. Вполне возможно, что в некоторых организациях к доменам **Server 2012** будут подключать не только компьютеры под управлением **Windows 8/10**, но и системы с установленной **Windows 7** (а иногда и с **Windows XP**).

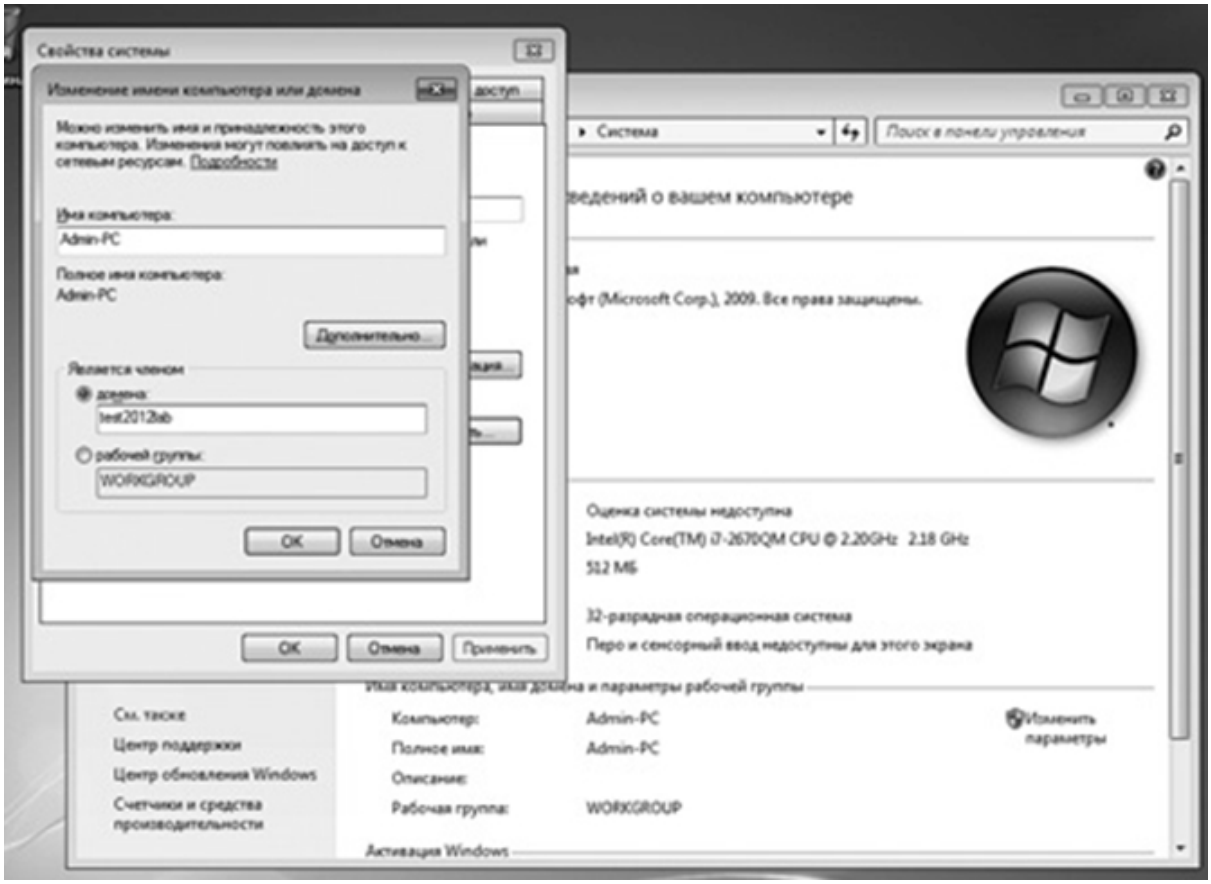
В крупных организациях клиентские компьютеры и серверы можно присоединить к домену с использованием автоматизированных методов, таких как использование сценариев и пакетных файлов. В этом разделе сосредоточимся на подключении к домену с применением интерфейса клиентской системы. Обычно этот метод используют в небольших сетях. Данная методика работоспособна и на виртуальных машинах.

Подключение к домену Server 2012 компьютера, работающего под управлением Windows 7

В панели управления (**Control Panel**) **Windows 7** в разделе “Система” (**System**) откройте окно “**Свойства системы**” (**System properties**), перейдите в нем на вкладку “**Имя компьютера**” (**Computer Name**) и нажмите кнопку “**Изменить**” (**Change**).



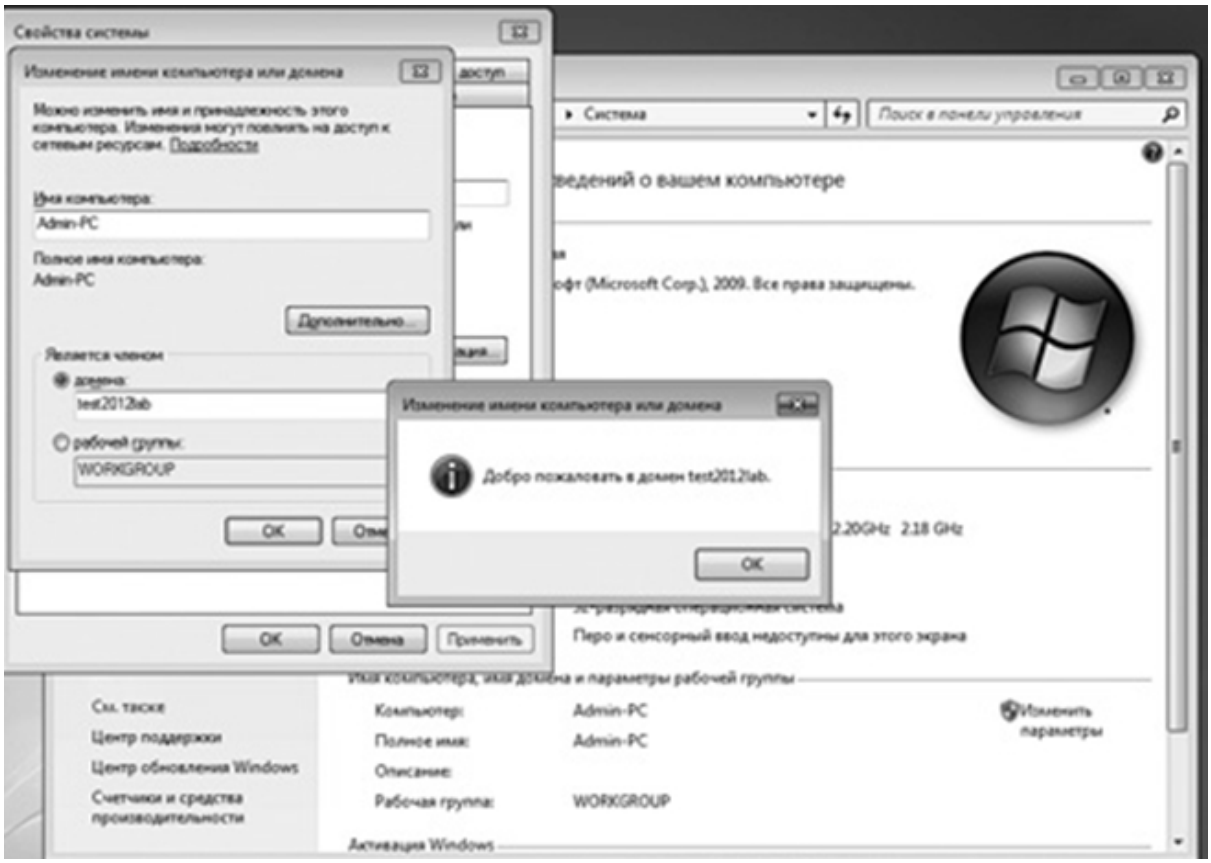
Введите имя домена в поле домена (**Domain**) переключателя “**Является членом**” (**Member of**).

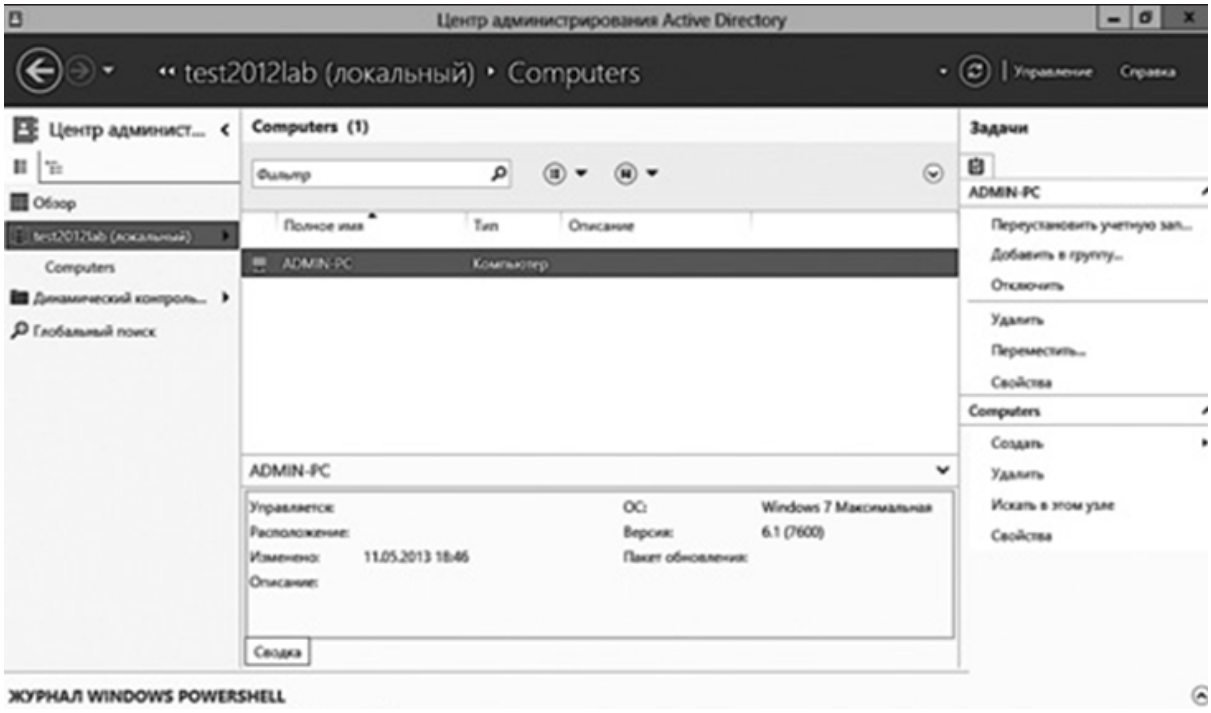


Введите учетные данные пользователя, имеющего разрешение на подключение компьютеров к домену. После того, как компьютер

будет успешно подключен к домену, появится соответствующее сообщение.

Как только клиентский компьютер присоединится к домену, вы сможете найти его в контейнере “**Computers**” (Компьютеры) центра администрирования “**Active Directory**” (**ADAC**).





Добавление компьютеров под управлением Windows 8 к домену Server 2012

С тех пор как была выпущена версия **Windows 8 Consumer Preview**, вокруг клиентских систем на **Windows 8** возникает множество споров, их подвергают критике. Многим кажется, что эта операционная система больше похожа на платформу мобильных устройств для обычных пользователей, чем на систему, ориентированную на бизнес-цели. Однако приход **Windows 8** (а теперь уже и **Windows 10**) в корпоративную среду неизбежен, и в первую очередь благодаря ее новым возможностям. Я помню себя администратором **Windows Server**, пытающимся подключить последнюю версию клиентской операционной системы к домену только потому, что некоторые руководители высшего звена хотели попробовать поработать с новой ОС.

У меня есть для вас и хорошие новости о подключении **Windows 8/10** к корпоративным доменам, и не очень хорошие. Мобильные устройства, оснащенные ARM-процессорами, на которые установлена **Windows 8/10**, так называемая **Windows RT**, официально не могут быть присоединены к домену **Windows**. Это не означает, что вы не можете подключать мобильные устройства на **Windows RT** к сетевой инфраструктуре организации. Для этой цели можно использовать как собственные возможности ОС, так и средства сторонних разработчиков для управления мобильными устройствами (**Mobile Device Management, MDM**).

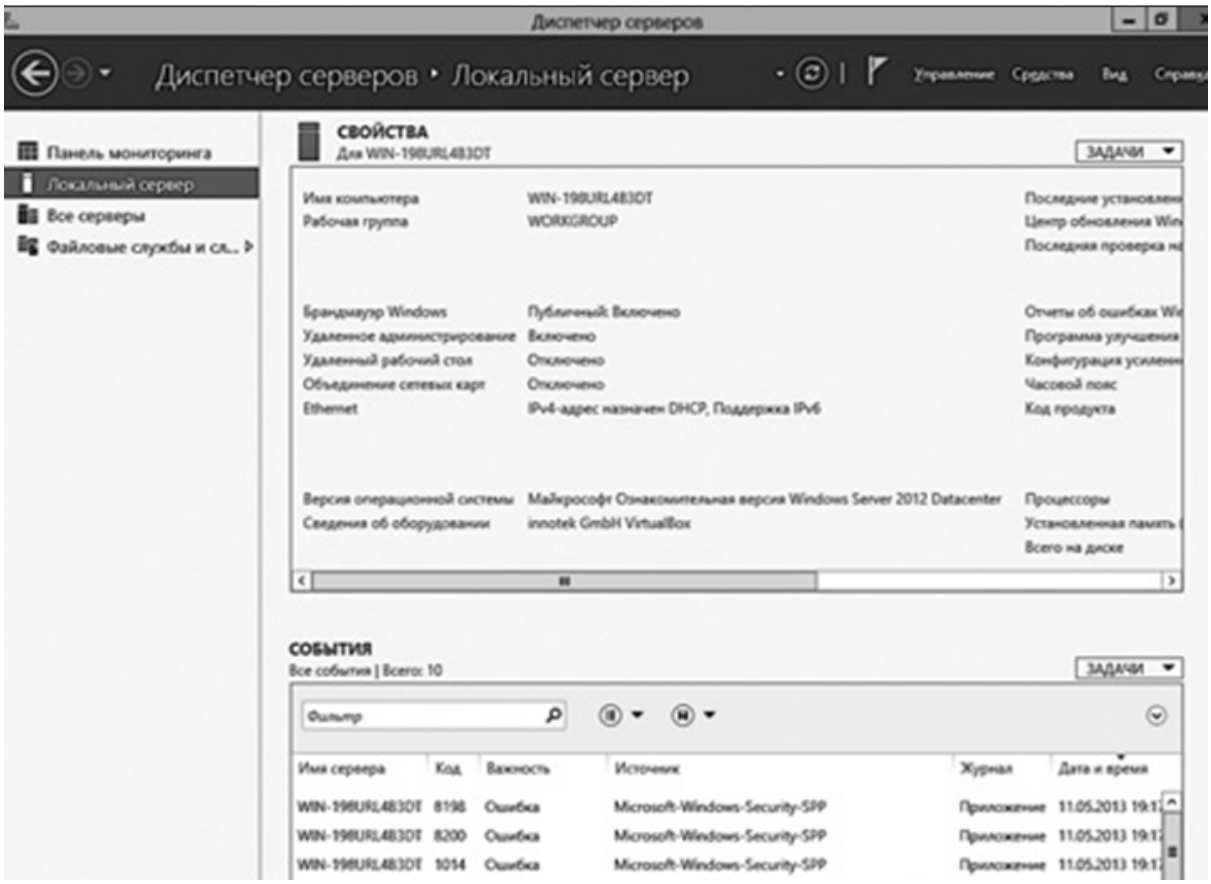
Однако вы можете подключать к доменам компьютеры и виртуальные машины с установленной **Windows 8/10**. Для того чтобы подключить компьютер с установленной **Windows 8/10** к домену уровня Server **2012**, достаточно выполнить те же действия, которые выполняются при подключении к домену **Windows 7** клиентов. Откройте панель управления (**Control Panel**), войдите в раздел “Система и безопасность” (**System and Security**). Далее перейдите в раздел “Система” (**System**) и щелкните на ссылке “Дополнительные параметры системы” (**Advanced system settings**).

В появившемся окне перейдите на вкладку” Имя компьютера” (**Computer Name**) и нажмите кнопку” Изменить” (**Change**). Введите имя домена в соответствующее поле. Как только компьютер

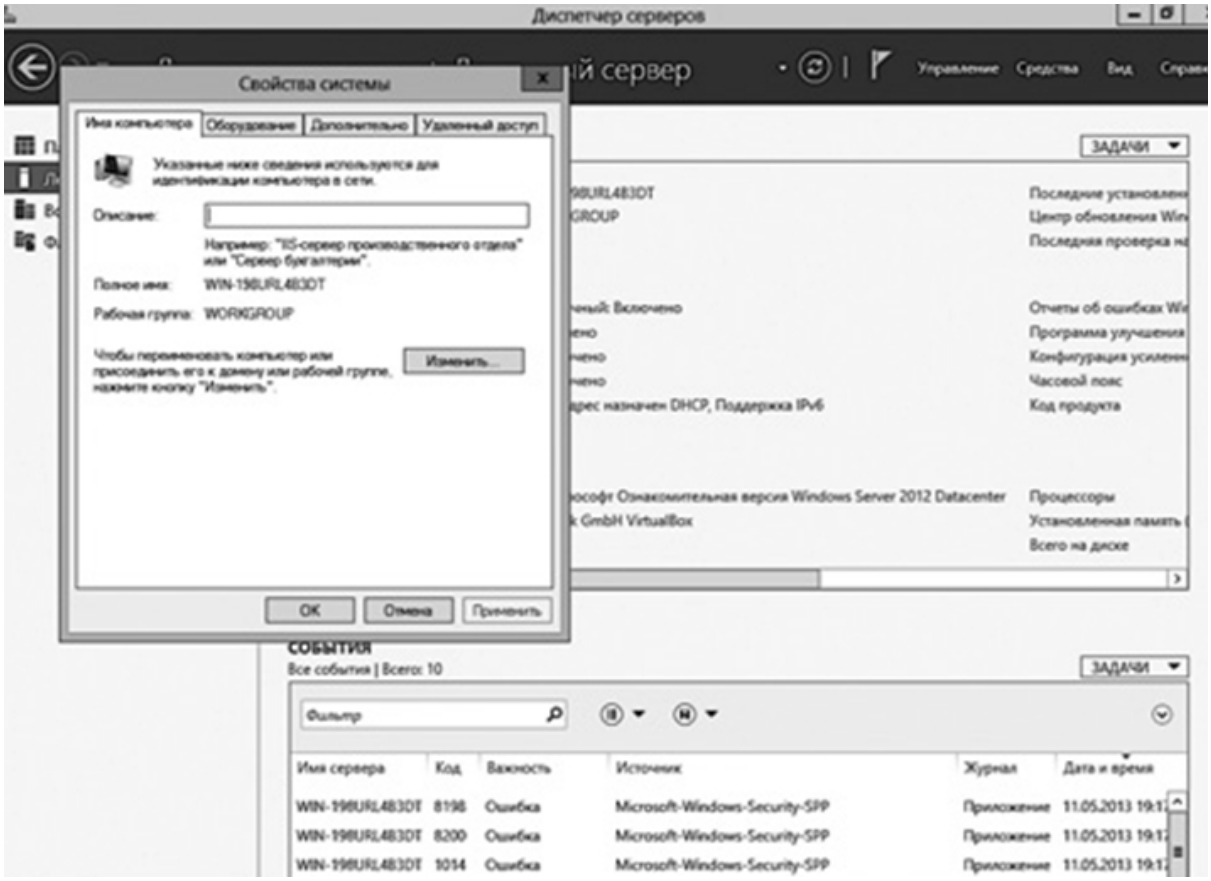
будет подключен к домену, его можно будет найти в контейнере" Компьютеры" (**Computers**) в" ADAC".

Для подключения к домену **Server 2012** систем, работающих под управлением Server **2003, 2008 и 2008 R2**, в качестве обычных серверов можно воспользоваться той же процедурой, которая используется для подключения клиентских **Windows**-систем.

Обновленный интерфейс диспетчера серверов в **Server 2012** позволяет вам быстро добавлять в домен компьютеры под управлением **Server 2012**. Для того чтобы это сделать, достаточно открыть диспетчер серверов, щелкнуть на ссылке "Настроить" этот локальный сервер" (**Configure this local server**). В группе параметров "Свойства" (**Properties**) нужно щелкнуть на пункте **WORKGROUP**).



Щелчок WORKGROUP приведет к открытию окна “Свойства системы” (**System Properties**), с помощью которого вы можете добавить сервер к домену.



Пожалуй, на этом всё. Перейдём к послесловию.

Послесловие к четвертой главе

И хотя это далеко не всё, но минимальный базис выглядит примерно вот так, как я его, собственно, описал выше. Впрочем, базис он на то и базис.

Не так страшно, как могло казаться, не так ли? Попробуйте, Вам понравится.

Вообще, в отличии от линейки пользовательских операционных систем от **Microsoft**, прогресс серверных меня определенно радует (хотя я и предпочитаю линуксы), ибо администрировать становится удобнее, быстрее и проще, а это крайне немаловажный фактор, ибо человекочас админской работы стоит достаточно дорого... в плане наладки и, тем более, восстановления работы компании.

Если повезет и размер книги не будет слишком большой, то я расскажу Вам о том, о чем обещал в первой части, а именно, об **Active Directory**, Hyper и других вещах, что были заявлены на этапе предзаказа (напоминаю, что там стояла пометка “это описание, есть ни что иное, как так называемая "рабочая" версия содержания, она не является финальной и оглавлением”).

Если не повезет, но книга всё равно Вам понравится, то ждите следующих частей... Или записывайтесь к нам на обучение - http://sonikelf.ru/obuchayu-nachinayushhix-i-specialistov/ - там больше всяческих вкусняшек ;)

А пока давайте двигаться дальше и посмотрим на **Linux** сервера, которые пугают пользователей сильнее, чем все **Windows** вместе взятые.

Глава 5. Linux Server. Базис

Предисловие к пятой главе

Что ж, с **Windows**'ами кратенько разобрались, можно и с моим любимым линуксом поработать, благо его, линукса, много (в смысле дистрибутивов), он хороший и идеально подходит для решения огромного количества задач.

Плюс ко всему, конечно же, чаще всего, бесплатность, стабильность, антивирусная безопасность, малая прожорливость ресурсов и, в идеальном случае, отсутствие графического интерфейса и полное управление с консоли (при условии, что Вы достаточно опытны). А уж использование оного под определенные виды серверных ролей вообще чистое, незамутненное удовольствие. Во всяком случае на мой искушенный вкус.

Приступим.

Linux Server - какой, почему и как

Linux Server - бесконечность дистрибутивов

Для начала давайте определимся с терминологий, т.е с тем, что же такое дистрибутив?

Дистрибутив - условно говоря “сборка чего-то от кого-то”, т.е так называют варианты операционной системы **Linux** (и не только), выпускаемые разными компаниями и разработчиками, которые включают в состав каждого дистрибутива своё ядро, драйверы, системные утилиты, и различного рода софт (который обычно весит больше всего, т.к, на самом деле, вес системы может исчисляться парой десятков мегабайт).

Linux родилась в начале **90**-х годов и численность дистрибутивов уже измеряется сотнями, если не тысячами, причем, условная простота сборки дистрибутивов, порождает множество, что называется, на-коленке-в-одно-лицо, в том числе и от наших многострадальных соотечественников. Справедливость ради стоит сказать, что всяких сборок-дистрибутивок **Windows** (вспомните вездесущий и популярнейший **Windows XP Zver Edition**) тоже предостаточно, но концепция там несколько иная, хотя принципы те же самые.

К счастью, особенно начиная знакомство со "свободной ОС" (как часто Линукс называют), нет нужды просматривать все

существующие её версии, иначе может поехать крыша, да и просто это, по сути бессмысленно, ибо, во-первых, большая часть из них основана на одном из типов ядер, собранном, под определенные нужды и цели, с соответствующим набором драйверов и софта.

Вообще говоря, базовая производительность и функциональность большей части дистрибутивов почти идентичны, да и подавляющее большинство популярных дистрибутивов содержат один и тот же софт, различаясь лишь мелочами: средствами для управления программами, процедурами установки, командами консоли, визуальной оболочкой, репозиториями и тп. Добавить недостающие приложения в систему после её инсталляции по силам каждому, даже начинающему пользователю, да и делается это не так сложно, как когда-то могло показаться, благо линуксы уже давно обзавелись визуальным графическим интерфейсом и тучей автоматизированных скриптов, если Вы, что называется, не разбираетесь в консоли.

Из вышесказанного следует важное правило: если что-то идет не так, то не всегда виноват дистрибутив и надо срочно всё бежать-переустанавливать и прочее, достаточно заручиться поддержкой умных книг, форумов или сайта **sonikelf.ru** :) Просто потому, что всё достаточно типовое и различается лишь подход+способ реализации задачи, и, в общем-то, нельзя сделать чего-то в одном дистрибутиве такого, чего нельзя в другом. Несколько утрировано, но для большинства случаев это факт.

Отсюда же следует, что, подбирая **Linux**-дистрибутив для сервера, лучше сосредоточиться не на технических характеристиках, а на цели, которую Вы преследуете (вспоминайте главу про сервера и слова о том, исходя из чего Вы делаете выбор железок и прочего).

Так как начинать с “тяжелых” вещей в первой части серии книг по серверам, собственно, нельзя, то мы рассмотрим вариант с одной из самых распространенных вариантов линукса, а заодно затронем базовые темы, вроде графического интерфейса (**GUI**), репозиториев и структуры файловой системы.

Начнем, пожалуй.

Установка, первичная настройка Ubuntu Server

Маленькая вводная

Так уж сложилось, что **Ubuntu** стал одним из самых популярных дистрибутивов линукса, что и хорошо, и плохо одновременно.

И хотя их курс в развитии своей **ОС** для домашней машины, мне, в общем-то, не очень нравится, надо признать, что серверная часть достаточно неплоха, проста и приятна, в установке и управлении. Хотя, лично автор, т.е я, предпочитает **CentOS**, но это уже совсем другая история.

Здесь и далее, я буду исходить из того, что мы делаем пробную установку на виртуальную машину, т.к для первого раза это

нагляднее, проще, а также позволит Вам использовать сразу связку **Linux + Windows** без лишних перезагрузок и других заморочек.

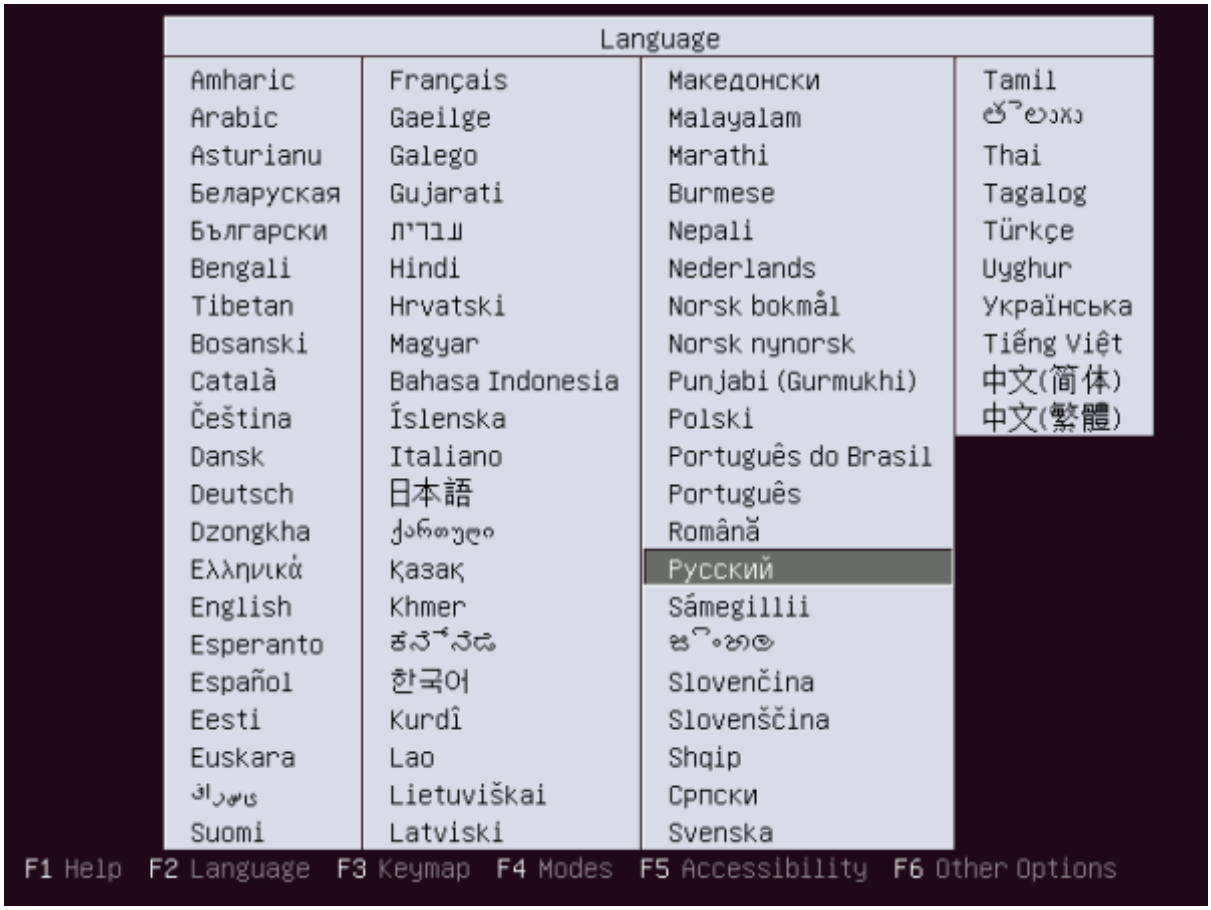
По поводу того, что такое виртуальная машина, виртуализация и прочее, я, собственно, писал на сайте проекта в статье: “**Система в системе: установка и использование Linux в Windows [виртуализация] [Virtual Box]**” и в какой-то из версий первой книги (да-да, их было несколько ;)) **ПНПК**.

Загрузка и установка Ubuntu Server

Начнем с загрузки официального дистрибутива **Ubuntu Server** с официального русского сообщества.

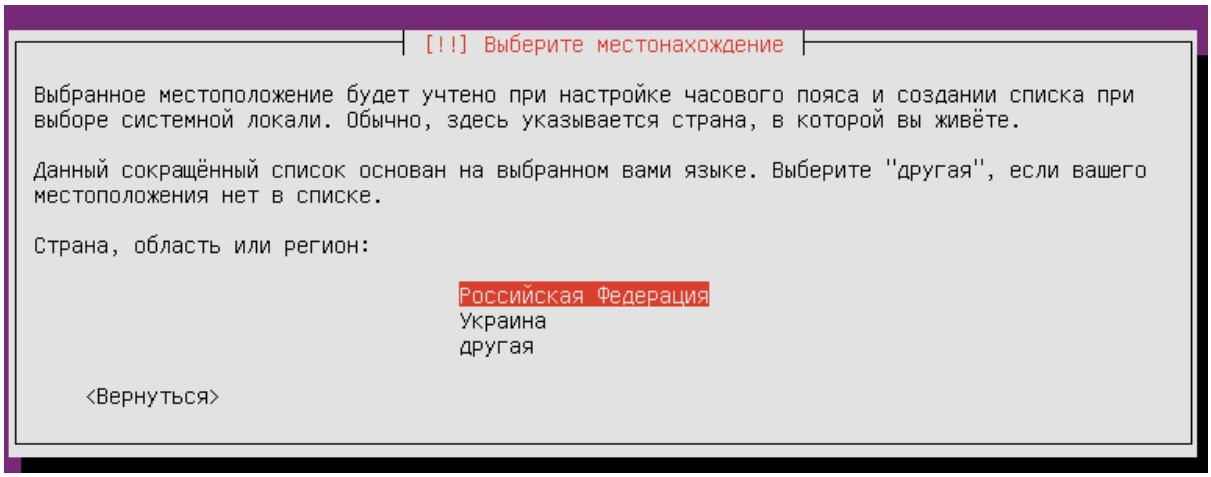
Наверняка на момент прочтения Вами этой книги вышло больше стабильных и старших версий, но я покажу на примере той, что есть сейчас (14.04.1 LTS), а там уже Вы сможете сами потренироваться на новых, что даст полезный опыт и практику. Да и наверняка установка-настройка не будет сильно отличается от описываемой ниже.

После подключения образа диска к виртуальной машине (или загрузки с внешнего носителя, смотря как именно Вы решили делать установку), запускаем её и если все нормально, вы должны увидеть окно выбора языка установки.

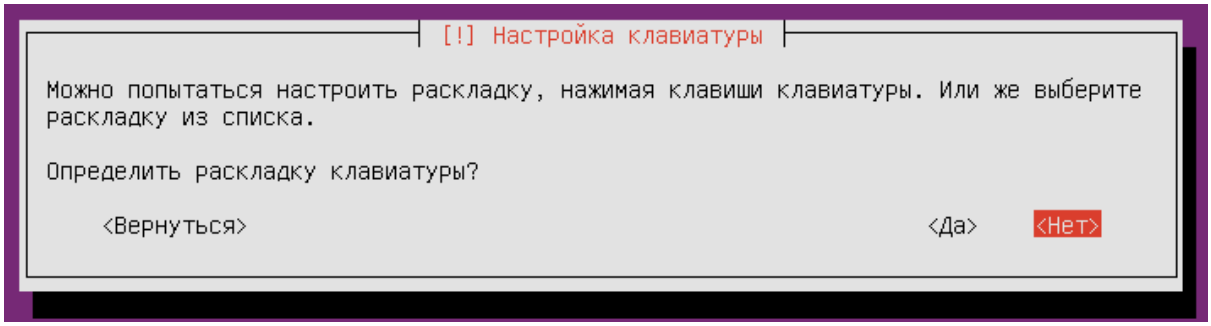


Выбираем Русский язык и жмем “**Enter**”... Далее, что логично, мы выбираем пункт “**Установить Ubuntu Server**”:

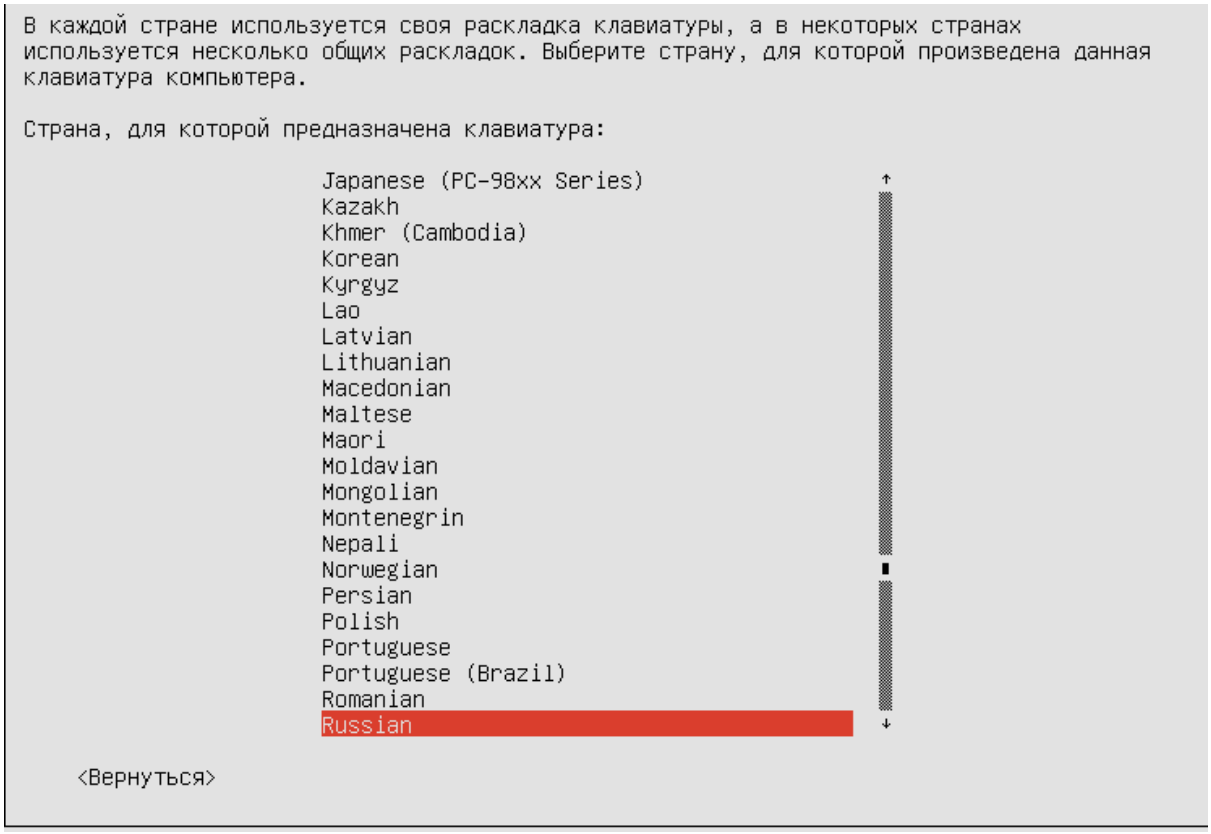
Далее нас спросят, где мы находимся. Отвечаем соответствующим образом:



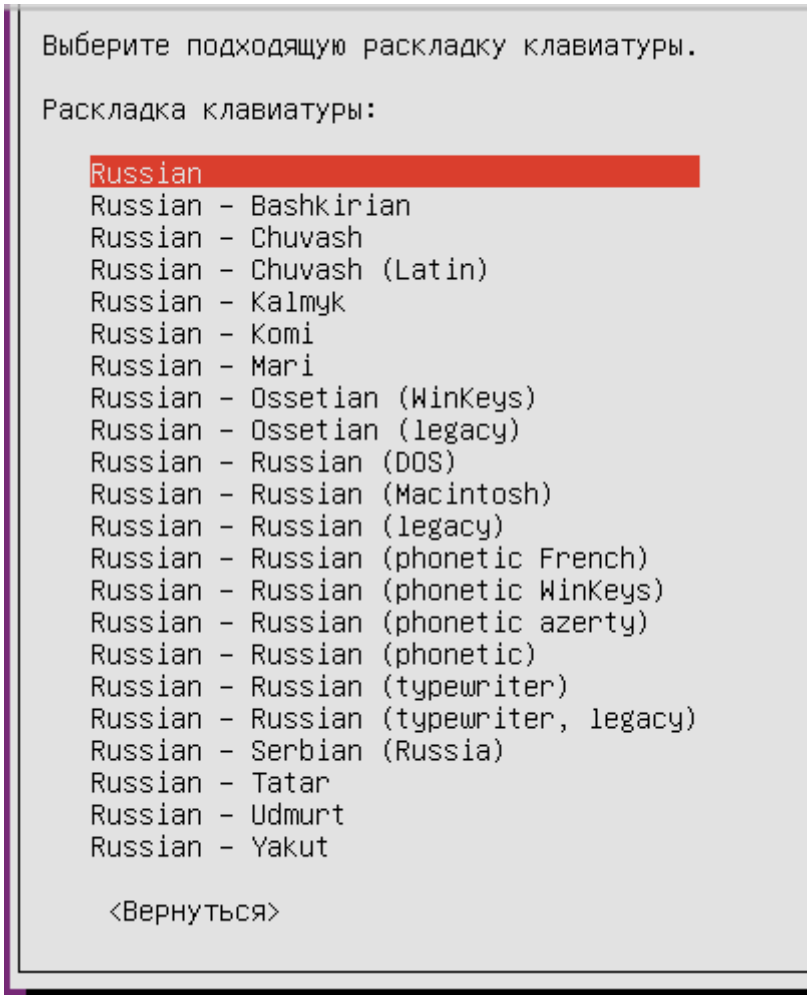
После, установщик предложит настроить клавиатуру или выбрать из списка. Жмем “**Нет**”, для выбора из списка:



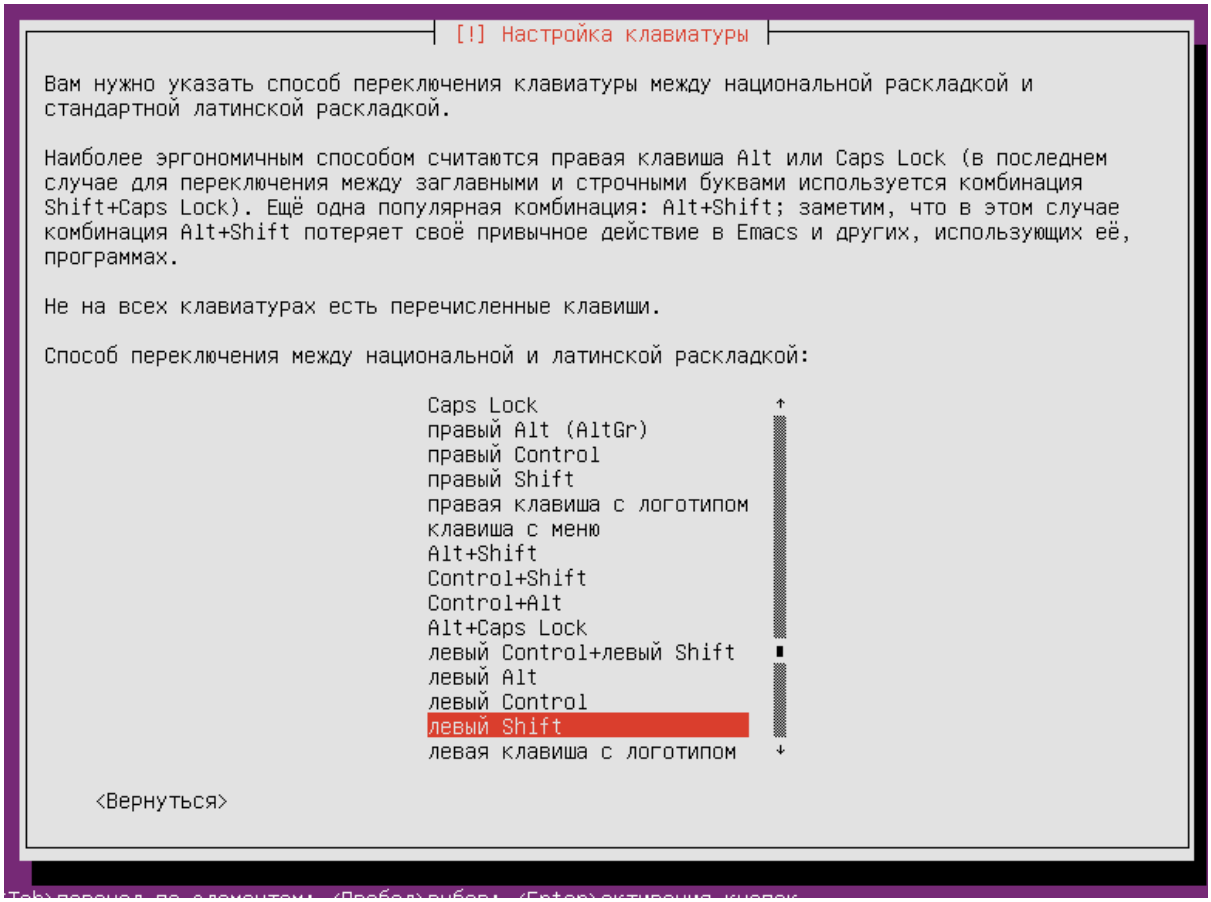
Выбираем страну, для которой предназначена клавиатура, точнее говоря, за какой Вы сидите:



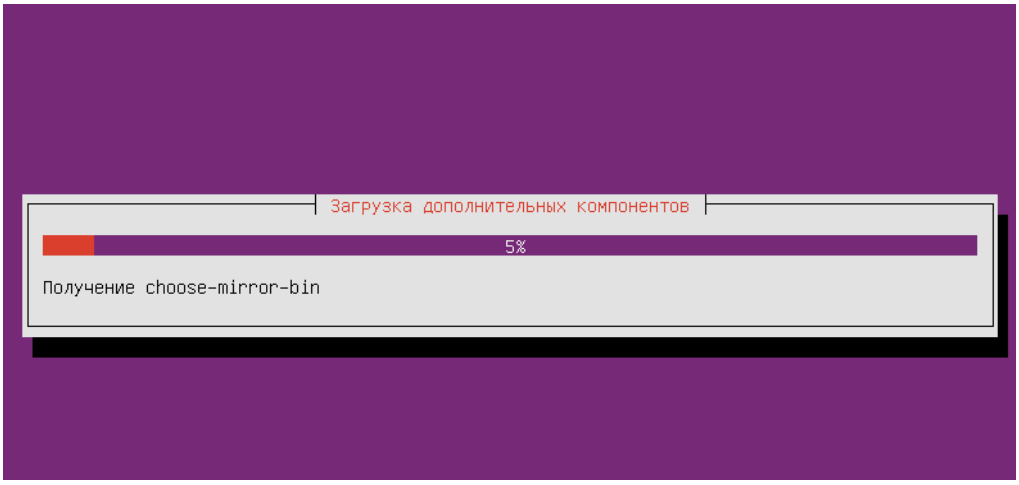
Выбираем раскладку для клавиатуры. Здесь стоит тыркнуть в пункт просто “**Русская**”, ну или выбрать нужную Вам:



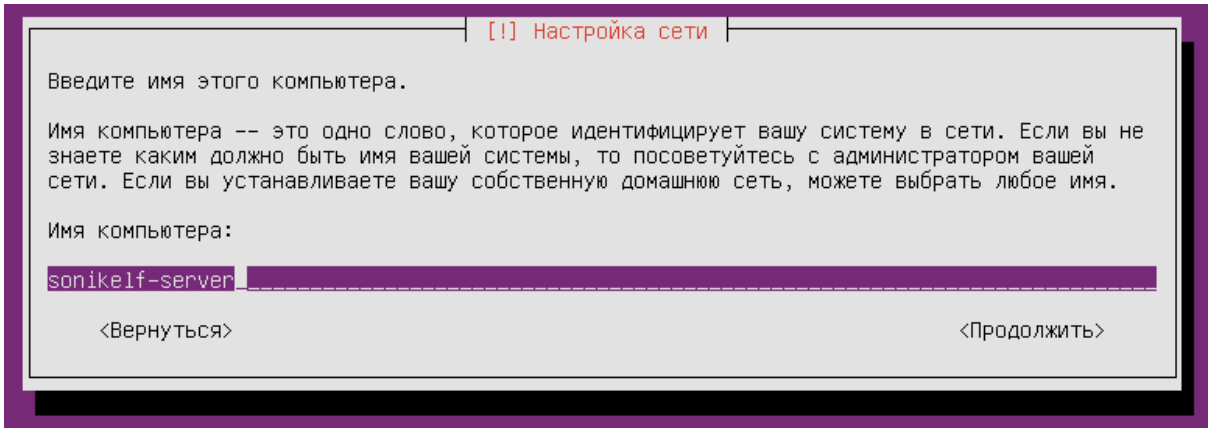
В следующем окне вам будет предложено настроить переключение раскладок. Выбирайте на свое усмотрение, я же выбрал левый **Shift**, так как уже привык именно к этой комбинации:



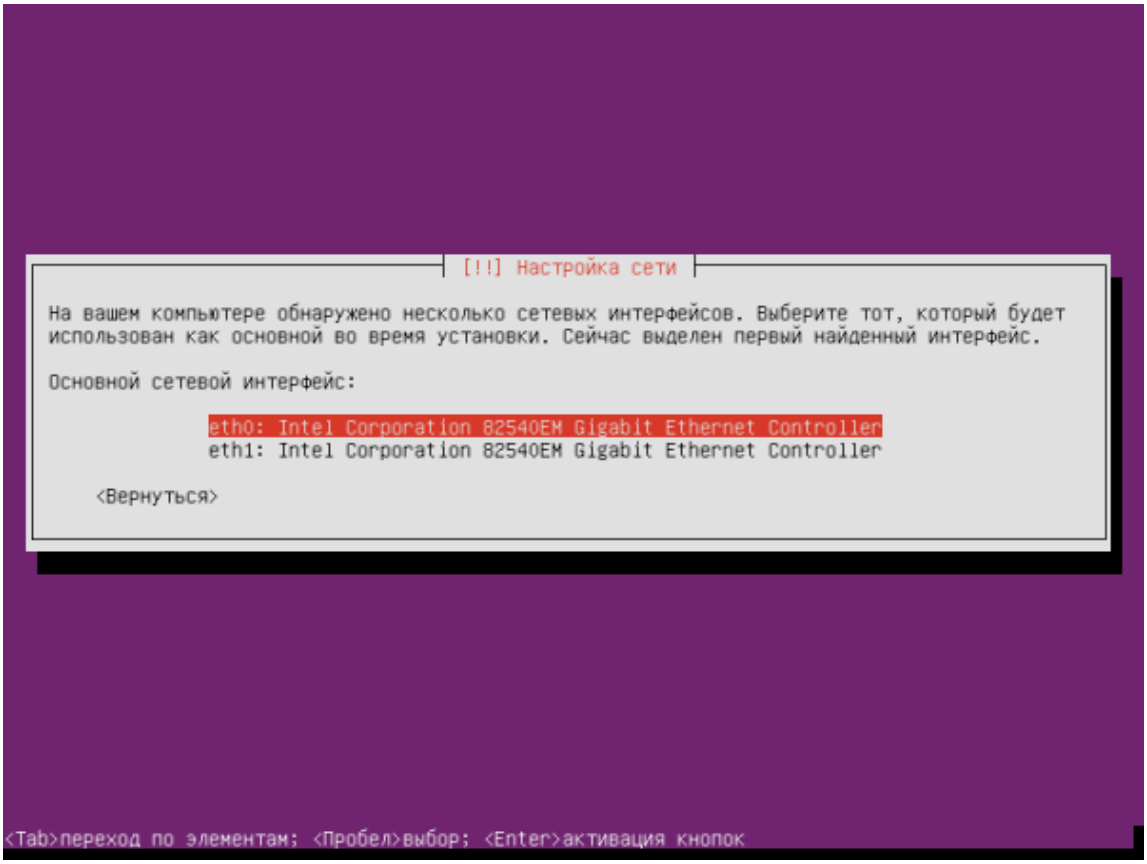
Далее потребуется подождать, прежде чем загрузятся дополнительные компоненты. После загрузки компонентов у вас появится окно выбора основного сетевого интерфейса.



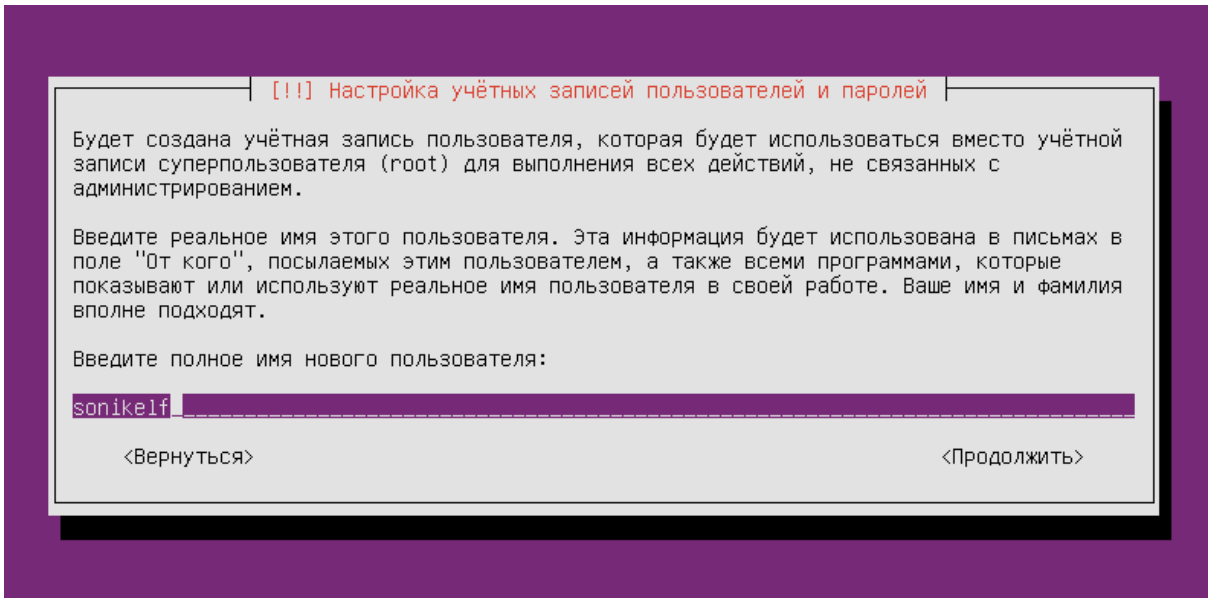
Следом потребуется задать имя компьютера, точнее говоря сервера, в сети (как мы с Вами задавали его для **Windows**, например):



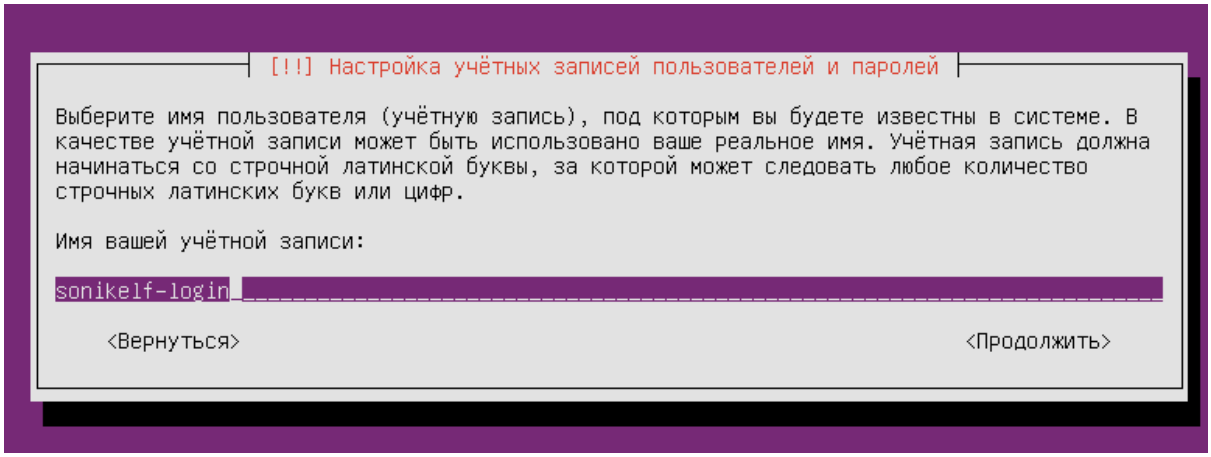
Возможно, что перед этим (или после) потребуется выбрать сетевой интерфейс (если у Вас несколько сетевых интерфейсов). Для примера выберу **etho**, Вам же надо будет сделать выбор, в соответствии с Вашими возможностями и потребностями, или не делать его вообще:



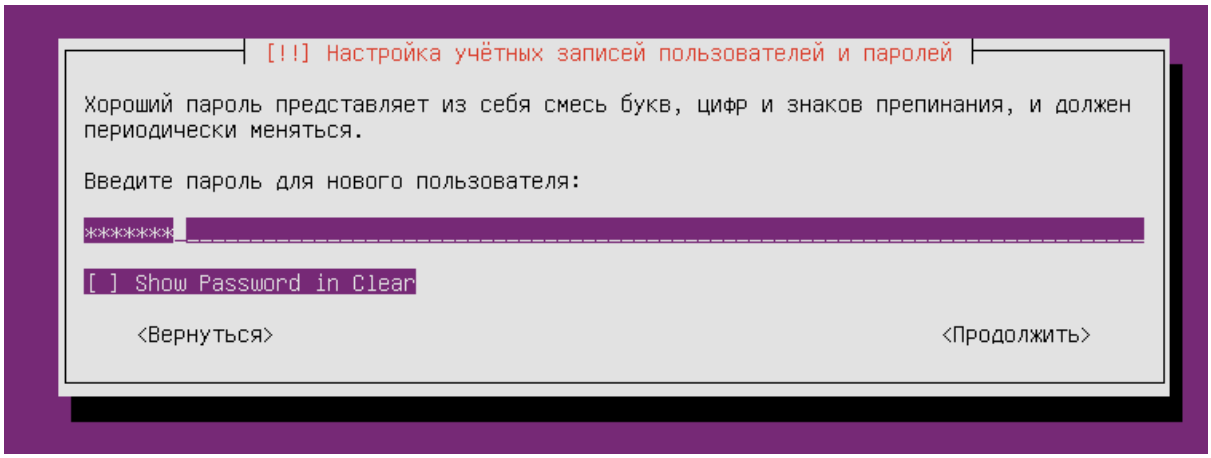
Следующим шагом потребуется указать имя пользователя (не путать с именем учетной записи и логином):



А вот в следующем окне укажите имя пользователя (логин), с помощью которого будет осуществляться вход в систему:

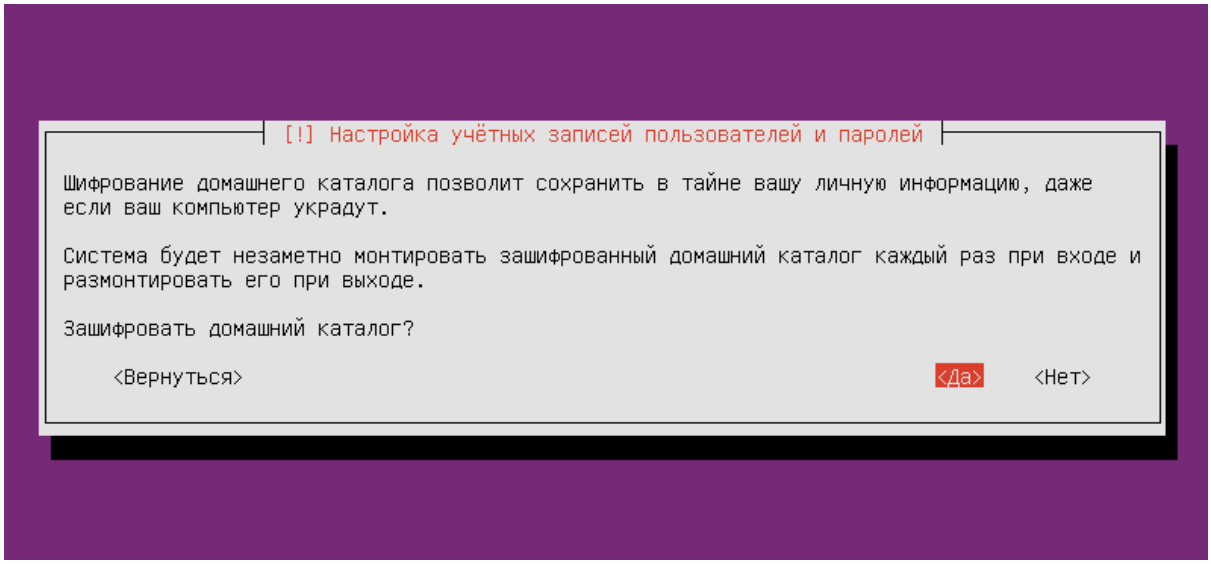


После ввода логина, придумайте и укажите пароль (желательно использовать пароль, содержащий строчные и прописные буквы, а также цифры и символы, т.е, опять же, см.прошлую книгу или статьи на сайте **sonikelf.ru**).

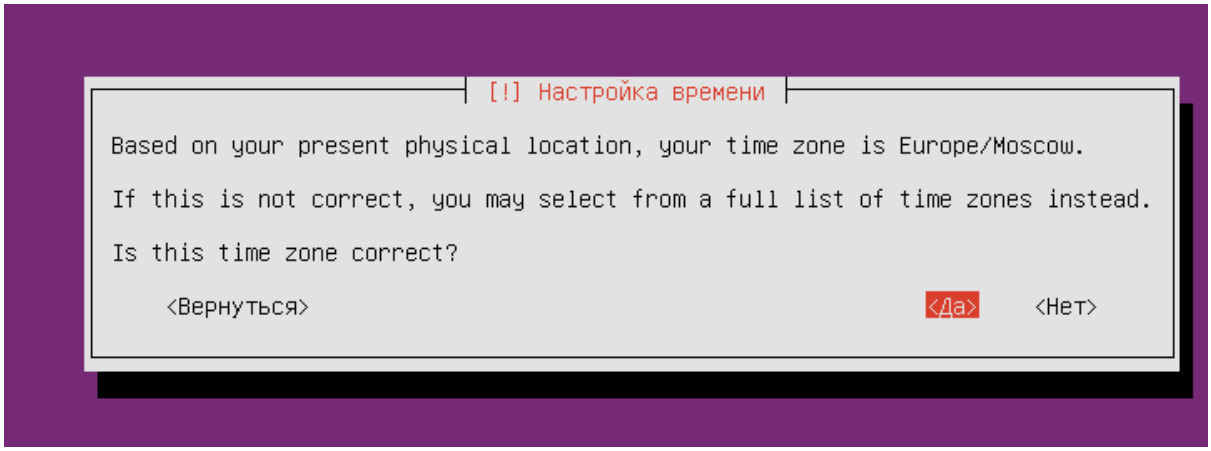


После ввода пароля, в следующем окне его нужно будет повторить, для избежания ошибок, в общем-то всё, как всегда.

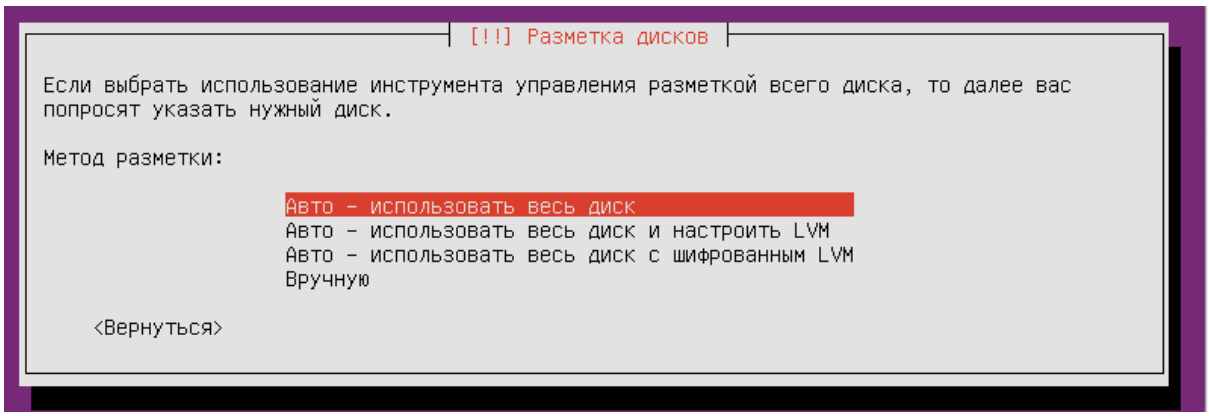
Дальше будет предложено сделать шифрование. Вообще говоря, это полезная процедура для сервера, да и рабочей станции вообще, но может сказаться на производительности. Выбор, как всегда, за Вами, но я предпочитаю шифрование, ибо безопасность наше всё.

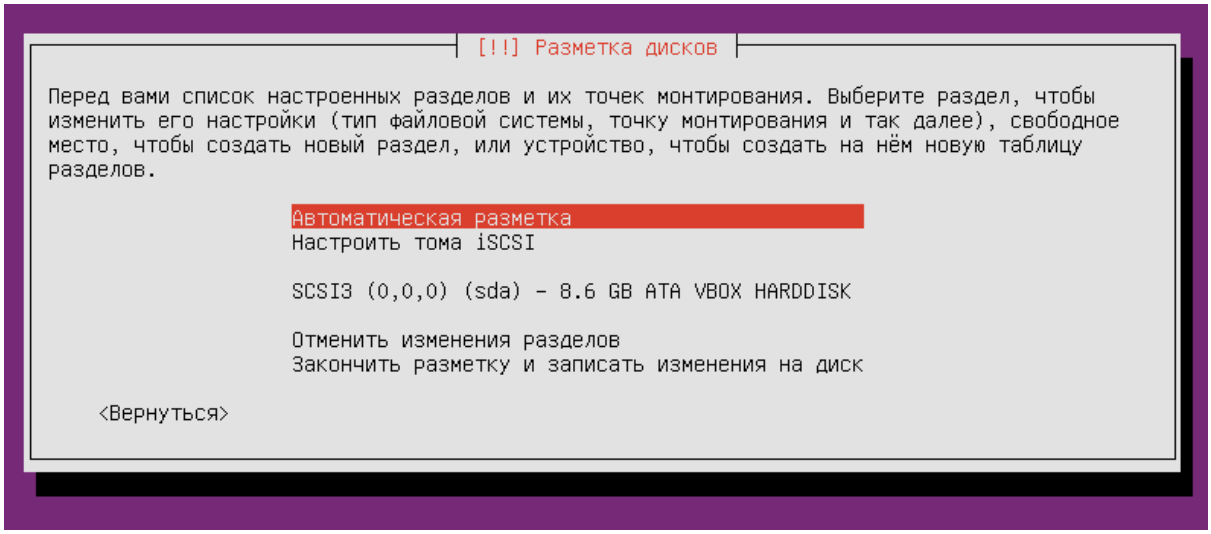


Далее нужно выбрать “часовую зону”. Как правило, если Вы задали сетевой интерфейс с доступом в интернет, оно покажет Вами корректное значение. Если же нет, то задайте его вручную.

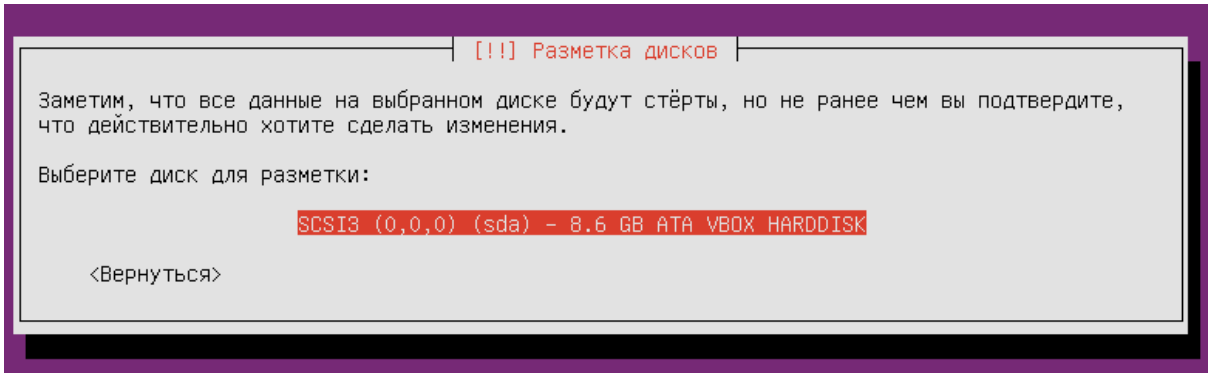


Следующим этапом будет идти разметка дисков, т.е куда будет устанавливаться система, на какой раздел, какие разделы будут вообще и всё такое прочее. Многих, в отличии от ситуации с **Windows**, этот этап вводит в ступор, но не нужно бояться:

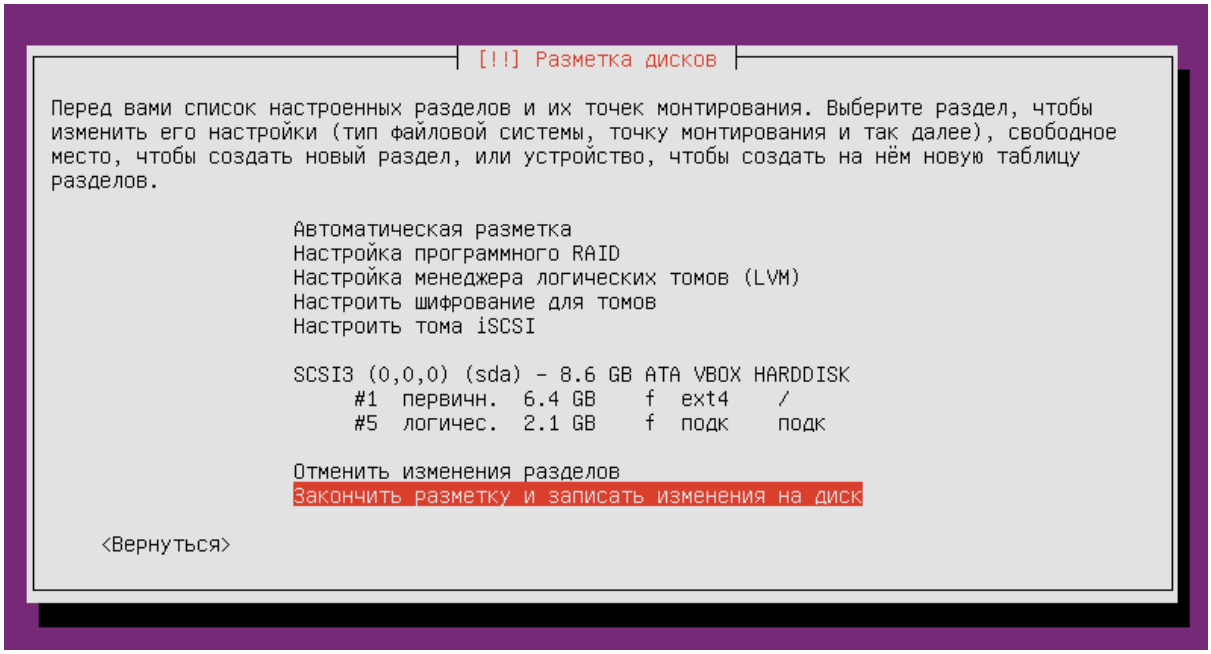




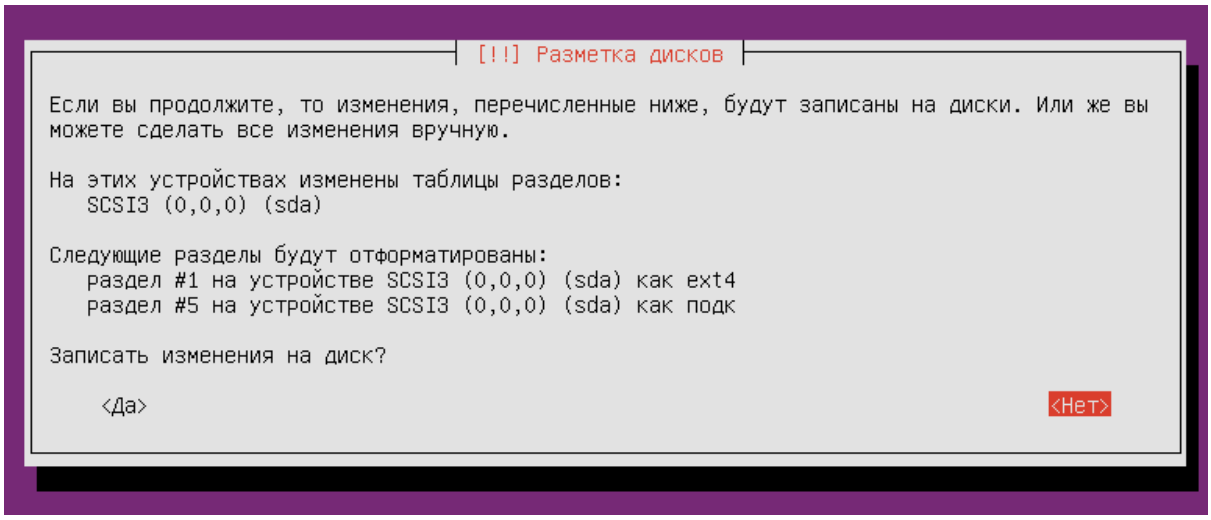
Так как мы устанавливаем данный сервер в целях обучения и в виртуалке, то выберем второй пункт “**Авто использовать весь диск**” и/или “**Автоматическая разметка**”, но если вы устанавливаете систему на реальный жесткий диск, советую устанавливать каталоги **/usr**, **/var**, **/home** на разные логические диски (разделы), что не сложно.



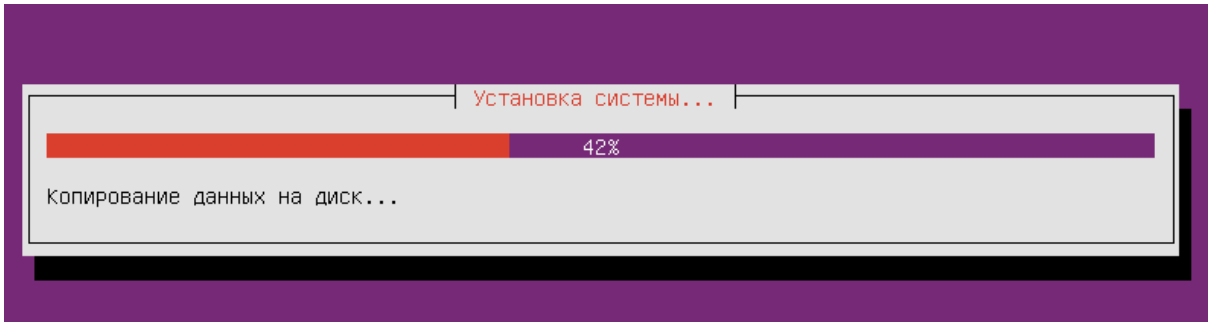
Далее надо будет выбрать диск, куда будем делать установку и автоматическую установку. В данном случае выбираем один единственный, но вариантов может быть и несколько (если в сервере стоит несколько дисков, что логично).



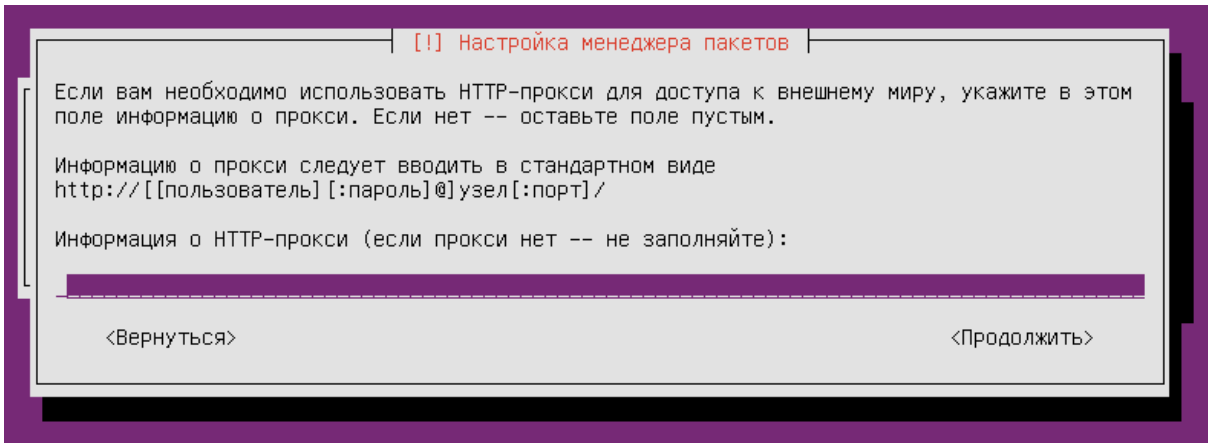
Закончив с разметкой, нужно будет согласиться на предупреждение о записи информации о разделах, жмем “Да” (прочитав предварительно уведомление, ибо информация на диске будет полностью утеряна из-за переразметки):



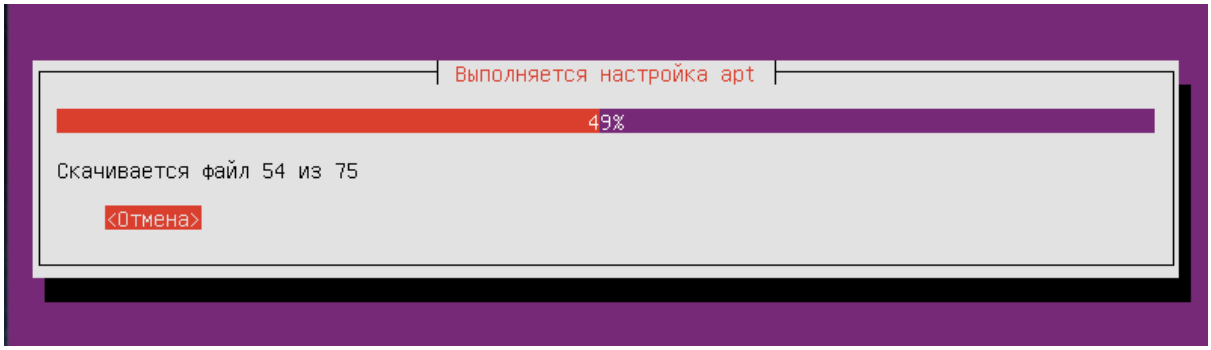
На сим процесс будет закончен и начнется установка системы, которую следует, что логично, дождаться:



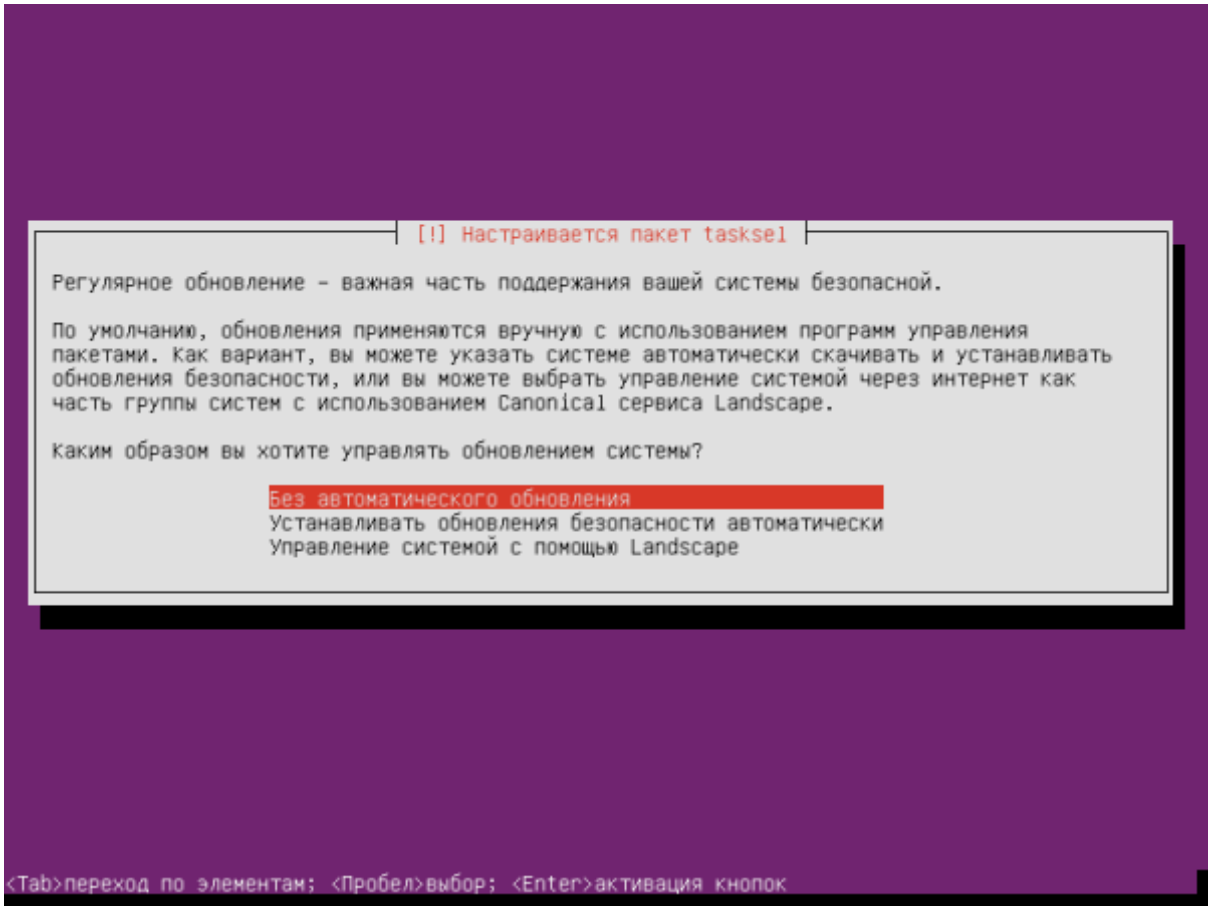
Если во время установки к сети виртуальной машины был подключен интернет, система попытается установить обновления, но перед этим спросит есть ли у вас прокси, если у вас нет прокси (о том, что это я писал в статье “**Что такое прокси и как сделать себя анонимным интернете [часть 1]**”), а интернет идет напрямую, то жмите “**Продолжить**”, ничего не указывая :



Далее начнется еще один процесс установки:

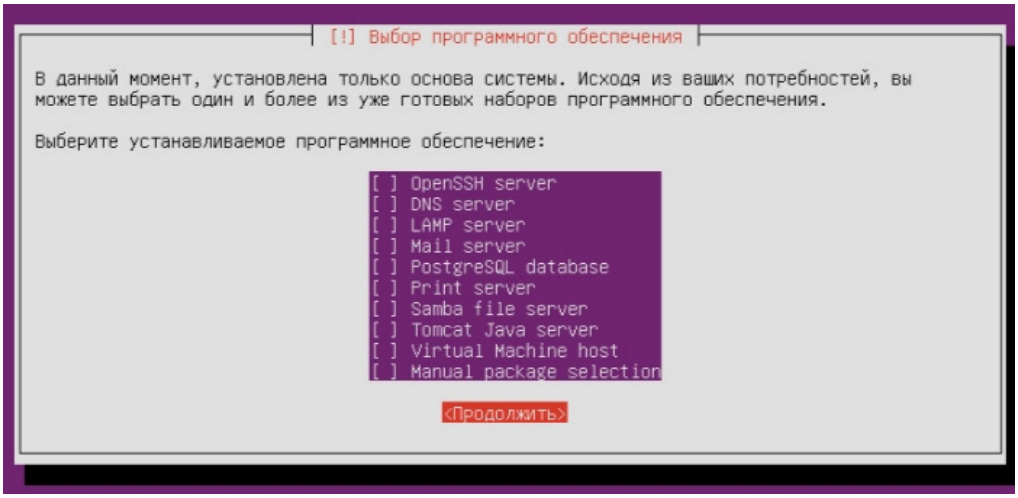


После чего Вас спросят про автоматические обновления. На данном этапе я обычно выбираю “**Без автоматического обновления**”, ибо не люблю, когда что-то ставится без моего ведома, кушает трафик, ресурсы и прочее, лучше всё сделать ручками.

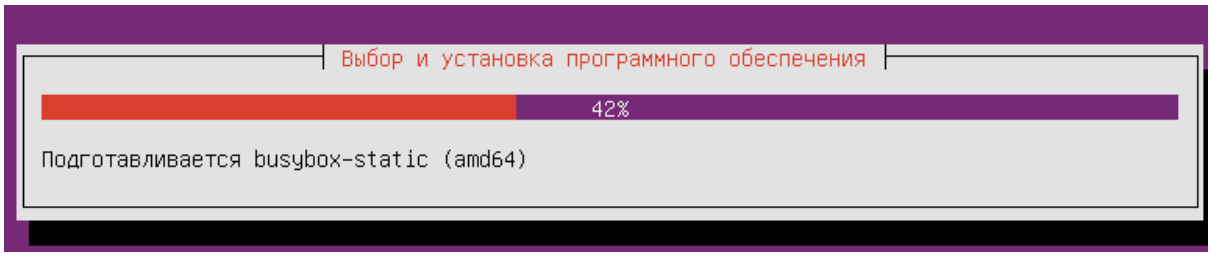


Следующим окном будет предложено выбрать, что именно устанавливать. Здесь как раз указана часть ролей, которые могут

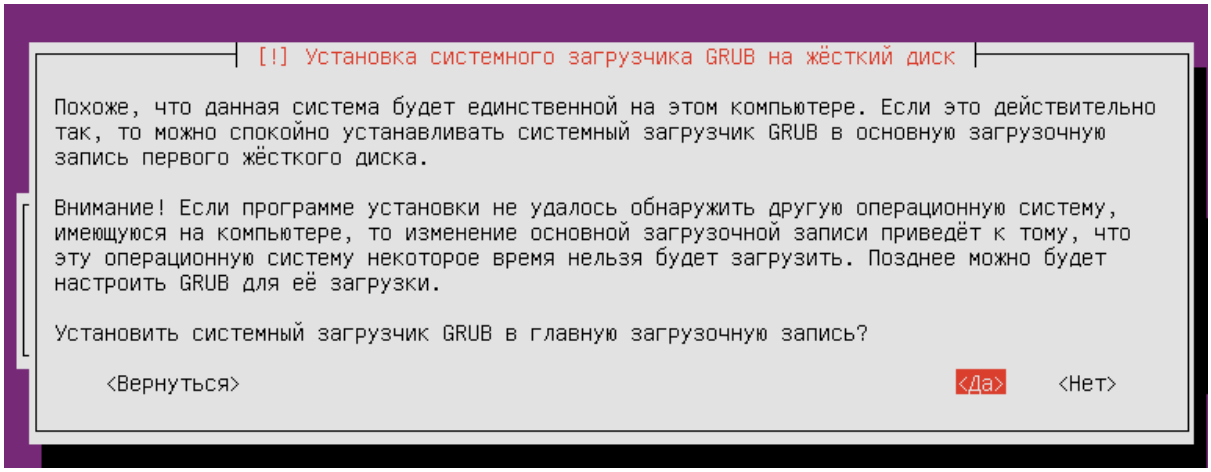
быть установлены, что называется “из коробки”, либо поставлены позже вручную.



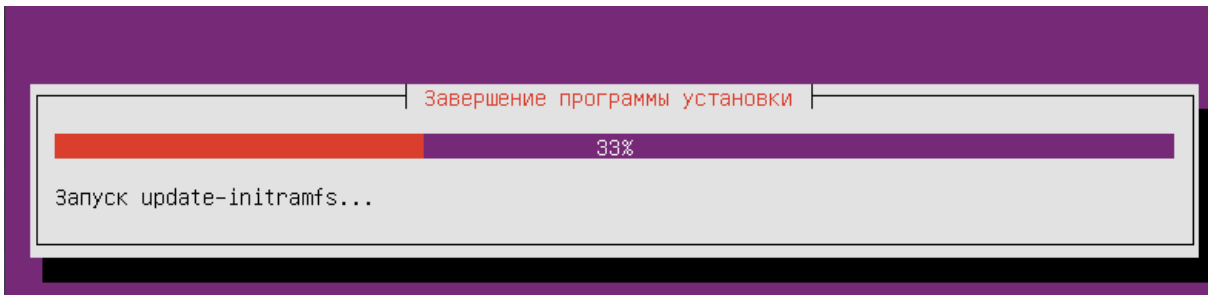
Я сразу выбрал здесь **LAMP** server (это роль веб-сервера и соответствующее программное обеспечение для него), **OpenSSH**-сервер (для удаленного доступа), **Samba File Server** (роль файлового сервера и соответствующее программное обеспечение для него) и “**Standard system utilities**”.



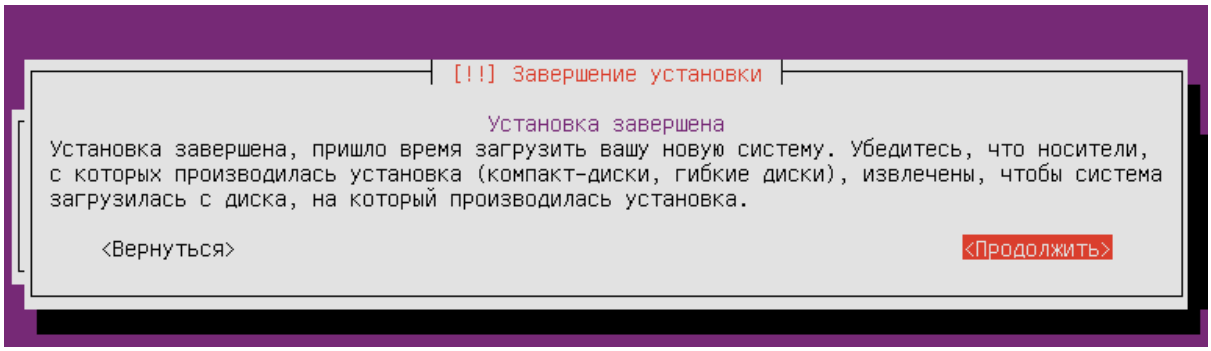
В самом конце установки сервера вам нужно согласиться с установкой загрузчика в главную загрузочную запись, что жизненно необходимо.



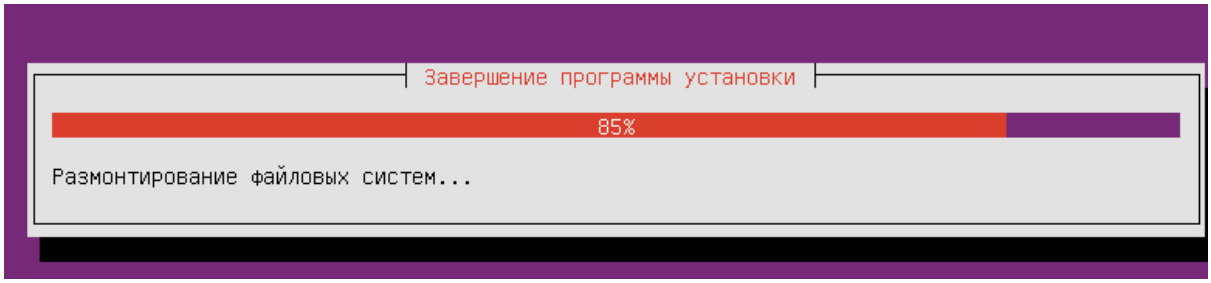
Если же Вы устанавливаете несколько систем наживую, то с загрузчиком надо быть осторожнее, возможно стоит оставить тот, что уже установлен **Windows** (если она у Вас стоит первой системой), а может и нет (зависит от версии тоже **Windows**, диска(ов), куда идет установка и других формальностей)



После установки загрузчика, если всё прошло успешно, Вы увидите сообщение об успешной установке системы:

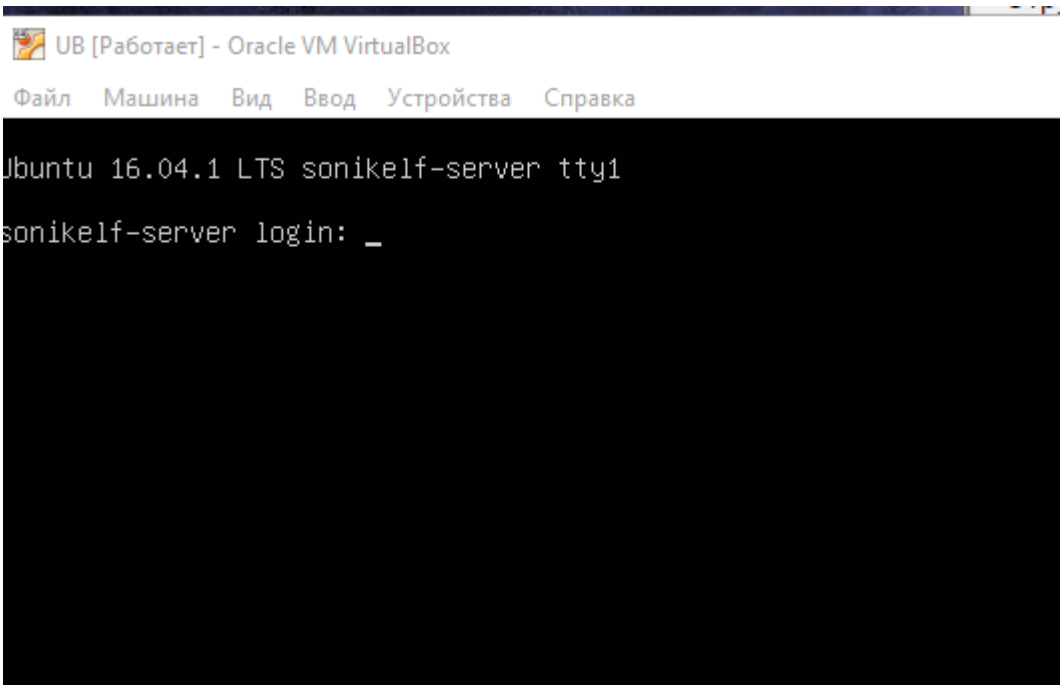


Теперь жмём в “**Продолжить**” и ждем, пока закончится процесс перезагрузки.



```
      Starting Flush Journal to Persistent Storage...
      Starting Load/Save Random Seed...
      Starting udev Coldplug all Devices...
 OK  ] Started Load/Save Random Seed.
 OK  ] Started Flush Journal to Persistent Storage.
 OK  ] Started udev Coldplug all Devices.
 OK  ] Started Dispatch Password Requests to Console Directory Watch.
 OK  ] Started Monitoring of LVM2 mirrors,...ng dmeventd or progress polling.
 OK  ] Reached target Local File Systems (Pre).
 OK  ] Reached target Local File Systems.
      Starting Tell Plymouth To Write Out Runtime Data...
      Starting Create Volatile Files and Directories...
      Starting Commit a transient machine-id on disk...
      Starting Set console font and keymap...
      Starting LSB: AppArmor initialization...
 OK  ] Started Create Volatile Files and Directories.
      Starting Network Time Synchronization...
      Starting Update UTMP about System Boot/Shutdown...
 OK  ] Started Tell Plymouth To Write Out Runtime Data.
 OK  ] Started Network Time Synchronization.
 OK  ] Started Update UTMP about System Boot/Shutdown.
 OK  ] Reached target System Time Synchronized.
 OK  ] Started Commit a transient machine-id on disk.
 OK  ] Found device 82540EM Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter).
 OK  ] Started Set console font and keymap.
 OK  ] Created slice system-getty.slice.
 OK  ] Found device VBOX_HARDDISK 5.
      Starting Cryptography Setup for cryptswap1...
 OK  ] Listening on Load/Save RF Kill Switch Status /dev/rfkill Watch.
```

После первой загрузки вы должны увидеть предложение ввода логина, указанного при установке, а также пароля:



Если вы ввели правильные логин и пароль, вы попадете в систему и увидите такой вот экран:

```
sonikelf-server login: sonikelf-login
Password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Могут быть обновлены 72 пакета.
22 обновления касаются безопасности системы.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sonikelf-login@sonikelf-server:~$
```

На этом базовую установку можно считать завершенной. Давайте обновим систему, так как, как видите, могут быть обновлены **72** пакета и **22** патча для безопасности. Можно сделать это, используя команды:

```
sudo apt-get update
```

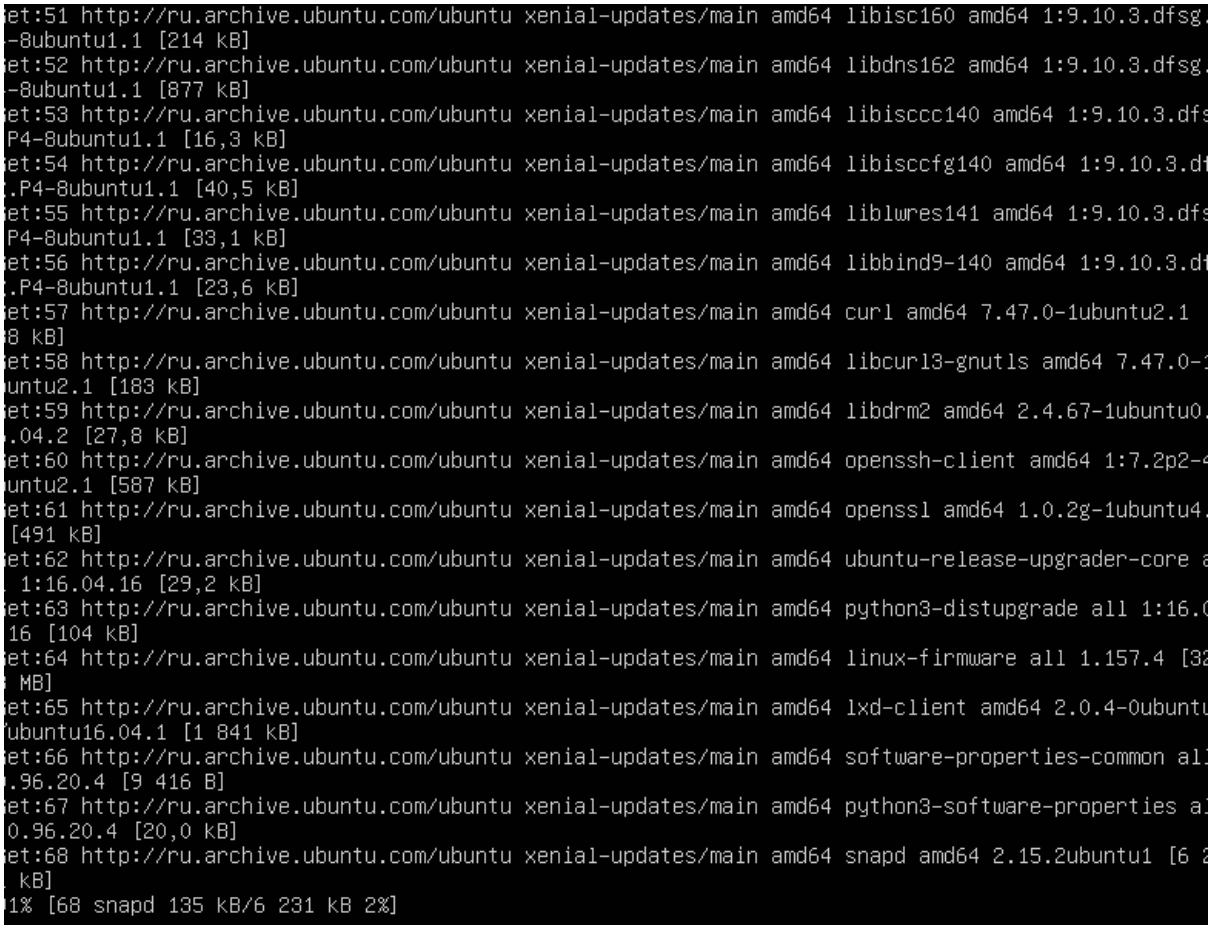
```
sudo apt-get upgrade
```

Первой командой мы обновим индекс обновлений, а второй поставим сами обновления:

```
sonikelf-login@sonikelf-server:~$ sudo apt-get update
Hit:1 http://ru.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://ru.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://ru.archive.ubuntu.com/ubuntu xenial-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [94,5 kB]
Получено 94,5 kБ за 0с (165 kБ/с)
Чтение списков пакетов… Готово
sonikelf-login@sonikelf-server:~$ _
```

```
sonikelf-login@sonikelf-server:~$ sudo apt-get update
Hit:1 http://ru.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://ru.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://ru.archive.ubuntu.com/ubuntu xenial-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [94,5 kB]
Получено 94,5 kБ за 0с (165 kБ/с)
Чтение списков пакетов… Готово
sonikelf-login@sonikelf-server:~$ sudo apt-get upgrade
Чтение списков пакетов… Готово
Построение дерева зависимостей
Чтение информации о состоянии… Готово
Расчёт обновлений… Готово
Пакеты, которые будут оставлены в неизменном виде:
  linux-generic linux-headers-generic linux-image-generic ubuntu-core-launcher
Пакеты, которые будут обновлены:
  accountsservice apparmor bind9-host cloud-initramfs-copymods cloud-initramfs-dyn-netconf curl
  dnsutils fuse gcc-5-base gnupg gpgv grub-legacy-ec2 ifupdown init init-system-helpers
  initramfs-tools initramfs-tools-bin initramfs-tools-core isc-dhcp-client isc-dhcp-common kbd
  klibc-utils language-selector-common less libaccountsservice0 libapparmor-perl libapparmor1
  libbind9-140 libcurl3-gnutls libdns-export162 libdns162 libdrm2 libfuse2 libgcrypt20 libidn11
  libisc-export160 libisc160 libisccc140 libisccfg140 libklibc liblwres141 liblxc1 libp11-kit0
  libpam-systemd libpython3.5 libpython3.5-minimal libpython3.5-stdlib libssl1.0.0 libstdc++6
  libsystemd0 libudev1 linux-firmware lxc-common lxcfs lxd lxd-client open-iscsi openssh-client
  openssl overlayroot python3-distupgrade python3-software-properties python3.5 python3.5-minimal
  snapd software-properties-common sudo systemd systemd-sysv tzdata ubuntu-release-upgrader-core
  udev
обновлено 72, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 4 пакетов не обновлено
.
Необходимо скачать 64,9 МБ архивов.
После данной операции, объём занятого дискового пространства возрастёт на 10,1 MB.
Хотите продолжить? [Д/н] y_
```

Система проверит в репозиториях, какие обновления требуется установить, сколько будет затрачено на закачку и тп, а также спросит Вашего разрешения на установку. Введите **Д** и нажмите Enter или **Y** и нажмите **Enter**.



После этого начнется процесс загрузки, распаковки, обновления и установки пакетов, обновления для них и прочего.

Теперь у нас есть сервер с установленными на нем ПО для различных ролей и тп. В принципе, можно начинать работать, но так как вряд ли Вы сходу разберетесь в консольных командах, давайте мы с Вами установим графический интерфейс.

Графический интерфейс в Ubuntu Server - установка и использование

Небольшая вводная

Про графические оболочки и другие нюансы, в виде базиса, было написано в статье “**Графические оболочки в Linux [основы основ, работа в KDE]**” на **sonikelf.ru**. Рекомендую ознакомиться, чтобы понимать, что **GUI** (графических оболочек), в отличии от **Windows**, бывает несколько и тд и тп.

Как Вы поняли, по умолчанию **GUI** отсутствует в серверной версии системы и её надо установить. Можно установить оболочку отдельно, но вообще говоря, новичкам я бы не рекомендовал это делать по ряду причин, одна из которых связана с некорректностью или неполной установкой, что приведет к множеству дальнейших проблем.

Установка оболочки и сопутствующих “хвостов”

Самый простой способ, хотя не самый удачный, это проапгрейдить систему до версии десктопной, со всем графическими причудами, т.е ввести в консоли команду:

```
sudo apt-get install ubuntu-desktop
```

Что установит графическую оболочку unity и кучу дополнительных пакетов от настольной версии Ubuntu. Я бы не рекомендовал ставить таковую на сервер в виду прожорливости и других факторов.

Можно еще ввести:

```
sudo apt-get install lubuntu-desktop
```

Это развернет **lubuntu**, куда встроена графическая оболочка **LXDE**, которая отличается низкими требованиями к системе и высоким быстродействием. Если вам нужен простой непрожорливый интерфейс без красивых визуальных эффектов, но со всем необходимым функционалом, то **lubuntu** вам подойдет.

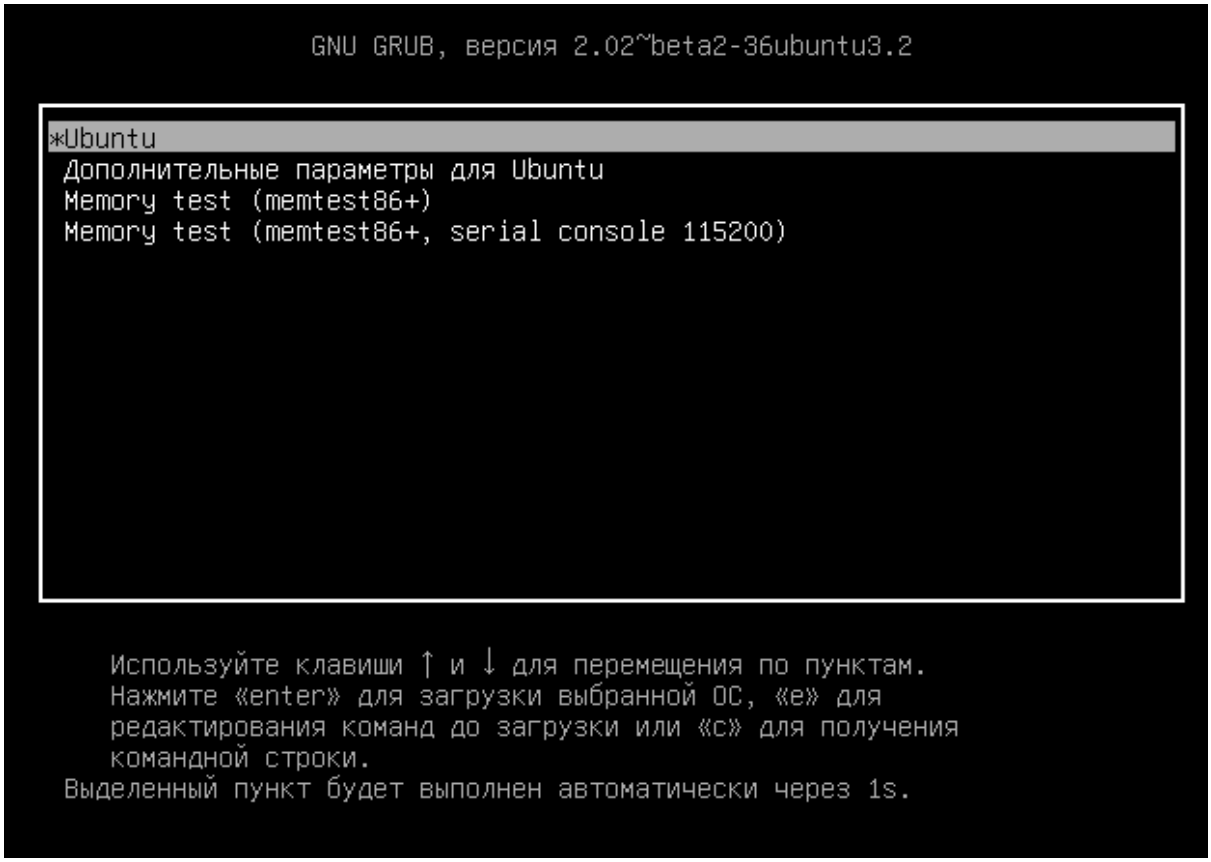
Но вообще, на мой вкус (хотя я вообще больше предпочитаю консоль), лучше всего подойдет **kubuntu**:

```
sudo apt-get install kubuntu-desktop
```

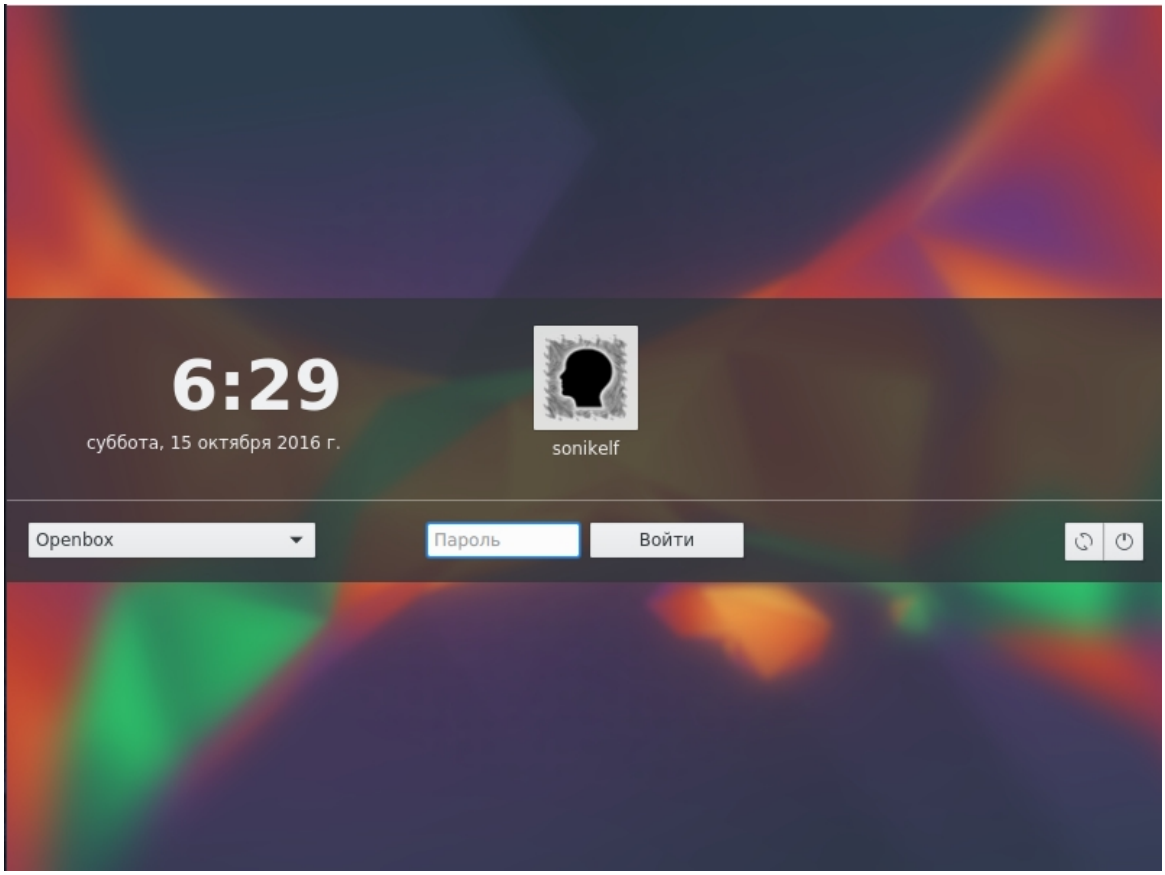
Как наиболее адекватную из всех представленных, да и в комплекте идет минимум необходимого ПО без лишних приблуд. Введите одну из команд в консоли, подтвердите установку и дождитесь её окончания.

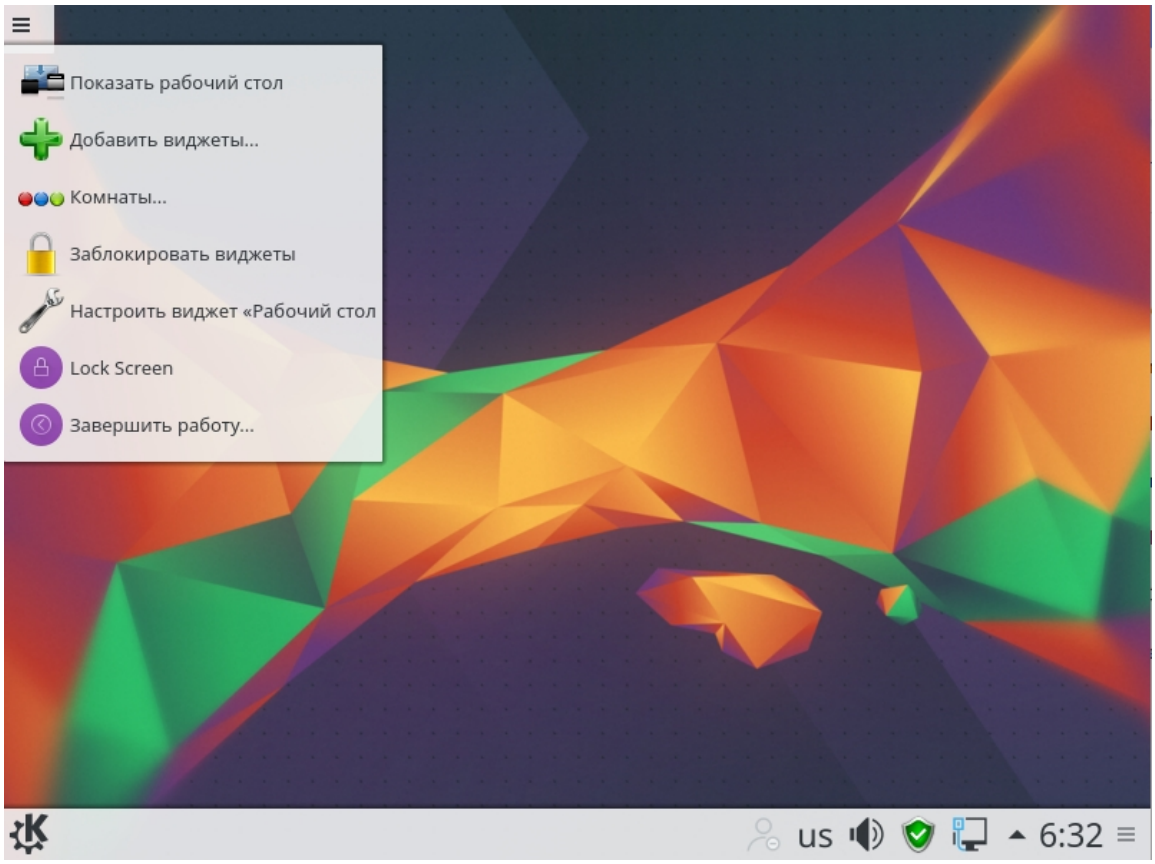
```
ыбор ранее не выбранного пакета kde-config-whoopsie.
одготовка к распаковке …/kde-config-whoopsie_15.10ubuntu2_amd64.deb …
аспаковывается kde-config-whoopsie (15.10ubuntu2) …
ыбор ранее не выбранного пакета liboxygenstyle5-5.
одготовка к распаковке …/liboxygenstyle5-5_4%3a5.5.5-0ubuntu1_amd64.deb …
аспаковывается liboxygenstyle5-5 (4:5.5.5-0ubuntu1) …
ыбор ранее не выбранного пакета liboxygenstyleconfig5-5.
одготовка к распаковке …/liboxygenstyleconfig5-5_4%3a5.5.5-0ubuntu1_amd64.deb …
аспаковывается liboxygenstyleconfig5-5 (4:5.5.5-0ubuntu1) …
ыбор ранее не выбранного пакета kde-style-oxygen-qt5.
одготовка к распаковке …/kde-style-oxygen-qt5_4%3a5.5.5-0ubuntu1_amd64.deb …
аспаковывается kde-style-oxygen-qt5 (4:5.5.5-0ubuntu1) …
ыбор ранее не выбранного пакета kio-mtp.
одготовка к распаковке …/kio-mtp_0.75+git20140304-2_amd64.deb …
аспаковывается kio-mtp (0.75+git20140304-2) …
ыбор ранее не выбранного пакета kscreen.
одготовка к распаковке …/kscreen_4%3a5.5.5-0ubuntu1_amd64.deb …
аспаковывается kscreen (4:5.5.5-0ubuntu1) …
ыбор ранее не выбранного пакета libauthen-sasl-perl.
одготовка к распаковке …/libauthen-sasl-perl_2.1600-1_all.deb …
аспаковывается libauthen-sasl-perl (2.1600-1) …
ыбор ранее не выбранного пакета libc6-dbg:amd64.
одготовка к распаковке …/libc6-dbg_2.23-0ubuntu3_amd64.deb …
аспаковывается libc6-dbg:amd64 (2.23-0ubuntu3) …
ыбор ранее не выбранного пакета libkf5modemmanagerqt6:amd64.
одготовка к распаковке …/libkf5modemmanagerqt6_5.18.0-0ubuntu1_amd64.deb …
аспаковывается libkf5modemmanagerqt6:amd64 (5.18.0-0ubuntu1) …
ыбор ранее не выбранного пакета libksane0.
одготовка к распаковке …/libksane0_4%3a15.08.2-0ubuntu1_amd64.deb …
аспаковывается libksane0 (4:15.08.2-0ubuntu1) …
ыбор ранее не выбранного пакета liblouis-data.
одготовка к распаковке …/liblouis-data_2.6.4-2_all.deb …
аспаковывается liblouis-data (2.6.4-2) …
ыбор ранее не выбранного пакета liblouis9:amd64.
одготовка к распаковке …/liblouis9_2.6.4-2_amd64.deb …
аспаковывается liblouis9:amd64 (2.6.4-2) …
```

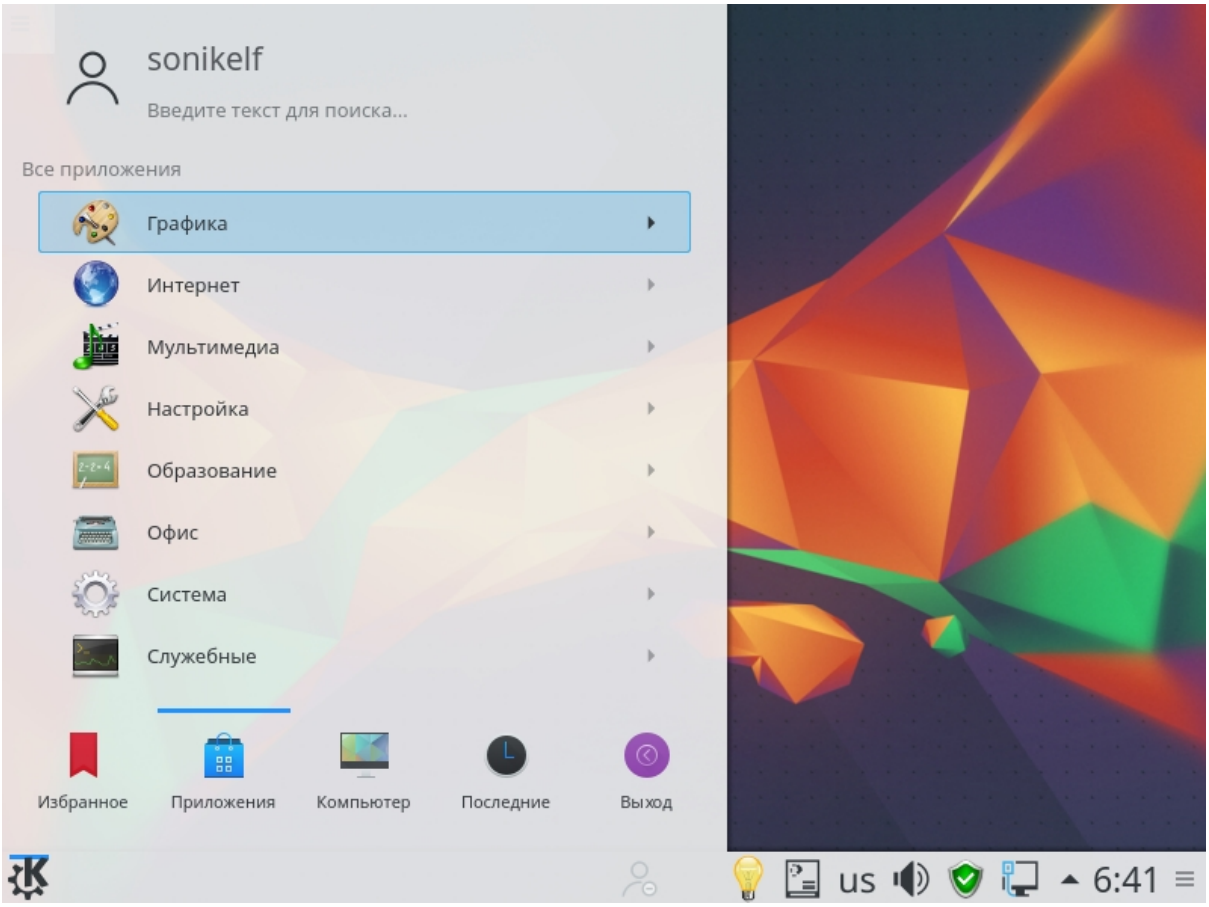
Процесс может занять достаточно большое количество времени, т.к ставится не только графическая оболочка, но и прочие необходимые вещи, потом все это настраивается, предварительно скачивается и прочее прочее.



После установки введите в консоли **reboot** и перезагрузите систему. По завершению перезагрузки Вы увидите окно со входом в профиль выбранного пользователя и выбором обработчика интерфейса (справа, где у меня написано **Openbox**).


6:29
суббота, 15 октября 2016 г.
sonikelf
Openbox
Пароль
Войти


Показать рабочий стол
Добавить виджеты...
Комнаты...
Заблокировать виджеты
Настроить виджет «Рабочий стол
Lock Screen
Завершить работу...
us
6:32



Теперь Вы можете управлять сервером с помощью визуальной составляющей, что пусть и не избавляет Вас полностью от необходимости использования консоли, но существенно упростит жизнь для новичков.

Теперь давайте кратенько познакомимся с базисом, который хорошо бы знать, прежде чем работать с системой дальше.

Root и другие пользователи [+группы] в Linux

Давайте познакомимся поближе с тем, как устроена система пользователей и управления в Linux.

Это многопользовательская система

Итак, еще раз. **Linux** является многозадачной многопользовательской (сколько много мног'ов :)) операционной системой. Это означает, что одновременно с системой могут работать несколько пользователей, да и каждый из них может запускать несколько приложений. И при этом, например, Вы можете зайти в систему локально, а другой пользователь – удаленно, воспользовавшись протоколами удаленного доступа (**telnet, ssh, ftp**).

Это очень удобно. Приведем пример, чтобы было понятно, о чем идет речь. Предположим, Вам понадобилось распечатать документ, который есть на рабочем компьютере, а на домашнем его нет. При наличии интернета и компьютера, настроенного соответствующим образом, это не составит особого труда, ибо Вы спокойно зайдете в систему, скопируете нужный Вам файл и распечатаете его там, где вам это удобно. И даже если кто-то в момент Вашего подключения уже работает с системой, то таки Вы не будете мешать друг другу.

В **Windows** такая возможность тоже есть, но только после установки соответствующего программного обеспечения, и при этом Вы не сможете войти в систему, чтобы "не помешать" другому пользователю, который уже в ней находится. Вместе вы работать не сможете, ибо будете мешать друг другу. Всё, что Вы будете делать, будет отражаться на мониторе пользователя (т.е. рабочий стол

получится как бы общий), и если не предупредить пользователя, что Вы удаленно зашли в систему, он подумает, что с его компьютером что-то не так (случай, конечно, не 100%, но в большинстве своём это факт).

> В **Linux**-системах несколько пользователей могут работать с системой и даже не подозревать о существовании друг друга, пока не воспользуются командой (например, **who**). **Linux** изначально разрабатывалась как многопользовательская система с возможностью удаленного подключения как в текстовом, так и в графическом режиме. **Windows** в основе своей всегда была (и, наверное, долго останется) системой персонального, ни с чем совсем сильно не связанного компьютера, а средства взаимодействия по локальной сети всегда шли в виде своего рода «довеска». Кроме того, **Linux** всегда опиралась на открытые протоколы, большая часть которых является стандартом в мире **Unix**-систем. **Microsoft**, напротив, делает ставку на собственные протоколы, естественно, закрытые и не всегда совместимые со своими предыдущими версиями.

Суперпользователь (root) в Linux

Вот мы и выяснили (точнее говоря, немного прояснили), что представляет из себя многопользовательская система. Теперь поговорим о том, кто же такой в **Linux**-системах пользователь **root**.

Условно и "сленгово" говоря - это царь и бог :). Пользователь root обладает ВСЕМИ полномочиями в системе, вплоть до её полного и безвозвратного удаления, т.е. это, так сказать, системный администратор в отдельно взятой операционной системе.

Любая команда, введенная в терминале, таки будет выполнена безоговорочно и бесповоротно. Так что работать под root-ом в любых **Linux (unix)**-системах не приветствуется и даже, можно сказать, не принято. Например, кто-то решил "подшутить" и выложил в интернете вредоносную программу, а Вы взяли и скачали её, да еще и начали устанавливать ее на своем компьютере (трудно представить себе таких экстремалов, но все же? - а вдруг найдутся такие). Если запуск этой программы будет произведен через суперпользователя, то Ваша система может быть легко уничтожена, а вот запуск этой программы от имени простого пользователя ни к чему не приведет - система просто откажется выполнять ее, сославшись на то, что у Вас не хватает полномочий. Это лишний повод присмотреться, а нужно ли Вам вообще это устанавливать. Вот поэтому в **Linux (unix)**-системах вход под именем пользователя **root** запрещен.

В одних дистрибутивах Вы не сможете войти как **root** в графическом режиме, но сможете зайти через консоль (грубо говоря, терминал) комбинацией клавиш Ctrl + Alt + F2 (как, например, в **Fedora**), а в других дистрибутивах Вы вообще не сможете войти как **root** - ни через графический режим, ни через комбинацию клавиш, ни через консоль (к такому дистрибутиву, например, относится **Ubuntu**). Встает вопрос - а как тогда устанавливать нужные программы, минуя этого суперпользователя? Для этого существует "временное разрешение"

на выполнение нужных операций в виде команд **sudo** или **su**. Чем же отличаются эти команды друг от друга?

Давайте начнем с **sudo**. Эта команда позволяет запустить любую команду с привилегиями **root**. Но использовать эту команду могут не все пользователи, а только те, которые внесены в файл **/etc/sudoers** (в этом файле в доступной форме расписывается каким пользователям или группам можно выполнять команду **sudo**, а это значит, что рядовой пользователь, не внесенный в этот файл, даже при сильном своем желании не сможет воспользоваться правами **root**).

> О каталоге /**etc** мы поговорим отдельно, но предварительно хочется сказать, что в этом каталоге содержатся почти все настройки системы (кроме пользовательских, пользовательские настройки хранятся в домашнем каталоге пользователя /**home**), при этом в каталоге /etc находится множество текстовых файлов. А поскольку файлы текстовые, то их можно редактировать любым текстовым редактором, что очень упрощает работу с системными файлами и повышает надежность системы. К слову, даже если удалить один из конфигурационных файлов каталога /**etc**, система продолжит работу как ни в чем не бывало! Конечно, она будет работать не так, как до удаления этого файла, но все же она, в отличие от **Windows**, будет работать.

Теория - это конечно хорошо, но давайте немного попрактикуемся. Например, вам нужно отредактировать файл **/etc/sudoers**, для чего заходим в терминал и вводим:

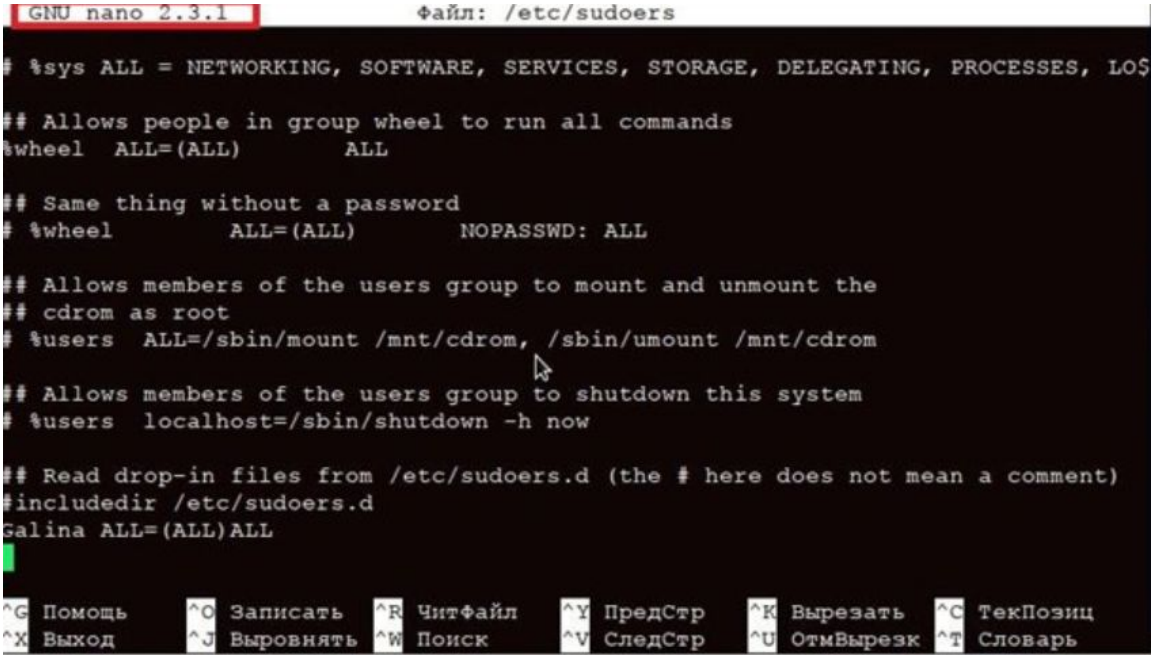
```
sudo gedit /etc/sudoers
```

```
Файл  Правка  Вид  Терминал  Переход  Справка
[galina@localhost ~]$ sudo gedit /etc/sudoers
[sudo] password for galina:
```

Откроется окно программы **gedit**:

```
root    ALL=(ALL)       ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
Galina ALL=(ALL)ALL
```

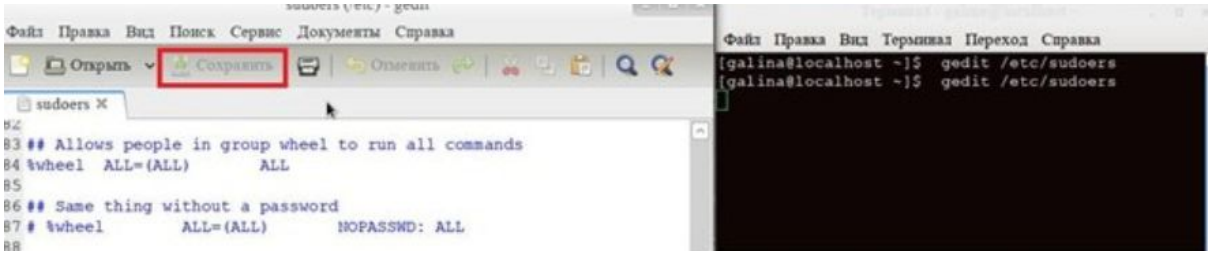
..Где мы производим нужное редактирование, сохраняем и закрываем этот файл. В **Linux**-системах очень много текстовых редакторов, с помощью которых можно редактировать файлы - это и вышеупомянутый **gedit**, и **nano**, и **vi**, и много других - посему Вы можете воспользоваться любым. Вот как будет выглядеть открытие файла через редактор **nano**:



Если мы введем без **sudo** ту же команду:

```
gedit /etc/sudoers
```

..то файл откроется, но сохранить изменения, внесенные Вами, не получится, поскольку у Вас не будет тех полномочий, которые требуются для редактирования.



Видите, кнопка «Сохранить» неактивна.

Теперь о команде su

Команда **su** позволяет получить доступ к консоли (терминалу) **root** ЛЮБОМУ пользователю (даже если этот пользователь не внесен в

список файла **/etc/sudoers**), но при условии, что он знает пароль **root**. Разницу прочувствовали?

Команда **su** предназначена для администраторов системы, а **sudo** - для остальных пользователей, которым только иногда нужны права root.

Для многих пользователей команда **su** - получение прав суперпользователя. Это не совсем верно. Команда **su** вызывает оболочку другого пользователя в системе, т.е. позволяет получить права ДРУГОГО пользователя, который должен быть указан аргументом к этой команде. Попробуйте ввести команду su root или su[имя_вашего_пользователя] и посмотрите, как среагирует эта команда. Я бы рекомендовал использовать один важный параметр "-" (тире), который существенно изменяет её работу.

При выполнении команды "**su** -" вызывается оболочка регистрации в системе. Это часто сбивает с толку новичков: **"su"** предоставляет Вам только права другого пользователя, "**su** -" позволяет войти под именем другого пользователя. Второе означает, что вместе с предоставлением прав у вас изменится текущий рабочий каталог, домашний каталог и переменные окружения. Использование команды su без параметра "-" чревато проблемами - запущенная таким образом программа сохраняет свои настройки в каталоге вызывающего, а не вызываемого пользователя (помните, Вам предоставляются только права).

Если этот пользователь **root** - Вы получите в своем домашнем каталоге файлы настроек программы, к которым не будете иметь доступа. Следует ли говорить, что запустив в дальнейшем ту же самую программу из-под своего аккаунта, Вы, в лучшем случае, не сможете сохранить свои настройки, в худшем - программа вообще не запустится. Лучше вообще не использовать команду su для запуска программ со своими переменными окружения, для этого более подходит команда sudo.

И, тем не менее, заходить в систему под суперпользователем можно всегда, даже если эта учетная запись ограничена или отключена, например, в Ubuntu нужно просто задать пароль для пользователя root командой sudo passwd root. Сначала система запросит ваш пользовательский пароль, затем новый пароль и его подтверждение. Учетную запись root в ubuntu можно потом отключить командой sudo passwd -lroot, но.. После закрытия учетной записи могут быть проблемы с входом в систему в режиме восстановления. На мой взгляд, в ubuntu суперпользователя лучше не включать, а пользоваться командой sudo -i.

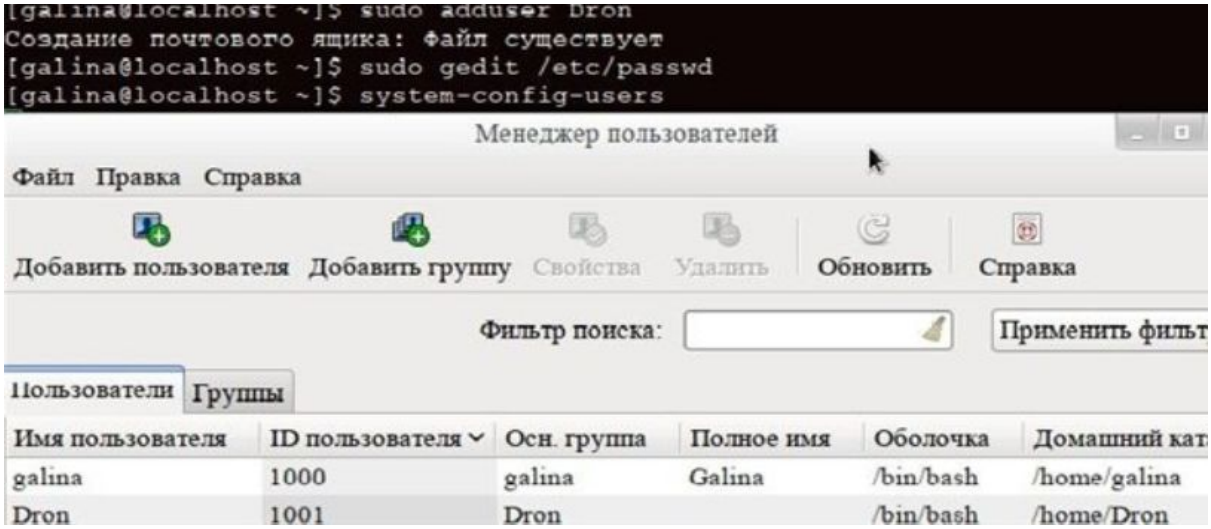
Немного практики

Давайте добавим нового пользователя в систему. Для этого нужно от суперпользователя ввести всего 2 команды:

```
adduser <имя пользователя> (добавляет пользователя)
```

```
passwd <имя пользователя> (задает/изменяет пароль)
```

Обратите внимание на знак # перед командой. Таким знаком обозначается, что команду отдает суперпользователь (он же root, он же бог и властелин, если перед командой стоит значок $ - то команда будет выполняться от простого пользователя).



Вот он, наш новый пользователь системы Dron. Мы его создали и завели пароль на эту учетную запись.

Если пароль задан слишком простой, то Вам выдается предупреждение, что пароль слишком доступный:

```
[galina@localhost ~]$ sudo passwd Dron
[sudo] password for galina:
Смена пароля для пользователя Dron.
Новый пароль :
НЕУДАЧНЫЙ ПАРОЛЬ: короткий
НЕУДАЧНЫЙ ПАРОЛЬ: слишком простой
Повторите ввод нового пароля :
Извините, но пароли не совпадают.
Новый пароль :
НЕУДАЧНЫЙ ПАРОЛЬ: слишком простой
Повторите ввод нового пароля :
```

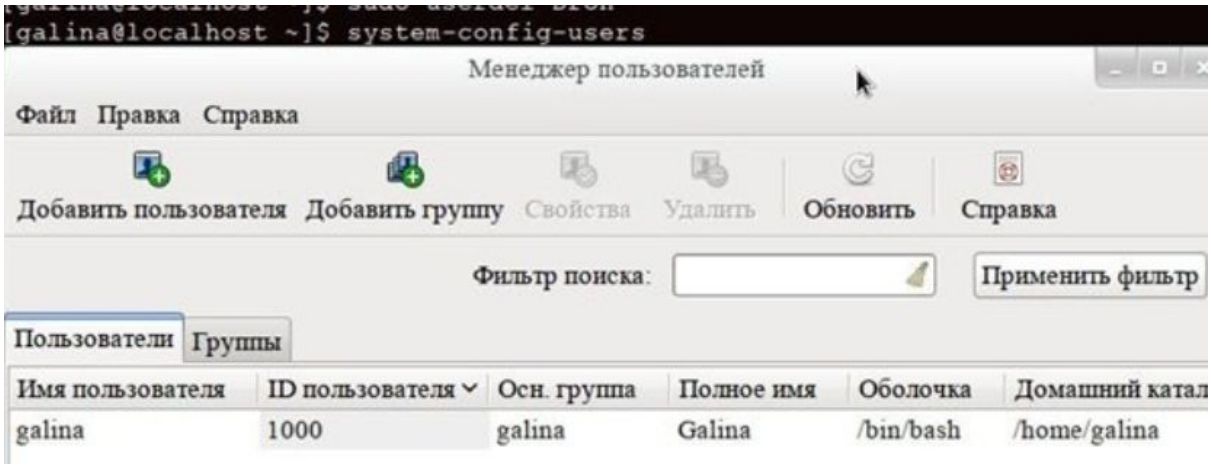
И так будет до бесконечности, пока Вы не создадите нормальный и таки взломоустойчивый пароль.

При создании нового пользователя, для него в системе создастся каталог /home <имя пользователя>, куда скопируется содержимое каталога /etc/skel (в нем находится весь "джентльменский" набор, который должен быть в любом пользовательском каталоге).

Помимо этого, в файле /etc/passwd создается запись, короче говоря, наш новый пользователь Dron будет полноправным пользователем системы.

Удалить пользователя из системы так же просто, как и завести его. Достаточно набрать команду:

userdel <имя пользователя>



Группы пользователей

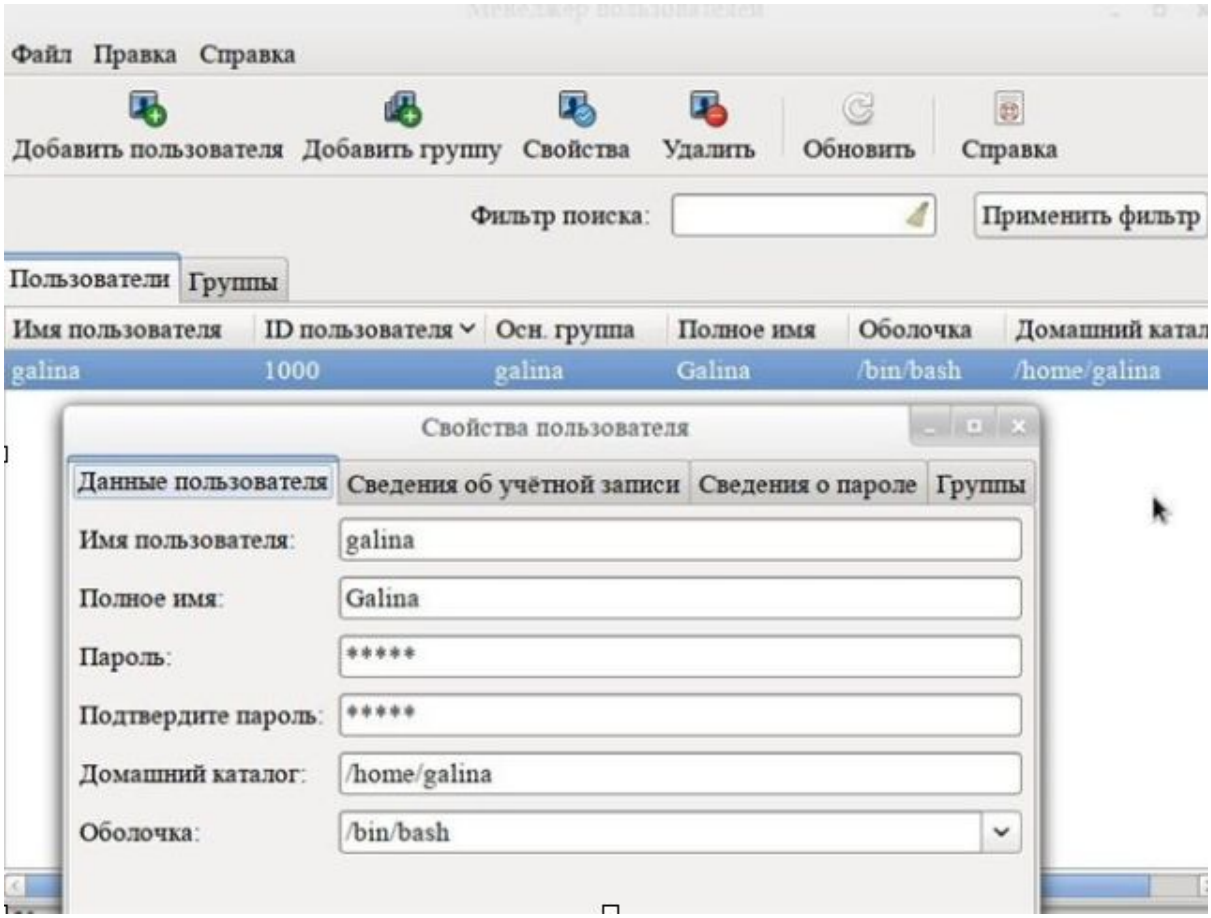
Иногда, для удобства работы, пользователей объединяют в группы. По умолчанию один пользователь не имеет доступа к домашнему каталогу другого пользователя (например, пользователь galina не

будет иметь доступ к домашнему каталогу Dron и наоборот, Dron не будет иметь доступа к домашнему каталогу galina).

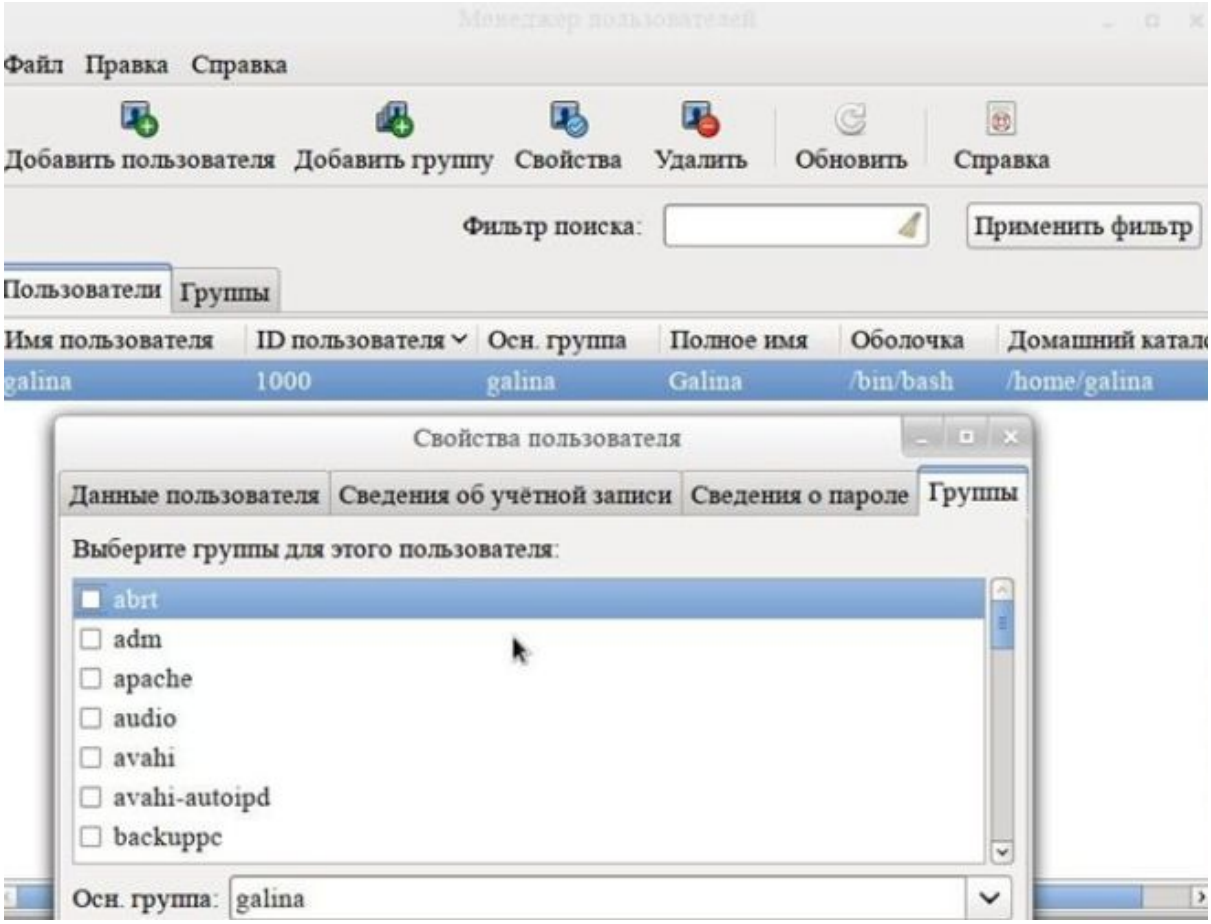
Но если нескольких пользователей объединить в одну группу, то у каждого из них будет доступ не только к своему домашнему каталогу, но и к домашним каталогам других пользователей. Полезность таких объединений очевидна.

Обычно добавлять/удалять/изменять учетные записи принято через командную строку, но хотелось бы затронуть графические конфигураторы (это маленькие утилиты, которые упрощают жизнь начинающим пользователям или тем, кто любит работать в графическом интерфейсе).

С другой стороны, я хочу показать, что работать в linux (unix)-системах можно разными способами: можно через командную строку, а можно использовать графический интерфейс - кому что больше по душе. В каждом дистрибутиве есть свои конфигураторы, и они могут отличаться. Например, в **Fedora** для редактирования пользователей и групп служит конфигуратор system-config-users.



Еще раз вернитесь к рисунку, где мы удаляли пользователя Dron из системы. В системе остался один пользователь – galina. Выделите его, нажмите на кнопку «Свойства», и Вы узнаете много интересного :). Например, обратите внимание на вкладку «Группы», где можете включить пользователя в ту или иную группу, просто поставив галочку:



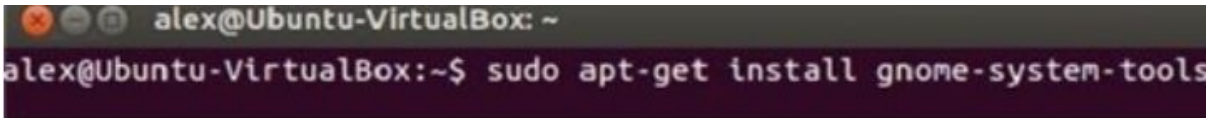
С каждым новым пользователем создаются и новые группы. Если у пользователя galina на вкладке «Группы» поставить галку возле группы «Dron», то у galina будет доступ к домашнему каталогу пользователя Dron (а вот у Dron таких полномочий не будет).

Хочу обратить Ваше внимание на один момент - с каждым новым выходом релиза конфигураторы из системы потихоньку пропадают (что, на мой взгляд, печально). Их или заменяют на другие, или вообще убирают из дистрибутива за ненадобностью. Некоторые из них могут быть установлены по умолчанию, некоторые надо устанавливать самим, это зависит от Ваших потребностей. Если при вводе команды запуска конфигуратора система скажет, что такой

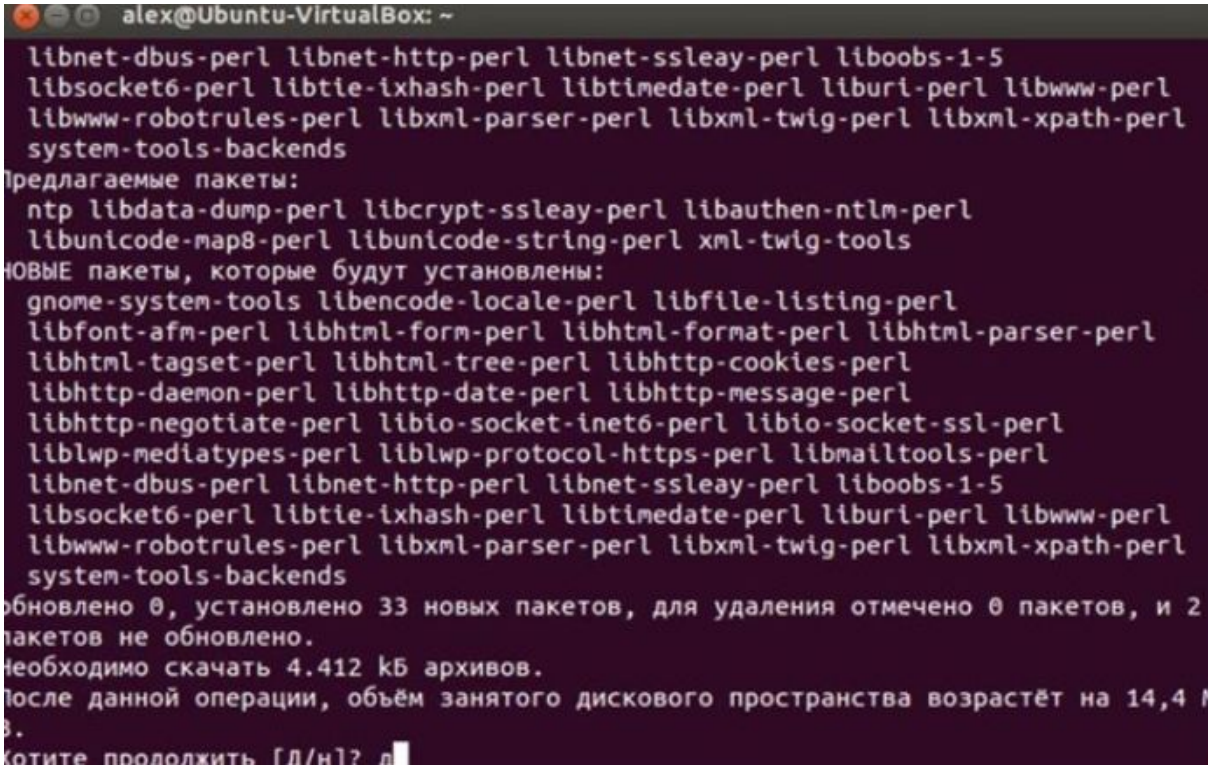
команды нет, скорей всего этот конфигуратор у Вас не установлен и можно попробовать установить его.

Давайте попробуем потренироваться на другом дистрибутиве (все предыдущие картинки относятся к дистрибутиву Fedora 16), например, на ubuntu 12.10. Отвлекусь и напомню, что в каком бы дистрибутиве Вы не сидели, смысл работы будет идентичен, отличаться будут только команды. В данном конкретном случае в ubuntu есть конфигуратор gnome-system-tools, и он по умолчанию не установлен в системе. Чтобы воспользоваться всеми прелестями этого конфигуратора, для начала установите его командой:

```
sudo apt-get install gnome-system-tools
```

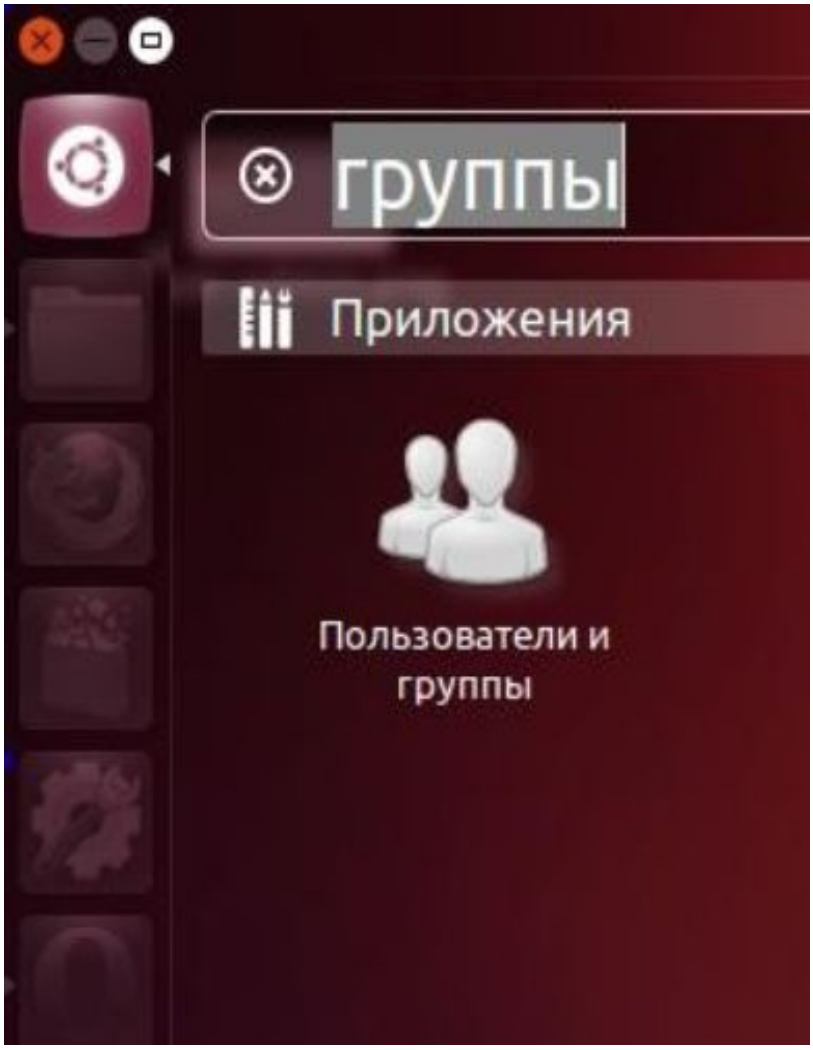


Пошел процесс установки данной утилиты:



Обратите внимание, что для установки данного пакета требуется разрешить зависимости (о зависимостях мы уже говорили в предыдущей статье, а теперь наглядно это продемонстрируем), т.е. будут установлены дополнительные пакеты для того, чтобы утилита корректно работала.

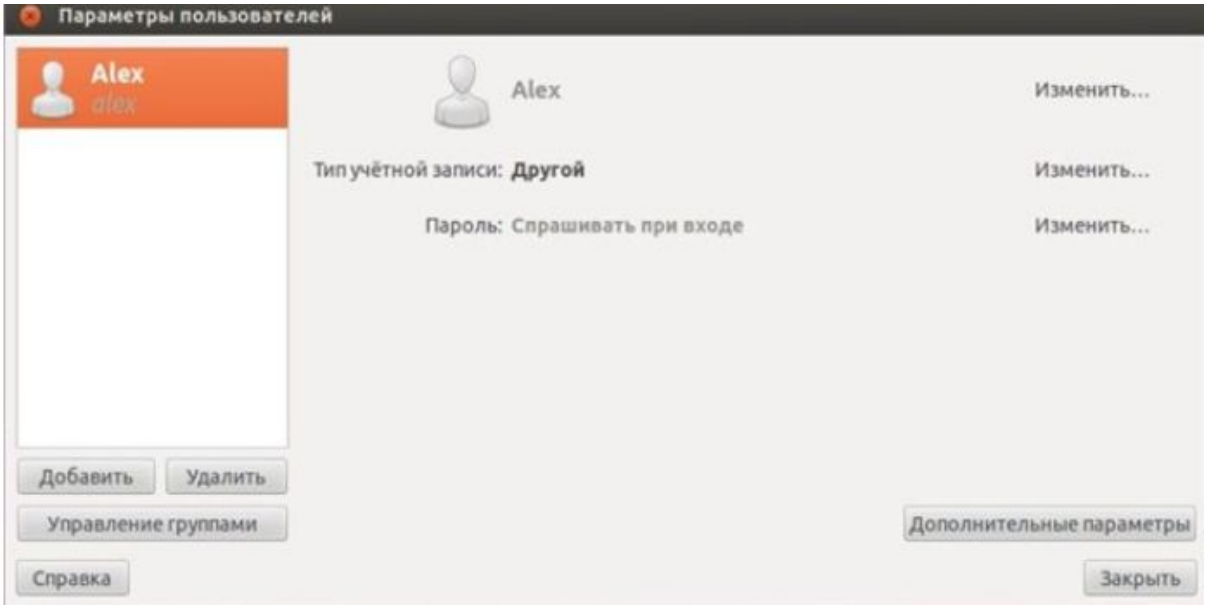
Потом в поиске можно набрать «группы» или «пользователи» и появится иконка утилиты gnome-system-tools



Или запускаем утилиту командой users-admin из терминала (командной строки):

```
alex@Ubuntu-VirtualBox:~$ users-admin
```

И нашему взору представляется графический интерфейс настроек пользователей и групп в **ubuntu**:



Все современные дистрибутивы Linux неплохо русифицированы, а поэтому более подробно останавливаться на рассмотрении этой утилиты мы не будем, но главные моменты осветим.

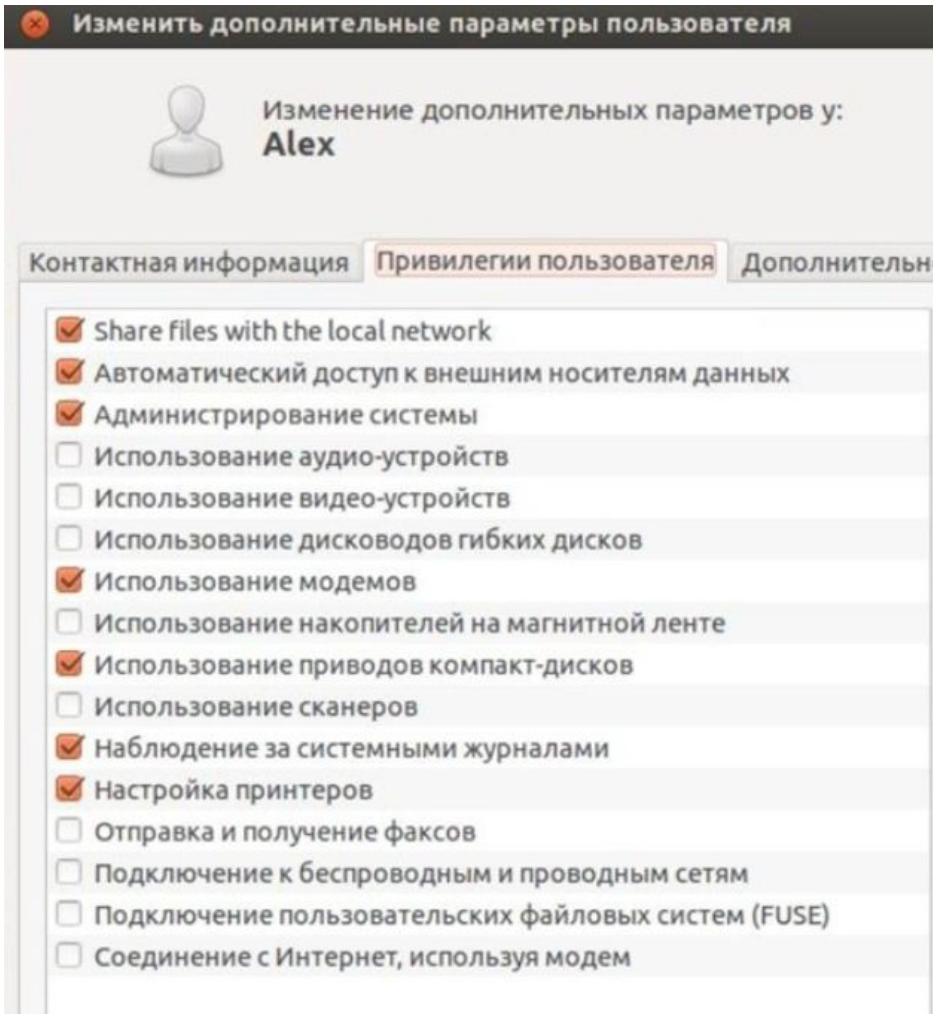
Понятно, что чтобы добавить нового пользователя, нужно нажать кнопку «Добавить», чтобы удалить созданного пользователя – нажать кнопку «Удалить». Чтобы изменить тип учетной записи – нужно щелкнуть по кнопке «Изменить», напротив поля «Тип учетной записи».

Существует 3 вида учетной записи:

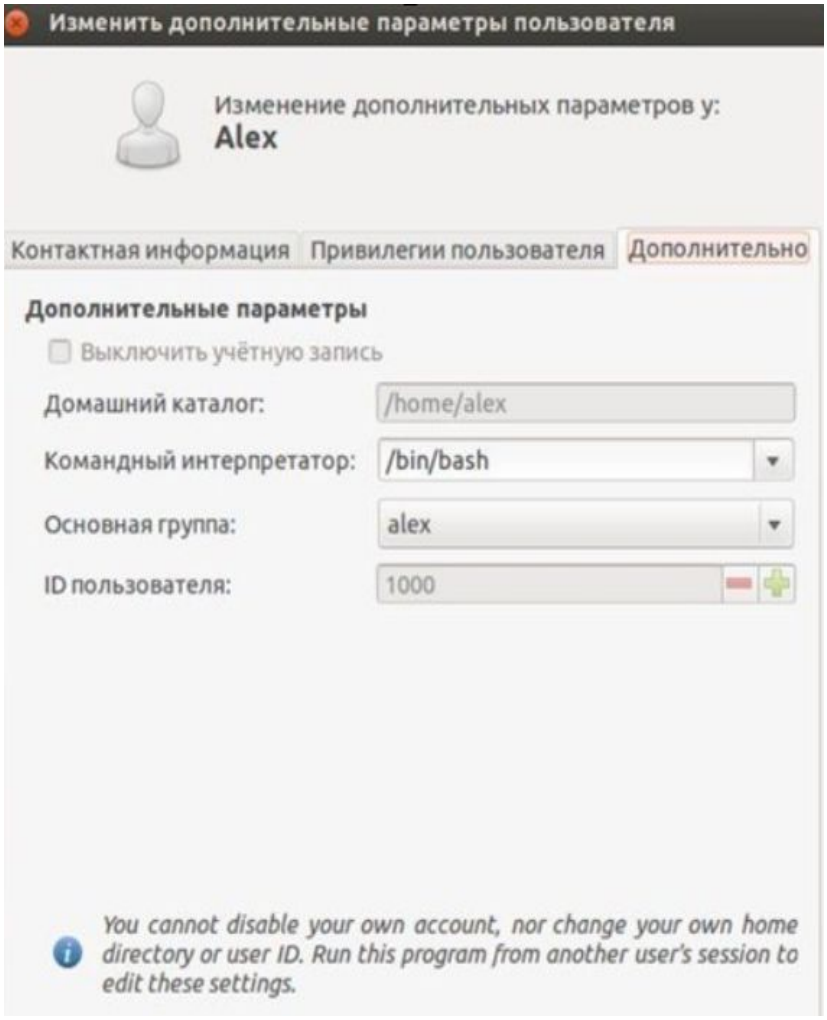
- **Администратор (root)**, – пользователь может администрировать систему;
- **Пользователь**, – пользователь может работать в системе, но не может администрировать её (использовать команду **sudo**,

устанавливать программы, управлять пользователями и тд, тп);

- **Другое**, – эта учетная запись с особыми параметрами доступа, которые устанавливаются вручную. Для этого нужно воспользоваться вкладкой «**Дополнительные параметры**». На вкладке «**Привилегии пользователя**» можно задать, какие операции может выполнять пользователь, а какие – нет.

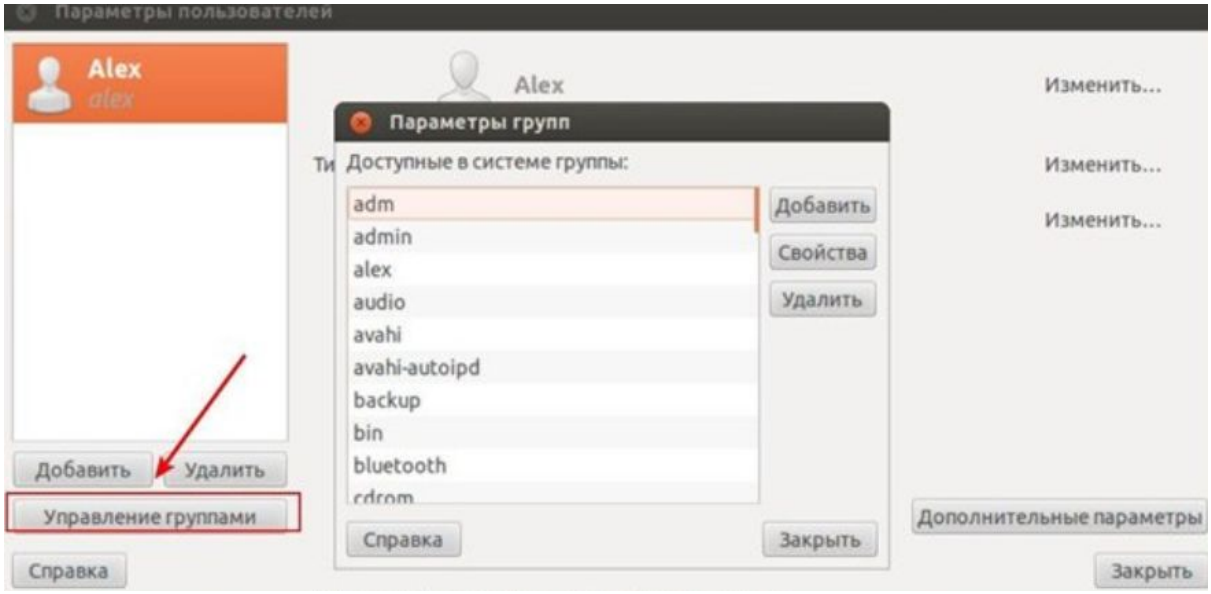


На вкладке «**Дополнительно**» Вы можете отключить учетную запись, изменить группу, **ID** пользователя, можете даже выбрать другой интерпретатор и домашний каталог.



Если Вам необходимо разрешить пользователю использовать **sudo**, то в окне «**Параметры пользователей**» выделите учетную запись пользователя, нажмите кнопку «**Дополнительные параметры**» и на вкладке «**Привилегии пользователя**» разрешите «**Администрирование системы**».

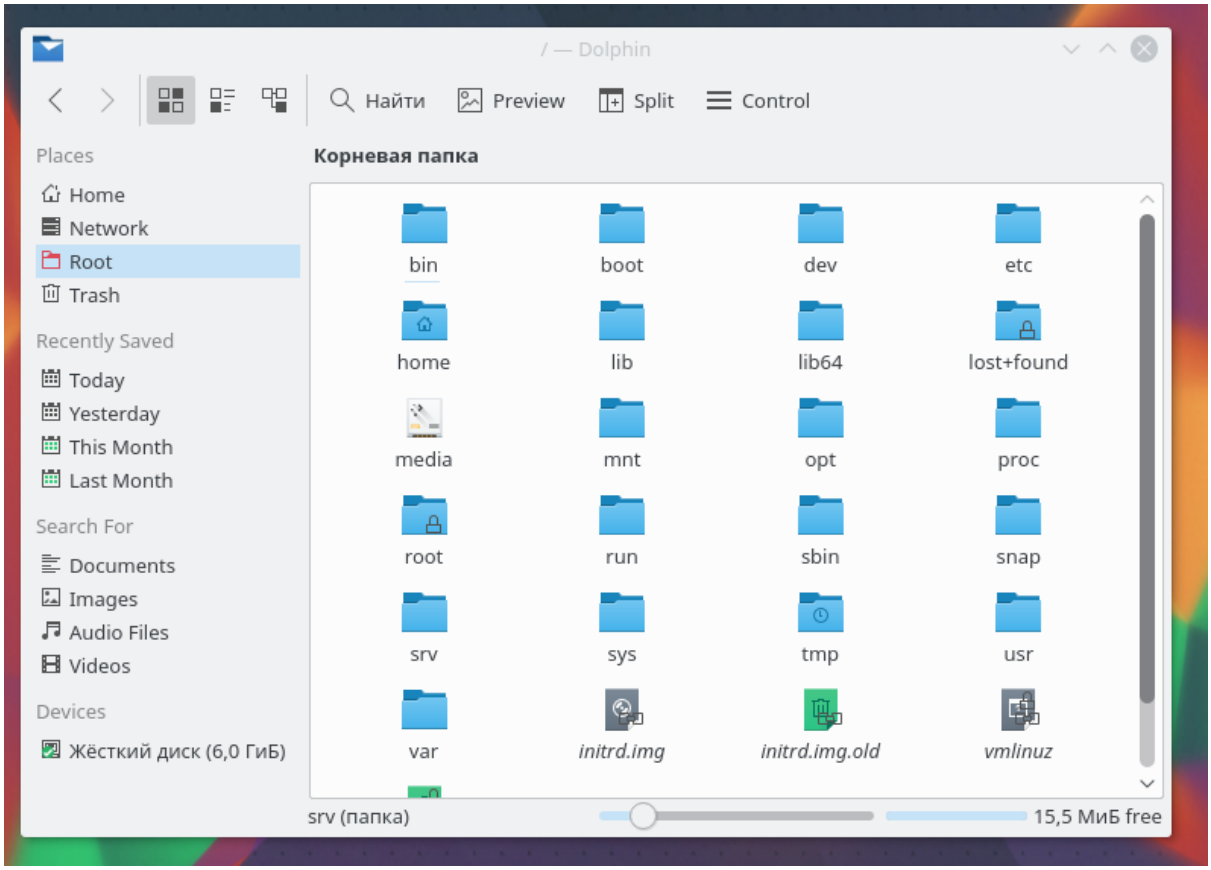
Чтобы управлять группами – нужно воспользоваться кнопкой «**Управление группами**».



Вы увидите список групп и кнопки «**Добавить**» (ну понятно же, что эта кнопка добавляет новую группу), «**Удалить**» (удаляет группу) и «**Свойства**» (с помощью этой кнопки Вы легко добавите/исключите членов группы). Все очень просто и интуитивно понятно.

Корневая файловая система Linux

И давайте кратенько про корневую файловую систему в линуксе, чтобы у Вас было понимание того, что за куча странных папок наблюдается в проводнике.

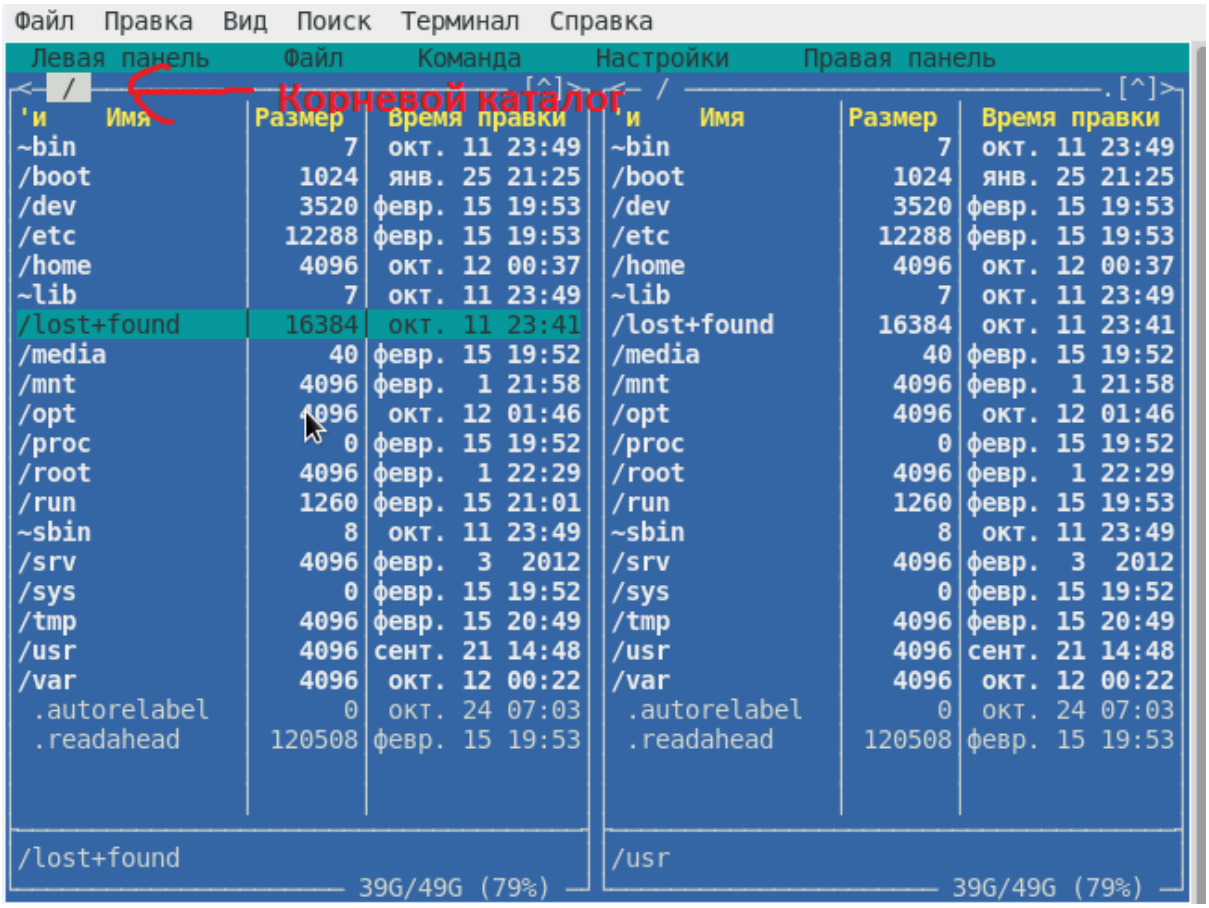


Сложно представить операционную систему (даже такую как **Linux**) без файловой системы. На ней лежит всесистемный порядок. Файловая система строго следит за организацией, хранением, а также именованием данных. Например, когда программа обращается к файлу, она ничего не знает о том, где расположена информация о данном файле, на каком физическом носителе она записана.

Единственное, что знает программа, это имя файла, к которому она обращается, его размер и другие параметры, которыми искомый файл отличается от множества других файлов (дату и время создания файла, имя владельца файла, права и метод доступа к файлу и т.д.). Все эти данные она получает от драйвера файловой системы.

Именно файловая система устанавливает, где и как будет записан файл на физическом носителе (например, на жёстком диске или

съемном устройстве). Все собранные в одном месте каталоги (по-другому папки), подкаталоги, а также файлы и есть данные файловой системы. Причем один из этих каталогов в **Linux**-системах является самым наиважнейшим, а именно «верхушкой» (по-другому - «корнем») файловой системы - в нём содержатся все остальные каталоги и файлы.



Для того, чтобы Вы понимали, о чем идет речь, я открыл каталоги файловой системы **Linux** через файловый менеджер **Midnight Commander** (MC). Верхушку файловой системы (корневой каталог) указал красной стрелкой - и вот именно к ней монтируются все другие файловые системы. Далее я попытаюсь более доступно описать все, что в данный момент Вы видите на рисунке (по тексту мы будем останавливаться на тех или иных понятиях, чтобы было понятно, о чем идет речь, и Вы, поглядев на рисунок, смогли сориентироваться).

Если жёсткий диск разбит на разделы, то на каждом из них возникают отдельные файловые системы с собственным корнем и с собственной структурой каталогов.

Любая операционная система поддерживает несколько файловых систем, например, **Windows** поддерживает такие известные всем файловые системы, как **FAT**, **FAT32**, **NTFS** и др. **Linux** не исключение. Однако важно различать файловые системы, которые могут использоваться в качестве корневой файловой системы и файловые системы, которые просто поддерживают операционную систему, но не используются для установки **Linux** или **Windows** (к таким можно отнести, например, **ISO9660** (эта модель обеспечивает совместимость носителей под разными операционными системами), **UDF** (а это особенный формат файловой системы, предназначенный для хранения файлов на оптических носителях) и др).

Мы не будем рассматривать файловые системы **Windows**, не будем останавливаться на файловых системах, не использующихся для установки операционных систем, а просто сделаем краткий обзор и познакомимся с некоторыми наиболее известными **корневыми файловыми системами Linux**.

- **ext — Extended File System** (расширенная файловая система) первая файловая система, разработанная специально для операционных систем на ядре **Linux**;
- **ext2 — Second Extended File System** (дословно: «вторая расширенная файловая система») - стандартная, но уже устаревшая файловая система **Linux**. По скорости и производительности работы она может служить эталоном в тестах производительности файловых систем. Единственный ее минус - она не является журналируемой файловой системой

(журналируемая файловая система — это система, в которой ведется запись журнала, и которая хранит список изменений, что в свою очередь, помогает сохранить целостность файловой системы при сбоях). Этот недостаток был устранён в следующей файловой системе **ext3**;

- **ext3 — Third Extended File System** (третья версия расширенной файловой системы) усовершенствованная версия файловой системы **ext2** с поддержкой ведения журнала, что, несомненно, повышает ее надежность;
- **ext4 — Fourth Extended File System** (четвёртая версия расширенной файловой системы) - новейшая файловая система Linux. Начиная с ядра **Linux 2.6.28** (вышедшем **25.12.2008 г.**) файловая система уже считается стабильной и используется по умолчанию во многих дистрибутивах **Linux**;
- Файловая система **ext4** рассматривается, как переходный шаг на пути к файловой системе следующего поколения **Btrfs**, которая уже сейчас претендует на звание основной файловой системы **Linux** в будущем.

Есть еще несколько файловых систем, например, **ReiserFS**, **JFS**, **XFS** и т.д., но использовать их в серверных условиях нецелесообразно, потому как **ReiserFS** чувствительна к сбоям и ее нужно регулярно дефрагментировать, **JFS** плохо работает с большими файлами, лучше использовать эту файловую систему под сервер баз данных, а **XFS** хорошо использовать на графических станциях для обработки видео.

Еще раз повторюсь (повторение - мать учения) :). При установке **Linux** на выбранном Вами разделе жесткого диска создается корневая файловая система, и для **Linux** именно она является самой важной. Именно к ней будут монтироваться все остальные файловые системы на других устройствах. Корневая файловая система тоже монтируется, но только не к другой файловой системе, а к «самой **Linux»**, причём точкой монтирования служит корневой каталог. Поэтому при загрузке системы прежде всего монтируется корневая файловая система, а при завершении она размонтируется в последнюю очередь.

Корневая файловая система содержит набор стандартных каталогов и утилит, без которых невозможна работа **Linux**. В **Linux** корневой каталог обозначается так - “/” (слеш). Полные имена (по-другому - пути) всех остальных каталогов получаются из “/”, к которому дописываются справа имена последовательно вложенных друг в друга каталогов. Имена каталогов в пути также разделяются символом “/”.

Например, запись /**home** обозначает каталог “**home**” в корневом каталоге (“/”), а **/home/Sonikelf** - каталог “**Sonikelf**” в каталоге “**home**”, который, в свою очередь, находится в корневом каталоге. Перечисленные таким образом каталоги, завершающиеся именем файла, и составляют полный путь к файлу.

В **Linux**, помимо полного пути, есть и относительный путь, который строится точно так же, как и полный — перечислением через слеш (“/”) всех названий каталогов, встретившихся при движении к искомому каталогу или файлу. Между полным и относительным путём есть только одно немаловажное различие: относительный путь начинается от текущего каталога, а полный путь всегда начинается от корневого каталога. **Linux** различает полный и относительный пути просто: если имя объекта начинается на “/” — это полный путь, в любом другом случае — относительный.

Основные каталоги корневой файловой системы

В разных дистрибутивах **Linux** каталоги тоже могут быть разные: в некоторых будут дополнительные файлы/каталоги конфигурации, а некоторые файлы конфигурации, вполне возможно, будут называться как-то по-другому. Я приведу названия каталогов, которые входят в дистрибутив **Fedora**. Итак, поехали:

/ - корневой каталог:

- /**bin** - содержит стандартные утилиты **Linux**. Или, скажем так, - этот каталог содержит основные исполняемые файлы (в **Windows** такие файлы имеют расширение .exe), доступные всем пользователям, а также содержит символьные ссылки на исполняемые файлы. Символьная ссылка - это специальный файл в файловой системе, внутри которого есть только одна-две строки с указанием полного пути, который должен быть открыт при попытке обратиться к данной ссылке (файлу);

- /**boot** - содержит конфигурационные файлы загрузчика **GRUB**, образы ядра (по умолчанию хранятся последние три), файлы **Initrd**. Чтобы было понятно, что это за такие мифические файлы **Initrd**, немного внесем ясность.

  Когда начинается начальная загрузка системы, сначала монтируется временный виртуальный диск, который содержит временную корневую файловую систему, с помощью которой, в свою очередь, осуществляется запуск ядра **Linux**.

  Файл **Initrd** - образ этой временной корневой системы, который хранится на загрузочном устройстве;
- /**dev** - содержит файлы устройств. В **Linux** устройством называется оборудование, которое подключается к системе в процессе загрузки ядра или в процессе работы системы.

  Эти устройства представляют методы для ввода или вывода информации. Например, жесткий диск - устройство для ввода (запись) и вывода (чтение), мышь - устройство ввода. Каждое движение или нажатие на кнопки мыши отправляет символ на устройство **/dev/mouse**, клавиатура - тоже устройство ввода.

  Большинство устройств в **Linux** представляют из себя файлы в особой файловой системе (исключение составляют сетевые

карты). И вот эти файлы хранятся в каталоге /dev, куда к ним обращается система для выполнения задач, связанных с вводом/выводом;

- /**etc** - содержит конфигурационные файлы операционной системы и всех сетевых служб. Данный каталог можно сравнить с реестром **Windows**, но в **Windows** общесистемные настройки хранятся в одном большом бинарном файле, а в **Linux** - в разных конфигурационных файлах, которые можно редактировать обычным текстовым редактором;
- /**home** - название каталога говорит само за себя. Здесь содержатся домашние каталоги всех пользователей, которые зарегистрированы в системе.

  В домашних каталогах пользователей хранятся пользовательские файлы, а также пользовательские настройки различных программ. Как я уже упоминал ранее, **Linux** является многопользовательской системой, и поэтому каждый пользователь имеет свой уникальный и неповторимый каталог для своих личных, персональных файлов.

  Этот каталог называется Домашним каталогом пользователя и обозначается, /**home/Имя_Пользователя**. Такое разделение пользовательских каталогов и файлов

операционной системы упрощает сохранение данных и повышает надежность самой операционной системы;

- /**lib** - здесь находятся различные библиотеки и модули ядра. В процессе установки различных программ в **Linux** (в том числе и драйверов) устанавливаются зависимости для корректной работы программы.

  Вот эти зависимости в большинстве случаев и есть библиотеки - набор собранных особым образом файлов, которые подключаются во время установки к устанавливаемой программе;
- /**lost+found** - этот каталог нужен для хранения испорченных файлов при проблемах с файловой системой, которые были восстановлены после, например, некорректного размонтирования файловой системы. Это очень ценный каталог, удалить его не получится (он всё равно снова появится);
- /**misc** - может содержать все что угодно;
- /**mnt** и /**media** - обычно в этих каталогах содержатся точки монтирования. В современных дистрибутивах **Linux** этот процесс обычно происходит автоматически. При этом в каталогах /**mnt** или /**media** создается подкаталог, имя которого совпадает с именем монтируемого тома;
- **/opt** - здесь обычно размещаются установленные программы, имеющие большой дисковый объем или вспомогательные

пакеты, например, у меня в этом каталоге всего две программы - это **Adobe Reader** и **Google Chrome**;

- **/proc** - это не совсем обычный каталог, это каталог псевдофайловой системы **procfs**, которая используется для предоставления информации о процессах (по-другому это виртуальная файловая система, которая обеспечивает связь с ядром и монтируется в каталог **/proc**).

  Да-да, в системе **Linux** присутствует виртуальный файловый объект, именуемый каталогом /**proc**. Он существует только во время работы системы в оперативной памяти компьютера. Каталог представляет интерес и с точки зрения безопасности. Многие из утилит, выводящие информацию о системе (например, команда **ps**), берут свои исходные данные именно из этого каталога;
- **/root** - каталог пользователя root (кто это такой, я писал выше);
- **/run** - это совершенно новый каталог, который появился совсем недавно. И создан он (по задумке разработчиков) для хранения данных, которые были запущены приложениями, требующимися в процессе работы (это могут быть и службы, запускаемые самой системой, и программы, которые Вы запускаете сами).
  Сюда входят:

    - Идентификаторы процессов **PID** (каждый запущенный в среде **Linux** процесс имеет свой уникальный идентификатор - по-другому это адрес (в числовом виде) процесса, с которым будут взаимодействовать другие запущенные процессы во время работы);
    - Информация о межпроцессорном взаимодействии (проще говоря, это обмен данными между запущенными процессами);
    - Заблокированные файлы (если в процессе работы приложение или не запускается, или работает нестабильно? это может означать, что некоторые файлы этого приложения заблокированы другими процессами и происходит это тогда, когда сразу несколько приложений пытаются использовать один общий ресурс); - ну и другие данные, необходимые во время работы.
- **/sbin** - набор утилит для системного администрирования, содержит исполняемые файлы, необходимые для загрузки системы и ее восстановления в различных щекотливых ситуациях. Запускать эти утилиты имеет право только **root**;
- **/tmp** - каталог, в котором хранятся временные файлы. **Linux**, в отличие от **Windows**, следит за чистотой и регулярно очищает этот каталог.
- **/usr** - содержит пользовательские программы, документацию, исходные коды программ и ядра. По размеру это один из

самых больших каталогов файловой системы. В этот каталог устанавливаются практически все программы. И его (этот каталог) с большой натяжкой можно сравнить с каталогом **Program Files** в **Windows**;

- **/var** - содержит файлы, которые подвергаются наиболее частому изменению. Например, кэши различных программ; файлы блокировки для недопустимости одновременного использования одной программы несколькими пользователями; файлы системных журналов; временные файлы (при выключении компьютера содержимое очищается); информация о различных программах; общая информация о состоянии системы с момента последней загрузки, входа в систему и т.д.; очередь печати, факсов, а также входящие почтовые ящики пользователей и т.д.

Кратенькие выводы

Файловая система в **Linux** это неразделимое целое, начинающееся с корневого каталога (так называемой "верхушки"), а внутри каталога могут находиться целые диски или их разделы (с другими файловыми системами), которые можно легко примонтировать (присоединить) и также легко отмонтировать (отсоединить) от единственного, единого, неразделимого, целого дерева. Специально выделил эти слова, чтобы было понятно отличие от файловой системы **Windows**.

Если кто не в курсе, в **Windows** каждый логический диск — это корневой каталог. Например, если есть три логических диска - C, D, F, значит будет три корневых каталога.

О программах, репозиториях, пакетах и зависимостях в Linux

Все программы в дистрибутивах **Linux** - это отдельные проекты, которые развиваются сами по себе. Вы должны представить себе некую цепочку: есть отдельные пакеты (программное обеспечение), есть зависимости. Цель же всего этого – собрать все эти программы, с их зависящими друг от друга библиотеками вместе, да не просто собрать, а сделать так, чтобы все это работало в комплексе.

У каждого дистрибутива есть свои разработчики (майнтейнеры). Эти люди занимаются тестированием различных пакетов на их нормальное функционирование, взаимную совместимость, а также часто добавляют собственные усовершенствования или не успевшие войти в официальную сборку и, в конечном итоге, отвечающие за включение пакета в дистрибутив патчи.

Т.е. разработчики берут программы из открытых исходных кодов и начинают подгонять их друг к другу, упаковывая в пакеты и соблюдая все зависимости, тестируя и удаляя ошибки из этих самых программ. Представили? Тогда, думаю, Вы понимаете, что это непростое занятие. Так вот, все эти подогнанные друг к другу

программы, библиотеки и нескучные обои, упакованные в пакеты со всеми зависимостями, – это и есть репозиторий Вашего дистрибутива, откуда программы и устанавливаются на Ваш компьютер.

О репозиториях в Linux. Что это и зачем нужно

Т.е. еще раз и чуть иначе: репозиторий в Линуксе – это все файлы пакетов, принадлежащие одному дистрибутиву (например, **Fedora**), одной его версии (например, 16), то бишь сие есть огромное хранилище пакетов, которое находится в сети Интернет и которым Вы можете спокойно воспользоваться (причем бесплатно). Те самые ISO-файлы образов для записывания на болванку и последующей установки содержат как раз репозитории пакетов со всеми зависимостями и менеджером пакетов плюс установочную программу, которая разметит жёсткий диск, всё поставит и приготовит Вам собственно сервер.

Для чего создаются репозитории? Ответ прост – для централизованного управления обновлением пакетов. Представим на секунду, что у нас нет репозиториев, и Вы установили **Linux** с диска с определенными (стандартными) программами. Однако время не стоит на месте, все программы обновляются и всё такое прочее. Как же тогда узнать – есть ли обновление для Вашей программы или нет? Естественно, придется посещать сайт разработчиков программы, чтобы выяснить это, что, согласитесь, не

совсем удобно, особенно, если программ у Вас установлено очень много. Ну и понеслось, Вы раз проверили, два проверили наличие обновлений, в третий раз забыли, а потом и вообще надоело каждый раз смотреть, вышло там обновление или нет. И тут раз..

Вспоминаем, для чего у нас существуют обновления? А для того, чтобы не просто иметь новый (и улучшенный старый) функционал в оных программах, но еще и залатывать дыры, которые нередко приводят к различным неприятностям, начиная от глюков программы/системы и заканчивая проблемами с безопасностью (я, например, очень не люблю "терять" пароль, скажем, от почты по вине дыр в софте). Поэтому-то разработчики **Linux** и создали репозитории, с помощью которых можно быстро и удобно отслеживать обновления тех или иных пакетов (да и вообще обновления всей системы в целом), устанавливать новые и обновленные и всё такое прочее. Кстати, почему для **Windows** оным еще не озадачились, решительно непонятно (хотя там частично спасают программы для обновления программ, пусть это и не совсем то).

О пакетах и менеджерах пакетов в Linux. Что это и зачем нужно

К слову, в заголовке выше я специально выделил четыре пункта, чтобы, так сказать, разбить содержание на несколько частей. О репозитории пакетов мы поговорили и теперь переходим к

следующему куску повествования, а именно, поговорим о том, что подразумевается под понятием пакеты в **Linux**.

Под пакетами в **Linux** подразумевается программное обеспечение (ПО), которое Вы хотите установить на компьютер. Скажем, например, в **Windows** софт устанавливается с помощью мастера (программы) установки – **setup.exe** или **install.exe**. Вы запускаете этот мифический экзешный файл, и процесс установки начинается едва ли не мгновенно после выбора пути и мелких побочных настроек.

Установка же программ в **Linux** несколько отличается тем, что здесь используются два основных способа инсталляции: с помощью пакетов или из исходных кодов. Пакет содержит собранную программу, информацию о том, какие требуется совершить действия для ее установки, информацию о зависимостях, а также, возможно, много других данных (в зависимости от вида пакета). Причем за установку (удаление, обновление) отвечает такая штука, как менеджер пакетов.

Обычно менеджер пакетов является сердцем дистрибутива, обеспечивая полный контроль целостности и работоспособности всей системы, и он же обеспечивает пользователю интерфейс для автоматизированного получения пакета, его зависимостей и его установку. Пакеты, как уже говорилось, собираются в репозитории, т.е. всё это можно сложить в одну цепочку: пользователь

запрашивает установку пакета – менеджер пакетов отслеживает зависимости – он же получает необходимые пакеты из репозитория(ев) – и он же устанавливает зависимости и требуемый пакет.

Практически каждый дистрибутив **Linux** имеет свои репозитории, зачастую несовместимые с другими дистрибутивами. Менеджер же пакетов – консольная утилита, однако обычно для нее существуют многочисленные графические оболочки, которые легко отыскать в каждом дистрибутиве, введя в поиск «Установка/удаление программ».

Пакетные менеджеры бывают разные. Для управления пакетами в разных дистрибутивах используются разные программы. В общем-то, их не так уж и мало, а посему выделим «основные», которые «умеют» разрешать зависимости. Фраза «умеют разрешать зависимости» означает следующее – если при установке пакета будет обнаружено, что для корректной его установки нужны дополнительные пакеты, то менеджер пакетов установит их сам, т.е. Вам не придется искать дополнительные пакеты в репозиториях. Те менеджеры пакетов, которые не обладают такой функцией (умением разрешать зависимости), мы рассматривать не будем, ибо оные только сообщат Вам, что пакет установить невозможно и выведут весь список файлов (именно файлов, а не пакетов), которые нужны для установки данного пакета. А уж какой файл в каком пакете находится, Вы будете догадываться и искать самостоятельно.

Вот небольшой список:

- **Yum (Yellow Dog Update Modified)** – мощный менеджер пакетов, основанный на rpm (простой МП, не умеет разрешать зависимости), работающий в текстовом режиме и умеющий разрешать зависимости, а также умеющий поддерживать репозитории (источники пакетов). Используется в **RedHat Linux**, а также в **Fedora**, **SuSe** и некоторых других;
- **APT [Advanced Package Tool]** создана для дистрибутивов **Linux**, основанных на **Debian**, используется в **Ubuntu** (и клонах), **ALT Linux** и др. Мощный менеджер пакетов, работающий в текстовом режиме. Умеет разрешать зависимости и поддерживает репозитории (источники пакетов);
- **Portage package management system** имеет много разновидностей, примером может служить дистрибутив **Gentoo**. Как вариант пакетного менеджера можно привести **emerge**.

К слову, пакетные менеджеры не просто ищут желаемые Вами программы по описаниям, но прежде нам нужно ввести еще один не раз уже упомянутый термин и объяснить его.

О зависимостях в Linux. Что это и зачем нужно

Например, Вы захотели установить программу и нажали кнопку «Установить», а она спрашивает Вас про какие-то мифические и

непонятные зависимости. Так давайте разберемся – а что же это такое?

Часто компоненты, используемые различными программами, выделяют в отдельные пакеты и помечают, что для работы ПО, предоставленного пакетом А, необходимо установить пакет Б. В таком случае говорят, что пакет А зависит от пакета Б или что между пакетами А и Б существует зависимость (обычно в роли зависимостей выступают какие-либо библиотеки, без которых программа не будет запускаться, поскольку использует функции этой библиотеки). Вот как раз отслеживанием зависимостей между такими пакетами и занимается уже неоднократно упомянутый менеджер пакетов.

Говоря просто, пакетный менеджер это такая программа, которая ведёт базу данных установленных приложений и их версий, и всегда знает, какие файлы куда установлены, чтобы можно было поставить новые программы, удалить старые или обновить всю систему целиком без переустановки и вычищения мусора оставшихся файлов.

Вся эта огромная куча пакетов с их ворохом зависимостей друг от друга, управляемая пакетным менеджером, как раз и составляет Ваш дистрибутив **Linux**. Но это не просто куча мусора, а упорядоченная система, которая называется - та-дам! - репозитории

пакетов программ. Круг замкнулся – мы вернулись к первому понятию – что такое репозиторий

Несколько слов о нюансах

Напоследок все-таки хочется сказать, что какой бы **Linux** не была устойчивой, стабильной и неубиваемой, всё же пользователь должен придерживаться определенной осторожности. Например:

1. Не надо искушать судьбу и ставить программы в **Linux** в обход менеджера пакетов, простой компиляцией. Работать они будут, но пакетный менеджер ничего о них не будет знать, из-за чего при обновлении системы или программ Вы рискуете получить больше проблем на свою голову, чем представляете. Устанавливайте программы только в виде пакетов;
2. Не надо подключать те репозитории, о которых имеете совсем смутное представление. Например, не надо подключать репозитории со словами **testing, debug** и тому подобными терминами, ибо эти репозитории в первую очередь предназначены для самих разработчиков дистрибутивов и далеко не всегда стабильны;
3. Не подключайте подряд все доступные репозитории, это тоже может сыграть с Вами злую шутку. Подключайте только самые необходимые, не надо жадничать

Например, при установке операционной системы **Fedora** по умолчанию сразу подключены два репозитория:

- **Fedora** (пакеты, которые подходят на любую комбинацию из компакт-дисков или DVD-дисков);
- **Updates** (обновленные пакеты, новее, чем репозиторий (хранилище) **Fedora**).

Для нормальной работы нужно подключить дополнительный репозиторий **rpmfusion** (без него Вам действительно не обойтись), что даст доступ к программам, которые не могли быть включены в дистрибутив из-за лицензионных ограничений (приложения, которые требуются для воспроизведения мультимедиафайлов, таких как mp3, dvd и т.д.; драйвера – к ним относятся проприетарные драйвера для **ATI** и **NVIDIA**; игры: **Bub's Brothers**, **Secret Maryo Chronicles**, **UFO: Alien Invasion**, **Wörms of Prey**, **xrick**, **GLtron** и многие, многие другие; эмуляторы: эмулятор **Commodore 64**, а также **Commodore 8 bit**, эмулятор **Amiga**, **Nestopia**, **ZSNES** и много других). Чтобы подключить этот репозиторий, достаточно в командной строке (терминале) от суперпользователя (**root**) ввести команды:

```
$ sudo rpm -ivh http://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-stable.noarch.rpm
```

```
$ sudo rpm -ivh http://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-stable.noarch.rpm
```

Обратите внимание, что репозиторий **rpmfusion** разделяется на две части: **free** и **nonfree**. Первый содержит чисто свободные программы в понимании **FSF**, распространяемые под **GPL** и совместимыми с ней лицензиями. Содержимое второго, вопреки названию, - также программы по преимуществу свободные, но попадающие под пресловутые патентные ограничения некоторых государств (например, аудио- и видеокодеки).

То же самое касается и менеджера пакетов, например, в **Fedora**. Для нормальной и удобной работы менеджера пакетов (**yum**) рекомендуется подключить дополнительный плагин **fastestmirror**. Этот плагин очень важен: он определяет не просто ближайшее зеркало, как это делают аналогичные утилиты из других систем управления пакетами, а устанавливает именно самое быстрое зеркало в данный момент – по времени отклика.

```
$ sudo yum install yum-plugin-fastestmirror
```

В двух словах как-то так. Пожалуй, перейдем к послесловию этой главы, иначе про линуксы я могу писать еще долго.

Послесловие к пятой главе

Чуть больше основ по **Linux**'ам есть на **sonikelf.ru** и других ресурсах. Лучше всего не полениться и почитать документацию и материалы сообществ выбранного Вами дистрибутива. В нашем

случае, тот же **Ubuntu Server** приведен как пример, еще и как раз за счет размеров этого самого сообщества и количества разного рода инструкций и форумов по нему в сети, так что разобраться сможет каждый.

Пока это самый минимум, который я сумел втиснуть в первую часть цикла книг по серверам. Я понимаю, что этого мало, но с другой стороны, достаточно, чтобы установить и начать делать первые шаги, а это, как Вы помните, первоцель этой книги, - показать, что не всё так страшно и дать опору для старта.

Двигаемся дальше.

Глава 6. Сети. Базис и не только

Предисловие к шестой главе

Эта глава, скорее всего, будет одной из самых тяжелых для понимания многих, т.к излагать я буду базовые вещи техническим языком, причем говорить не столько про программную часть, сколько про физическую, в каковой, как раз таки, многие плавают.

Мы поговорим про монтаж, категории, стандарты, нюансы и другие хвосты, затронем беспроводные сети и черт знает, что еще. Читать лучше вдумчиво, ибо понятно, с наскоку, может быть не сразу

Поехали.

СКС как часть информационных систем

Понятие СКС, типология и категории

СКС - структурированные кабельные сети. Собственно, жизнь практически любой компании в современном мире начинается с построения этих сетей, которые, собственно, обеспечивают обмен данными между устройствами, выход в интернет и всё такое прочее.

От качества этих, с позволения сказать, каналов связи, не в последнюю очередь зависит стабильность и надежность работы

всего на свете, да и вообще существование рабочего процесса как такового, а значит и поступление финансов в компанию.

Согласно действующим правилам, структурированные кабельные сети (СКС) должны проектироваться и монтироваться специализированными организациями, имеющими государственную лицензию на данный вид деятельности.

Однако, на практике, во всяком случае в этой стране, оное почти никем не используется (и не только в силу высокой стоимости этих самых работ) и монтаж выполняется собственными силами, благо это не так сложно (хотя и времязатратно, если офис небольшой) как могло бы казаться.

Обычно провода прокладываются вдоль плинтусов или по стенам (в основном по эстетическим соображениям), поверх устанавливаются специализированные короба (при наличии подвесных потолков и проходов между комнатами, можно применять и их).

При этом далеко не всегда выполняются требования, предъявляемые стандартами, но что поделать (особенно когда компания небольшая и часто меняет офисы, например, у автора, за год компания переезжала порядка 7 раз и каждый раз приходилось снимать-растягивать заново сеть на 50 с хвостиком машин).. работать-то надо, еще и в сроки, поставленные руководством укладываться.

Самая худшая сеть, - это та, которая валяется на полу и об неё постоянно спотыкаются. Печальная практика, но таковых достаточно много, ибо русское “работает и ладно” у нас существует везде и всегда, что очень удручает профессионалов, когда они это видят.

Поверьте, ничто не радует сердце хорошего айтишника (в работе и вообще), как идеально проложенные и промаркированные кабели… Но, как я говорил, встречается это… редковато.

Категории сетей

СКС, они же кабельные системы (слово-то какое), характеризуются категорией, которая определяет качество. Из основных можно выделить следующие.

Категория	Максимальная частота сигнала (**МГц**)	Область применения
3	**16**	Телефонные каналы, локальные сети **Ethernet 10 Base-T**
5	**100**	Локальные сети со скоростью передачи до **100 Мбит/с**
5e	**100**	Локальные сети со скоростью передачи до **1000 Мбит/с**
6	**250**	Локальные сети со скоростью передачи до **1000 Мбит/с**
7	**600**	Локальные сети со скоростью передачи до **1000 Мбит/с**, сети **ATM**, кабельное телевидение

Кабельную сеть можно отнести к определенной категории, если при её создании использованы элементы (розетки, разъемы, кабели и тп), удовлетворяющие требованиям данной или более высокой категории, а проектирование и монтаж выполнены в соответствии с требованиями стандартов (ограничения на длины, число точек коммутации и тд). Хотя, опять же, на практике всё это выполняется не строго :)

В настоящее время, точнее на момент написания этого текста в **2015**-ом, всё еще большинство эксплуатируемых кабельных систем относятся к категории **5**, которая допускает передачу данных по сети со скоростью до **100 Мбит/с**. Категория **5e** вводит небольшие дополнительные ограничения, которые позволяют использовать каналы передачи данных с гигабитными сетевыми картами. На практике, кстати, аккуратно выполненная проводка на элементах категории **5** позволяет осуществлять передачу на скорости до **1 Гбит/с**.

Волоконно-оптические сети

При построении кабельной системы крупных фирм часто возникает необходимость подключения устройств, разнесенных на большие расстояния (**100** и более метров). Как правило, это пробросы каналов между зданиями (или в рамках одного крупного) и другие хвосты.

В этом случае, при наличии средств и возможностей, частенько успешно применяются волоконно-оптические линии связи.

Если говорить просто, то обычно одно волокно кабеля используется для передачи сигнала, другое — для приема. В частных случаях, существует оборудование, позволяющее за счет использования различных диапазонов излучения, передавать и принимать данные по одному волокну, но оно применяется не часто. Обычно

оптические кабели проектируются с большим запасом по числу волокон.

При расстояниях 100—**200 м** применяются многомодовые оптические кабели, причем стоимость прокладки и эксплуатации такой линии практически соизмерима со стоимостью линии на витой паре. При длинах соединений свыше **200 м** используется одномодовый оптический кабель. Соответствующее оборудование (приемники и передатчики оптического сигнала) по стоимости в несколько раз дороже, чем модели для многомодового кабеля.

> Существуют технологии оптического уплотнения (**CWDM — Coarse Wavelength-division multiplexing**, грубое мультиплексирование с разделением по длине волны, и **DWDM — Dense Wavelength Division Multiplexing**, плотное спектральное мультиплексирование), позволяющие по одному волокну передавать несколько каналов данных. Но они оправданы только в случае сверхдлинных линий связи и не применяются на уровне предприятий.

Если же расстояние, на которое предполагается передавать данные, измеряется десятками километров, то необходимо использовать передатчики повышенной мощности (они присутствуют в линейках продукции практически всех изготовителей), что логично само по себе.

Для подключения волоконно-оптических линий применяются специальные модули. Существуют различные модели такого

оборудования, на практике в последнее время наиболее распространены **SFP**-модули, подключаемые в соответствующие порты активного оборудования. Но поскольку на практике эксплуатируются различные технологии, при проектировании расширения сети следует обращать внимание на совместимость используемых решений. Например, следует учитывать типы оптических разъемов, которые могут отличаться на оборудовании различных вендоров, принимать во внимание тип оптического кабеля (выпускаются более чем по **5** стандартам), длину волны и тп.

> **SFP (small form-factor pluggable)** — компактный сменный приемопередатчик, регламентирован международным стандартом. В конструктиве SFP выпускаются модули для подключения как витой пары, так и оптических каналов (одномодовых и многомодовых) на различные частоты линии связи.

Эксплуатация волоконно-оптических каналов связи требует повышенной аккуратности со стороны пользователей. Не допускается резко изгибать кабель (и даже не резко), следует оберегать оптические поверхности от засорения (не трогать их руками, всегда закрывать концы кабеля и гнезда в модуле подключения специальными заглушками и тд). В случае повреждения кабеля обычно требуется приглашение монтажных фирм, поскольку приобретение оборудования для поиска и определения мест неисправности (рефлектометры и тд и тп) доступно только специализированным организациям.

В общем говоря, при работе с оптикой нужны хорошие знания её монтажа и понимание как оно работает.

Так как оптика не совсем тема этой книги, и если говорить проще, то она, а) используется, б) имеет высокую скорость передачи в) позволяет передавать данные далеко. Сама же по себе тема интересная, рекомендую почитать про оптику где-нибудь отдельно, принцип работы достаточно интересен.

Сети 10G

Логично, что увеличение объемов информации заставляет постоянно повышать скорости передачи данных. Внедрение мультимедийных приложений (**IP**-телефония, видеоконференции, системы видеонаблюдения и тп) приводит к тому, что величина скорости передачи в **1 Гбит/с** становится недостаточной для магистральных каналов предприятия. Объединение каналов не позволяет увеличить это значение на порядок.

Результатом усилий ряда компаний стала технология передачи со скоростью **10 Гбит/с**, которая, кстати, малоизвестна в рядах просто пользователей, не знакомых со структурами сетей.

В настоящее время стандартизовано использование в данных целях как медного кабеля (витой пары), предназначенного для небольших расстояний, так и оптического. На практике соединения **10G** применяются для связи коммутаторов уровня ядра или при

оснащении дата-центров, где имеют место самые высокие требования к объемам и скорости передачи данных.

Обжимка кабеля пятой категории

На тему обжимки кабеля и всего, что с этим связано, я писал некогда в соответствующей статье “**Как и чем обжимать сетевой кабель [RJ-45]**”. Продублирую в книге, добавив немного новой информации, которая может быть многим полезна, ибо рассматривать сети без этого, собственно, не резонно.

Что нам понадобится из сетевого оборудования:

1. Не обжатый кабель (да и обжатый тоже, ведь он одним легким движением ножниц превращается в ~~элегантные шорты~~ не обжатый вид). Купить можно в любом компьютерном магазине по цене около **5-10** рублей за метр или бухтой в **100** метров сразу;
2. Пара (как показывает практика - лучше больше) коннекторов **RJ-45**. Стоят в районе **5** рублей за штуку. Выглядят так:

Что нам понадобится из инструментов:

1. Ножик небольшой (полноценный кухонный тесак для разделки мяса лучше не использовать ;-)). Он нам понадобится, чтобы снять внешнюю изоляцию.
2. "Обжимка", т.е. то, чем мы будем обжимать. Здесь есть два варианта:
    - Кримпер. Стоит в тех же магазинах около **500** р. Позволяет легко и быстро обжимать кабели. Выглядит так:

3.

- Отвертка обыкновенная (шлицевая). В этом случае часто требуется дополнительный фиксатор для коннекторов. В качестве него можно использовать руки напарника.

Примечание: ~~в принципе некоторые обжимают еще зубами~~ в случае использования отвертки желательно быть осторожнее. Были прецеденты :)

Приступаем к процессу обжатия кабеля RJ-45

Сам процесс элементарен и прост. Сначала снимаем внешнюю изоляцию и всё, что с ней связано. Есть два подхода, один чисто русский, один продуманный.

Вариант 1. Чисто русский

Ножиком, слегка нажимая, аккуратно проходимся по контуру нашего кабеля (если давить слишком сильно, то Вы повредите жилу или случайно отрежете пол кабеля).

Вариант 2. Продуманный

В кабеле, как правило, есть специальная нить (обычно белого цвета из капрона), беремся за неё и тянем вдоль кабеля. Таким образом она будет разрезать внешнюю изоляцию, а лишнее потом можно отрезать всё тем же ножичком.

Вариант 3. Лучший

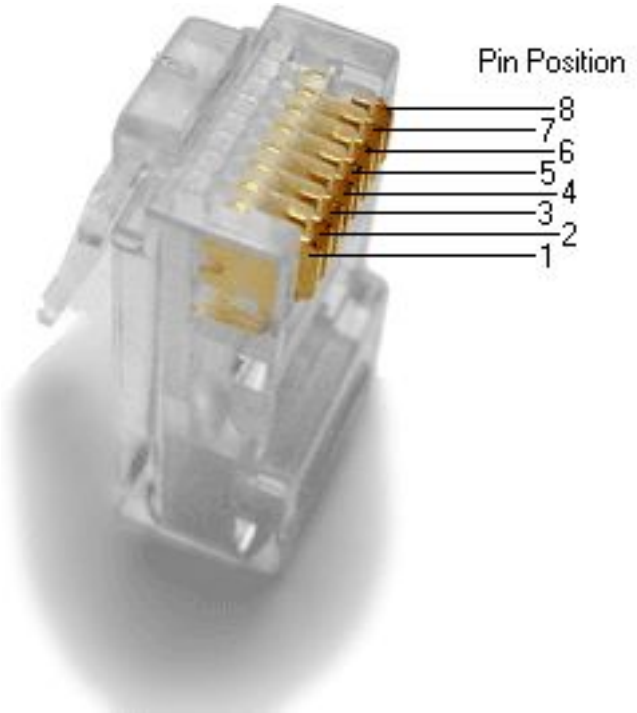
В ~~особо продвинутых~~ большинстве кримперов давно уже есть встроенное лезвие для снятия изоляции, - достаточно вставить кабель в отверстие, сжать рукоятки инструмента и снять потом руками обмотку. Удобно, ровно и изоляция срезается, можно сказать, тютелька в тютельку.

Теперь наш кабель без изоляции представляет собой **4** пары скрученных жил и капроновую нить. Нить Вы сразу можете обрезать. Она Вам не понадобится. Далее распутываем жилы и пробуем их по возможности расправить.

Остается определить какой тип кабеля мы будем делать. Их существует два:

- **Прямой**, - для соединения порта сетевой карты со свитчем/хабом;
- **Перекрестный (Crossover)**, - для соединения напрямую двух сетевых плат, установленных в компьютеры, а также для соединения некоторых старых моделей хабов/свитчей.

В большинстве случаев достаточно использовать прямую последовательность обжима.



Если Вы не уверены в том, какой именно тип использовать Вам, то узнайте в тех.поддержке Вашего провайдера в каком порядке должны быть жилы:

- Если Вы хотите соединить **2** компьютера напрямую, то используйте перекрестный тип;
- Если Вы хотите подключить компьютер к хабу/свитчу/роутеру, то достаточно использовать прямой тип.

Схема обжимки прямого кабеля

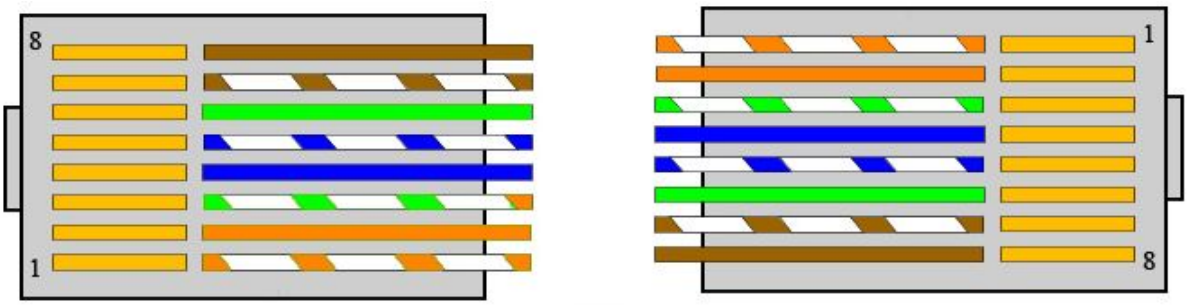
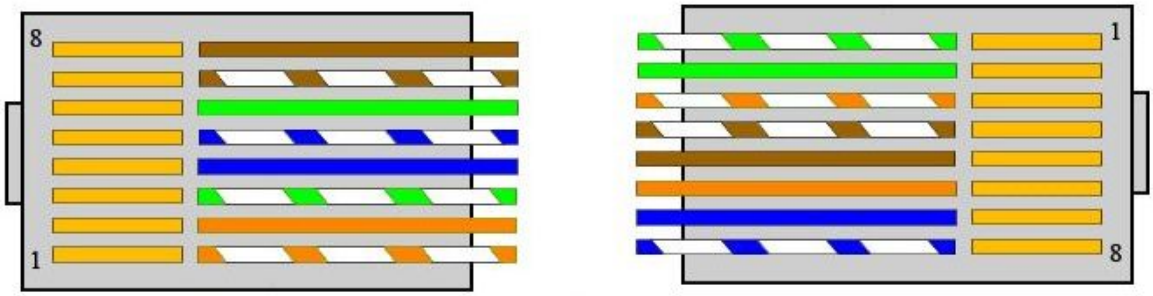


Схема обжимки перекрестного кабеля



После того, как Вы определились со схемой, надо выстроить наши жилы (проводки) в одном из вышеперечисленных положений. После этого нужно выровнять жилы так, чтобы они были максимально параллельно и близко друг к другу, а именно чтобы нигде не расходились ни сами жилы, ни их кончики.

После этого аккуратно, плавно, нежно и медленно вводим провода в коннектор, тщательно контролируя процесс, а именно, следя за тем, чтобы цвета не перепутались или две жилы не залезли в одну дорожку. Удачно введя в оба коннектора, разложенные и разровненные заранее жилы, мы засовываем их (коннекторы) в кримпер и крепко зажимаем рукоятки.

Если всё сделали правильно и осторожно, то провод готов.

В случае с отверткой всё несколько сложнее. Нам надо закрепить коннектор где-нибудь и закрепить провода, чтобы при обжимке жилы не вылезли из коннектора. Берем отвертку и продавливаем желтенькие защелки до конца (они должны продавить изоляцию жилы, чтобы иметь соединение с проводом). Трудность в том, что у Вас может выпасть провод, когда Вы уже зажмете защелку, или если жила не дошла до защелки, а Вы ее уже продавили. Тогда придется использовать другой коннектор, ибо этот сломан.

Примечание-повторение-дополнение

Для подключения компьютера к коммутатору обычно используется расшивка разъема, показанная в вышеупомянутой статье или, теперь уже, еще и выше по тексту книги.

На практике есть два применимых варианта расшивки кабеля — A и B. Кстати говоря, один используется в основном в США, другой — A — в странах Европы.

К слову говоря, если Вы администрируете не первый, то в условиях одной организации можно встретить оба варианта разделки разъема, например, вследствие выполнения работ предыдущими администраторами или монтажными компаниями.

Вообще говоря, разделка разъема не сказывается на работе оборудования за исключением случая, когда один конец кабеля разделан по варианту A, а другой — по варианту B. На практике такая ситуация чаще всего возникает при использовании розеток и кросс-шкафов (когда разводку в шкафу делает один человек, а розеток — другой) или при укорачивании существующего кабеля (если исполнитель не проверит тип расшивки отрезаемого разъема).

Для передачи сигналов на линиях связи **100 Мбит/с** используются только две пары из четырех (в **Gigabit Ethernet** на скорости **1000 Мбит/с** задействованы все четыре пары), имеющихся в кабеле. Хотя это и не разрешается стандартами, администратор может задействовать оставшиеся пары, например, использовать средние контакты (пара голубой — бело-голубой провода) под телефон, а коричневую пару в качестве замены одной из сигнальных при обнаружении обрыва в кабеле. Следует учитывать, что для питания

устройств поверх сети **Ethernet (PoE)** используется центральная пара и что в гигабитных каналах задействованы все проводники.

Отдельно повторюсь, что вопреки навязанным, последнее время, маркетологами мифам, два компьютера можно соединить витой парой без коммутатора, т.е напрямую. Как это сделать, опять же, см.выше по тексту.

Внимание: качество патч-корда

Часто в условиях фирмочек экономят на кабелях (патч-кордах), покупая самые дешевые бухты (или куски метража) из возможных. Я бы рекомендовал всеми силами избегать покупки сомнительных **noname** кабелей с кривой разметкой (автор даже встречал отбраковку вовсе без неё), без экранирования и в общем и целом сомнительного качества.

Запомните, от физического качества кабеля напрямую зависит качество работы сети, скорость в этой сети, потеря пакетов и то, как долго, собственно, эта сеть протянет.

Обратите внимание, что по данным одного из производителей патч-кордов, две трети их не проходят тестирования после изготовления в промышленных условиях (да даже больше, если судить по практике автора), а посему трудно ожидать стабильности характеристик от изделий, изготовленных в кустарных условиях,

поэтому стоит комплектовать рабочие места только нормальными известными производителями.

Говоря проще - избегайте экономить на СКС, ибо, учитывая стоимость в человекочасах поиска последующей после экономии проблемы и перемонтажа из-за какого-нибудь кривого разрыва в середине очередного кабеля, потом выходит на несколько порядков дороже, а уж сколько нервов Вы пережгете, словами, собственно, описать сложно.

Составные СКС и наращивание длины

Стандарты СКС не предусматривают возможности составления линий передачи данных из нескольких участков. Это значит, что соединение компьютера с коммутатором должно состоять из патч-корда, информационной розетки, кабеля, коммутационной панели (на которую "расшит" кабель), патч-корда до разъема на коммутаторе. Если по каким-либо причинам длины кабеля недостаточно (кабель поврежден или сотруднику выделено другое место), то наращивать его нельзя.

Я вообще кстати не рекомендую наращивание как таковое.. Однако, к сожалению, в суровой жизненной практике, достаточно часто приходится сталкиваться с необходимостью удлинения кабеля.

Ни в коем случае не рекомендуется постоянно применять на практике нижесказанное (ибо скрутки всегда плохо), но раз уж

порой приходится, то должен отметить, что на 100-мегабитных сетях кабели вполне удовлетворительно функционируют при наличии скрутки (или, что еще лучше, паяного соединения) двух кусков. В общем.. При реализации наращивании или устранения разрыва, стоит минимизировать длину участка кабеля, на котором нарушен шаг витой пары. И поскольку качество контактов скрутки может существенно снизиться через некоторый промежуток времени из-за воздействия внешних условий (например, окисления под воздействием влаги), следует **обязательно** позаботиться о надежной изоляции такого соединения (хотя бы типовой изолентой).

Требования пожарной безопасности

Основные требования пожарной безопасности при прокладке кабелей в офисе заключаются в следующем:

- Кабели, каналы, розетки и т. п. должны соответствовать определенной категории пожароустойчивости; обычно это выполняется при помощи современных элементов СКС;
- Силовые и информационные кабели при прокладке в одном канале должны быть разделены сплошной перегородкой. Минимальное расстояние от силовых кабелей до информационных определяется по специальным нормативам в зависимости от нагрузки, но обычно не должно быть менее 12—15 см;

- Отверстия, выполненные для прокладки кабелей между помещениями, должны быть закрыты легкоудаляемым негорючим материалом, например, цементом или гипсом низкой прочности, минеральной ватой и тп;
- При прокладке кабелей в пространстве над навесным потолком недопустимо использовать горючие материалы.

В случае монтажа кабельной системы под фальшполом нормативами налагаются более строгие ограничения. Например, разделение пространства под фальшполом на зоны, отделяемые друг от друга несгораемыми материалами и тп.

Топология сети

Под топологией понимают схему расположения и соединения устройств сети. Обычно выделяют две топологии: физическую, которая описывает реальное расположение устройств и наличие каналов связи между ними, и логическую, создаваемую поверх физической и описывающей пути передачи данных.

Если при построении сети внутри здания обычно удается придерживаться иерархии связей "здание — этаж — рабочее место", то в случае размещения предприятия в нескольких зданиях структура сети в значительной степени определяется возможностями прокладки внешних кабелей. Наличие кабельной канализации, воздушных линий связи, кабельных эстакад и тп.

достаточно жестко определяют возможные направления каналов передачи данных.

Поскольку стоимость прокладки кабелей между зданиями достаточно высока, обычно прокладывается лишь минимум связей, которые обеспечивают отказоустойчивость сетевой структуры. При этом весьма часто используется кольцевая структура, иногда снабжаемая "перемычкой" для снижения числа промежуточных узлов между двумя узлами распределения.

Что-то меня понесло. Давайте перейдем к беспроводным СКС.

Беспроводные сети (“Вафелька”)

Выполнение полного цикла работ по проектированию и монтажу структурированной кабельной сети (СКС) является весьма затратной операцией. Стоимость проектирования, стоимость самих материалов, оплата монтажных работ — эти расходы могут составить весьма значительную сумму. Даже в небольших организациях для создания сети необходимо приобрести несколько сотен метров кабеля. Для средних и крупных организаций счет уже ведется на тонны "меди". Кроме того, достаточно велики стоимости розеток, каналов и других элементов. При этом в силу большой трудоемкости в проекты обычно закладываются необходимые резервы, рабочие места, которые могут быть созданы в перспективе, и тп.

Для небольших офисов существует решение, которое уже сейчас успешно конкурирует с существующими кабельными сетями — беспроводные линии связи.

Для создания такой сети каждый компьютер должен быть оборудован беспроводным адаптером, а в офисе необходимо установить специальные устройства, — так называемые точки доступа, выполняющие, кроме того, роль хабов сети.

С другой стороны, в настоящее время все ноуты имеют **Wi-Fi**, а стоимость адаптера, устанавливаемого в "обычный" компьютер, сравнима со стоимостью стандартной сетевой платы. Если учесть, что стоимость точки доступа аналогична стоимости небольшого коммутатора, то переход на беспроводную сеть может быть экономически оправдан для многих организаций, а в некоторых случаях (например, аренда помещения без права выполнения монтажно-строительных работ, наличие мобильных сотрудников, к примеру, официанты могут использовать мобильные компьютеры для приема заказов, врачи иметь с собой ноутбуки при проведении обхода и тд.) использование беспроводной сети может стать и единственным приемлемым решением.

Стандарты беспроводной сети

В настоящее время устройства для беспроводной сети выпускаются на основе нескольких стандартов. Я не пишу о последних версиях,

пока только о самых старых и базовых, т.к их еще большинство и далеко не все они вышли из оборота и производства.

Параметр	Стандарт		
	802.11a	802.11b	802.11g
Диапазон частот, ГГц	5	2,4	2,4
Число каналов	8	3	3
Макс.скорость передачи, Мбит/с	54	11	54 (108 со сжатием)
Совместимость	-	802.11 g	802.11 b

На практике, при проектировании лучше выбрать один стандарт беспроводного оборудования и реализовывать всё беспроводное на нём, т.к даст больше совместимости, скорости и меньше проблем, сбоев и утерь данных. По возможности хорошо бы уходить на **5 ГГц** сети, но, к сожалению, большая часть мобильных устройств всё еще живет на **2,4** передатчиках.

Проектирование беспроводной сети предприятия

С помощью беспроводных технологий можно соединять компьютеры (по принципу "точка — точка"), отдельные сегменты сетей и тп. Наиболее часто в локальных сетях устройства беспроводного доступа ставятся в качестве точки доступа (**Wireless Access Point, AP**). В этом случае персональные компьютеры подключаются к точкам доступа, через которые осуществляют доступ как в локальную сеть организации, так и в Интернет, при этом точка доступа выступает аналогом концентратора локальной сети.

После выбора стандарта беспроводной сети следует определить зоны покрытия. Одно стандартное устройство **AP** "покрывает" зону радиусом около **75—100** м. Хотя существуют различные оценки для расчетов диаграмм зон покрытия, эти величины существенно зависят от конкретных условий: планировки помещений, материала стен и тп. Лучшим способом является проведение тестовых измерений на местности, с использованием соответствующего оборудования.

Как правило, это весьма дорогостоящая операция, поэтому часто ограничиваются тестированием уровня сигнала встроенными средствами беспроводного адаптера (штатными средствами **Windows**). При этом следует учесть, что существующее на предприятии оборудование при своей работе может создать помехи беспроводной сети и предусмотреть необходимые технологические резервы. И даже при отсутствии постоянных помех используемые в беспроводной сети программы должны быть устойчивы к кратковременному исчезновению связи. Например, при работе в **1С** могут наблюдаться случаи аварийного завершения программы из-за кратковременной потери связи с сервером.

На количество устанавливаемых точек доступа будут влиять также требования к скорости передачи данных.

Следует учитывать, что скорость передачи данных снижается на максимальных расстояниях при слабом уровне сигнала. Установка дополнительных точек доступа позволит распределить между ними пользователей и повысить скорость обмена данными. В связи с этим обычно рекомендуется устанавливать одну точку доступа приблизительно на **10** клиентов, хотя технический предел подключений беспроводных устройств, как правило, составляет не одну сотню систем.

Беспроводные решения могут помочь соединить, например, два здания. Для этого созданы специализированные беспроводные мосты и направленные антенны.

Диапазоны, расстояние и шифрование

Перед установкой оборудования хорошо бы просканировать диапазон (каналы) частот **Wi-Fi** и посмотреть, какие из них наиболее загружены (т.к их несколько), и выбрать наиболее свободный, что даст хорошую стабильность соединения, скорость и наименьшее количество помех. О том, как это сделать я писал в статье “**Что такое канал Wi-Fi и как его настроить/использовать**”.

Логично, что **Wi-Fi** сети надо защищать от вторжения, что делается шифрованием. Описание шифрования я давал в статье: “**Несколько слов о шифровании Wi-Fi протокола: что к чему и зачем?**”.

Обе статьи, как Вы понимаете, рекомендуются к прочтению и вставлять их сюда я не имею возможности ввиду ограниченности размеров книги.

Модель OSI и сети

С целью систематизации СКС часто используется модель **OSI** (**Open Systems Inter-connection model**, модель взаимодействия открытых систем), условно разбивающая сетевое взаимодействие на семь уровней.

Таблица OSI и детали

Кратенько оно выглядит так:

Уровень OSI	Назначение	Примеры	Необходимое сетевое оборудование
Application (7)	Обеспечение служб сетевых приложений	Протоколы SMTP, HTTP, FTP и т. п.	—
Presentation (6)	Службы кодирования и преобразования данных, используемых на уровне приложений	Стандарты кодирования изображений (GIF, JPEG, TIFF и т. п.), аудио и видео (MPEG) и т. п.	—
Session (5)	Обеспечение коммуникаций между приложениями более высокого уровня (согласование, поддержка, завершение сессий)	Session Control Protocol (SPC) Remote Procedure Call Zone Information Protocol (AppleTalk)	—
Transport (4)	Обеспечивает передачу данных от одной точки до другой	TCP (используются соединения) UDP (передача данных без создания соединения)	—
Network (3)	Обеспечивает логическую структуру сети (сетевые адреса)	IP	Маршрутизаторы Маршрутизирующие коммутаторы
Data Link (2)	Обеспечивает передачу данных по тем или иным *физическим* каналам связи	Ethernet Token Ring FDDI Point-to-Point Protocol Frame Relay	Коммутаторы Мосты
Physical (1)	Определяет физические, механические, электрические и другие параметры физических каналов связи (напряжение, частота, максимальные длины участков и т. п.)	LAN категории 3 LAN категории 5 V.35	Концентраторы

OSI и практика

Знание уровней **OSI** обычно требуется при сдаче тех или иных сертификационных экзаменов, но на практике такое деление уже очень давно потеряло свое значение. Если первые три уровня еще можно достаточно хорошо вычленить при анализе того или иного сетевого проекта, то классифицировать функциональность оборудования по остальным уровням достаточно сложно и часто не имеет практического смысла, если конечно Вы понимаете суть.

К слову говоря, в маркетинговых целях часто указывают в описаниях коммутаторов, что они работают, например, на уровне 4 или 7. На практике это означает только то, что при реализации определенного функционала в коммутаторах осуществляется анализ пакета данных по характеристикам, относящимся к соответствующим уровням. Например, это происходит при операциях маршрутизации группового трафика (коммутатор анализирует пакет на принадлежность той или иной программе), приоритезации пакетов и тп.

В общем, если хотите заморочиться, то можете почитать на эту тему ту же самую Википедию. По факту же, я не уверен, что Вам это нужно в том виде, в каком оно даётся.

Терминология компьютерной сети

Неполным была бы тема сетей без описания терминологии. В данном случае, к сожалению, статью по этой теме придется привести полностью, т.к тема безумно основополагающая, т.е базис, т.е то, что является целью этой книги.

Сетевые протоколы TCP/IP, NWLink IPX/SPX, NetBEUI

Сетевой протокол - это набор программно реализованных правил общения между компьютерами. Этакий язык, на котором компьютеры разговаривают друг с другом и передают информацию. Ранее компьютеры были, так сказать, многоязычны и в старых версиях Windows использовался целый набор протоколов, - **TCP/IP, NWLink IPX/SPX, NetBEUI**. Ныне же пришли к общей договоренности, и стандартом стало использование исключительно протокола **TCP/IP**, а посему речь далее пойдет именно о нем.

Когда говорят о **TCP/IP**, то обычно подразумевают под этим именем множество различных… правил или, скажем, стандартов, которые прописаны с использованием (или под использование) этого протокола. Так, например, есть правила, по которым осуществляется обмен сообщениями между почтовыми серверами и есть правила, по которым конечный пользователь получает в свой ящик письма.

Имеются правила для проведения видеоконференций и правила для организации "телефонных" переговоров по Интернету. Фактически, это даже не то чтобы правила... Скорее этакая грамматика, что ли. Ну знаете, в английском одна структура построения диалогов, в французском другая. Вот и в **TCP/IP** нечто подобное, т.е. некая связка различных грамматических правил представляет собой как раз цельный протокол **TCP/IP** или, точнее говоря, стек протоколов **TCP/IP**.

Что такое UDP, TCP, ICMP

В рамках протокола **TCP/IP** для передачи данных используются протоколы - **TCP** и **UDP**. Многие наверняка слышали, что есть порты как **TCP**, так и **UDP**, но не все знают в чем разница и что это вообще. И так...

Передача данных по протоколу **TCP** (**Transmission Control Protocol** - Протокол Управления Передачей) предусматривает наличие подтверждений получения информации. "-Ну, мол, - получил? -Получил!" Если же передающая сторона не получит в установленные сроки необходимого подтверждения, то данные будут переданы повторно.

Поэтому протокол **TCP** относят к протоколам, предусматривающим соединение, а **UDP** (**User Datagram Protocol** — Протокол Пользовательских Датаграмм) - нет. **UDP** применяется в тех случаях, когда не требуется подтверждения приема (например,

DNS-запросы или **IP**-телефония (яркий представитель которой, - **Skype**)). То есть разница заключается в наличии подтверждения приема. Казалось бы, "Всего то!", но на практике это играет важную роль.

Есть еще также протокол **ICMP** (**Internet Control Message Protocol** — межсетевой протокол управляющих сообщений), который используется для передачи данных о параметрах сети. Он включает в себя служебные типы пакетов, таки как **ping, distination unreachable, TTL** и пр.

IP-адрес - что это такое и зачем нужно?

У всех он есть, но не все имеют представление что за адрес такой и почему вообще без него нельзя. Рассказываю.

IP-адрес - 32-х битное число, используемое для идентификации компьютера в сети. Адрес принято записывать десятичными значениями каждого октета этого числа с разделением полученных значений точками. Например, 192.168.101.36.

IP-адреса уникальны, - это значит, что каждый компьютер имеет свое собственное сочетание цифр, и в сети не может быть двух компьютеров с одинаковыми адресами. **IP**-адреса распределяются централизованно, интернет-провайдеры делают заявки в национальные центры в соответствии со своими потребностями. Полученные провайдерами диапазоны адресов распределяются

дальше между клиентами. Клиенты, в свою очередь, сами могут выступать в роли провайдера и распределять полученные **IP**-адреса между субклиентами и т.д.

При таком способе распределения **IP**-адресов компьютерная система точно знает "расположение" компьютера, имеющего уникальный **IP**-адрес; - ей достаточно переслать данные в сеть "владельца", а провайдер в свою очередь проанализирует пункт назначения и, зная, кому отдана эта часть адресов, отправит информацию следующему владельцу поддиапазона **IP**-адресов, пока данные не поступят на компьютер назначения.

Для построения же локальных сетей выделены спец. диапазоны адресов. Это адреса 10.x.x.x, 192.168.x.x, 10.x.x.x, с 172.16.x.x по 172.31.x.x, 169.254.x.x, где под x- имеется ввиду любое число это от 0 до 254. Пакеты, передаваемые с указанных адресов, не маршрутизируется, иными словами, попросту не пересылаются через Интернет, а поэтому в различных локальных сетях компьютеры могут иметь совпадающие адреса из указанных диапазонов. Т.е., в компании ООО "Рога и копыта" и ООО "Вася и компания" могут находится два компьютера с адресами 192.168.0.244, но не могут, скажем, с адресами 85.144.213.122, полученными от провайдера интернета, т.к. в интернете не может быть два одинаковых **IP**-адреса. Для пересылки информации с таких компьютеров в Интернет и обратно используются спец.

программы и устройства, которые заменяют локальные адреса реальными при работе с интернетом.

Иными словами, данные в Сеть пересылаются с реального **IP**-адреса, а не с локального. Этот процесс происходит незаметно для пользователя и называется трансляцией адресов. Хочется так же упомянуть, что в рамках одной сети, скажем, компании, ООО "Рога и копыта", не может быть два компьютера с одним локальным **IP**-адресом, т.е., в указанном выше примере имелось ввиду, что один компьютер с адресом 192.168.0.244 в одной компании, второй с таким же адресом - в другой. В одной же компании два компьютера с адресом 192.168.0.244 попросту не уживутся.

Вы наверняка слышали такие термины как внешний **IP** и внутренний **IP**, постоянный (статический **IP**) и переменный (динамический) **IP**. В двух словах о них:

- Внешний **IP**, - это как раз тот самый **IP**, который выдает Вам провайдер, т.е. Ваш уникальный адрес в интернете, например, - 85.144.24.122;
- Внутренний **IP**, - это локальный **IP**, т.е. Ваш **IP** в локальной сети, например, - 192.168.1.3;
- Статический **IP**, - это **IP**, который не меняется с каждым подключением, т.е. закреплен за Вами твердо и навсегда;
- Динамический **IP**, - это плавающий **IP**-адрес, который меняется с каждым подключением.

Тип Вашего **IP** (статический или динамический) зависит от настроек провайдера.

Маска адреса или подсеть - что это такое и зачем нужно?

Понятие подсети введено, чтобы можно было выделить часть **IP**-адресов одной организации, часть другой и тд. Подсеть представляет собой диапазон **IP**-адресов, которые считаются принадлежащими одной локальной сети. При работе в локальной сети информация пересылается непосредственно получателю. Если данные предназначены компьютеры с **IP**-адресом, не принадлежащим локальной сети, то к ним применяются специальные правила для вычисления маршрута для пересылки из одной сети в другую.

Маска, - это параметр, который сообщает программному обеспечению о том, сколько компьютеров объединено в данную группу (подсеть). Маска адреса имеет такую же структуру, как и сам **IP**-адрес: это набор из четырех групп чисел, каждое из которых может быть в диапазоне от 0 до 255. При этом, чем меньше значение маски, тем больше компьютеров объединено в данную подсеть. Для сетей небольших компаний маска обычно имеет вид 255.255.255.x (например, 255.255.255.224).

Маска сети присваивается компьютеру одновременно с **IP**-адресом. Так, например, сеть 192.168.0.0 с маской 255.255.255.0 может содержать в себе компьютеры с адресами от 192.168.0.1 до

192.168.254. А сеть 192.168.0.0 с маской 255.255.255.128допускает адреса от 192.168.0.1 до 192.168.0.127. Думаю, смысл понятен. Как правило сети с небольшим возможным числом компьютеров используются провайдерами с целью экономии **IP**-адресов. Например, клиенту, может быть назначен адрес с маской 255.255.255.252. Такая подсеть содержит в себе только два компьютера.

После того как компьютер получил **IP**-адрес и ему стало известно значение маски подсети, программа может начать работу в данной локальной подсети. Однако же, чтобы обмениваться информацией с другими компьютерами в глобальной сети, необходимо знать правила, куда пересылать информацию для внешней сети. Для этого служит такая характеристика как адрес шлюза (**Gateway**).

Что такое Шлюз (Gateway) - что это такое и зачем нужно?

Шлюз - это устройство (компьютер или маршрутизатор), которое обеспечивает пересылку информации между различными **IP**-подсетями. Если программа определяет (по **IP** и маске), что адрес назначения не входит в состав локальной подсети, то она отправляет эти данные на устройство, выполняющее функции шлюза. В настройках протокола указывают **IP**-адрес такого устройства.

Для работы только в локальной сети шлюз может не указываться.

Для индивидуальных пользователей, подключающихся к Интернету, или для небольших предприятий, имеющих единственный канал подключения, в системе должен быть только один адрес шлюза, - это адрес того устройства, которое имеет подключение к Интернету. При наличии нескольких маршрутов будет существовать несколько шлюзов. В этом случае для определения пути передачи данных используется таблица маршрутизации.

Таблицы маршрутизации - что это такое и зачем нужно?

Организация или пользователь может иметь несколько точек подключения к Интернету (например, резервные каналы на случай, если у первого провайдера что-то выйдет из строя, а интернет таки очень нужен) или содержать в своей структуре несколько **IP**-сетей. В этом случае, чтобы система знала каким путем (через какой шлюз) посылать ту или иную информацию, используются таблицы маршрутизации.

В таблицах маршрутизации для каждого шлюза указываются те подсети Интернета, для которых через них должна передаваться информация.

При этом для нескольких шлюзов можно задать одинаковые диапазоны, но с разной стоимостью передачи данных: например, информация, будет пересылаться по каналу, имеющему самую низкую стоимость, а в случае выхода его из строя по тем или иным

причинам, автоматически будет использоваться следующее доступное наиболее дешевое соединение.

Порты - что это такое и зачем нужно?

При передаче данных кроме **IP**-адресов отправителя и получателя пакет информации содержит в себе номера портов. Пример: 192.168.1.1:80, - в данном случае 80 - это номер порта. Порт, - это некое число, которое используется при приеме и передаче данных для идентификации процесса (программы), который должен обработать данные. Так, если пакет послан на 80-й порт, то это свидетельствует, что информация предназначена серверу **HTTP**.

Номера портов с 1-го до 1023-й закреплены за конкретными программами (так называемые **well-known**-порты). Порты с номерами 1024-65 535 могут быть использованы в программах собственной разработки. При этом возможные конфликты должны решаться самими программами путем выбора свободного порта. Иными словами, порты будут распределяться динамически: возможно, что при следующем старте программа выберет иное значение порта, если, конечно, Вы вручную через настройки не задавали ей порт.

Что есть MAC-адрес и зачем он нужен

Дело в том, что пересылаемые пакеты в сети адресуются компьютерам не по их именам и не на **IP**-адрес. Пакет

предназначается устройству с конкретным адресом, который и называется MAC-адресом.

MAC-адрес, - это уникальный адрес сетевого устройства, который заложен в него изготовителем оборудования, т.е. это этакий проштампованный номер Вашей сетевой карты. Первая половина MAC-адрес представляет собой идентификатор изготовителя, вторая - уникальный номер данного устройства.

Как правило, MAC-адрес бывает требуется для идентификации, скажем, у провайдера (если провайдер использует привязку по мак-адресу вместо логина-пароля) или при настройке маршрутизатора.

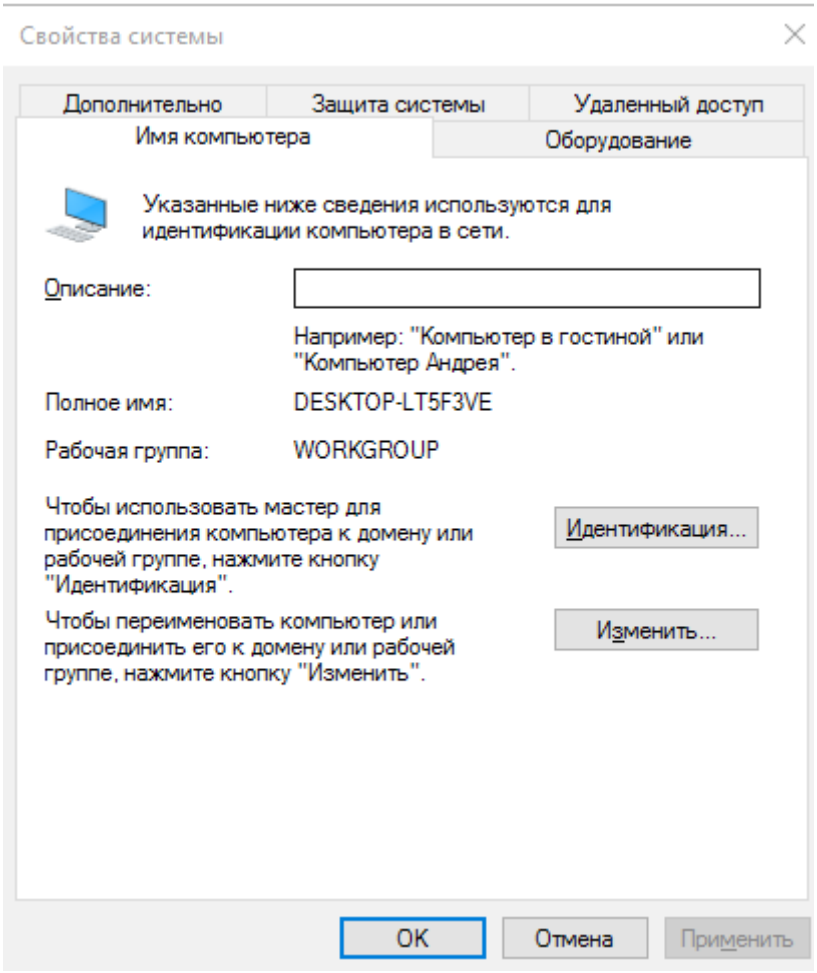
Как сделать сеть на программном уровне?

Сети предприятий строятся по простому принципу, - уровень рабочих групп или уровень домена. Про добавление компьютеров на сервер и в домен, мы рассматривали на примере **Windows Server** в соответствующей главе.

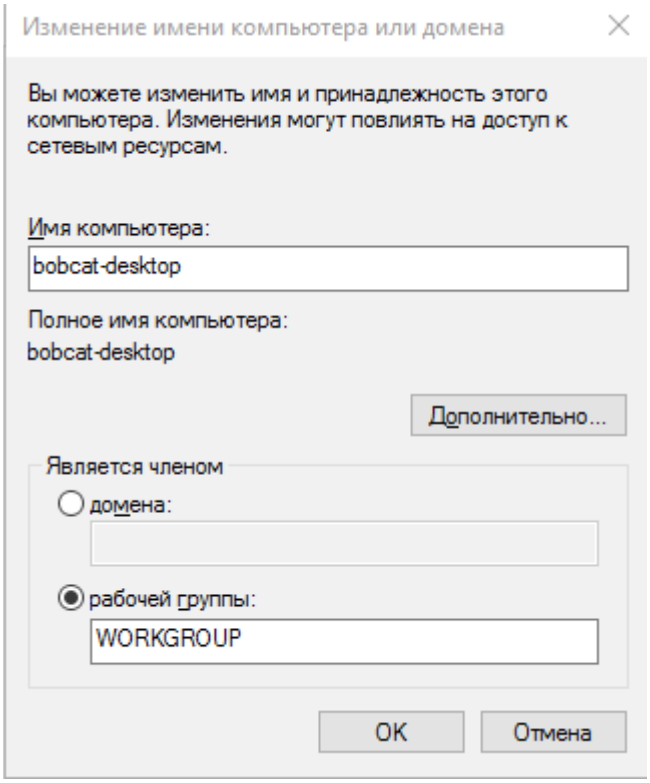
Общие принципы на уровне рабочих групп, я кратко опишу ниже. Глобально, в **Windows**, начиная с XP (или даже пораньше) есть мастер создания локальной сети на уровне рабочих групп, который позволяет буквально в два-три клика сделать сеть и чем дальше, тем проще это происходит.

Создание локальных сетей на уровне рабочих групп

Всё просто. Для начала нам надо установить одинаковое имя рабочей группы для всех компьютеров, которые должны быть подключены к локальной сети.

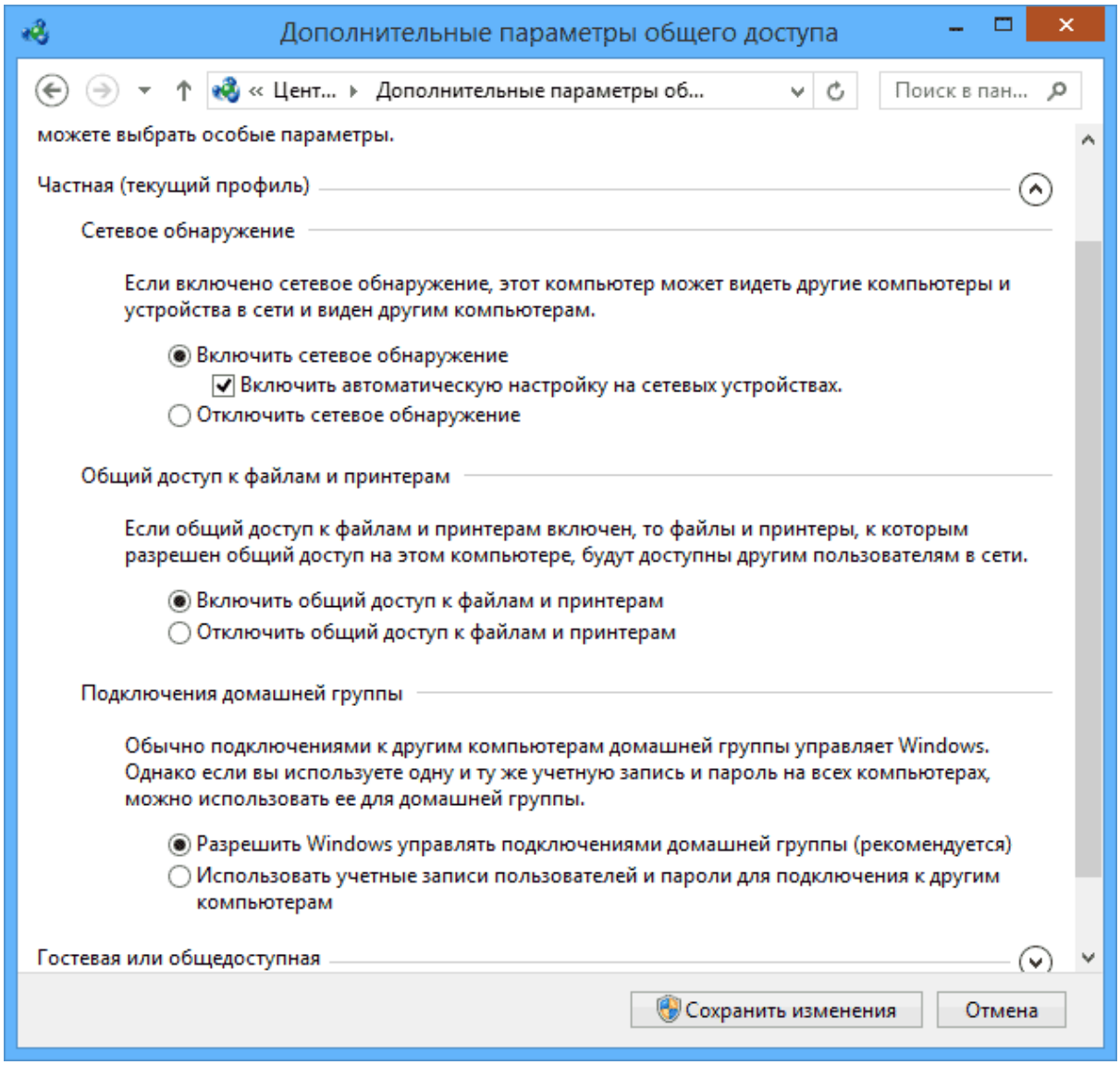


Для этого жмем в “**Win + R**” на клавиатуре (или “**Пуск - Выполнить**”) и вводим команду **sysdm.cpl**, чтобы открыть свойства системы (это можно сделать также из панели управления и иными способами).

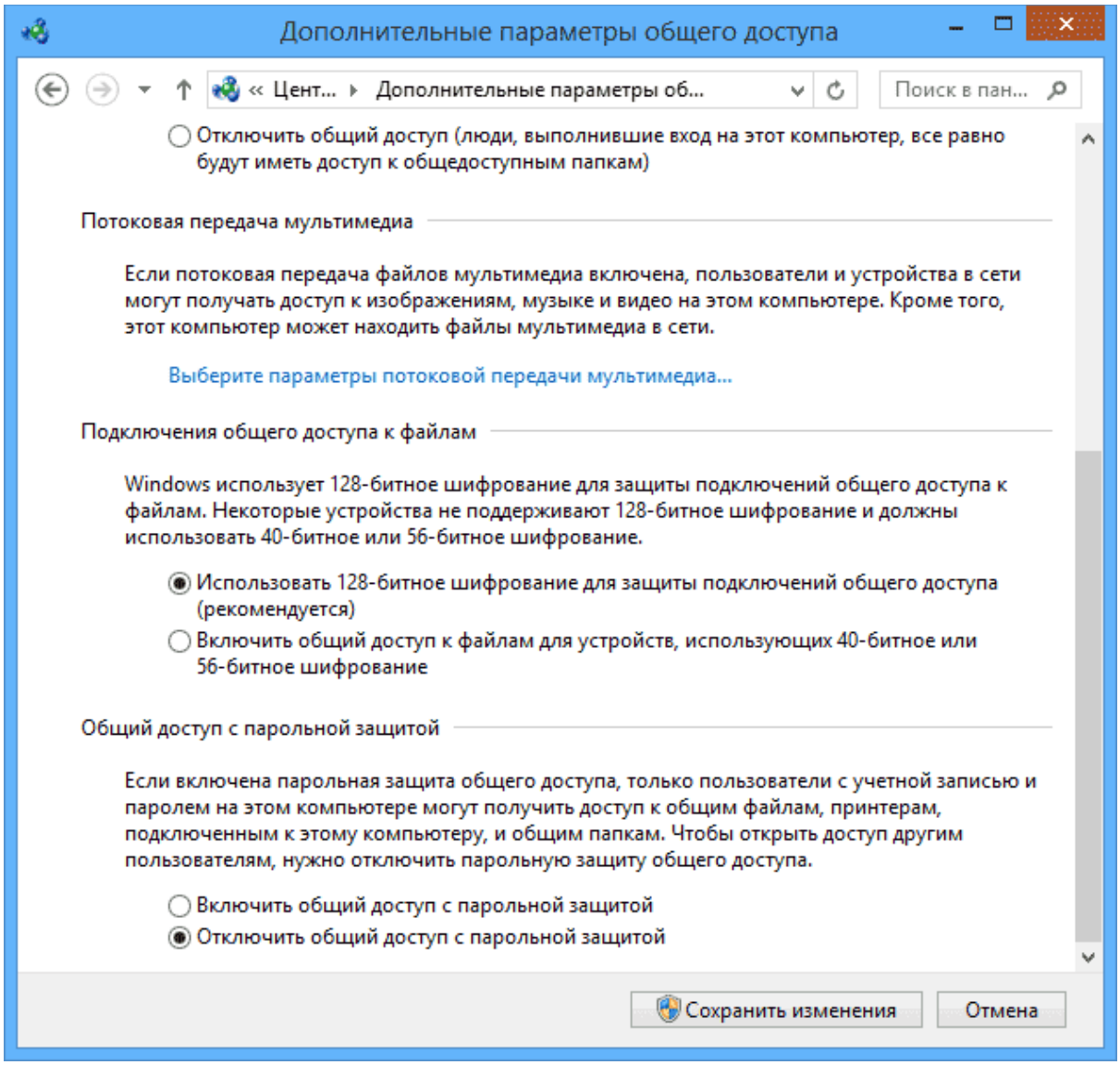


Откроется как раз нужная нам вкладка, в которой можно увидеть, к какой рабочей группе принадлежит компьютер, как правило это **WORKGROUP**. Для того, чтобы изменить имя рабочей группы, нажмите **«Изменить»** и задайте новое имя (не используйте кириллицу, т.е русскоязычный текст). Как я уже сказал, имя рабочей группы на всех компьютерах должно совпадать, т.е процедуру стоит проделать на всех компьютерах.

Далее зайдите в “**Центр управления сетями и общим доступом**” (есть где-то начиная с **Windows Vista**), который также живет в панели управления.



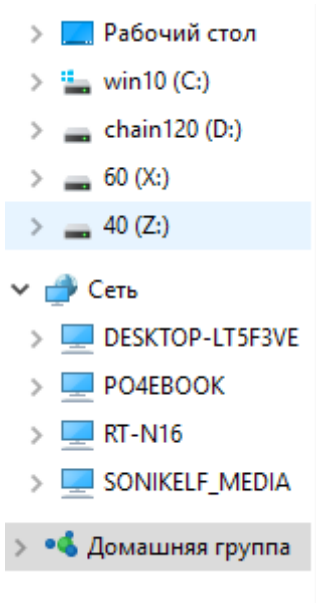
Для всех профилей сети включите сетевое обнаружение,



автоматическую настройку, а также общий доступ к файлам и принтерам.

Далее перейдите к пункту «**Дополнительные параметры общего доступа**» - «**Все сети**» и в последнем пункте тыркните в «**Общий доступ с парольной защитой**», где в обычном случае, выберите «**Отключить общий доступ с парольной защитой**» и сохраните изменения.

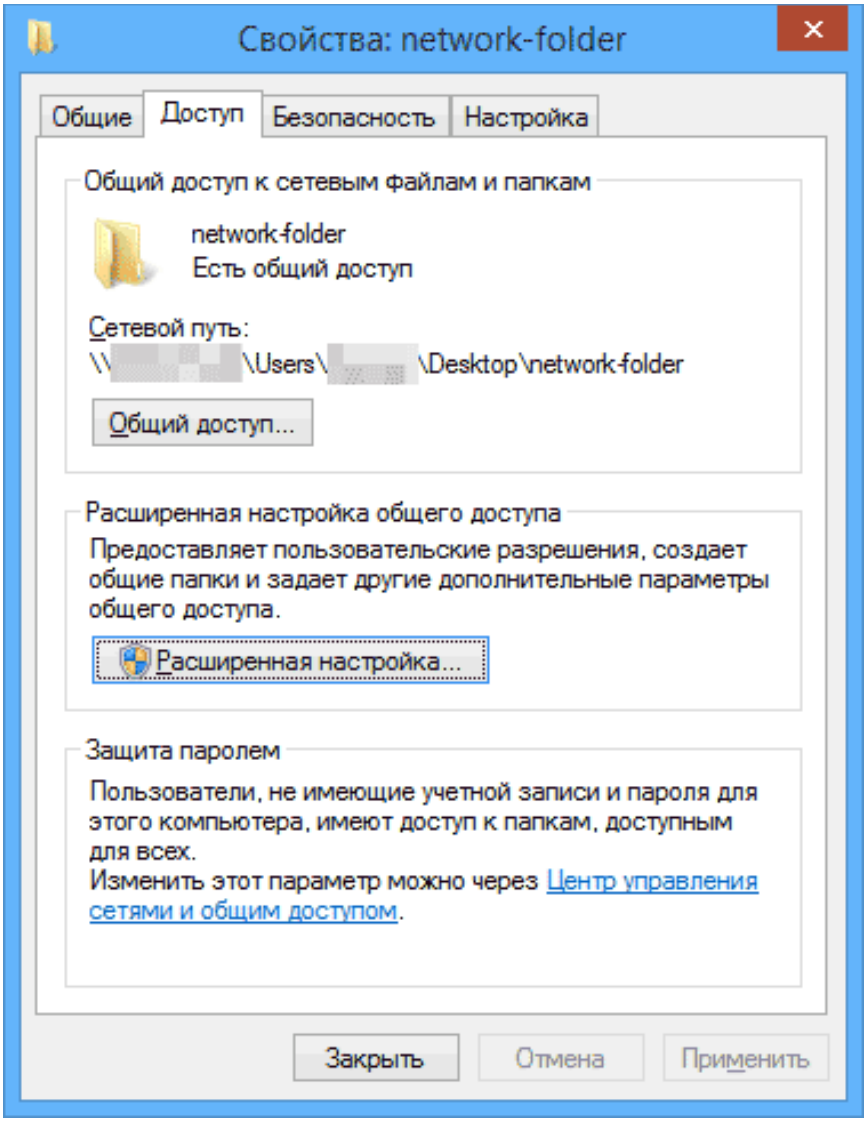
Как предварительный итог: на всех компьютерах локальной сети должно быть установлено **одно имя рабочей группы,** а также сетевое обнаружение; на компьютерах, папки с которых должны быть доступны в сети, следует включить общий доступ к файлам и принтерам и отключить общий доступ с парольной защитой.



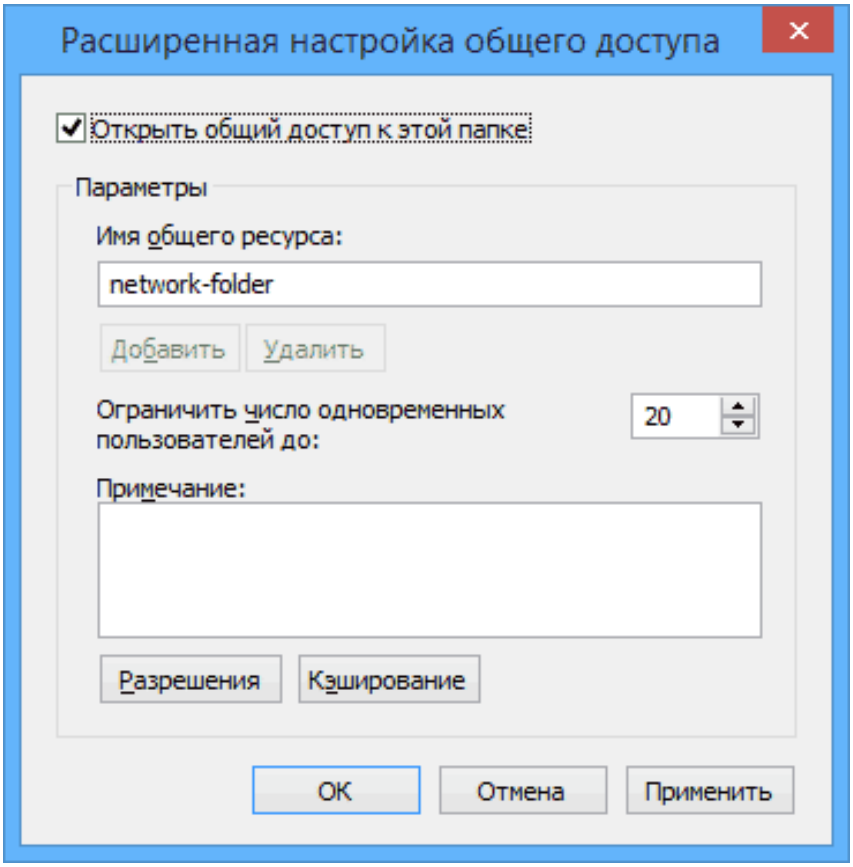
Вышеописанного достаточно, если все компьютеры в вашей домашней сети подключены к одному роутеру, а роутер раздает адреса через **DHCP** (сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети **TCP/IP**). При иных вариантах подключения может потребоваться задать статический **IP**-адрес в одной подсети в свойствах подключения **LAN**.

Предоставление доступа к файлам и папкам на компьютере

Давайте разберемся как, в случае рабочих групп, “шарить” (share) папки, т.е предоставлять общий доступ к ним.

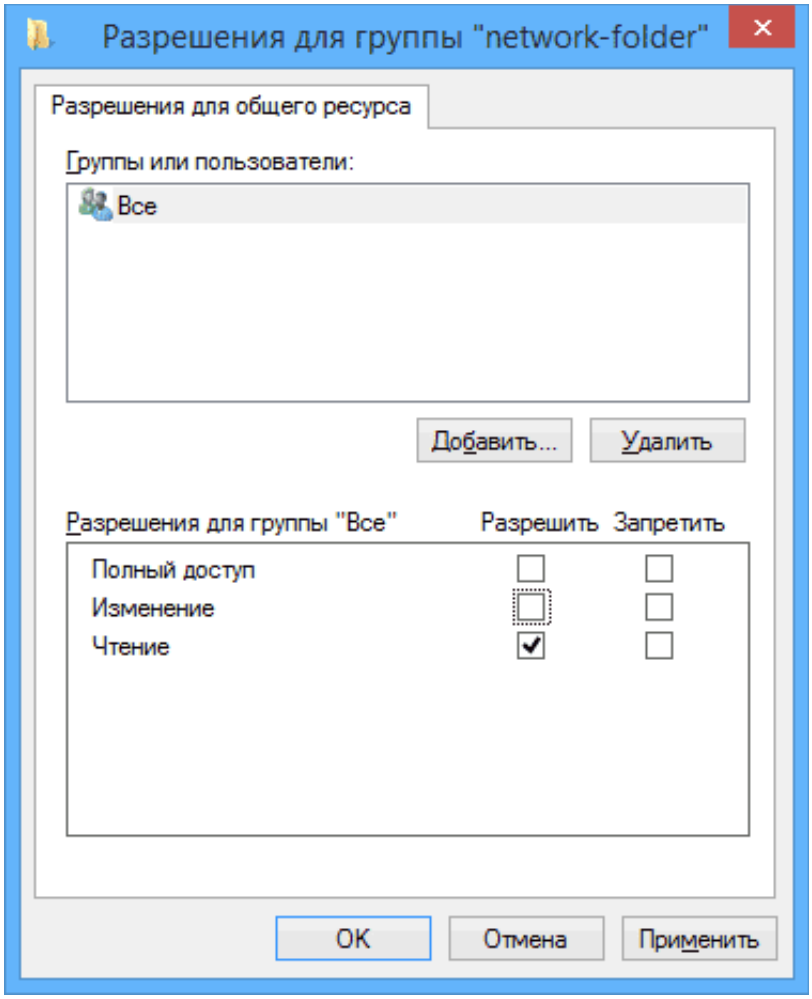


Для того, чтобы предоставить общий доступ к папке **Windows** в локальной сети, кликните правой кнопкой мыши по этой папке и



выберите пункт **«Свойства»** и перейдите к вкладке **«Доступ»**, на ней нажмите кнопку «**Расширенная настройка**».

Установите отметку «**Открыть общий доступ к этой папке**», после чего нажмите **«Разрешения»**.



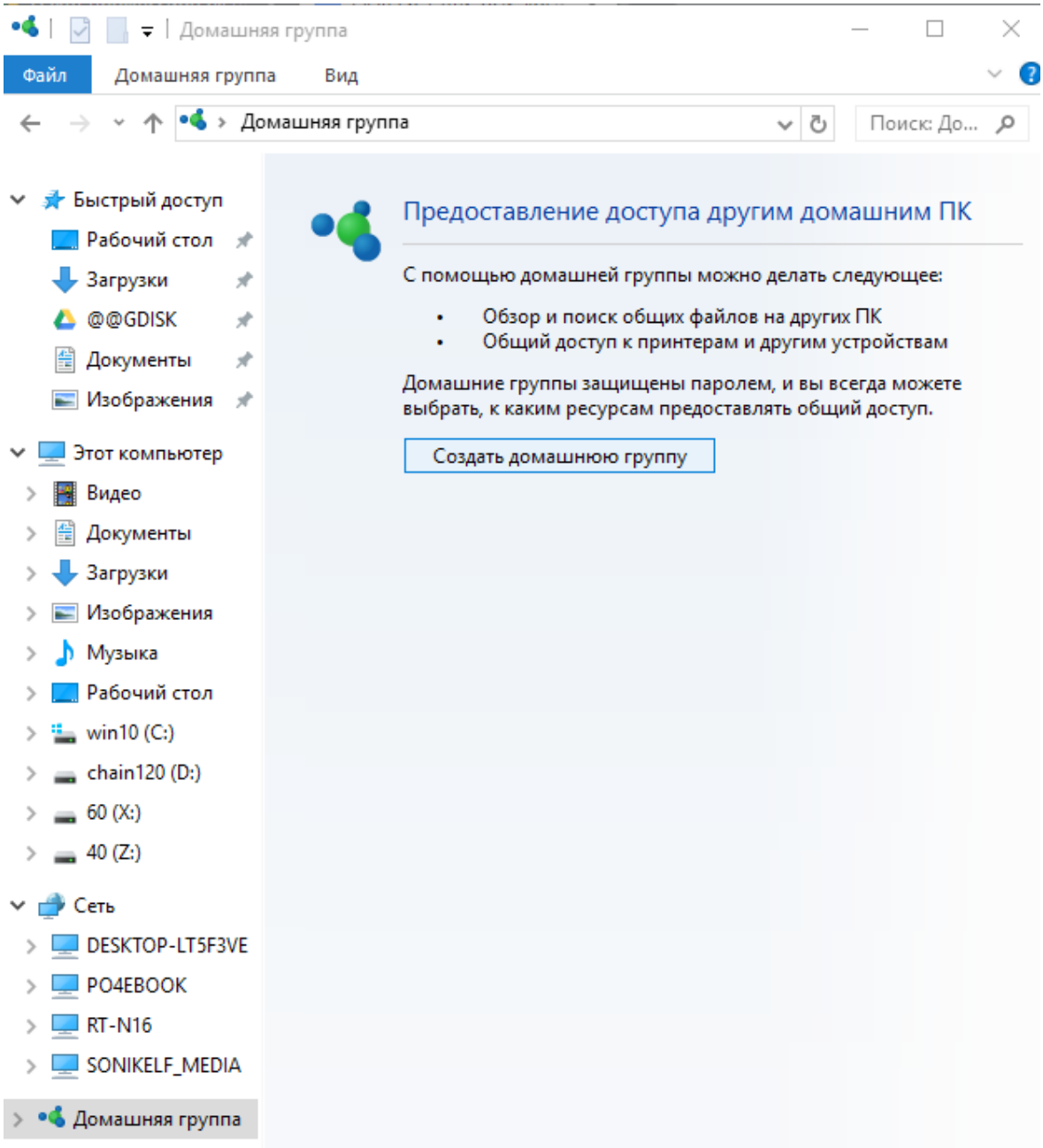
Отметьте те разрешения, которые необходимы для этой папки. Если требуется возможность только чтения, можете оставить значения по умолчанию. Примените сделанные настройки.

После этого, в свойствах папки откройте вкладку «**Безопасность**» и нажмите кнопку «**Изменить**», а в следующем окне — «**Добавить**».

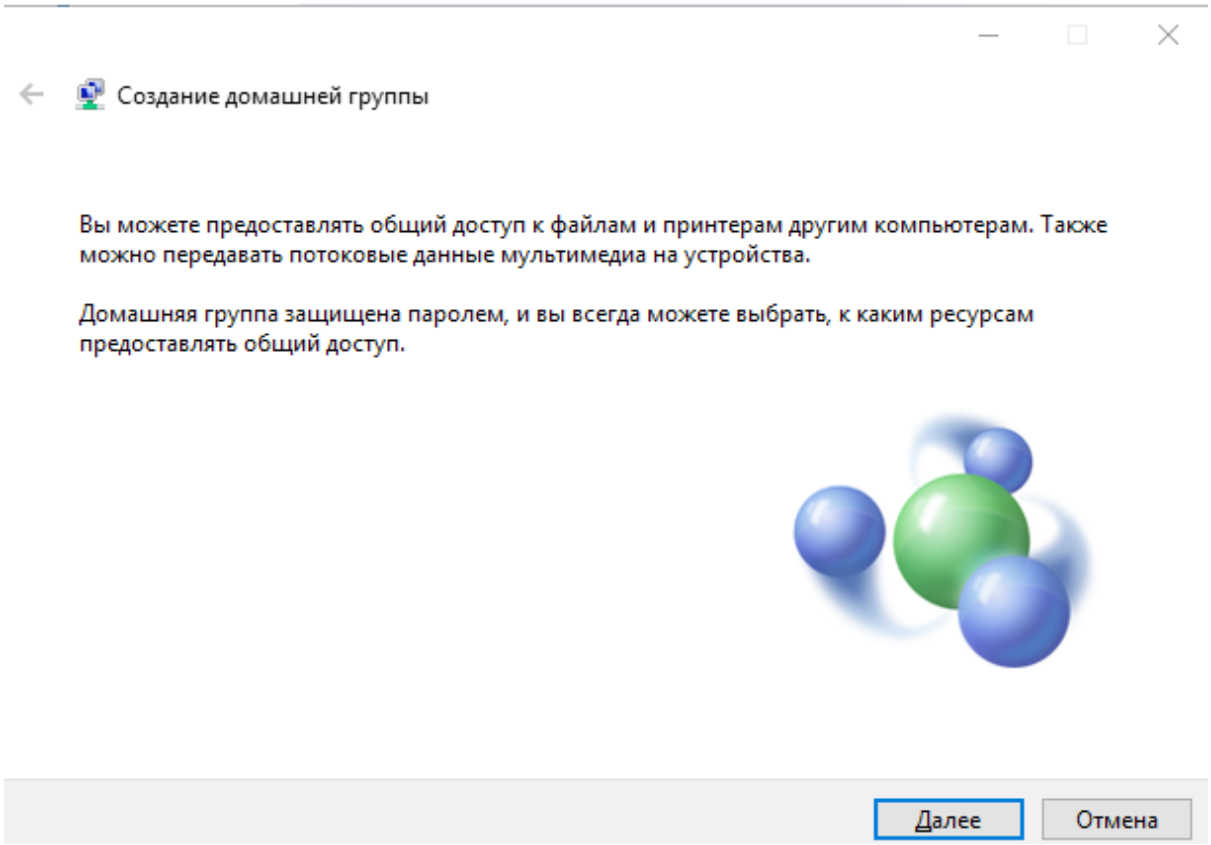
В двух словах как-то вот так и мы простейшим образом создали рабочую группу в рамках компьютеров, воткнутых в один роутер.

Использование мастера для создания сети

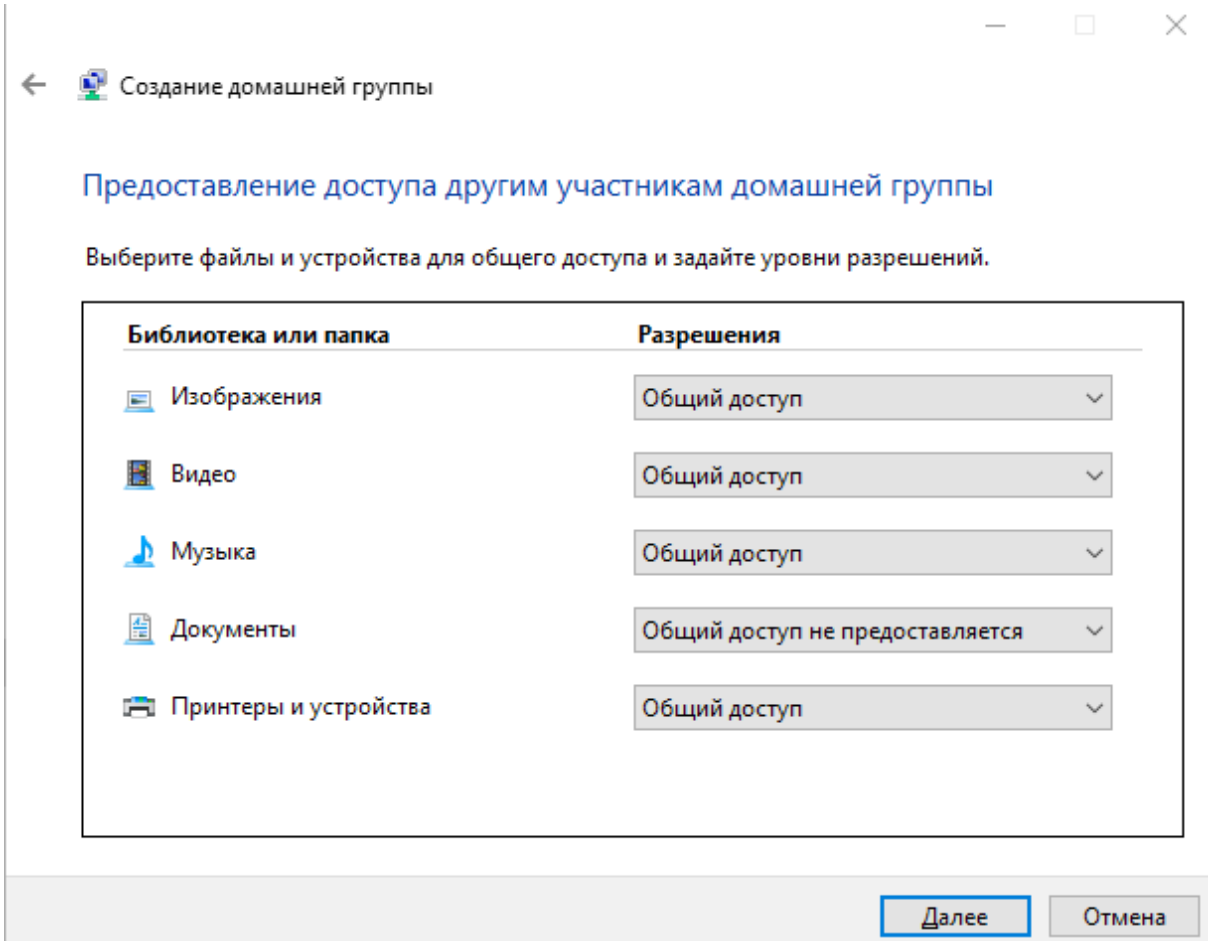
Также в **Windows** есть мастер создания рабочих групп, позволяющий делать это еще быстрее и проще. Обитает в том же центре управления сетями, местами в панели управления, либо, например в **Windows 10**, прямо в проводнике:



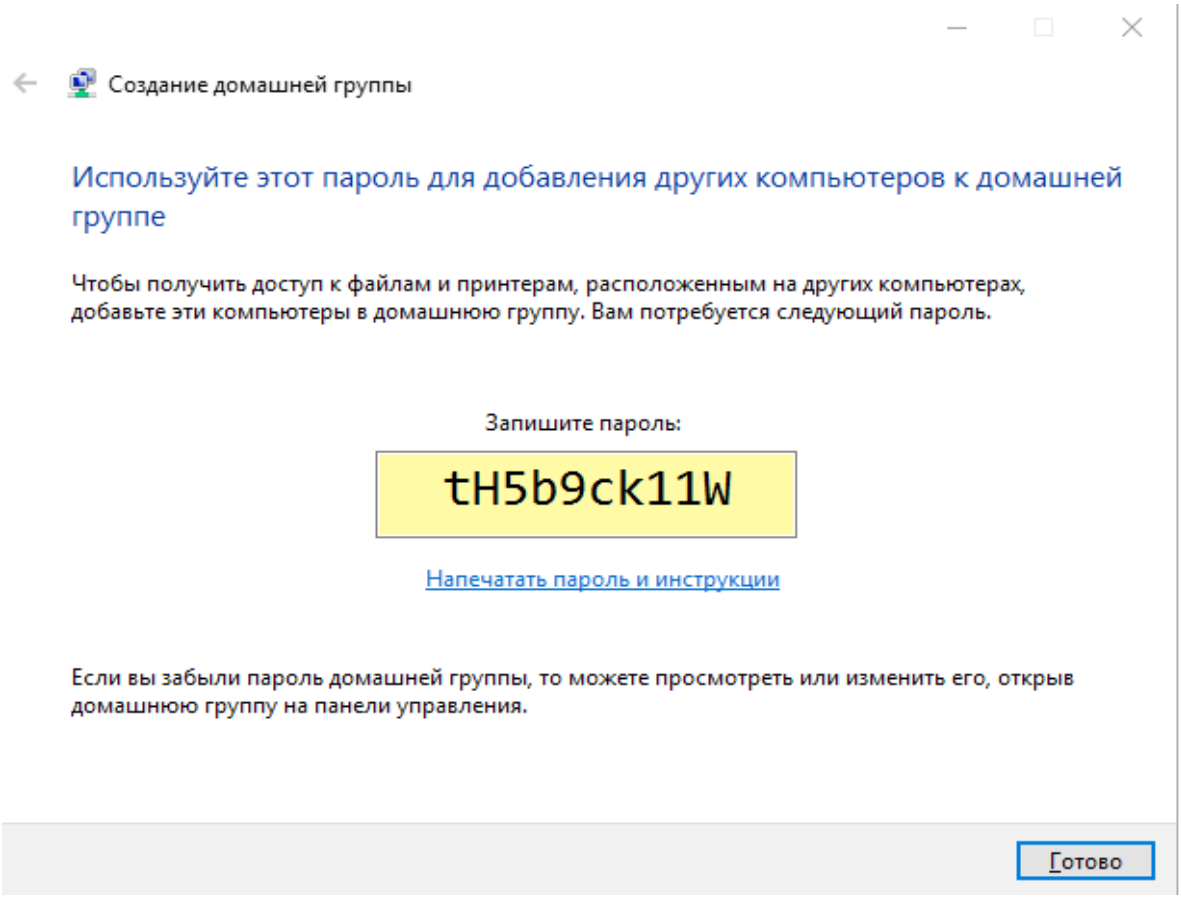
Читая написанное на экране и следуя его шагам, можно моментально настроить и создать рабочую группу или присоединить к ней компьютер:



Он же позволит быстро дать общий доступ к различным файлам:



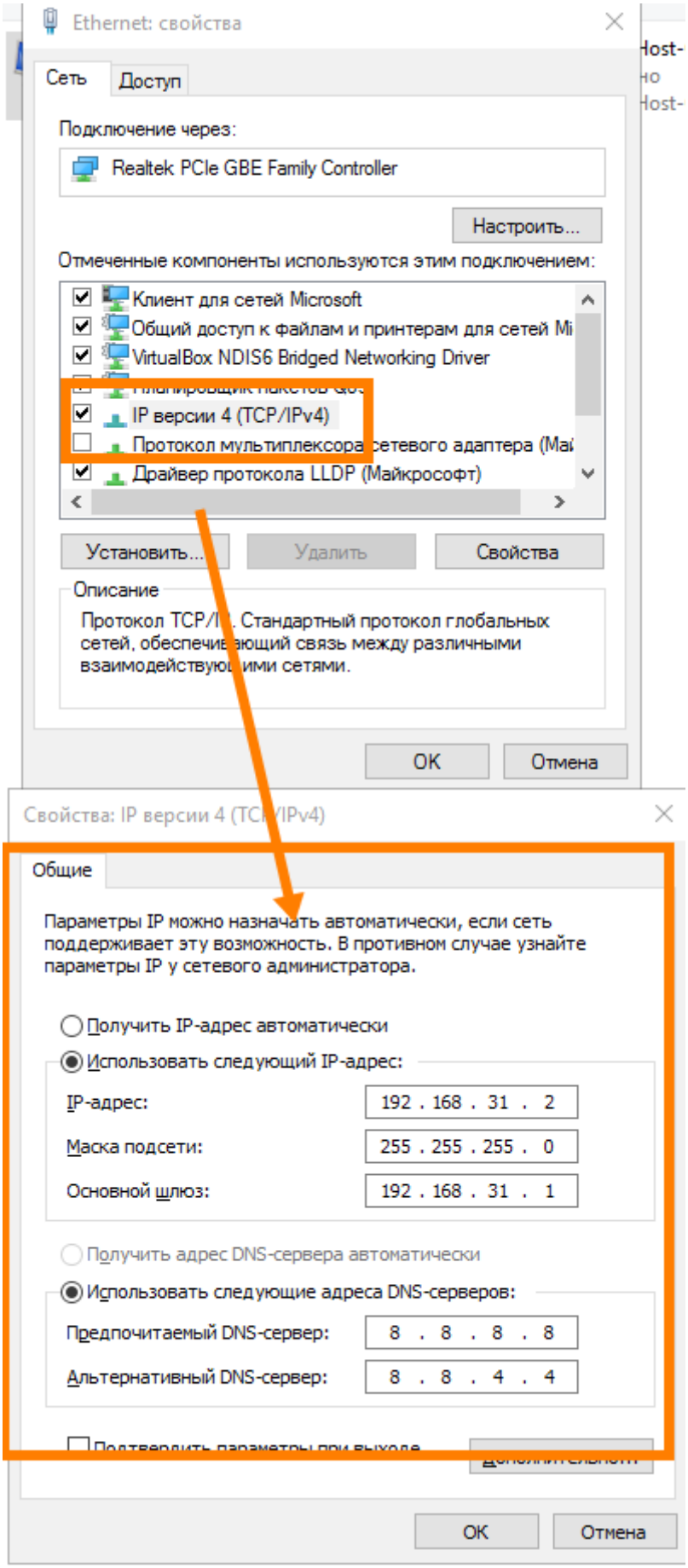
И подключить все остальные компьютеры буквально в два клика, введя данный пароль, который автоматически поможет подобрать настройки и объединить машины в сеть:



Можно даже распечатать пароль и инструкции. В общем, всё что нужно, - это просто уметь читать.

Общие принципы, которые стоит понимать

Компьютеры в сети должны находится в одном **IP**-диапазоне, подсети и желательно иметь общий шлюз и **DNS**.


Ethernet: свойства
Сеть
Доступ
Подключение через:
Realtek PCIe GBE Family Controller
Настроить...
Отмеченные компоненты используются этим подключением:
Клиент для сетей Microsoft
Общий доступ к файлам и принтерам для сетей Mi
VirtualBox NDIS6 Bridged Networking Driver
IP версии 4 (TCP/IPv4)
Драйвер протокола LLDP (Майкрософт)
Установить...
Удалить
Свойства
Описание
OK
Отмена
Свойства: IP версии 4 (TCP/IPv4)
Общие
Параметры IP можно назначать автоматически, если сеть поддерживает эту возможность. В противном случае узнайте параметры IP у сетевого администратора.
Получить IP-адрес автоматически
Использовать следующий IP-адрес:
IP-адрес: 192 . 168 . 31 . 2
Маска подсети: 255 . 255 . 255 . 0
Основной шлюз: 192 . 168 . 31 . 1
Получить адрес DNS-сервера автоматически
Использовать следующие адреса DNS-серверов:
Предпочитаемый DNS-сервер: 8 . 8 . 8 . 8
Альтернативный DNS-сервер: 8 . 8 . 4 . 4
OK
Отмена

Т.е, если Вы настраиваете всё ручками, то один компьютер у Вас имеет **IP**, например, **192.168.31.2** и подсеть **255.255.255.0**, то второй должен иметь отличный, от данного **IP,** что называется, на “конце” (в последнем, четвертом октете), т.е, например, **192.168.31.3**, такую же подсеть, в идеале шлюз и **DNS**. Если же второй компьютер будет иметь **192.169.1.2** и подсеть **255.255.0.0**, то он не будет в одной сети с первым. Как и в случае, если у них указана разная рабочая группа и тд и тп.

Это, если говорить кратко, на уровне, что называется, “на пальцах”. Как видите, в создании сетей на программном уровне и/или присоединении компьютеров к серверу нет никаких сложностей, в отличии от понимания написанного в первой части этой главы, т.е физических составляющих СКС.

Послесловие к шестой главе

Тема сетей крайне обширна и ничуть не меньше серверов, как таковых. Необходимый базис я, как смог, описал, но по сетям вообще говоря, надо писать отдельную книгу или даже цикл книг.

И опять же, повторюсь, что, если всё пойдет хорошо и первая часть Вам понравится, то со временем может получится такая себе “**Война и Мир**”, но только по серверам и сетям, ибо рассказать мне еще есть что, а в одну книгу уместить всё достаточно накладно, посему… Получается то, что получается.

Такие дела.

Давайте, пожалуй, закругляться.

Глава 7. Послесловие

Предисловие к послесловию

На сим, пожалуй, всё. Есть еще несколько моментов, которые я хотел Вам сказать и показать, но, как я уже многократно говорил, в первую часть цикла книг они попросту не поместились.

Тем не менее, вместо бонусной темы, я хотел бы добавить несколько важных слов, которые часто игнорируются многими администраторами или не знакомы многим не администраторам.

Что хотелось бы сказать еще

Регламентация, документация и деятельность

Деятельность системного администратора и любого айтишника, как такового — это, де факто, непрерывный процесс возникновения проблем, их решения, появления новых вопросов и тд. Качество же работы практически пропорционально его "незаметности": чем стабильнее работает система (нет проблем у пользователей) и чем быстрее разрешаются инциденты, тем профессиональнее специалист, обслуживающий такую систему.

Стабильная работа информационной системы не получается сама собой. Обычно она является результатом планомерной работы системного администратора по обслуживанию оборудования и программных средств, по анализу возникающих событий и выработкой решений, предупреждающих тот или иной отказ (так называемый, проактивный мониторинг) и тд и тп

Из практики хочу отметить, что всегда имеет смысл регламентировать порядок разрешения возможных инцидентов, в том числе с пользователями. Под регламентированием я имею ввиду создание ведущих документов, определяющих сделанное, делаемое и то, что сделать планируется, также регистрацию заявок пользователей, возникших проблем, дат и способа их разрешения, каких-то заметок и мыслей по работе информационной системы и тд и тп. В конце концов заведите себе обычную тетрадку **A4** (а лучше штук **20** таковых).

Как простой пример можно определить время ввода в эксплуатацию нового компьютера, срок восстановления системы из резервной копии на новом оборудовании, даты плановых тех.работ и перезагрузок серверов и тп, и сделать всё это на основе банальной и своевременной статистики, диагностики производительности, профилактики и оптимизации.

Чем более подробно будут классифицированы возможные ситуации, тем меньше претензий потенциально возникнет у пользователей в отношении уровня их обслуживания.

Не стоит оценивать факт наличия подобного регламента лишь с точки зрения контроля над айтишниками, сис.админами (если Вы руководитель и тп). Данный “документ” может служить аргументом, например, для переноса сроков завершения работ в случае одновременного возникновения нескольких неисправностей. Кроме того, подобный регламент может быть основанием для решений руководителей по закупке нового оборудования и/или в случае конфликтных ситуаций, как с пользователями, так и с самим руководством.

Бумага бумагой, но конечно же еще неплохо, если вы сможете развернуть ту или иную автоматизированную форму для фиксирования обращений пользователей и контроля их исполнения. Эта же программа может сослужить хорошую службу администраторам, только приступающим к работе в организации, в качестве базы знаний данного предприятия, по которой можно осуществлять предварительное обучение нового специалиста. Как правило, для этого используются тикет-системы, большая часть из которых существует в бесплатном виде и элементарно разворачиваются на сервере, позволяя решать множество проблем.

Подобрать, кстати, тикет-системы и иной софт такого уровня не составляет особого труда, поскольку большинство требований, предъявляемых на малых и средних предприятиях к данному классу ПО, реализовано в программах с открытым кодом. Чуть попозже, может, напишу что-либо на эту тему на “Заметках Сис.Админа” (если еще не написал на момент чтения Вами этой книги).

Немного про этику и сопутствующие нюансы

Думаю, что для Вас не секрет, что по роду своей деятельности системный администратор имеет потенциальный доступ практически ко всей информации, хранящейся на предприятии в электронном виде. Мне бы хотелось кратко затронуть тут Вам этот аспект, ибо он будет полезен как для кармы, так и может избавить Вас от разного рода проблем (в частности уберечь от потери работы, зарплаты и уважения).

Именно барьеры этического плана должны удерживать Вас от соблазна узнать чужую зарплату или прочесть чью-либо корреспонденцию. Корректность также имеет большое значение в работе системного администратора. Например, многим айтишникам приходится применять софт, перехватывающие экран и клавиатуру компьютера пользователя. У меня, например, такой софт часто настроен таким образом, что при удаленном подключении на экране пользователя всегда выводится

соответствующее предупреждение. Я специально акцентирую на этом внимание, поскольку встречал в прессе высказывания "специалиста" о том, как ему нравится наблюдать за реакцией пользователей при удаленном перехвате управления, когда компьютер переставал "слушаться" владельца.

В немалой степени от админа зависят способы реализации корпоративных политик в области безопасности. С одной стороны - это желание руководителей осуществлять полный контроль над деятельностью подчиненных (перлюстрация корпоративной электронной почты, контроль посещения страниц Интернета и т. п.), с другой — право каждого на личную тайну. По данным статистики, желание полностью контролировать сотрудников чаще всего возникает у руководителей малых предприятий.

Часто Вы вынуждены будете быть дипломатом и поддерживать хорошие отношения как с руководством, так и с коллективом сотрудников, находя компромиссные решения противоречивых ситуаций.

Особенно это касается общения с пользователями, - в моей практике было множество ситуаций, когда человек был откровенно говоря неадекватен и создавал множество проблем, провоцировал на ссоры и откровенно выходил за рамки. И с этим нужно было разбираться максимально адекватно и стараясь сберечь свои (и

чужие) нервы, время и другие. Но об этом мы поговорим ближе в другой раз.

Немного о мистике и всём таком прочем

И в заключение немного лирики о мистике. Автор проекта и книги неоднократно замечал взаимосвязь между своим внутренним состоянием пользователя/айтишника и стабильностью работы системы.

Если Вы садитесь за компьютер, а особенно, сервер, в плохом настроении, то не ждите, что он ответит вам "полным пониманием" и будет работать быстро, стабильно и так далее. Учитывайте это в работе - оно Вам пригодится :)

Почему так мало и будет ли еще

Что с размером?

Само собой, что многие наверняка задаются вопросом на тему того, почему так мало страниц в книге и тому подобное. Отвечаю.

Во-первых, страниц мало по той простой причине, что стоимость печати, со времен выхода ПНПК и первого её тиража, возросла на порядок из-за кризиса в стране, всех этих санкций, курса доллара и тп. Опять же, стоимость напрямую зависит от количества

заказанных экземпляров и, вопреки ожиданиям, голосованию на сайте и Вашим обещаниям, печатных вариантов заказано куда меньше, чем ожидалось, следовательно, стоимость возросла.

По факту, за книгу и её отправку я уже не первый раз доплачиваю из своего кармана, с другой же стороны, не считаю разумным поднимать цену книги еще дороже или требовать дополнительных денег с Вас (т.к книги делаются не с точки зрения прибыли, а с точки зрения пользы), поэтому пока страдает размер книги.

Во-вторых, в отличии от тем первой книги, в серверной части достаточно сложно удержаться от того, чтобы не уйти в далёкие сложные технические дали, и при этом дать тот базис и таким стилем, который позволит Вам с чего-то начать, что, повторюсь, уже в который раз, является самоцелью книги.

Будет ли еще?

Как Вы прочитали в книге (если прочитали), то если отзывы (смотрим последнюю страницу с контактами) будут положительны, и Вы захотите, то цикл будет продолжен и у книги будет несколько частей.

Такие дела.

Послесловие к послесловию

Часть лирическая

Как было и с первой книгой, я не знаю, понравится Вам эта книга или нет, получилась она или не очень, но я очень хотел её для Вас написать, пусть даже опираясь на существующие и/или уже известные Вам статьи и информацию. Посему я повторю лирическую часть, которую включил к послесловию первой книги.

Только время и реальность покажет, стоила оно того или нет, получилась или не совсем, но одно я знаю точно, - я вложил в неё (как и в проект), свой кусочек души. И хотел бы пожелать Вам такой же страсти к чему-либо в жизни, как сейчас она есть у меня, именно в этот момент, когда я набираю эти строки.

Чего бы я реально хотел Вам сказать здесь такого не компьютерного, а вот от души? Я хотел бы Вам пожелать вечной тяги к жизни, умения ставить цели и достигать их. Умения жить. Часто мы сбиваемся с пути и забываем про самые основы основ.

Да, я знаю, что Вы не просили учить Вас здесь жизни или давать советы, да еще и в книге про компьютеры... Просто меня часто спрашивают, как стать спецом или как освоить компьютер в совершенстве, или как сделать проект подобного рода с нуля... Или что-то еще в таком духе. Ответ один - делать. Даже когда что-то не получается, и когда Вы вылезаете из зоны комфорта. Если реально

хочется, то пробуйте, идите, стремитесь и достигайте. И все получится, несмотря на то, хотели бы Вы стать настоящим айтишником или просто научиться чему-то в этой сфере.

“Хочу проиллюстрировать сказанное историей одного великого человека, у которого однозначно можно учиться силе духа.

Каждый раз, смотря на бои Джорджа Форемана, я испытываю к нему безграничное уважение. Посуди сам: олимпийский чемпион по боксу, чемпион мира, в 1973 году попал под раздачу джебов от Мухаммеда Али и был нокаутирован в восьмом раунде (его первое поражение). В 1977 году он проиграл Янгу и ушел и профессионального бокса, ему было 28 лет. Для большинства боксеров на этом заканчивается все.

В возрасте сорока лет он дал слово стать снова чемпионом мира. Народ над ним не то чтобы посмеивался, но всерьез точно не принимали. Вес Джорджа составлял под 140 килограмм, десять лет перерыва в боксе – ну что тут может быть такого?

После возвращения на ринг Форман провел 23 боя. Все выиграл нокаутом, и в итоге вырубил на ринге действующего чемпиона мира, который был младше его почти на 20 лет. В итоге – отбил обратно корону, двадцать лет спустя после потери. И стал самым возрастным в истории бокса чемпионом мира.

Ты думаешь, такой подход к реализации своего обещания характерен для многих людей? Да ладно тебе. Большинство ломаются после одной-трех попыток, сваливая все на обстоятельства непреодолимой силы, то есть отсутствие времени, лень, плохой прогресс и так далее.

“Я попадал мимо корзины более 9000 раз за свою карьеру. Я проиграл почти 300 матчей. 26 раз мне было доверено сделать решающий бросок,

и я промахивался. Я очень часто терпел неудачи в своей жизни. Именно поэтому я преуспел” (с) (Майкл Джордан).

“Люди чаще сдаются, чем терпят поражение” (с) (Генри Форд)”.

Как-то так. Это важно. Спасибо.

Часть не совсем лирическая

В этой книге существует как авторский материал, так и использованы тексты Википедии, статей и книг с моих / сторонних ресурсов, части уроков с моих курсов обучения и тд и тп, так или иначе, дополненных, переписанных или частично скопированных.

Всё это сделано на благо пользователя и донесения той информации, которая должна быть донесена, там и так, как это должно быть сделано, попросту потому, что подать иначе некоторый материал невозможно по той простой причине, что он одинаков во всех смыслах и/или совпадает с моим его видением.

В случае наличия претензий любого рода, пожалуйста, свяжитесь со мной через форму обратной связи, сайта **sonikelf.ru**

Спасибо за понимание.

Послесловие к послесловию к послесловию

Вот, наверное, и всё.

Всё для Вас.

Потому, что Вы лучшие.

Спасибо за уделенное внимание.

С уважением.

ОГЛАВЛЕНИЕ:

Антипиратская позиция.....2

Глава 0. Вводная.....6

Несколько слов лирики. О книге и авторе.....6

Об авторе: кто, зачем, почему и всё такое прочее.....7

Информация для критиков.....9

Общая информация о книге.....11

Послесловие.....12

Глава 1. Системное администрирование.....13

Предисловие к первой главе.....13

Концепция: человек и технологии.....13

Что за “зверь” такой и каков он есть/должен быть.....13

Но ведь IT-сфера очень огромна. Что же делать?.....15

Общее послесловие по теме первой главы.....17

Глава 2. Сервера и операционные системы.....19

Предисловие к второй главе.....19

Что есть сервер и зачем он нужен.....19

Производительность сервера.....21

Надежность сервера.....22

Масштабируемость серверов.....23

Управляемость сервера.....24

Классификация серверов в “природе”. Серверные роли.....25

Сервер рабочей группы и/или файл-сервер.....25

Контроллер домена.....26

Прокси сервер (+-VPN) (+-шлюз) и тп.....26

Сервер электронной почты.....27

Веб-сервер или сервер приложений.....28

Терминальный сервер.....29

Сервер баз данных.....30

Сервера приложений.....30

Сервера-брандмауеры (фаерволлы).....31

Сервер-DHCP.....32

Сервера FTP.....32

Принт-сервера.....33

Прочие серверные роли.....34

Аппаратная специализация и нюансы.....34

Размеры и другие детали внешнего исполнения.....34
Альтернативные решения.....36
Ресурсы серверов.....37
Послесловие ко второй главе.....38
Глава 3. Сервера и оборудование.....39
Предисловие к третьей главе.....39
Как выбрать серверное оборудование.....39
Общие правила выбора.....40
Процессор сервера.....41
Дисковый массив сервера.....46
Оперативная память сервера.....52
Всё остальное, что нужно.....53
Выбор вендора (производителя) оборудования.....54
Сервисные контракты.....56
Запасные компоненты на любой случай.....57
Послесловие к третьей главе.....57
Глава 4. Windows Server. Базис.....59
Предисловие к четвертой главе.....59
Windows Server - какой, почему и как.....60

Windows Server 2012 - общее описание и прочее.....60

Essentials как одна из редакций Win Server.....63

Установка Windows Server 2012 Essentials.....65

Предисловие к режимам установки.....65

Непосредственно процесс установки (в картинках).....66

Настройка Windows Server 2012 Essentials.....72

Настройка в первичном виде.....80

Создание пользователей.....82

Добавление папок сервера.....86

Настройка удаленного доступа.....91

Подключение рабочих станций.....103

Active Directory - что к чему и почему.....117

Небольшая вводная.....117

Предварительная настройка сервера для установки AD.....119

Установка роли Active Directory на Windows Server 2012.....121

Настройка доменных служб Active Directory.....124

Добавление компьютеров к домену Server 2012.....132

Подключение к домену Server 2012 компьютера, работающего под управлением Windows 7.....133

Добавление компьютеров под управлением Windows 8 к домену Server 2012.....137
Послесловие к четвертой главе.....140
Глава 5. Linux Server. Базис.....142
Предисловие к пятой главе.....142
Linux Server - какой, почему и как.....143
Linux Server - бесконечность дистрибутивов.....143
Установка, первичная настройка Ubuntu Server.....145
Маленькая вводная.....145
Загрузка и установка Ubuntu Server.....146
Графический интерфейс в Ubuntu Server - установка и использование.....167
Небольшая вводная.....167
Установка оболочки и сопутствующих “хвостов”.....167
Root и другие пользователи [+группы] в Linux.....172
Это многопользовательская система.....173
Суперпользователь (root) в Linux.....174
Теперь о команде su.....178
Немного практики.....180

Группы пользователей.....182
Корневая файловая система Linux.....191
Основные каталоги корневой файловой системы.....197
Кратенькие выводы.....203
О программах, репозиториях, пакетах и зависимостях в Linux. .204
О репозиториях в Linux. Что это и зачем нужно.....205
О пакетах и менеджерах пакетов в Linux. Что это и зачем нужно
.....206
О зависимостях в Linux. Что это и зачем нужно.....209
Несколько слов о нюансах.....211
Послесловие к пятой главе.....213
Глава 6. Сети. Базис и не только.....215
Предисловие к шестой главе.....215
СКС как часть информационных систем.....215
Понятие СКС, типология и категории.....215
Категории сетей.....217
Волоконно-оптические сети.....219
Сети 10G.....222
Обжимка кабеля пятой категории.....223

Приступаем к процессу обжатия кабеля RJ-45.....224

Примечание-повторение-дополнение.....229

Внимание: качество патч-корда.....230

Составные СКС и наращивание длины.....231

Требования пожарной безопасности.....232

Топология сети.....233

Беспроводные сети (“Вафелька”).....234

Стандарты беспроводной сети.....236

Проектирование беспроводной сети предприятия.....236

Диапазоны, расстояние и шифрование.....238

Модель OSI и сети.....239

Таблица OSI и детали.....239

OSI и практика.....240

Терминология компьютерной сети.....241

Сетевые протоколы TCP/IP, NWLink IPX/SPX, NetBEUI.....242

Что такое UDP, TCP, ICMP.....243

IP-адрес - что это такое и зачем нужно?.....244

Маска адреса или подсеть - что это такое и зачем нужно?.....246

Что такое Шлюз (Gateway) - что это такое и зачем нужно?.....248

Таблицы маршрутизации - что это такое и зачем нужно?.....248
Порты - что это такое и зачем нужно?.....249
Что есть MAC-адрес и зачем он нужен.....250
Как сделать сеть на программном уровне?.....251
Создание локальных сетей на уровне рабочих групп.....251
Предоставление доступа к файлам и папкам на компьютере…257
Использование мастера для создания сети.....260
Общие принципы, которые стоит понимать.....263
Послесловие к шестой главе.....265
Глава 7. Послесловие.....267
Предисловие к послесловию.....267
Что хотелось бы сказать еще.....267
Регламентация, документация и деятельность.....267
Немного про этику и сопутствующие нюансы.....270
Немного о мистике и всём таком прочем.....272
Почему так мало и будет ли еще.....272
Что с размером?.....272
Будет ли еще?.....273
Послесловие к послесловию.....273

Часть лирическая.....273

Часть не совсем лирическая.....276

Послесловие к послесловию к послесловию.....276

Завершающий лист с информацией.....286

Завершающий лист с информацией

Где оставлять отзывы

- Пожалуйста, по прочтению, оставьте отзыв на форуме в соответствующем разделе, а именно, здесь -

Контакты

- Почта - forall@sonikelf.ru
- Форма связи - http://sonikelf.ru/forma-svyazi/
- Скайп - Sonikelf
- Соц.сети - найдете, это не сложно.

Кто был рядом

- Обложка книги и многое из дизайна на сайте - "Василий aka KreOzoT". Очень талантливый и рукастый. Рекомендую.
- Духовный наставник, любимая женщина и вообще человек стоявший со мной всегда рядом - “Ксюха”, она же “Почь”, она же “Белова Ксения”
- Статьи, поддержка, многое из текстов/скриншотов и всё на свете, что не давало проекту пропасть и существовать - “Члены команды. Всех не перечислить, но можно искать на форуме по никам или спросить меня лично”.
- Автор - “Белов Андрей Александрович”, он же ”Sonikelf”
- Читатели - Все Вы.

Проект и сайт

- [Sonikelf's Project's] - еще услышите о нём в оффлайне. В каком формате - покажет время.

Как поддержать

- Яндекс - 41001245078461
- Webmoney - R674154538203
- Qiwi и телефон - 89161748826
- Остальное - http://sonikelf.ru/helpme/