Санкт-Петербургское государственное бюджетное
профессиональное образовательное учреждение
«Академия управления городской средой, градостроительства и печати»

УТВЕРЖДАЮ
Заместитель директора
по учебно-производственной работе
О.В. Фомичева
«26» 12 2023 г.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
по выполнению практических работ
по МДК.05.01 Технологии виртуализации и автоматизации
ПМ.05 ЭКСПЛУАТАЦИЯ ОБЛАЧНЫХ СЕРВИСОВ

для специальности

09.02.06 Сетевое и системное администрирование

Санкт-Петербург
2023 г.

Методические рекомендации рассмотрены на заседании методического совета СПб ГБПОУ «АУГСГиП»
Протокол № 2 от «29» 11 202 3 г.

Методические рекомендации одобрены на заседании цикловой комиссии информационных технологий
Протокол № 4 от «21» 11 202 3 г.

Председатель цикловой комиссии: Караченцева М.С. _____

Разработчики: преподаватели СПб ГБПОУ «АУГСГиП»

С О Д Е Р Ж А Н И Е

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА....4
1. Перечень практических работ по МДК.05.01 «Технологии виртуализации и автоматизации» 6
2. Описание порядка выполнения практических работ....8
Практическая работа №1 Работа с Hypervisor: Установка и настройка hosted.... 8
Практическая работа № 2 Работа с Hypervisor: Установка и настройка нативного Hypervisor. 8
Практическая работа № 3 Работа с Hypervisor: Установка и настройка виртуальных машин. 19
Практическая работа № 4 Работа с Hypervisor: Настройка виртуальной маршрутизации.19
Практическая работа № 5 Работа с Hypervisor: Автоматизциия развёртывания виртуальных машин 20
Практическая работа № 6 Работа с Hypervisor: Конфигураця ресурсов виртуальных машин26
Практическая работа № 7 Работа с Hypervisor: Развёртывание сервисов для конечного пользователя (Базы данных, HostePanel, Серверов сертификации и аутентификации).....26
Практическая работа № 8 Установка Kubernetes в среде Proxmox VE....27
Практическая работа №9 Настройка Kubernetes в среде Proxmox VE.... 28
Практическая работа № 10 Работа с контейнерами Kubernetes в среде Proxmox VE 34
Практическая работа № 11 Оркестрациия Kubernetes в среде Proxmox VE 36
Практическая работа № 12 Настройка логирования контейнеров. 43
Практическая работа № 13 Настройка виртуальных машин для шлюза удалённого рабочего стола 44
Практическая работа № 14 Настройка межплатформенный бесклиентский шлюз удаленного рабочего стола 53
Практическая работа № 16 Работа с Облачными бизнес-моделями IaaS: Автоматизация. развёртывание виртуальной машины.....77
Практическая работа № 17 Работа с Облачными бизнес-моделями IaaS: Балансировщик нагрузки виртуальных машин.... 77

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Рабочая тетрадь по выполнению практических работ предназначена для организации работы на практических занятиях по МДК.05.01 «Технологии виртуализации и автоматизации», которая является важной составной частью в системе подготовки специалистов среднего профессионального образования по специальности 09.02.06 «Сетевое и системное администрирование».

Практические занятия являются неотъемлемым этапом изучения учебной дисциплины и проводятся с целью:

- формирования практических умений в соответствии с требованиями к уровню подготовки обучающихся, установленными рабочей программой учебной дисциплины;
- обобщения, систематизации, углубления, закрепления полученных теоретических знаний;
- готовности использовать теоретические знания на практике.

Практические занятия по МДК.05.01 «Технологии виртуализации и автоматизации» способствуют формированию в дальнейшем при изучении профессиональных модулей, следующих общих и профессиональных компетенций:

ОК 1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 2. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;

ОК 4. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 5.1. Осуществлять развертывание облачной инфраструктуры

ПК 5.2. Проводить документирование требований и технических возможностей облачных инфраструктур

ПК 5.3. Проводить настройку виртуальных машин с использованием механизмов автоматического масштабирования и распределения нагрузки.

В рабочей тетради предлагаются к выполнению практические работы, предусмотренные учебной рабочей программой МДК.05.01 «Технологии виртуализации и автоматизации».

При разработке содержания практических работ учитывался уровень сложности освоения студентами соответствующей темы, общих и профессиональных компетенций, на формирование которых направлена дисциплина.

Выполнение практических работ в рамках МДК.05.01 «Технологии виртуализации и автоматизации» позволяет освоить комплекс работ по выполнению практических заданий по всем темам МДК.05.01 «Технологии виртуализации и автоматизации».

Рабочая тетрадь по МДК.05.01 «Технологии виртуализации и автоматизации» имеют практическую направленность и значимость. Формируемые в процессе практических занятий умения могут быть использованы студентами в будущей профессиональной деятельности.

Рабочая тетрадь предназначена для студентов колледжа, изучающих МДК.05.01 «Технологии виртуализации и автоматизации».

Оценки за выполнение практических работ выставляются по пятибалльной системе. Оценки за практические работы являются обязательными текущими оценками и выставляются в журнале теоретического обучения.

1. Перечень практических работ по МДК.05.01 «Технологии виртуализации и автоматизации»

№ раздела, темы	Освоение умений в процессе занятия	Формируемые ОК и ПК	Тема практического занятия	Кол-во часов
Тема 1.1. Платформы виртуализации на основе кластерного подхода	– Определять общие модели развертывания облачной инфраструктуры; – Поддерживать облачные конфигурации в актуальном состоянии и вести учет контроля версий; – Определять, насколько данные модели соответствуют требованиям, специфичным для организации; – Пользоваться преимуществами облачной инфраструктуры для снижения операционных нагрузок при развертывании служб; – Документировать ключевые требования бизнес-приложений и то, как они соотносятся миграцией в облачную инфраструктуру; – Переводить бизнес-цели и задачи в спецификации, а также презентовать их заинтересованным сторонам; – Проводить оценку, выбор и внедрение передовых облачных сервисов, таких как сервисы управления данными, сервисы кэширования и сервисы автоматического масштабирования и обеспечения доступности; – Создавать внутренние руководящие документы и требования к процедурам, необходимым	ПК 5.1-ПК 5.3 ОК 1-9	Практическое занятие 1. Работа с Hypervisor: Установка и настройка hosted	2
			Практическое занятие 2. Работа с Hypervisor: Установка и настройка нативного Hypervisor.	2
			Практическое занятие 3. Работа с Hypervisor: Установка и настройка виртуальных машин.	2
			Практическое занятие 4. Работа с Hypervisor: Настройка виртуальной маршрутизации	2
			Практическое занятие 5. Работа с Hypervisor: Автоматизциия развёртывания виртуальных машин	2
			Практическое занятие 6. Работа с Hypervisor: Конфигураця ресурсов виртуальных машин	2
			Практическое занятие 7. Работа с Hypervisor: Развёртывание сервисов для конечного пользователя (Базы данных, HostePanel, Серверов сертификации и аутентификации)	2
			Практическое занятие 8. Установка Kubernetes в среде Proxmox VE	2
			Практическое занятие 9. Настройка Kubernetes в среде Proxmox VE	2
			Практическое занятие 10. Работа с контейнерами Kubernetes в среде Proxmox VE	2
			Практическое занятие 11. Оркестрациия Kubernetes в среде Proxmox VE	2
			Практическое занятие 12. Настройка логирования контейнеров.	2

№ раздела, темы	Освоение умений в процессе занятия	Формируемые ОК и ПК	Тема практического занятия	Кол-во часов
	для создания, обновления, удаления и получения доступа к инфраструктуре и ресурсам общедоступного облака; – Проводить оценку, выбирать и внедрять базовые облачные сервисы, таких как вычислительная среда, сеть и хранилище; – Разрабатывать и внедрять процессы проверки подлинности на уровне подразделения и компании в целом, контролировать доступ к системе управления общедоступным облаком; – Анализировать и интерпретировать показатели производительности вычислений, хранения данных, уровня сети и приложений для использования в дизайне общедоступной облачной инфраструктуре; – Использовать методы и пакеты настройки производительности для обеспечения оптимального использования ресурсов;		Практическое занятие 13. Настройка виртуальных машин для шлюза удалённого рабочего стола	2
			Практическое занятие 14. Настройка межплатформенный бесклиентский шлюз удаленного рабочего стола	2
			Практическое занятие 15. Работа с Облачными бизнес-моделями IaaS: Установка.	2
			Практическое занятие 16. Работа с Облачными бизнес-моделями IaaS: Автоматизация. развёртывание виртуальной машины.	2
			Практическое занятие 17. Работа с Облачными бизнес-моделями IaaS: Балансировщик нагрузки виртуальных машин.	2

2. Описание порядка выполнения практических работ

Практическая работа №1 Работа с Hypervisor: Установка и настройка hosted

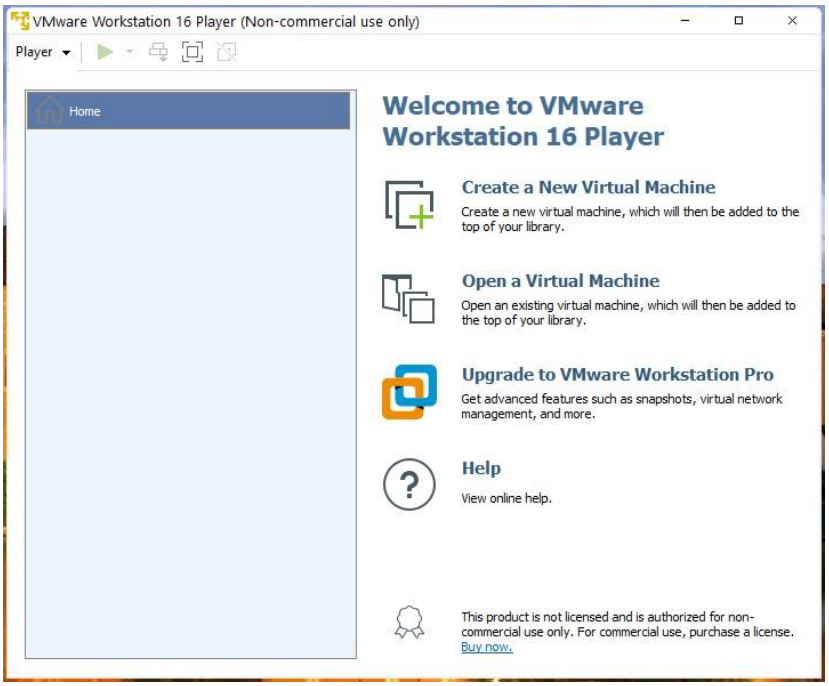
Задание:

1. Скачайте официальную версию Proxmox https://www.proxmox.com/en/proxmox-virtual-environment/overview
2. Смонтируйте образ, получив сменный диск.
3. Решите, на какой диск будете устанавливать Proxmox.
4. Если нужно изменить параметры, это можно сделать в разделе Option. Если в этом нет необходимости, оставьте настройки по умолчанию.
5. Поставьте региональные настройки, нужно выбрать Россию.
6. Задайте сетевые настройки: IP-адрес, маску подсети и другие.
7. Перезагрузите устройство.
8. Запустите веб-интерфейс с помощью команды в таком формате, указав актуальные значения.

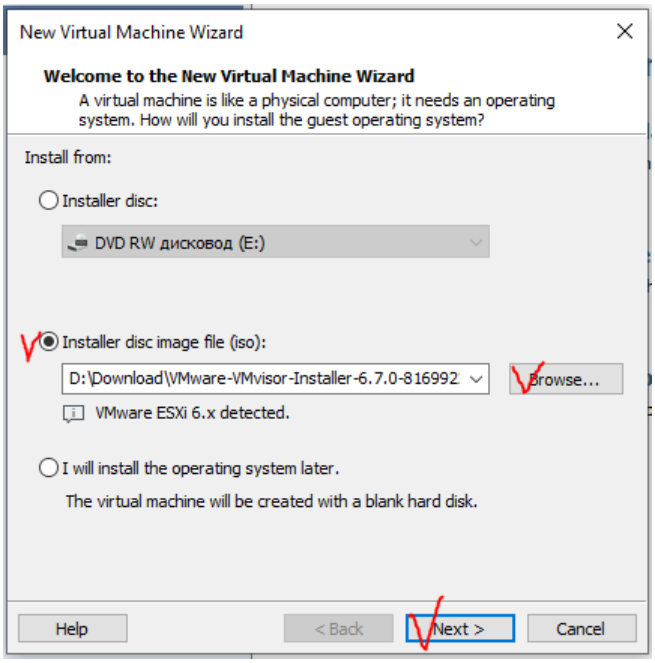
Практическая работа № 2 Работа с Hypervisor: Установка и настройка нативного Hypervisor.

Задание:

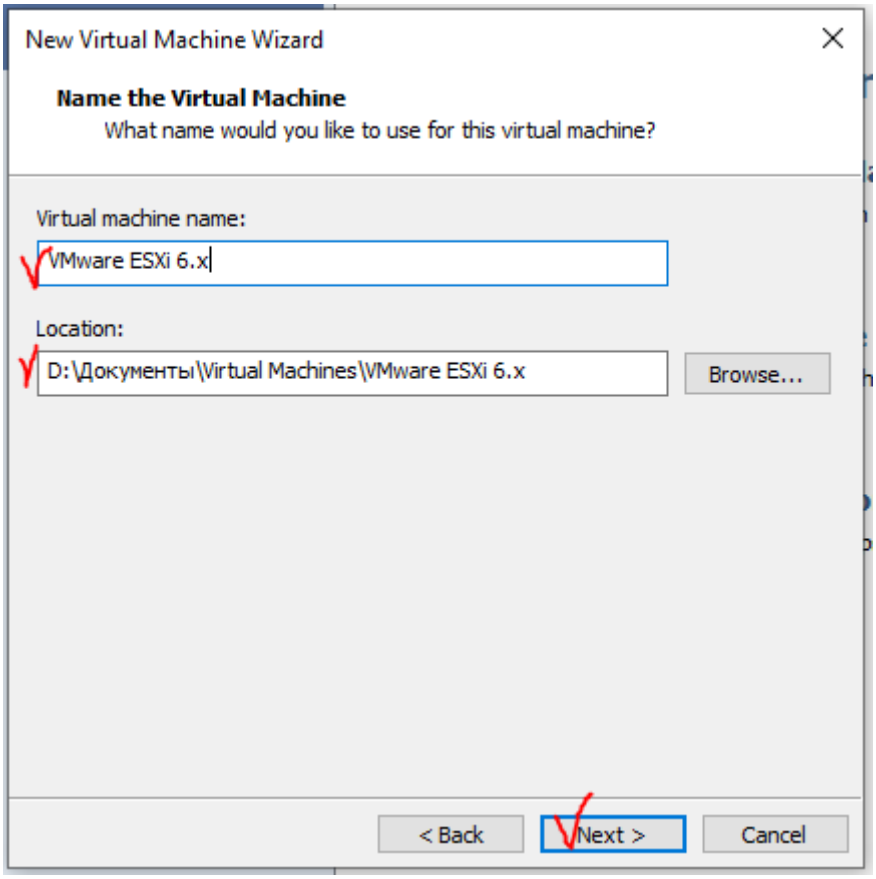
1. Установить гипервизор VMware Player 16. Скачать инсталляции можно по ссылке https://yadi.sk/d/J70iVaTwiPJJ1w



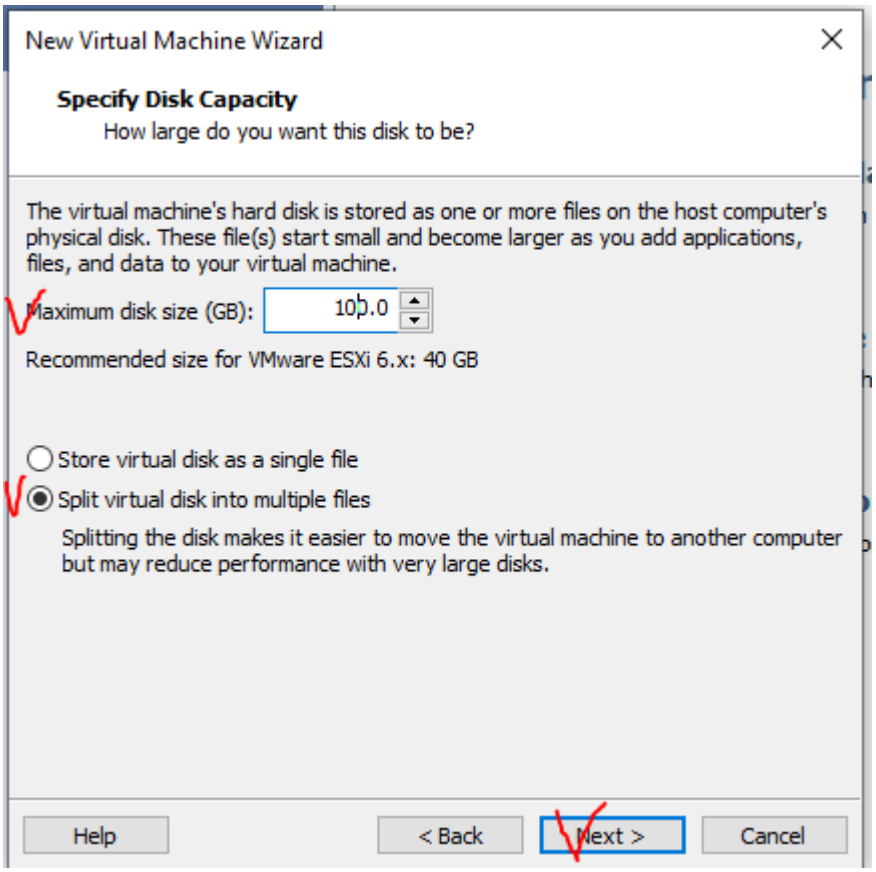
2. Установить систему виртуализации VMware ESXi 6.7. Скачать инсталляции можно по ссылке https://yadi.sk/d/1CqbT5_7m8g0pQ
    a. Выбрать пункт *Create a New Virtual Machine*
    b. Выбрать второй пункт *Installer disc image file (iso)*



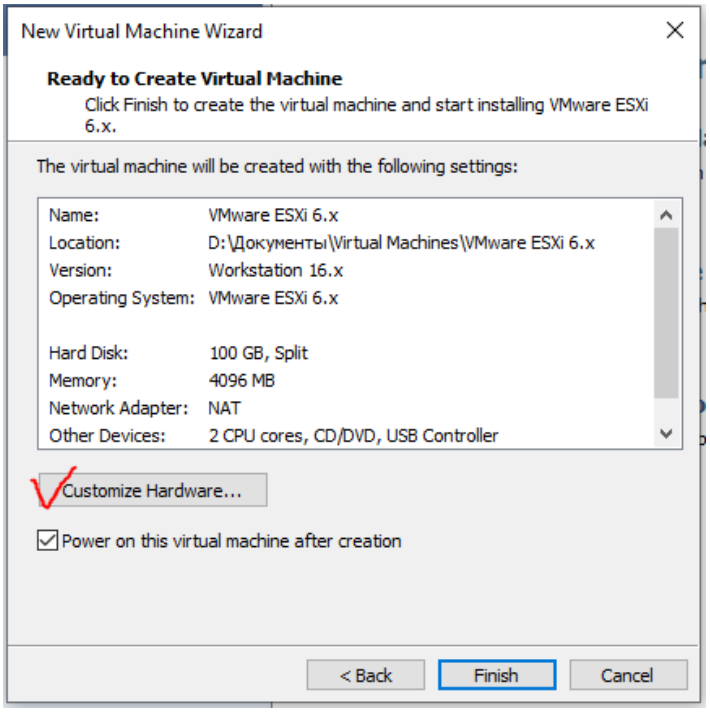
c. Указать имя и место хранения файлов виртуальной машины или оставить по умолчанию.



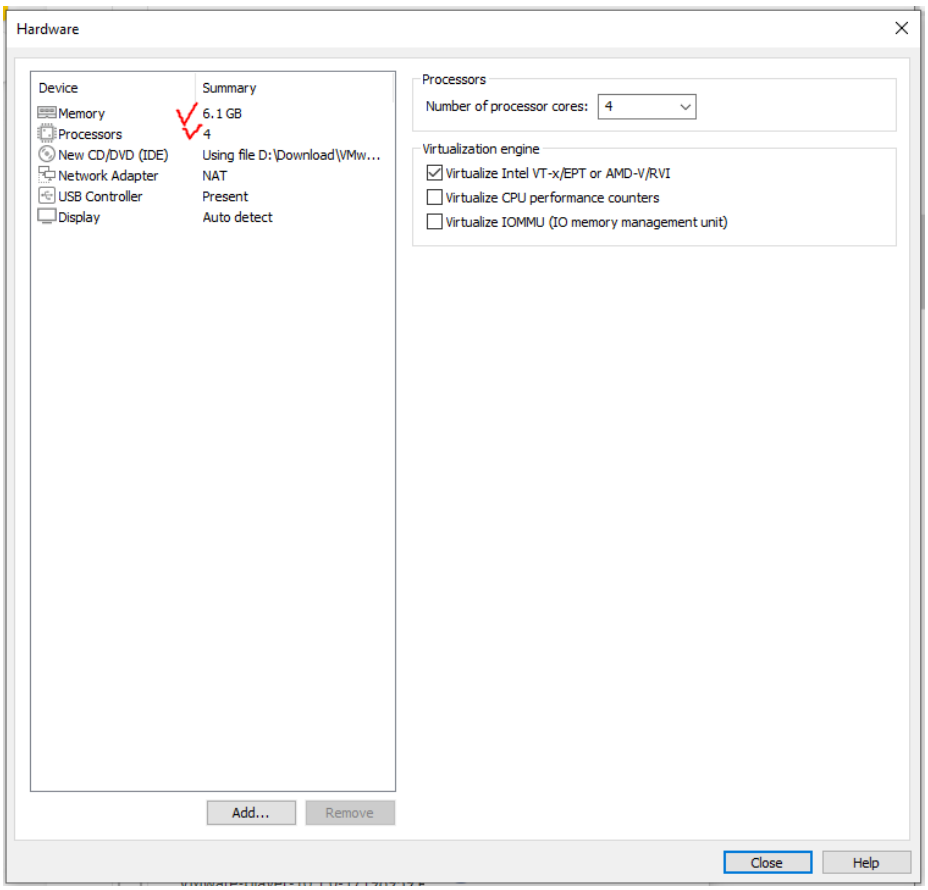
d. Указать размер диска для виртуальной машины. Здесь необходимо указать максимально возможный размер, который позволяет ваша система.



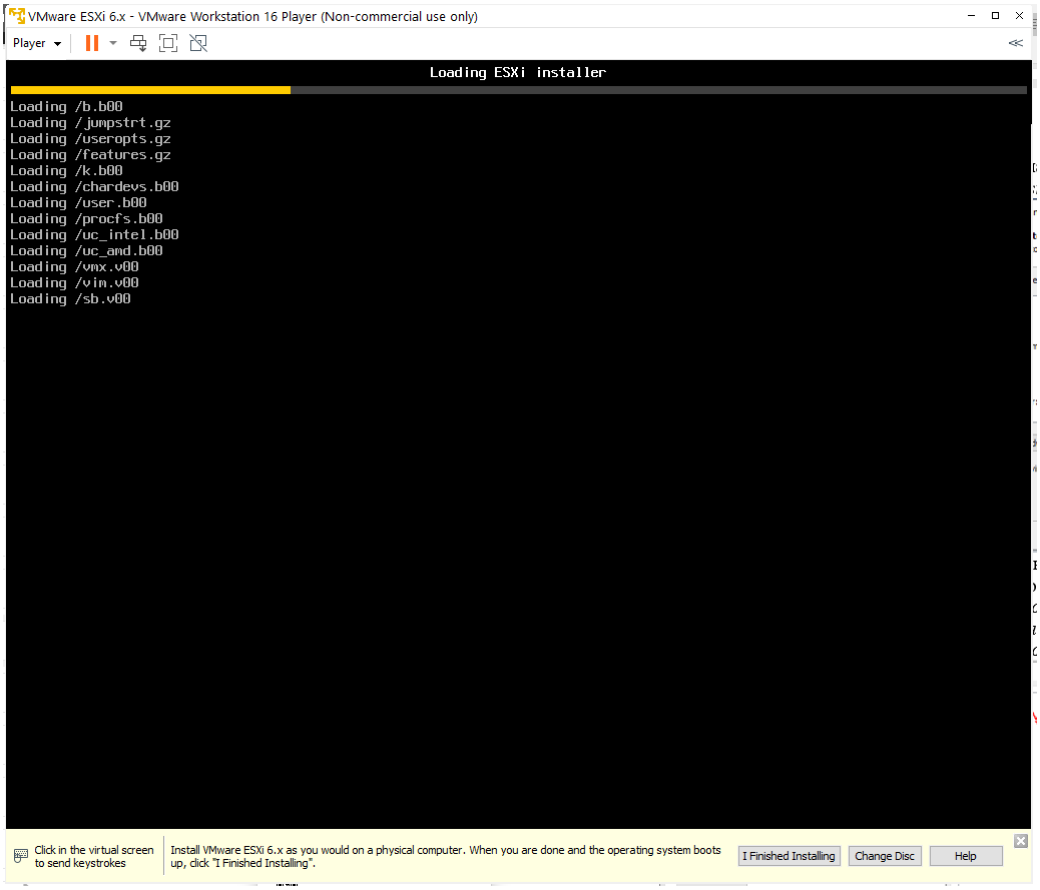
e. Провести настройку параметров виртуальной машины, нажав кнопку *Customize Hardware*

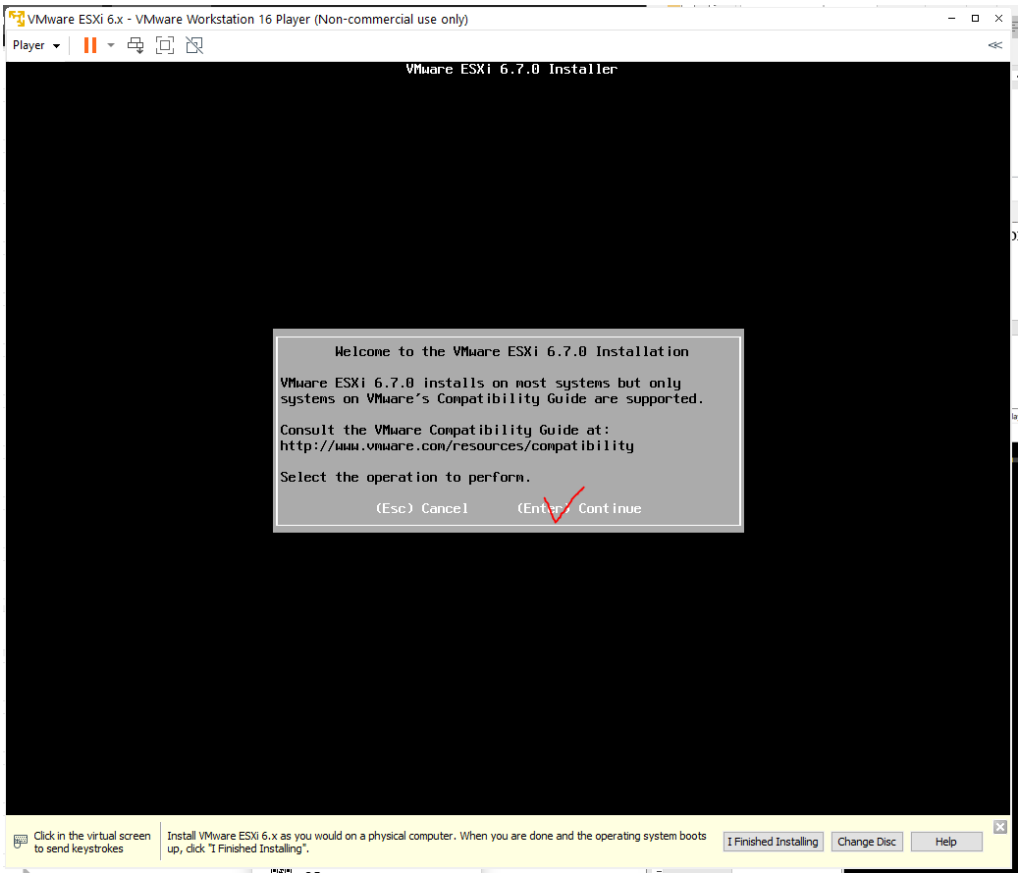


Указать максимально возможное количество ОЗУ и количество процессоров, которое позволяет выбрать ваша система. *Количество ОЗУ для виртуальной машины должно быть меньше имеющейся физической памяти в системе иначе произойдет падение производительности системы в целом*.

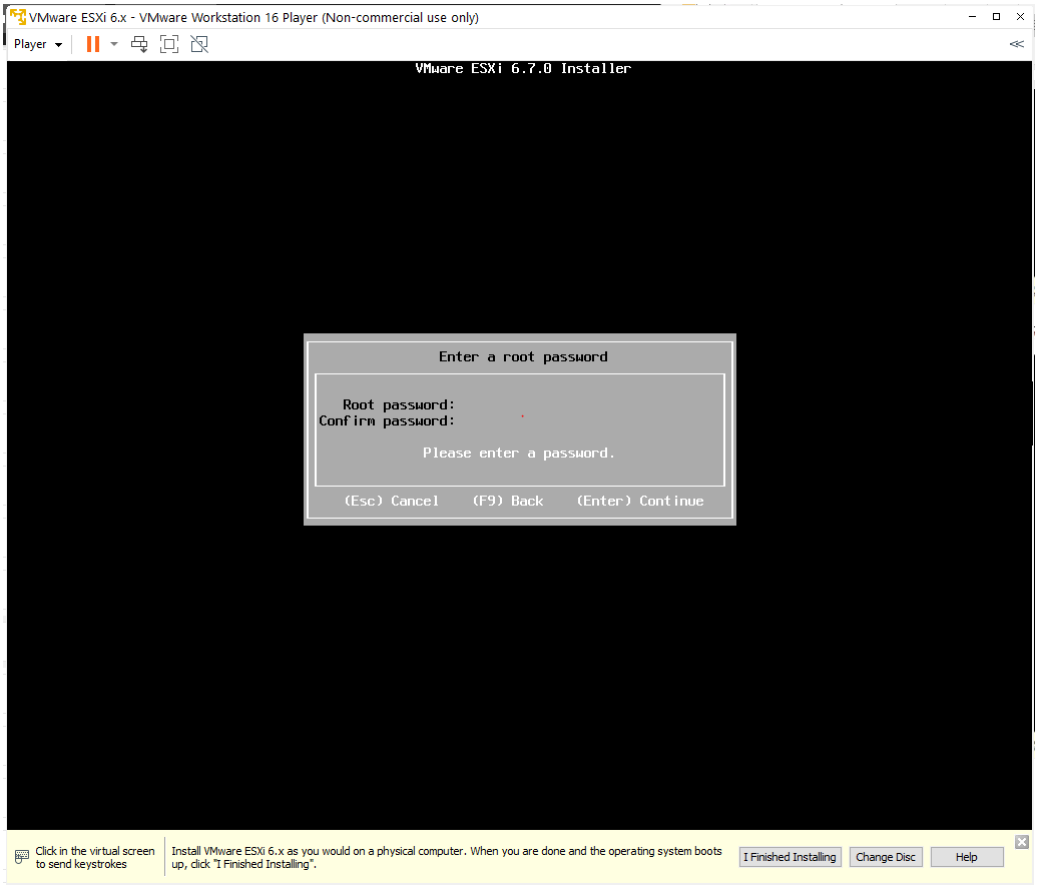


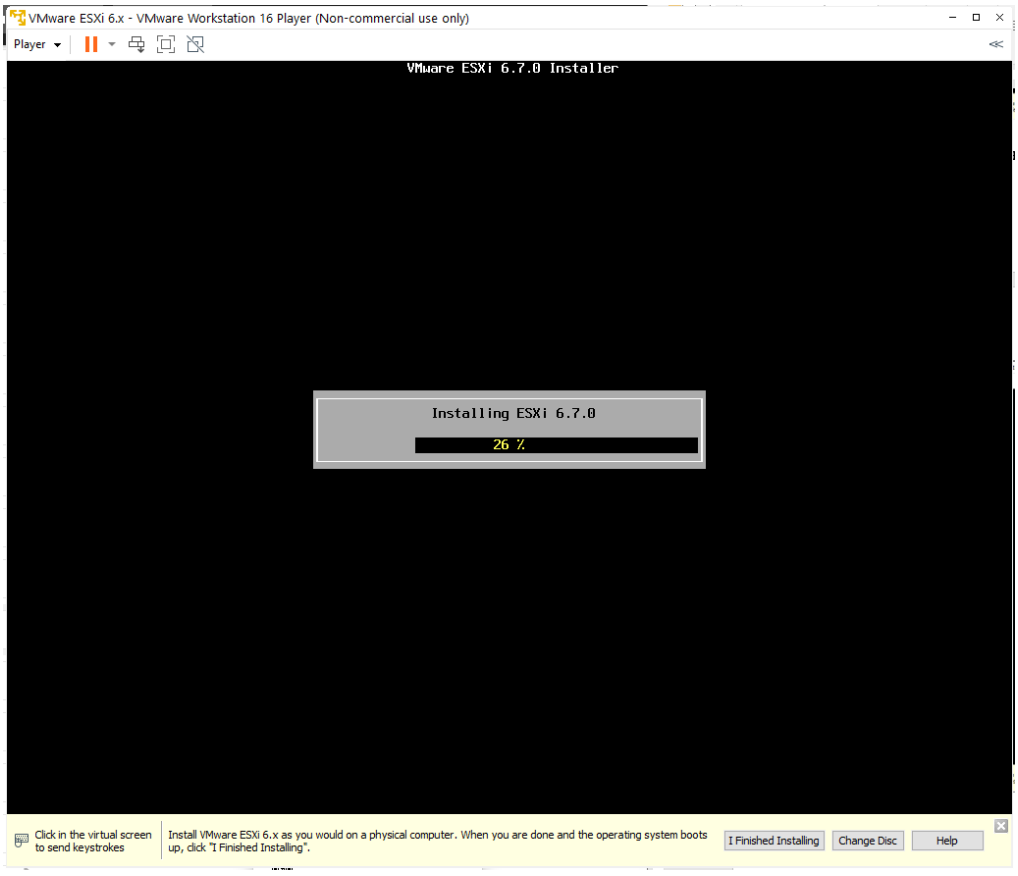
f. Нажать кнопку *Finish*. Запустится процесс установки



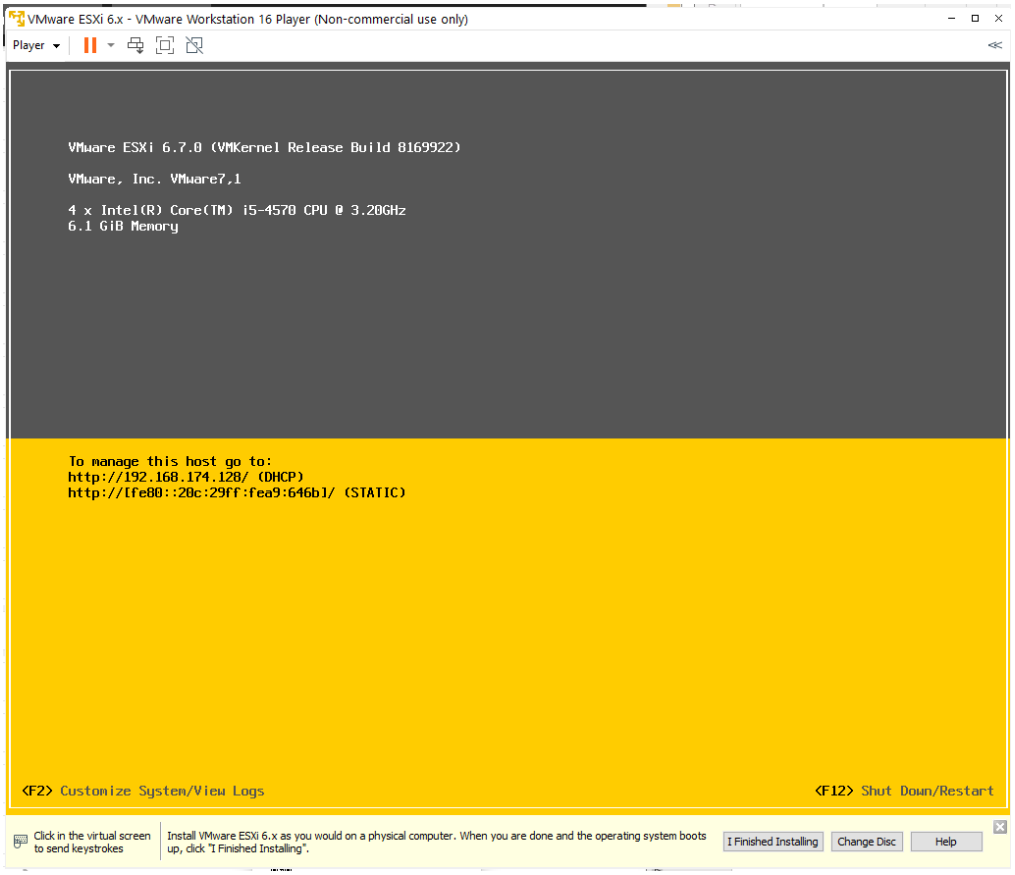


Указать пароль администратора

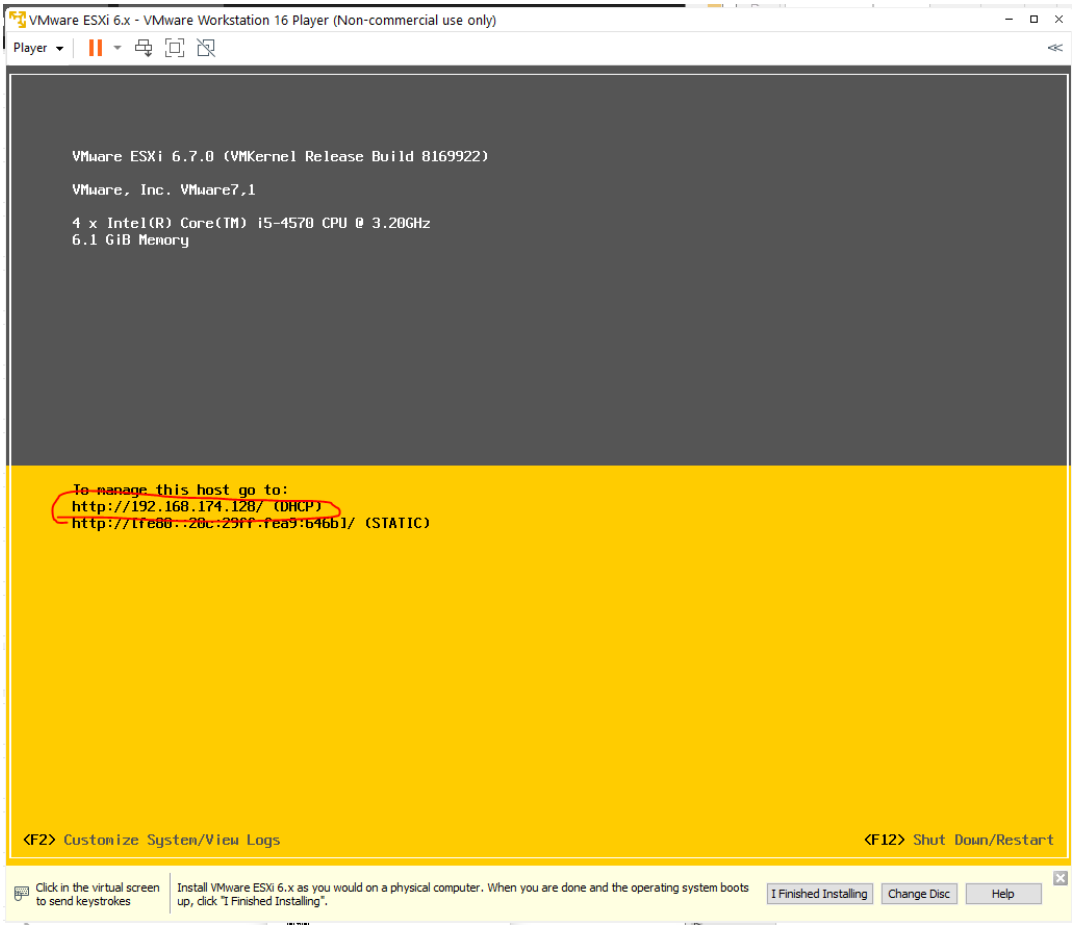




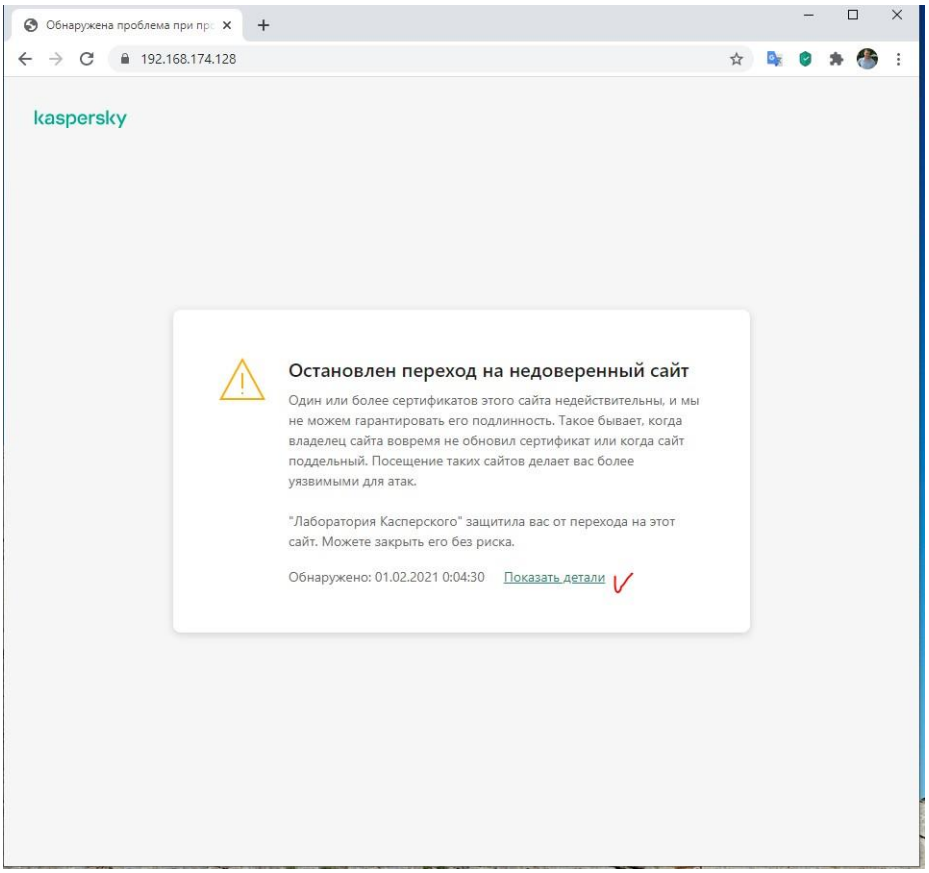
После перезагрузки система готова к работе

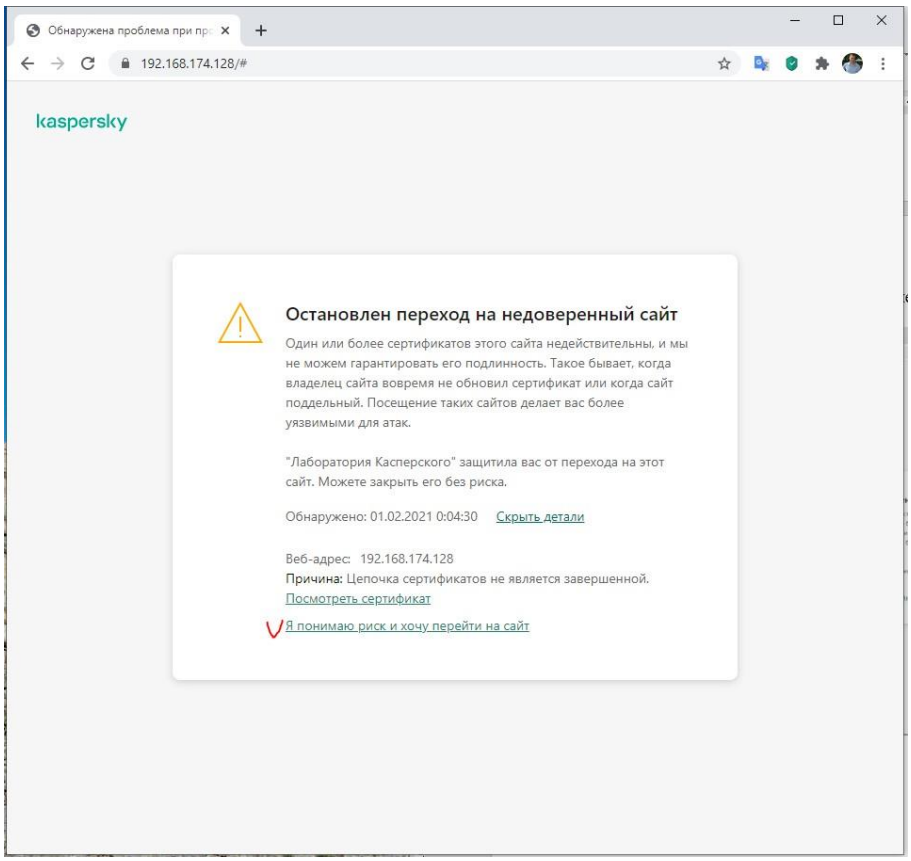


3. Доступ к системе осуществляется с помощью Web-браузера. В адресной сроке необходимо указать IP адрес, указанный на экране гипервизора

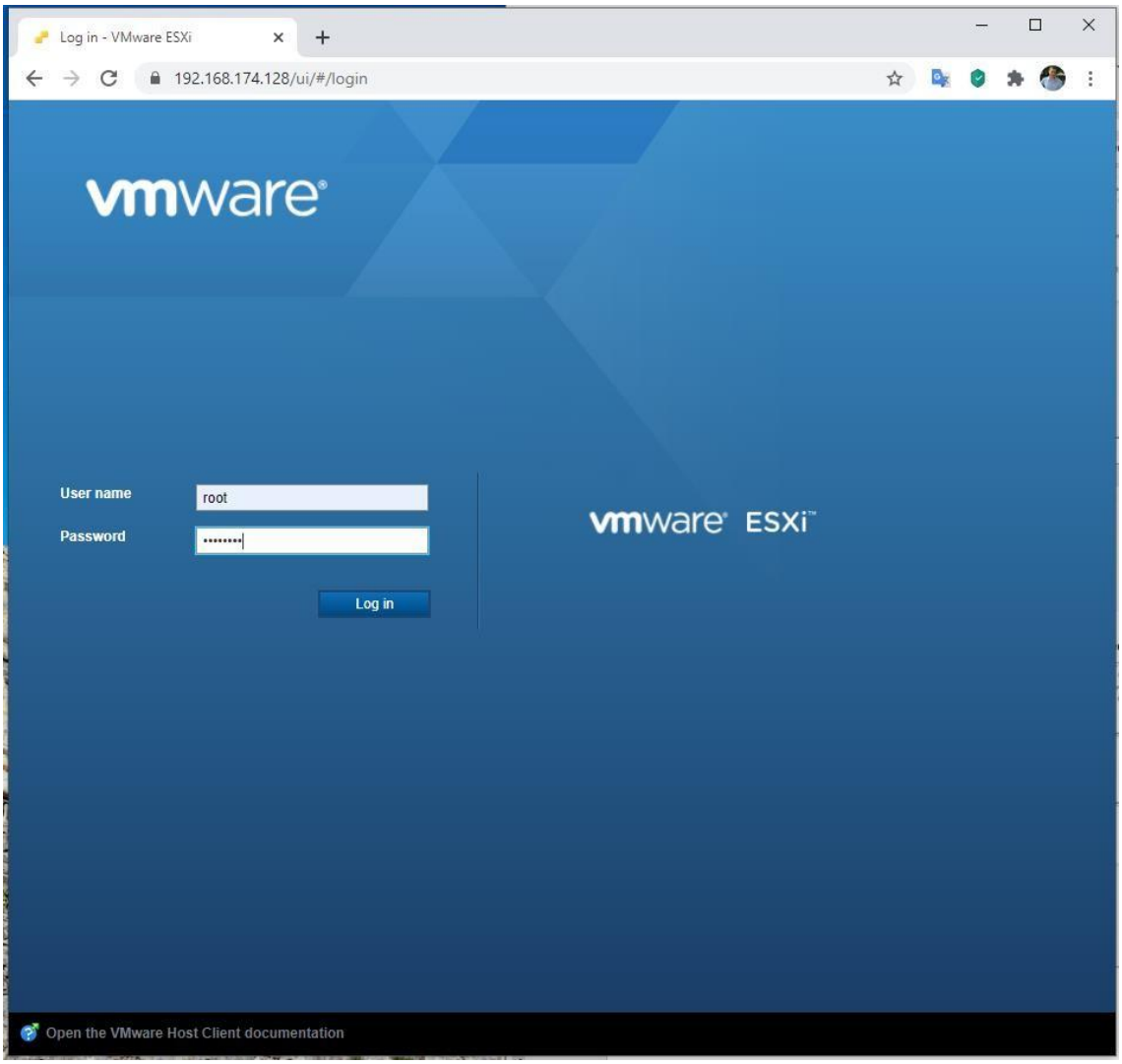


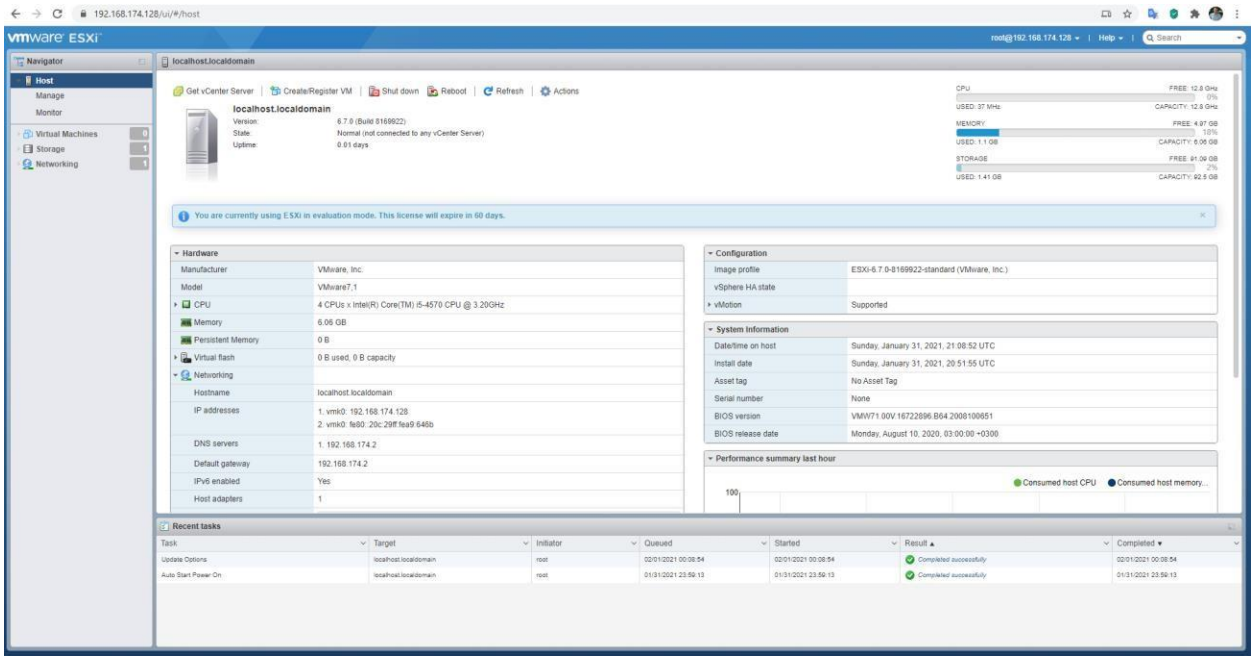
4. Возможно сообщение антивируса о небезопасном использовании этой веб страницы





5. Ввести логин и пароль





Создание сетевой инфраструктуры

Для проведения лабораторных работ будет использована схема сети, представленная на рисунке

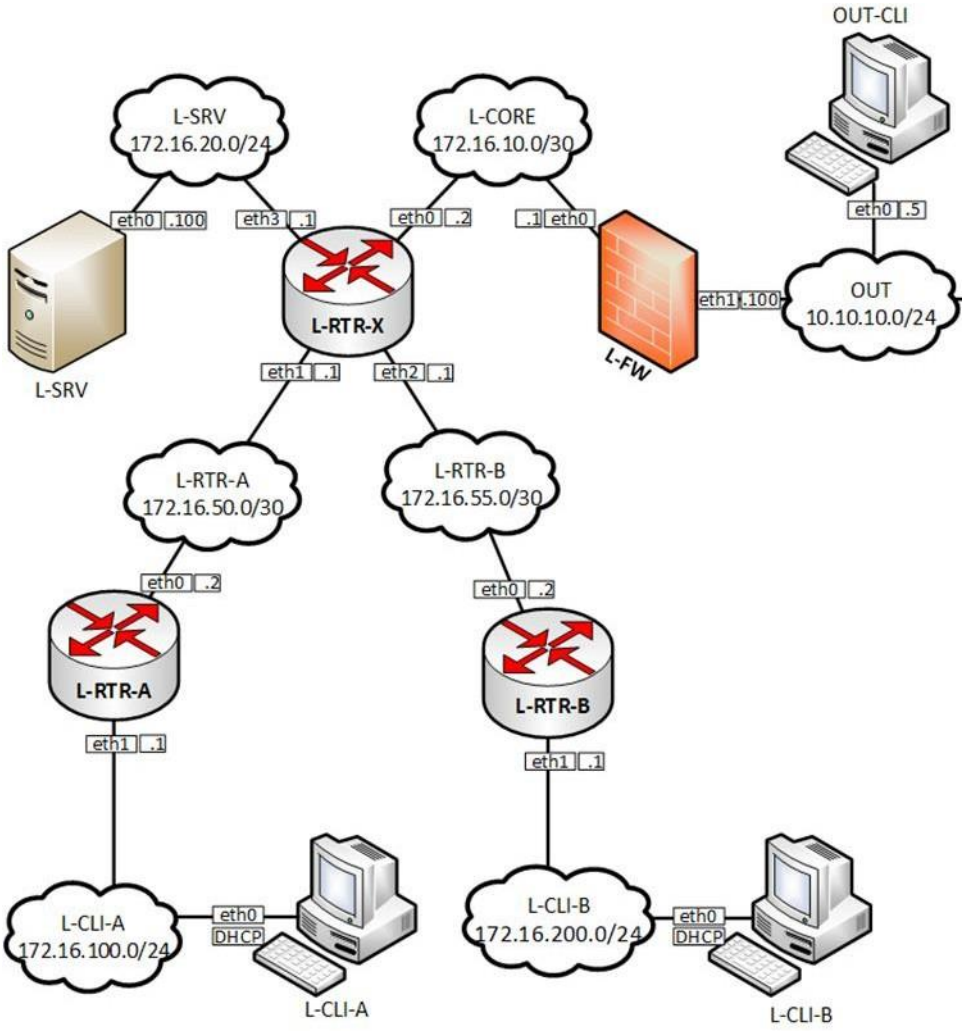
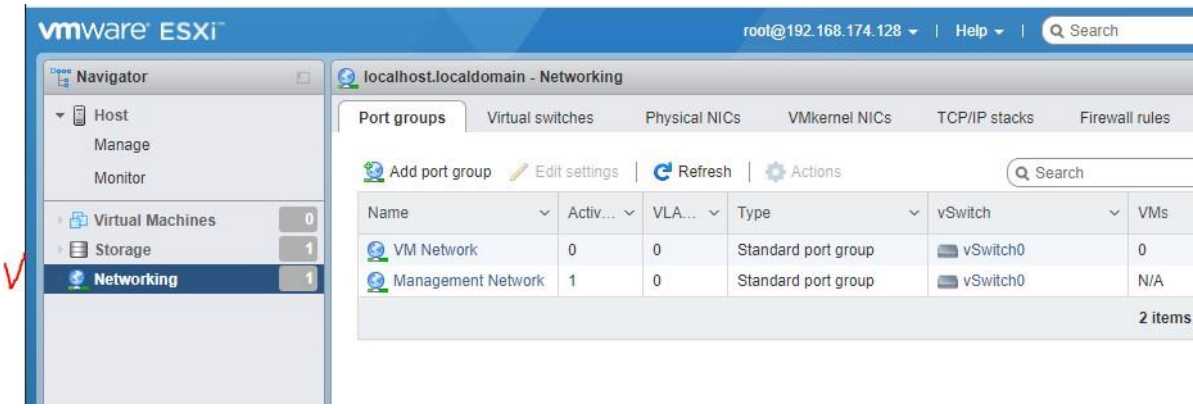


Рисунок 1. Топология сети

Каждое «облако» на схеме представляет собой отдельную подсеть со своим адресным пространством. В системе VMware ESXi реализовать такую топологию возможно с использованием механизма **Виртуальных коммутаторов** и **Групп портов**.

Для создания виртуального коммутатора и группы портов необходимо:

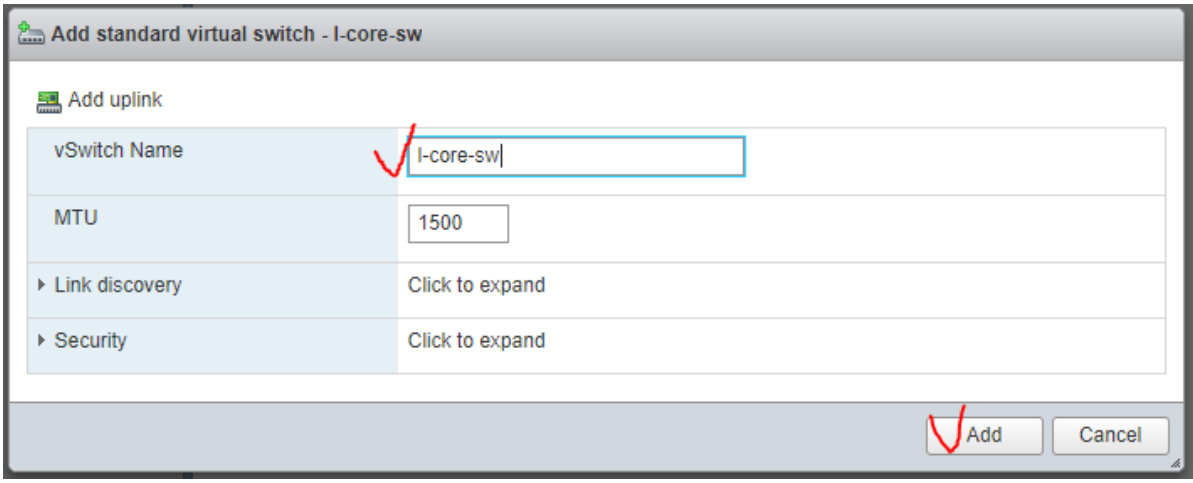
1. Нажать на пункт *Networking* в древовидном списке слева



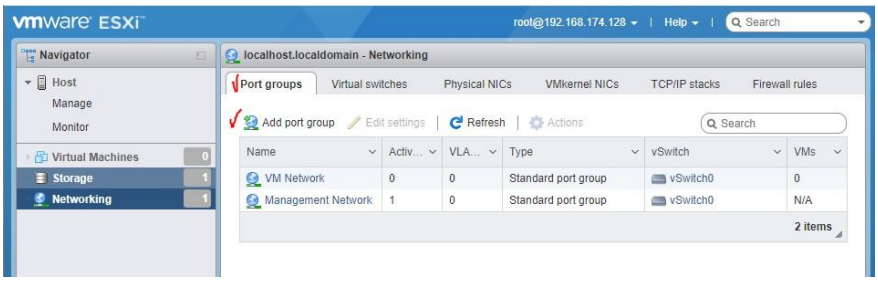
2. Открыть вкладку *Virtual switches* и нажать *Add standard virtual switch*

3. Указать имя нового коммутатора, например, *l-core-sw*, и нажать кнопку

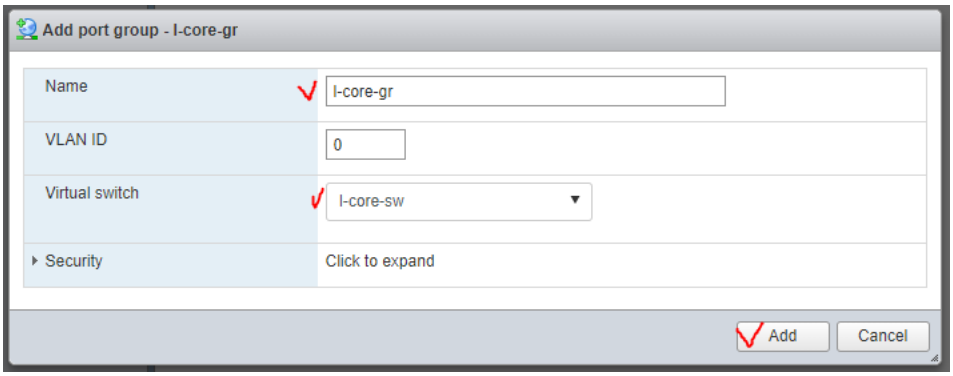
   Add



4. Открыть вкладку *Port groups* и нажать *Add port group*



5. Указать имя новой группы портов, например, L-core-gr, и выбрать из списка виртуальных коммутаторов нужный, в нашем случае – это L- core-sw, и нажать кнопку *Add*



Таким образом, с помощью указанного механизма реализуется следующее правило: группа портов подключается только к одному виртуальному коммутатору, а один виртуальный коммутатор может содержать несколько групп портов.

Подключение виртуальных машин к виртуальным коммутаторам, т.е. **создание подсетей**, осуществляется с помощью **присоединения сетевого адаптера виртуальной машины к нужной группе портов**.

Практическая работа № 3 Работа с Hypervisor: Установка и настройка виртуальных машин.

Задание:

1. Установите в любом гипервизоре следующие виртуальные машины:

   Windows 10 RPO

   Astra Linux SE

   Ubuntu 22.04

   Windows Server 2019

2. Зафиксируйте процесс установки скриншотами.

Практическая работа № 4 Работа с Hypervisor: Настройка виртуальной маршрутизации

Задание:

1. Настройка виртуальной сети в гипервизоре ESXI 6.0 | 🖧 Полигон 218 (poligon218.ru)

Практическая работа № 5 Работа с Hypervisor: Автоматизциия развёртывания виртуальных машин

Задание:

1. Создаем проект тестовой лаборатории и формируем нижеуказанную структуру папок и файлов, где:

●**LAB** — название корневой папки проекта.
●**Vagrantfile** — файл (подробнее дальше) с указаниями по разворачиванию виртуальных машин.
●**provision** — папка, где будут лежать сценарии DSC (будут использоваться для конфигурации виртуальных машин).
●**Файлы с расширением .ps1** — непосредственно сценарии DSC для конфигурирования ОС.
Ниже определим назначение каждого сценария .ps1
●**ConfigureRebootOnNode.ps1** — конфигурирует перезапуск хоста, если это необходимо.
●**DSCInstallDHCP.ps1** — устанавливает роль DHCP, RSAT-DHCP, авторизует DHCP сервер в AD после установки и конфигурирует область.
●**DSCInstallDomain.ps1** — устанавливает роль AD DS, RSAT-ADDS, конфигурирует домен PARTY.HARD.
●**DSCJoinDomain.ps1** — добавляет хост в домен Active Directory.
●**PreDSCInstall.ps1** — устанавливает необходимые для работы модули DSC.
Ниже прикрепляю визуализацию структуры.

```
LAB
│   Vagrantfile
│
└───provision
        ConfigureRebootOnNode.ps1
        DSCInstallDHCP.ps1
        DSCInstallDomain.ps1
        DSCJoinDomain.ps1
        PreDSCInstall.ps1
```

Копируем файл Vagrantfile в папку с проектом.
Vagrantfile

```
# -*- mode: ruby -*-
# # vi: set ft=ruby :
Vagrant.configure("2") do |config|
config.vagrant.plugins = "vagrant-reload"
config.vm.define "lab-dc1" do |subconfig|
subconfig.vm.box = "gusztavvargadr/windows-server"
subconfig.vm.hostname = "lab-dc1"
```

```
subconfig.vm.provider :virtualbox do |vb|
vb.gui = false
end
subconfig.vm.network "private_network", ip: "192.168.11.2",
virtualbox__intnet: true
subconfig.winrm.username = "vagrant"
subconfig.winrm.password = "vagrant"
subconfig.winrm.transport = :plaintext
subconfig.winrm.basic_auth_only = true
#Install DSC Modules
subconfig.vm.provision "shell",
path: "provision\\PreDSCInstall.ps1"
#https://github.com/dsccommunity/ActiveDirectoryDsc
subconfig.vm.provision "shell",
path: "provision\\DSCInstallDomain.ps1"
#Restart VM to finish Active Directory Domain Services installation
subconfig.vm.provision :reload
#https://github.com/dsccommunity/xDhcpServer
subconfig.vm.provision "shell",
path: "provision\\DSCInstallDHCP.ps1"
end
config.vm.define "lab-test" do |subconfig|
subconfig.vm.box = "gusztavvargadr/windows-server"
subconfig.vm.hostname = "lab-test"
subconfig.vm.provider :virtualbox do |vb|
vb.gui = false
end
subconfig.vm.network "private_network", ip: "127.0.0.2",
auto_config: false,
virtualbox__intnet: true
subconfig.winrm.username = "vagrant"
subconfig.winrm.password = "vagrant"
subconfig.winrm.transport = :plaintext
subconfig.winrm.basic_auth_only = true
subconfig.vm.provision "shell",
path: "provision\\ConfigureRebootOnNode.ps1"
subconfig.vm.provision "shell",
path: "provision\\PreDSCInstall.ps1"
subconfig.vm.provision "shell",
path: "provision\\DSCJoinDomain.ps1"
end
end
```

Копируем файл PreDSCInstall.ps1 в папку provision

```
PreDSCInstall.ps1
Install-Module -Name ActiveDirectoryDsc -Force
```

```
Install-Module -Name PSDscResources -Force
Install-Module -Name ComputerManagementDsc -Force
Install-Module -Name xDhcpServer -Force
```

Копируем файл DSCInstallDomain.ps1 в папку provision

DSCInstallDomain.ps1

```
Configuration ADDomain_NewForest_Config
{
param
(
[Parameter(Mandatory = $true)]
[ValidateNotNullOrEmpty()]
[System.Management.Automation.PSCredential]
$Credential,
[Parameter(Mandatory = $true)]
[ValidateNotNullOrEmpty()]
[System.Management.Automation.PSCredential]
$SafeModePassword
)
Import-DscResource -ModuleName PSDscResources
Import-DscResource -ModuleName ActiveDirectoryDsc
Import-DscResource -ModuleName ComputerManagementDsc
Node 'localhost'
{
WindowsFeature 'Active Directory Domain Services'
{
Name = 'AD-Domain-Services'
Ensure = 'Present'
}
WindowsFeature 'RSAT-ADDS'
{
Name = 'RSAT-ADDS'
Ensure = 'Present'
}
ADDomain 'party.hard'
{
DomainName = 'party.hard'
Credential = $Credential
SafemodeAdministratorPassword = $SafeModePassword
ForestMode = 'WinThreshold'
}
}
}
#Next block is using to allow password as plain text
$cd = @{
AllNodes = @(
```

```
@{
NodeName = 'localhost'
PSDscAllowPlainTextPassword = $true
}
)
}
#Define user and password for ADDomain deployment (also used for restore).
$password = ConvertTo-SecureString "RestorePassword123$" -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential('vagrant',$password)
#Create MOF
ADDomain_NewForest_Config -Credential $cred -SafeModePassword $cred -ConfigurationData
$cd
#Execute MOF
Start-DscConfiguration -Path .\ADDomain_NewForest_Config -Force -Wait -Verbose
```

Копируем файл DSCInstallDHCP.ps1 в папку provision

```
Configuration xDhcpsServerScope_NewScope
{
Import-DscResource -ModuleName xDHCpServer
#Define NIC IP
$IP = Get-NetIPAddress -InterfaceAlias "Ethernet 2" | Where-Object {$_.IPAddress -notlike "*:*"
} | select -ExpandProperty IPAddress
Node 'localhost'
{
WindowsFeature 'RSAT-DHCP'
{
Name = 'RSAT-DHCP'
Ensure = 'Present'
}
WindowsFeature 'DHCP'
{
Name = 'DHCP'
Ensure = 'Present'
}
xDhcpServerAuthorization RemoteServerActivation
{
Ensure = 'Present'
DnsName = $env:COMPUTERNAME + '.party.hard'
IPAddress = $IP
}
xDhcpServerScope Scope
{
ScopeId = '192.168.11.0'
Ensure = 'Present'
IPEndRange = '192.168.11.254'
IPStartRange = '192.168.11.10'
```

```
Name = '11.0/24'
SubnetMask = '255.255.255.0'
LeaseDuration = ((New-TimeSpan -Hours 8 ).ToString())
State = 'Active'
AddressFamily = 'IPv4'
}
xDhcpServerOption Option
{
Ensure = 'Present'
ScopeID = '192.168.11.0'
DnsDomain = 'party.hard'
DnsServerIPAddress = '192.168.11.2'
AddressFamily = 'IPv4'
Router = '192.168.11.1'
}
}
}
xDhcpsServerScope_NewScope
Start-DscConfiguration -Path .\xDhcpsServerScope_NewScope -Force -Wait -Verbose
```

Копируем файл DSCJoinDomain.ps1 в папку provision

```
DSCJoinDomain.ps1
Configuration JoinDomainConfiguration
{
param
(
[Parameter(Mandatory = $true)]
[ValidateNotNullorEmpty()]
[System.Management.Automation.PSCredential]
$Credential
)
Import-DscResource -Module ComputerManagementDsc
Node 'localhost'
{
Computer JoinDomain
{
Name = $env:COMPUTERNAME
DomainName = 'PARTY'
Credential = $Credential # Credential to join to domain
}
}
}
#Next block is using to allow password as plain text
$cd = @{
AllNodes = @(
@{
```

```
NodeName = 'localhost'
PSDscAllowPlainTextPassword = $true
}
)
}
#Define user and password for ADDomain deployment (also used for restore).
$password = ConvertTo-SecureString "vagrant" -AsPlainText -Force
$cred = New-Object Sys-
tem.Management.Automation.PSCredential('party.hard\vagrant',$password)
#Create MOF
JoinDomainConfiguration -Credential $cred -ConfigurationData $cd
#Execute MOF
Start-DscConfiguration -Path .\JoinDomainConfiguration -Force -Wait -Verbose
```

Копируем файл ConfigureRebootOnNode.ps1 в папку provision

```
Configuration ConfigureRebootOnNode
{
Node 'localhost'
{
LocalConfigurationManager
{
RebootNodeIfNeeded = $true
}
}
}
#Create MOF
ConfigureRebootOnNode
Set-DscLocalConfigurationManager .\ConfigureRebootOnNode -Verbose
Get-DscLocalConfigurationManager
```

Структура проекта сформирована, что дальше?
Мы с вами выполнили, пожалуй, самую сложную часть, поздравляю. Теперь, чтобы автоматически развернуть нашу небольшую тестовую лабораторию достаточно выполнить команду:

```
vagrant up
```

Все. Идем пить чай. В зависимости от производительности вашей машины и скорости интернет, возможно, придется выпить не одну кружку 
По возвращению Вы обнаружите маленькое виртуальное окружение, которое состоит из двух узлов. Мы можем убедиться в этом выполнив команду:

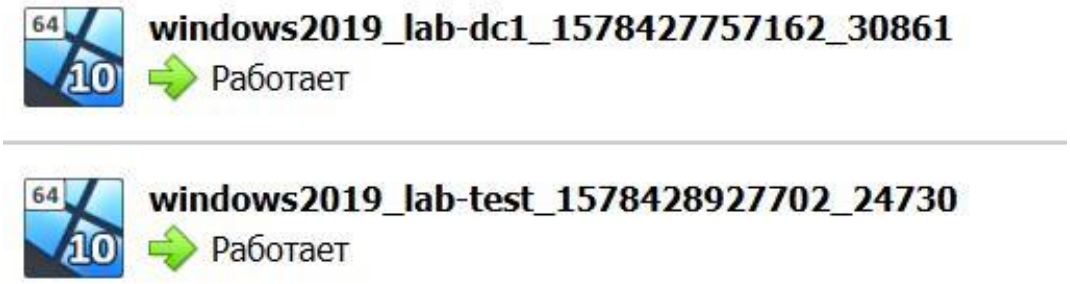
```
vagrant status
```

Мы увидим список запущенных виртуальных машин

```
Current machine states:
lab-dc1 running (virtualbox)
lab-test running (virtualbox)
```

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
VM, run `vagrant status NAME`.

Либо, открыв менеджер управления виртуальными машинами VirtualBox. Мы увидим те же ВМ и тот же статус.



2.

Практическая работа № 6 Работа с Hypervisor: Конфигураця ресурсов виртуальных машин

Задание:

1. Измените ресурсы созданных ВМ:

   Добавьте 8ГБ ОЗУ

   Увеличьте диск на 100ГБ

   Создайте общую папку для всех машин и подключите ее.

   Добавьте второй сетевой интерфейс NAT

Практическая работа № 7 Работа с Hypervisor: Развёртывание сервисов для конечного пользователя (Базы данных, HostePanel, Серверов сертификации и аутентификации)

Задание:

1. Разверните сервисы конечного пользователя:

   – Базы данных,
   – HostePanel,
   – Серверов сертификации и аутентификации

Практическая работа № 8 Установка Kubernetes в среде Proxmox VE

Задание:

Так как по умолчанию контейнерам не разрешено самостоятельно загружать модули ядра, вы должны настроить их загрузку непосредственно на гипервизорах.

Мы будем использовать overlay драйвер для docker, так что это все что нам нужно:

```
echo overlay >> /etc/modules
```

Теперь нам нужно добавить больше привилегий для нашего контейнера, что бы разрешить ему запускать другие контейнеры внутри, добавьте эти строки в конфиг вашего контейнера:

```
lxc.apparmor.profile: unconfined
lxc.cap.drop:
lxc.cgroup.devices.allow: a
lxc.mount.auto: proc:rw sys:rw
```

Начиная с версии **v11.0** kubelet требует shared mode для всех mounts с хоста.

Этот грязный хак позволит вам достичь этого, внутри LXC-контейнера запустите:

```
echo '#!/bin/sh -e
mount --make-rshared /' > /etc/rc.local
```

Это действие добавит команду mount --make-rshared / в /etc/rc.local и будет запускать ее каждый раз при загрузке контейнера.

Также если вы планируете использовать HA-manager в proxmox, знайте что на данный момент есть неприятный bug#1842, который принудительно убивает процессы контейнера во

время миграции, что может породить зомби-процессы или даже заблокировать ваше хранилище.

Это не есть хорошо, к счастью есть простое решение:

```
sed -i 's/forceStop => 1/forceStop => 0/' /usr/share/perl5/PVE/HA/Resources/PVECT.pm
```

В дополнение можно добавить следующие опции для вашего docker:

```
--storage-driver overlay2
--iptables=false
--ip-masq=false
```

Скопируйте docker.service из /lib в /etc для переопределения его параметров:

```
cp /{lib,etc}/systemd/system/docker.service
```

Теперь добавьте эти опции в ExecStart секцию.

Практическая работа №9 Настройка Kubernetes в среде Proxmox VE

Шаг 1: Подготовьте хост proxmox

Убедитесь, что загружены следующие модули:

```
# cat /proc/sys/net/bridge/bridge-nf-call-iptables
```

Теперь убедитесь, что возможность подкачки равна 0, чтобы swap не использовался, иначе kubernetes не запустится:

cat /proc/sys/vm/swappiness

[should be 0]

Определите новый

sysctl vm.swappiness=0

Отключите SWAP, для очистки области SWAP потребуется некоторое время

#s wapoff -a

Теперь дождитесь, пока swap опустеет.

Шаг 1: Создание контейнера kubernetes

Создайте новый контейнер в proxmox, убедившись, что ему присвоен 0 swap, и сделайте его привилегированным контейнером

Отредактируйте конфигурационный файл /etc/pve/lxc/$ID.conf и добавьте следующую часть:

lxc.apparmor.profile: unconfined

lxc.cgroup.devices.allow: a

lxc.cap.drop:

lxc.mount.auto: "proc:rw sys:rw"

Если вы используете zfs в proxmos, обязательно создайте том ext4, поскольку zfs не поддерживается kubeadm See: https://github.com/corneliusweig/kubernetes-lxd

zfs create -V 50G mypool/my-dockervol

zfs create -V 5G mypool/my-kubeletvol

mkfs.ext4 /dev/zvol/mypool/my-dockervol

mkfs.ext4 /dev/zvol/mypool/my-kubeletvol

Затем обязательно смонтируйте его внутри контейнера:

mp0: /dev/zvol/mypool/my-dockervol,mp=/var/lib/docker,backup=0

mp1: /dev/zvol/mypool/my-kubeletvol,mp=/var/lib/kubelet,backup=0

Затем убедитесь, что conntrack работает в контейнере

$ sudo conntrack -L

Теперь мы можем настроить необходимый нам VPN, смотрите документацию по установке wireguard

$ sudo add-apt-repository ppa:wireguard/wireguard

$ sudo apt-get update

$ sudo apt-get install wireguard

Создайте конфигурацию:

$ cat > /etc/wireguard/wg0.conf

[Interface]

Address = 10.0.0.1/32

ListenPort = 55555

PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

PrivateKey = WNeaIBT40mN/asu9zXrPeSYA+4pFmZA9lUBvHTx+TG8=

MTU = 1500

[Peer]

server2

PublicKey = NSWzZOIUHPqRxOxUmB/A7+Gs6oECYGojREvGs/ZEi2o=

AllowedIPs = 10.0.0.2/32

[Peer]

server3

PublicKey = JhT41so2SiITMe2uqPoNB40kkwxRqyklWiILyhT1uVY=

AllowedIPs = 10.0.0.3/32

И запустите vpn:

$ wg-quick up wg0

$ wg show

Чтобы убедиться, что мы запускаем vpn при загрузке, и исправить некоторые другие мелкие проблемы, создайте следующий файл rc.local:

$ cat > /etc/rc.local

#!/bin/sh -e

Kubeadm 1.15 needs /dev/kmsg to be there, but it's not in lxc, but we can just use /dev/console instead

see: https://github.com/kubernetes-sigs/kind/issues/662

if [! -e /dev/kmsg]; then

    ln -s /dev/console /dev/kmsg

fi

Make sure our VPN is setup so we can connect to the other nodes

wg-quick up wg0

```
# https://medium.com/@kvaps/run-kubernetes-in-lxc-container-f04aa94b6c9c

mount --make-rshared /' > /etc/rc.local


exit 0
```

Установите разрешения и перезагрузите компьютер.

```
$ chmod +x /etc/rc.local

# sudo reboot
```

Теперь убедитесь, что мы загрузились должным образом:

```
$ ls -l /dev/kmsg

[this should exist]

$ wg show

[this should show wireguard is started]
```

Теперь мы можем приступить к установке kubernetes:

```
$ apt-get update && apt-get install -y apt-transport-https curl

$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -

$ cat <<EOF >/etc/apt/sources.list.d/kubernetes.list

deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF

$ apt-get update

$ apt-get install -y kubelet kubeadm kubectl

$ apt-mark hold kubelet kubeadm kubectl
```

Чтобы убедиться, что kubernets подключается с правильным IP и будет использовать VPN для подключения, мы должны сообщить kubelet, чтобы он использовал ip vpn-сервера:

$ echo "KUBELET_EXTRA_ARGS=--node-ip=10.0.0.1" >> /etc/default/kubelet

Теперь мы можем настроить kubeadm:

Обязательно укажите pod-network-cidr и service-cidr, если они перекрываются с вашей обычной локальной сетью, как это и есть в моем случае, поскольку я запускаю 192.168.x.x в своей локальной сети proxmox. Обратите внимание, что я добавляю дополнительный apiserver-cert-extra-sans, поэтому я могу просто подключиться к серверу api с его ip-адреса локальной сети.

Чтобы убедиться, что это сработает, нам нужно игнорировать все предполетные ошибки, но это должно работать просто отлично.

$ kubeadm init --pod-network-cidr=10.250.0.0/16 --service-cidr=172.31.0.0/16 --apiserver-advertise-address 10.0.0.1 --apiserver-cert-extra-sans k8s.mydomain.com --apiserver-cert-extra-sans 192.168.1.13 --apiserver-cert-extra-sans 10.0.0.1 --ignore-preflight-errors=all

Далее скопируйте конфигурацию kubectl:

$ mkdir -p $HOME/.kube

$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

$ sudo chown $(id -u):$(id -g) $HOME/.kube/config

И примените конфигурацию calico

$ curl https://docs.projectcalico.org/v3.8/manifests/calico.yaml -O

$ vim calico.yaml

Change:

IP_AUTODETECTION_METHOD: "interface=wg*"

CALICO_IPV4POOL_CIDR: "10.250.0.0/16"

Значение IP_AUTODETECTION_METHOD должно быть wg0, чтобы все узлы подключались через VPN, это гарантировало, что он нормально работает без NAT, и все будет в безопасности.

CALICO_IPV4POOL_CIDR Необходим, чтобы убедиться, что все модули созданы в правильной сети.

Когда это правильно, мы можем применить конфигурацию сети:

$ kubectl apply -f calico.yaml

Обратите внимание на команду join, чтобы мы могли добавить еще один узел.

Мне нравится, что k8s также планирует модули на главном сервере, чего по умолчанию не происходит, поэтому мне нужно запустить:

$ kubectl taint nodes --all node-role.kubernetes.io/master-

Практическая работа № 10 Работа с контейнерами Kubernetes в среде Proxmox VE

Задание:

Установка того, с каким Kubernetes-кластером взаимодействует kubectl и изменяет конфигурационную информацию. Подробную информацию о конфигурационном файле смотрите на странице Authenticating Across Clusters with kubeconfig.

kubectl config view # показать объединённые настройки kubeconfig

использовать несколько файлов kubeconfig одновременно и посмотреть объединённую конфигурацию из этих файлов

KUBECONFIG=~/.kube/config:~/.kube/kubconfig2

kubectl config view

```
# получить пароль для пользователя e2e
kubectl config view -o jsonpath='{.users[?(@.name == "e2e")].user.password}'

# показать первого пользователя
kubectl config view -o jsonpath='{.users[].name}'

# получить список пользователей
kubectl config view -o jsonpath='{.users[*].name}'

# показать список контекстов
kubectl config get-contexts

# показать текущий контекст (current-context)
kubectl config current-context

# установить my-cluster-name как контекст по умолчанию
kubectl config use-context my-cluster-name

# добавить новую конфигурацию для кластера в kubeconf с базовой аутентификацией
kubectl config set-credentials kubeuser/foo.kubernetes.com --username=kubeuser --
password=kubepassword

# сохранить пространство имен для всех последующих команд kubectl в этом контексте.
kubectl config set-context --current --namespace=ggckad-s2

# установить контекст, используя имя пользователя и пространство имен.
kubectl config set-context gce --user=cluster-admin --namespace=foo \
&& kubectl config use-context gce
```

удалить пользователя foo

kubectl config unset users.foo

Практическая работа № 11 Оркестрациия Kubernetes в среде Proxmox VE

Задание:

1 Разворачивание мастер-ноды Kubernetes

Собственно мастер-нода мало чем отличается от воркеров в процессе установки, по этому после начальной установки всех необходимых пакетов ее можно будет использовать как шаблон для воркер-нод.

В данной статье не будет описания того как работать с Proxmox, подразумевается что у вас уже есть опыт работы с данным продуктом

1.1 Создание шаблона CloudInit с Ubuntu 20.04

Чтобы не заморачиваться каждый раз с долгой установкой системы создадим CloudInit-шаблон Ubuntu 20.04

Скачиваем образ

wget https://cloud-images.ubuntu.com/focal/current/focal-server-cloudimg-amd64.img

Создаем новую ВМ

qm create 9000 --memory 2048 --cores 4 --net0 virtio,bridge=vmbr0 --name Ubuntu-20.04-CloudInit-template

Импортируем загруженный образ диска в хранилище local-lvm

qm importdisk 9000 focal-server-cloudimg-amd64.img local-lvm

Подключаем вновь созданный диск к ВМ

qm set 9000 --scsihw virtio-scsi-pci --scsi0 local-lvm:vm-9000-disk-0

Подключаем CloudInit CD-ROM диск к ВМ

qm set 9000 --ide2 local-lvm:cloudinit

Дополнительные параметры

qm set 9000 --boot c --bootdisk scsi0

qm set 9000 --serial0 socket --vga serial0

Преобразуем в шаблон

qm template 9000

1.2 Создание шаблона для Kubernetes нод

Теперь создаем первую ВМ, в которой будет все необходимое для разворачивания ноды Kebernetes, которую мы затем так же преобразуем в шаблон

1

qm clone 9000 9001 --name kube-template --full --storage local-lvm

qm set 9001 --memory 4096 --agent enabled=1

qm resize 9001 scsi0 3G

Запускаем машину клона - манину 9001 и устанавливаем всё необходимаое, а именно - docker и kubernetes:

Qemu-agent:

sudo apt-get install qemu-guest-agent

Docker:

sudo apt-get remove docker docker-engine docker.io containerd runc

sudo apt-get update

sudo apt-get install ca-certificates curl gnupg lsb-release

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io

Kubernetes:

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl open-iscsi nfs-common

sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg

echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update

sudo apt-get install kubeadm=1.23.0-00 kubelet=1.23.0-00 kubectl=1.23.0-00 kubernetes-cni=0.8.7-00

sudo apt-mark hold kubeadm kubectl kubelet kubernetes-cni

Выключаем нашу машину и делаем из нее шаблон:

qm template 9001

1.3 Создание и настройка Master-ноды

Теперь приступаем к созданию мастер-ноды Kubernetes:

qm clone 9001 100 --name kube-master --full --storage local-lvm

qm resize 100 scsi0 15G

Инициализация кластера:

Создаем файл параметров, заменив <server_ip_address> на свой:

cat <<EOF >kubeadm-config-master.yaml

kubeadm-config-master.yaml

apiVersion: kubelet.config.k8s.io/v1beta1

kind: KubeletConfiguration

cgroupDriver: cgroupfs

featureGates:

  NodeSwap: true

failSwapOn: false

memorySwap:

SwapBehavior: LimitedSwap

---

apiVersion: kubeadm.k8s.io/v1beta2

kind: ClusterConfiguration

api:

advertiseAddress: <server_ip_address>

networking:

podSubnet: 10.244.0.0/16

EOF

Запускаем процесс создания кластера:

sudo kubeadm init --config kubeadm-config-master.yaml --ignore-preflight-errors=swap

После успешной инициализации выполняем рекомендации по добавлению конфига для kubectl в конце вывода

Устанавливаем CNI (Container Network Interface):

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

И убеждаемся что наша мастер-нода готова:

kubectl get nodes

Мастер-нода должна иметь статус Ready

Для того чтобы работало автодополнение выполняем:

kubectl completion bash >/etc/bash_completion.d/kubectl

1.4 Создание и подключение Worker-нод к Master-ноде

Создаем соответствующие виртуальные машины:

qm clone 9001 101 --name kube-worker1 --full --storage local-lvm

qm resize 101 scsi0 40G

qm clone 9001 102 --name kube-worker2 --full --storage local-lvm

qm resize 102 scsi0 40G

qm clone 9001 103 --name kube-worker3 --full --storage local-lvm

qm resize 103 scsi0 40G

После запуска каждой из гостевых машин выполняем следующие действия:

sudo rm /etc/machine-id

sudo systemd-machine-id-setup

sudo reboot now

На мастер ноде получаем комманду для подключения выполнив:

sudo kubeadm token create --print-join-command

И затем используем полученную команду на воркерах:

1

sudo kubeadm join <ip_мастер_ноды>:6443 --token m2aw6a.czbgzvd4bqy2p3ed --discovery-token-ca-cert-hash sha256:537b55a1a82596580ddfcb0a9511b0ae3795e0a2ecfa6f8da44ac24f4da61553

2 Настройка кластера

2.1 Установка MetalLB

MetalLB нужен чтобы ingress-контроллер знал в каком диапазоне IP-адресов он может работать. В моем случае это будет лишь один адрес, но тем ни мение - на всякий случай, лучше чтобы он был. Для его установки, сначала, нам понадобится создать файл конфигурации, где вместо <IP_ingress-node> указать IP мастер-ноды:

cat <<EOF > metallb-values.yaml

metallb-values.yaml

configInline:

  address-pools:

   - name: default

     protocol: layer2

     addresses:

     - <IP_ingress-node>/32

EOF

затем немного изменить конфигурацию сервиса kube-proxy

```
kubectl get configmap kube-proxy -n kube-system -o yaml | \
sed -e "s/strictARP: false/strictARP: true/" | \
kubectl apply -f - -n kube-system
```

после чего можно установить сам MetalLB по средствам helm. Для этого делаем слудующее:

```
helm repo add metallb https://metallb.github.io/metallb
helm install metallb metallb/metallb -f metallb-values.yaml --namespace metallb-system --create-namespace
```

2.2 Установка ingress-контролера Nginx

```
helm upgrade --install ingress-nginx ingress-nginx \
  --repo https://kubernetes.github.io/ingress-nginx \
  --namespace ingress-nginx --create-namespace
```

2.3 Установить менеджера сертификатов cert-manager

Данный менеджер упрощает, как следует из названия, процесс создания TLS-сертификатов. И на мой взгляд, просто необходим, если вы планируете публикацию сервисов в интернете с использованием центров сертификации, например - LetsEncrypt.

```
helm repo add jetstack https://charts.jetstack.io
helm repo update
kubectl apply -f https://github.com/jetstack/cert-manager/releases/download/v1.6.1/cert-manager.crds.yaml
helm install \
  cert-manager jetstack/cert-manager \
  --namespace cert-manager \
  --create-namespace \
  --version v1.6.1
```

Так же можно сразу создать эмитент для кластера, который будет подписывать наши сертификаты, на примере Let’s Encrypt. Для этого создаем манифест примерно такого содержания:

```
cat <<EOF > letsencrypt-issuer.yaml
```

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: letsencrypt
spec:
  acme:
    # URL ACME сервера
    server: https://acme-v02.api.letsencrypt.org/directory
    # Email адрес используемый для ACME регистрации
    email: your@mail.com
    # Имя секрета используемого для хранения приватного ключа ACME-аккаунта
    privateKeySecretRef:
      name: letsencrypt
    # Добавление HTTP-01 challenge provider
    solvers:
    - http01:
        ingress:
          class: nginx
EOF
kubectl apply -f letsencrypt-issuer.yaml
```

Проверить список эмитентов можно выполнив kubectl get clusterissuers. Результат вывода примерно такой:

```
NAME          READY   AGE
letsencrypt   True    1h
```

2.4 Установка Rancher

```
helm repo add rancher-latest https://releases.rancher.com/server-charts/latest
helm install rancher rancher-latest/rancher \
  --namespace cattle-system \
  --set hostname=rancher.my.org \
```

--set replicas=3 \

--create-namespace

Как получить пороль будет написано после установки

2.5 Установка Longhorn

Установка Longhorn осуществляется в самом Rancher’е нажатием одной кнопки “Install”

На этом установку MVP Kubernetes кластера можно считать законченной

Обновлена 2022-06-27

Практическая работа № 12 Настройка логирования контейнеров.

Задание:

Dokku — это легковесный инструмент управления контейнерами Docker, который упрощает развертывание и управление приложениями. Для начала работы с Dokku выполните следующие шаги:

Инструкция по установке Dokku:

1. Загрузка скрипта установки:

`wget -NP . https://dokku.com/bootstrap.sh`

2. Запуск установщика:

`sudo DOKKU_TAG=v0.32.3 bash bootstrap.sh`

3. Настройка домена:

`dokku domains:set-global dokku.me`

4. Добавление SSH-ключа: Замените "your-public-key-contents-here" на ваш публичный SSH-ключ.

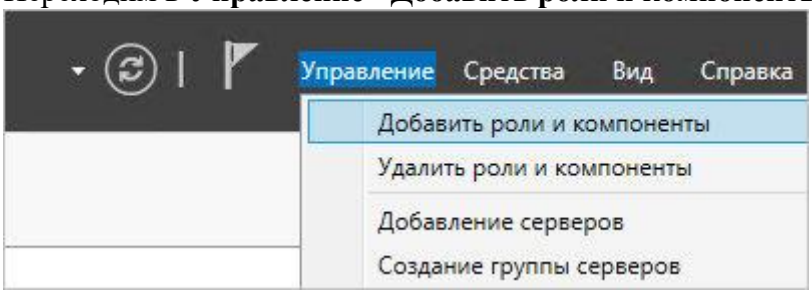
`echo "your-public-key-contents-here" | dokku ssh-keys:add admin`

Практическая работа № 13 Настройка виртуальных машин для шлюза удалённого рабочего стола

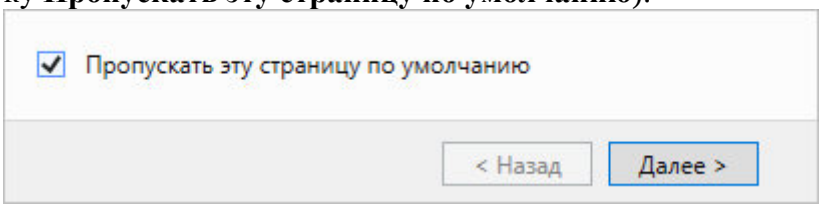
Задание:

Открываем **Диспетчер серверов**:

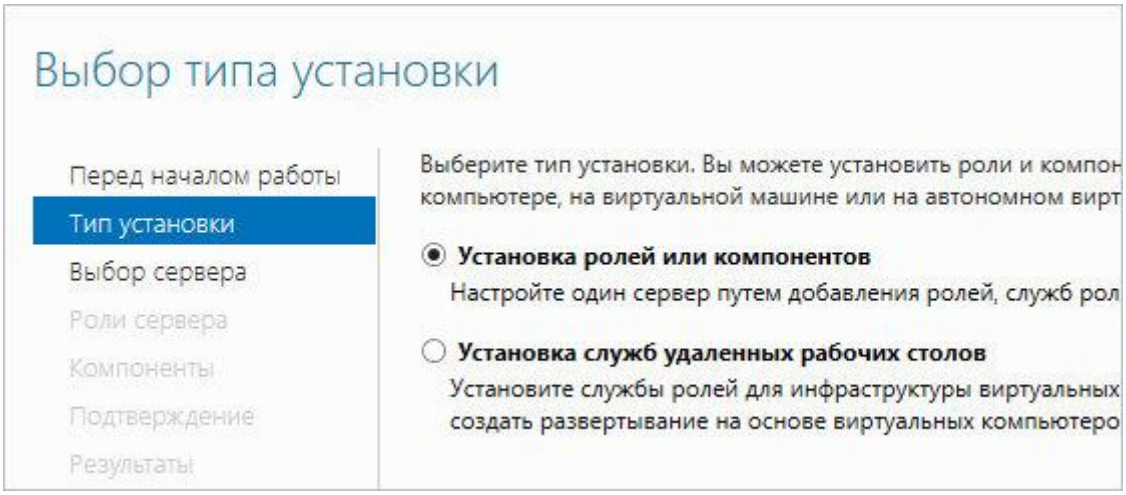
Переходим в **Управление - Добавить роли и компоненты**:



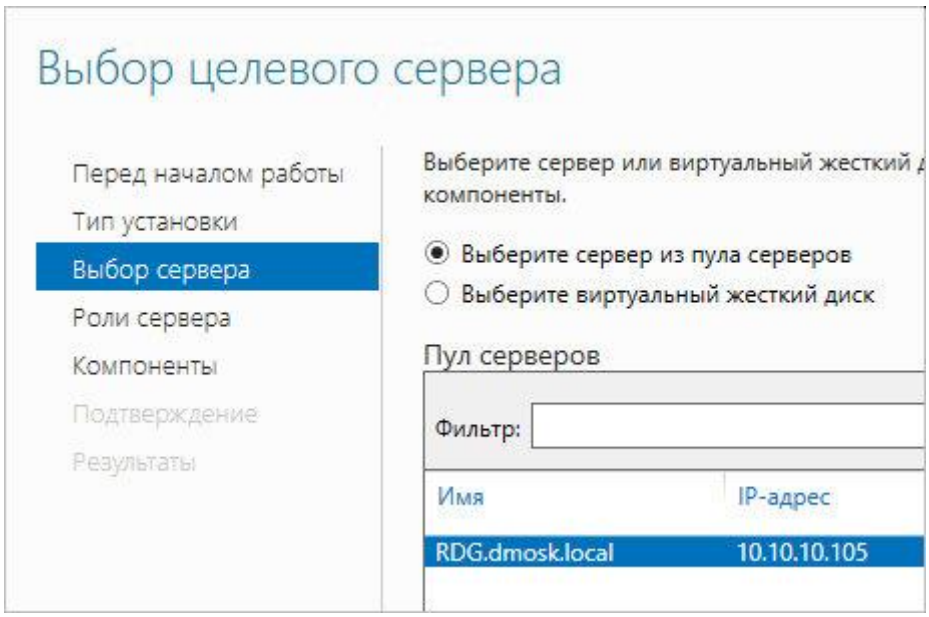
При появлении окна приветствия нажимаем **Далее** (при желании, можно поставить галочку **Пропускать эту страницу по умолчанию**):



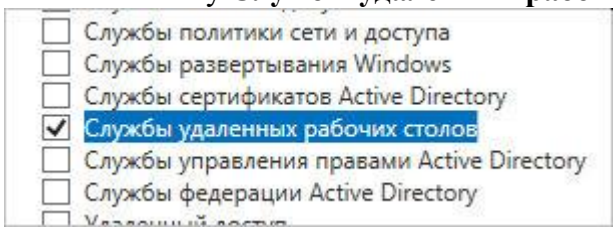
На страницы выбора типа установки оставляем выбор на **Установка ролей или компонентов**:



Выбираем целевой сервер — если установка выполняется на сервере локально, то мы должны увидеть один сервер для выбора:



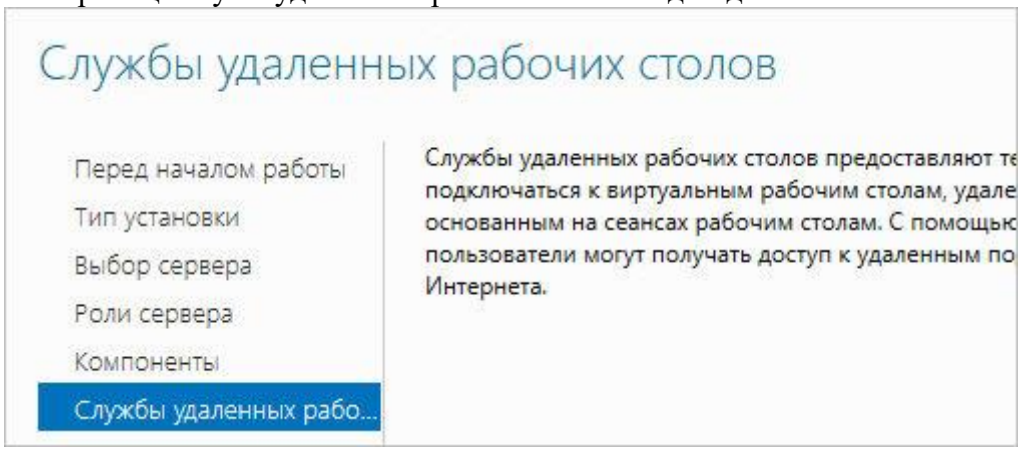
Ставим галочку **Службы удаленных рабочих столов**:



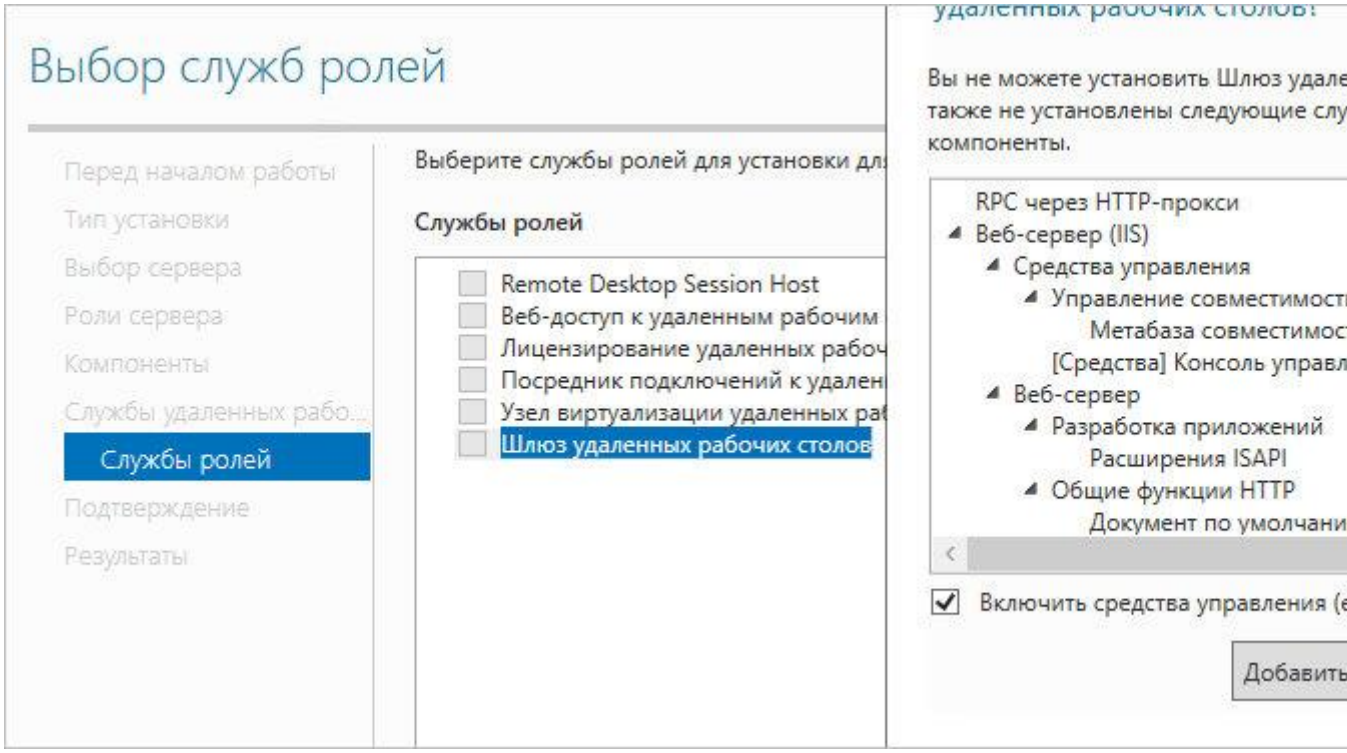
Дополнительные компоненты нам не нужны:

... просто нажимаем **Далее**.

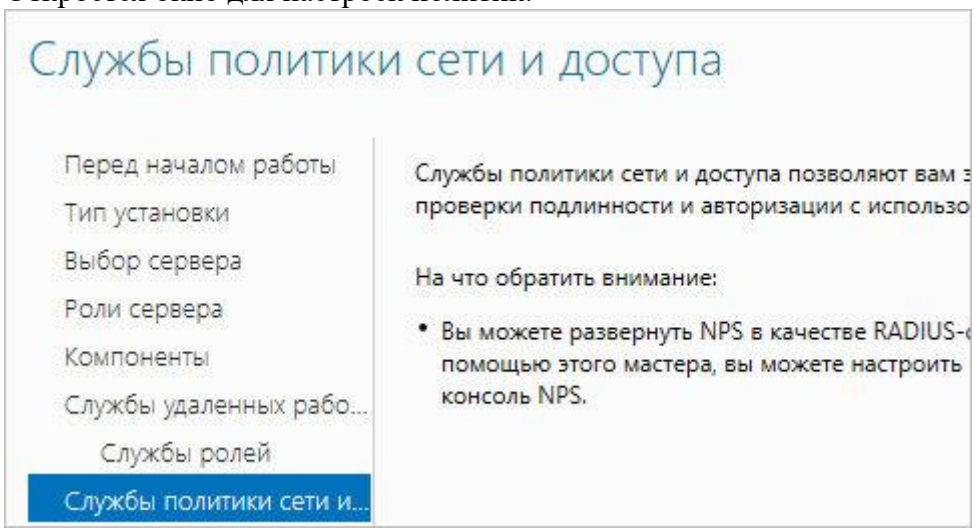
На странице служб удаленных рабочих столов идем дальше:



Выбираем конкретные роли — нам нужен **Шлюз удаленных рабочих столов**. После установки галочки появится предупреждение о необходимости поставить дополнительные пакеты — кликаем по **Добавить компоненты**:

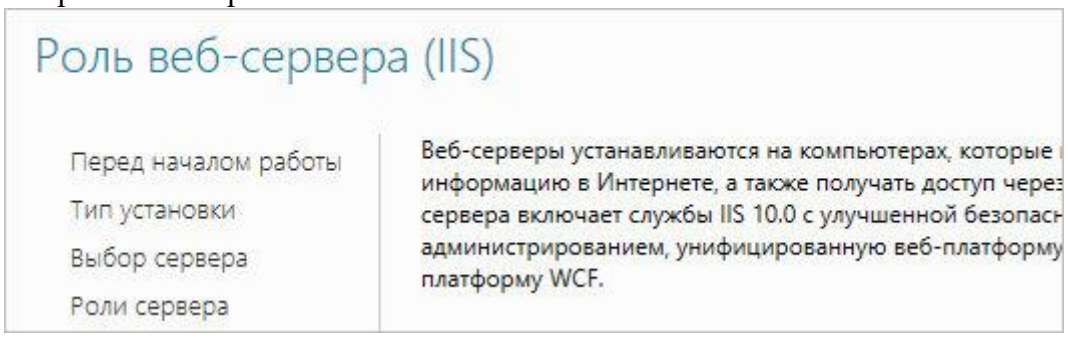


Откроется окно для настроек политик:



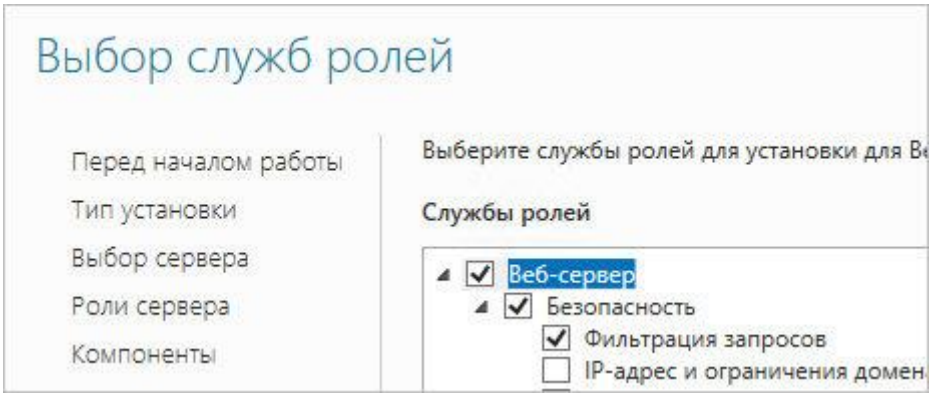
... нажимаем **Далее**.

Откроется окно роли IIS:



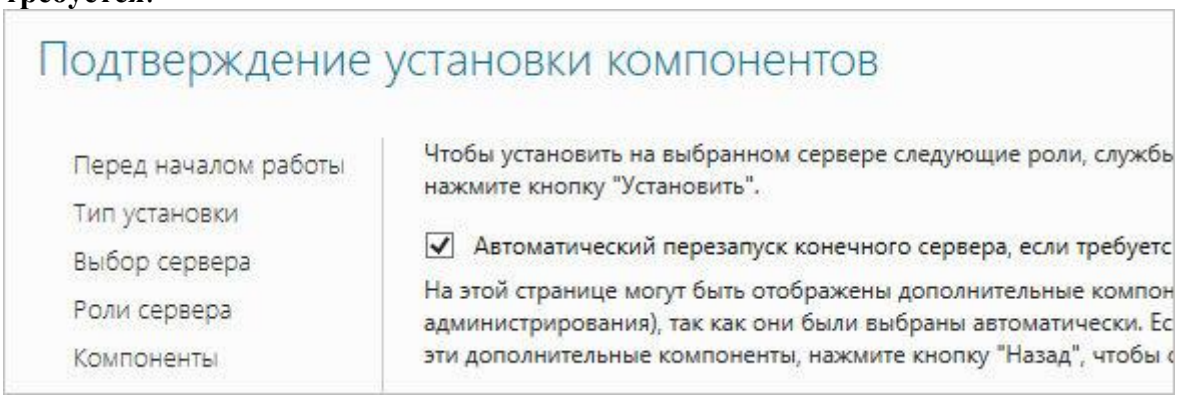
... также нажимаем **Далее**.

При выборе служб ролей веб-сервера ничего не меняем:



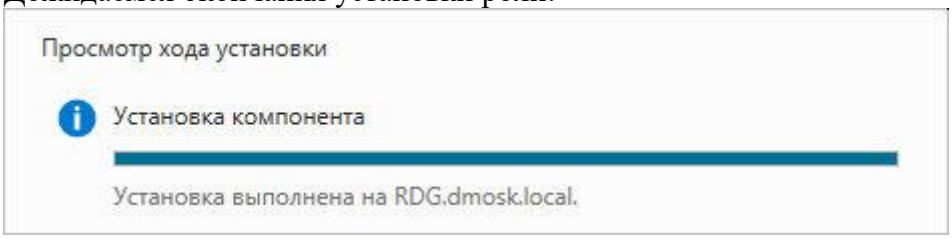
... и идем дальше.

В последнем окне ставим галочку **Автоматический перезапуск конечного сервера, если требуется**:



Нажимаем **Установить**:

Дожидаемся окончания установки роли:



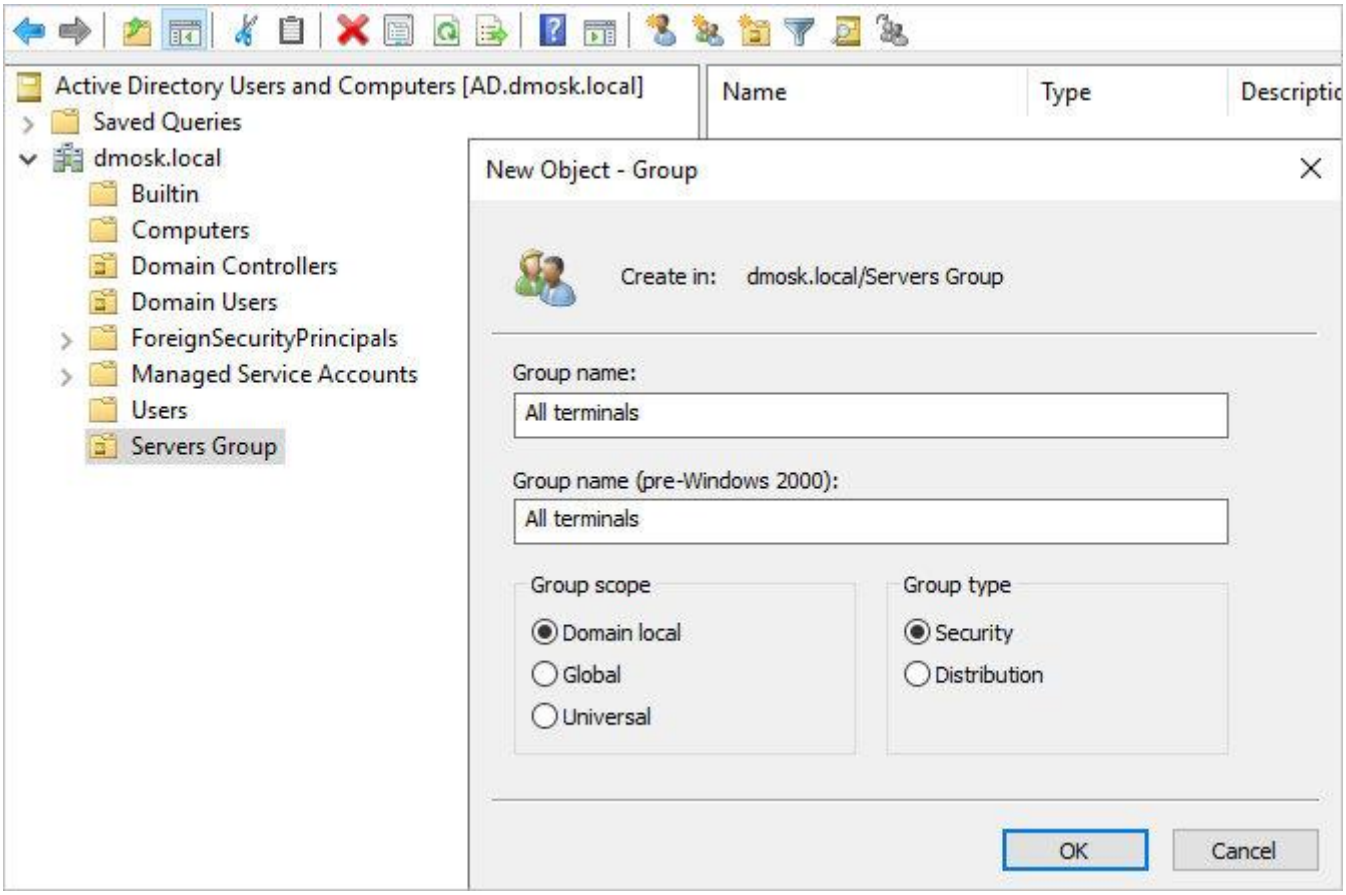
Сервер может уйти в перезагрузку.

Настройка RDG

Для настройки Microsoft Remote Desktop Gateway мы создадим группу компьютеров в Active Directory, настроим политику для RDG и создадим сертификат.

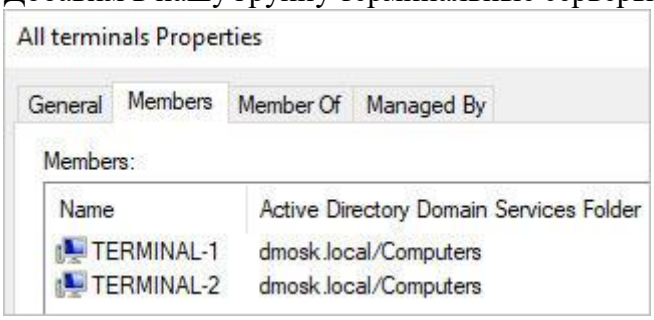
Создание групп для терминальных серверов

Политика ресурсов позволит задать нам конкретные серверы, на которые терминальный шлюз позволит нам подключаться. Для этого мы откроем консоль **Active Directory - Users and computers** (Пользователи и компьютеры Active Directory) и создаем группу:



** в данном примере мы создаем группу **All terminals** в организационном юните **Servers Group**. Это группа безопасности (**Security**), локальная в домене (**Domain local**).*

Добавим в нашу группу терминальные серверы:



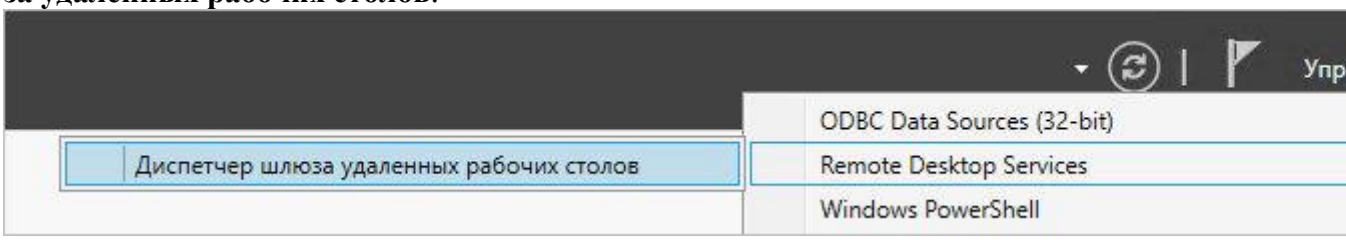
** в данном примере у нас используются два сервера — **Terminal-1** и **Terminal-2**.*

Закрываем консоль Active Directory - Users and computers.

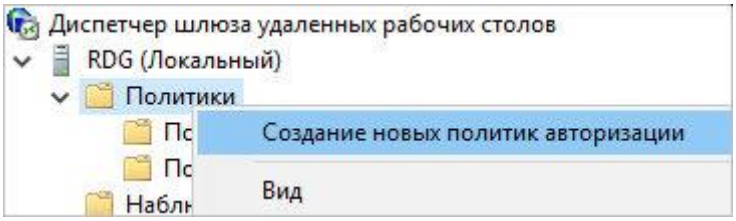
Настройка политик

Для предоставления доступа к нашим терминальным серверам, создадим политики для подключений и ресурсов.

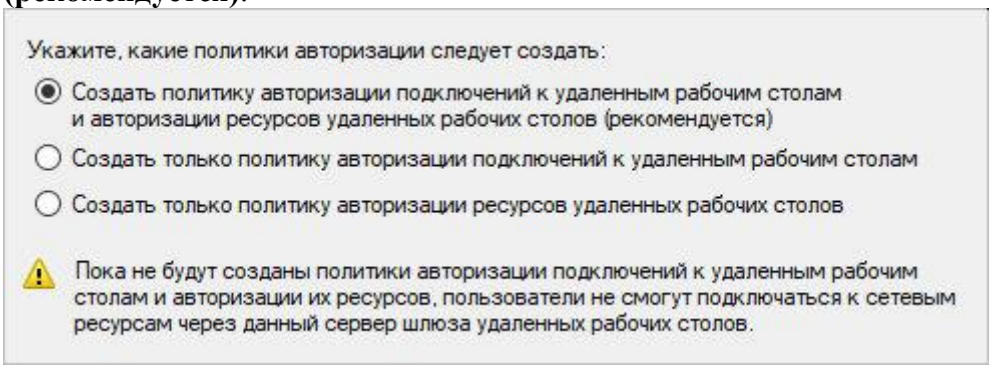
В диспетчере сервера переходим в **Средства** - **Remote Desktop Services** - **Диспетчер шлюза удаленных рабочих столов**:



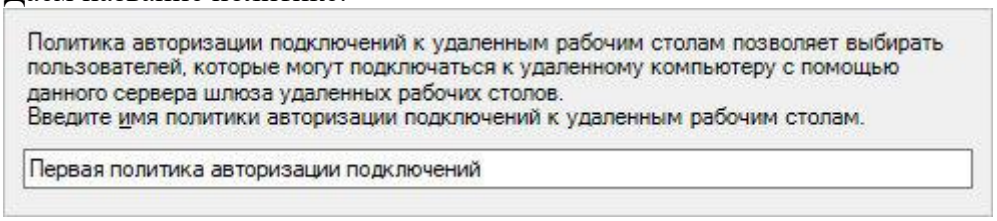
Раскрываем сервер - кликаем правой кнопкой по **Политики** - выбираем **Создание новых политик безопасности**:



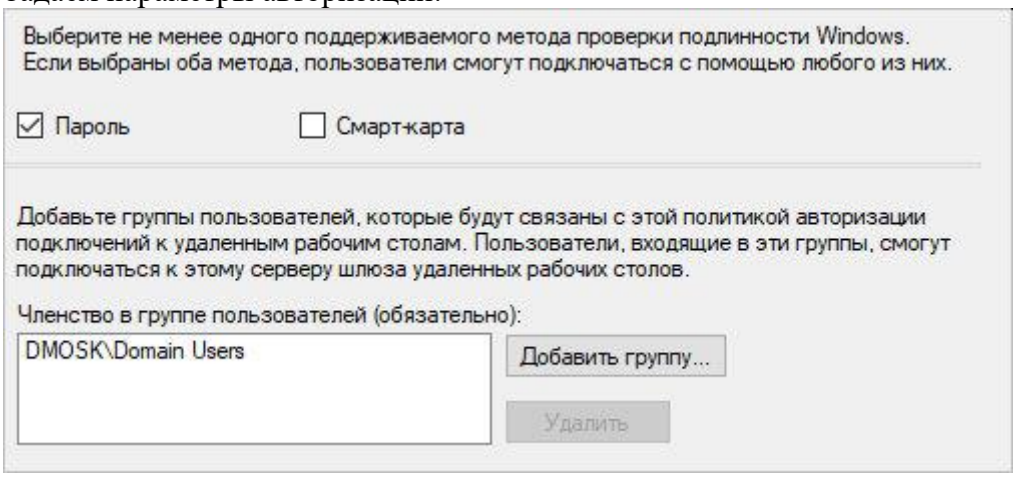
Устанавливаем переключатель в положении **Создать политику авторизации подключений к удаленным рабочим столам и авторизации ресурсов удаленных рабочих столов (рекомендуется)**:



Даем название политике:

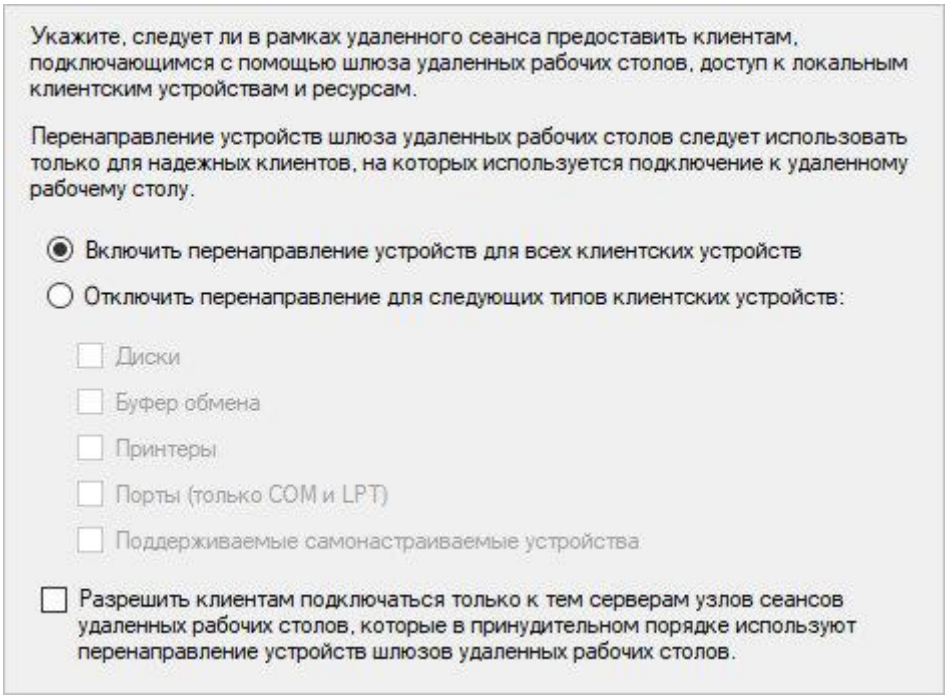


Задаем параметры авторизации:



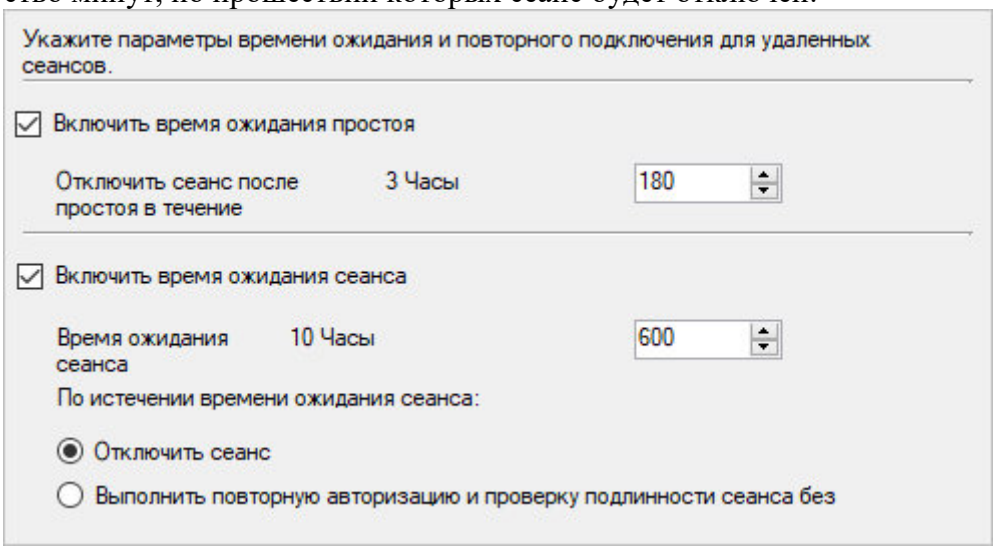
** мы указали, что пользователи должны подтверждать право вводом пароля, также мы указали, что для применения политики они должны принадлежать группе* ***Domain Users.***

В следующем окне есть возможность настроить ограничения использования удаленного рабочего стола. При желании, можно их настроить:



** в нашем случае ограничений нет. При необходимости, устанавливаем переключатель в положение **Отключить перенаправление для следующих типов клиентских устройств** и оставляем галочки пункты для ограничений.*

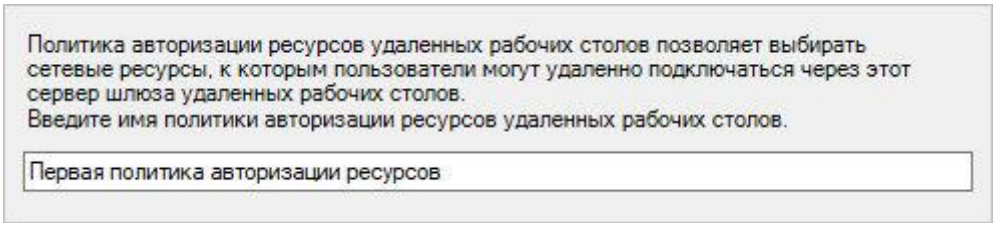
Далее настраиваем временные ограничения использования удаленного подключения. Если в этом есть необходимость, оставляем галочки в состоянии **Включить** и указываем количество минут, по прошествии которых сеанс будет отключен:



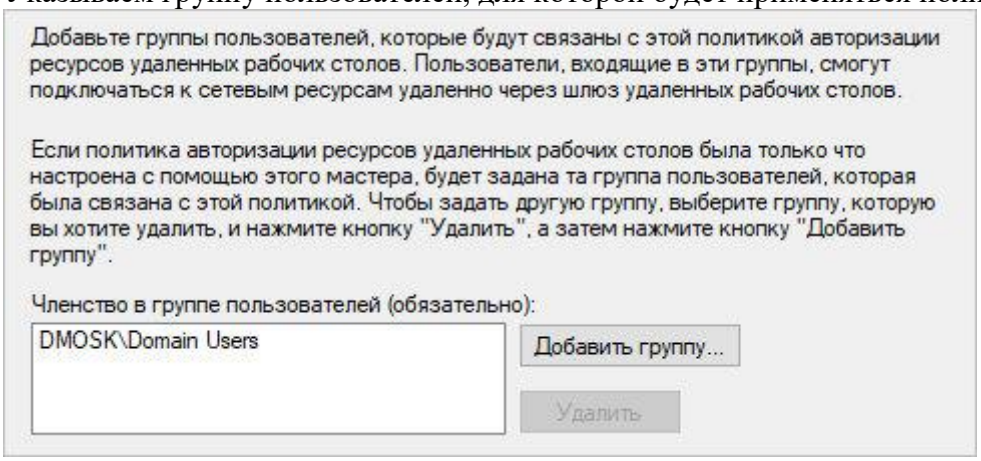
В следующем окне мы увидим вне введенные настройки:

Идем далее.

Откроется страница создания политики для авторизации ресурса — задаем для нее название:

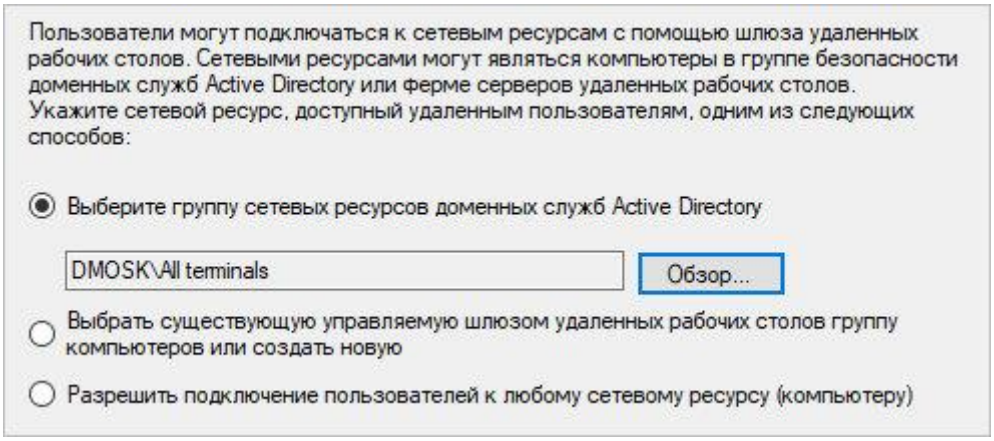


Указываем группу пользователей, для которой будет применяться политика:



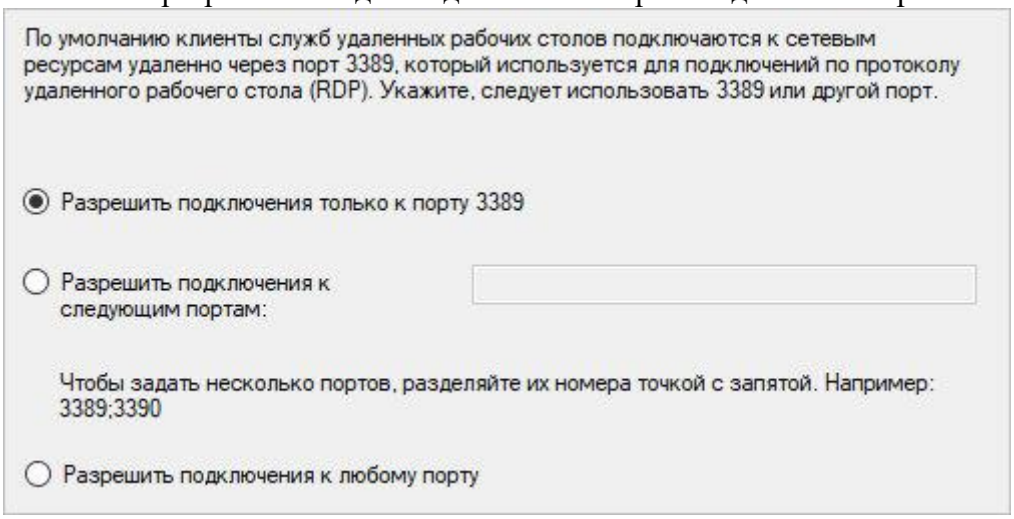
** как и при создании первой политики, мы добавили группу **Domain Users**.*

Теперь выбираем группу ресурсов, на которую будет разрешен доступ со шлюза терминалов:



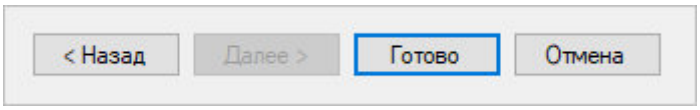
** мы выбрали группу, созданную нами ранее в AD.*

Указываем разрешенный для подключения порт или диапазон портов:



** в данном примере мы разрешим подключение по порту **3389**, который используется по умолчанию для RDP.*

Нажимаем **Готово**:

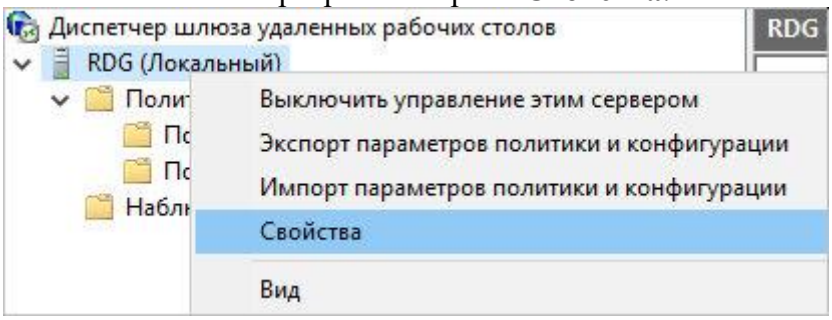


Политики будут созданы.

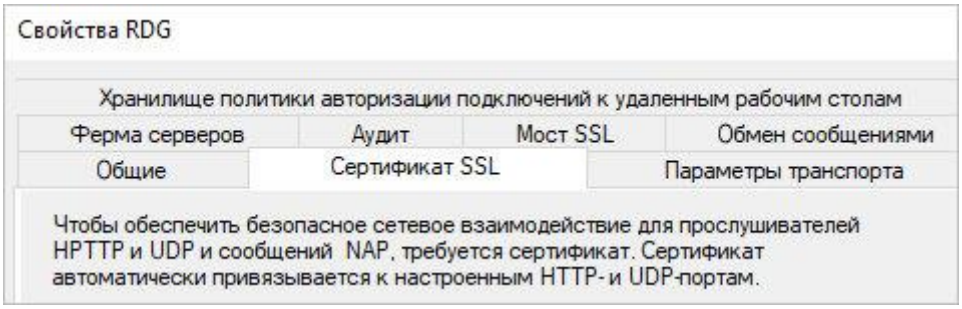
Настройка сертификата

Для работы системы нам необходим сертификат, который можно купить или получить [бесплатно от Let's Encrypt](). Однако, с некоторыми неудобствами, будет работать и самоподписанный. Мы рассмотрим вариант настройки с ним.

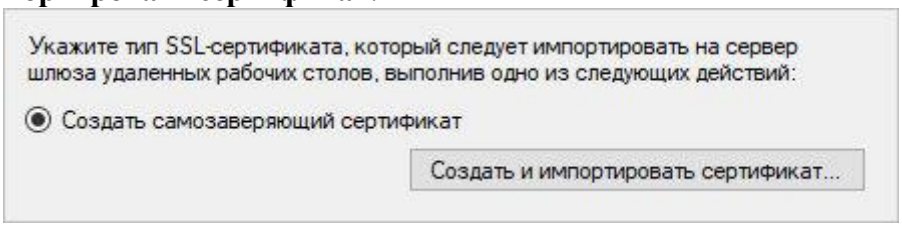
Запускаем «Диспетчер шлюза удаленных рабочих столов» - кликаем правой кнопкой по названию нашего сервера - выбираем **Свойства**:



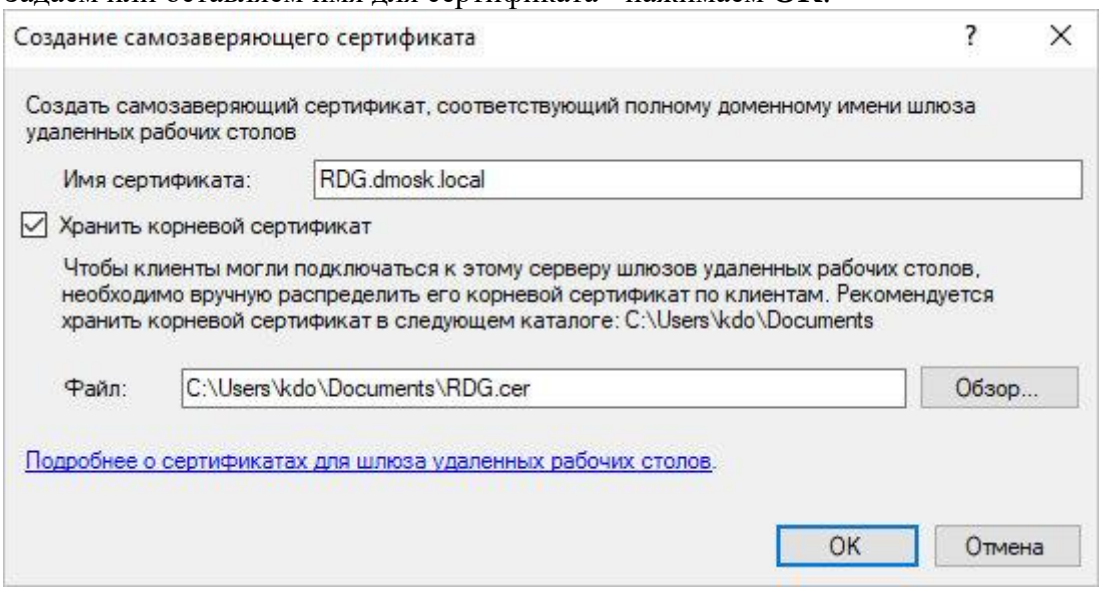
Переходим на вкладку **Сертификат SSL**:



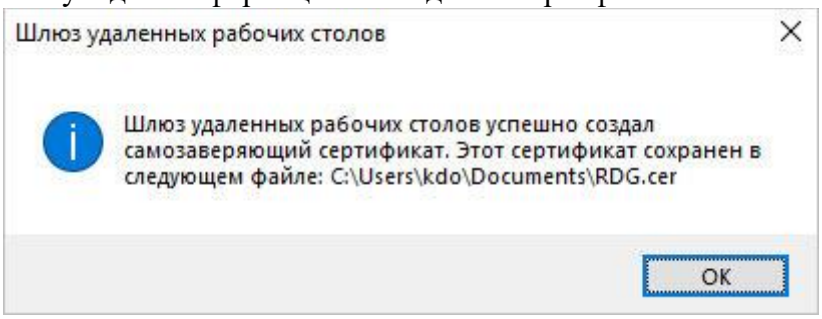
Выбираем вариант **Создать сомозаверяющий сертификат** и кликаем по **Создать и импортировать сертификат**:



Задаем или оставляем имя для сертификата - нажимаем **OK**:



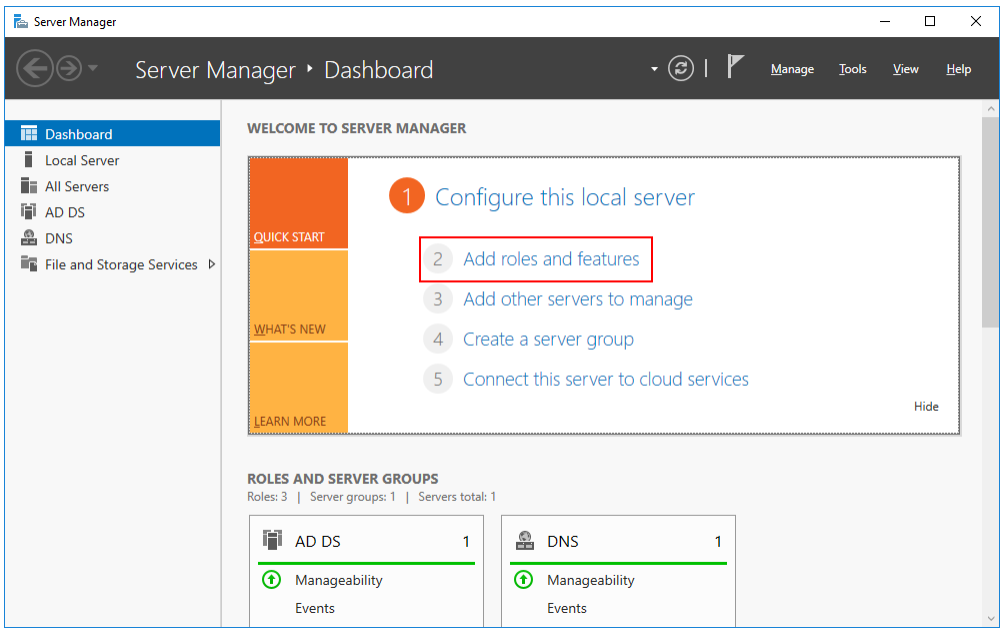
Мы увидим информацию о создании сертификата:



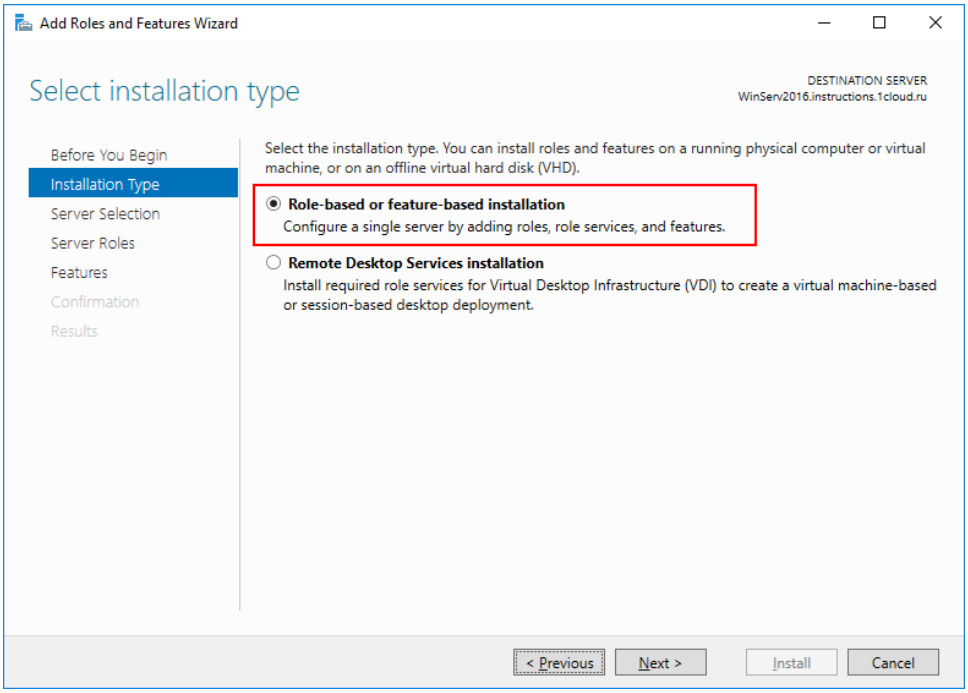
Практическая работа № 14 Настройка межплатформенный бесклиентский шлюз удаленного рабочего стола

Задание:

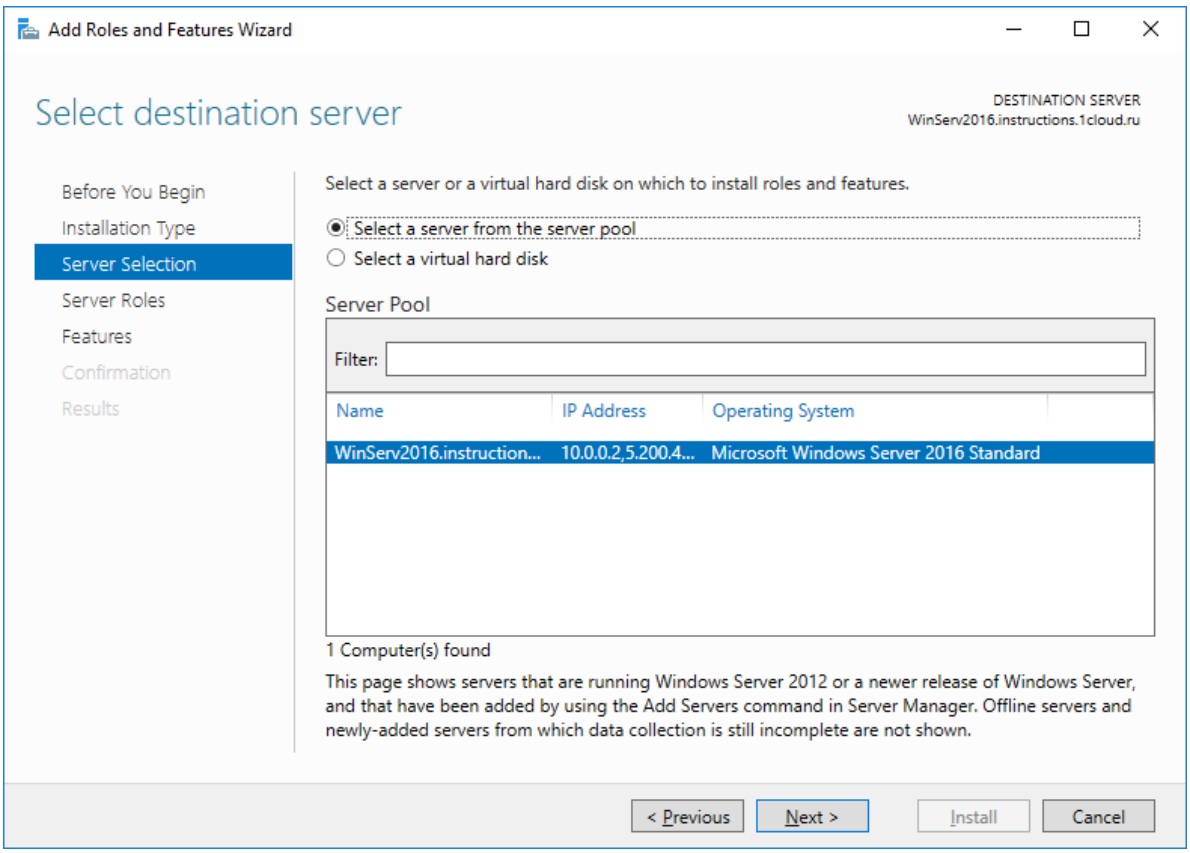
Откройте **Диспетчер серверов** и выберите пункт **Add roles and features**.



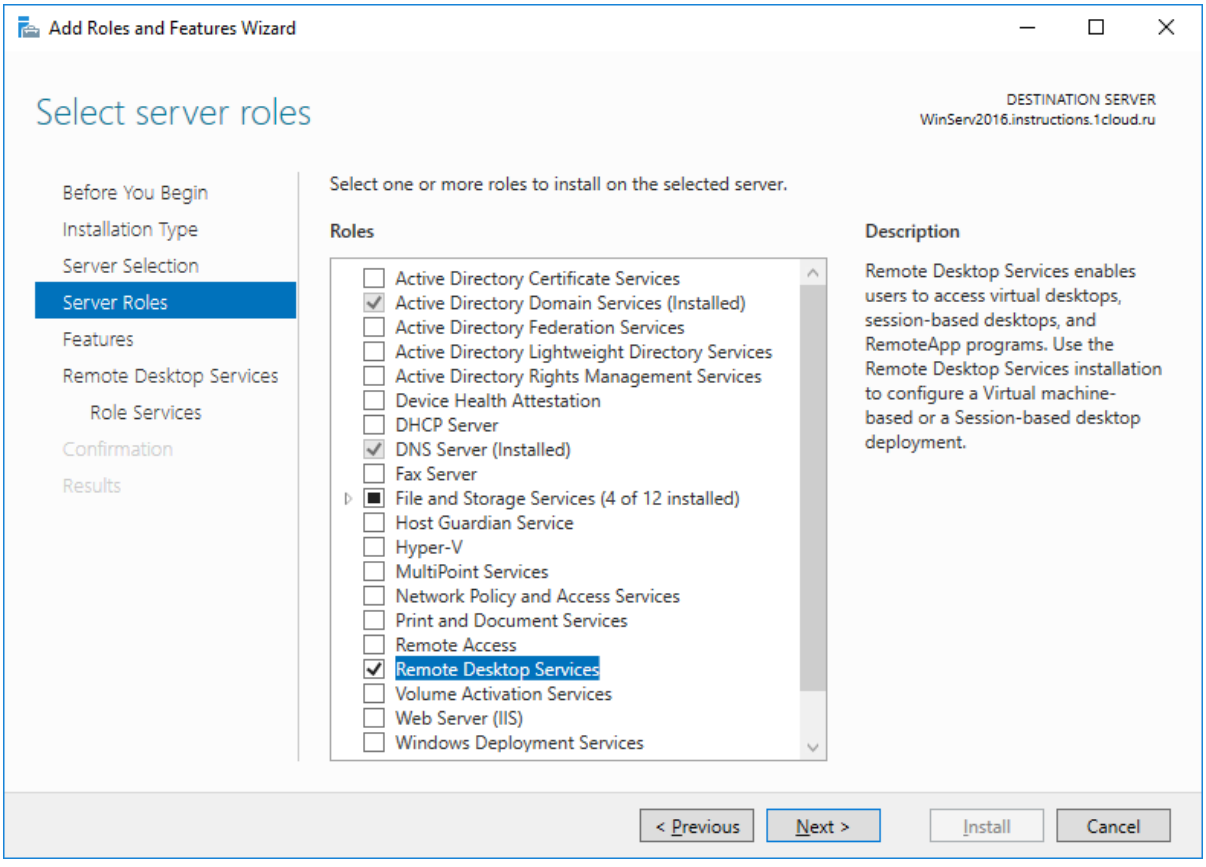
В качестве типа установки укажите **Role-based or feature-based installation**.



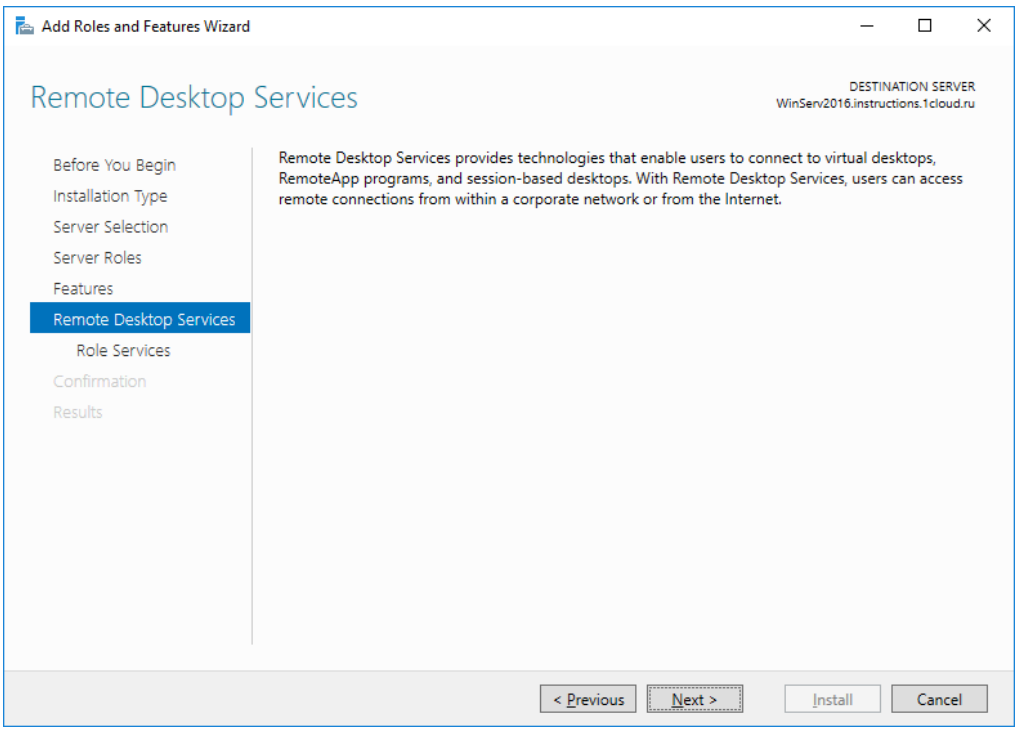
Выберите ваш сервер из пула.



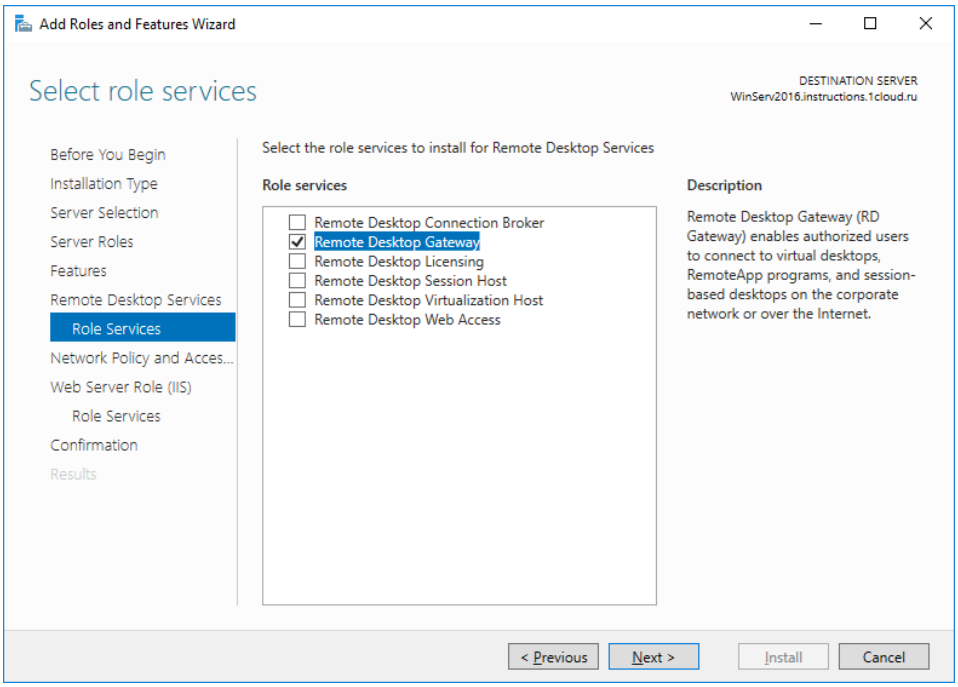
В следующем окне отметьте **Remote Desktop Services**.



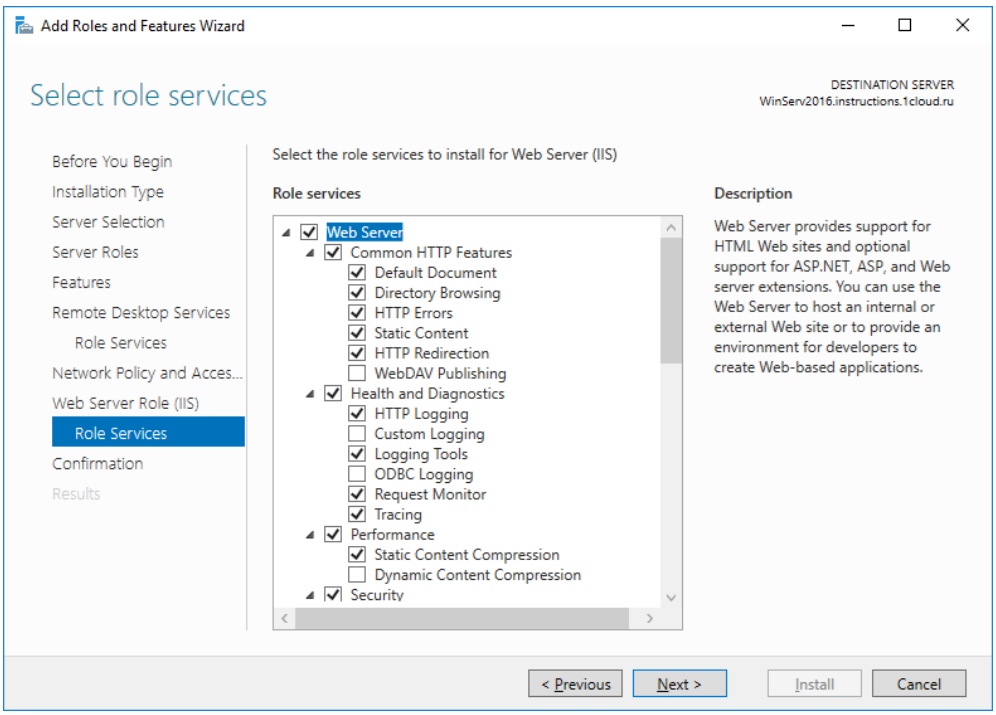
Далее вы увидите краткую информацию о роли.



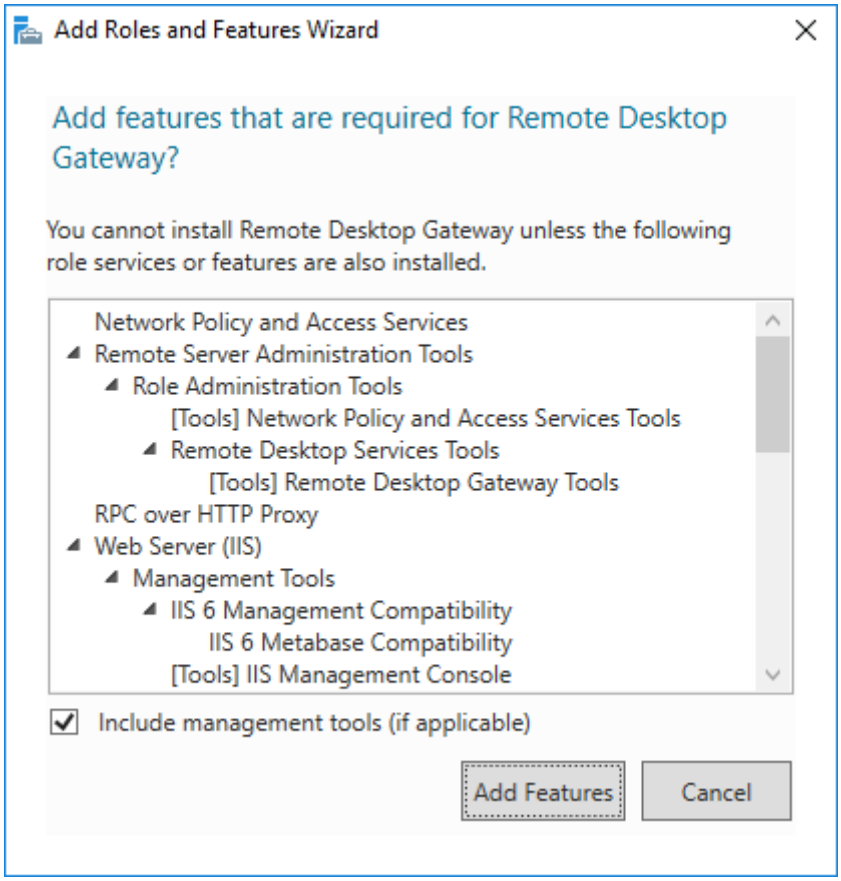
Далее добавьте сервис **Remote Desktop Gateway**.



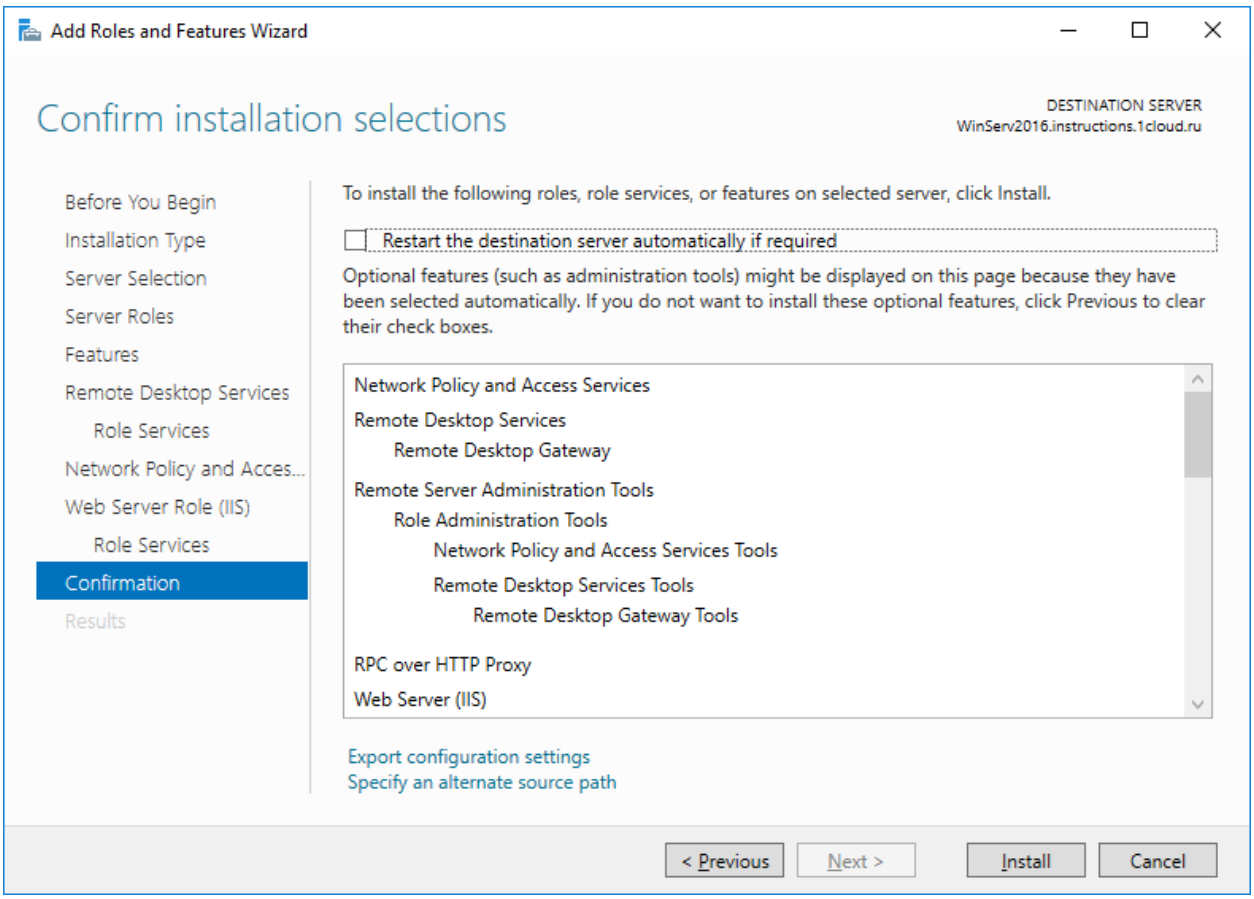
Для работы этого сервиса необходимо веб-сервер IIS и дополнительные административные инструменты, они будут предложены автоматически, если не были установлены ранее.



Добавьте данные функции.

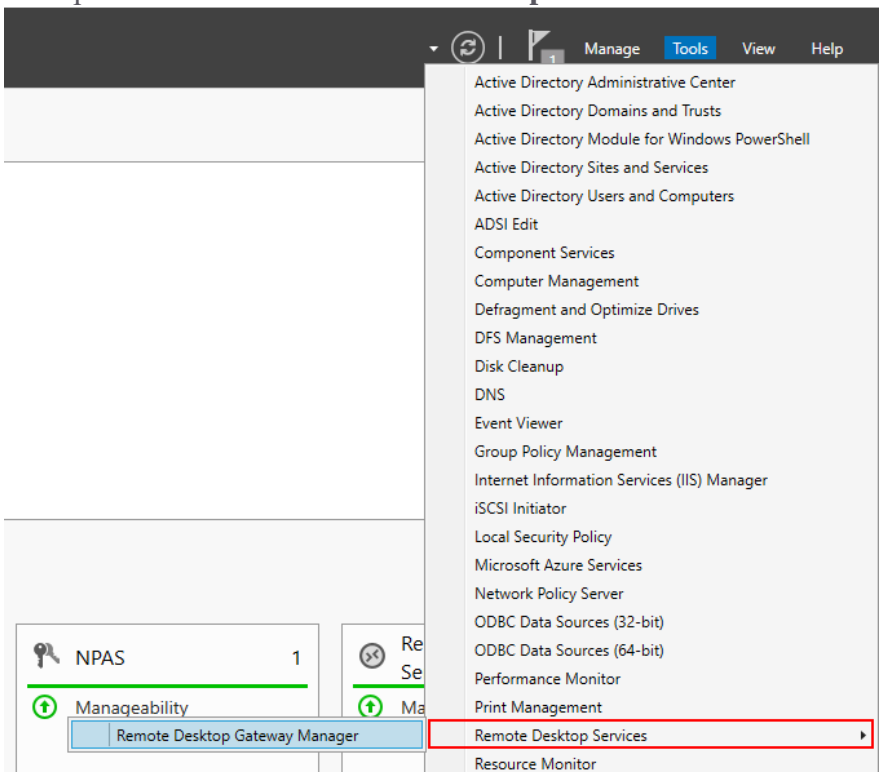


Установите все выбранные компоненты на VPS с помощью кнопки **Install**.

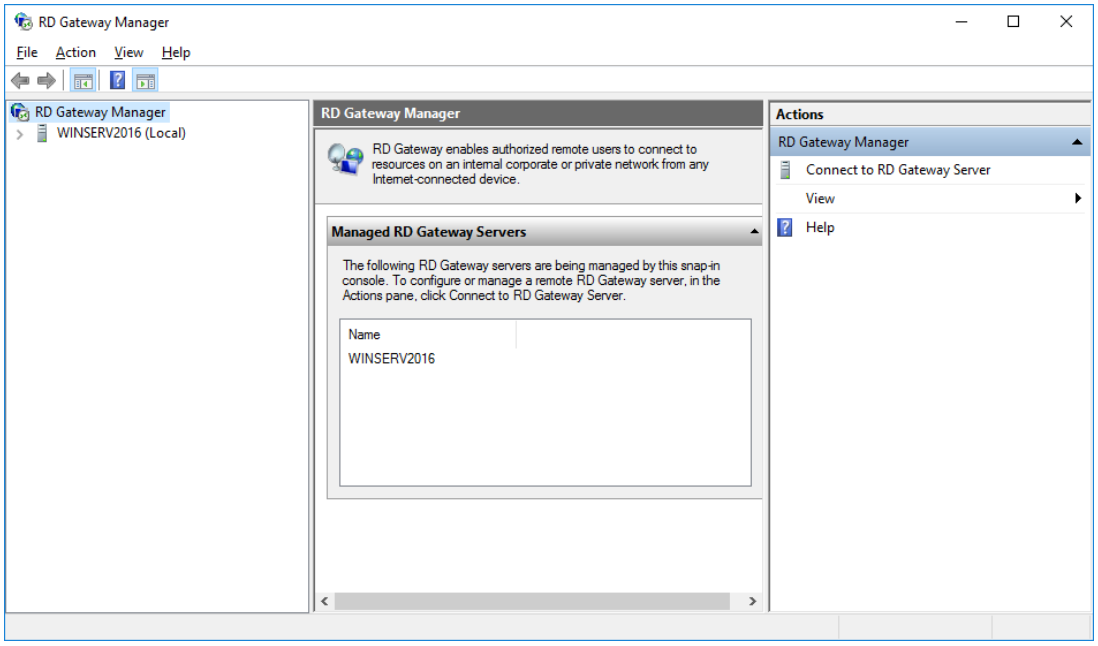


Создание политики авторизации подключения и ресурсов

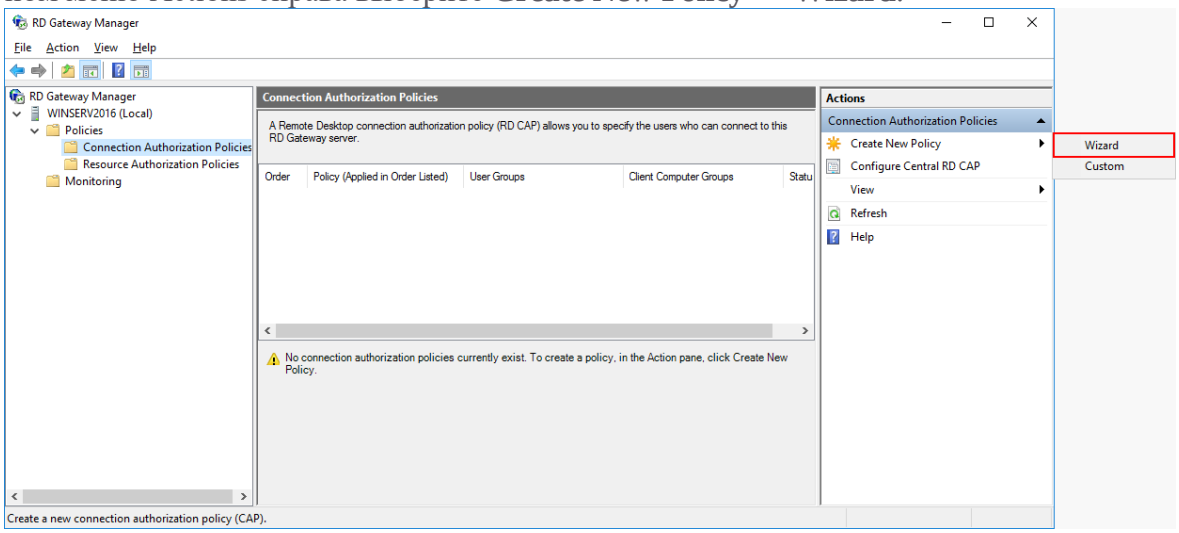
Чтобы открыть Remote Desktop Gateway Manager, в **Диспетчере серверов** выберите Tools и в открывшемся списке **Remote Desktop Services → Remote Desktop Gateway Manager**.



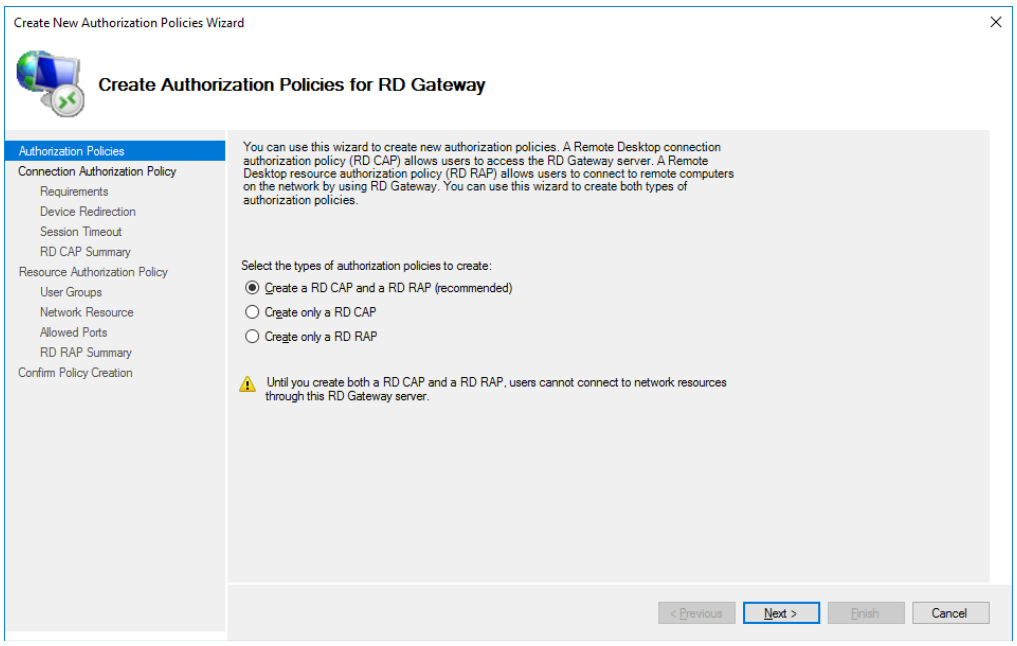
Перед вами откроется менеджер шлюза.



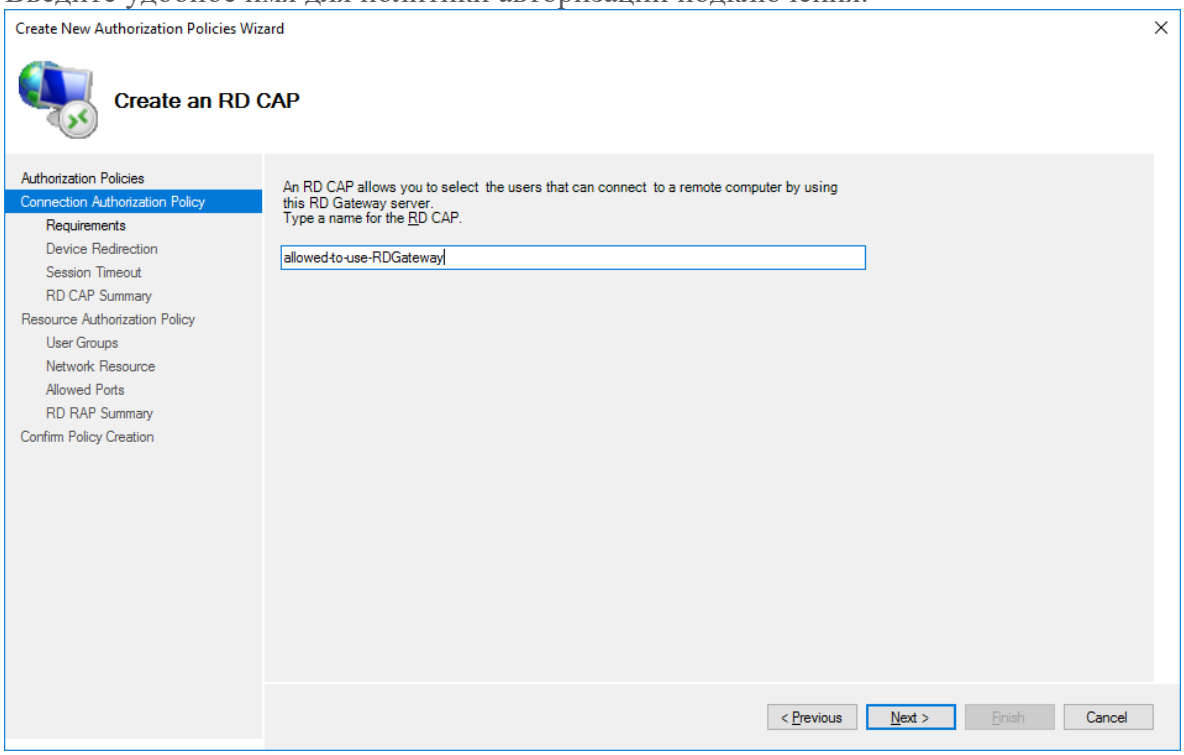
Для создания политик авторизации в древовидной структуре откройте **RD Gateway Manager → <Имя сервера> → Policies → Connection Authorization Policies**. В вертикальном меню Actions справа выберите **Create New Policy → Wizard**.



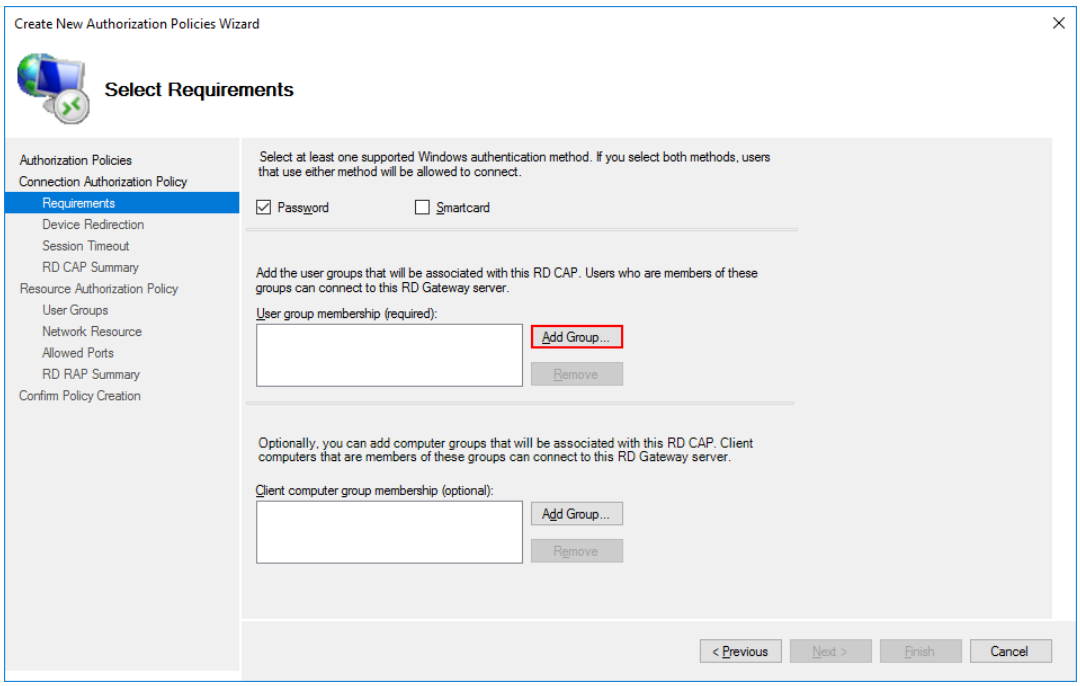
В открывшемся окне выберите Create RD CAP and RD RAP (recommended), чтобы с помощью одного процесса настроить обе политики.



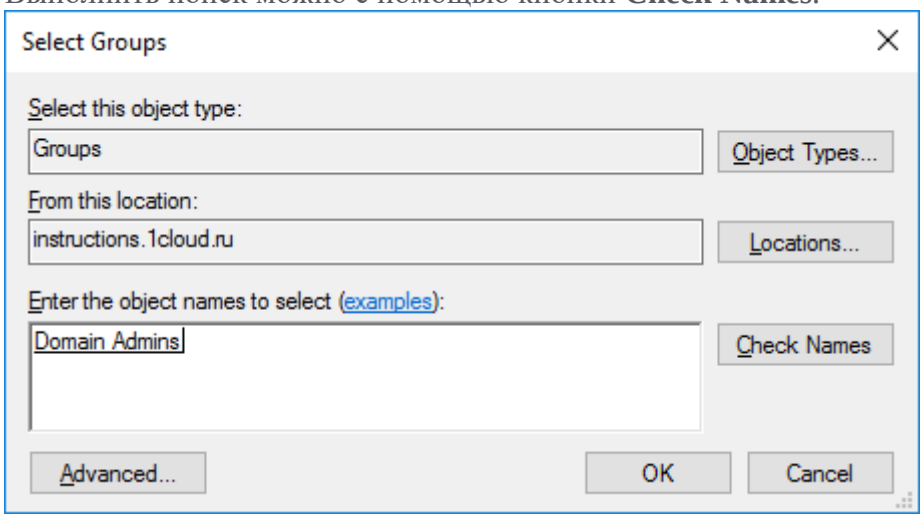
Введите удобное имя для политики авторизации подключения.



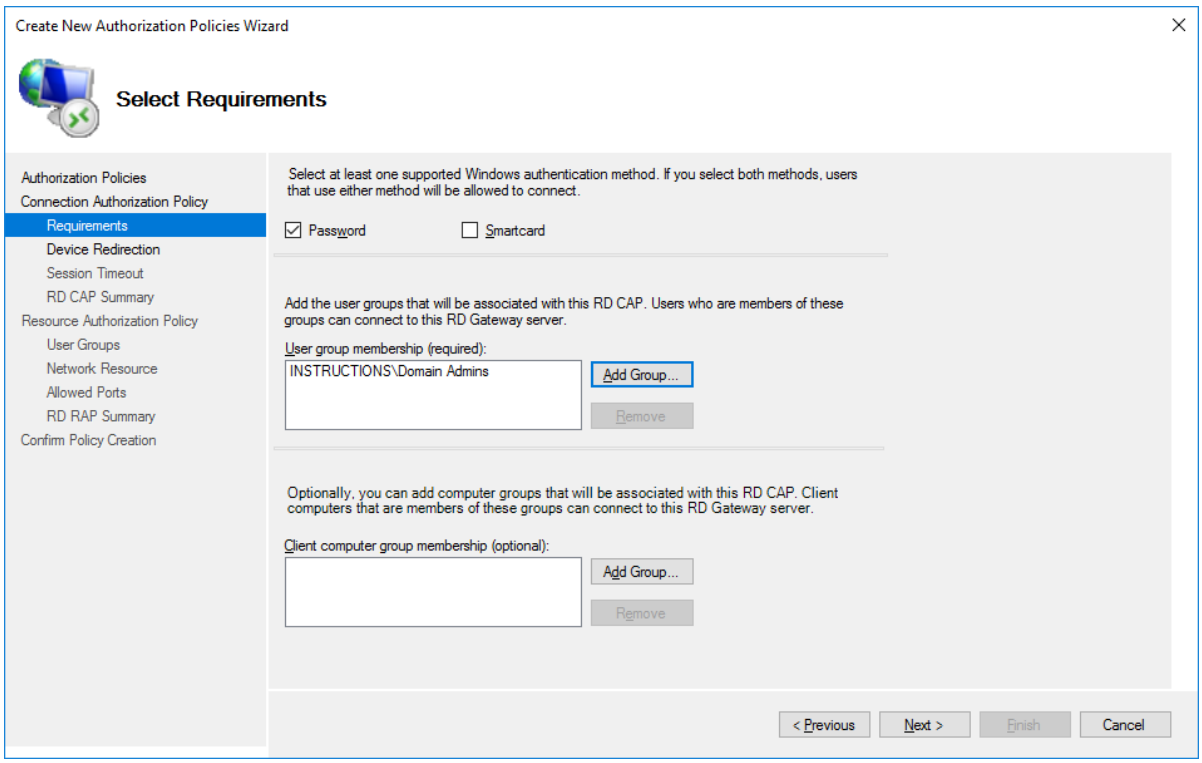
На следующем шаге выберите наиболее удобный метод аутентификации: пароль или smartcard. Далее добавьте группы пользователей которые смогут подключаться к этому RD Gateway серверу, для это нажмите **Add Group**.



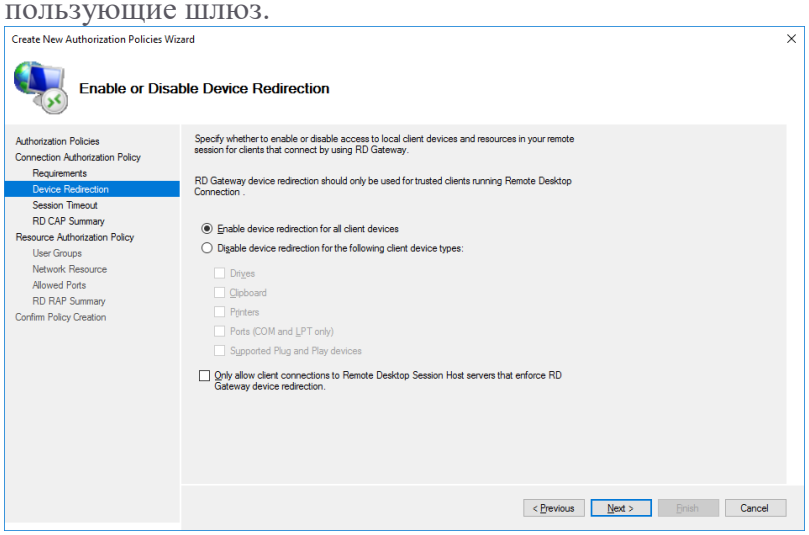
Выберите нужную группу, например администраторов домена или контроллеры домена. Выполнить поиск можно с помощью кнопки **Check Names**.



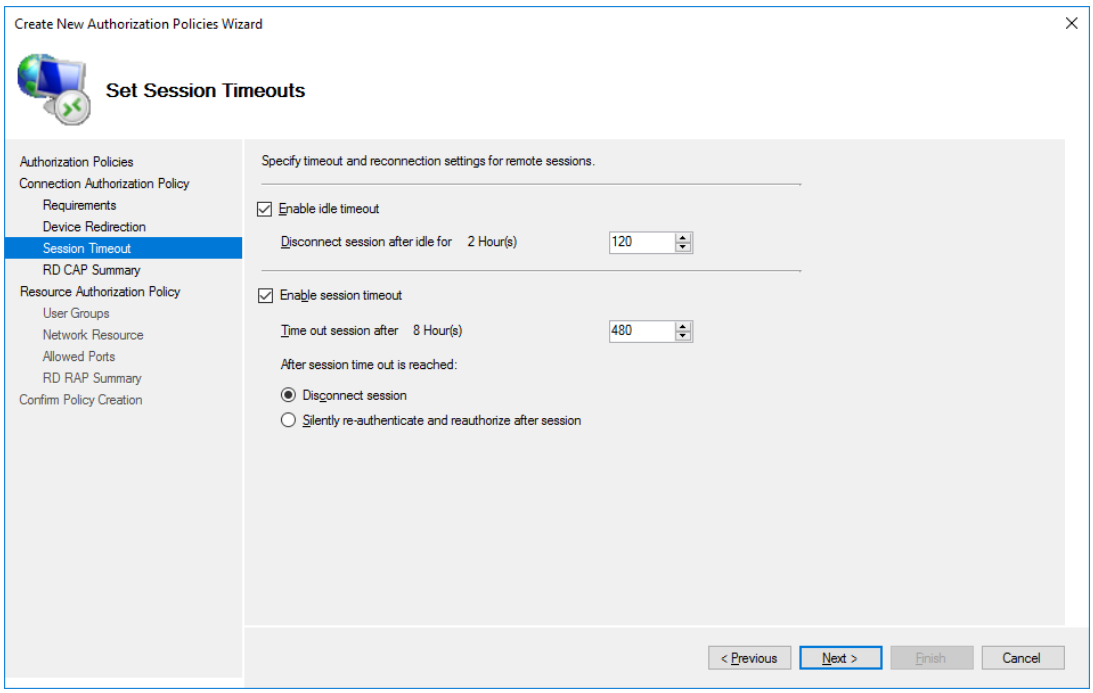
После добавления групп можно переходить к следующему действию.



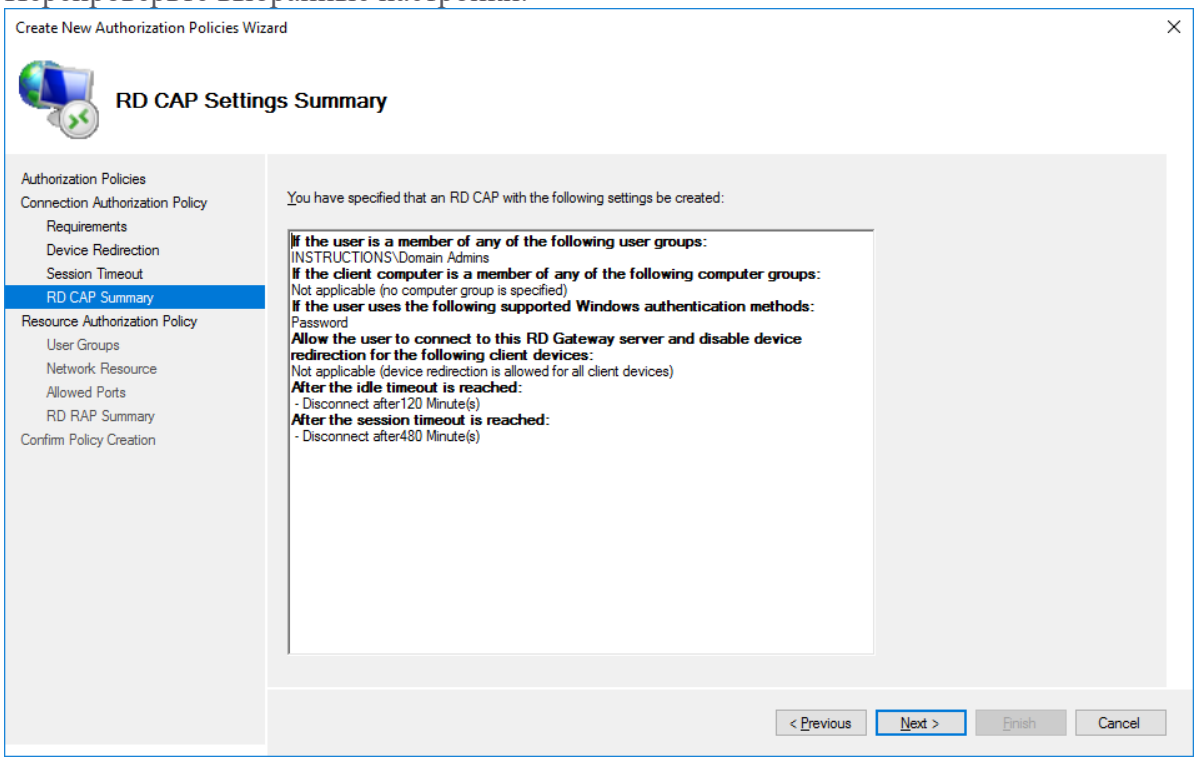
Выберите устройства и ресурсы удаленной сессии, которые будут доступны клиентам использующие шлюз.



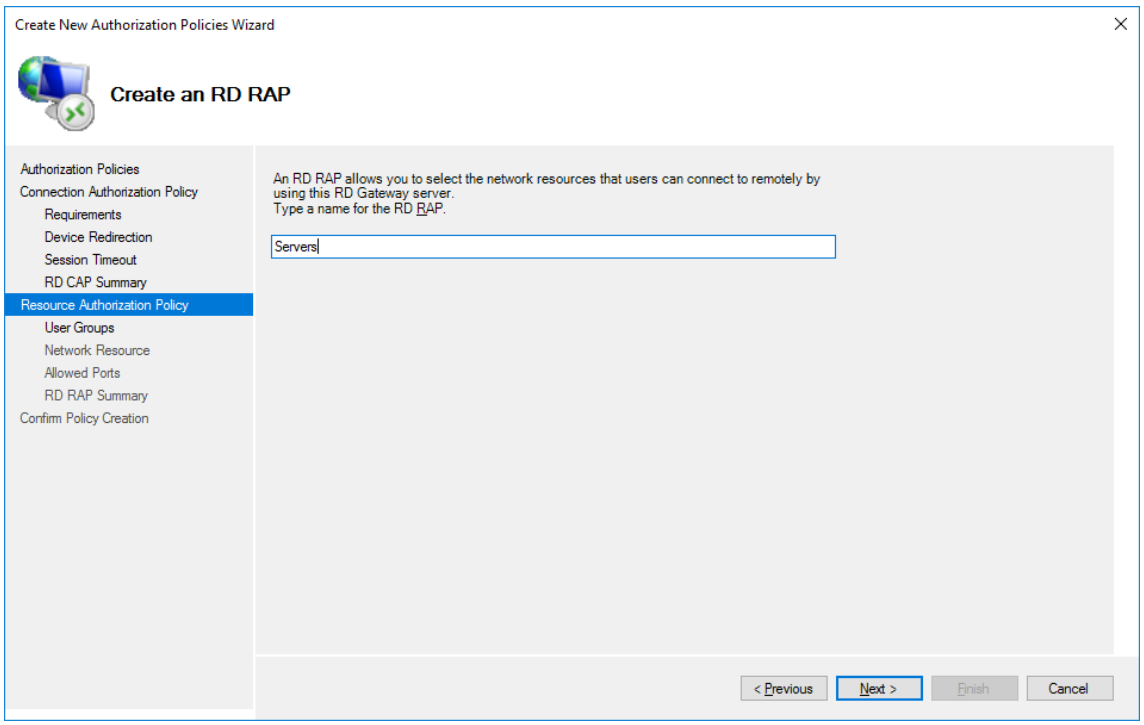
Выберите нужные для вас значения таймаутов: времени простоя и времени работы сессии.



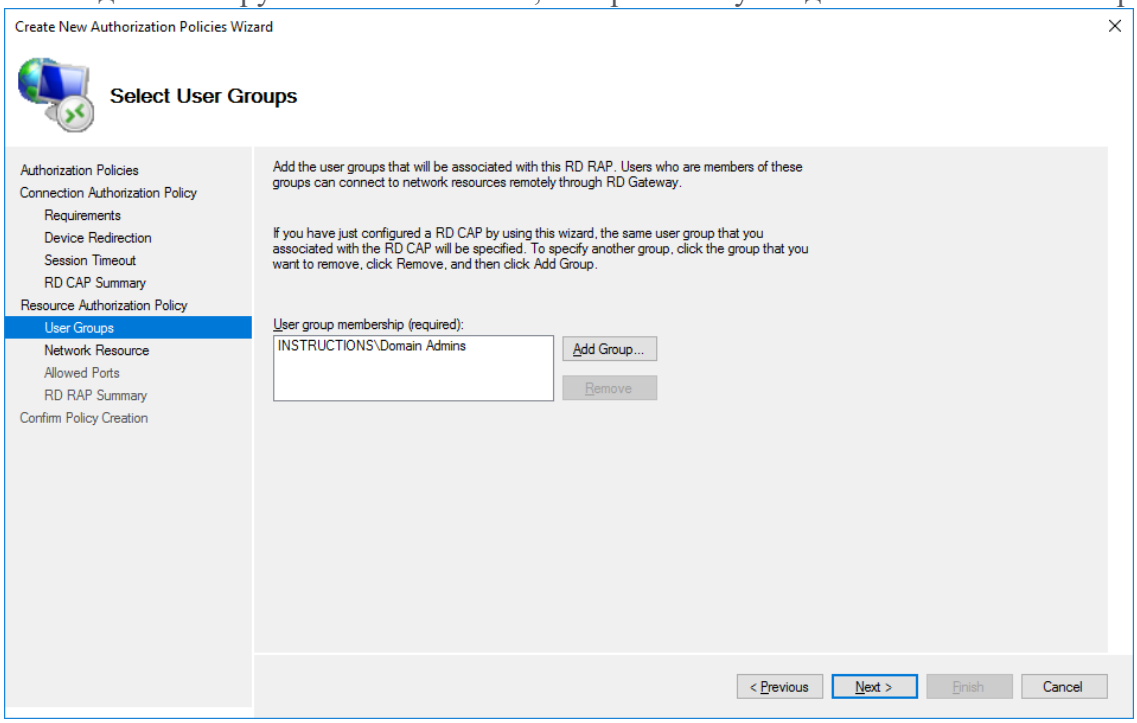
Перепроверьте выбранные настройки.



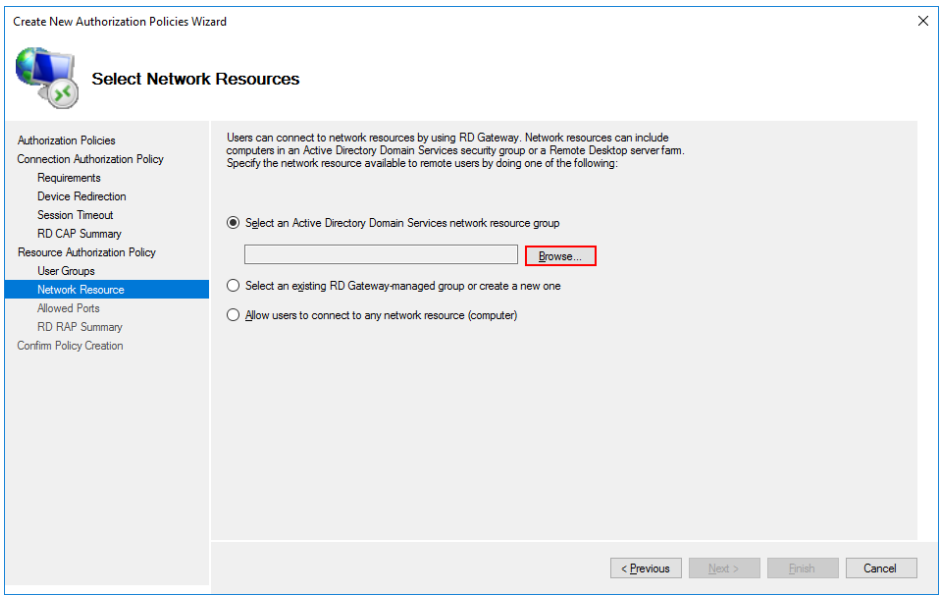
Далее вы перейдете к настройке политики авторизации ресурсов. Введите удобное имя политики.



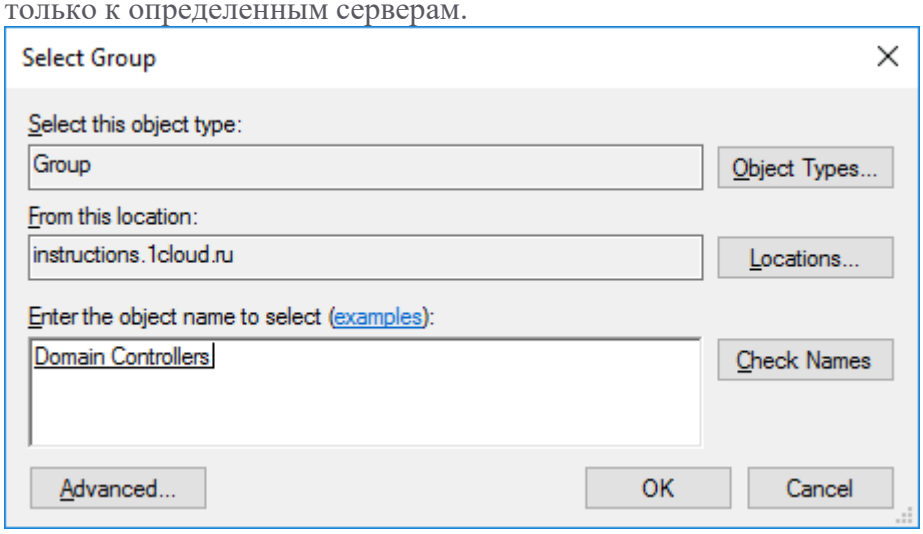
Также добавьте группы пользователей, которые смогут подключаться к сетевым ресурсам.



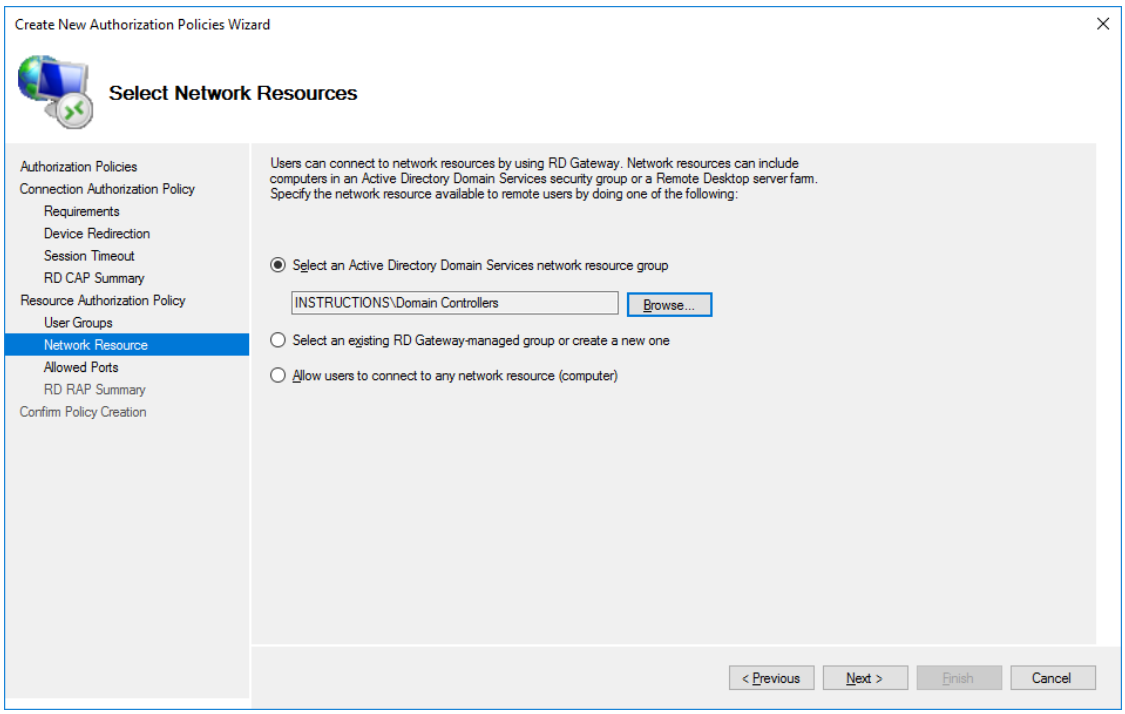
Выберите группу, содержащую серверы, на которых указанные группы пользователей могли бы работать с удаленным рабочим столом. Для этого нажмите **Browse**.



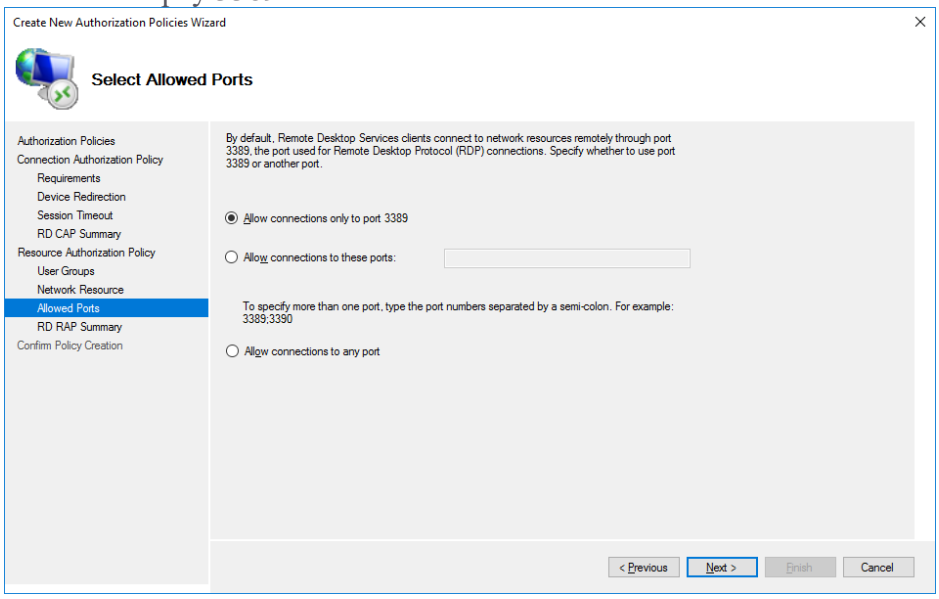
В этом примере используется встроенная группа под названием Domain Controllers. Вы можете создавать дополнительные группы, содержащие серверы, которые связаны или принадлежат к определенным отделам или сотрудникам. Таким образом, на предыдущих шагах вы можете назначать группы на основе потребностей пользователей и разрешать им доступ только к определенным серверам.



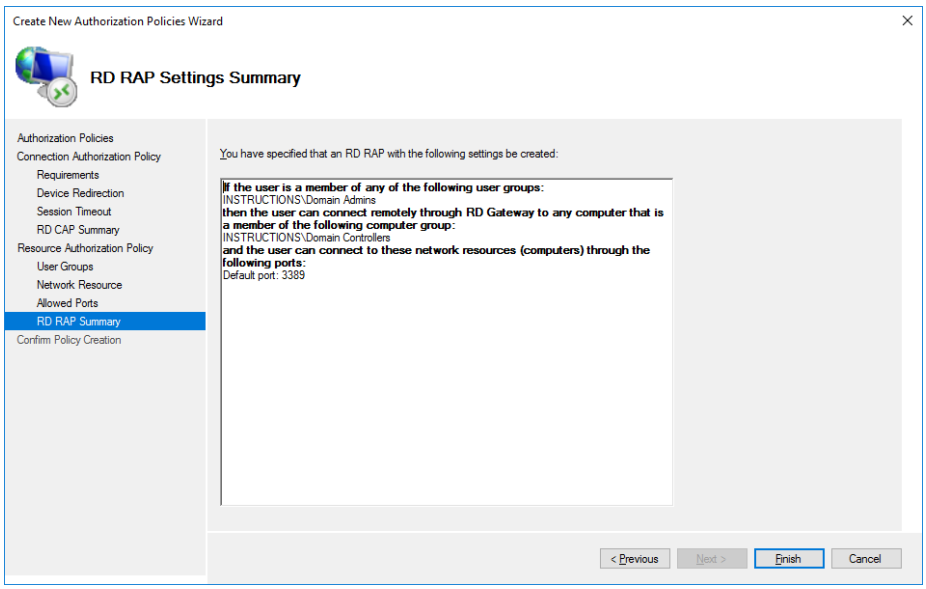
Убедитесь, что добавлена нужная группа.



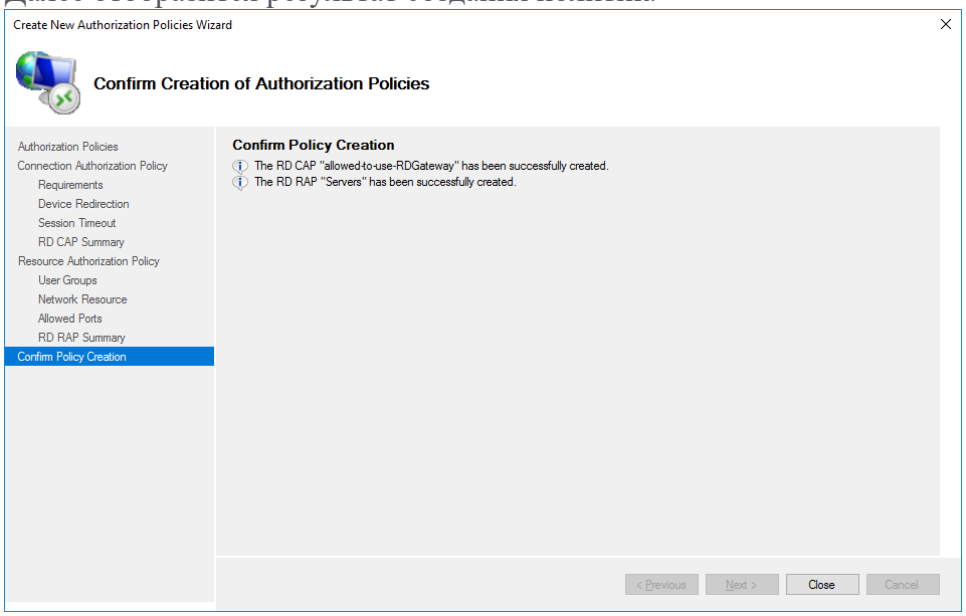
Если порт по умолчанию удаленного рабочего стола на серверах был изменен, используйте эту страницу для указания порта. В противном случае выберите разрешение подключения только к порту 3389.



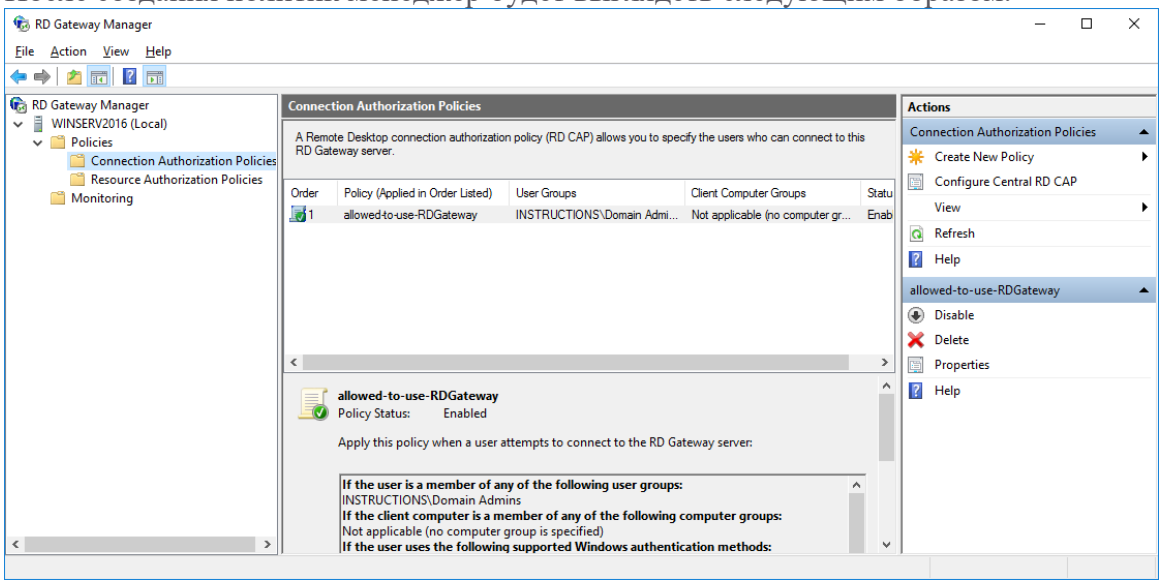
Проверьте указанные настройки для политики.



Далее отобразится результат создания политик.

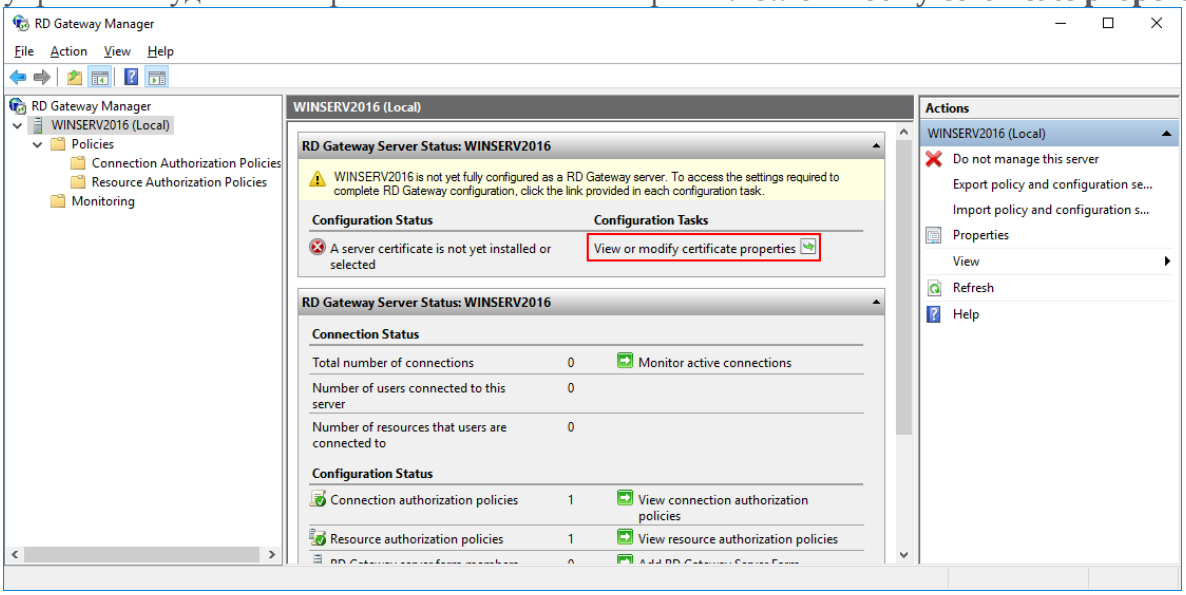


После создания политик менеджер будет выглядеть следующим образом.

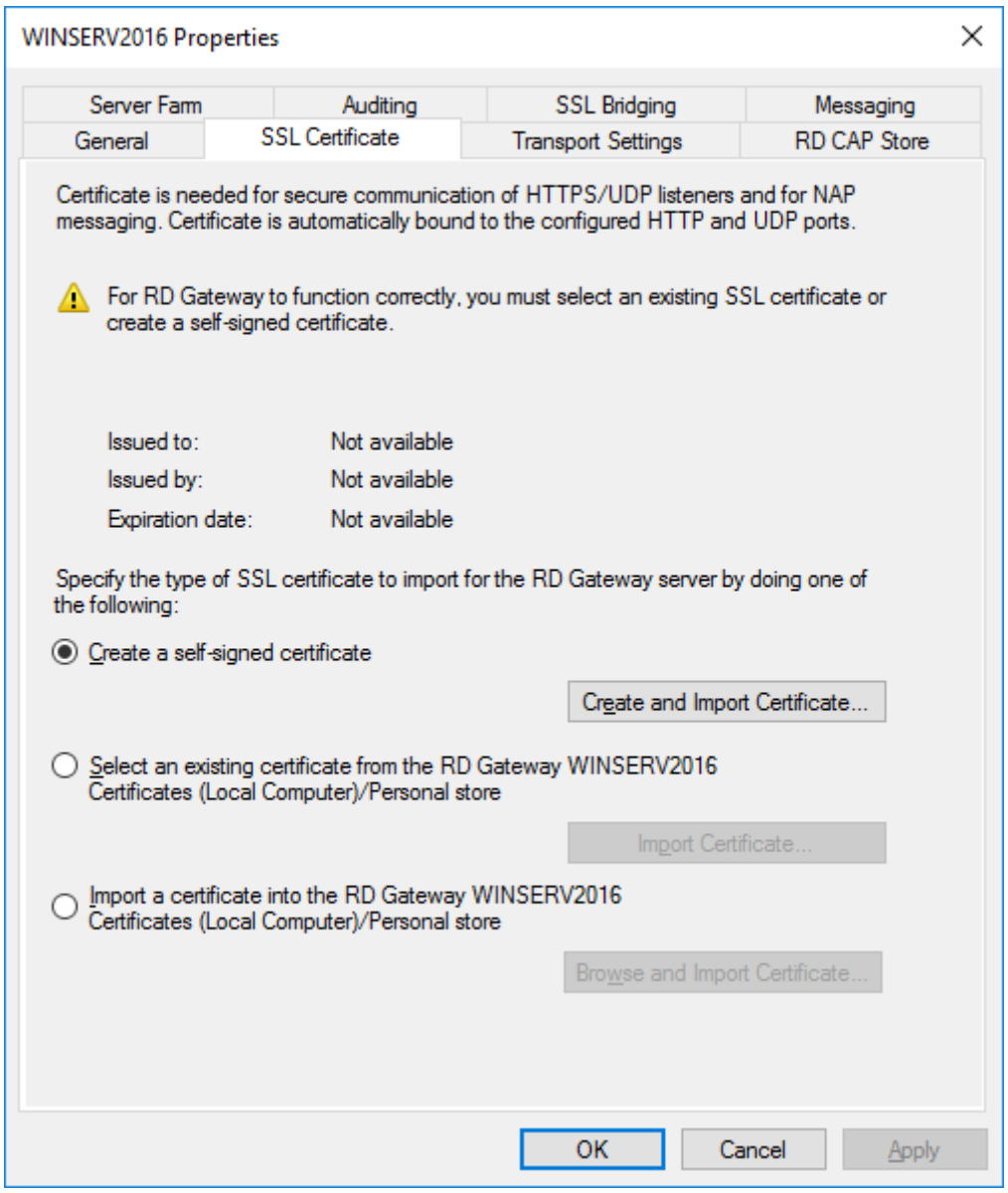


Установка SSL-сертификата

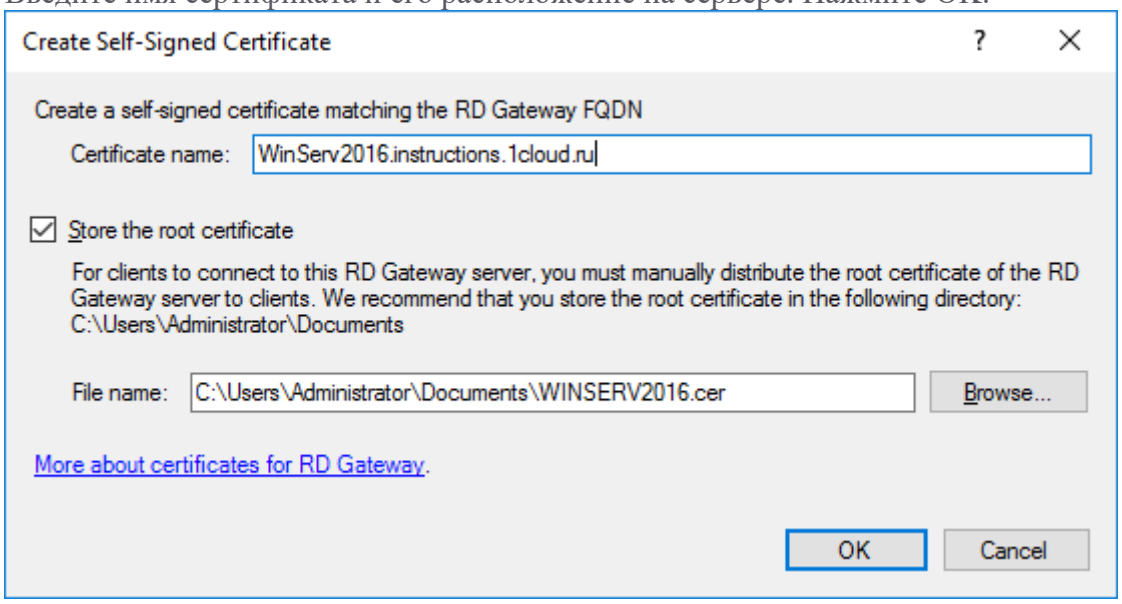
Для шлюза удаленного рабочего стола должен быть установлен SSL-сертификат. Чтобы установить SSL-сертификат, щелкните имя сервера удаленного рабочего стола в консоли управления удаленным рабочим столом и выберите **View or modify certificate properties**.



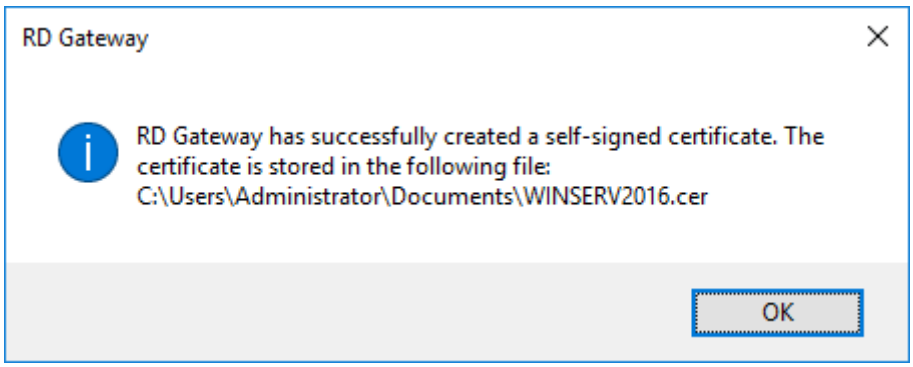
Возможно 3 способа импорта сертификатов:
создание самоподписанного сертификата и его импорт;
импорт ранее загруженного сертификата (самоподписанного или стороннего);
загрузка стороннего сертификата (например, Comodo) и его импорт;
Выберите подходящий вам способ, в нашем примере мы рассмотрим первый случай с генерацией и импортом самоподписанного сертификата.



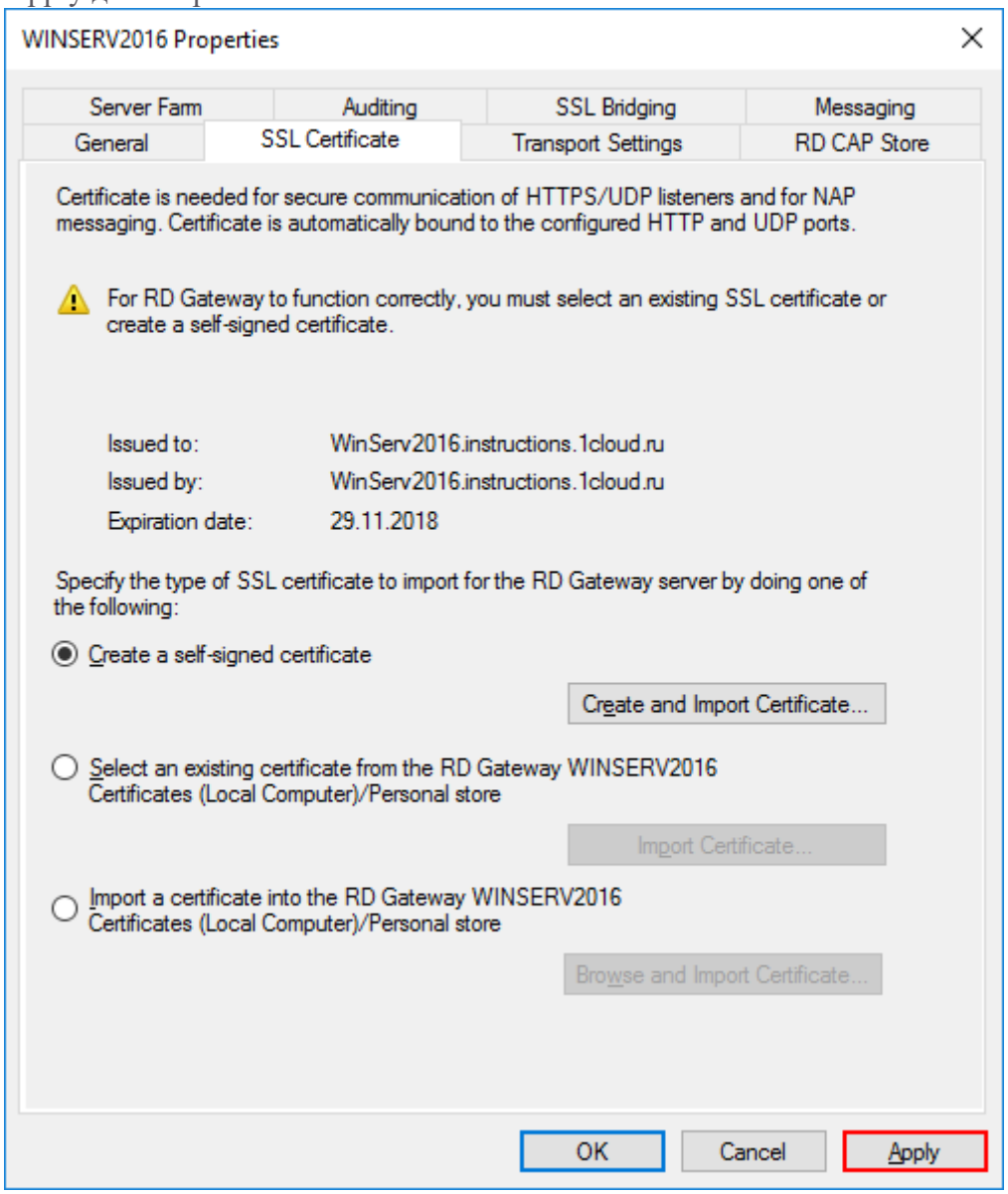
Введите имя сертификата и его расположение на сервере. Нажмите OK.



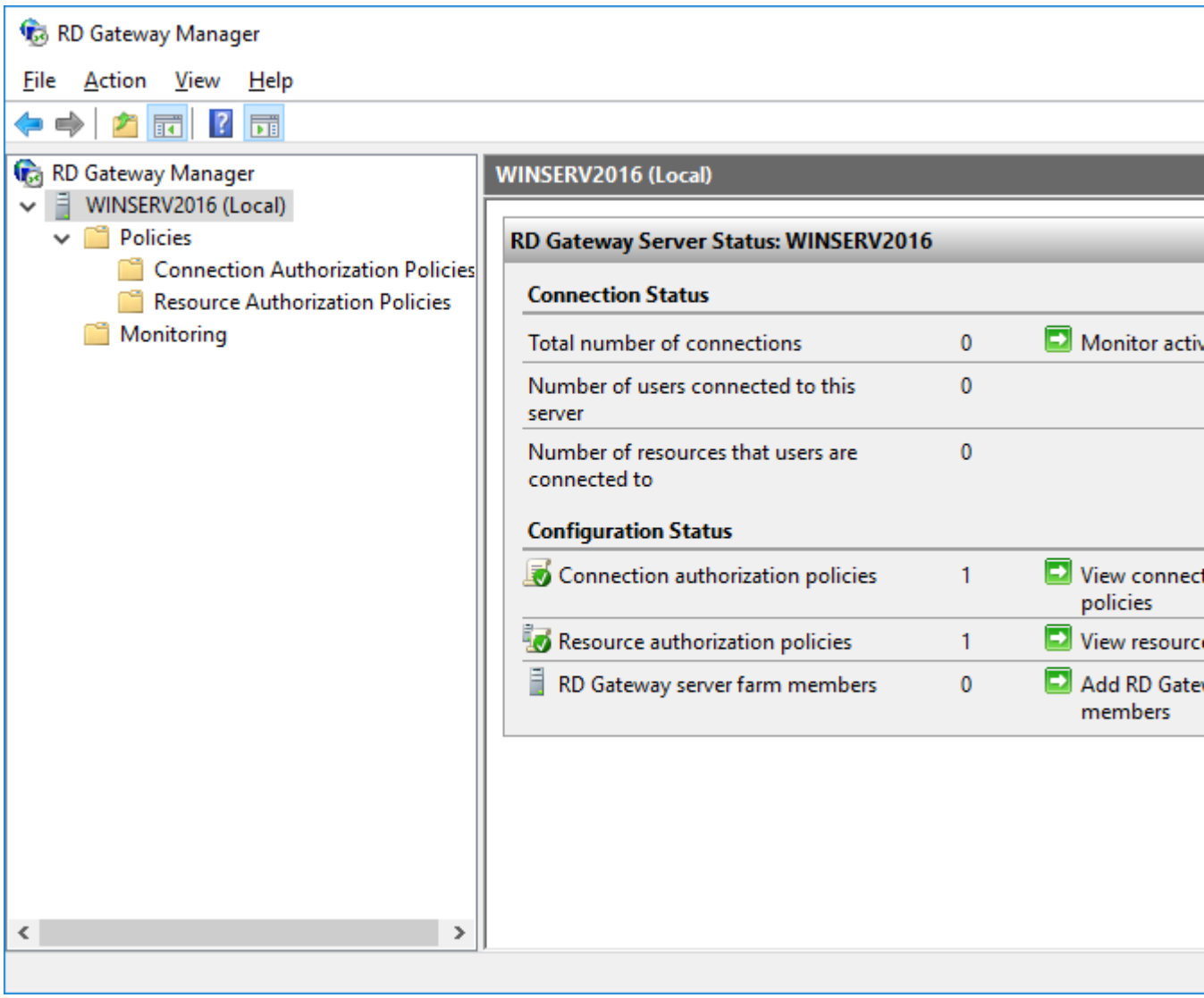
Сертификат будет сгенерирован.



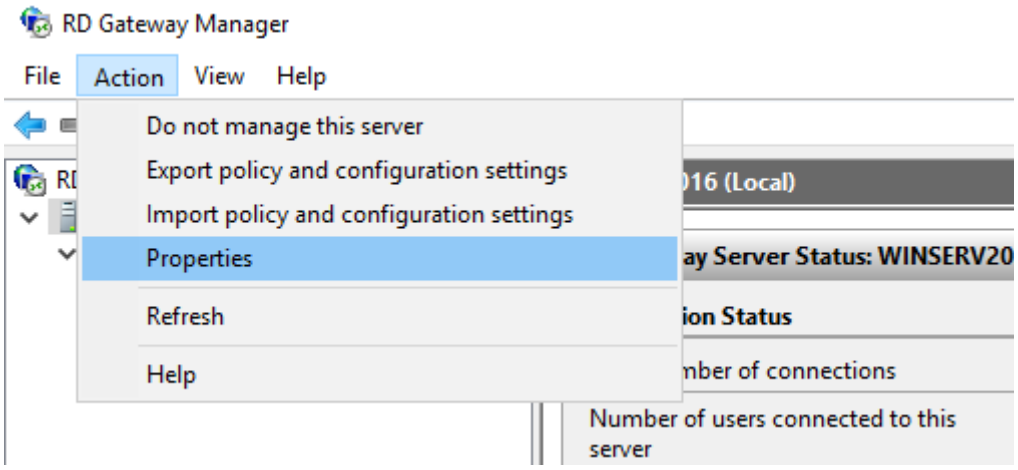
В результате отобразится - кому, кем и до какого числа выдан ssl-сертификат. Нажмите Apply для сохранения изменений.



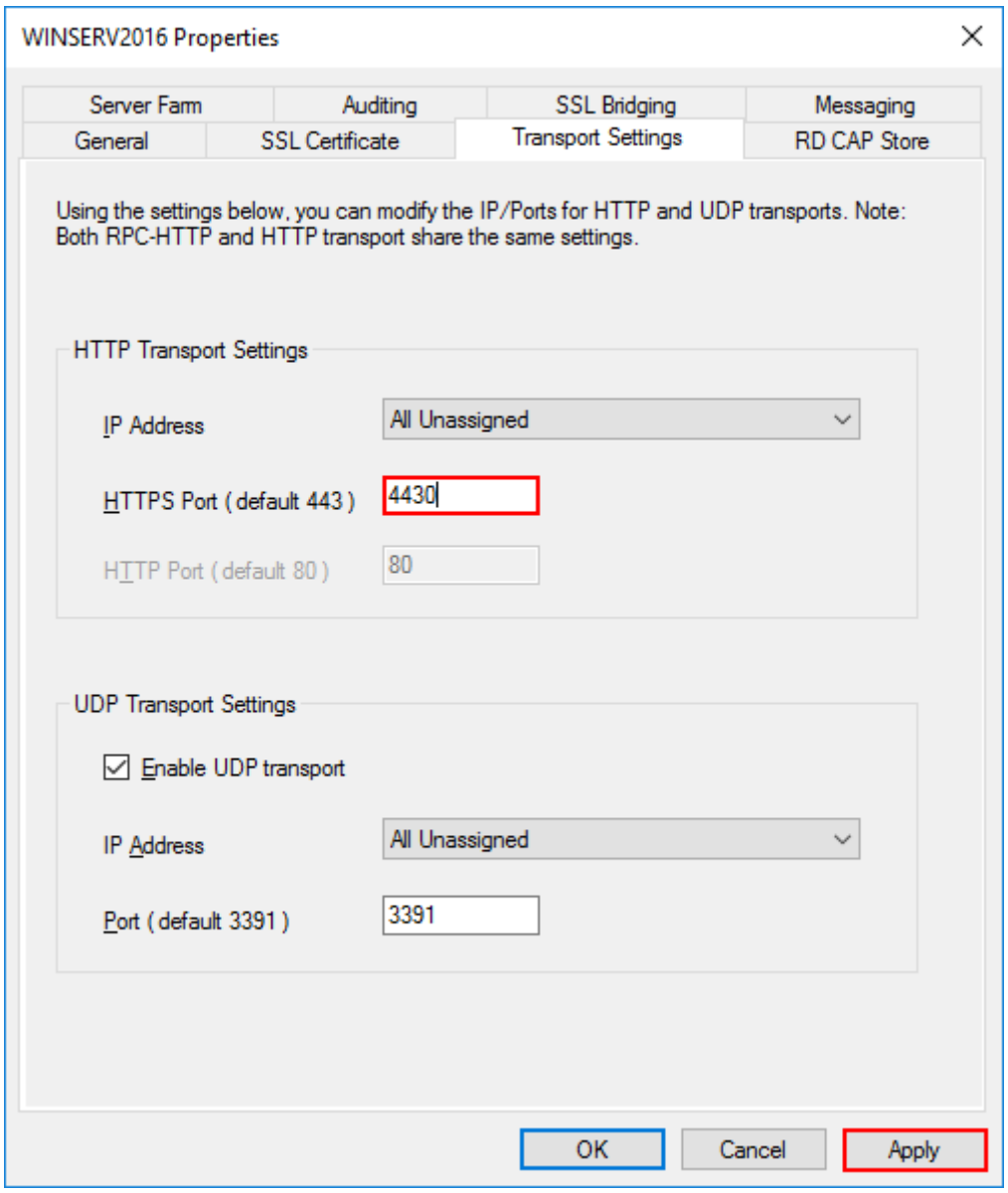
Теперь самоподписанный SSL-сертификат успешно установлен на TCP-порт 443 (порт SSL по умолчанию).



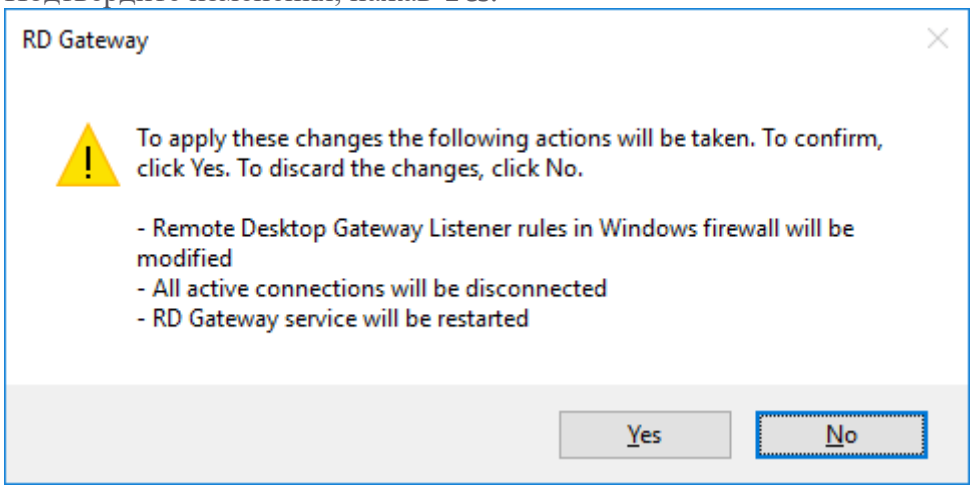
В целях безопасности рекомендуется изменить порт SSL для шлюза удаленных рабочих столов на другой номер. Обычно компании делают это, чтобы попытаться обмануть хакеров, которые могут ориентироваться на стандартный порт 443.
Чтобы изменить номер порта для шлюза RD, щелкните правой кнопкой мыши имя сервера и выберите свойства в консоли управления удаленным рабочим столом (Action →Properties).



Измените значение HTTP-порта на любое удобное значение и сохраните изменения.

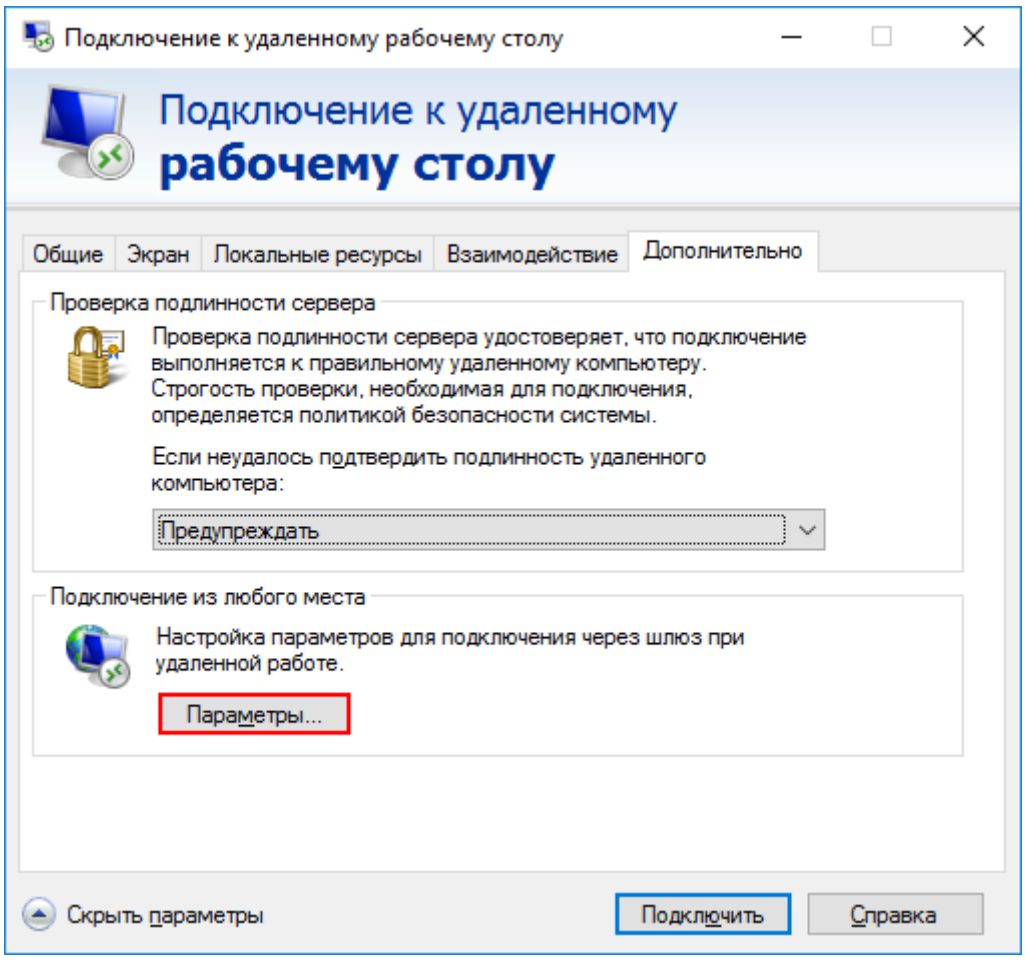


Подтвердите изменения, нажав **Yes**.

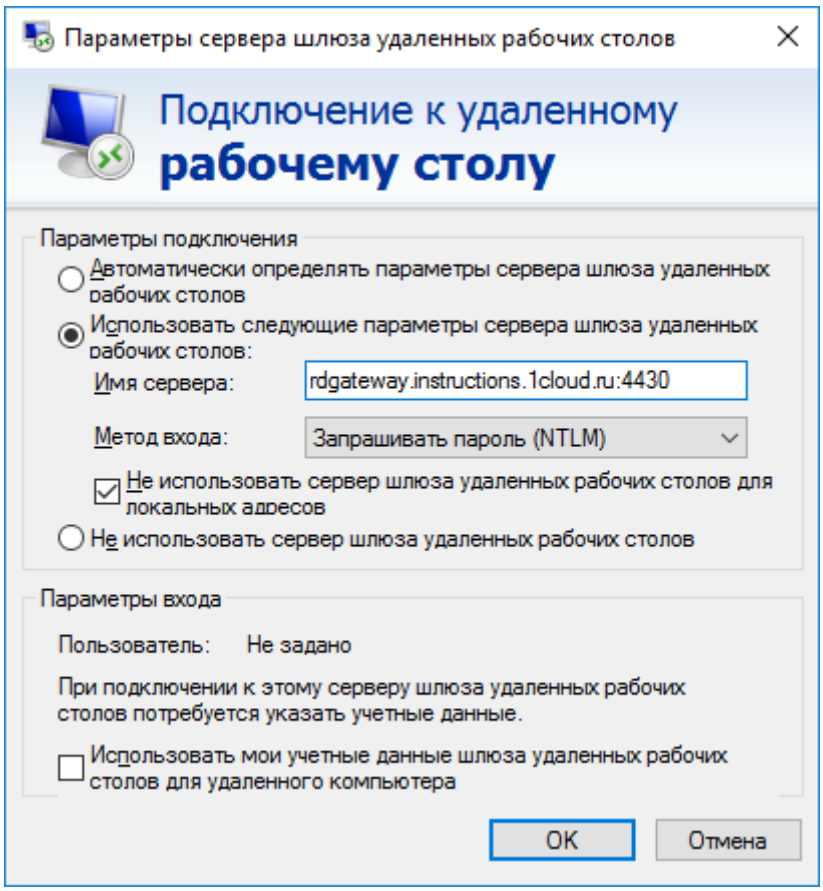


Подключение через шлюз

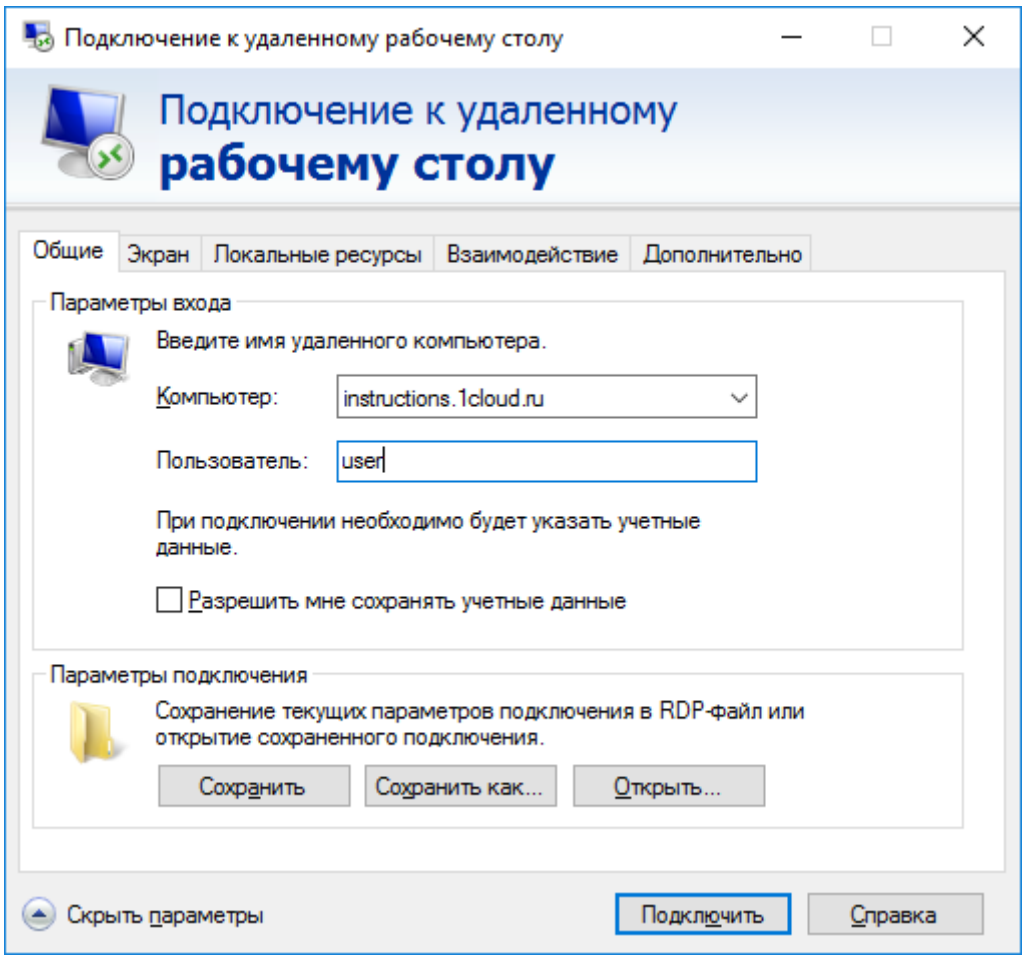
Для подключения откройте стандартное приложение Windows **Подключение к удаленному рабочему столу** (mstsc.exe). На вкладке **Дополнительно** нажмите на кнопку **Параметры**.



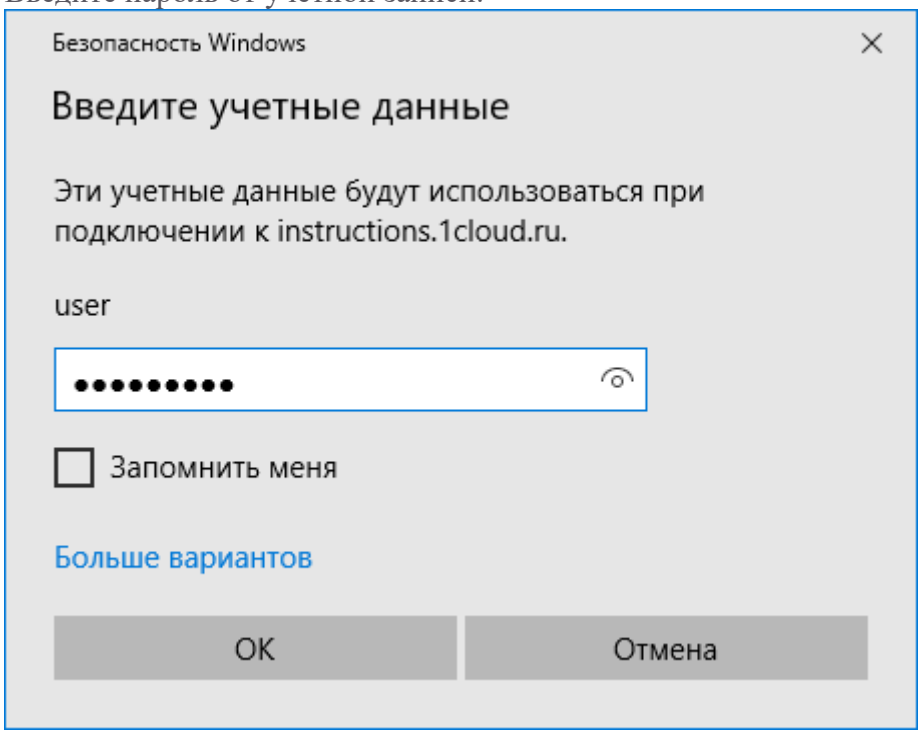
В открывшемся окне выберите Использовать следующие параметры сервера шлюза удаленных рабочих столов. Введите имя сервера в следующем формате и нажмите OK: rdgateway.<ваш домен>:<порт>



На вкладке **Общие** в поле **Компьютер** введите домен, в поле **Пользователь** имя пользователя и нажмите **Подключить**. При необходимости можете сохранить параметры входа.
Примечание: пользователь должен иметь права подключения через шлюз, которые были настроены ранее.



Введите пароль от учетной записи.



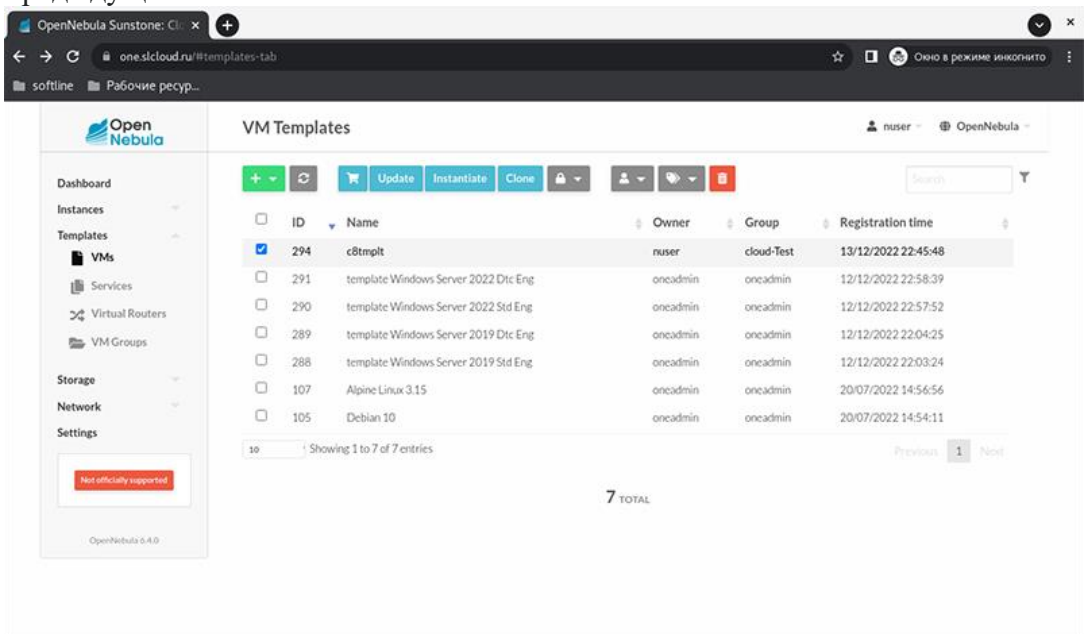
В результате будет произведено подключение к удаленному рабочему столу через шлюз RD Gateway. Это можно проверить с помощью команды tracert:

```
PS C:\Users\user> tracert 1cloud.ru

Tracing route to 1cloud.ru [5.200.50.90]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  5.200.47.1
  2     3 ms     2 ms     8 ms  5.200.46.254
  3    12 ms    <1 ms    <1 ms  fw-5-200-46-220.it-grad.ru [5.200.46.220]
  4     1 ms    <1 ms    <1 ms  5.200.50.90

Trace complete.
```

Практическая работа № 15 Работа с Облачными бизнес-моделями IaaS: Установка.

Задание:

На странице **Templates > VMs** в списке доступных шаблонов выберите созданный на предыдущем шаге и нажмите **Instantiate**.



Страница параметров ВМ

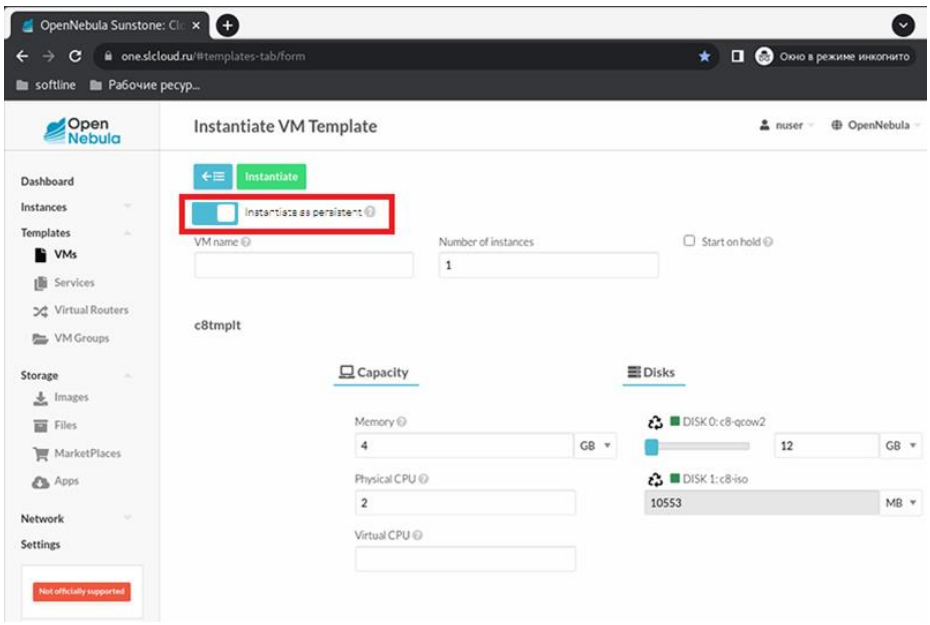
На странице создания ВМ **обязательно включите параметр «Instantiate as persistent».**

Заполните поля:

VM name – укажите имя ВМ.

Memory – укажите объём RAM.

Physical CPU, Virtual CPU – укажите количество CPU. Рекомендуется указывать значение Physical CPU равное Virtual CPU. Если вы хотите сэкономить ресурсы, то для тестовых и маловажных ВМ можно указать значение Physical CPU меньше, чем Virtual CPU.

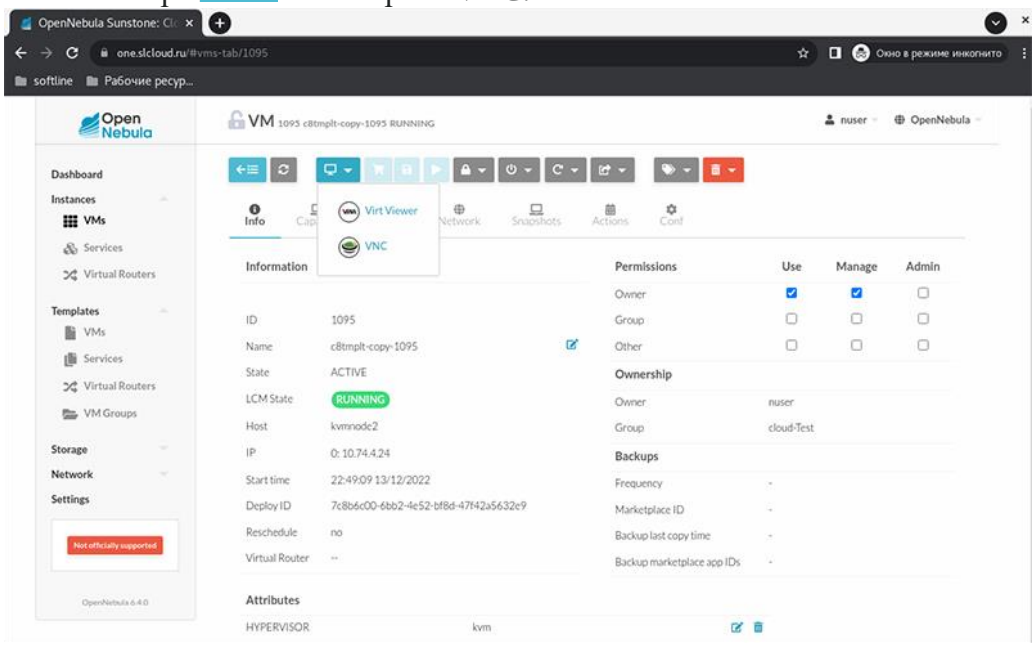


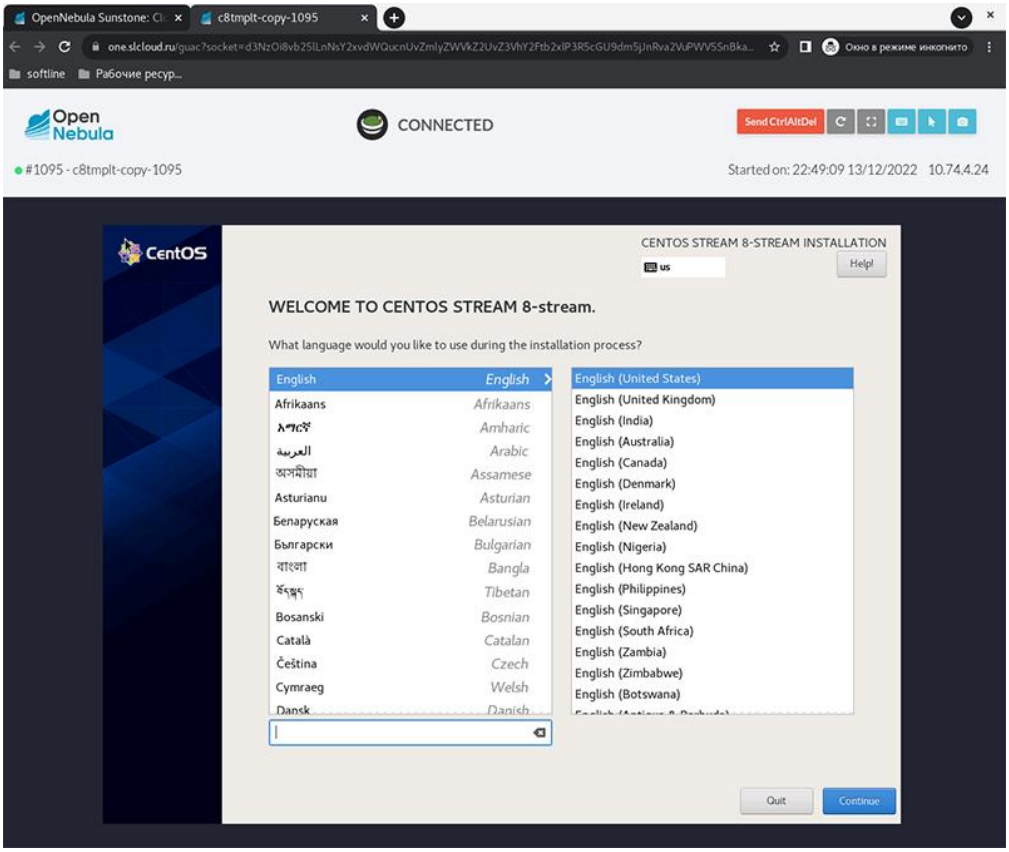
Нажмите Instantiate.

Практическая работа № 16 Работа с Облачными бизнес-моделями IaaS: Автоматизация. развёртывание виртуальной машины.

Задание:

Новую ВМ можно увидеть на странице Instances > VMs. Дождитесь статуса ВМ Running. Для установки ОС пользуйтесь консолью ВМ. Для этого нажмите значок монитора напротив имени ВМ или на странице ВМ нажмите значок монитора и выберите VNC.





Практическая работа № 17 Работа с Облачными бизнес-моделями IaaS: Балансировщик нагрузки виртуальных машин

Задание:

1. Выберите один из представленных подходов к балансировке нагрузки ВМ:

NO-Migrations (NOM) — это подход только локального перераспределения ресурсов CPU для виртуальных машин (без живой миграции).

Краткосрочное прогнозирование, Short-Term Detection (SHT-D), определяет состояние перегрузки, если фактические и прогнозируемые значения загрузки CPU в следующих двух временных интервалах превышают порог перегрузки. Кроме того, тот же алгоритм используется для определения состояний без перегрузки и недогрузки. Как ожидается, подход будет весьма чувствительным к коротким скачкам нагрузки.

Long-Term Detection (LT-D) основан на долгосрочных прогнозах использования CPU в следующих 7 контрольных интервалах.

Long-Term Probabilistic Detection (LT-PD) основан на долгосрочном прогнозировании использования CPU в следующих 7 контрольных интервалах, но также учитывает неопределенность прогнозирования с помощью моделирования плотности распределения ошибки предсказания.

Long-Term Decision Theory Detection (LT-DTD) — это LT-PD с применением теории принятия решений с целью минимизации издержек живой миграции.

Последний подход, называемый Local Regression Detection (LR-D), является подходом, используемым в смежной работе другими авторами. Его выбрали как репрезентативную современную технологию, поскольку она обеспечивает наилучшую производительность, по сравнению с другими методами, по результатам смежного исследования на эту тему.

2. Обоснуйте свой выбор.

3. Реализуйте балансировку нагрузки ВМ выбранным методом.