How to integrate Ruby on Rails application with Microsoft AD FS Single Sign On

Definitive Guide

Author: Maciej Arkit, maciej.arkit@gmail.com

Update: Sept 7, 2018

Introduction

What is this guide about?

If you need to configure AD FS instance on Azure and pair it with your Ruby on Rails application using Devise and OmniAuth gems, this document is for you.
Each section describes configuration step by step - you can follow these details and use tested solution.

Tutorial is divided into three modular sections:

- **Chapter 1:** Setup Windows 2012 Virtual Machine on Azure
- **Chapters 2, 3:** Setup and configure AD FS service and IIS server
- **Chapter 4:** Configure, deploy and test Ruby on Rails application

Sample code for Chapter 4 is available on GitHub:
https://github.com/maciej-arkit/RoR_integration_with_ADFS_SSO_example

This tutorial **is not** an introduction to Single Sign On or SAML.

Why you may need your own instance of AD FS?

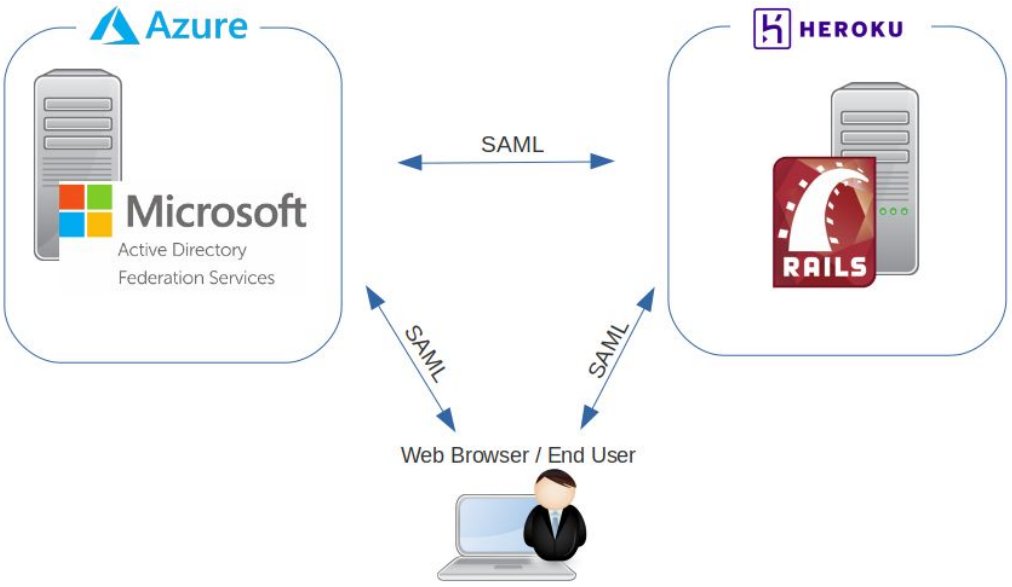
Microsoft AD FS is quite popular in many big institutions for authentication and authorization, often in conjunction with Active Directory. It may happen that your application will have to be integrated with AD FS service hosted by your customer.

Unfortunately, it is not always easy for the following reasons:

- no test environment provided by external company
- limited or no support from external company to help with the integration

Hence, setting your own AD FS instance may required. Own instance will help to execute initial integration, but it also is very useful later - ex. for troubleshooting and testing.

At the end of tutorial you will have configured full, end to end environment including Windows 2012 Virtual Machine with AD FS service running on it and example Ruby on Rails application deployed on Heroku and fully integrated with AD FS SSO.



Prerequisites

- Knowledge about SSO: what is SSO, how it works, what are the basic login/logout flows
- Average Ruby on Rails knowledge and general programming skills
- Basic knowledge: general computer science, computer networks, operating systems and HTTP protocol

3rd party providers used in this tutorial

MS Azure will be used to setup Windows 2012 VM and configure AD FS service on it.
Heroku will be used to deploy Ruby on Rails application.

*Please note, that you can replace **Azure** with any other hosting solution which provides Windows 2012 machines. Same for **Heroku** - you can replace it with any other provider you wish - ex. You can deploy RoR application on Google Compute Engine or Amazon EC2.*

1. Setup Windows 2012 Virtual Machine

For the purpose of this tutorial we will setup Windows 2012R2 Virtual Machine on MS Azure (you can use any other VM provider or even physical machine if you wish).
Windows machine is required to install and configure AD FS service.

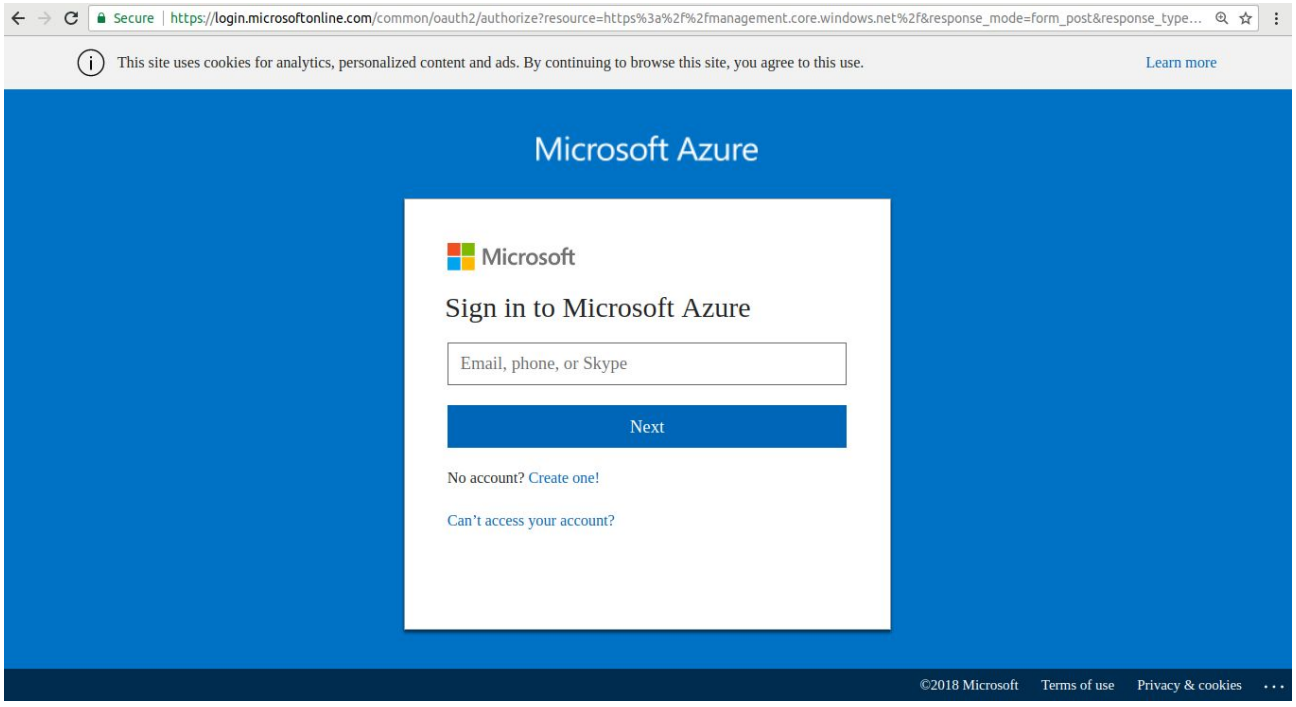
Prerequisites for Windows VM

- Public IP address
- DNS address. Machine have to be available via DNS ex. https://adfstutorial.westeurope.cloudapp.azure.com

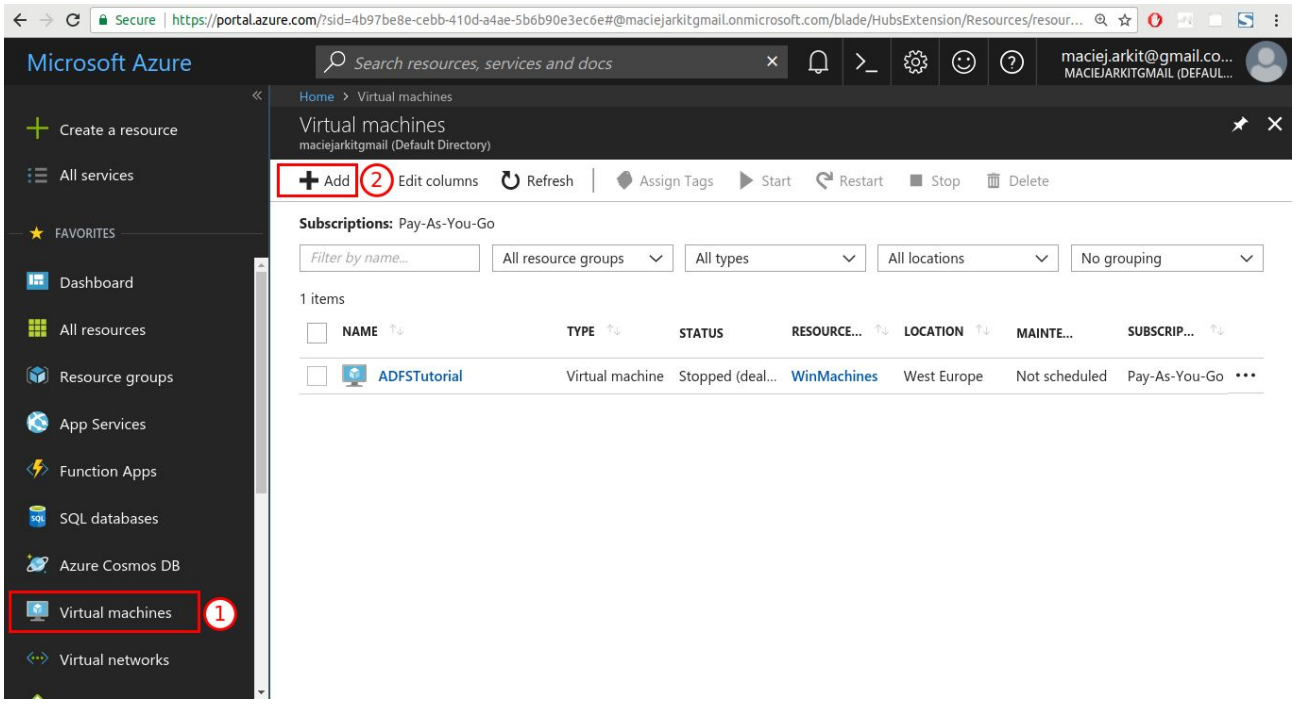
If you have already Windows 2012 VM or standalone machine, which satisfies mentioned prerequisites, you can skip this step, and move directly to “2. Setup and Configure AD FS service”.

1.1 Step by step

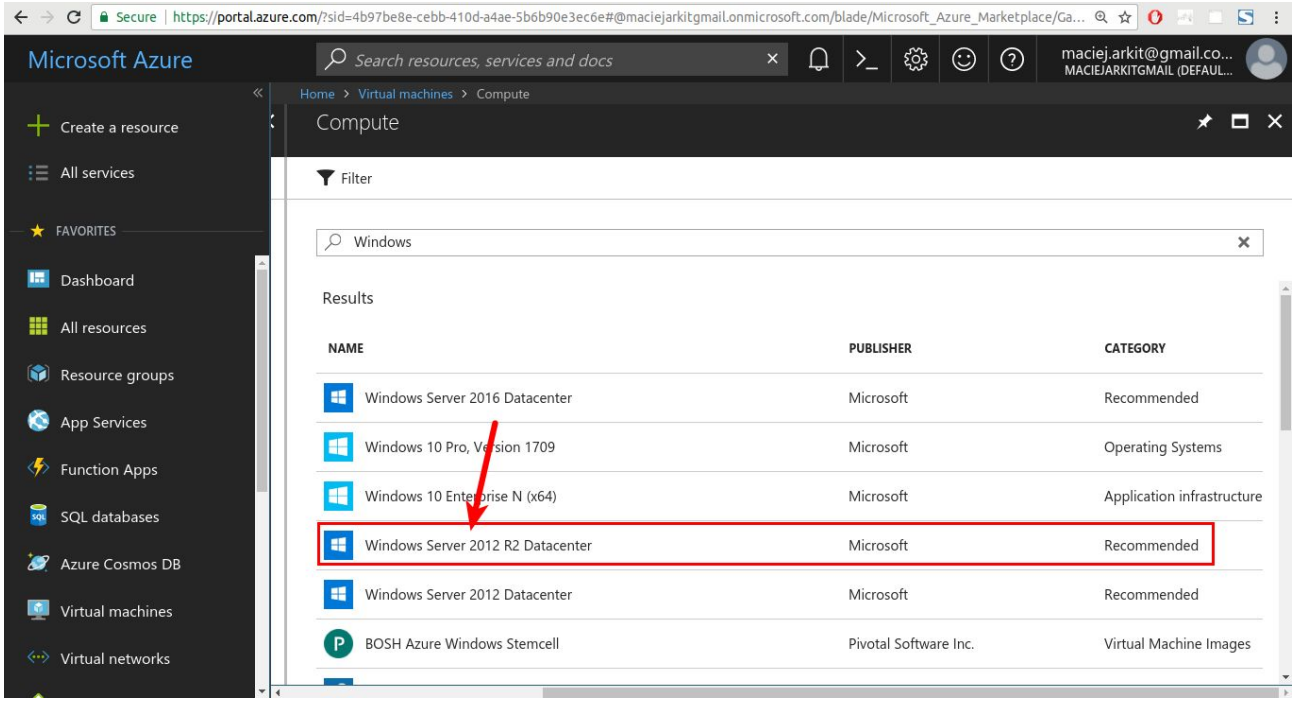
1. Go to https://portal.azure.com and sign in with your account.
   If you don’t have one, please follow instructions provided by Azure.



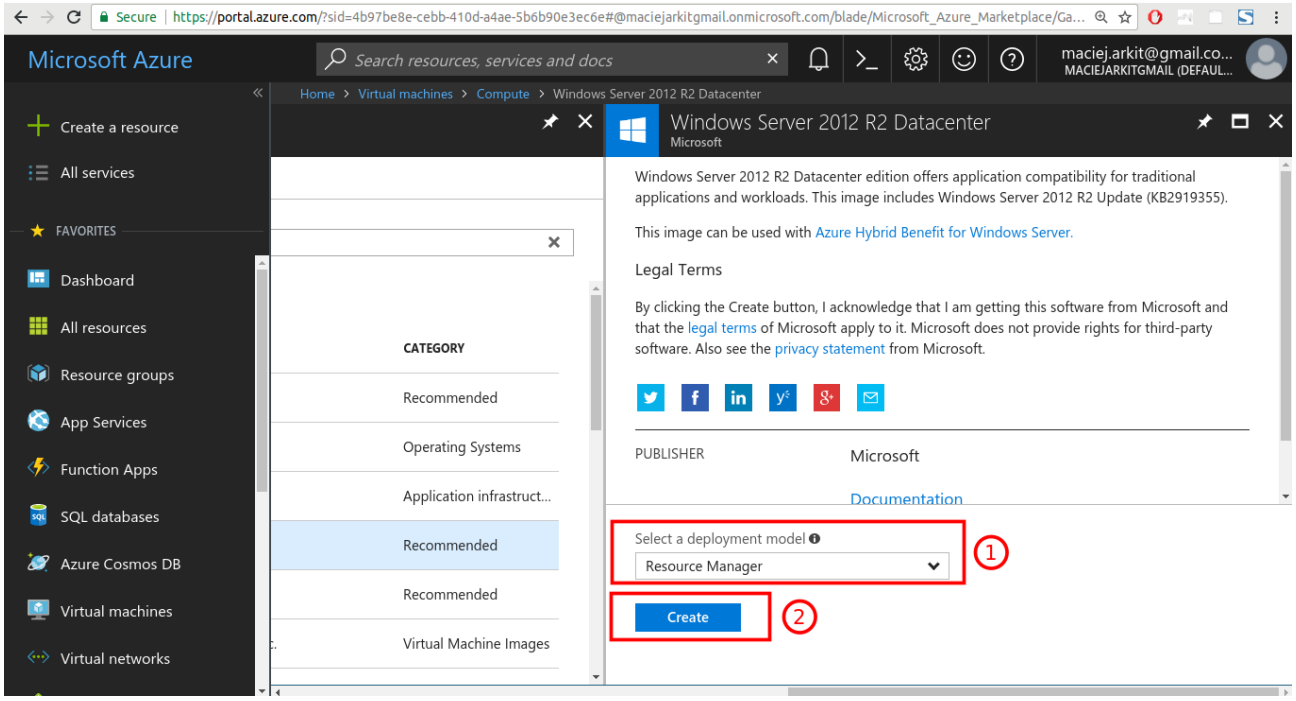
2. After logging in, you will see Azure “Dashboard”
3. Please click on “Virtual Machines” **(1)** section on the side menu
4. Please click “Add” button **(2)**



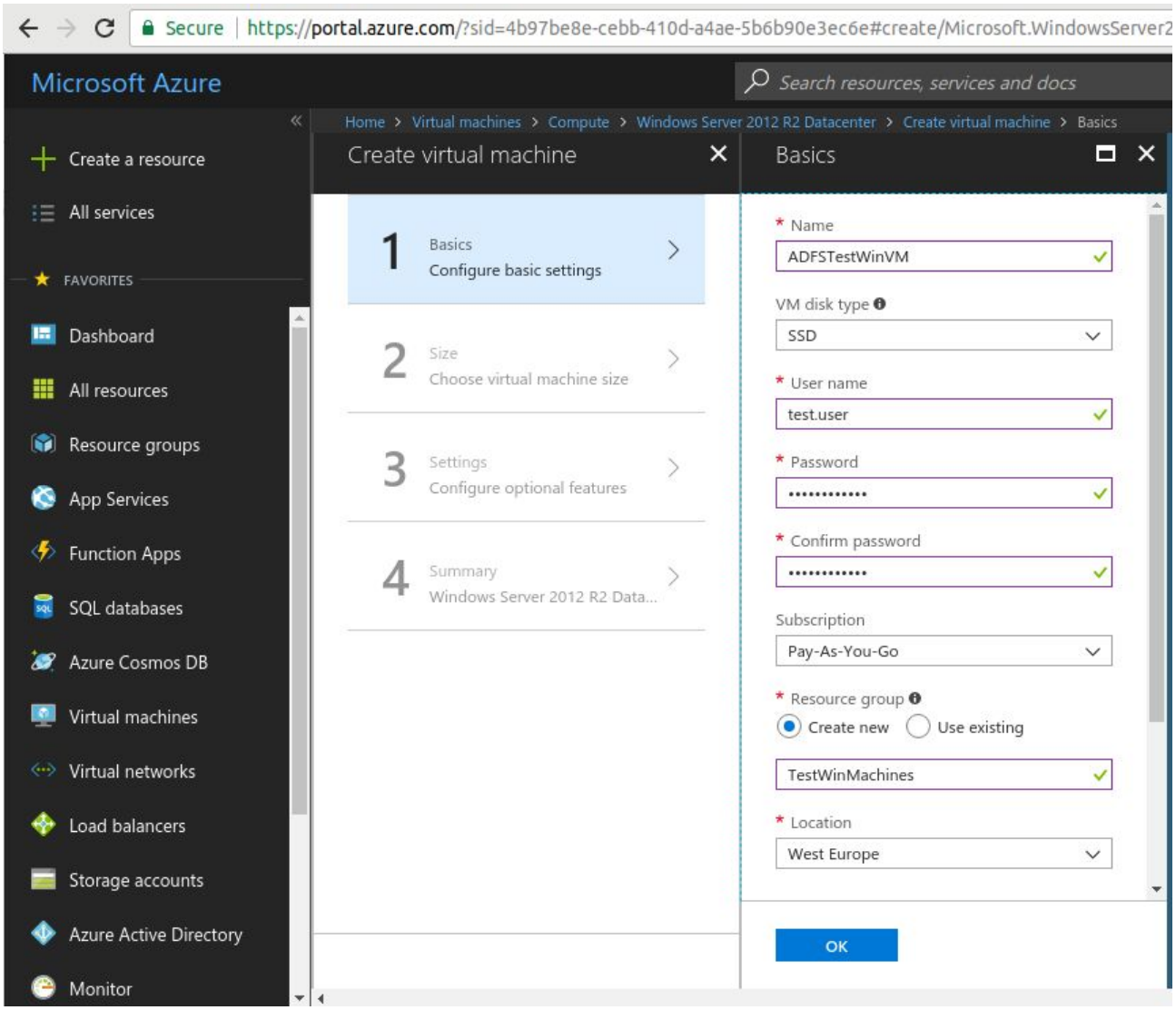
5. In the search input, type: **“Windows”**
6. From the “Results” list, please choose **“Windows Server 2012 R2 Datacenter”**



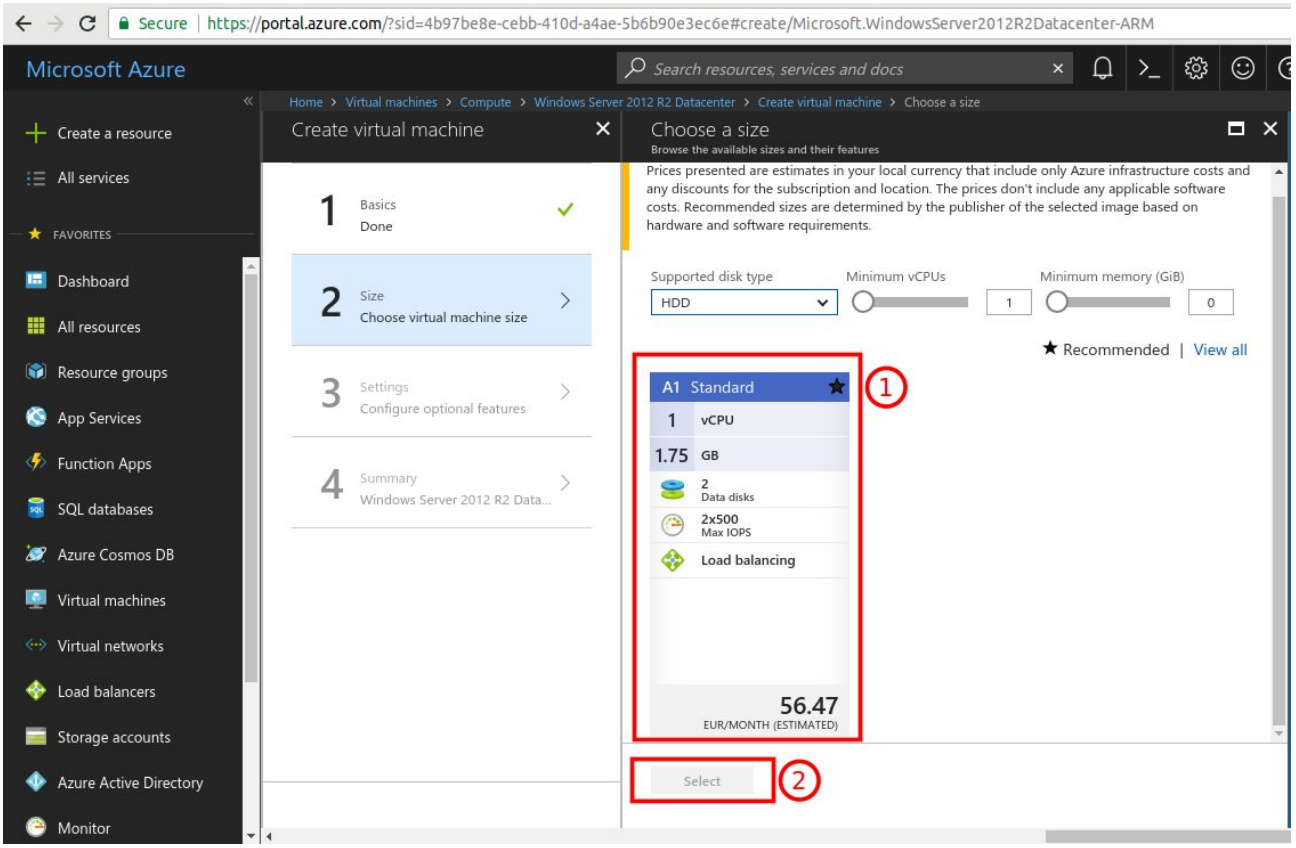
7. After selecting virtual machine, please select **“Resource Manager”** in **“Select a deployment model” (1)** and click **“Create”. (2)**



8. **NOTE:** At this point, if you don't have an existing subscription Azure will ask you to **"Sign up for a new subscription"** - please follow steps suggested by Azure.
   To start I recommend to subscribe **"Pay-as-You-Go"** subscription **without technical support**.
   During the subscription process you will be asked to provide your personal data and your credit card details for billing purposes.
   PRICING: If you would like to compute estimated cost of running VM please use pricing calculator: https://azure.microsoft.com/pl-pl/pricing/calculator/
9. Once you're done with your subscription, you can go back to VM configuration and provide details for virtual machine.
10. **Step 1: Basics** - provide basic details for your VM.
    At this step you will provide username and password for Windows user. **Remember it** so you can login to VM.
    HINT: If you already have a resource group, it is cheaper to reuse it.
    Be sure that group which you want to reuse is not dedicated for critical applications. Otherwise your test VM may impact performance of more important VMs.


Secure | https://portal.azure.com/?sid=4b97be8e-cebb-410d-a4ae-5b6b90e3ec6e#create/Microsoft.WindowsServer2
Microsoft Azure
Search resources, services and docs
Create a resource
All services
FAVORITES
Dashboard
All resources
Resource groups
App Services
Function Apps
SQL databases
Azure Cosmos DB
Virtual machines
Virtual networks
Load balancers
Storage accounts
Azure Active Directory
Monitor
Home > Virtual machines > Compute > Windows Server 2012 R2 Datacenter > Create virtual machine > Basics
Create virtual machine
1 Basics
Configure basic settings
2 Size
Choose virtual machine size
3 Settings
Configure optional features
4 Summary
Windows Server 2012 R2 Data...
Basics
* Name
ADFSTestWinVM
VM disk type
SSD
* User name
test.user
* Password
* Confirm password
Subscription
Pay-As-You-Go
* Resource group
Create new
Use existing
TestWinMachines
* Location
West Europe
OK

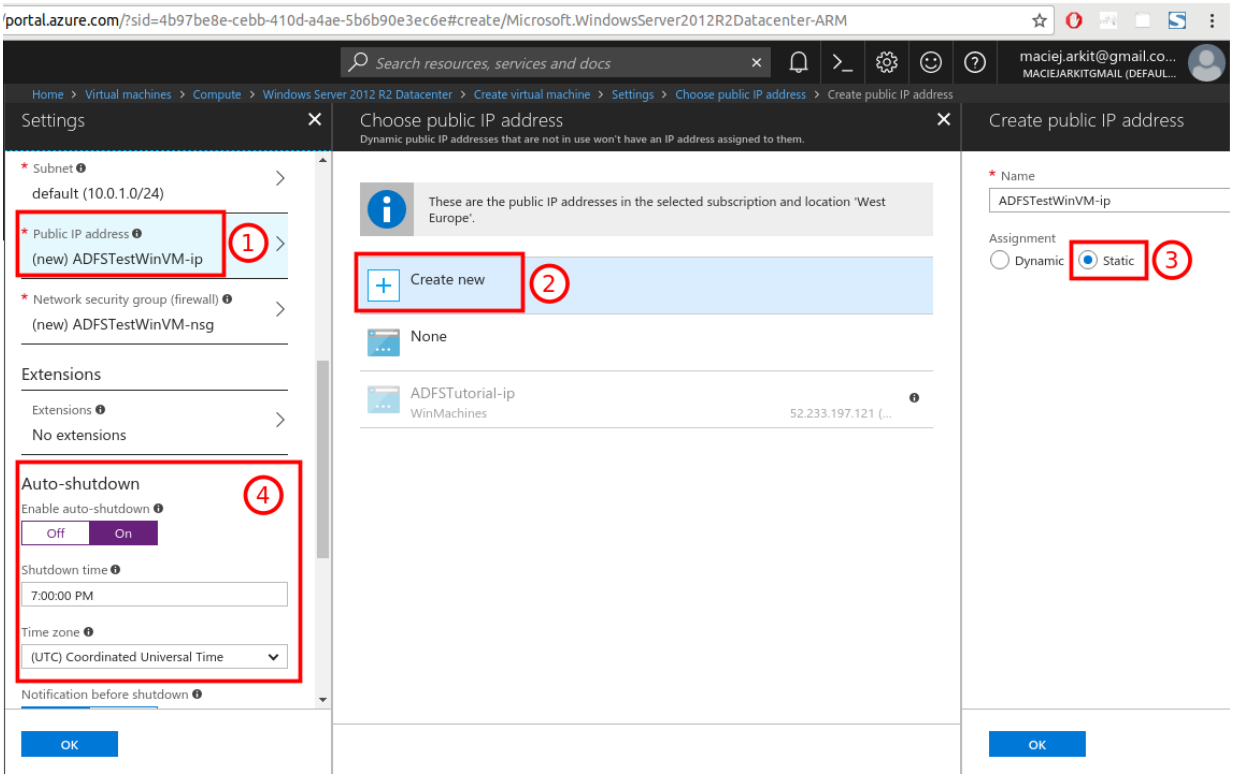
11. **Step 2: Choose VM size**. For the purpose of this tutorial you can choose minimal configuration. Please select **“A1 Standard” (1)** and click **“Select” (2)** button.



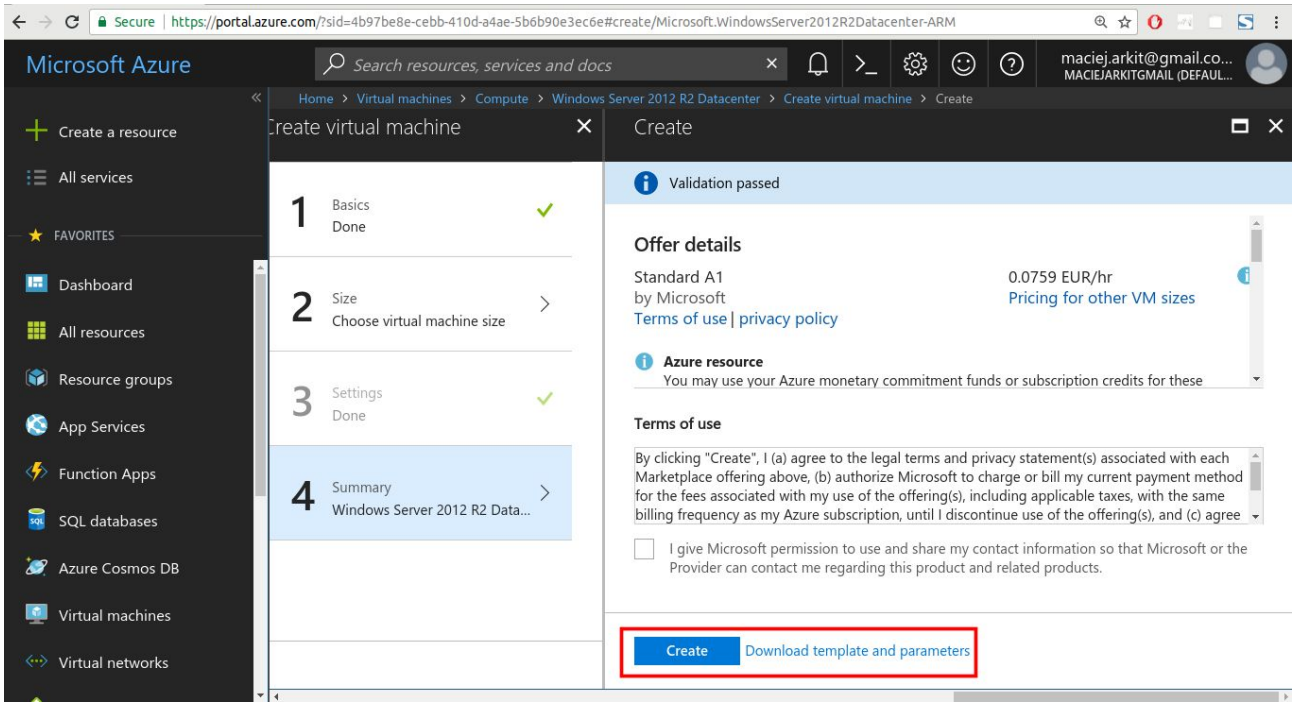
12. **Step 3 Settings**:
    Please click on **“Public IP address” (1)** and please select **“Create new” (2)** and **“Static” (3)** (Static IP address is preferred option for test VM. If you aim to use this VM for other purposes you can leave “Dynamic”).

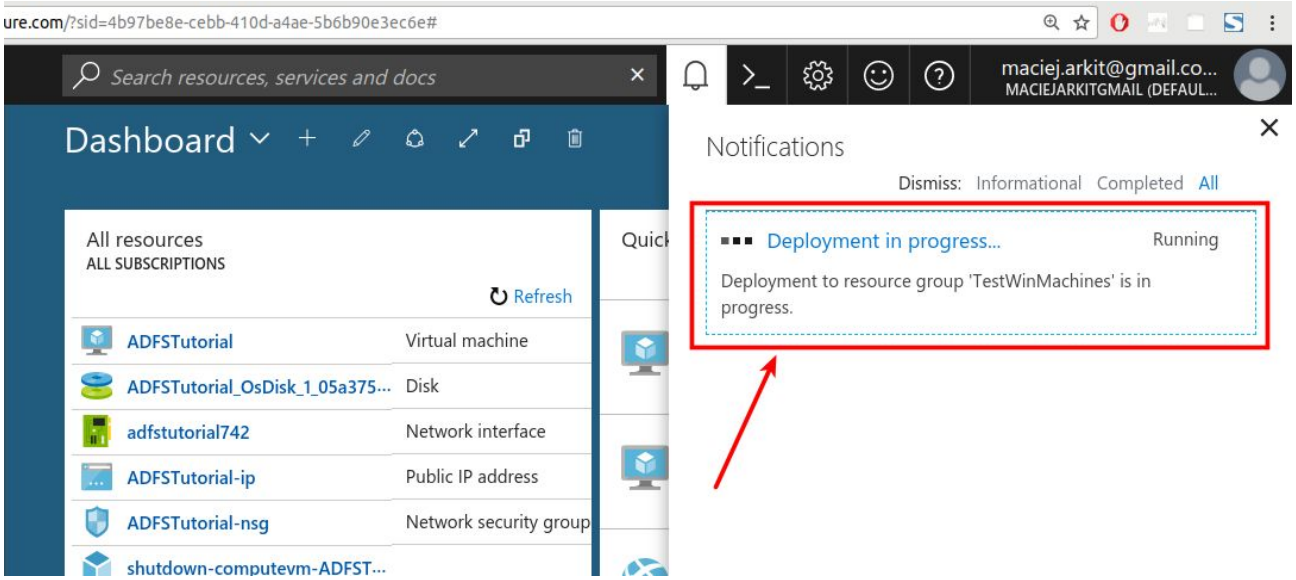
    If you don’t aim to use this VM frequently, I recommend to setup automatic shutdown at certain time of the day **(4)**. It will significantly reduce monthly cost. When you’re done with

“Settings”, please click **“OK”**.



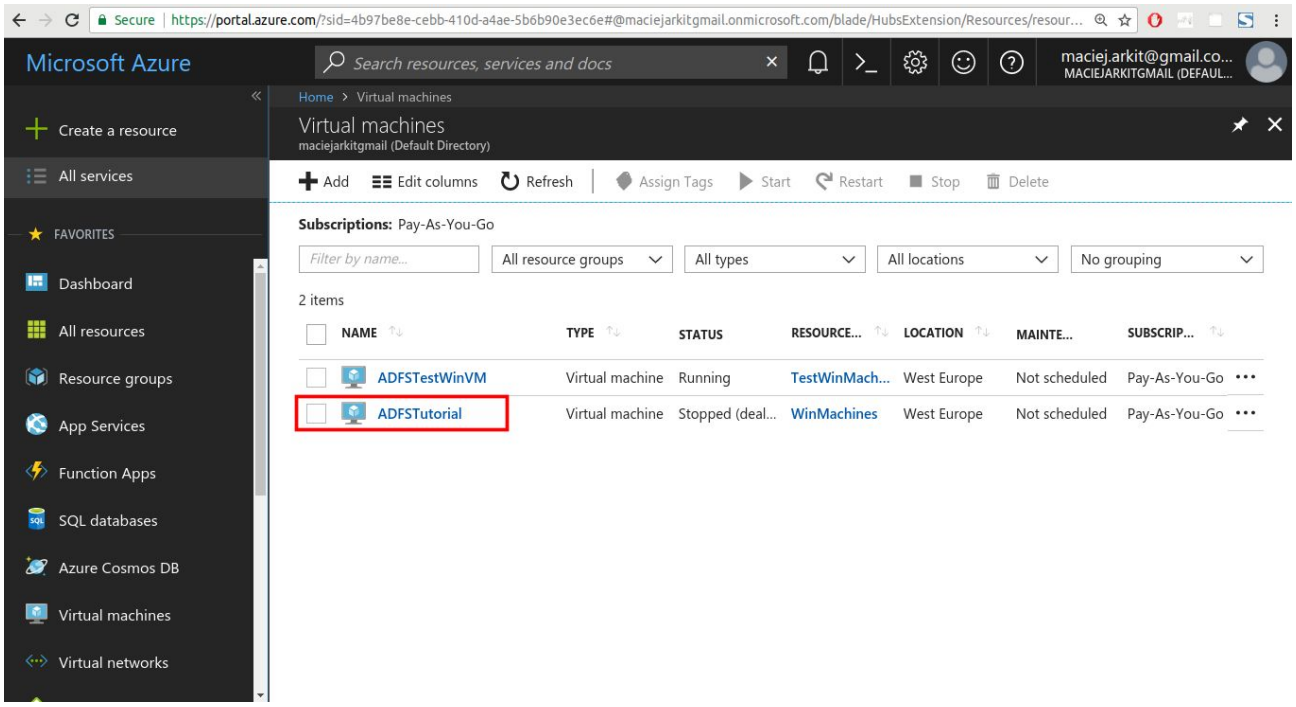
13. **Step 4 Summary:** Please review your VM details and click **“Create”** to finish creation process. Your VM will be created.



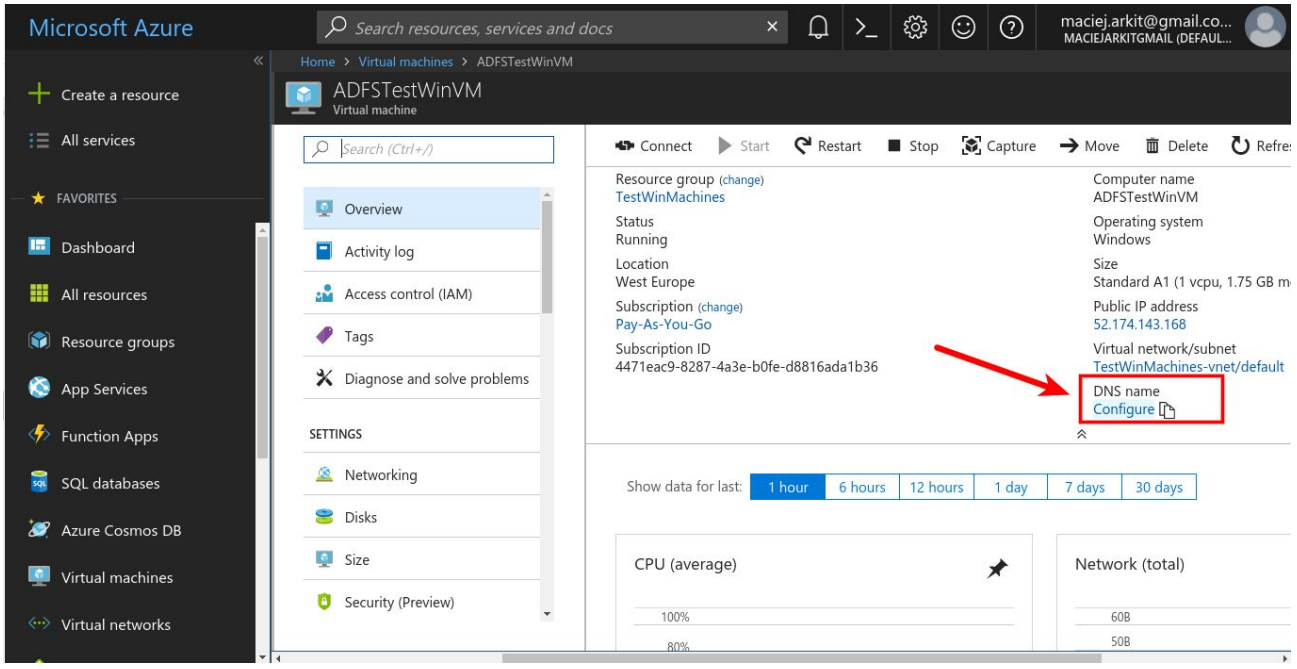


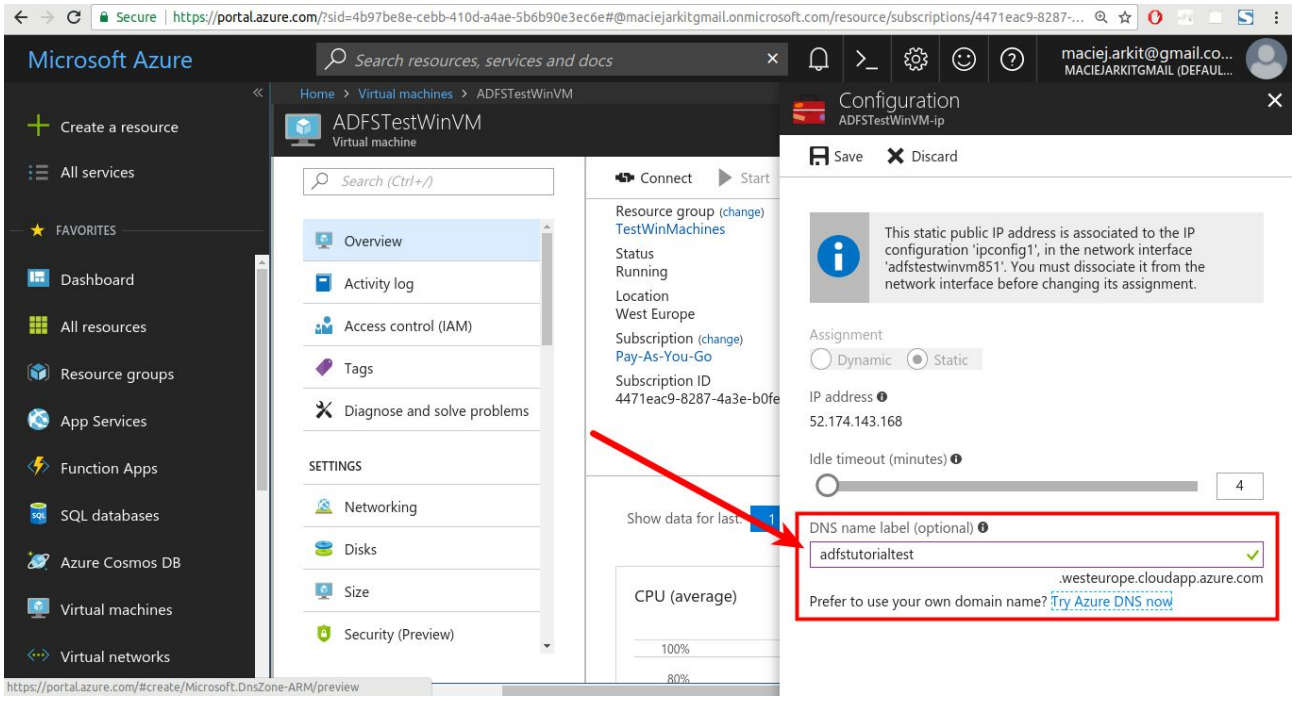
1.2 Finish VM configuration - configure DNS name

After couple minutes your machine will be deployed and ready to use. Now you should configure DNS name. It is required to setup self signed certificate.
Click on your VM name in **"Virtual Machines"** section.



Then click "DNS name Configure". Then provide DNS name and click "Save".





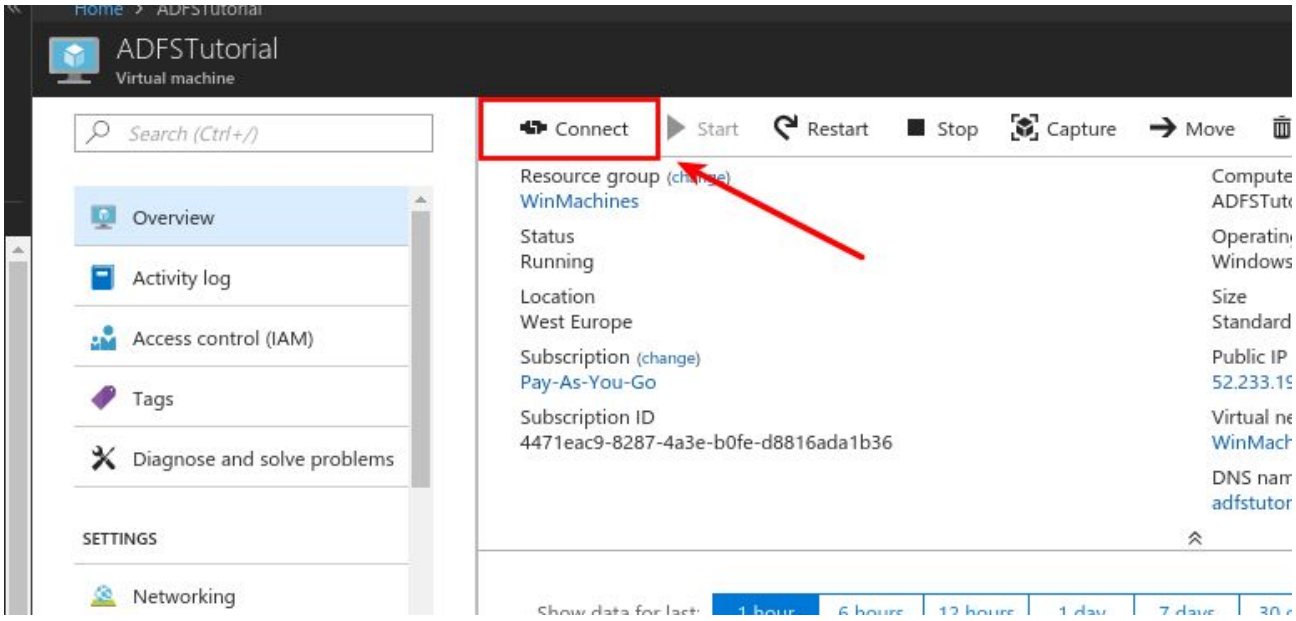
After a while VM will be available at: ${NAME}.westeurope.cloudapp.azure.com
It may take a few minutes until DNS information will get propagated. If you cannot reach your VM using newly configured DNS, try to refresh DNS service":

Linux
`sudo /etc/init.d/nscd restart`

Windows
`c:\> ipconfig /flushdns`

1.3 Connect to VM

At this step you should try to connect to VM via RDP.
Use any Remote Desktop client you like - ex. “Remmina” for Linux or “MS Remote Desktop” for Mac OS.
To configure RDP connection, open your Virtual Machine dashboard in Azure, and click “Connect” button, to download file with configuration. Use this file to configure connection in your RDP client. Then try to connect using username and password provided during VM initial configuration.



Sample RDP file

```
full address:s:${NAME}.westeurope.cloudapp.azure.com:3389
prompt for credentials:i:1
administrative session:i:1
```

You should be able to access your VM and see Windows desktop.
At this point your VM is configured and ready to setup and configure AD FS service.

---

2. Setup and configure AD FS service

Once Windows 2012 machine is configured, we can move step forward and configure services which are needed for AD FS.
We will have to configure:

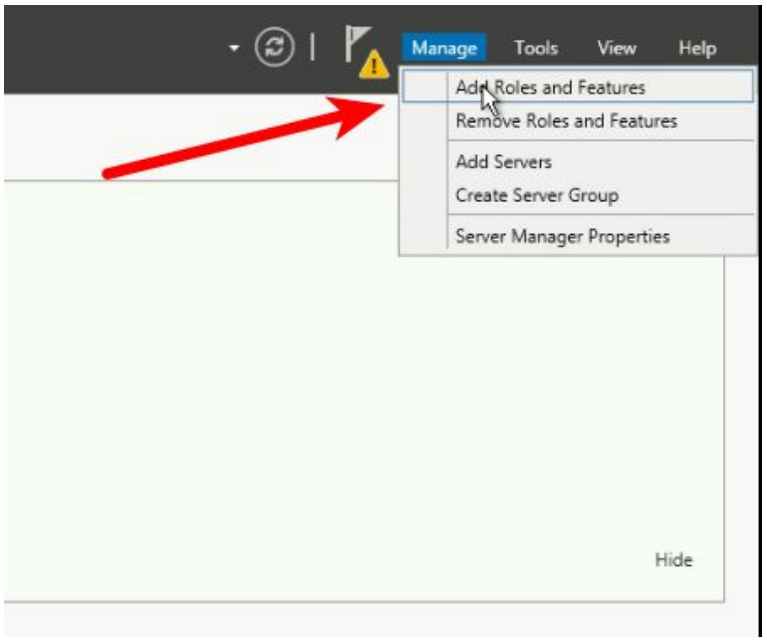
- Configure Active Directory Services and promote server to domain controller
- Generate self-signed certificate
- Install and configure AD FS

- Install and configureIIS server which will be processing HTTP traffic and forwarding it to AD FS

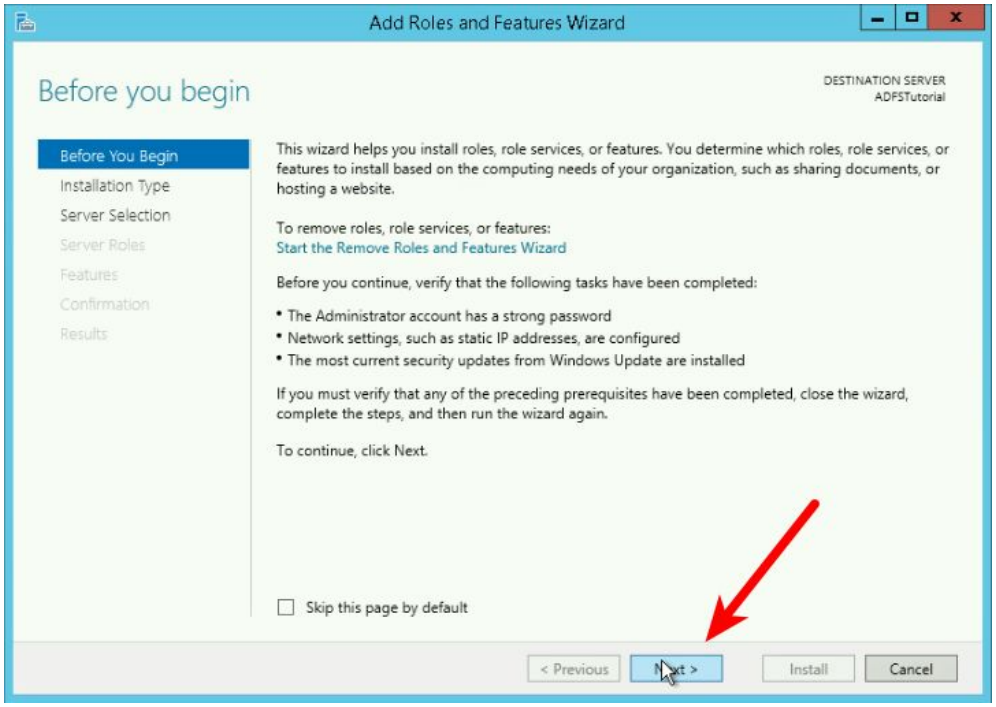
2.1 Configure Active Directory Domain Services and promote server to domain controller

Before configuring AD FS Service, it is required to configure Active Directory Domain Services and promote server to a domain controller.

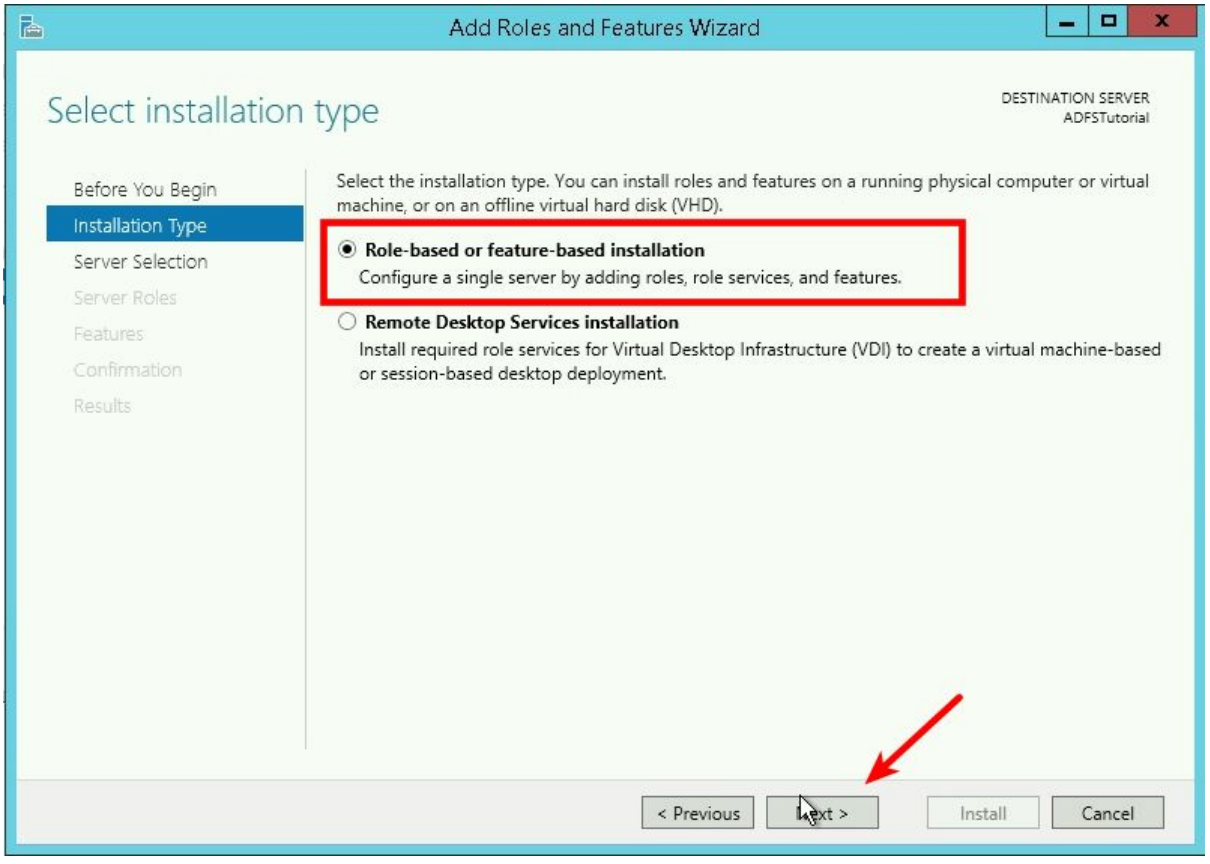
1/ Open “Server Manager”
2/ Click “Add Roles and Features”



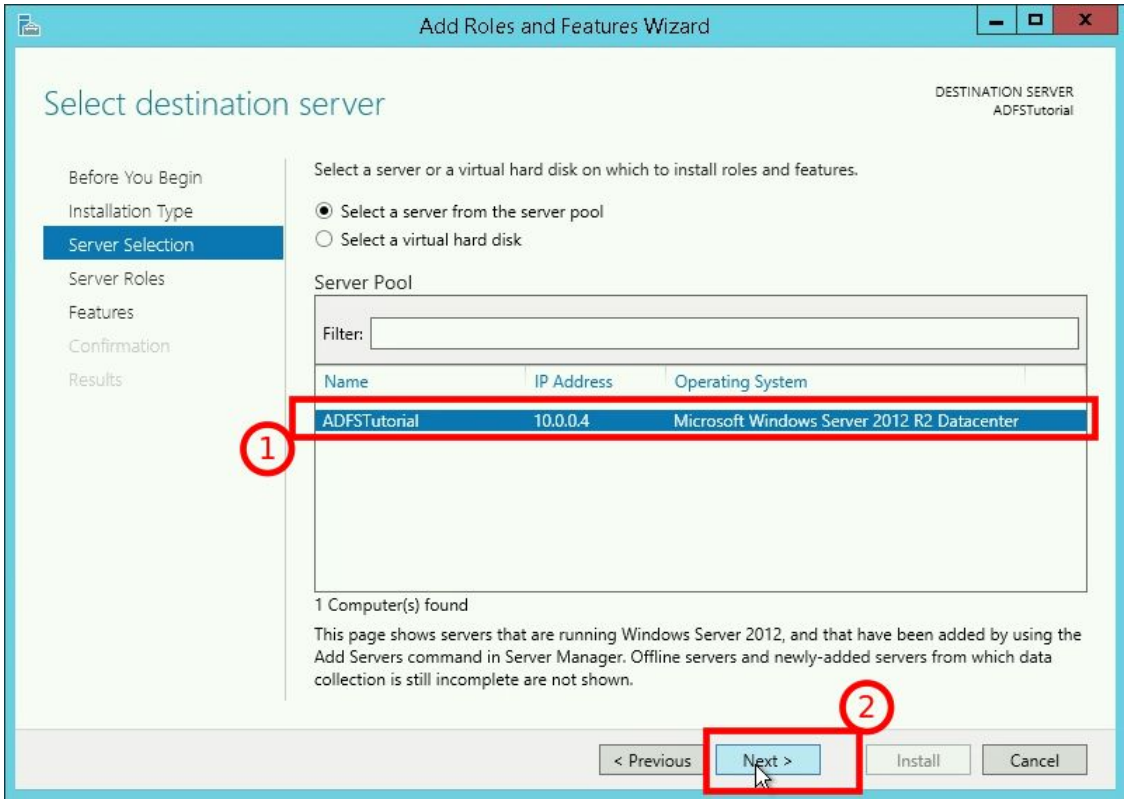
3/ Click “Next” to skip “Before you begin” page:



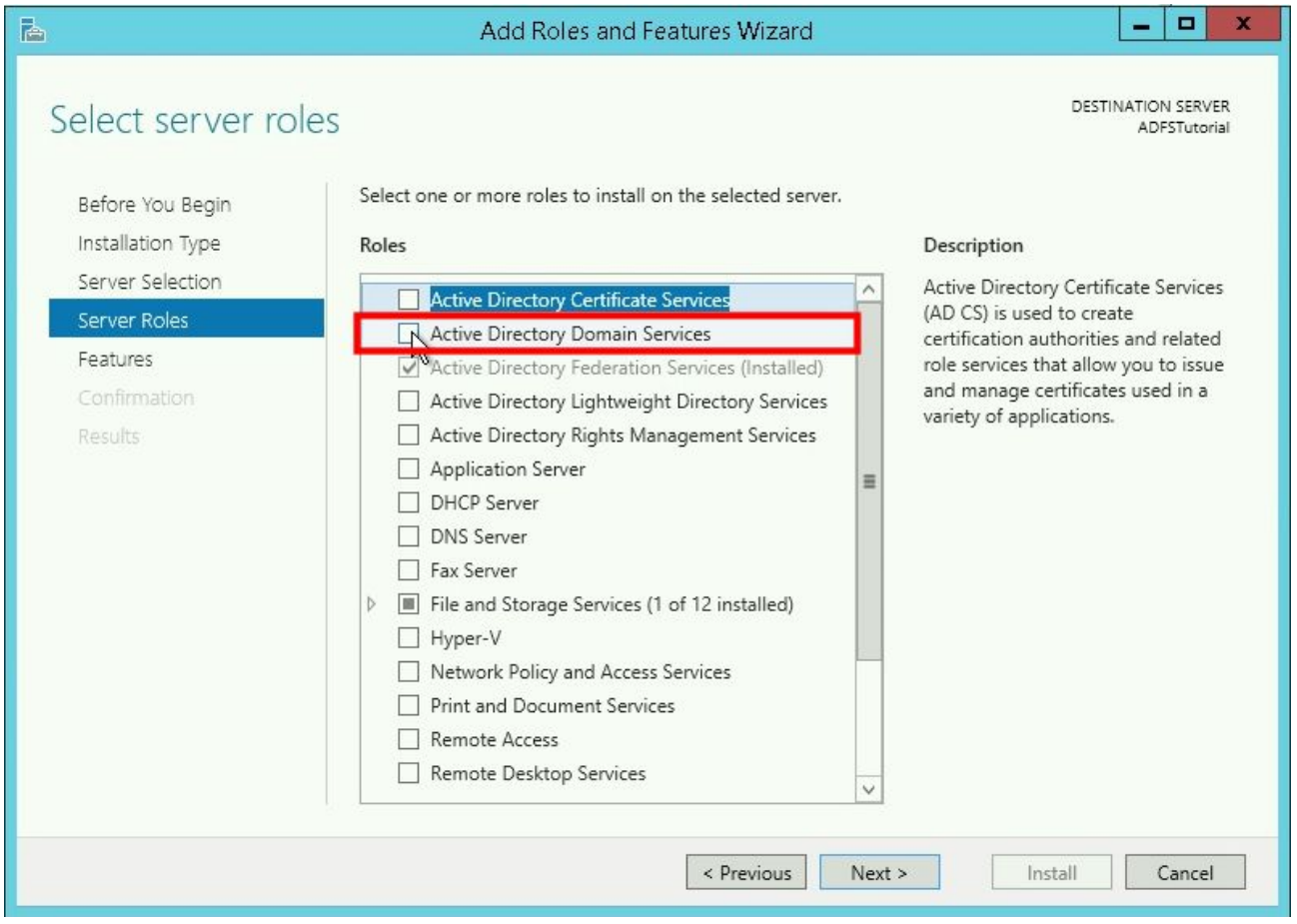
4/ On “Installation Type” page, select “Role-based or feature-based installation” and click “Next”:



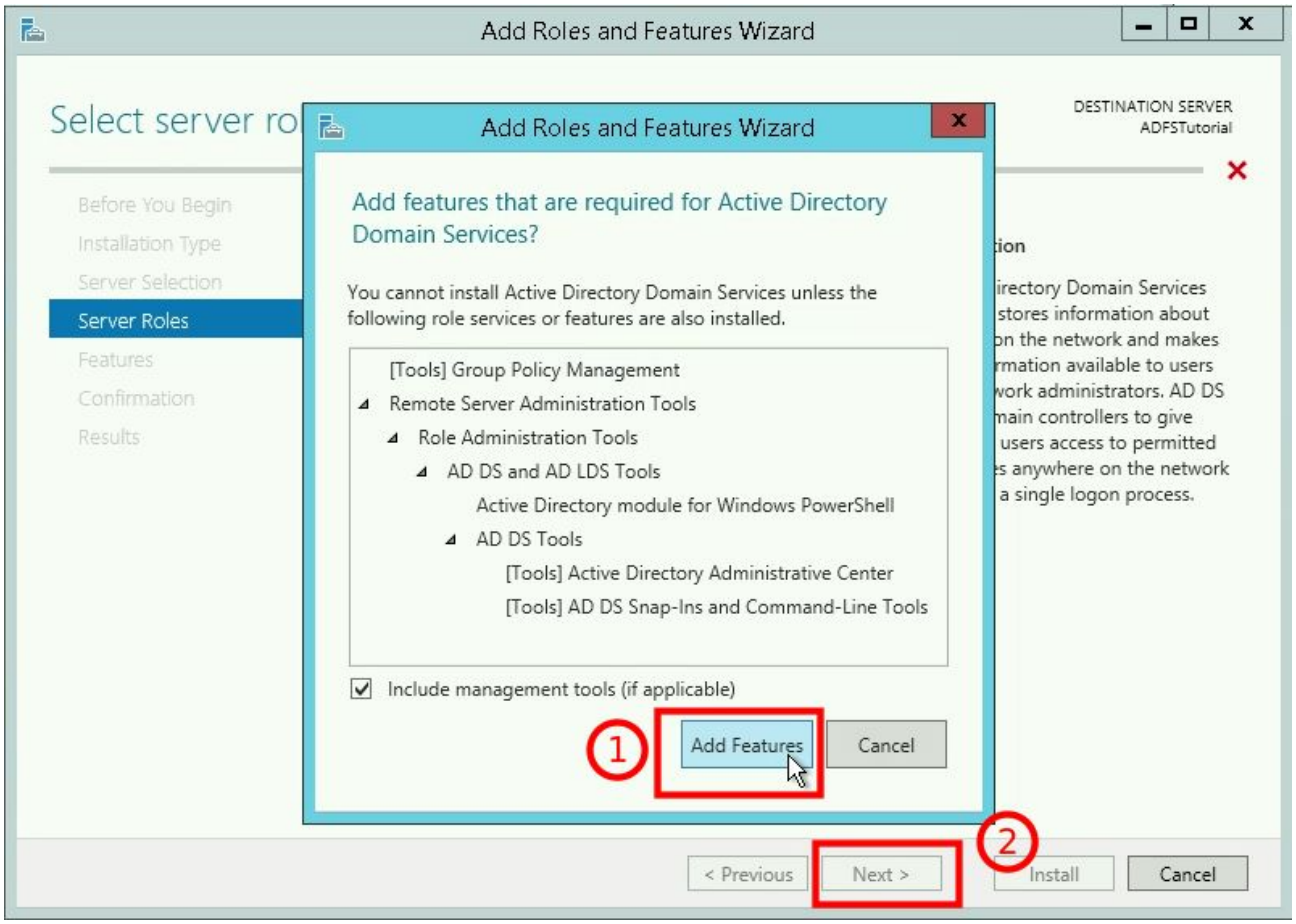
5/ On the “Server Selection” page, select your server, and click “Next”:



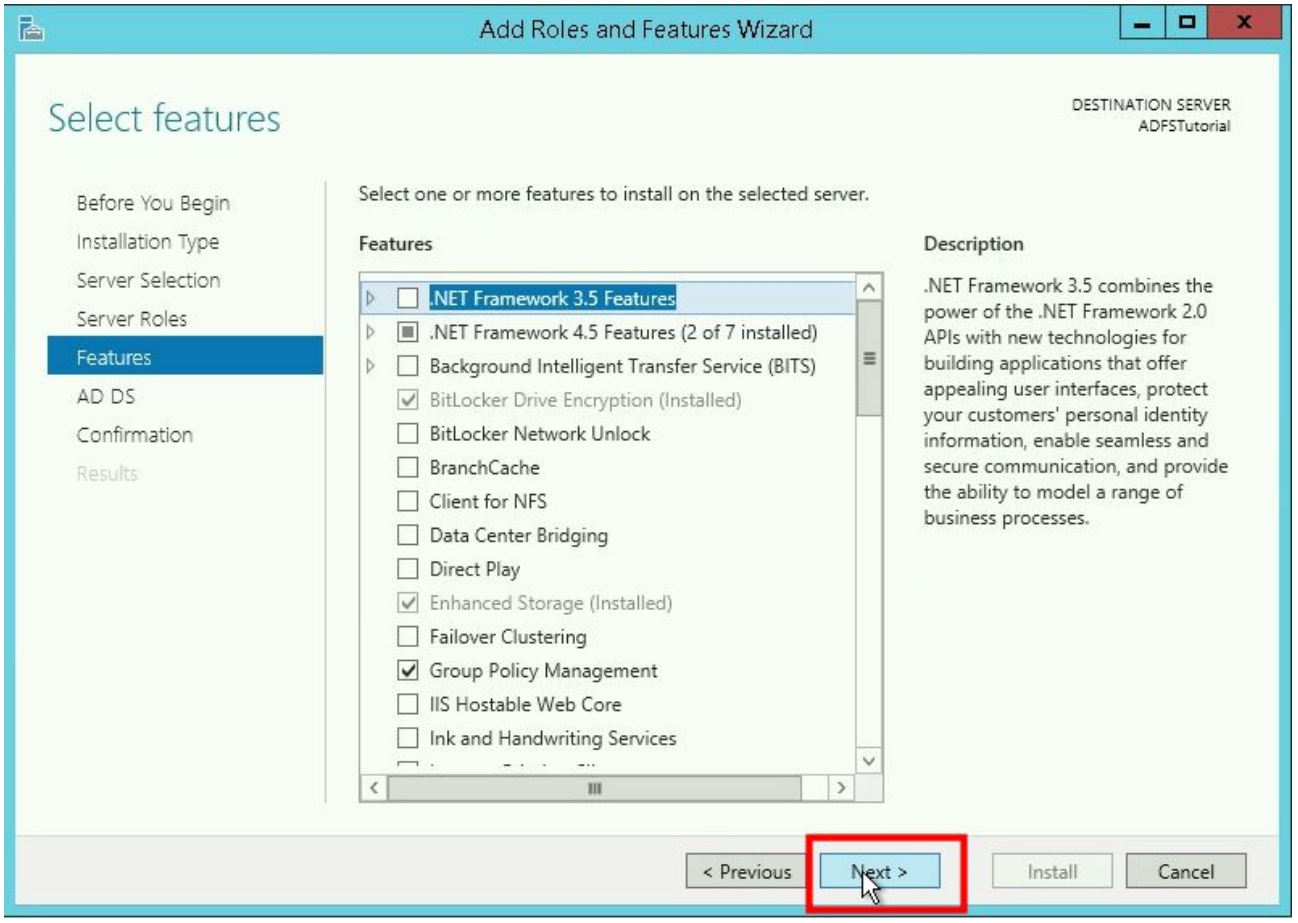
6/ On “Server Roles” page, select “Active Directory Domain Services”.



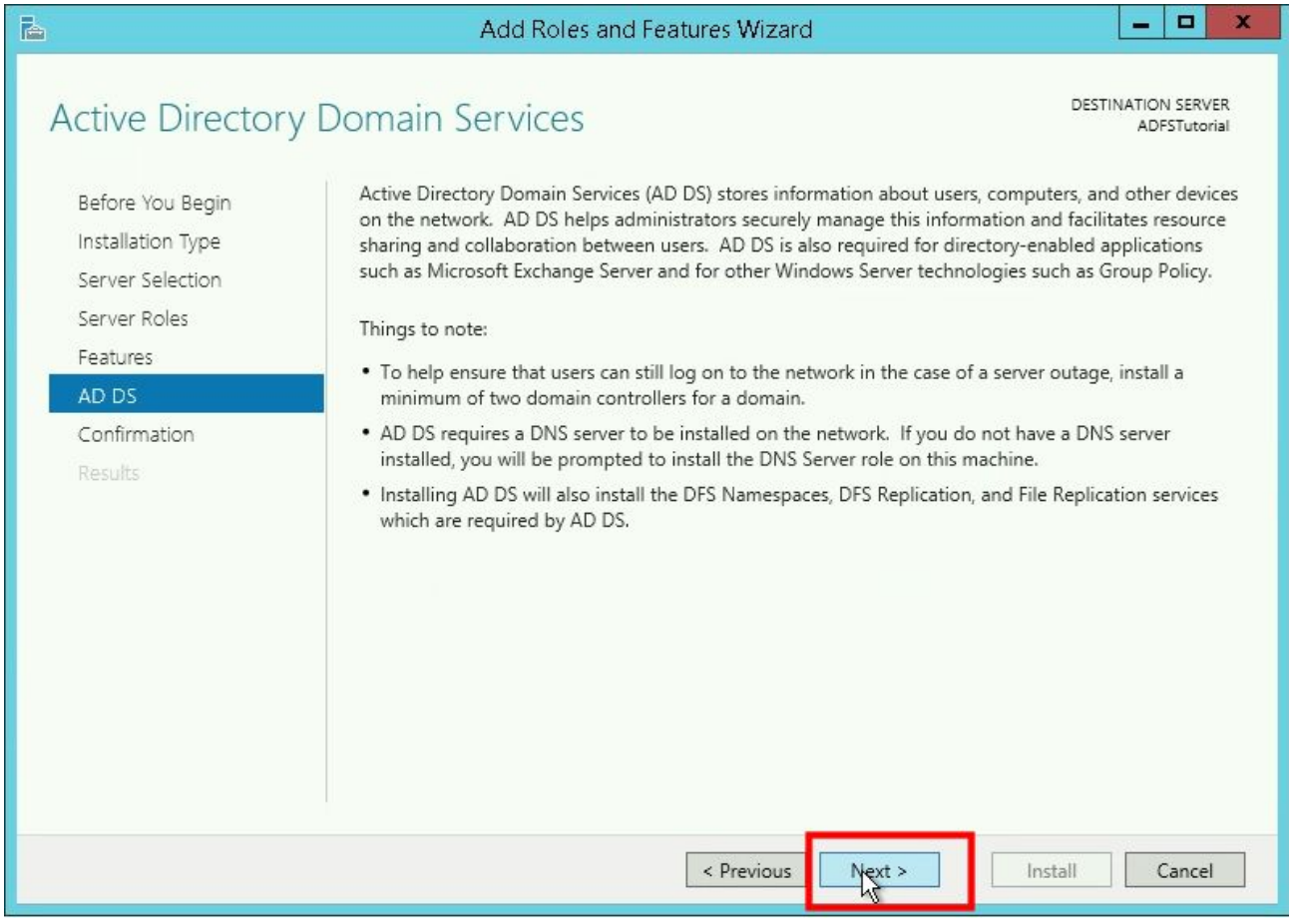
7/ When selected “Active Directory Domain Services”, a modal dialog with question “Add features that are required for Active Directory Domain Services?” will pop up. Please click “Add Features”, and then click “Next”:



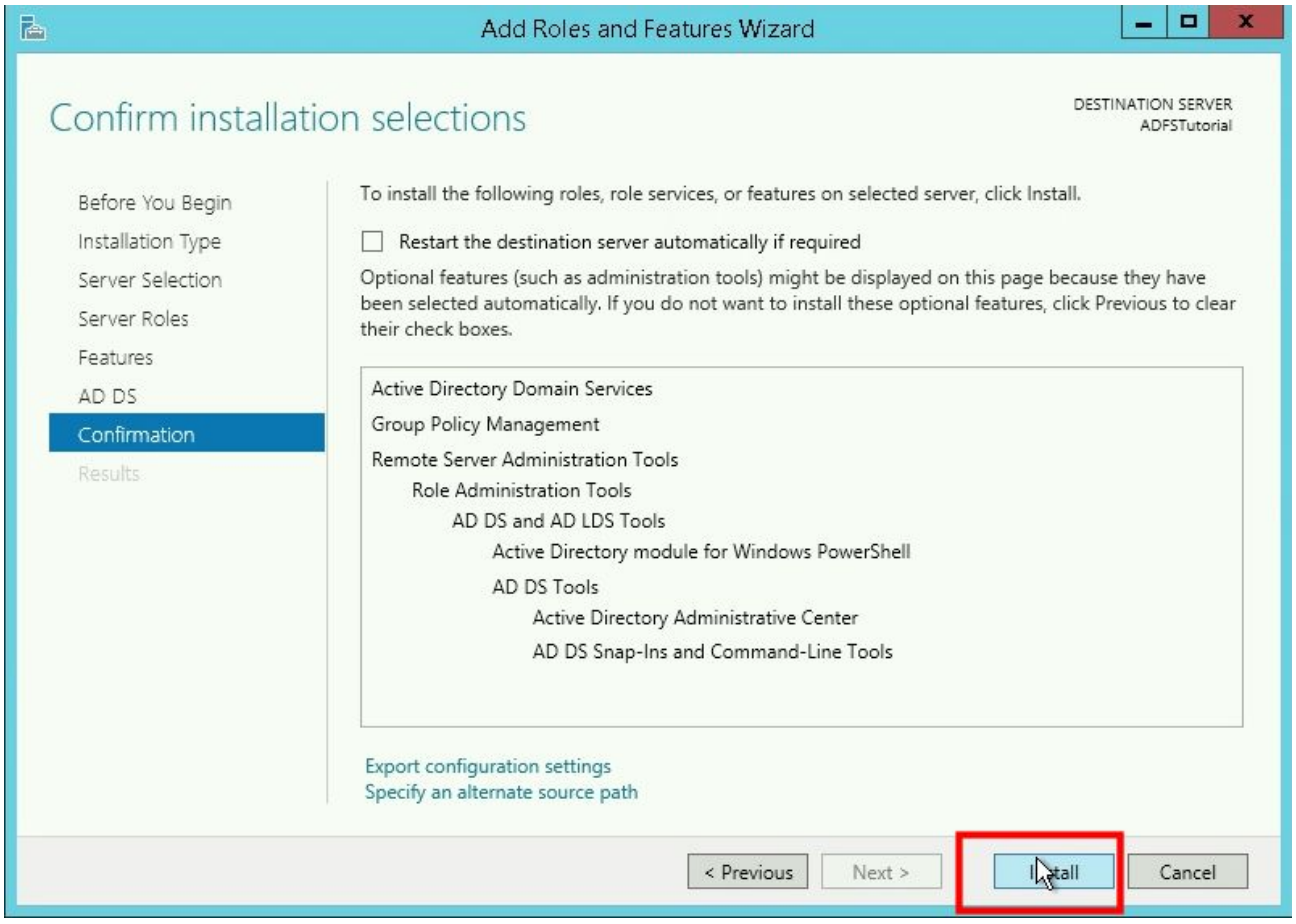
8/ On “Features” page please do not change anything. Please click “Next”:



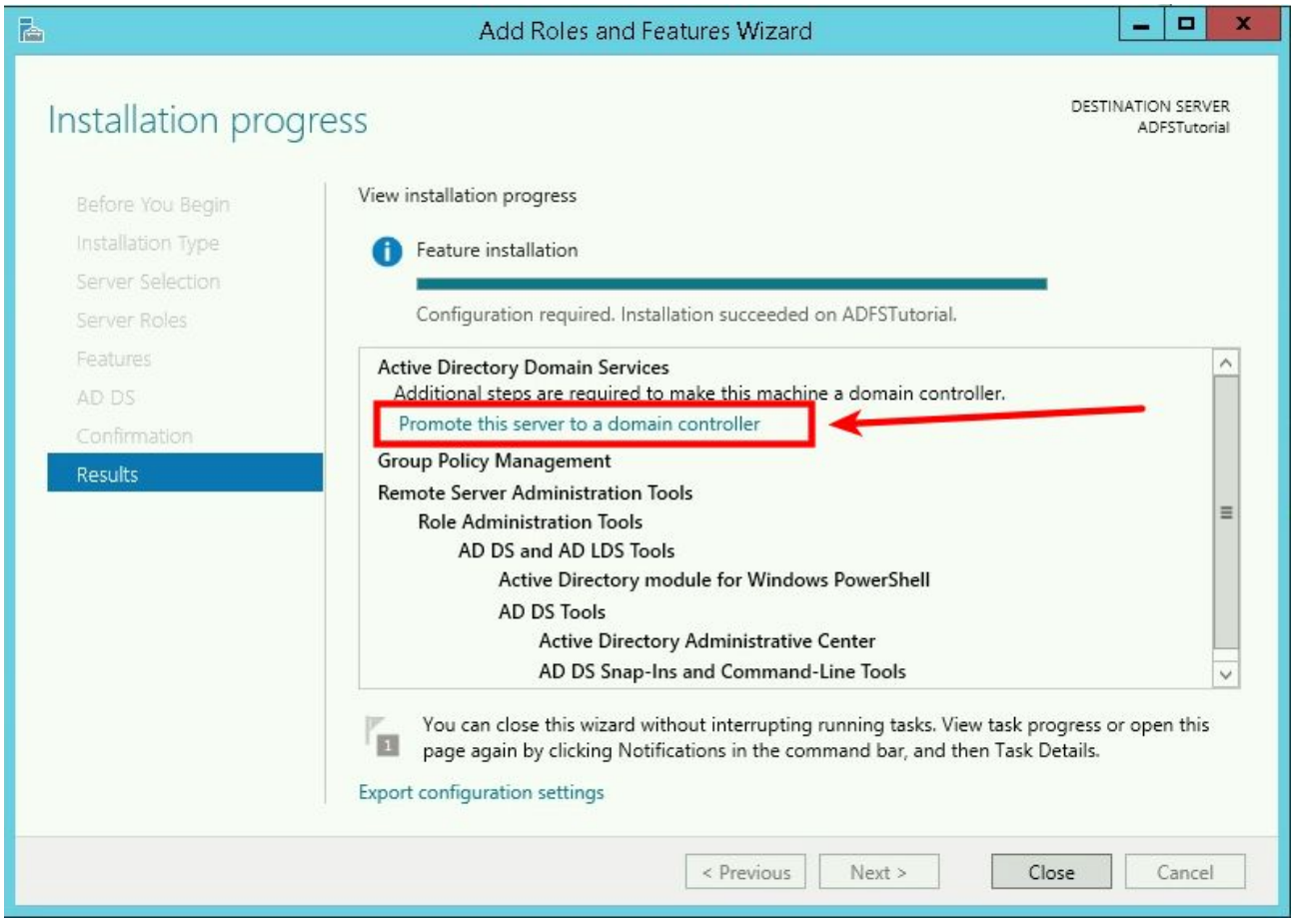
9/ On “AD DS” please click “Next”:



10/ On “Confirmation” page, please click “Install” and please stand by while installation is in progress:

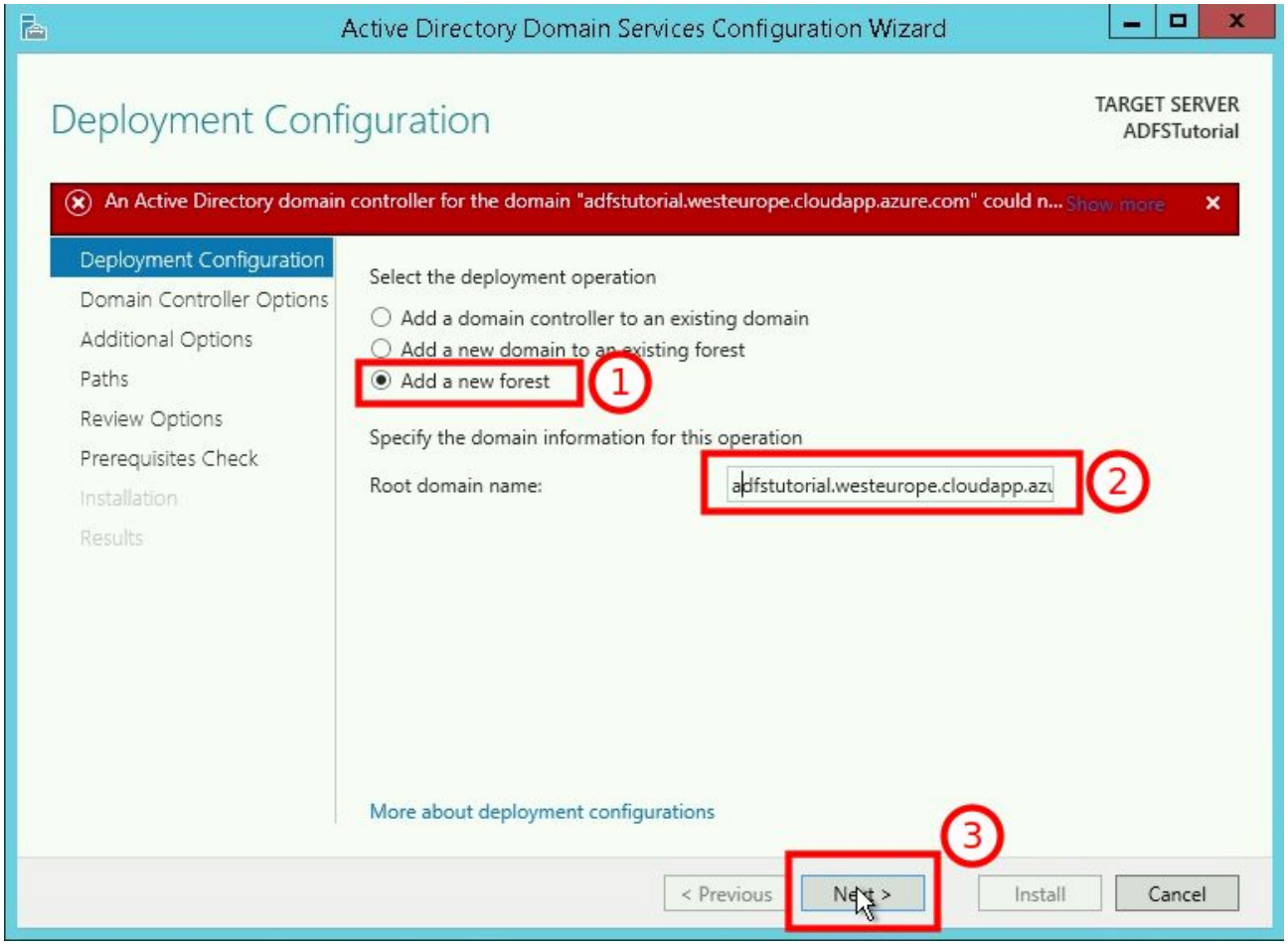


11/ When installation is finished, please click **“Promote this server to a domain controller”**:

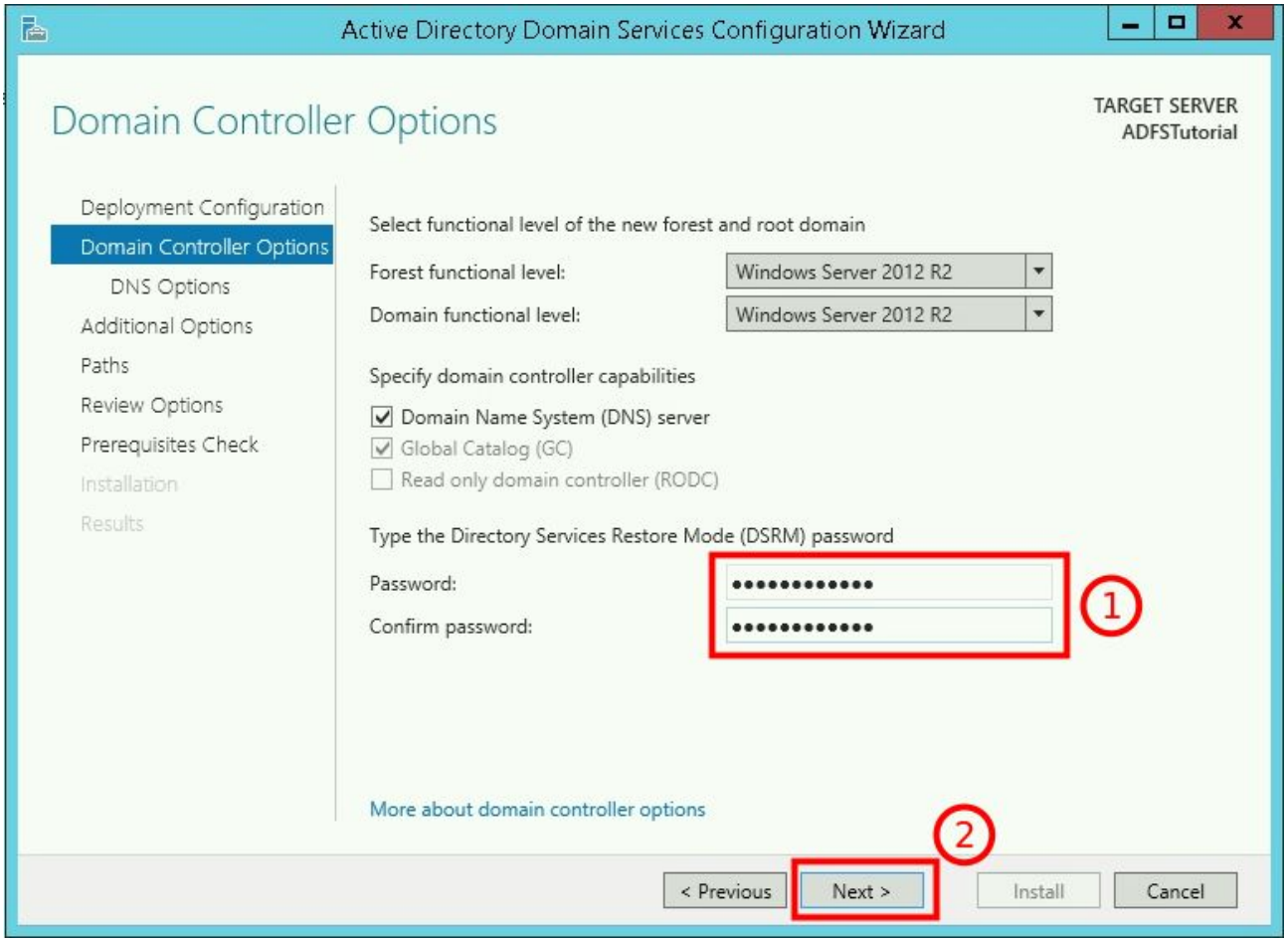


12/ After clicking **"Promote this server to a domain controller"**, you will see **"Active Directory Domain Services Configuration Wizard"**.

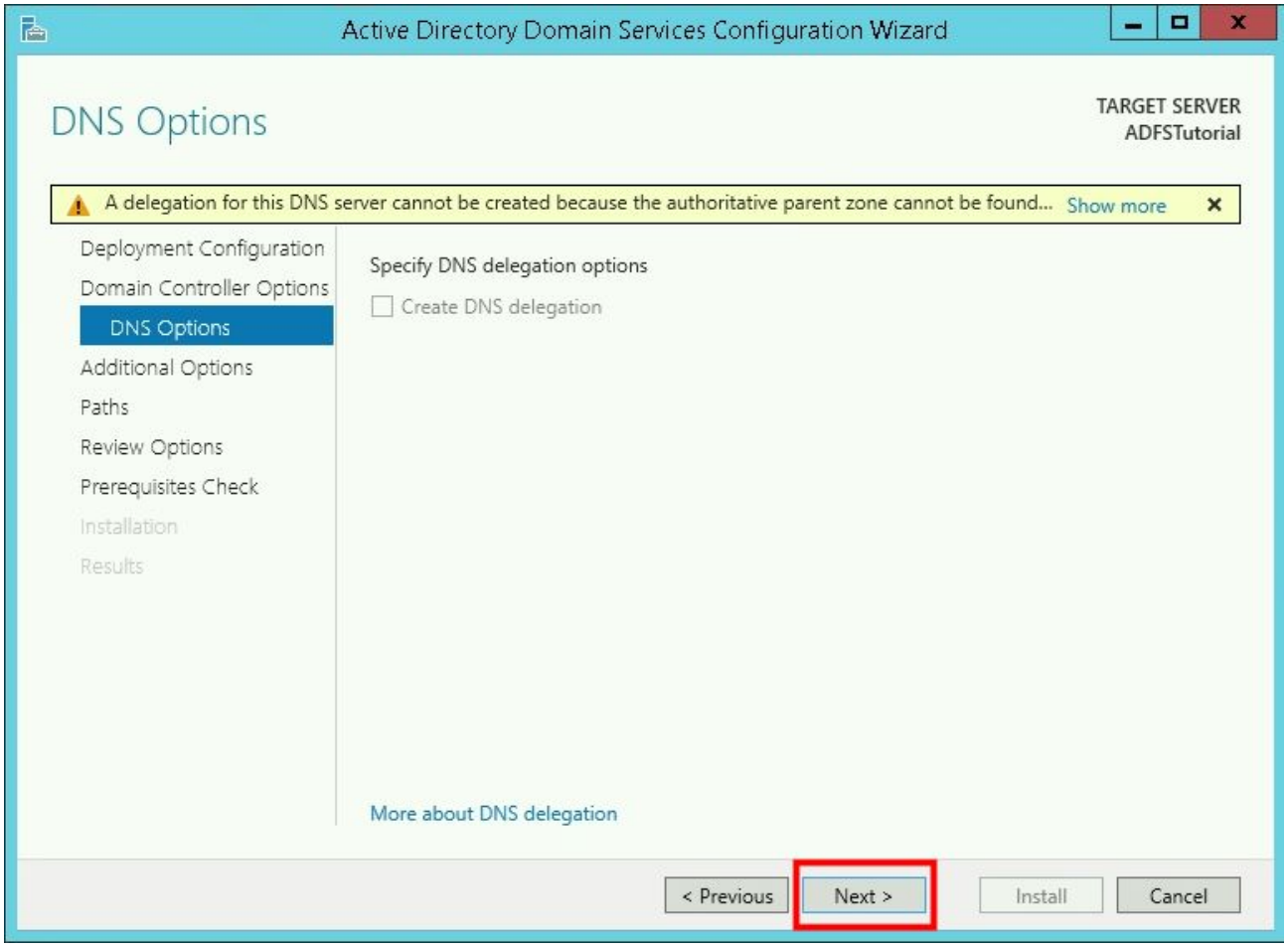
1. Please select **"Add a new forest"**
2. Provider **"Root domain name"** - you can use same name as DNS name assigned to your server - ex. ABCD.westeurope.cloudapp.azure.com
3. Click **"Next"** (**NOTICE:** It will take few minutes before new forest will be created and you will be redirected to page "Domain Controller Options")



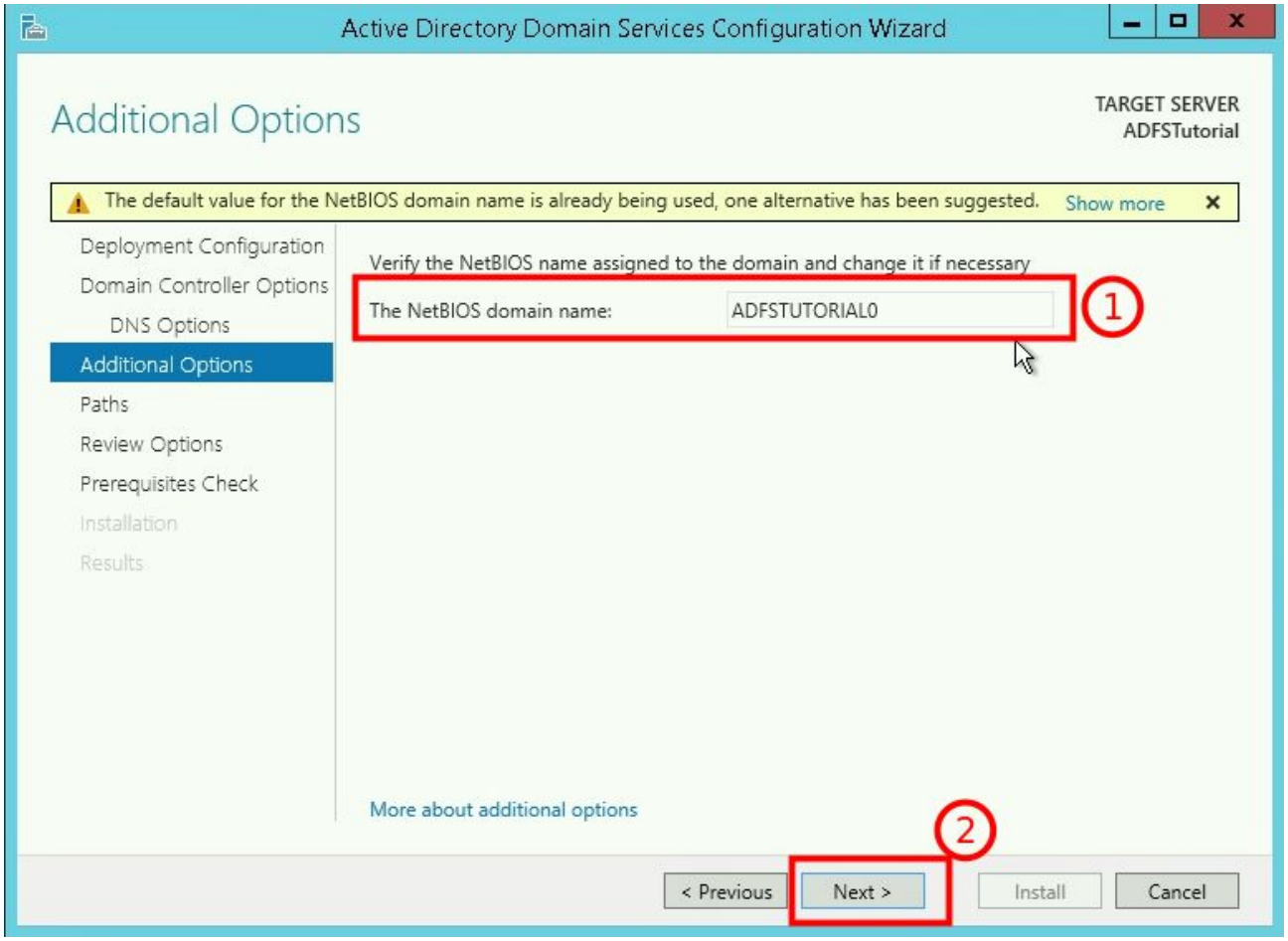
13/ On “Domain Controller Options” page, please **“Type the Directory Services Restore Mode (DSRM) password” (1)** and click **“Next” (2)**:



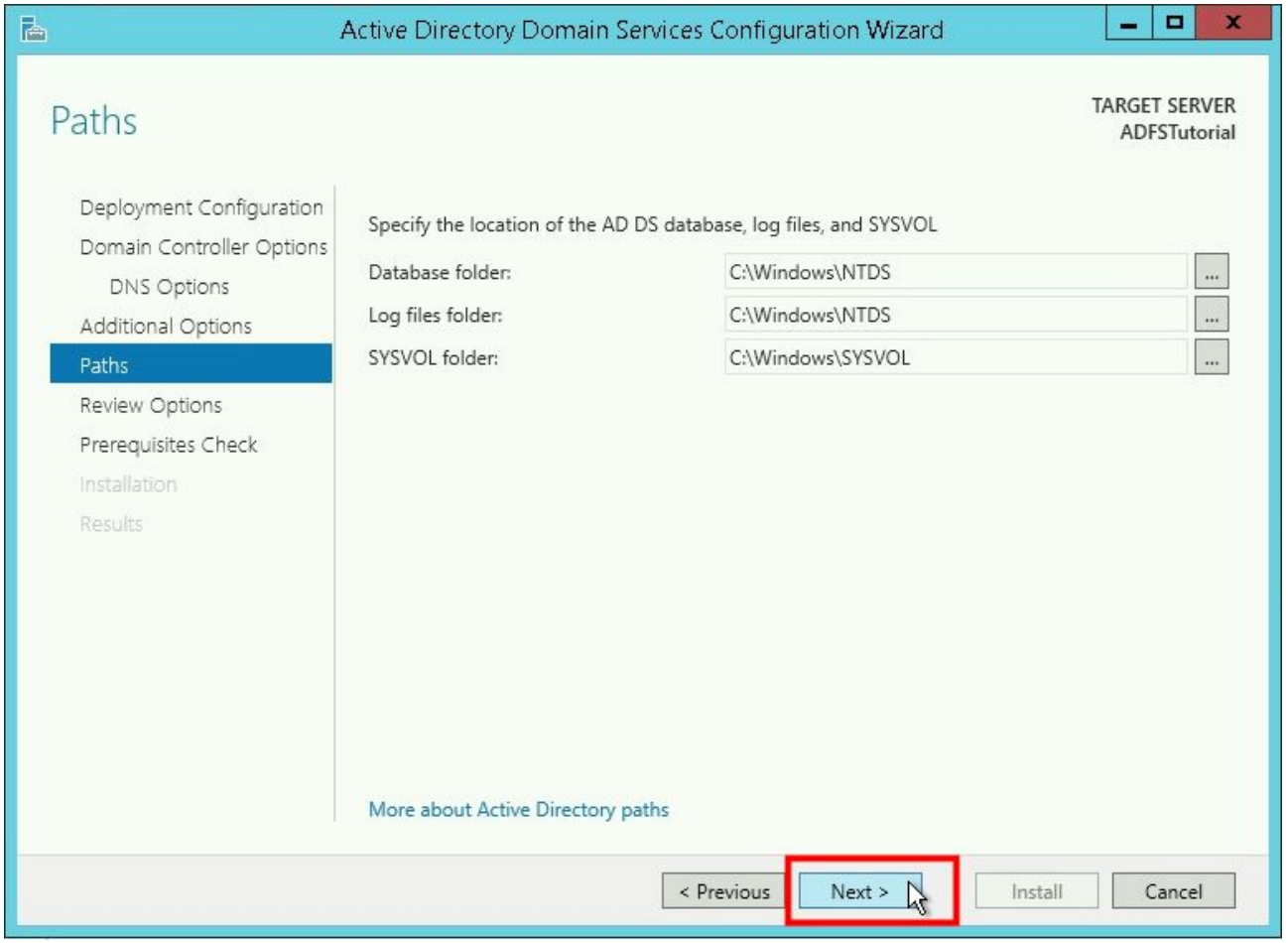
14/ On **“DNS Options”** page, please click **“Next”**:



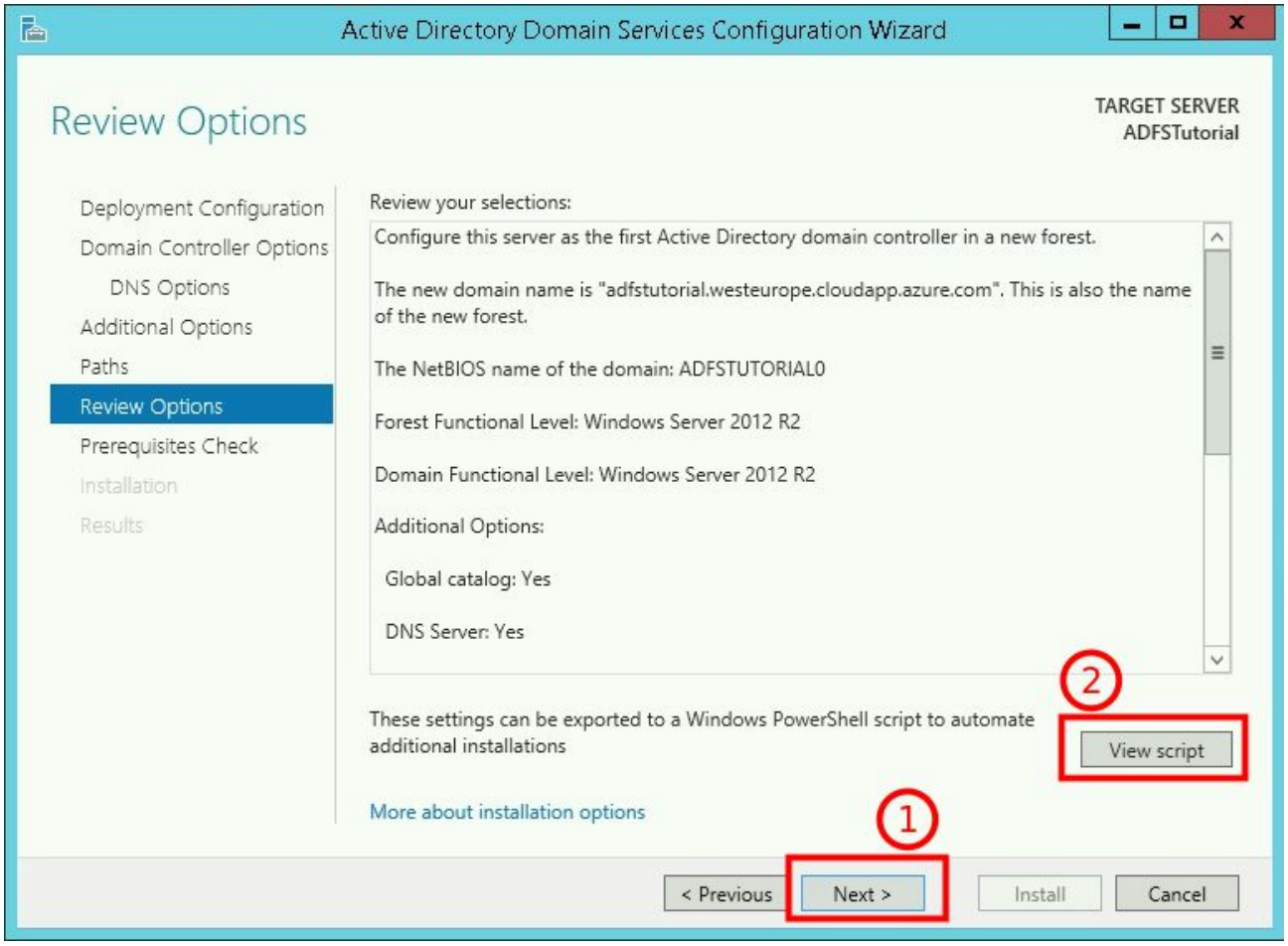
15/ On “Additional Options” page, please verify **“The NetBIOS domain name” (1)** - change it if you need (you can leave suggested one), and click **“Next” (2)**



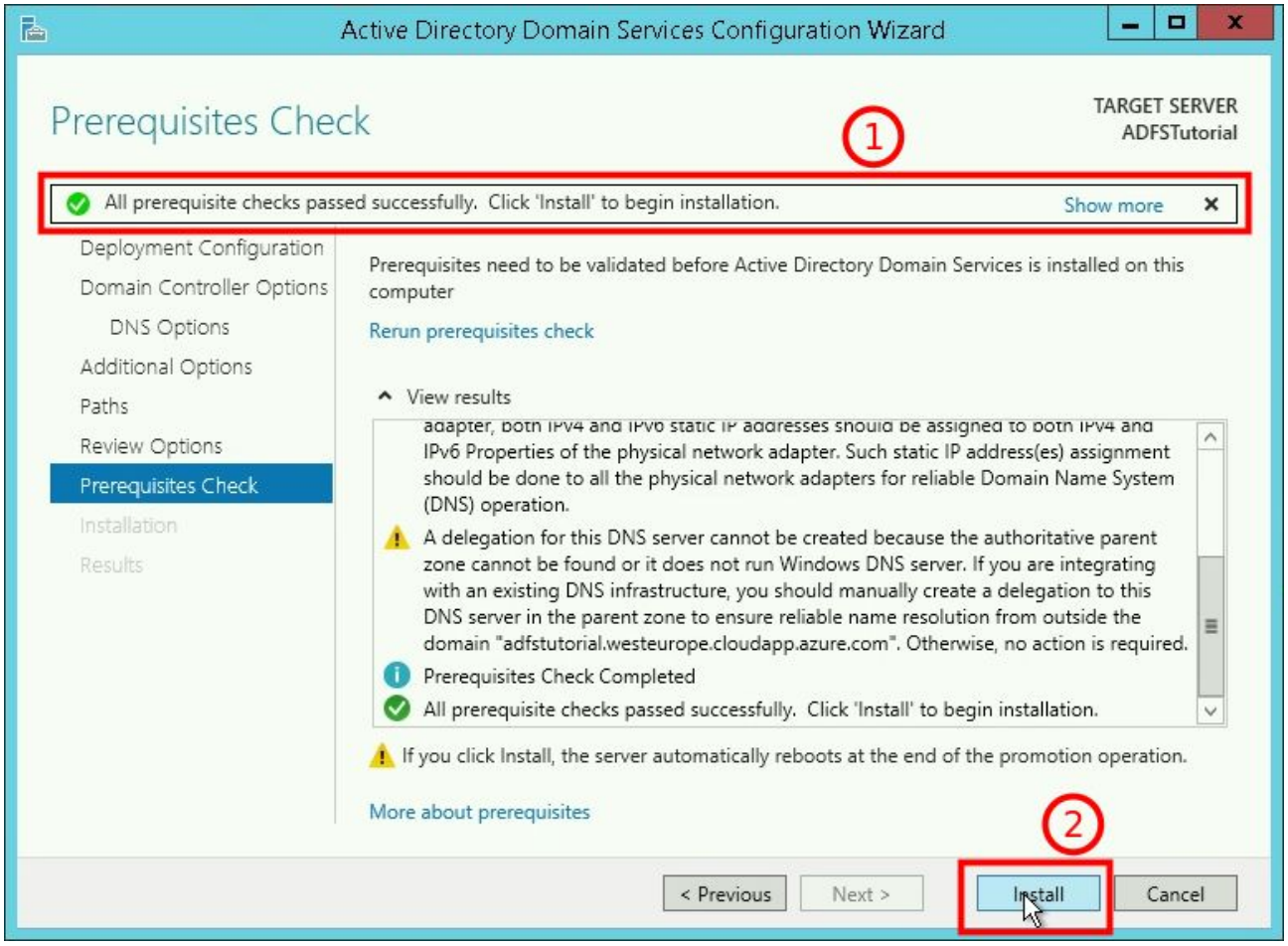
16/ On **“Paths”** page, please leave default values and click **“Next”**



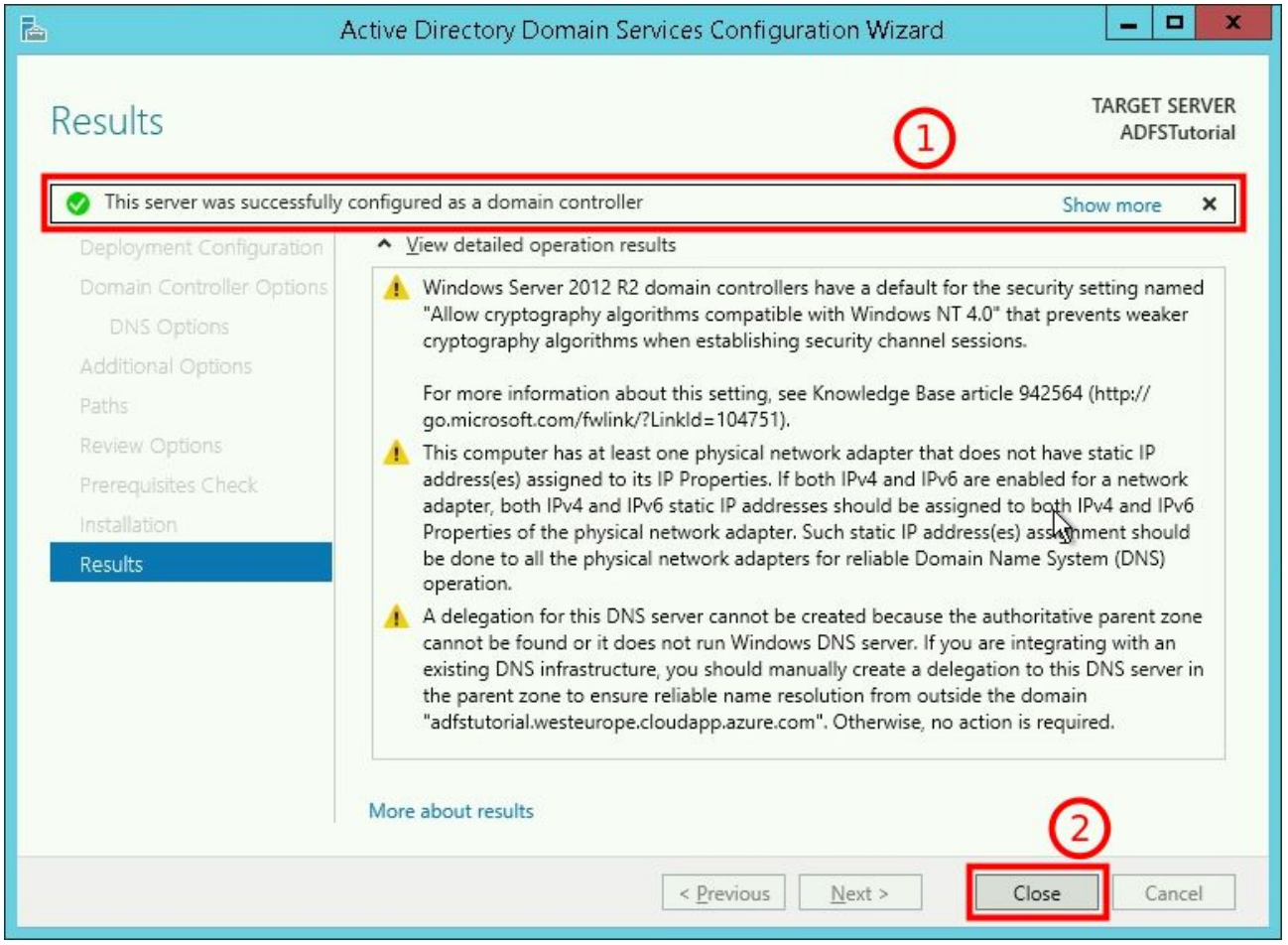
17/ Please **“Review Options”** and click **“Next” (1)**. At this step you can click **“View script” (2)** and save it for reference or to use for further installations (PowerShell script).



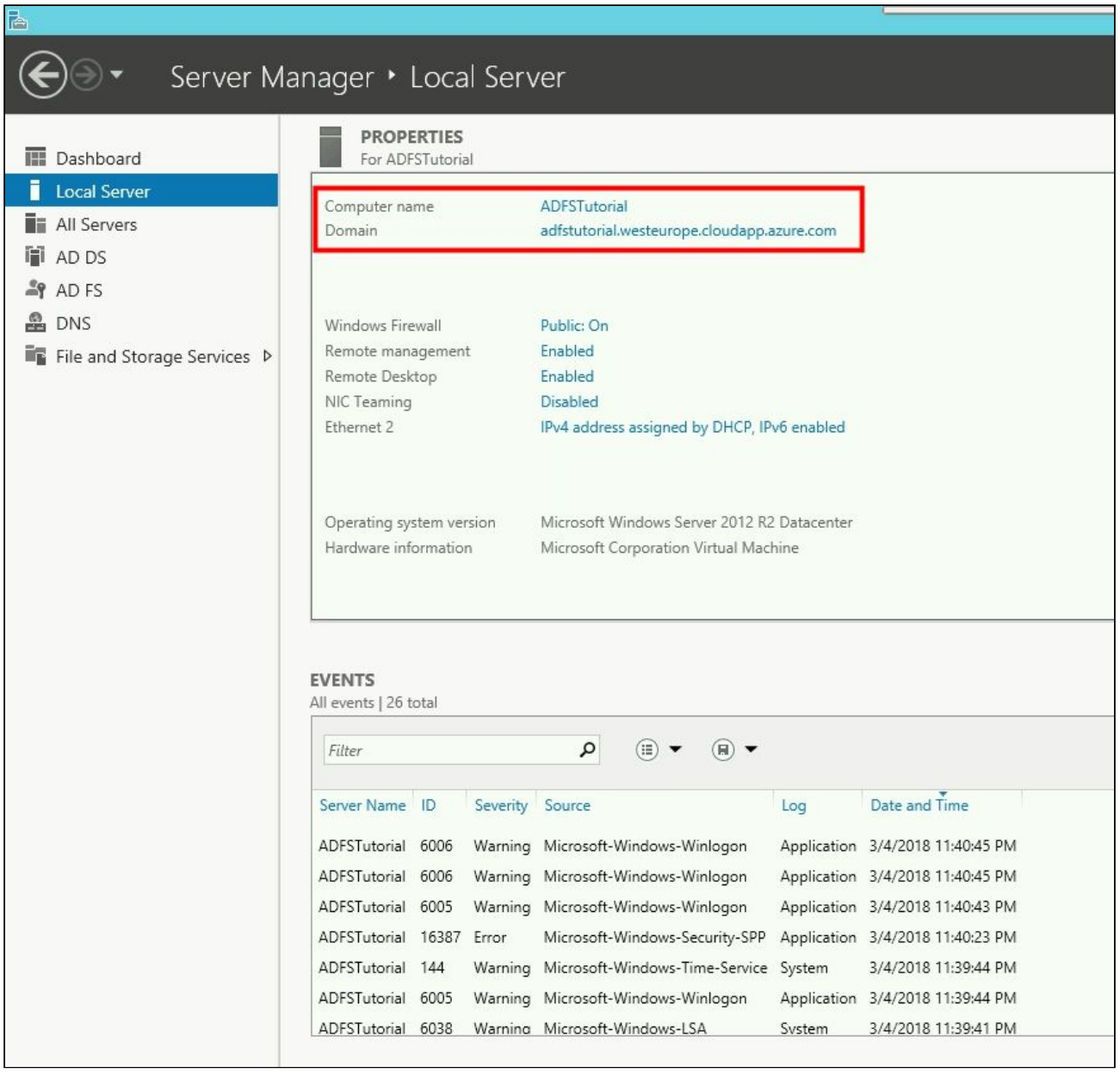
18/ On **"Prerequisites Check"**, please verify that all prerequisite checks passed **(1)** and click **"Install" (2)** to start AD DS installation process.



19/ On **"Results"** page, please ensure that installation finished successfully **(1)** and please click **"Close" (2)**. **NOTE:** After clicking "Close" your server will have to be restarted.


Active Directory Domain Services Configuration Wizard
Results
TARGET SERVER
ADFSTutorial
1
This server was successfully configured as a domain controller
Show more
Deployment Configuration
Domain Controller Options
DNS Options
Additional Options
Paths
Review Options
Prerequisites Check
Installation
Results
View detailed operation results
Windows Server 2012 R2 domain controllers have a default for the security setting named "Allow cryptography algorithms compatible with Windows NT 4.0" that prevents weaker cryptography algorithms when establishing security channel sessions.
For more information about this setting, see Knowledge Base article 942564 (http://go.microsoft.com/fwlink/?LinkId=104751).
This computer has at least one physical network adapter that does not have static IP address(es) assigned to its IP Properties. If both IPv4 and IPv6 are enabled for a network adapter, both IPv4 and IPv6 static IP addresses should be assigned to both IPv4 and IPv6 Properties of the physical network adapter. Such static IP address(es) assignment should be done to all the physical network adapters for reliable Domain Name System (DNS) operation.
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 "adfstutorial.westeurope.cloudapp.azure.com". Otherwise, no action is required.
More about results
2
< Previous
Next >
Close
Cancel

At this point Active Directory Domain Services are configured and server is promoted to a domain controller.



2.2 Generate Self Signed certificate for a domain

To configure AD FS instance we will need to generate self-signed certificate following the steps below:

1. Download ***New-SelfsignedCertificateEx*** module zip file from: https://gallery.technet.microsoft.com/scriptcenter/Self-signed-certificate-5920a7c6
2. Unzip file and import ***New-SelfsignedCertificateEx*** module to PowerShell by invoking command:

```
Import-Module %PATH%/New-SelfSignedCertificateEx.ps1
```

3. Generate self-signed certificate by invoking command below (replace "***adfstutorial.westeurope.cloudapp.azure.com***" by your domain name)

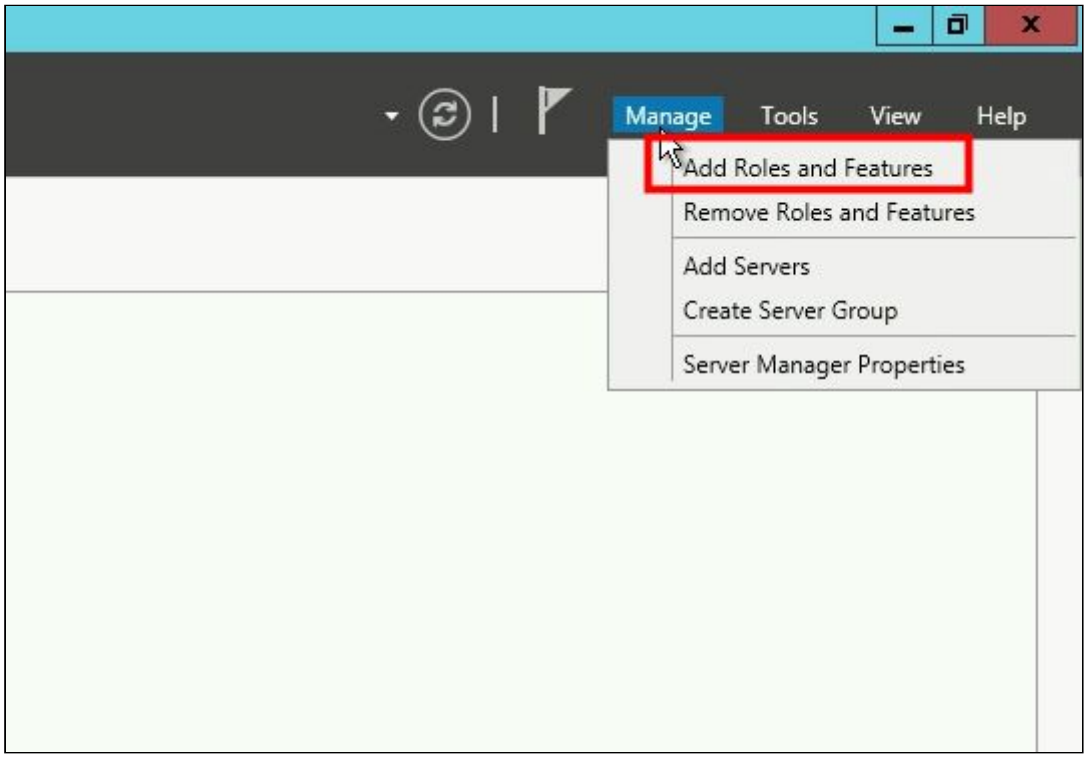
```
New-SelfSignedCertificateEx  -Subject 'CN=adfstutorial.westeurope.cloudapp.azure.com'
-ProviderName "Microsoft Enhanced RSA and AES Cryptographic Provider" -KeyLength 2048
-FriendlyName 'adfstutorial.westeurope.cloudapp.azure.com' -SignatureAlgorithm sha256
-EKU "Server Authentication", "Client authentication" -KeyUsage "KeyEncipherment,
DigitalSignature" -Exportable -StoreLocation "LocalMachine"
```

NOTE:
New-SelfsignedCertificateEx generates legacy compatible certificate which is required by AD FS. If you will use regular New-SelfsignedCertificate, the key generated will not work with AD FS.

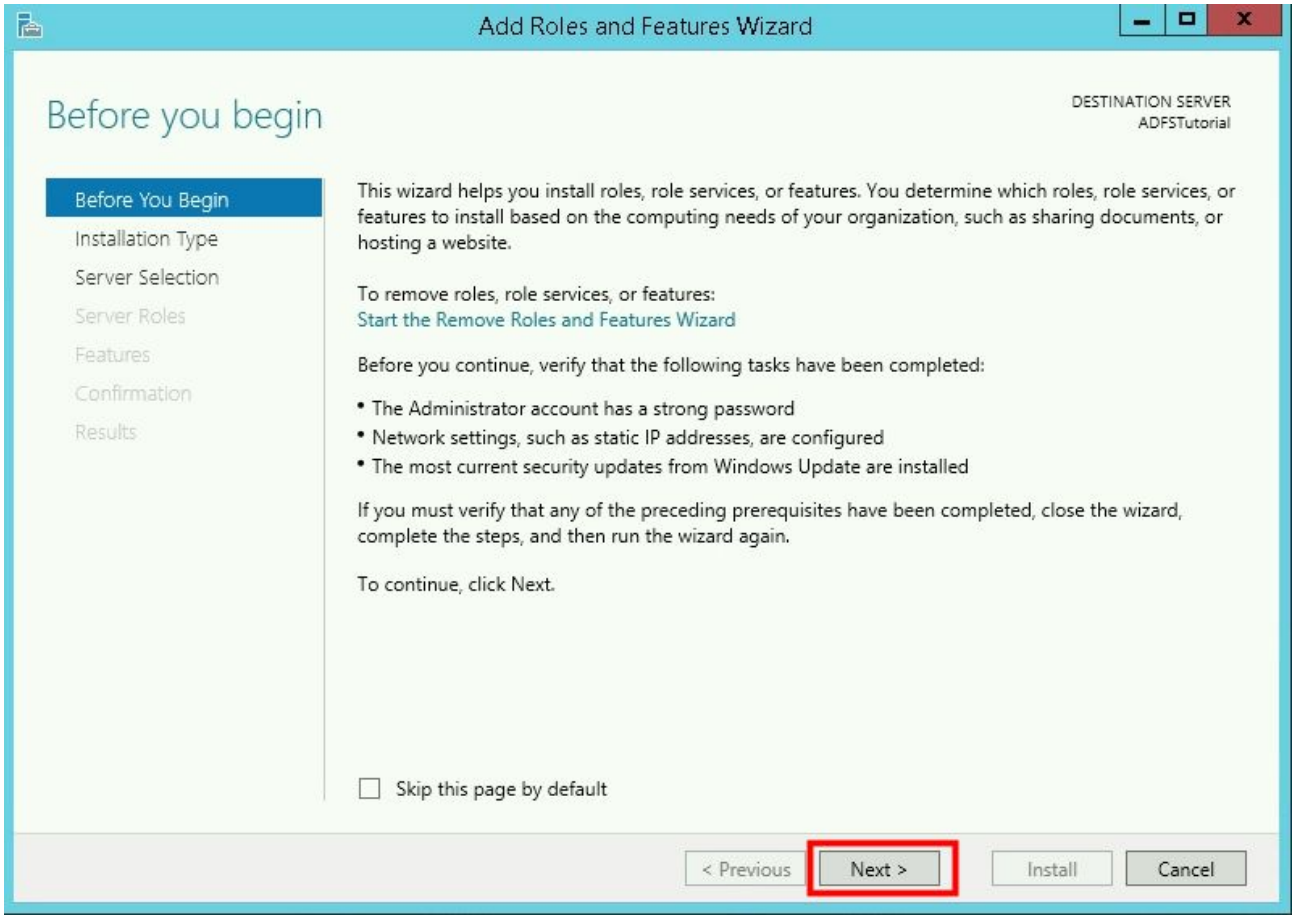
2.3 Install AD FS service

Once we have Active Directory Domain Services installed, our server is promoted to domain controller and self-signed certificate is generated, we can install and configure AD FS instance.

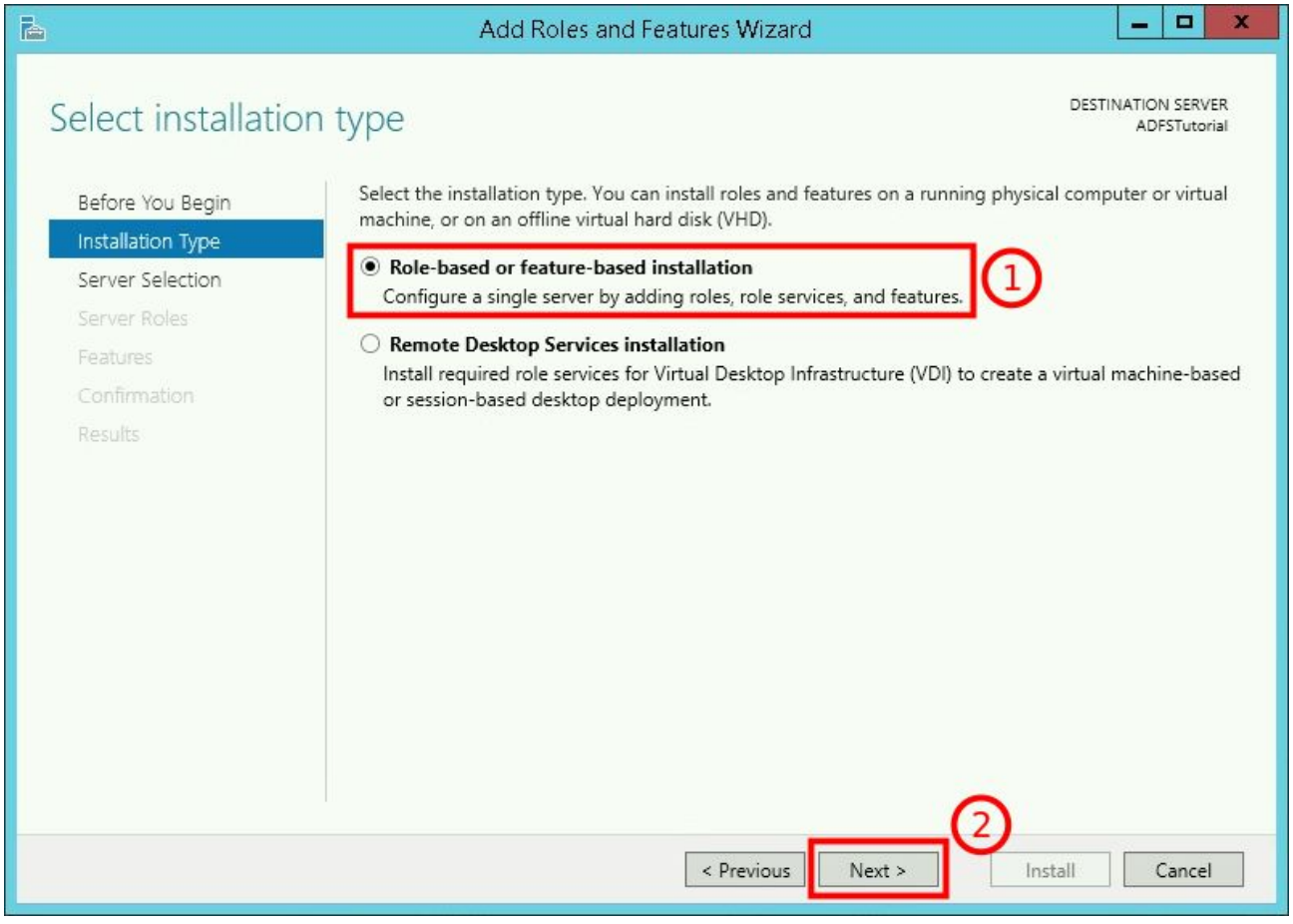
1/ Open "Server Manager"
2/ Click "Manage" and then "Add Roles and Features" in right upper corner:



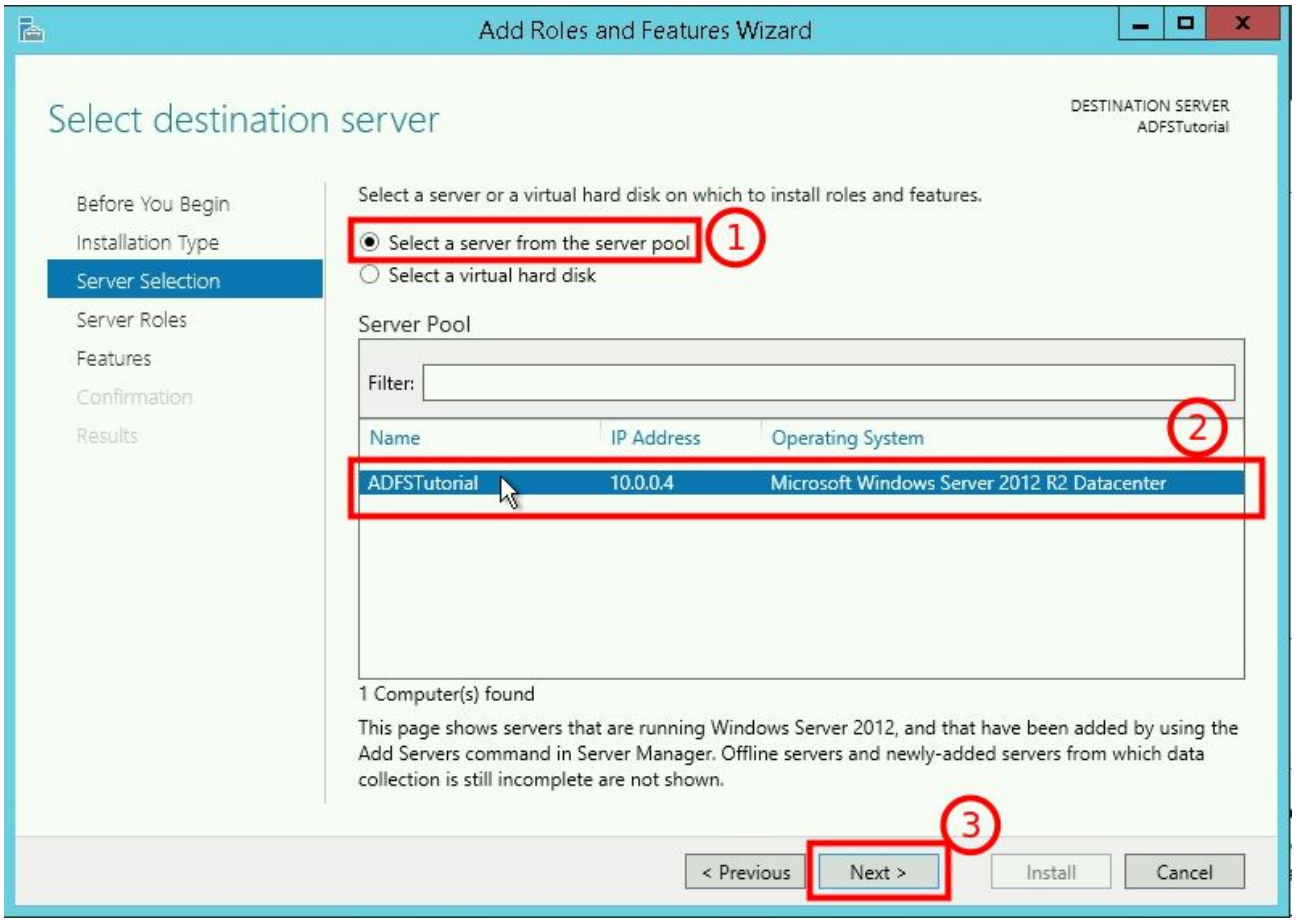
3/ On "Before You Begin" page, click "Next"



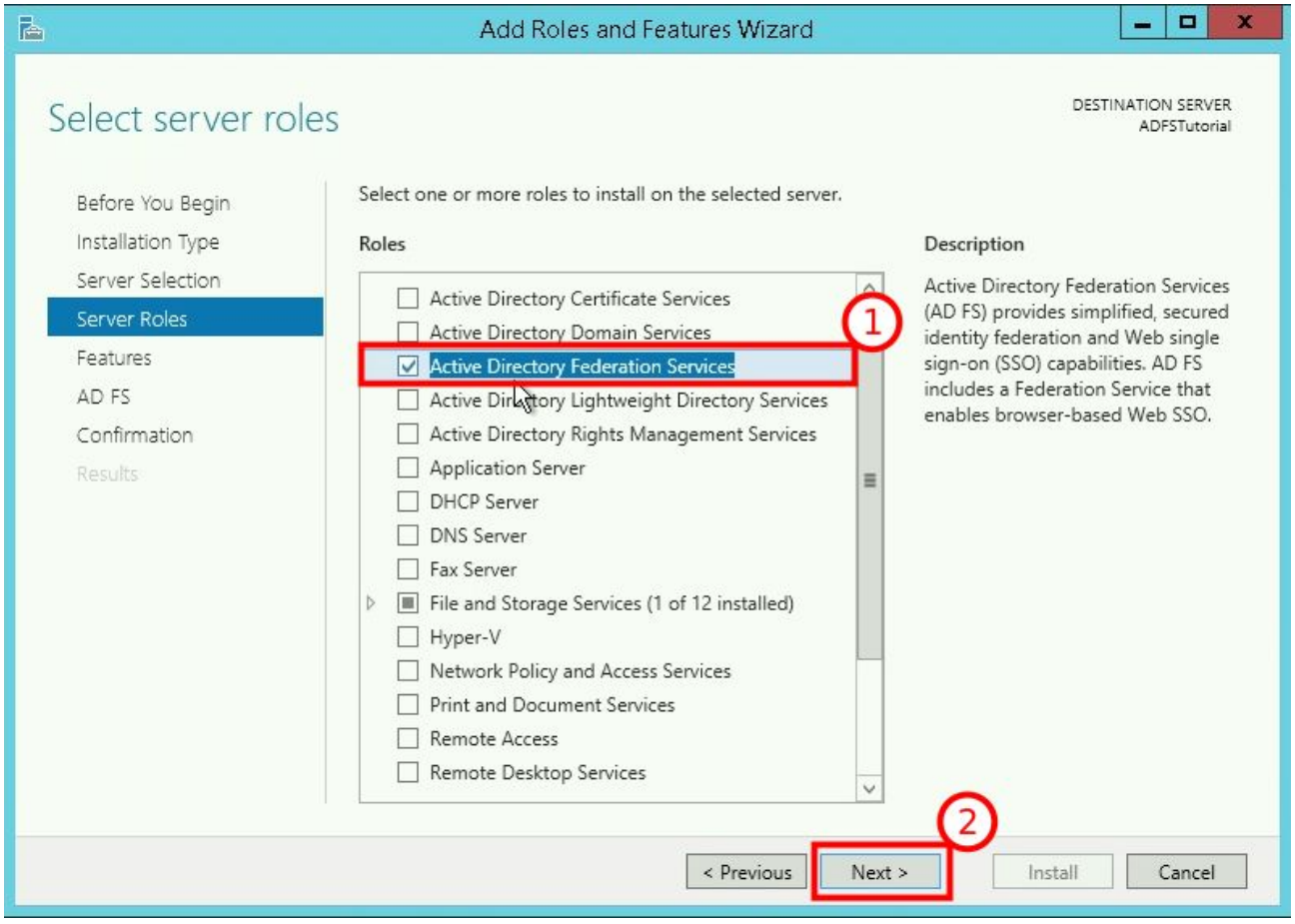
4/ On **"Installation Type"** page, please select **"Role-based or feature-based installation" (1)** and click **"Next" (2)**



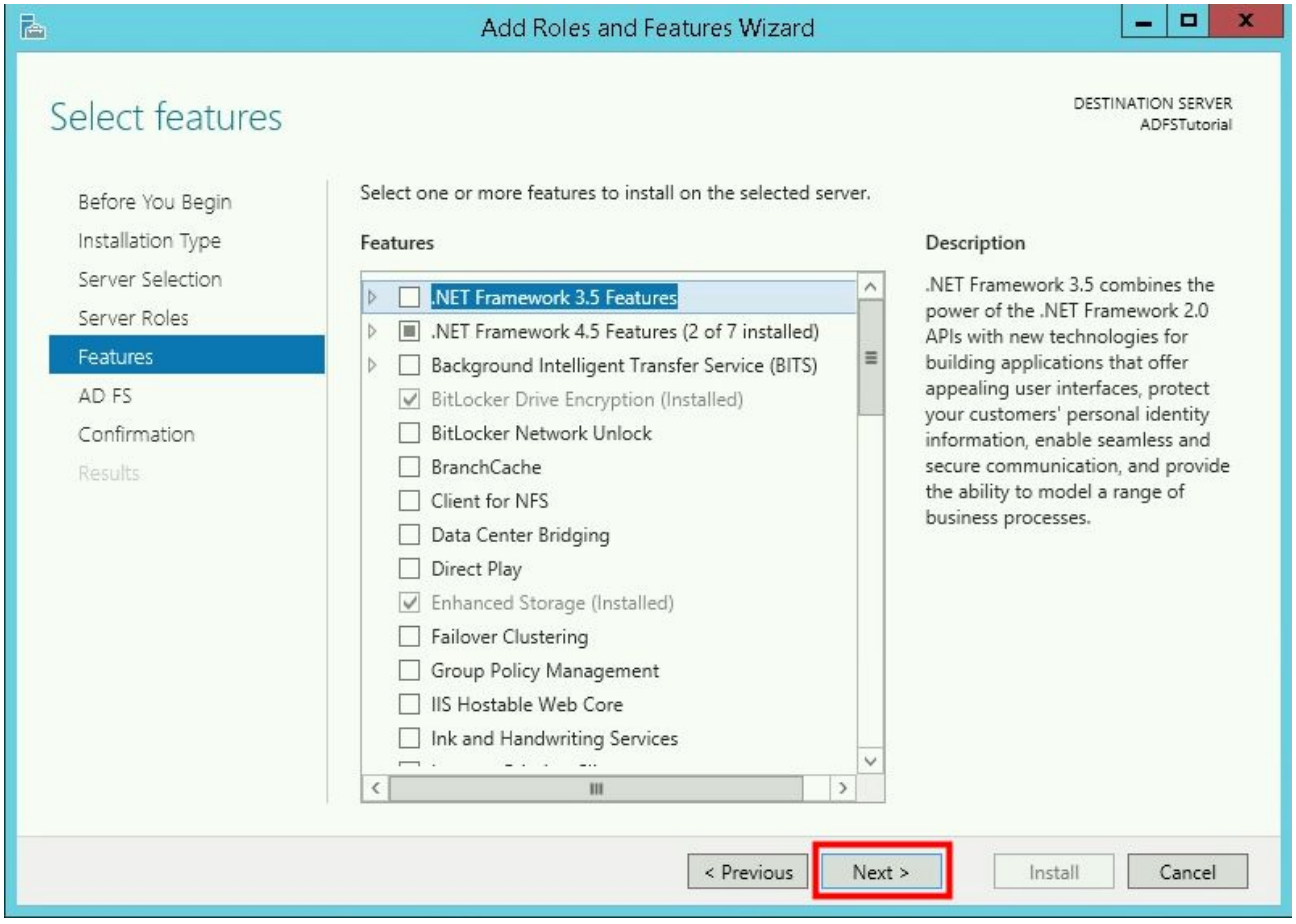
5/ On “Server Selection” page, please select “Select a server from the server pool” (1), then select your server instance from “Server Pool” list (2) and click “Next” (3)\



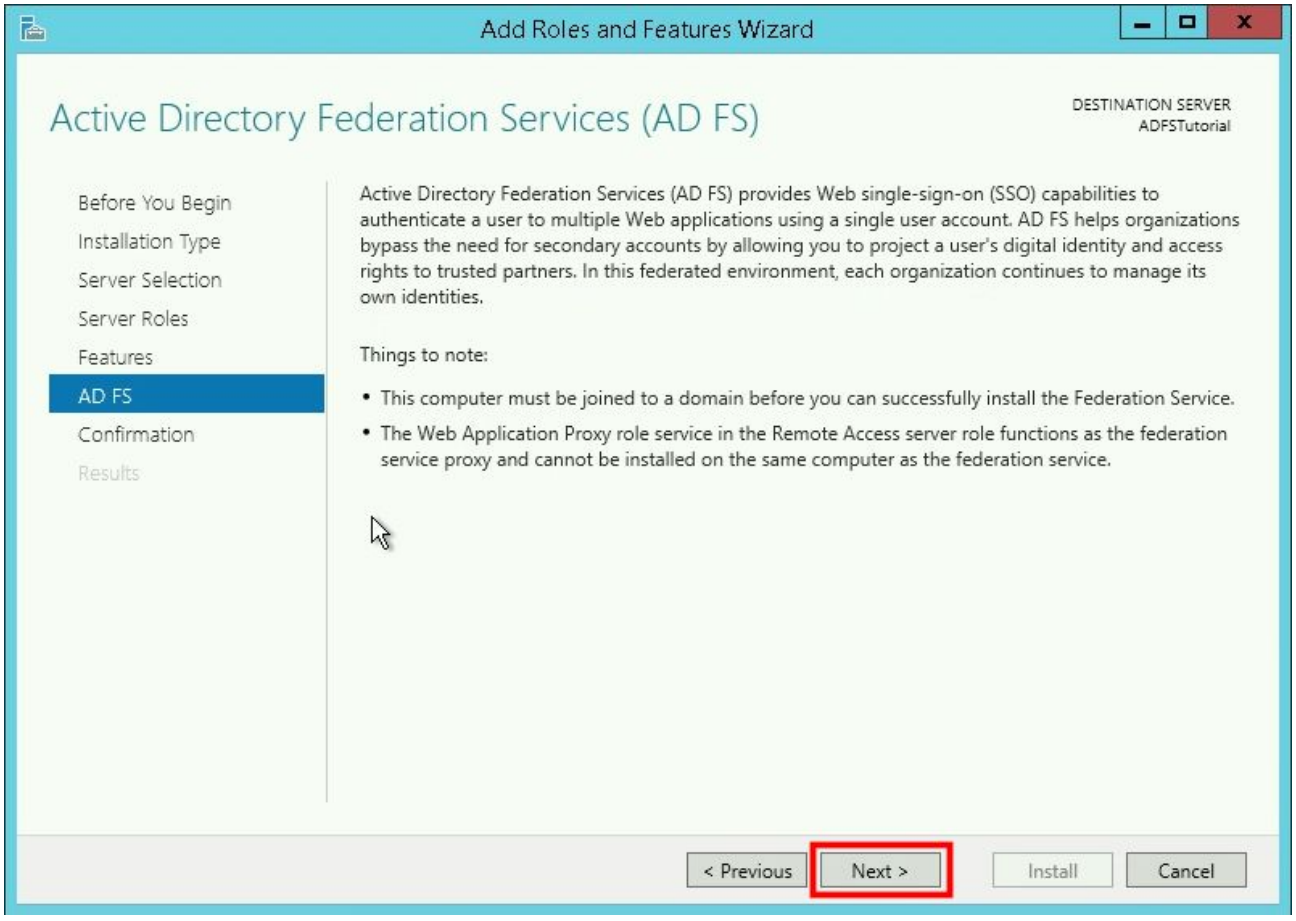
6/ On **“Server Roles”** page, please select **“Active Directory Federation Services” (1)** and click **“Next” (2)**



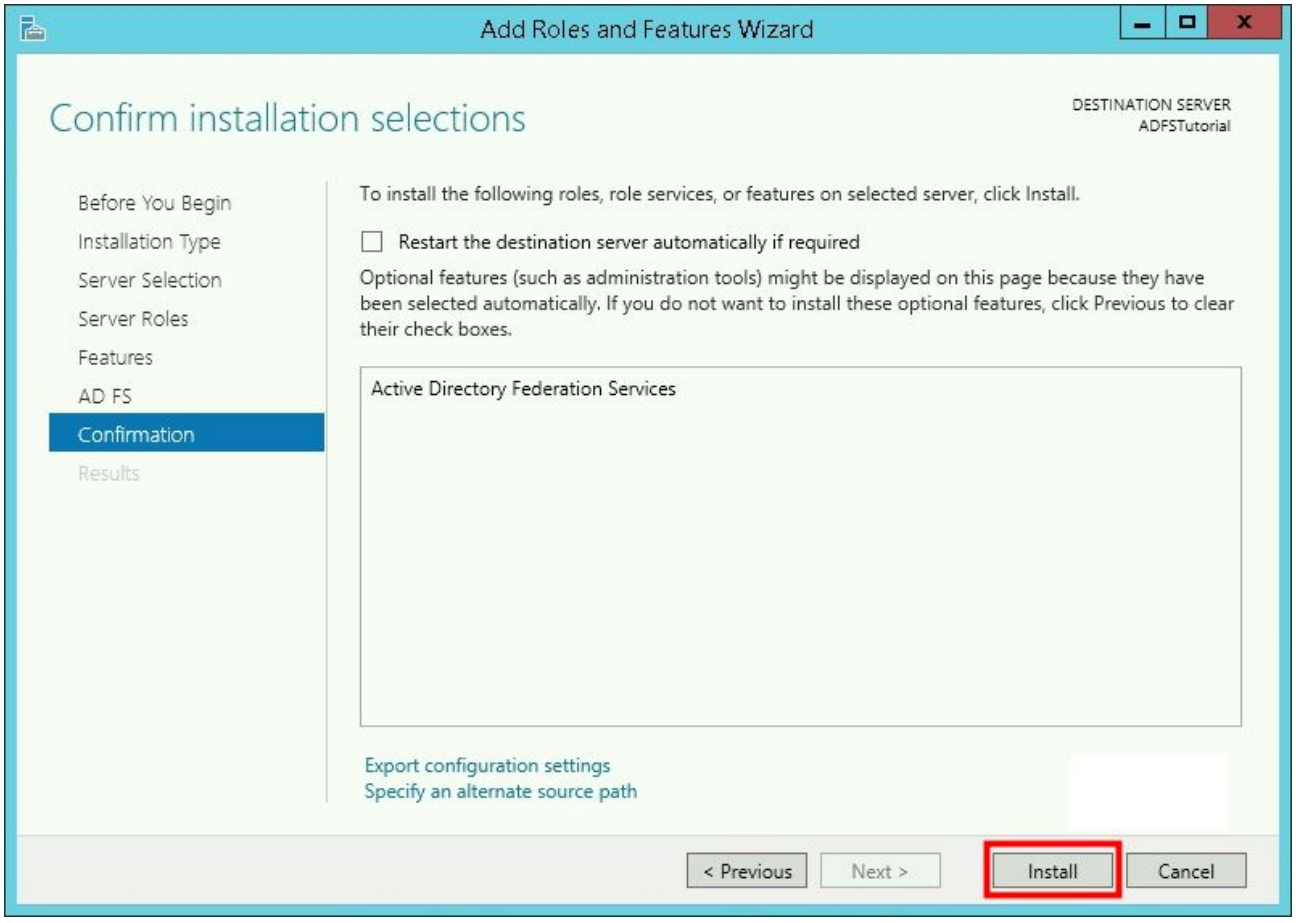
7/ On **“Features”** page, please leave default selections and click **“Next”**



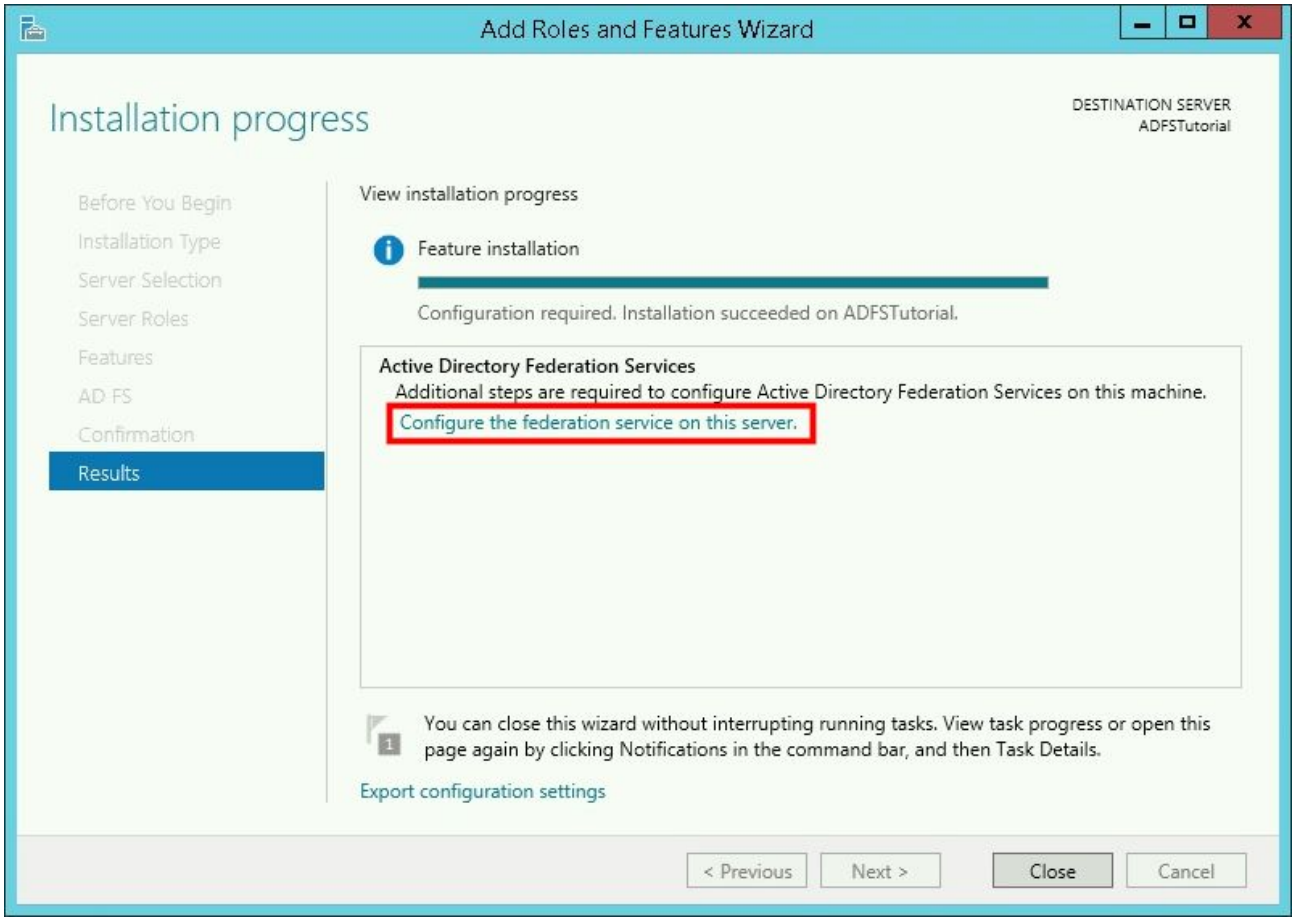
8/ On **“AD FS”** page, please click **“Next”**



9/ On **“Confirmation”** page, please click **“Install”** to begin installation process

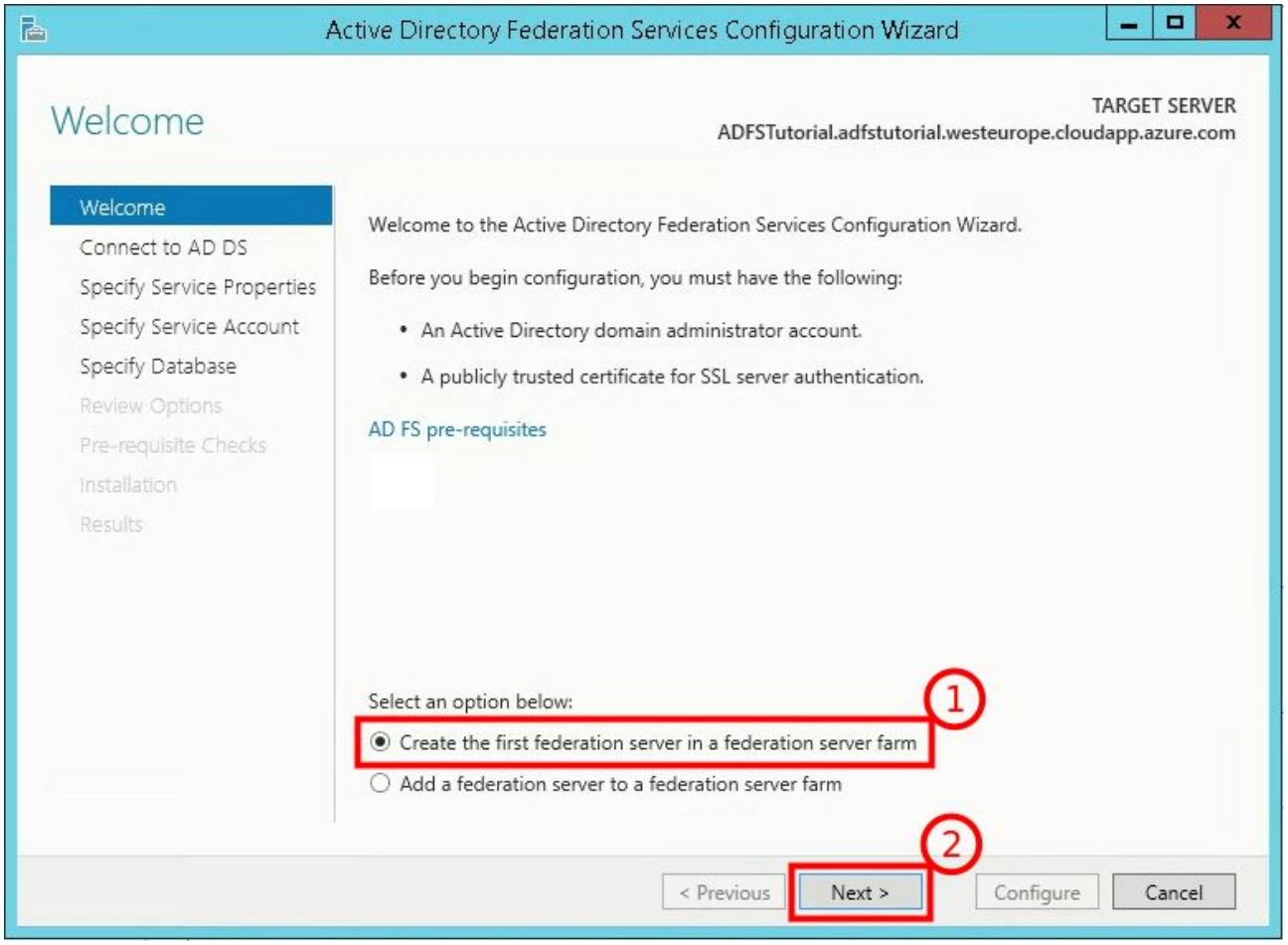


10/ When installation is finished, on **"Results"** page, please click **"Configure the federation service on this server."**

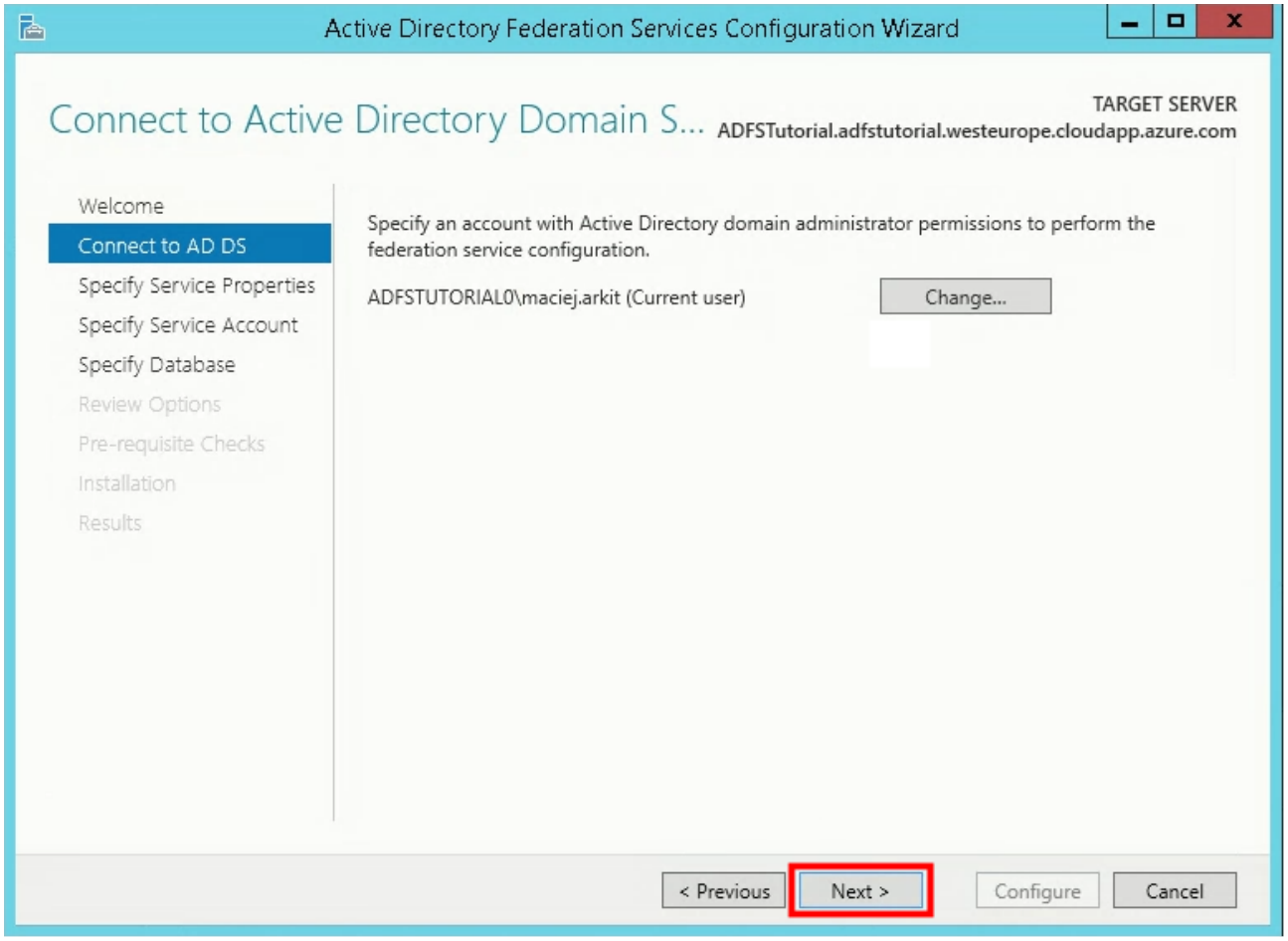


2.4 Configure AD FS service

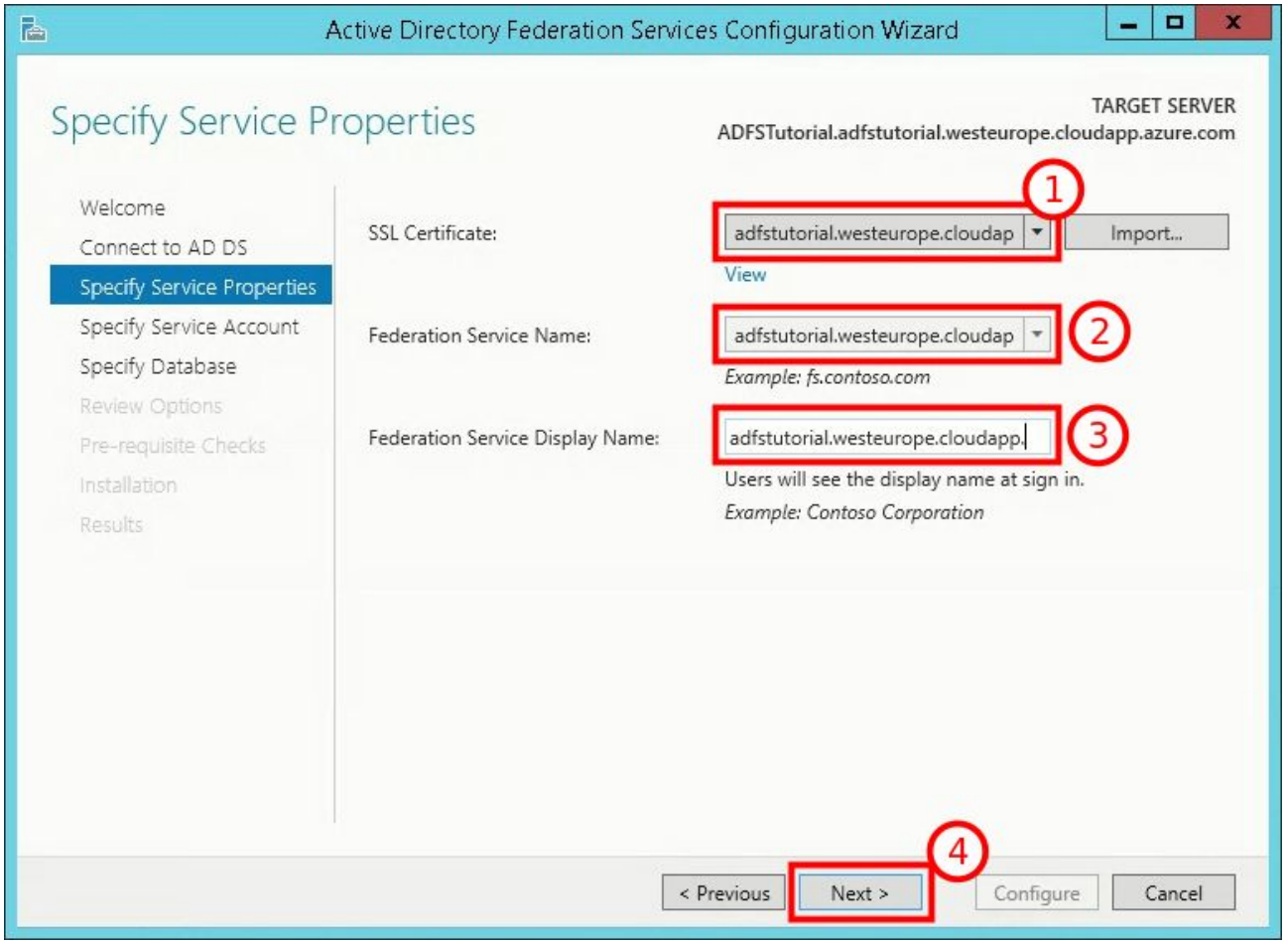
1/ On **“Welcome”** page please select **“Create the first federation server in a federation server farm” (1)** and click **“Next” (2)**



2/ On “Connect to “AD DS” page leave default selections and click **“Next”**

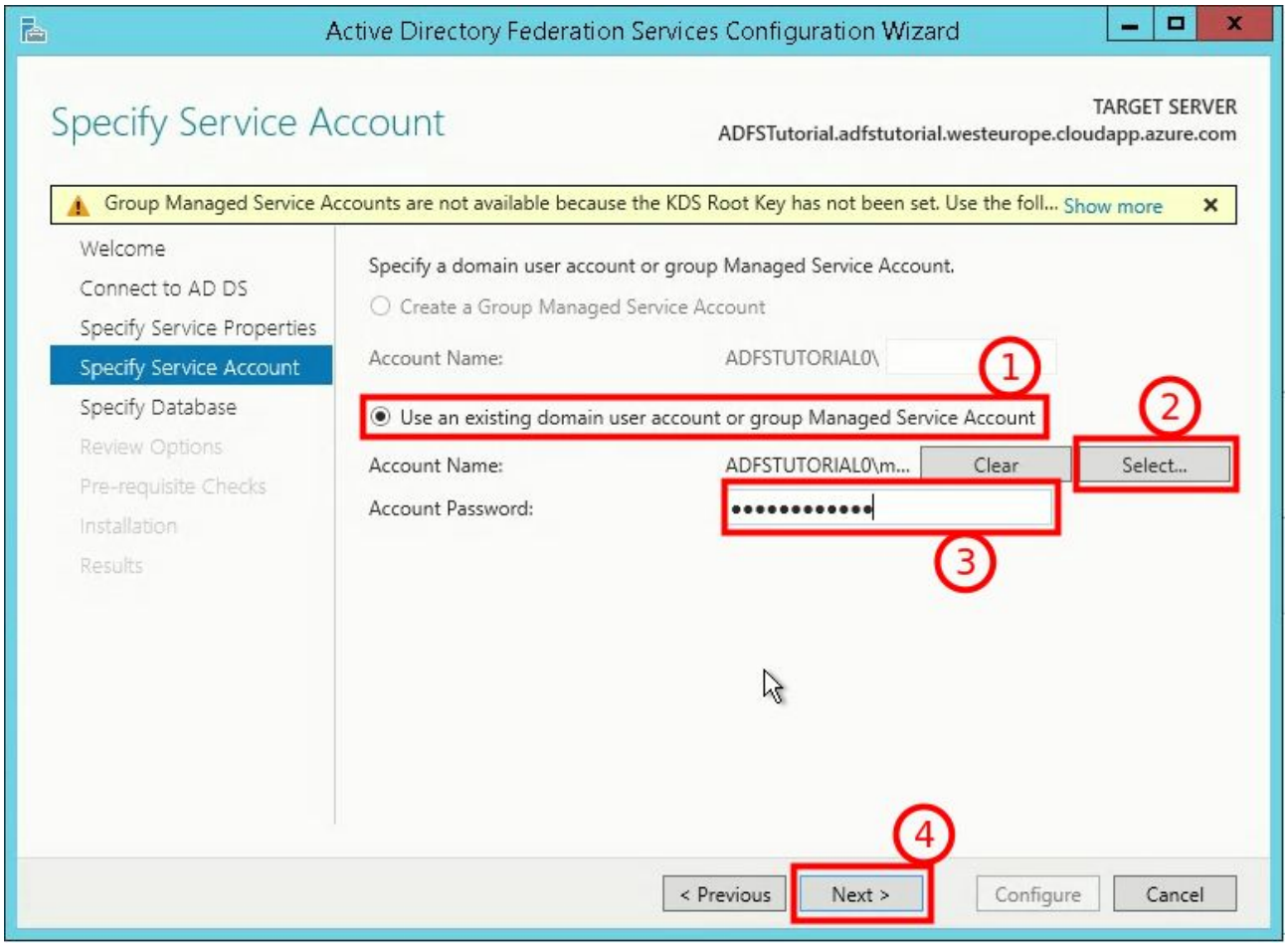


3/ On **“Specify Service Properties”** page please select “SSL Certificate” (1) [SSL certificate has been configured in 2.2 Generate Self Signed certificate for a domain].
Please select **“Federation Service Name” (2)** and provide **“Federation Service Display Name” (3)** - you can just use domain name for simplicity. Please click **“Next” (4)**

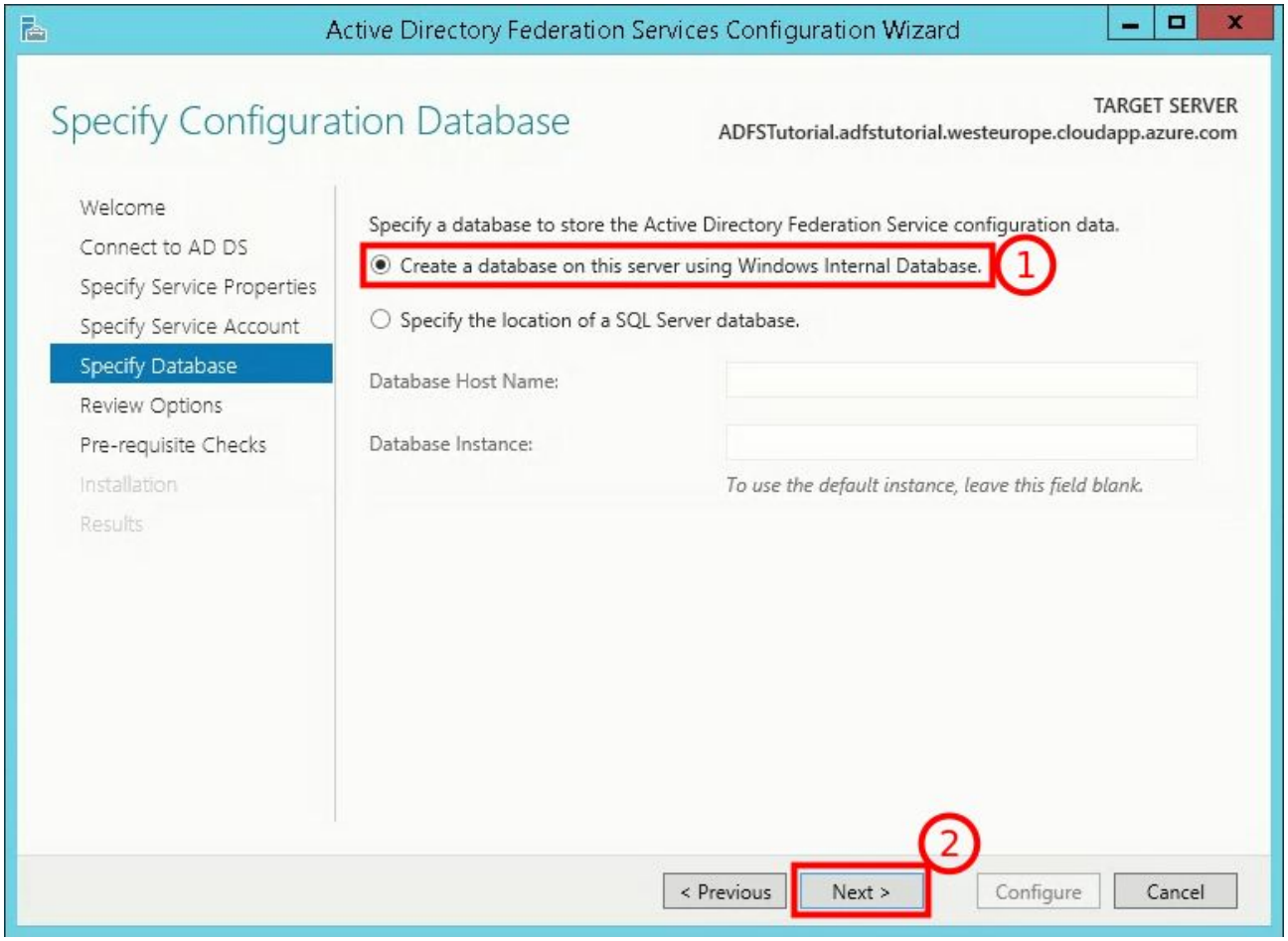


4/ On **"Specify Service Account"** page, please select **"Use an existing domain user account or group Managed Service Account" (1)** and then click **"Select" (2)**. You will be asked to provide account name. Please provide your current user name, and then please provide your password **(3)**.

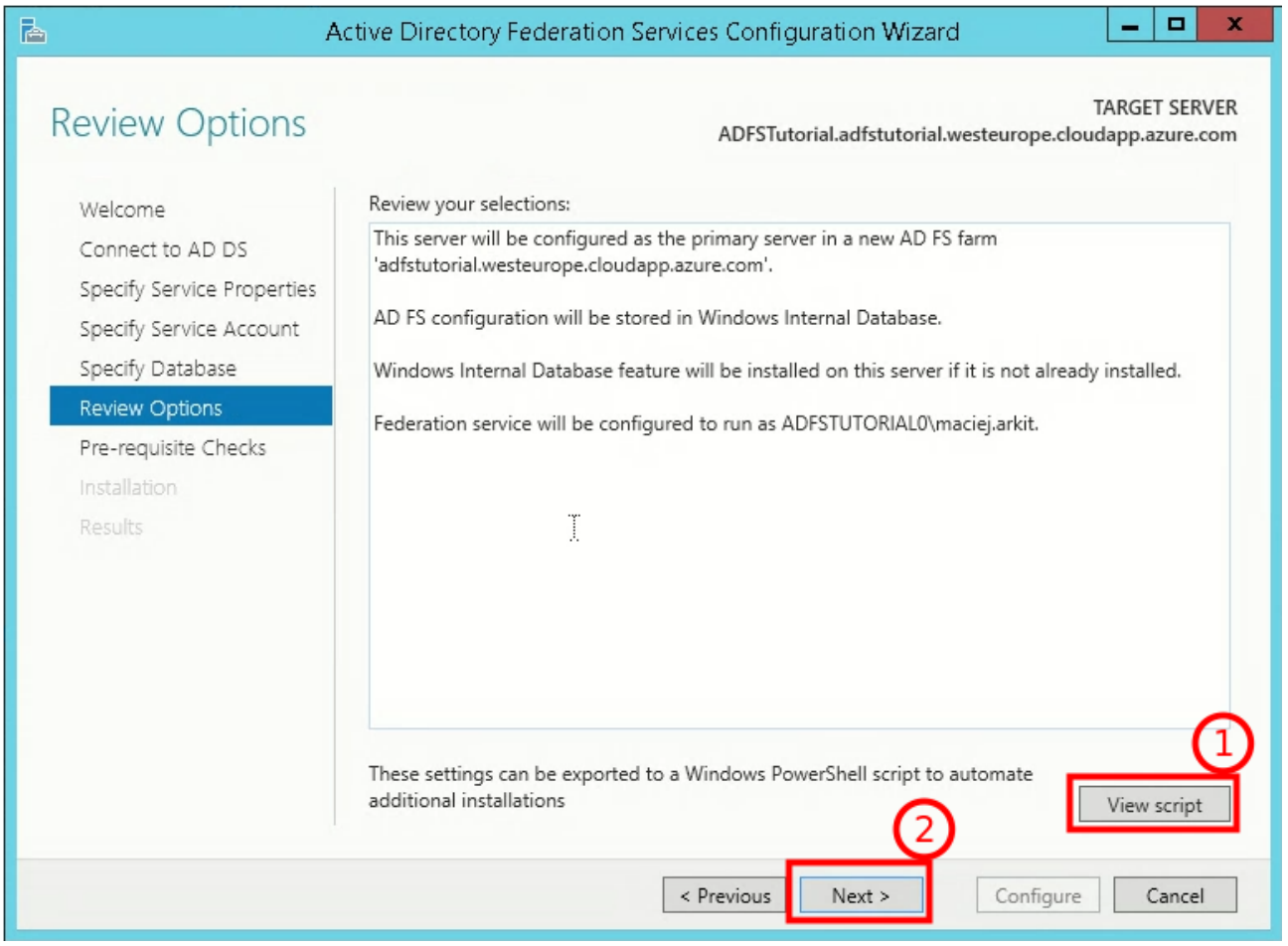
Then please click **“Next” (4)**



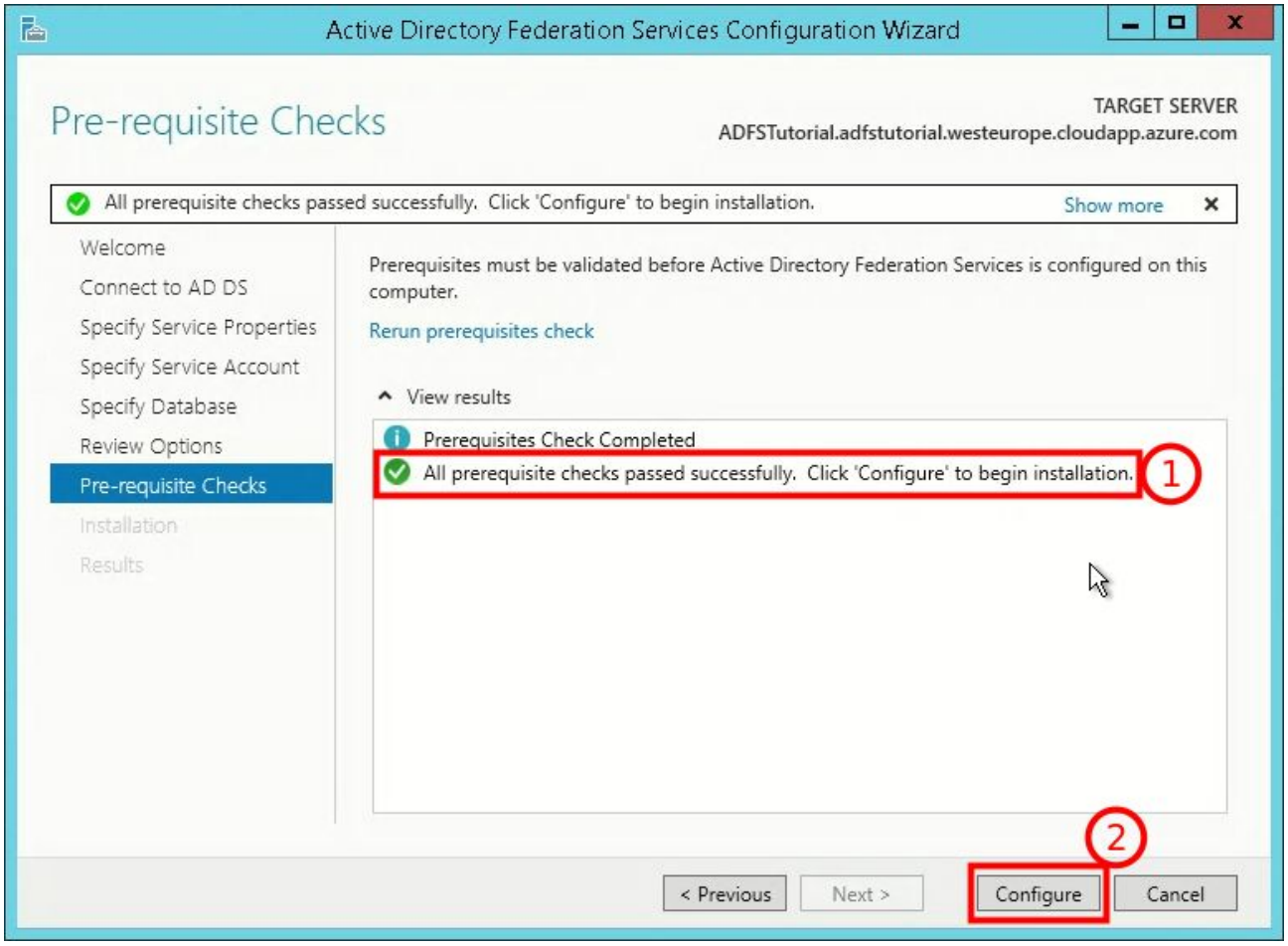
5/ On “Specify Database” page, please select **“Create a database on this server using Windows Internal Database” (1)** and click **“Next” (2)**



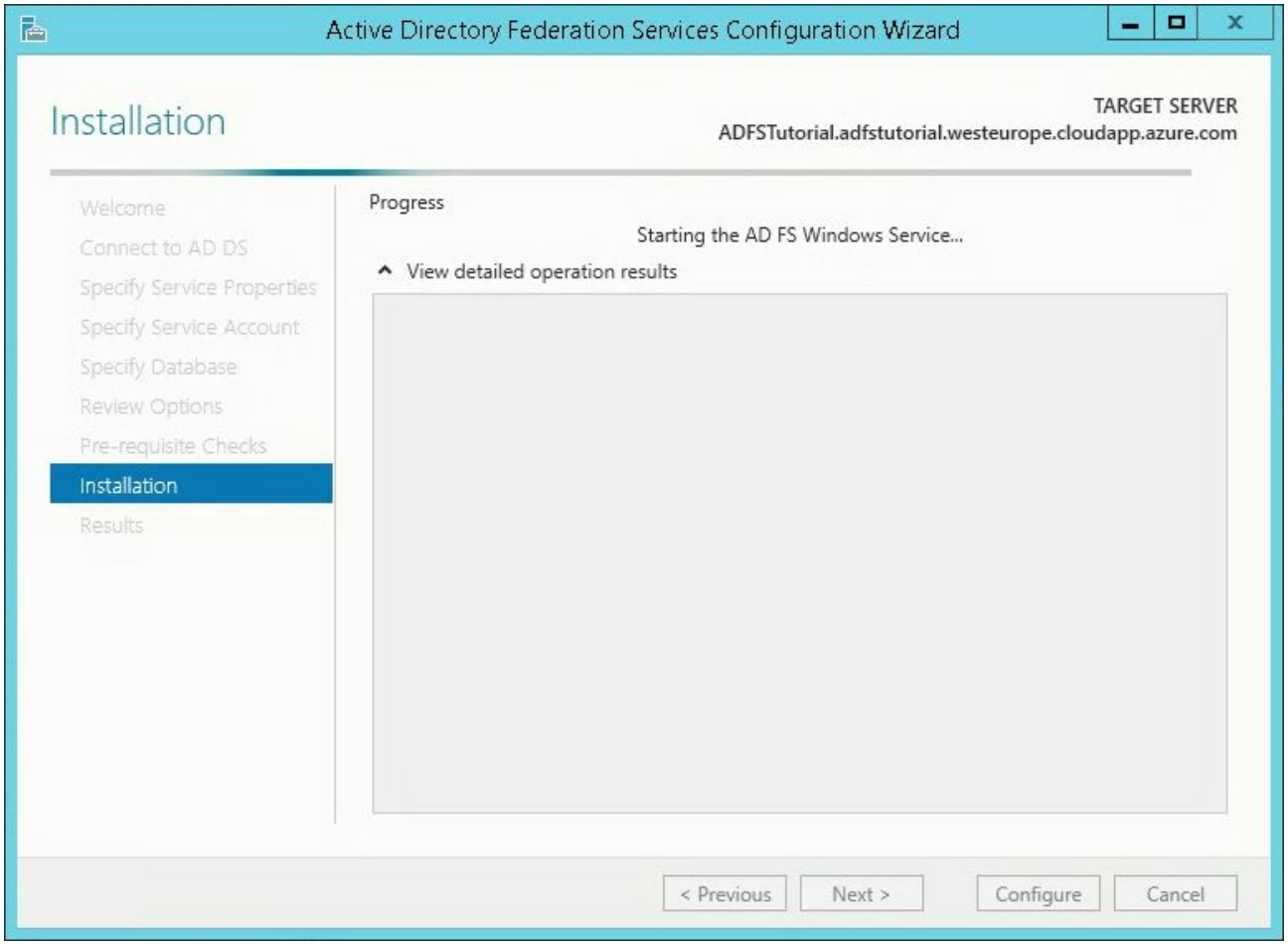
6/ On **“Review Options”** page you can click **“View script” (1)** and save it for reference or to use for further installations as a PowerShell script. Then please click **“Next” (2)**.



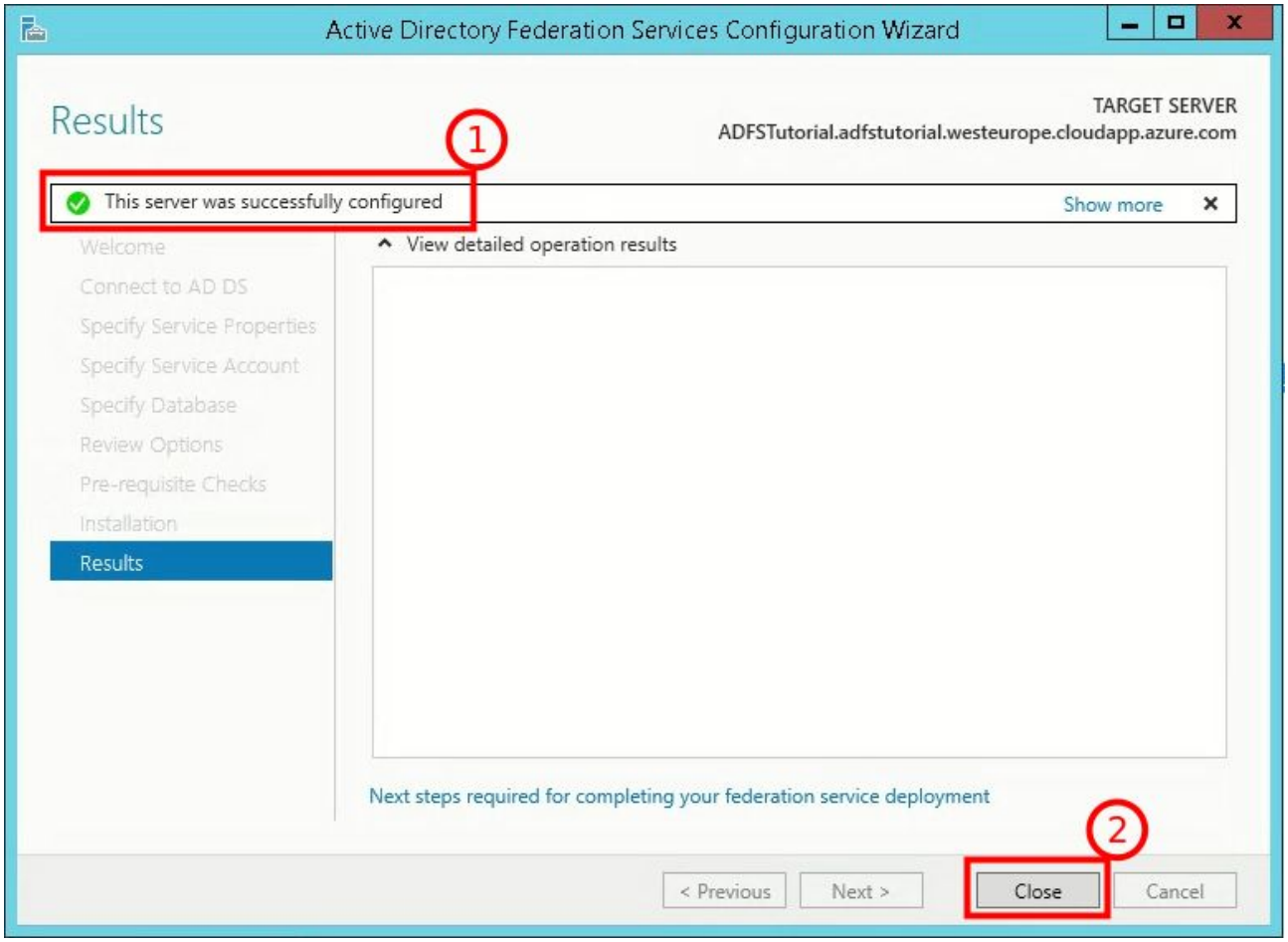
7/ On “Pre-requisite Checks” page, please ensure that all checks passed **(1)** and then please click **“Configure” (2)**

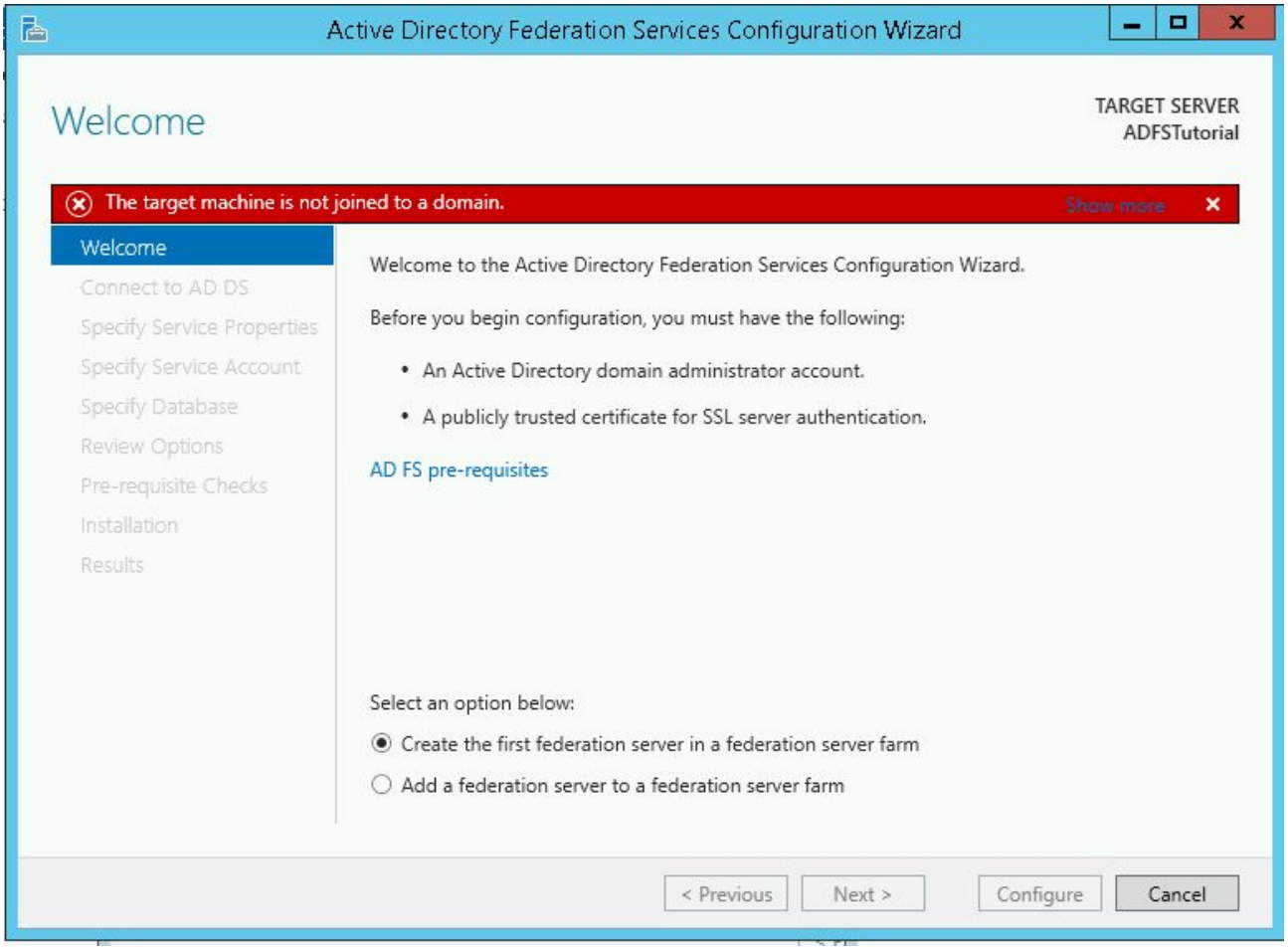


8/ On “Installation” page, please stand by while installation is in progress. It may take couple minutes to finish.



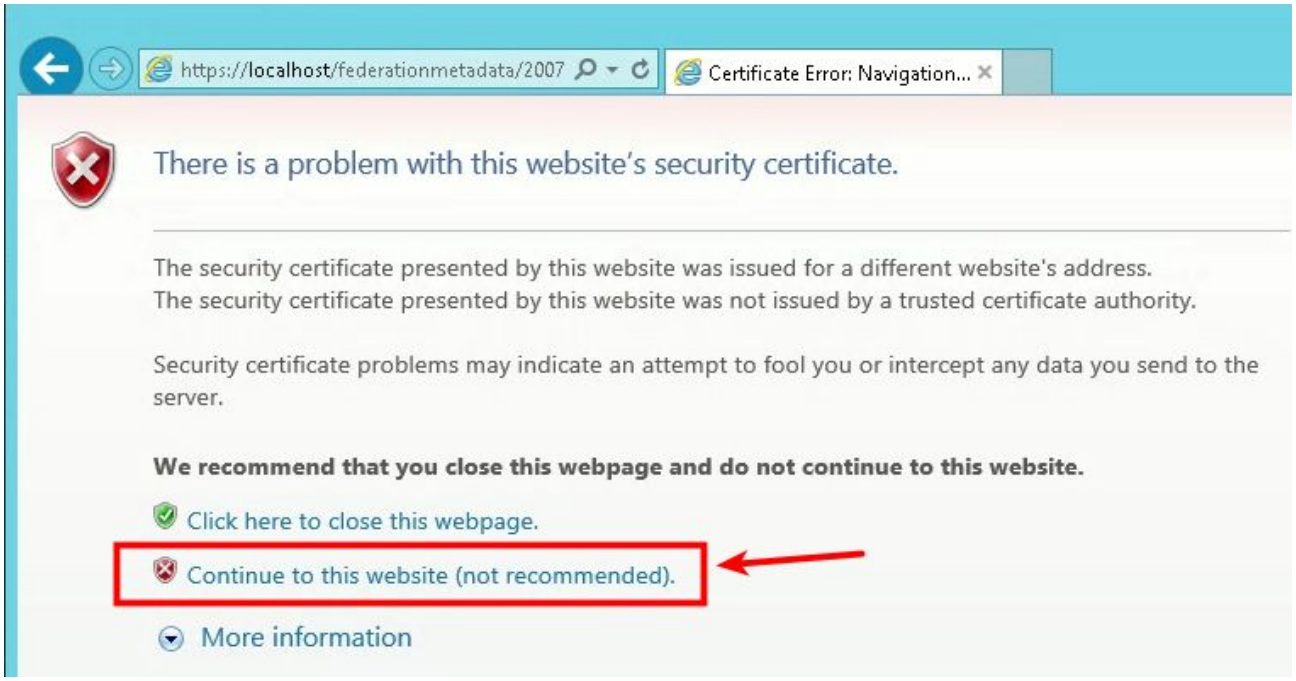
9/ When installation will be finished, you will see “Results” page. Ensure that installation was successful **(1)**, and click **“Close” (2)**


Active Directory Federation Services Configuration Wizard
Results
TARGET SERVER
ADFSTutorial.adfstutorial.westeurope.cloudapp.azure.com
1
This server was successfully configured
Show more
Welcome
Connect to AD DS
Specify Service Properties
Specify Service Account
Specify Database
Review Options
Pre-requisite Checks
Installation
Results
View detailed operation results
Next steps required for completing your federation service deployment
2
< Previous
Next >
Close
Cancel

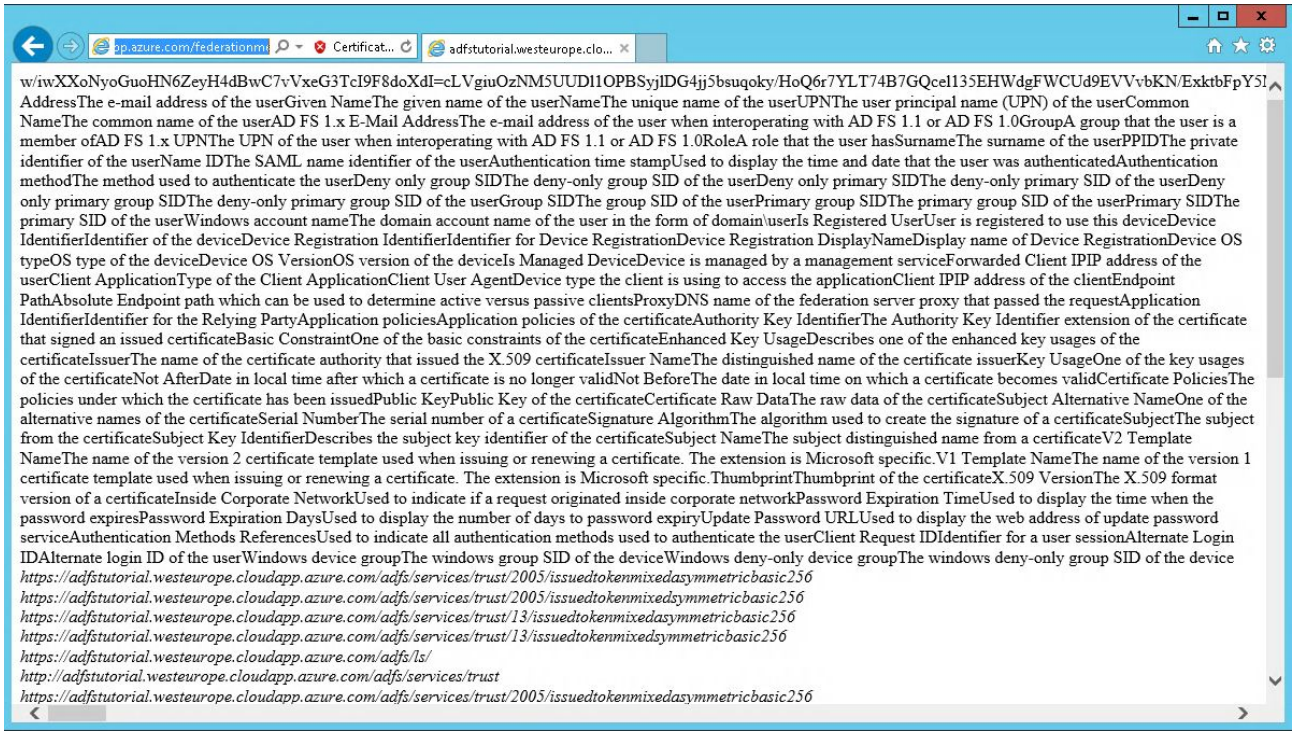


2.5 Verify ADFS installation

Please visit: https://localhost/federationmetadata/2007-06/federationmetadata.xml to verify AD FS setup completeness.
(NOTE: Due to fact that we use self signed SSL certificate, you will see warning. Please ignore this and click “Continue to this website”)



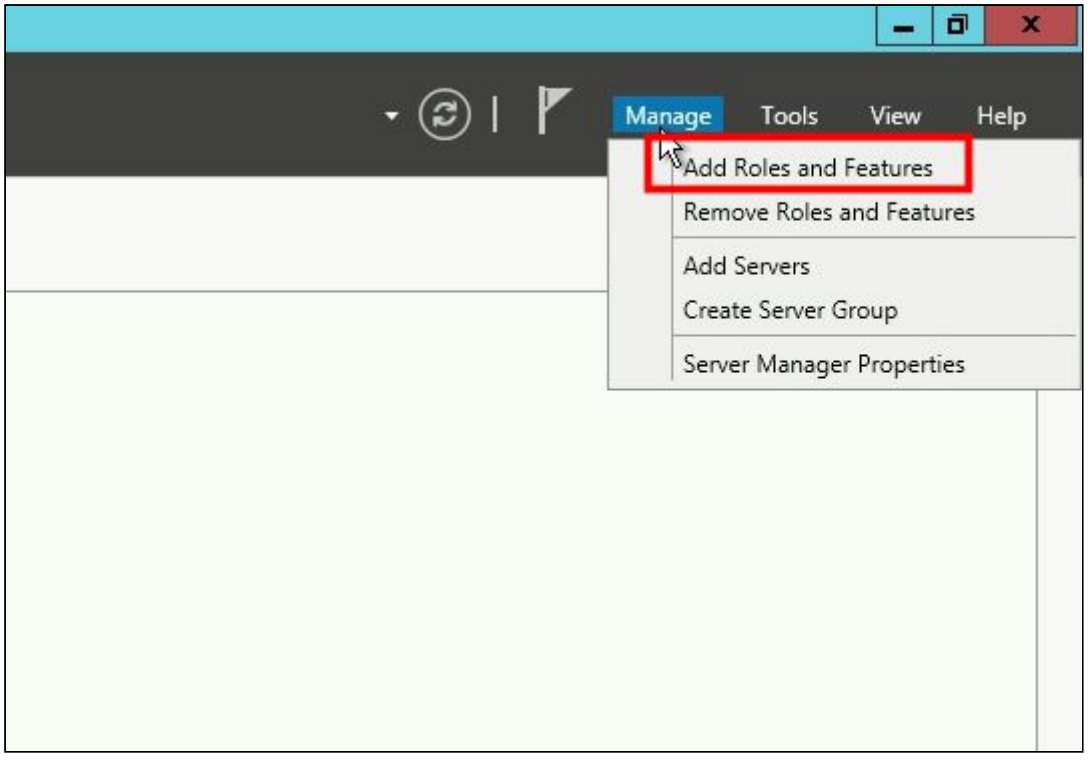
If everything went well you should see federation metadata:



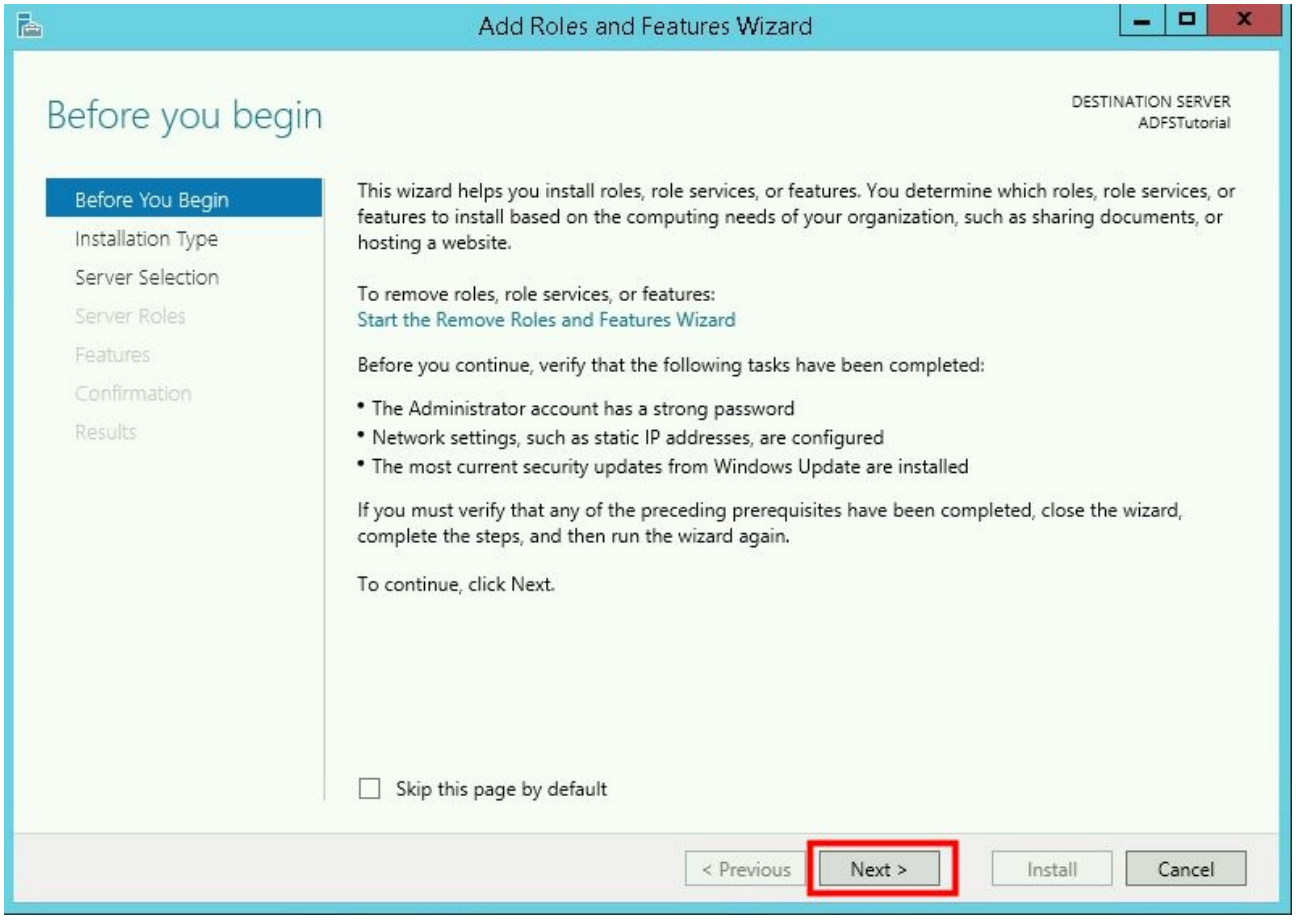
3. Configure IIS server

3.1. Install IIS Server

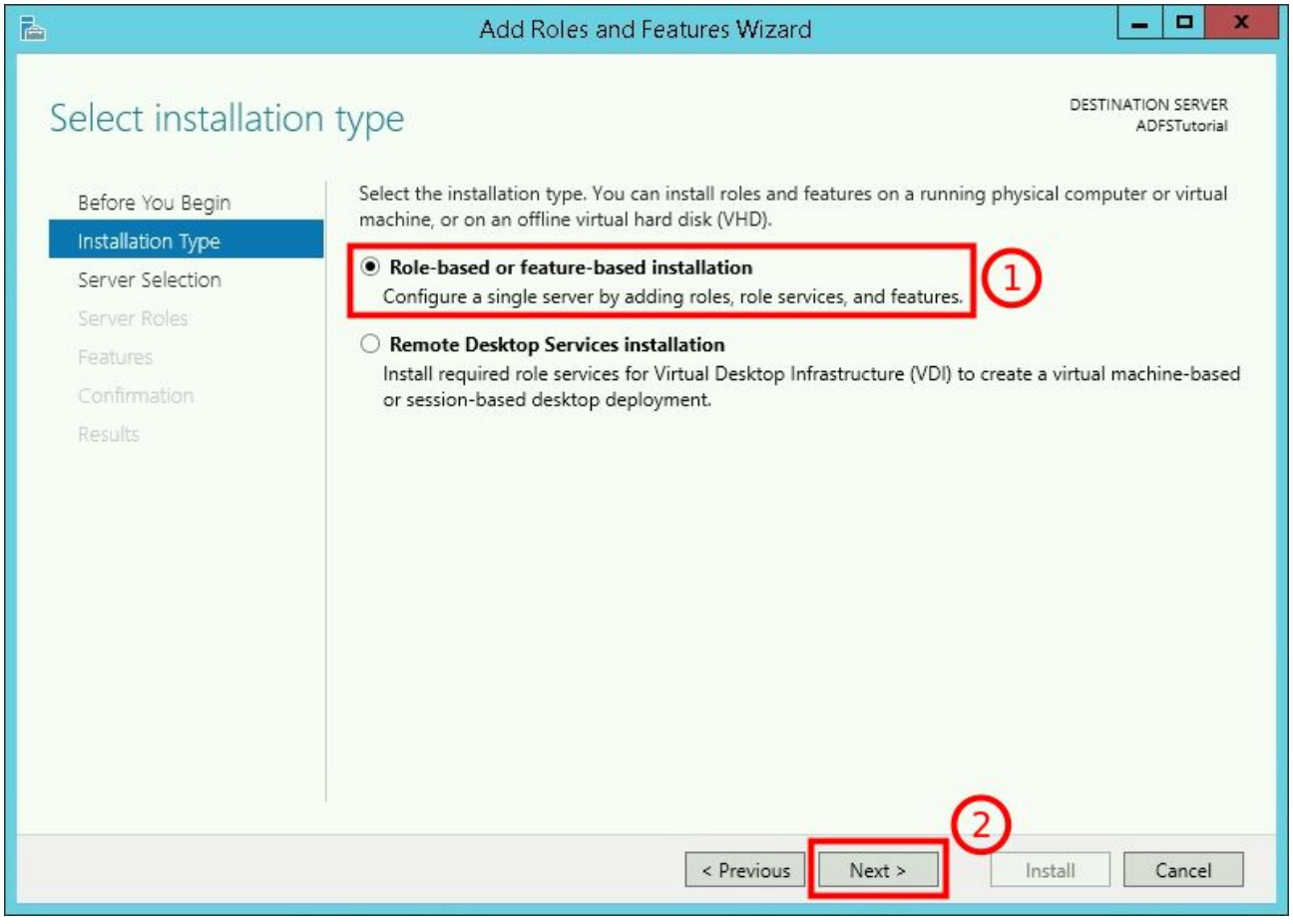
1/ Open “Server Manager”
2/ Click “Manage” and then “Add Roles and Features” in right upper corner:



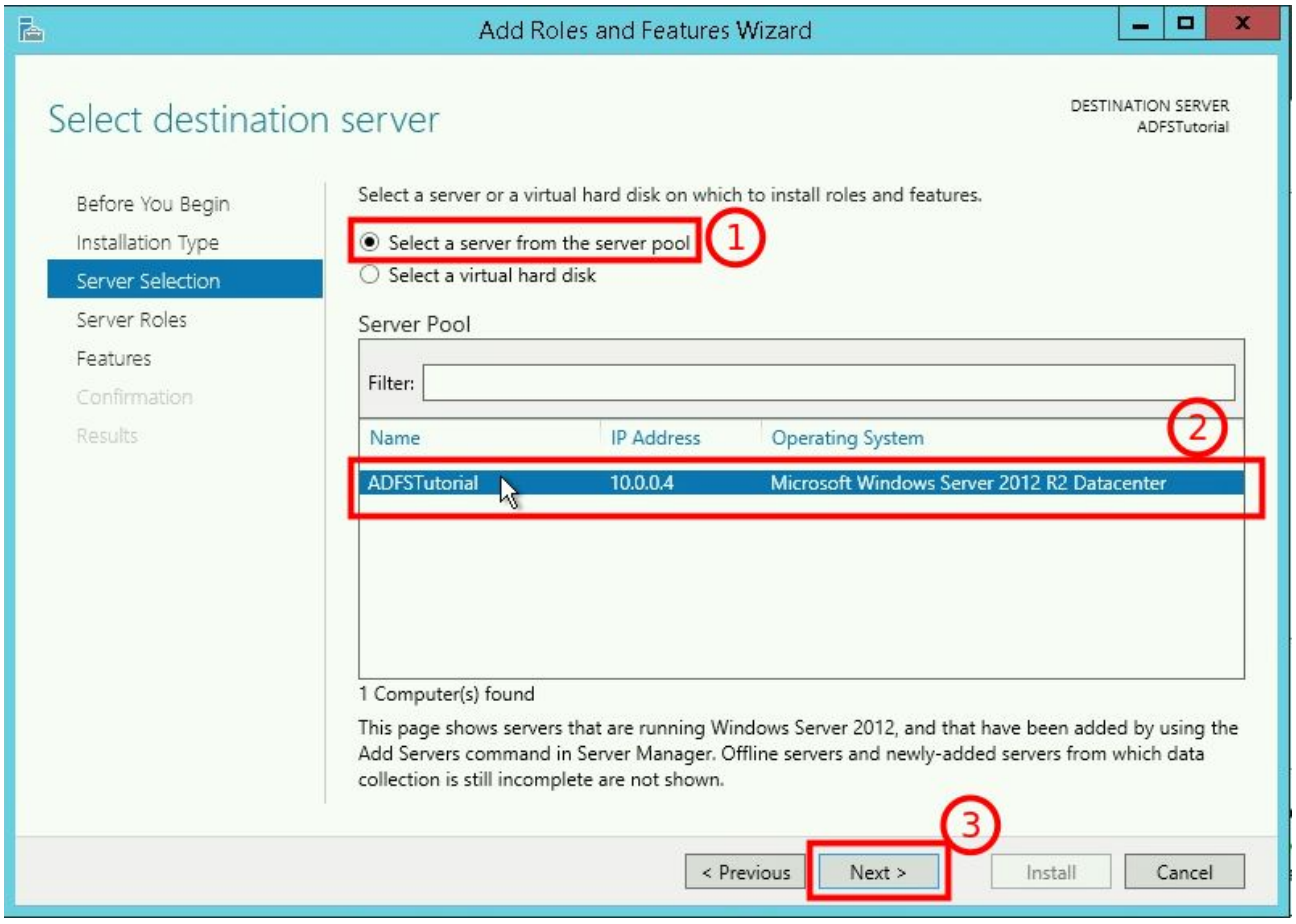
3/ On “Before You Begin” page, click “Next”



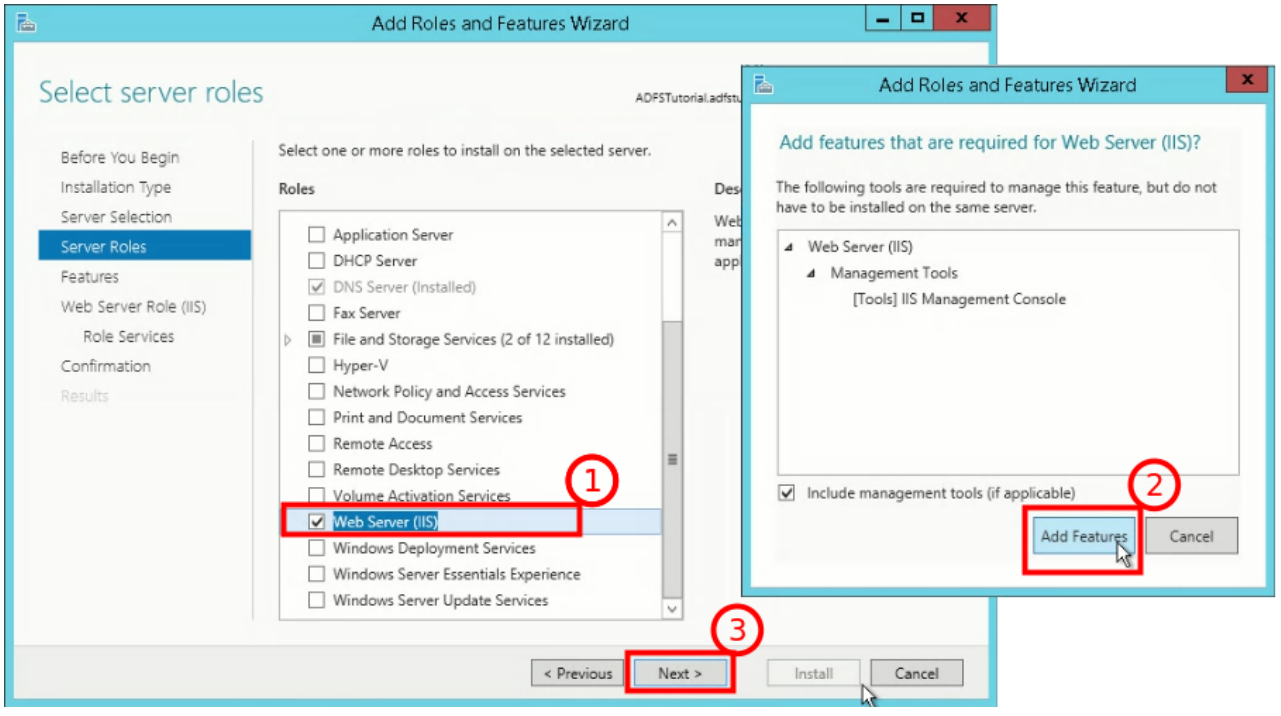
4/ On **"Installation Type"** page, please select **"Role-based or feature-based installation" (1)** and click **"Next" (2)**



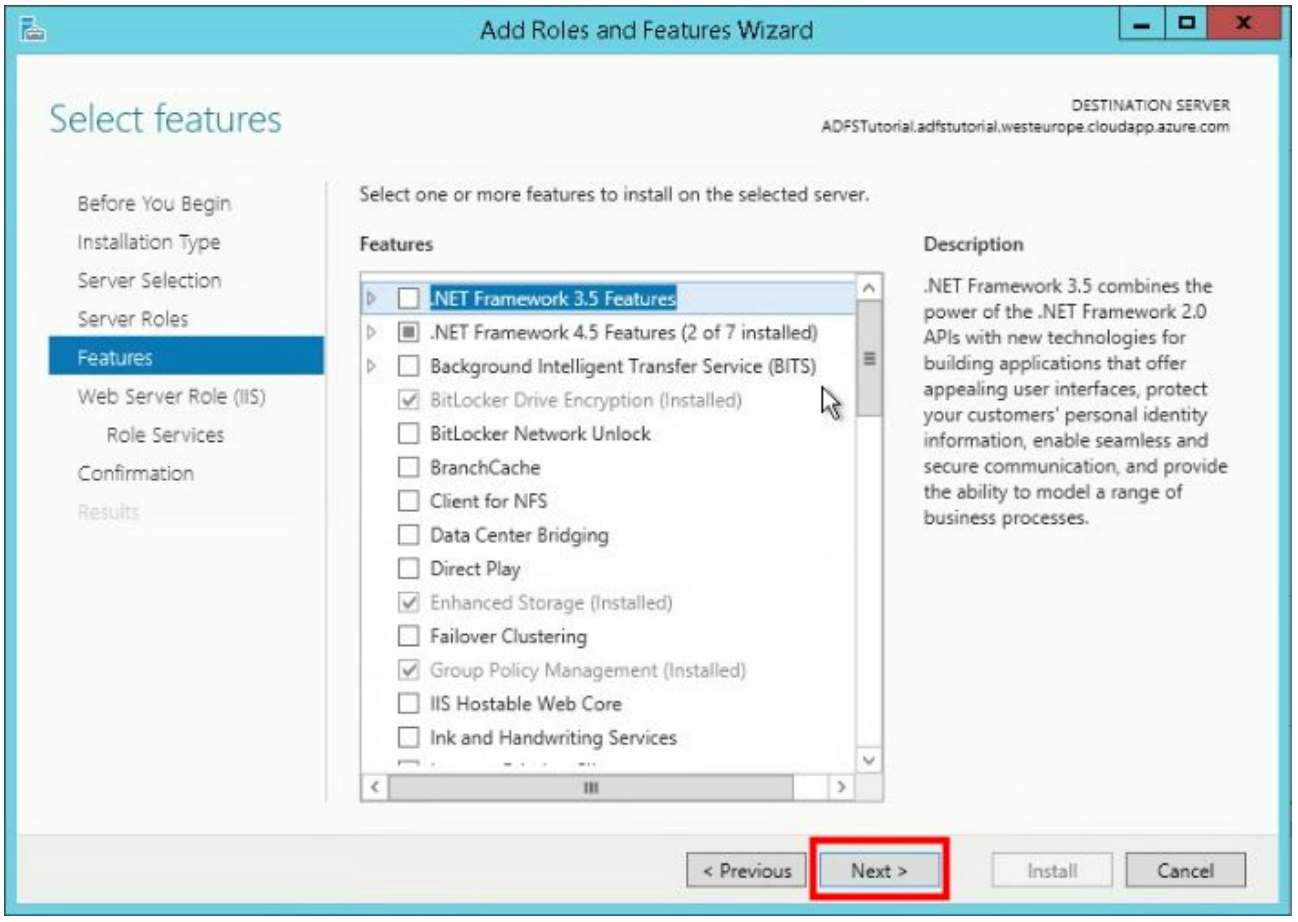
5/ On “Server Selection” page, please select “Select a server from the server pool” (1), then select your server instance from “Server Pool” list (2) and click “Next” (3)\



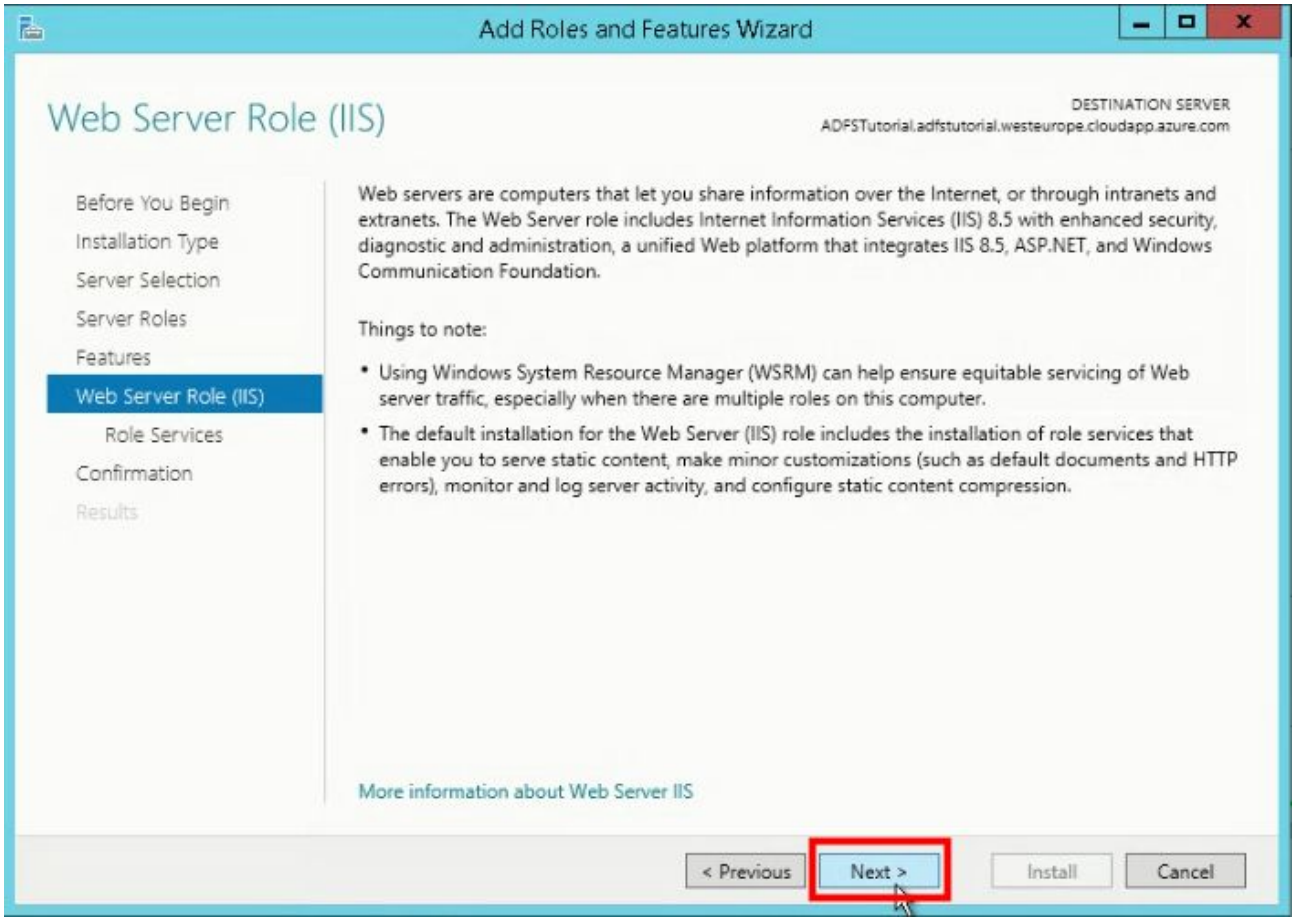
6/ On **“Server Roles”** page, please select **“Web server (IIS)” (1). When asked “Add features that are required for Web Server (IIS)?”, please confirm and click “Add Features”(2).** Then please click **“Next” (3)**



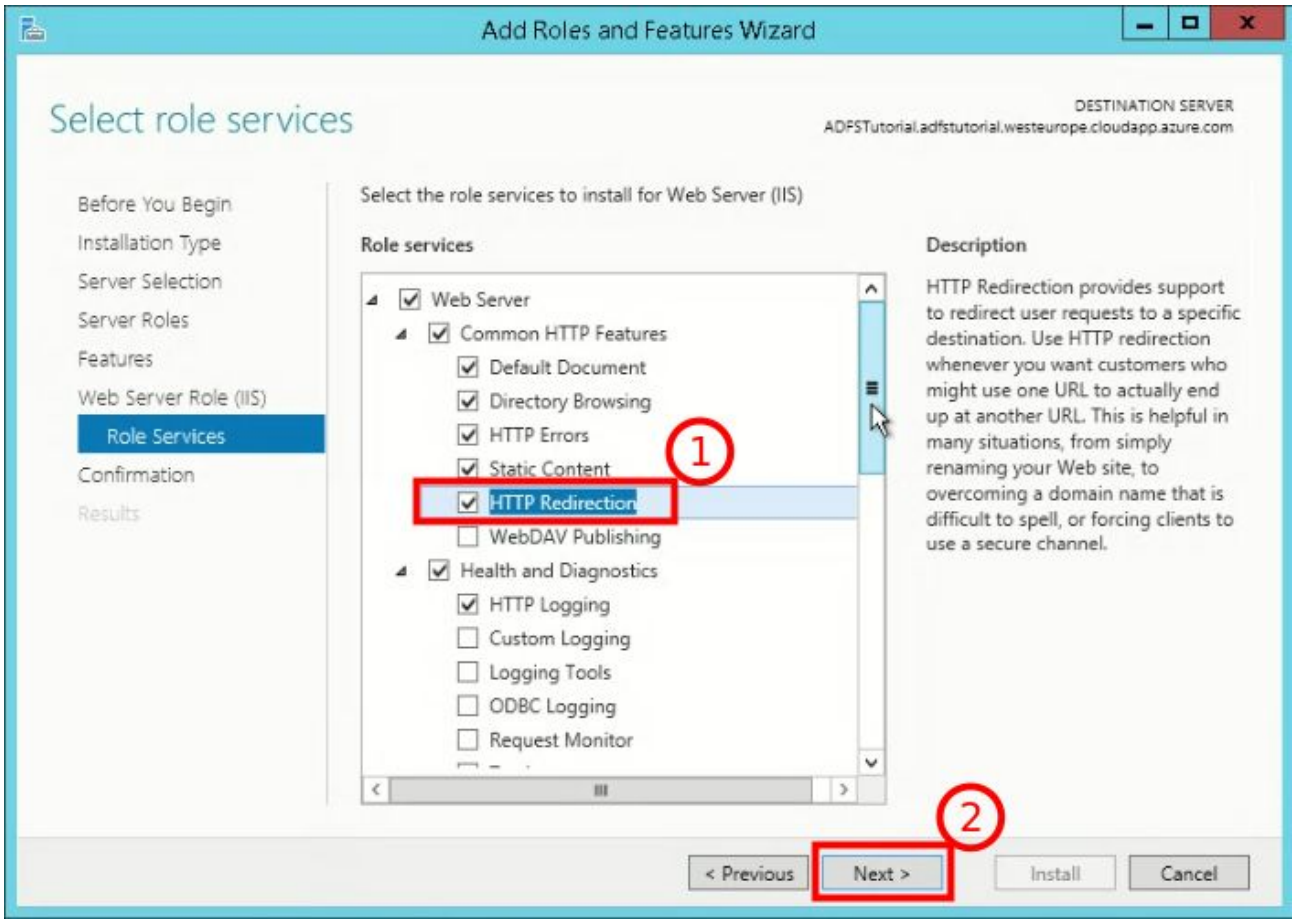
7/ On **“Features”** page, please leave default selections and click **“Next”**



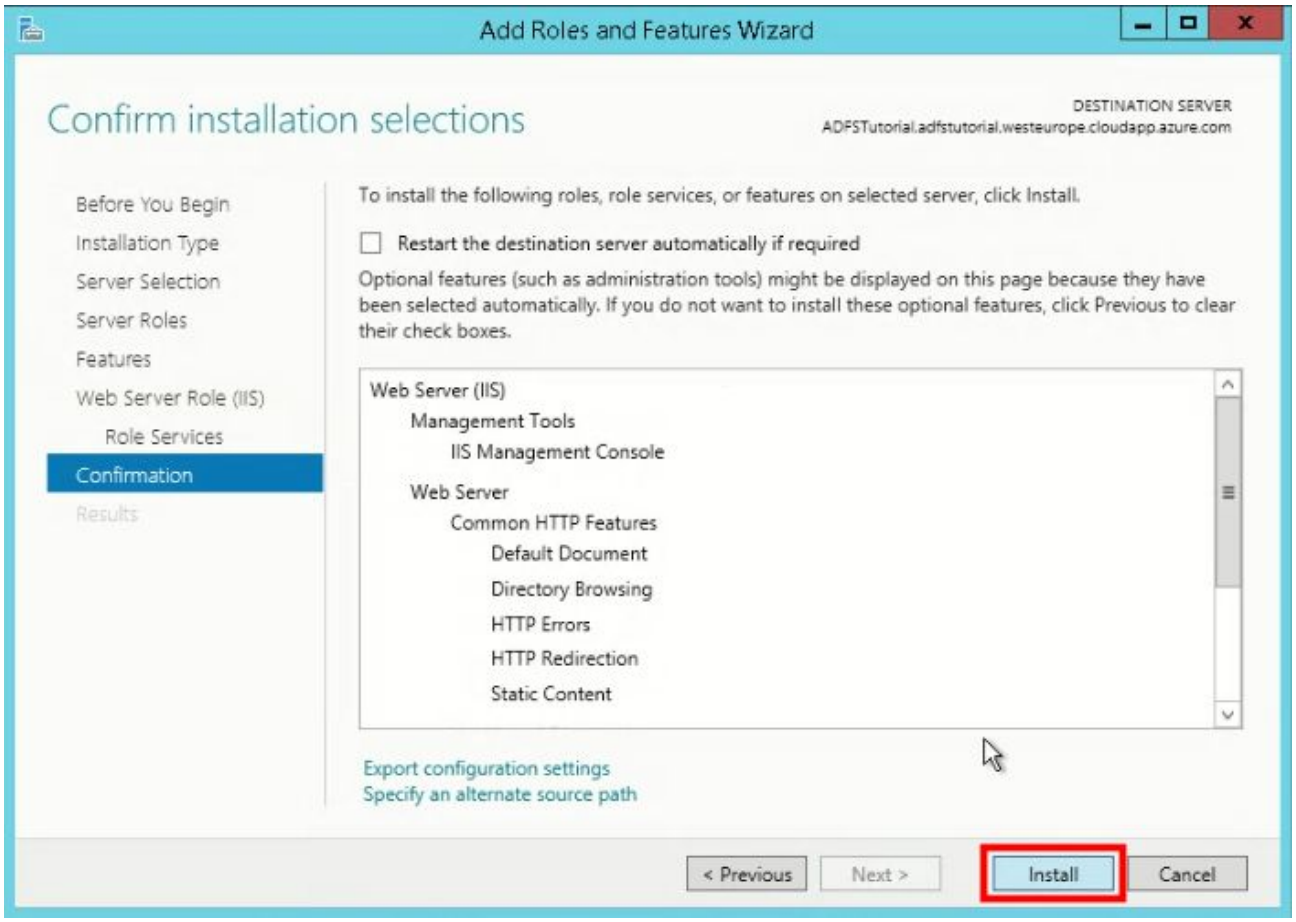
8/ On **“Web Server Role (IIS)”** please click “Next”



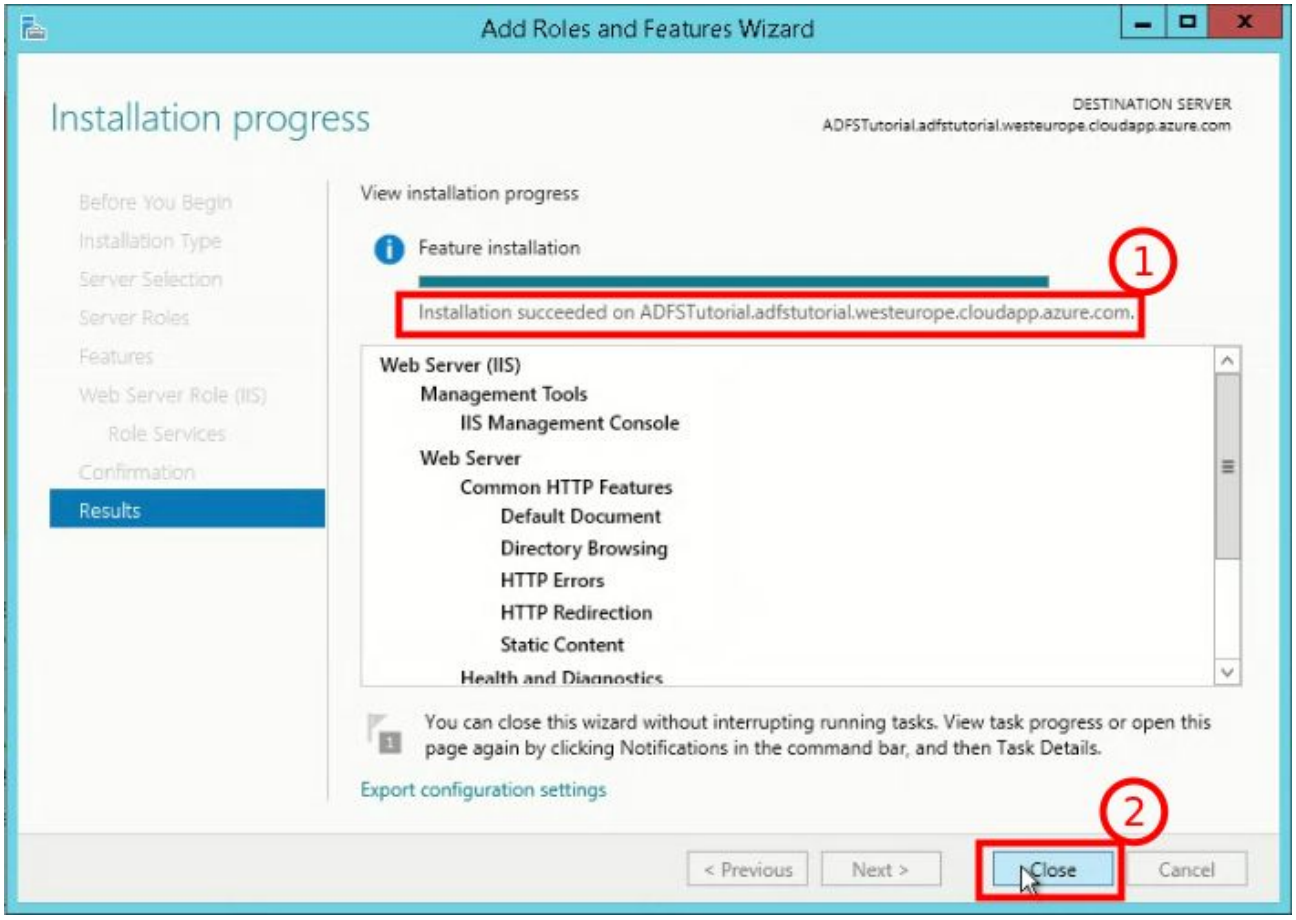
9/ On “Role Services” page, please select **“HTTP Redirection” (1)** and click **“Next” (2)**



10/ On **“Confirmation”** page, please click **“Install”**, and please stand by while installation is in progress.



11/ Installation may take couple minutes. When installation is finished **(1)**, ensure that it is successful, and click **“Close” (2)**



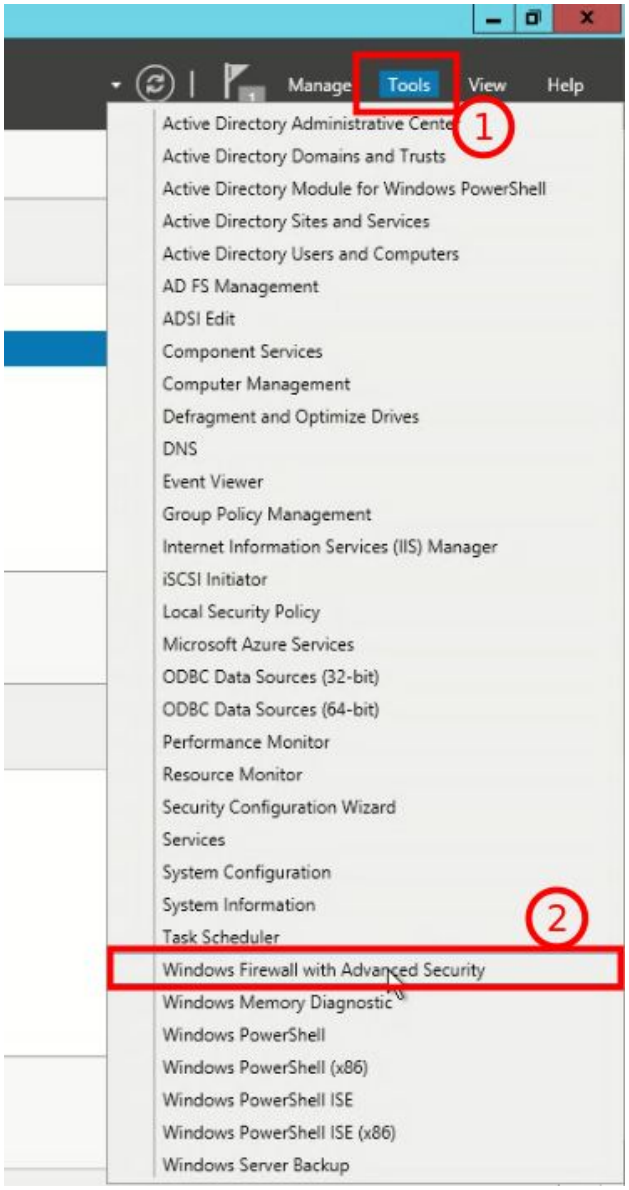
IIS Server is installed, now we can configure it.

3.2. Configure Firewalls

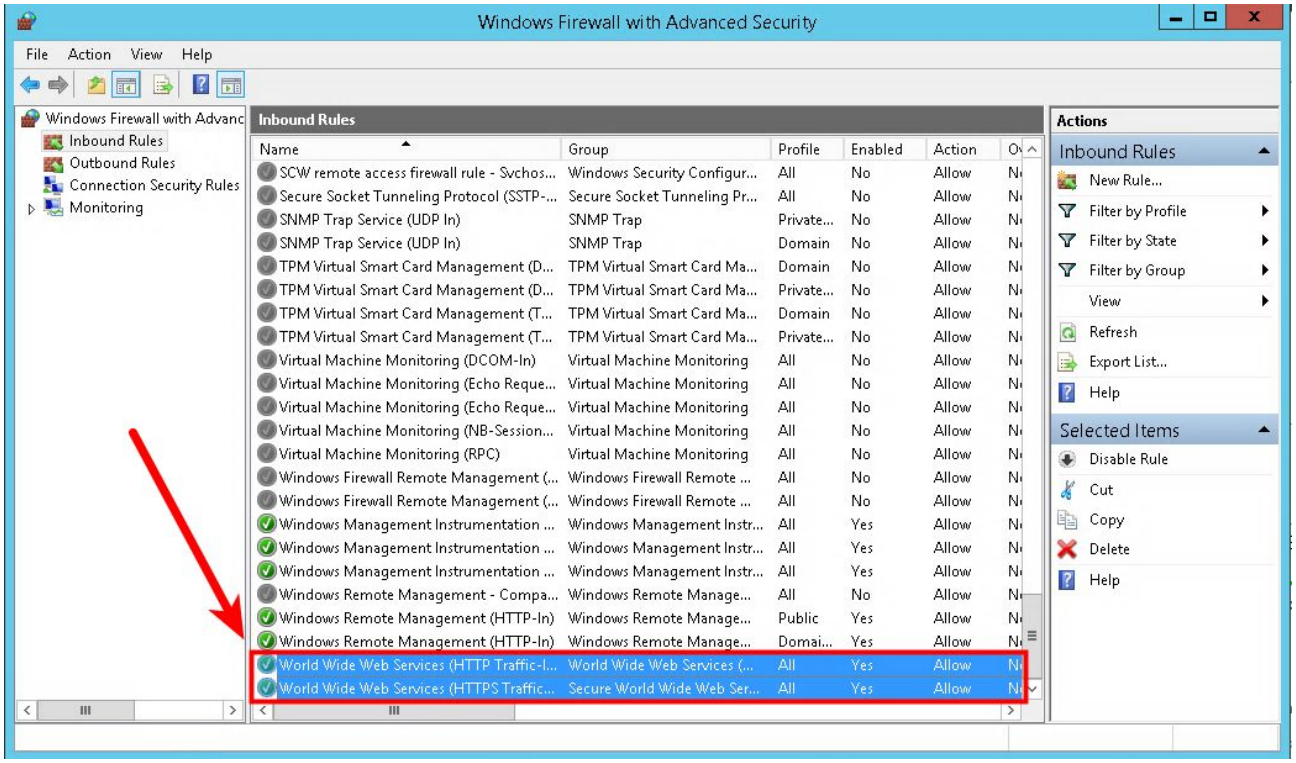
After installing IIS Server, we need to ensure that HTTP and HTTPS traffic is enabled.
To enable traffic, we need to allow HTTP/HTTPS in Windows and MS Azure firewalls.

3.2.1. Configure/Verify Windows Firewall Settings

Open **“Server Manager”** and then please click **“Tools” (1)** and **“Windows Firewall with Advanced Security” (2)**

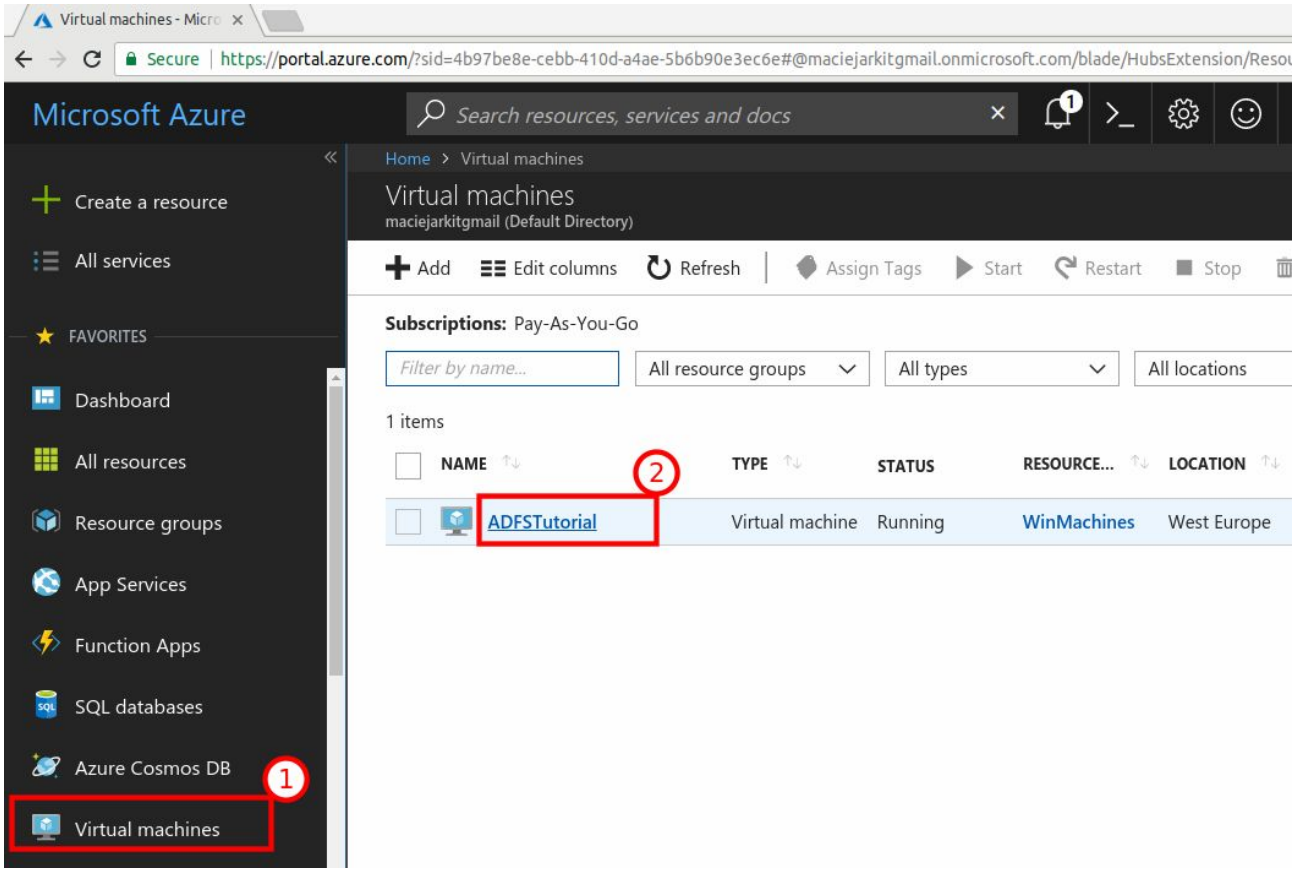


Ensure that HTTP/HTTPS traffic is enabled in Windows Firewall:

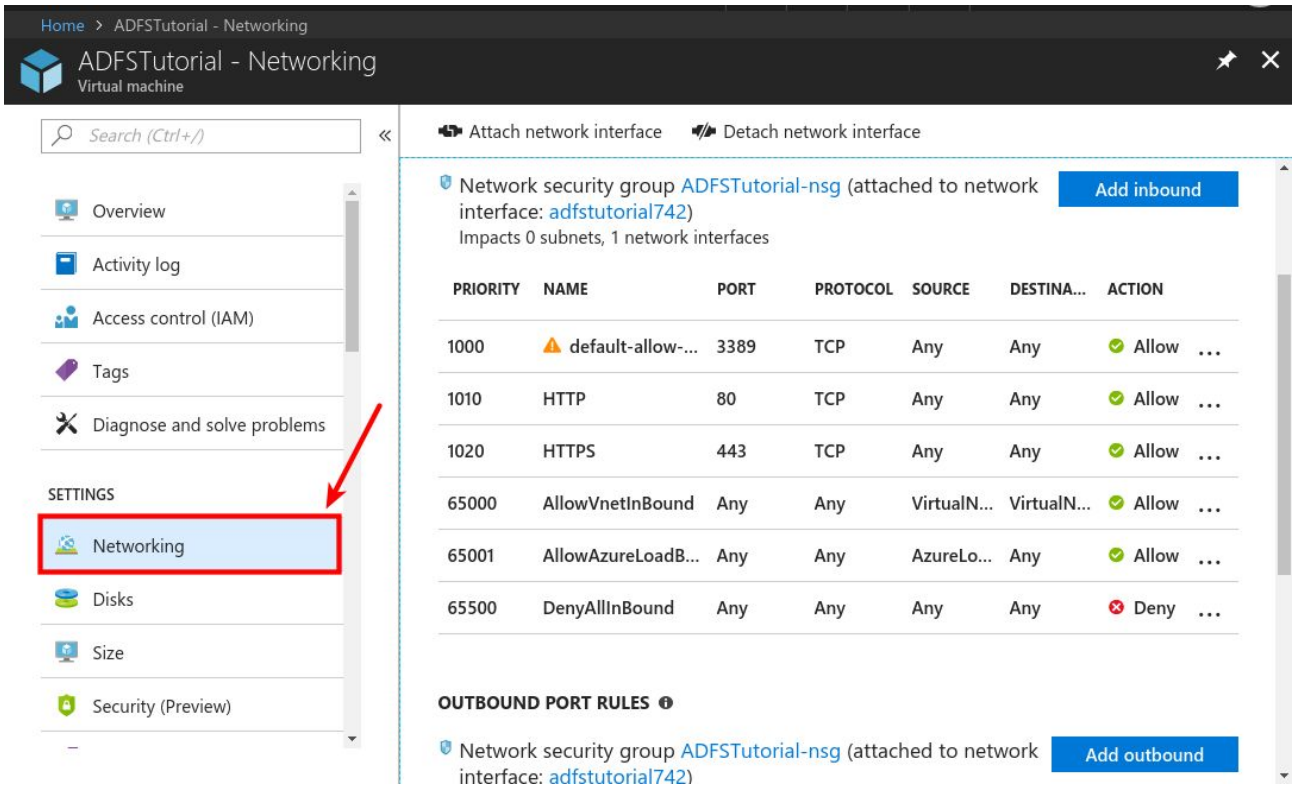


3.2.2. Configure MS Azure Firewall

Open Azure dashboard on https://portal.azure.com, select “Virtual Machines” (1), and select your virtual machine (2):

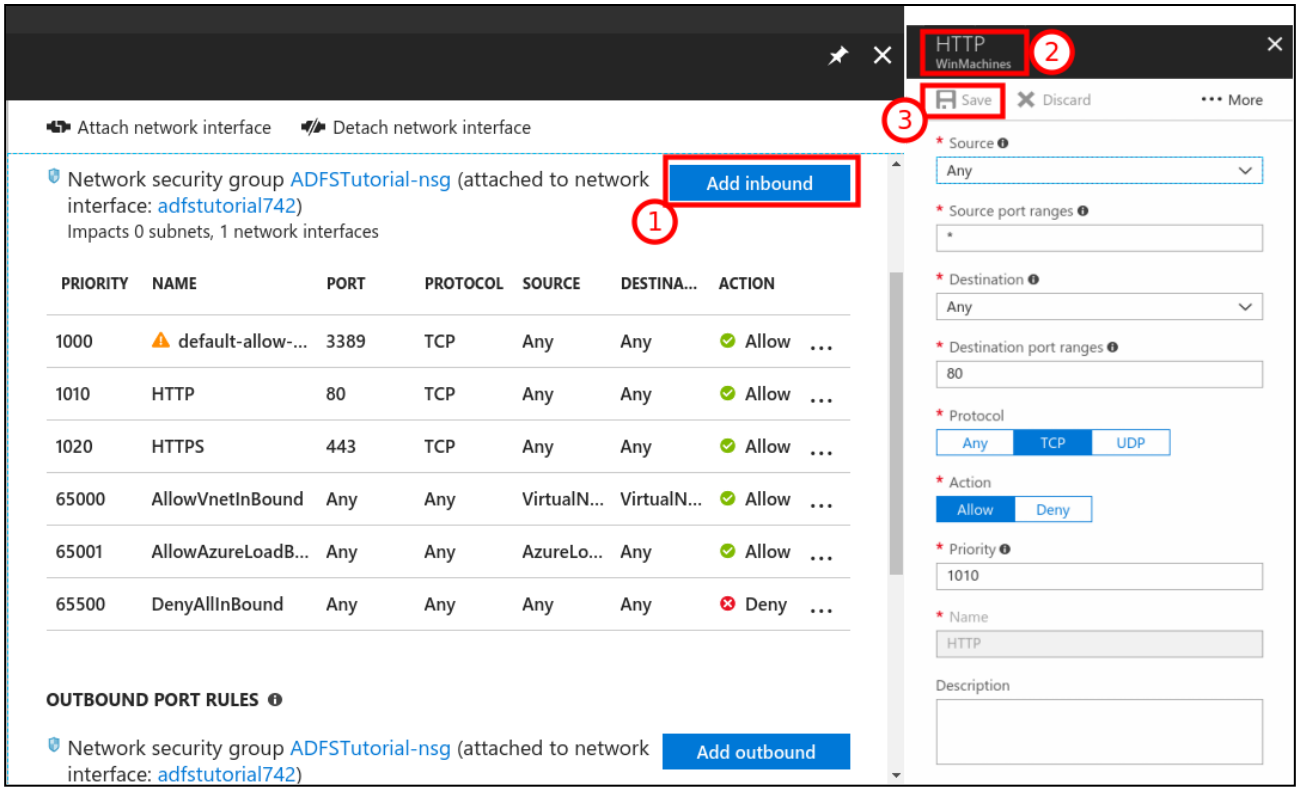


Then, please select “Networking”:

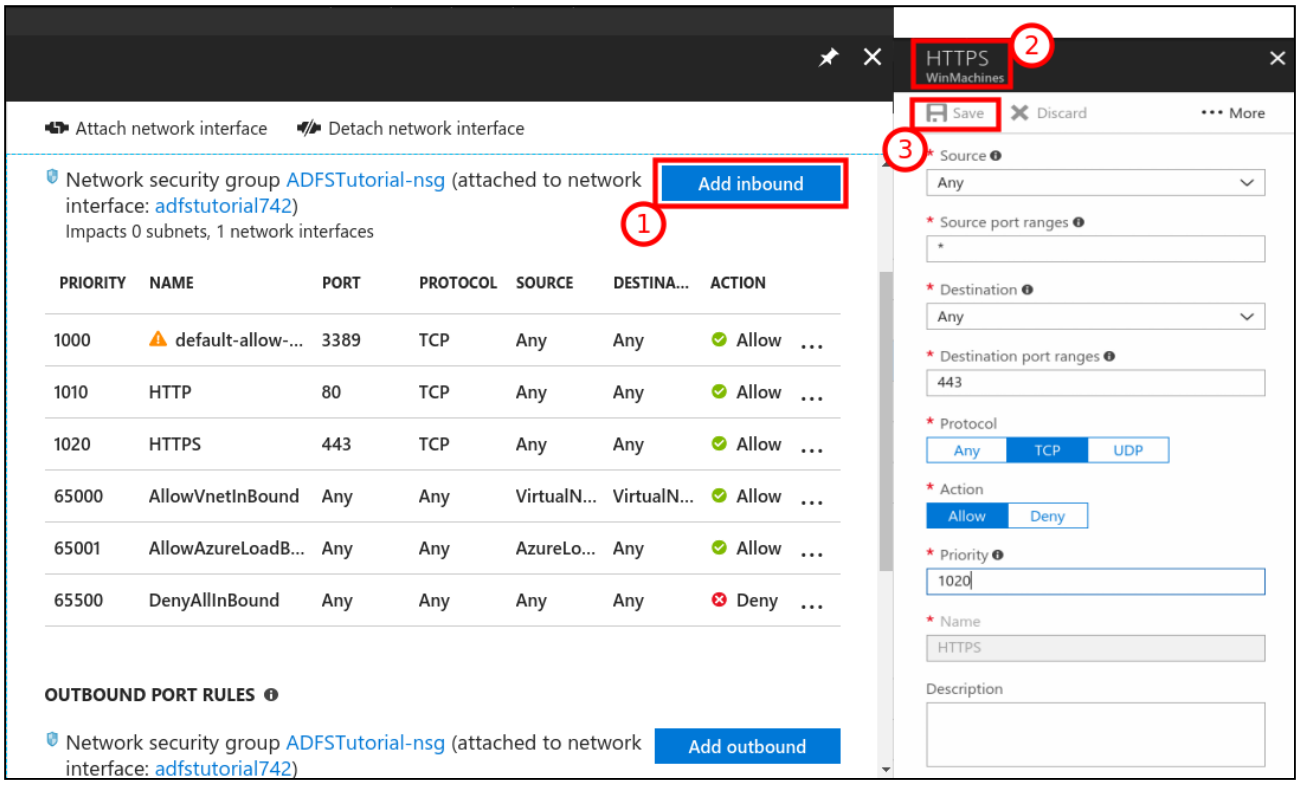


Configure Inbound HTTP/HTTPS traffic

1/ Please click **“Add inbound” (1)**, then fill out the form **(2)** to allow incoming HTTP traffic, and click **“Save” (3)**:

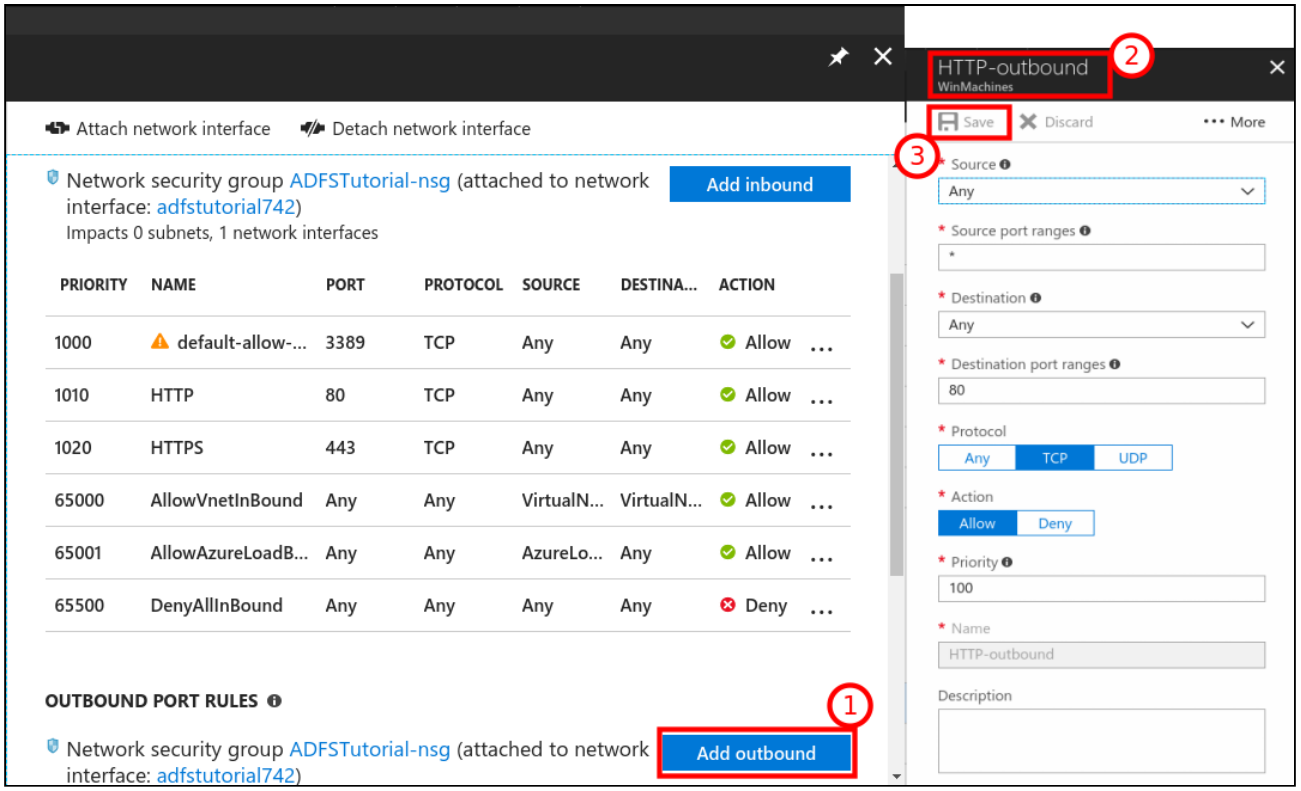


2/ Please click **“Add inbound”** (1) again, then fill out the form **(2)** to allow incoming **HTTPS** traffic, and click **“Save” (3)**:

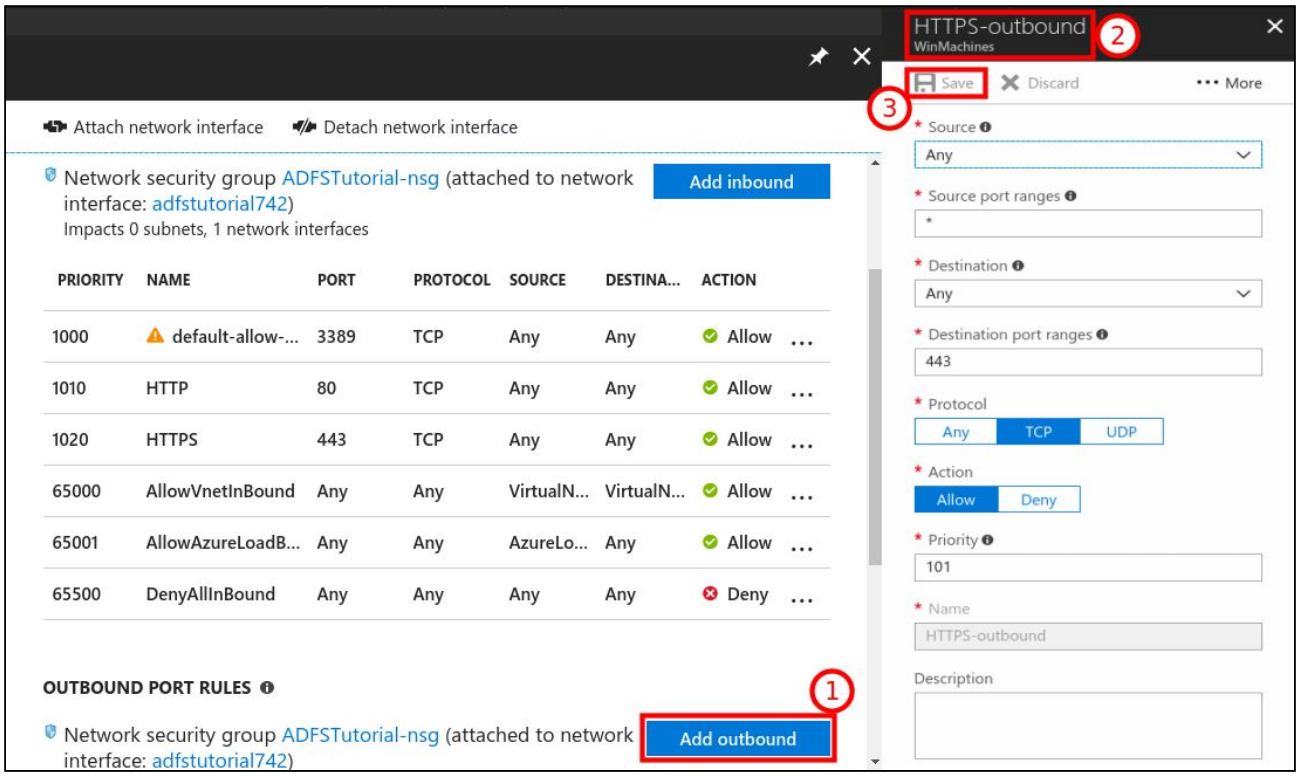


Configure Outbound HTTP/HTTPS

1/ Please click **“Add outbound”** (1), then fill out the form **(2)** to allow outgoing **HTTP** traffic, and click **“Save” (3)**:



2/ Please click **“Add outbound” (1)** again, then fill out the form **(2)** to allow outgoing **HTTPS** traffic, and click **“Save” (3)**:

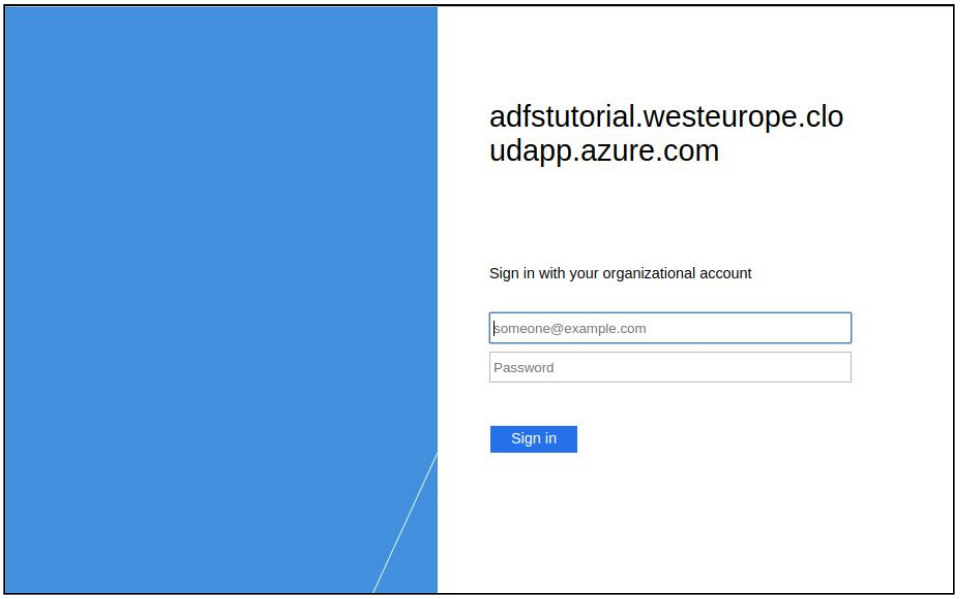


3.2.3. Verify HTTP(s) connections

1/ Please visit federation metadata url to verify that HTTPS connection works properly.
After visiting this URL, your browser should be able to download federation metadata XML:
https://**$MACHINE_NAME.$REGION_NAME**.cloudapp.azure.com/federationmetadata/2007-06/federationmetadata.xml
Ex:
https://adfstutorial.westeurope.cloudapp.azure.com/federationmetadata/2007-06/federationmetadata.xml

2/ Verify that you can access AD FS login form:
https://**$MACHINE_NAME.$REGION_NAME**.cloudapp.azure.com/adfs/ls/idpinitiatedsignon.aspx
Ex: https://adfstutorial.westeurope.cloudapp.azure.com/adfs/ls/idpinitiatedsignon.aspx

After visiting this URL you should see standard AD FS login page:



---

4. Integrate Ruby on Rails application with AD FS Single Sign On

When AD FS server is configured we can move to sample Rails application.
In this part of the tutorial we will do following things:

- Use Devise to configure basic authentication and OmniAuth to add Single Sign On authentication and integrate it with AD FS server
- Deploy application to Heroku server
- Move back to Windows VM to create “Relying Party Trust” based on metadata exposed by our Rails application
- Create sample test account in Active Directory
- Create attributes mapping for “Relying Party Trust” - i.e. tell AD FS which data about the Active Directory account should be passed to our Rails application during authentication process (ex. Email address, first name, last name, etc)
- Create sample account in Rails application which will be paired with our Active Directory account
- Test SSO login and logout

If you prefer you can follow steps below and create sample application from scratch.
You can also download working example from https://github.com/maciej-arkit/RoR_integration_with_ADFS_SSO_example, and adjust it to work with your AD FS instance (it requires to update URLs).

4.1. Generate RoR application and configure Devise and OmniAuth

1/ Generate application:

```
$ rails new sso_integration_with_adfs_and_omniauth
```

2/ Update Gemfile - add omniauth gems:

Gemfile

```
# Devise for authentication:
gem 'devise'
# OmniAuth gems:
gem 'omniauth'
gem 'omniauth-saml'
```

3/ Ensure that all connections to your Rails applications are securet - configure force SSL. Otherwise AD FS server will report an error, when it will receive a call from your application.

config/environments/production.rb

```
[…]
# Force all access to the app over SSL, use Strict-Transport-Security, and use secure cookies.
config.force_ssl = true
[...]
```

4/ Execute:

```
bundle install
rails generate devise:install
```

5/ Edit **`devise.rb`** and add following configuration:

config/initializers/devise.rb

```
config.omniauth :saml,
          issuer: "#{ENV['ISSUER'] || 'adfstutorial.herokuapp.com'}",
          idp_sso_target_url: "#{ENV['IDP_SSO_TARGET_URL'] ||
'https://adfstutorial.westeurope.cloudapp.azure.com/adfs/ls/idpinitiatedsignon.aspx'}",
          idp_cert_fingerprint: "#{ENV['IDP_CERT_FINGERPRINT'] || 'TODO-PROVIDE-CERTIFICATE-FINGERPRINT'}",
          attribute_statements: {email: ['User.Email'], name: ['User.LastName'], first_name: ['User.FirstName'], last_name:
['User.LastName']},
          single_logout_service_url: "https://#{ENV['ISSUER']}/users/auth/saml/slo",
          idp_slo_target_url: "#{ENV['IDP_SLO_TARGET_URL'] ||
'https://adfstutorial.westeurope.cloudapp.azure.com/adfs/ls/?wa=wsignout1.0&wreply=https://adfstutorial.herokuapp.com'}"
```

"issuer" - Usually it should be name or url of your application. This name will be used to lookup corresponding configuration in ADFS service by “identifier”.

"idp_sso_target_url" - URL of SSO service. User will be redirected to this service upon SSO login attempt (unless already logged in).

"idp_cert_fingerprint" - fingerprint of a certificate used to create ADFS SSO configuration.

"attribute_statements" - Used to map Attribute Names in a SAMLResponse to entries in the OmniAuth info hash. For example, if your SAMLResponse contains an Attribute called 'EmailAddress', specify {:email => ['EmailAddress']} to map the Attribute to the corresponding key in the info hash. More information: https://github.com/omniauth/omniauth-saml

"single_logout_service_url" - required for Identity Provider initiated logout. Logout request from IdP will be sent to this URL

"idp_slo_target_url" - requried for Service Provider (i.e. your web app) initiated logout URL. Once user initiates logout action in your application, corresponding request will be sent as well to IdP, to this URL.

6/ Update `routes.rb` configuration

config/routes.rb

```
devise_for :users, controllers: {
  omniauth_callbacks: 'omniauth_callbacks',
  sessions: 'users/sessions'
}
```

7/ Update User.rb - add omniauthable annotation and attributes

app/models/user.rb

```
class User < ApplicationRecord
 # Include default devise modules. Others available are:
 # :confirmable, :lockable, :timeoutable and :omniauthable
 devise :database_authenticatable, :registerable,
       :recoverable, :rememberable, :trackable, :validatable,
       :omniauthable, omniauth_providers: [:saml]
end
```

8/ Create Rails migration to add **provider** and **uid** field to **User** model. These fields will be needed to keep meta information for SSO authentication and pair User with SSO account (ex. In Active Directory):

```
$ rails g migration AddOmniauthToUsers provider:string uid:string
$ rake db:migrate
```

9/ Add OmniauthCallbacksController:

app/controllers/users/omniauth_callbacks_controller.rb

```
# frozen_string_literal: true
class Users::OmniauthCallbacksController < Devise::OmniauthCallbacksController

 def saml
  # You need to implement the method below in your model (e.g. app/models/user.rb)
  @user = User.from_omniauth(request.env["omniauth.auth"])

  if @user
    sign_in_and_redirect @user, event: :authentication #this will throw if @user is not activated
    set_flash_message(:notice, :success, kind: 'ADFS SSO') if is_navigational_format?
  else
    failure
  end
 end

 def failure
  redirect_to root_path
 end
end
```

10/ Add from_omniauth method to user.rb

app/models/user.rb

```ruby
class User < ApplicationRecord
 # Include default devise modules. Others available are:
 # :confirmable, :lockable, :timeoutable and :omniauthable
 devise :database_authenticatable, :registerable,
        :recoverable, :rememberable, :trackable, :validatable,
        :omniauthable, omniauth_providers: [:saml]

   # Lookup for user matching authentication data from ADFS
   # More info: https://github.com/plataformatec/devise/wiki/OmniAuth:-Overview
   def self.from_omniauth(auth)
    where(provider: auth.provider, uid: auth.uid).
       first_or_create(email: auth.uid, provider: auth.provider, uid: auth.uid, password: generate_password)
   end

   private

   def self.generate_password
    Devise.friendly_token.first(10)
   end
end
```

11/ Implement SessionController to support Single Logout:

NOTE:
To avoid exception:
NameError (uninitialized constant Users::SessionsController::SAML_SETTINGS),
ensure that **after_sign_out_path_for()** uses following IF statement:

```ruby
if session['saml_uid'] && Devise.omniauth_configs[:saml].options[:idp_slo_target_url]
```

instead of:

```ruby
if session['saml_uid'] && SAML_SETTINGS.idp_slo_target_url
```

app/controllers/users/sessions_controller.rb

```ruby
class Users::SessionsController < Devise::SessionsController
 # before_action :configure_sign_in_params, only: [:create]

 def destroy
  # Preserve the saml_uid in the session
  saml_uid = session["saml_uid"]
  super do
    session["saml_uid"] = saml_uid
  end
 end

 def after_sign_out_path_for(_)
  if session['saml_uid'] && Devise.omniauth_configs[:saml].options[:idp_slo_target_url]
    user_saml_omniauth_authorize_path + "/spslo"
  else
    super
  end
 end
end
```

11/ Configure logger - add following line to **devise.rb**:

config/initializers/devise.rb

```
# Rails application example
OmniAuth.config.logger = Rails.logger
```

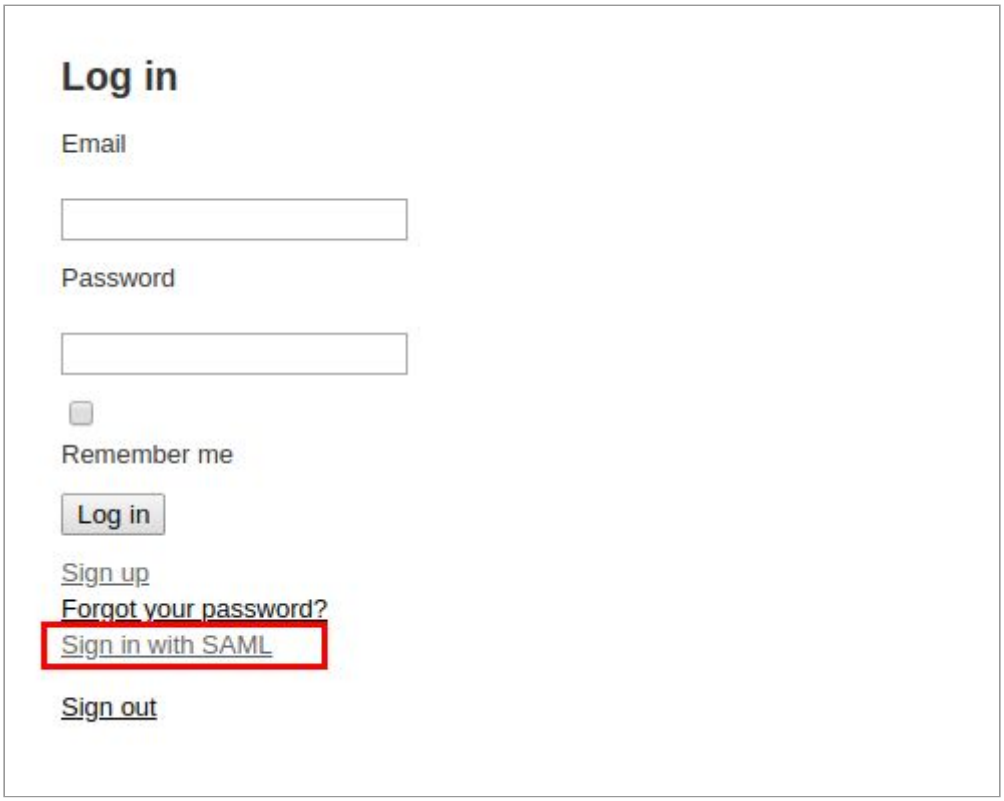
4.2. Test application locally

Start application server

```
rails s
```

and open https://localhost:3000

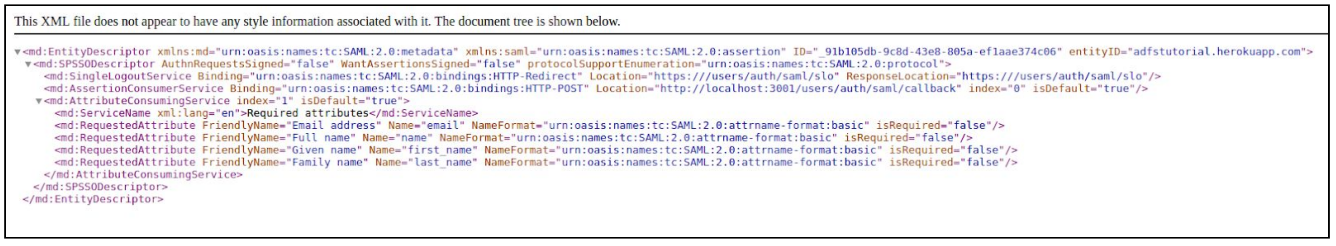
You should see updated login page with “Sign in with SAML” option:



Test SAML metadata:

http://localhost:3000/users/auth/saml/metadata

You should see metadata XML, similar as below:



4.3. Deploy application to Heroku

```
git push heroku master
```

```
heroku run rails db:migrate -a $YOUR_APP_ID
```

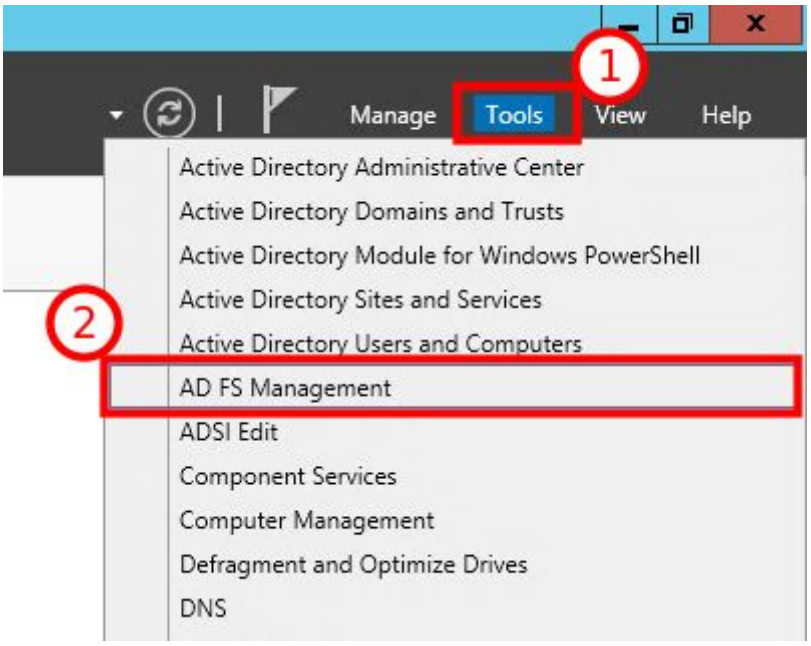
NOTE: Please ensure to configure following environmental variables which are used by *devise.rb*:

- ISSUER
- IDP_SSO_TARGET_URL
- IDP_CERT_FINGERPRINT
- IDP_SLO_TARGET_URL

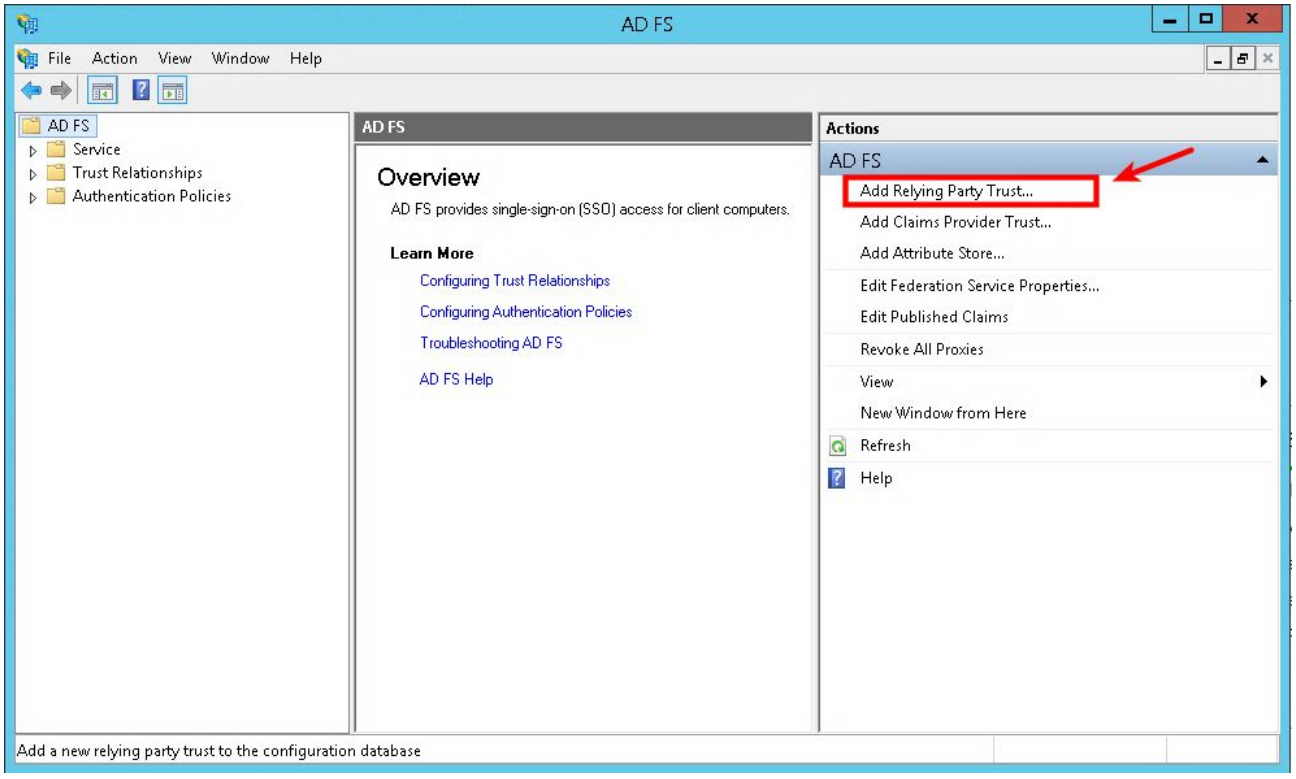
For details and explanation of each variable, please refer to 4.1. Generate RoR application and configure Devise and OmniAuth

4.4. Create “Relying Party Trust” in AD FS

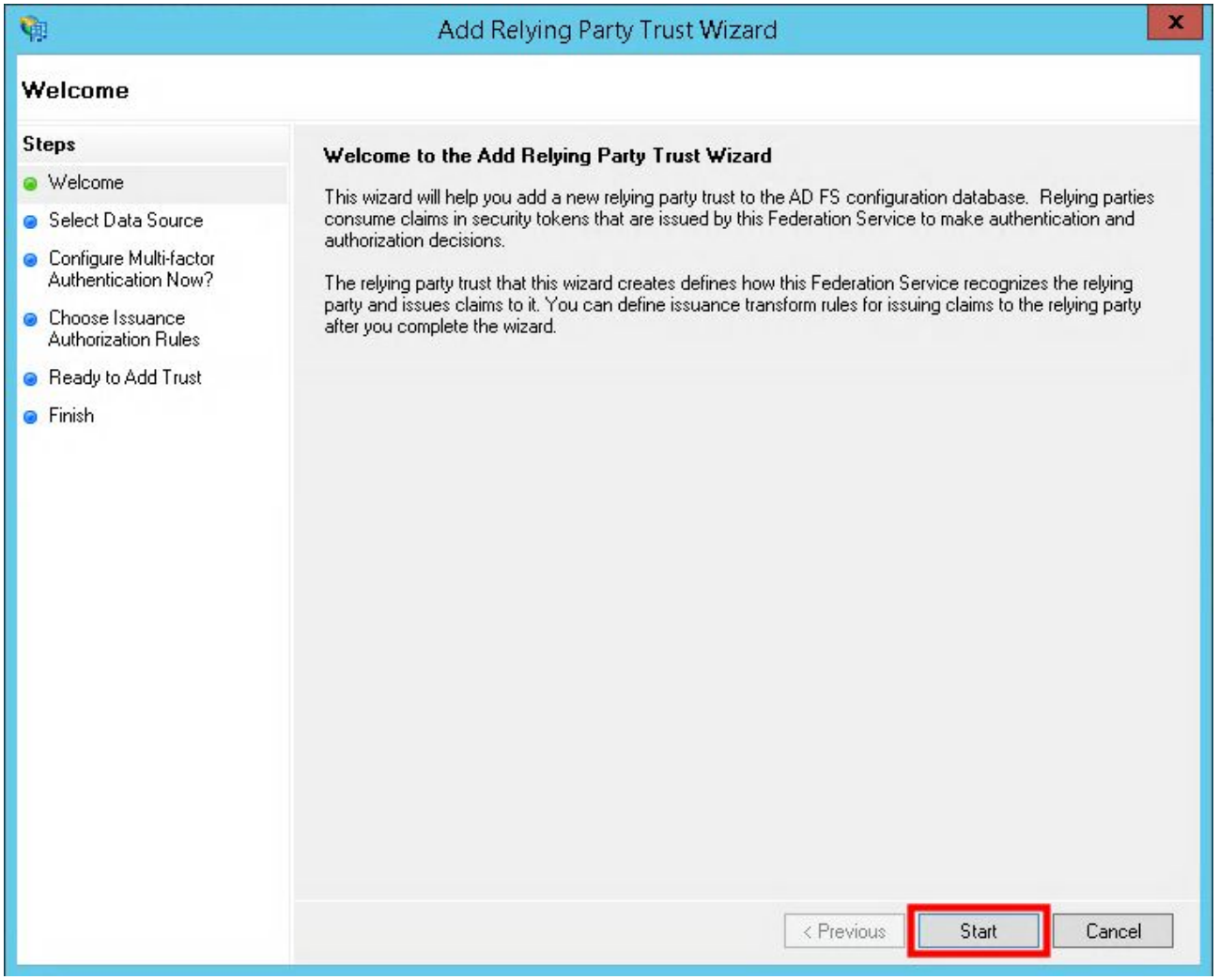
1/ From “Server Manager” please click “Tools” -> “AD FS Management”



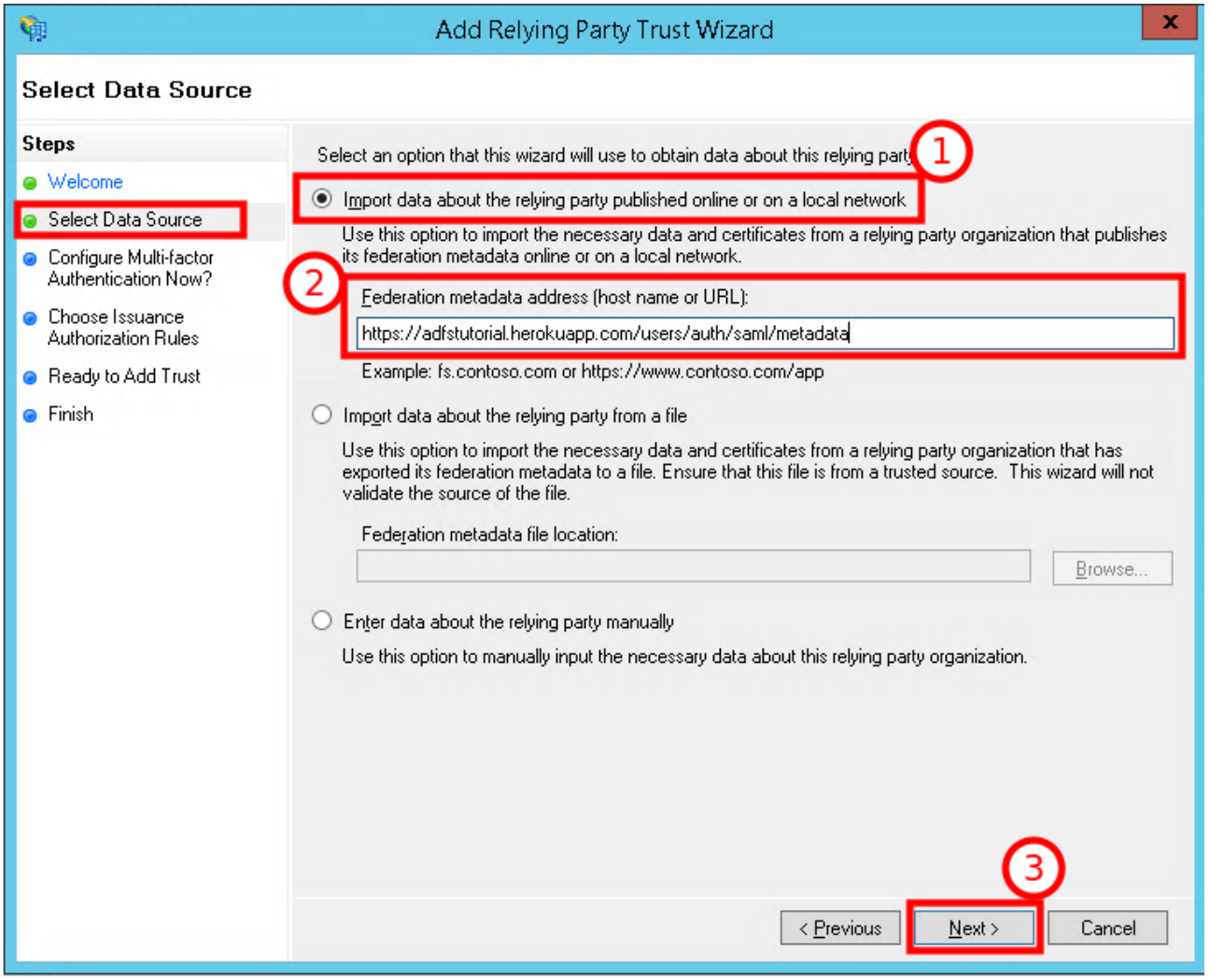
2/ Click “Add Relying Party Trust …” to open “Add Relying Party Trust Wizard”



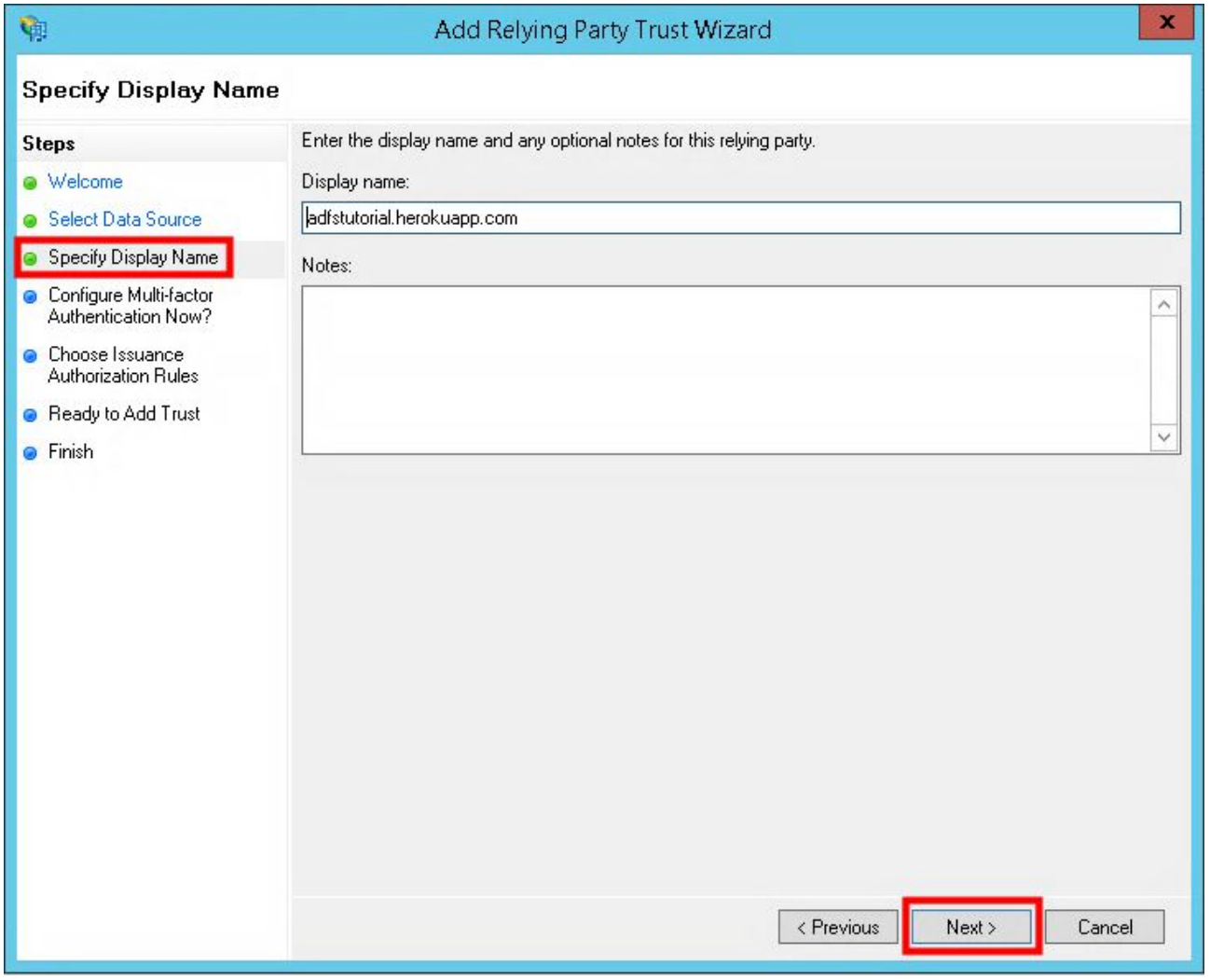
3/ On “Welcome” page of “Add Relying Party Trust Wizard” please click “Start”



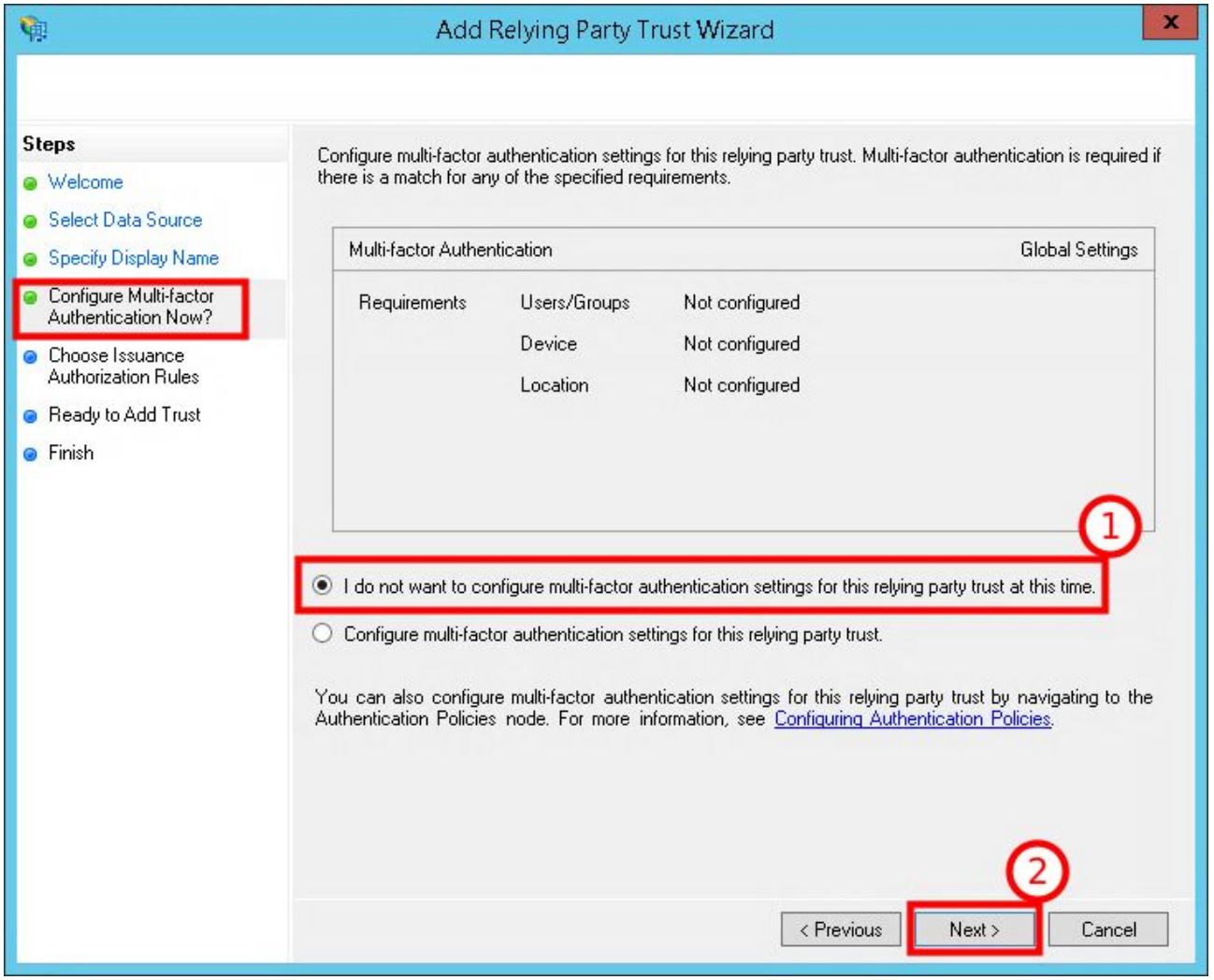
4/ On “Select Data Source” page please select “Import data about the relying party published online or on a local network” (1). Then please provide URL to your Ruby on Rails application metadata in “Federation metadata address (host name or URL)” (2) and click “Next” (3):



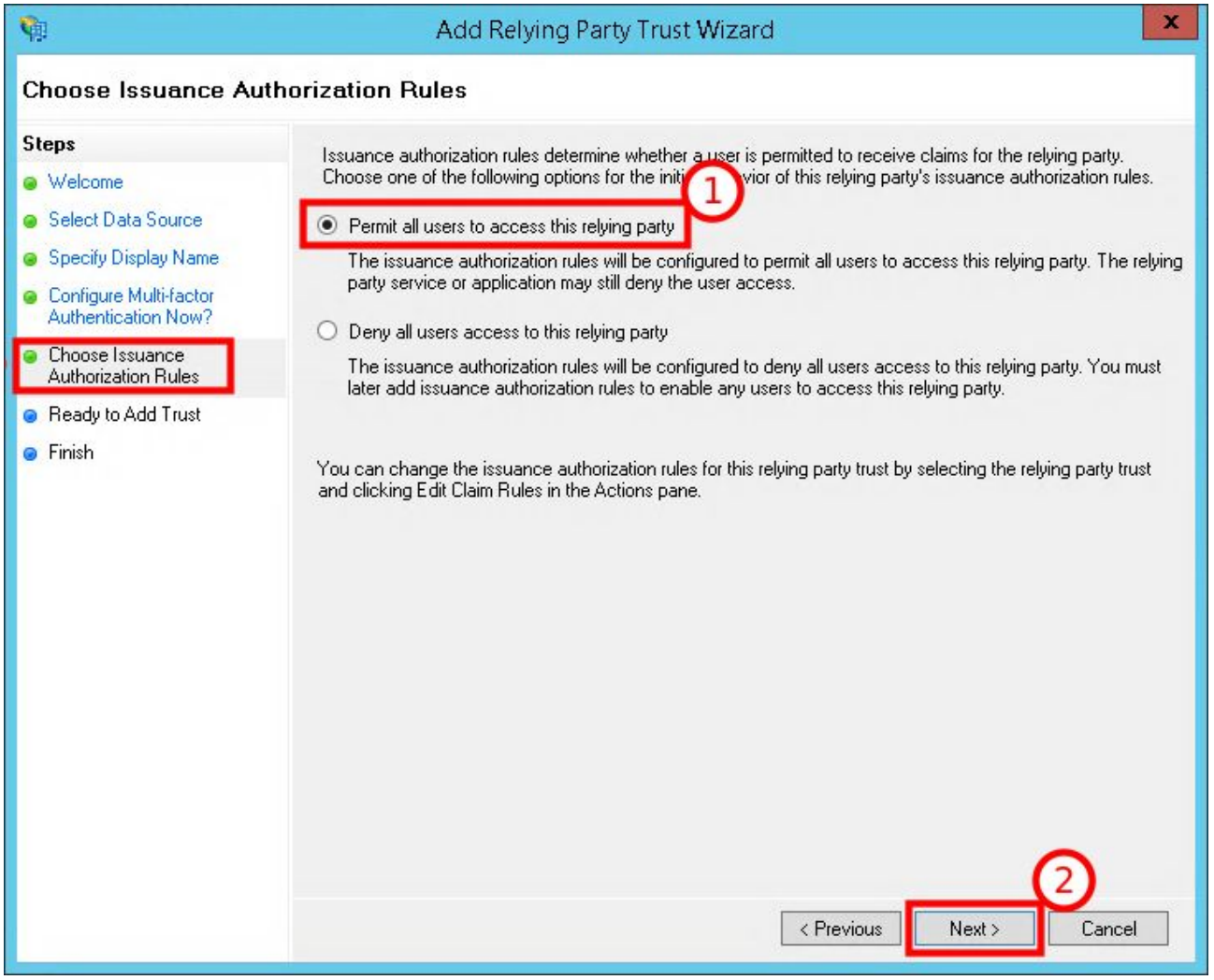
5/ On “Specify Display Name” page, please leave default values (unless you need to change it for some reason), and click “Next”:



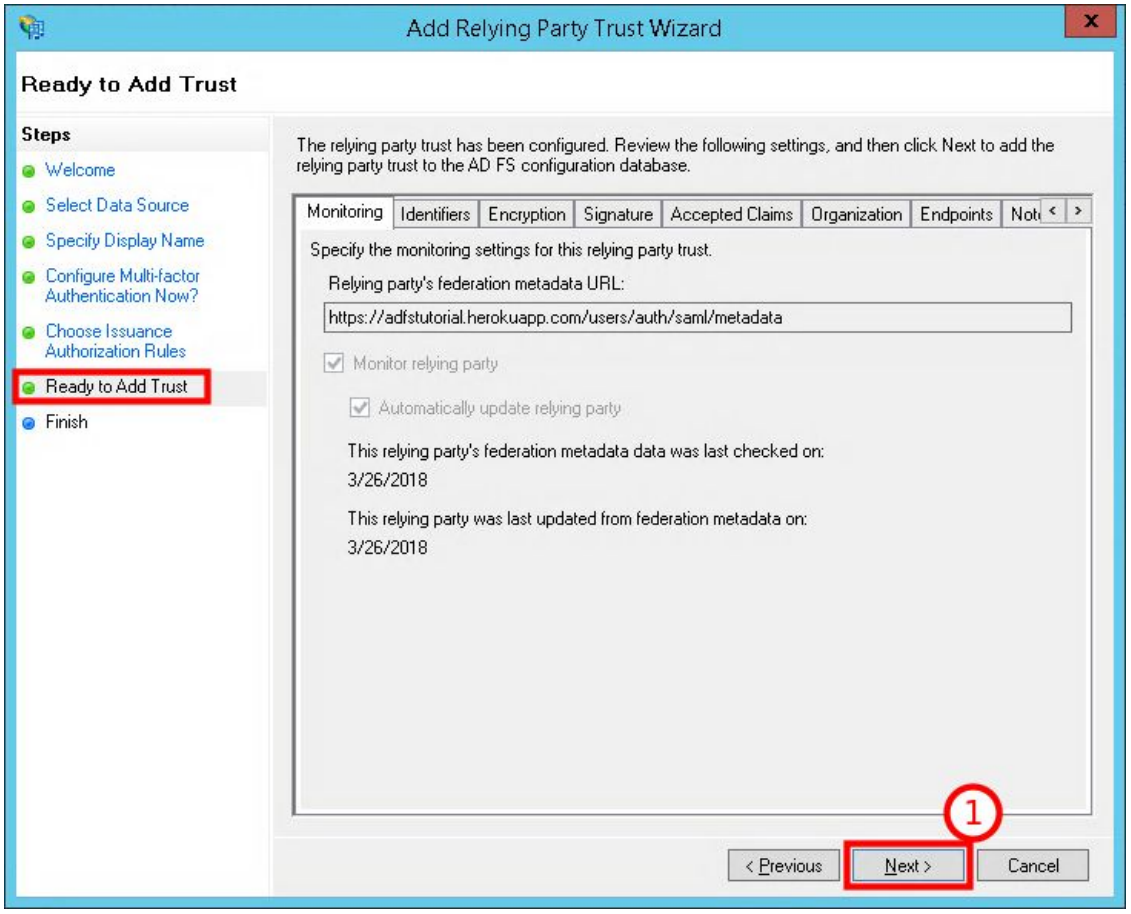
6/ On “Configure Multi-factor Authentication Now?” page, please select **“I do not want to configure multi-factor authentication settings for this relying party trust at this time.” (1)** and click **“Next” (2)**:

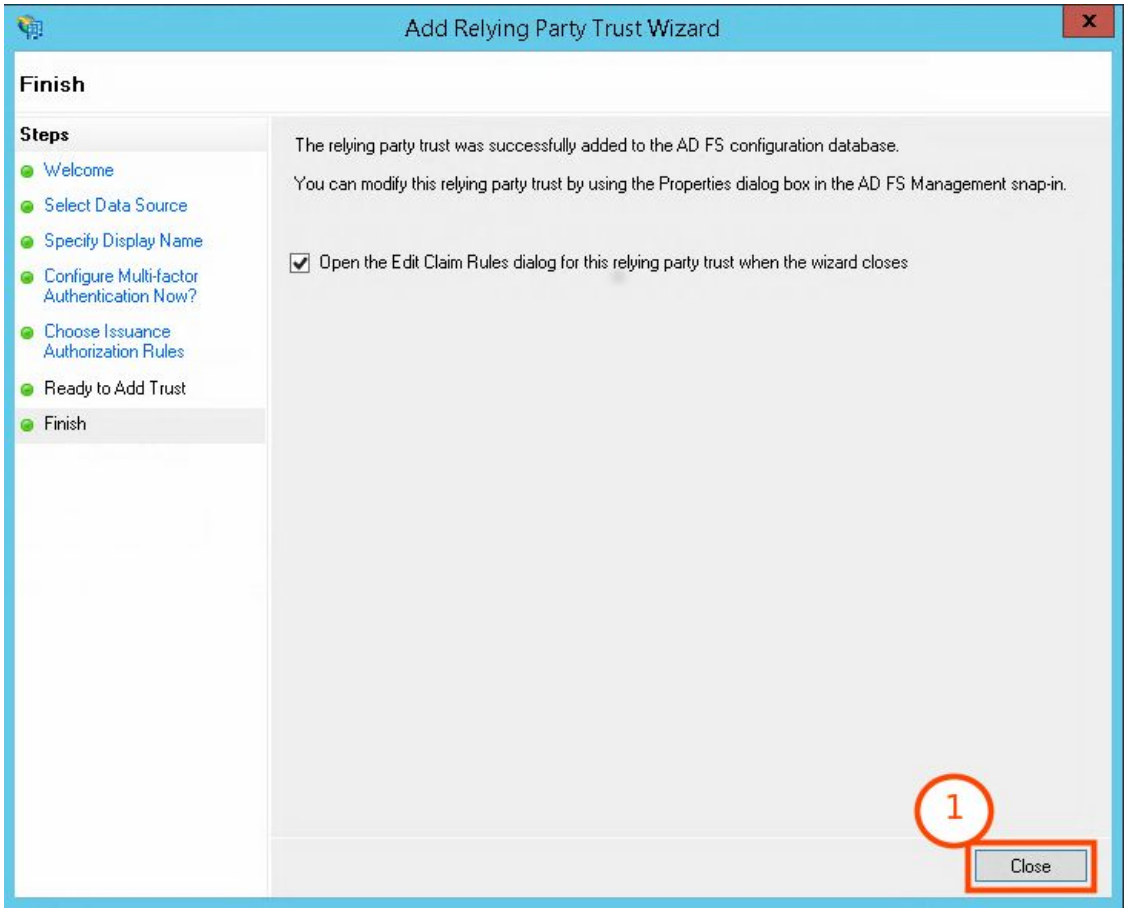


7/ On **“Choose Issuance Authorization Rules”** please select **“Permit all users to access this relying party” (1)** and please click **“Next” (2)**:



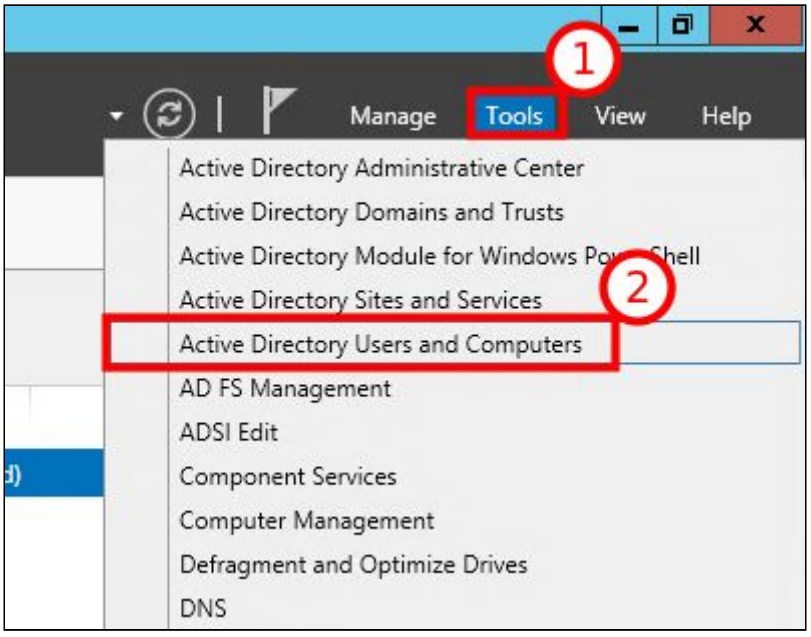
8/ On “Ready to Add Trust” please click “Next”, and then on “Finish” page please click “Close”


Add Relying Party Trust Wizard
Ready to Add Trust
Steps
Welcome
Select Data Source
Specify Display Name
Configure Multi-factor Authentication Now?
Choose Issuance Authorization Rules
Ready to Add Trust
Finish
The relying party trust has been configured. Review the following settings, and then click Next to add the relying party trust to the AD FS configuration database.
Monitoring
Identifiers
Encryption
Signature
Accepted Claims
Organization
Endpoints
Specify the monitoring settings for this relying party trust.
Relying party's federation metadata URL:
https://adfstutorial.herokuapp.com/users/auth/saml/metadata
Monitor relying party
Automatically update relying party
This relying party's federation metadata data was last checked on:
3/26/2018
This relying party was last updated from federation metadata on:
3/26/2018
1
< Previous
Next >
Cancel

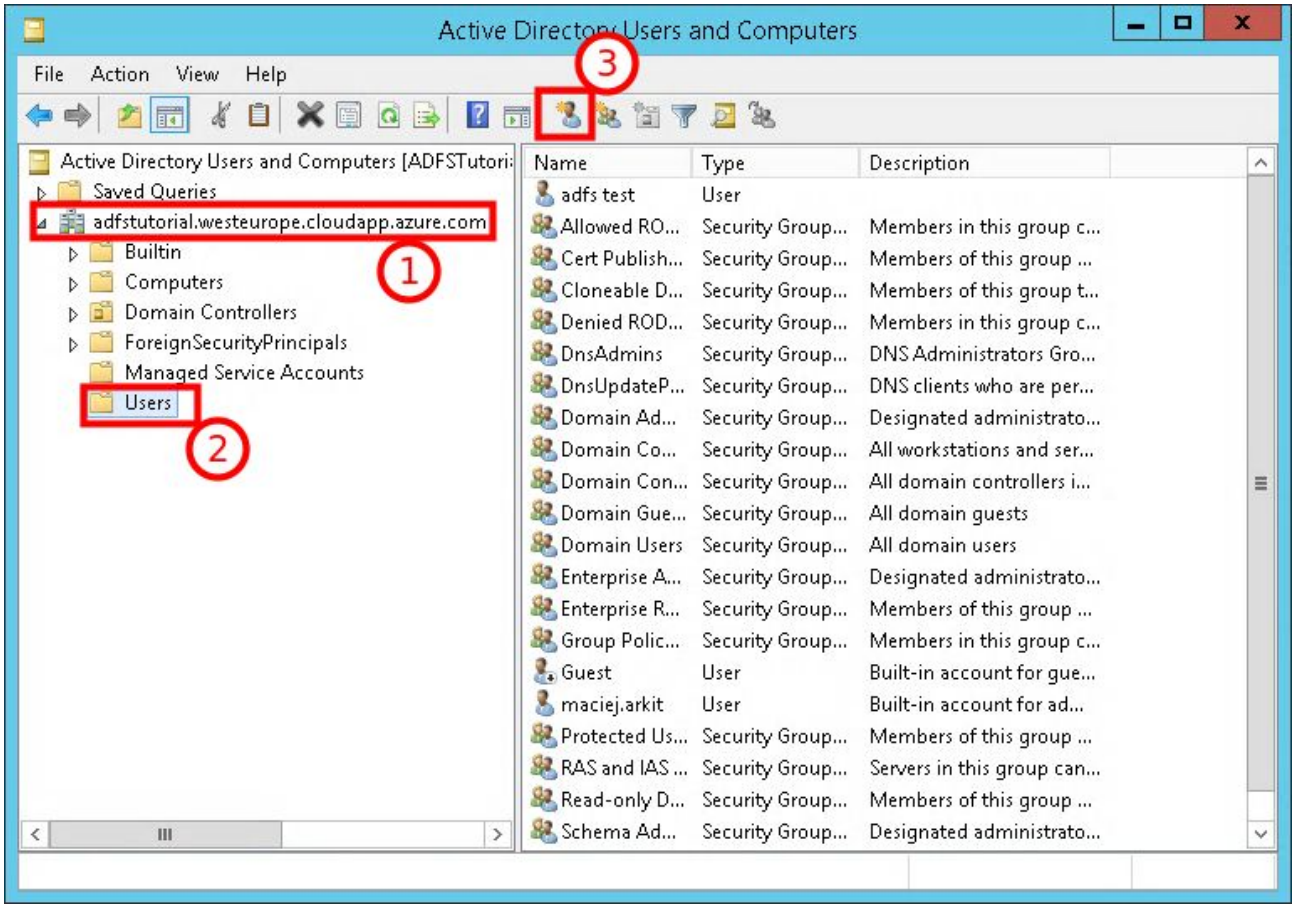

Add Relying Party Trust Wizard
Finish
Steps
Welcome
Select Data Source
Specify Display Name
Configure Multi-factor Authentication Now?
Choose Issuance Authorization Rules
Ready to Add Trust
Finish
The relying party trust was successfully added to the AD FS configuration database.
You can modify this relying party trust by using the Properties dialog box in the AD FS Management snap-in.
Open the Edit Claim Rules dialog for this relying party trust when the wizard closes
1
Close

4.5. Create user account in Active Directory

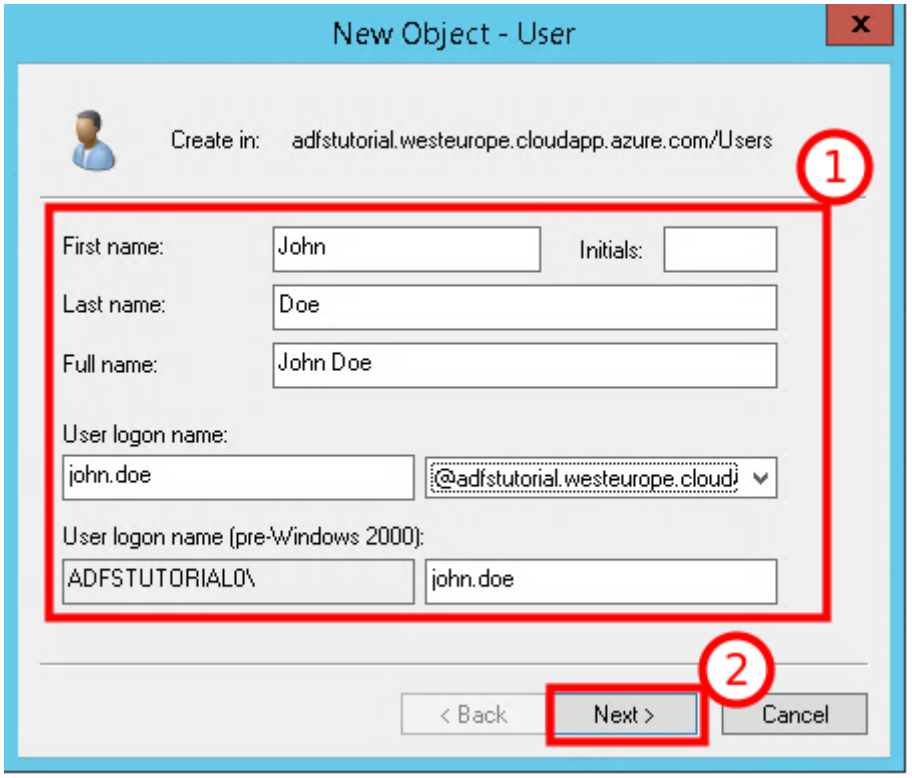
1/ Open **“Server Manager”** and then please click **“Tools” (1)** and **“Active Directory Users and Computers” (2)**



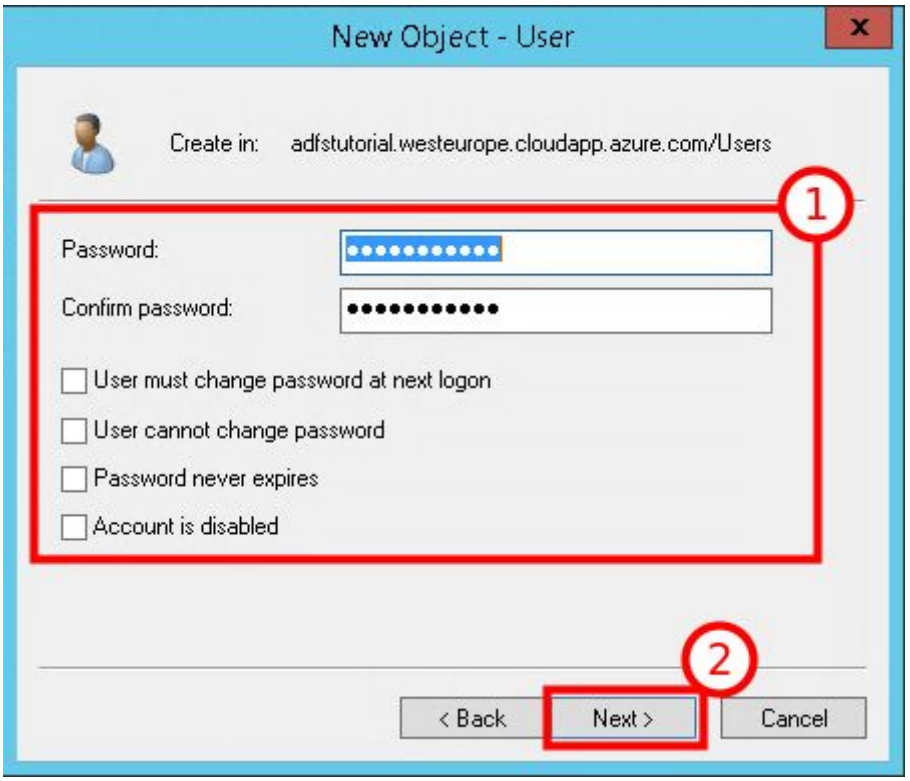
2/ On **“Active Directory Users and Computers”** please navigate to node representing your domain **(1)** and select **“Users”** node **(2)**. Then please click **“Create a new user in the current container” (3):**



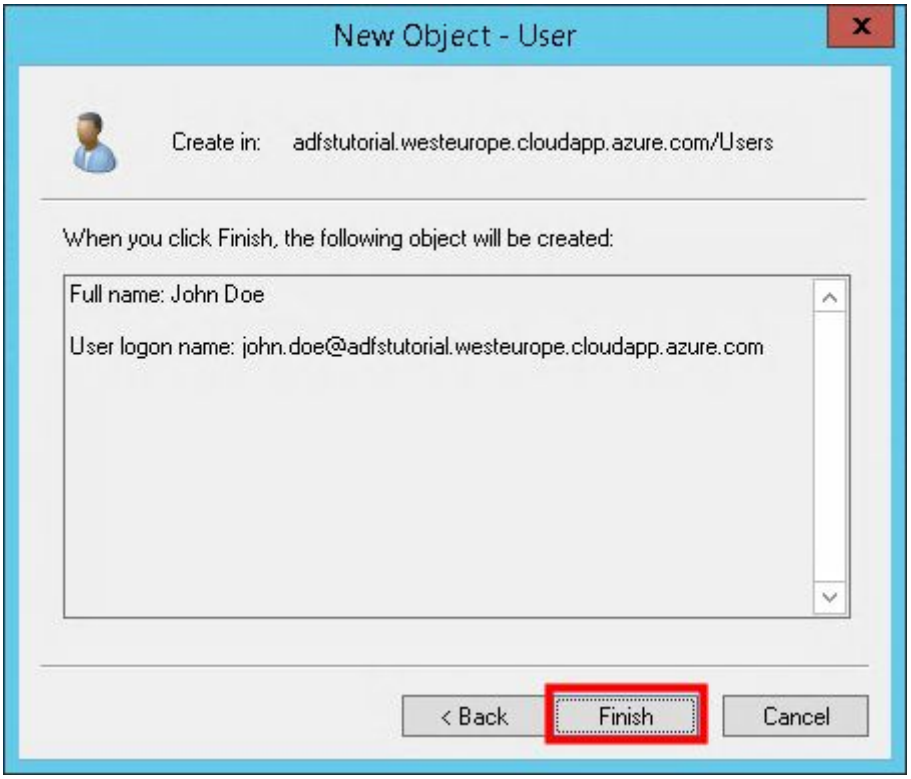
3/ In **“New Object - User”** dialog, please provide personal data for new user **(1)** and then click **“Next” (2)**:



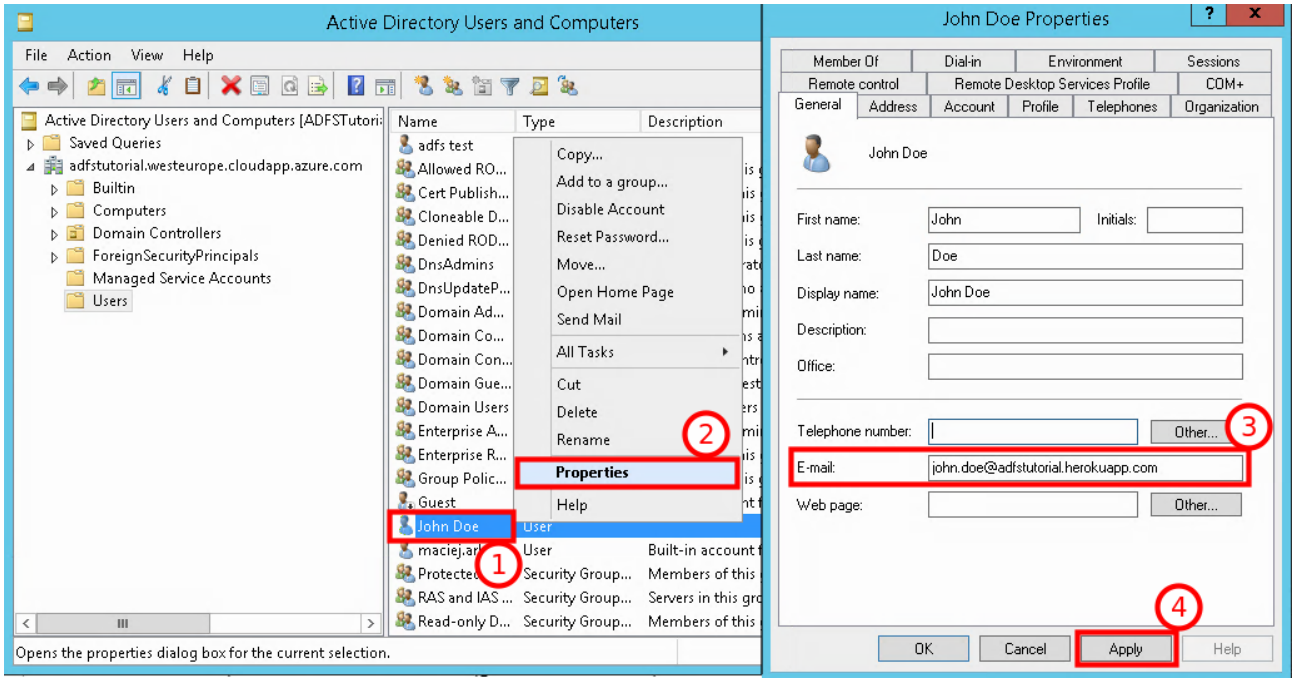
4/ On next screen, please provide password and select account options **(1)** and then click **“Next” (2)**



5/ On the summary screen, please click “Finish”:



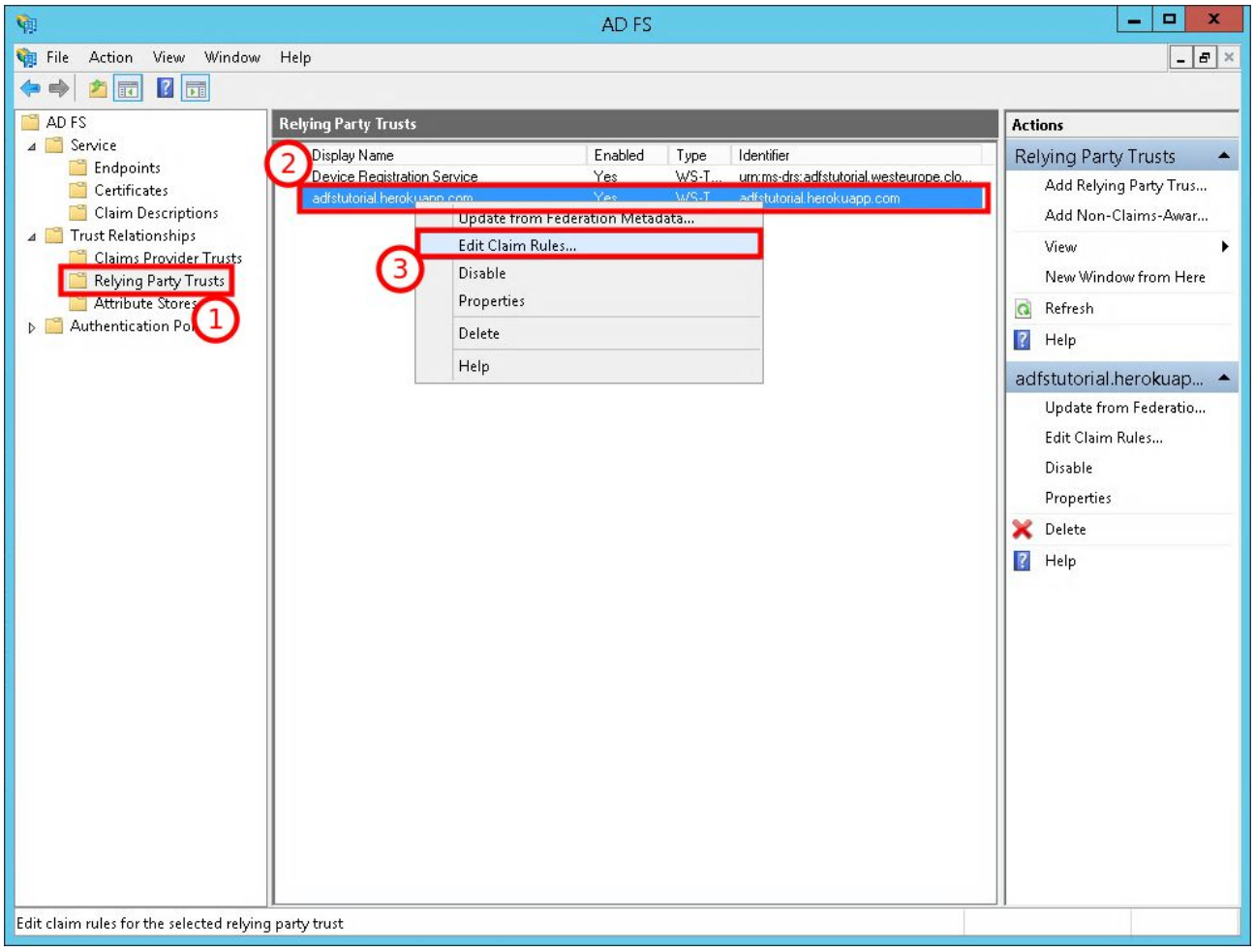
6/ When account will be created, please open **"Properties"** dialog, fill in **e-mail** address **(3)** and click **"Apply" (4)**.



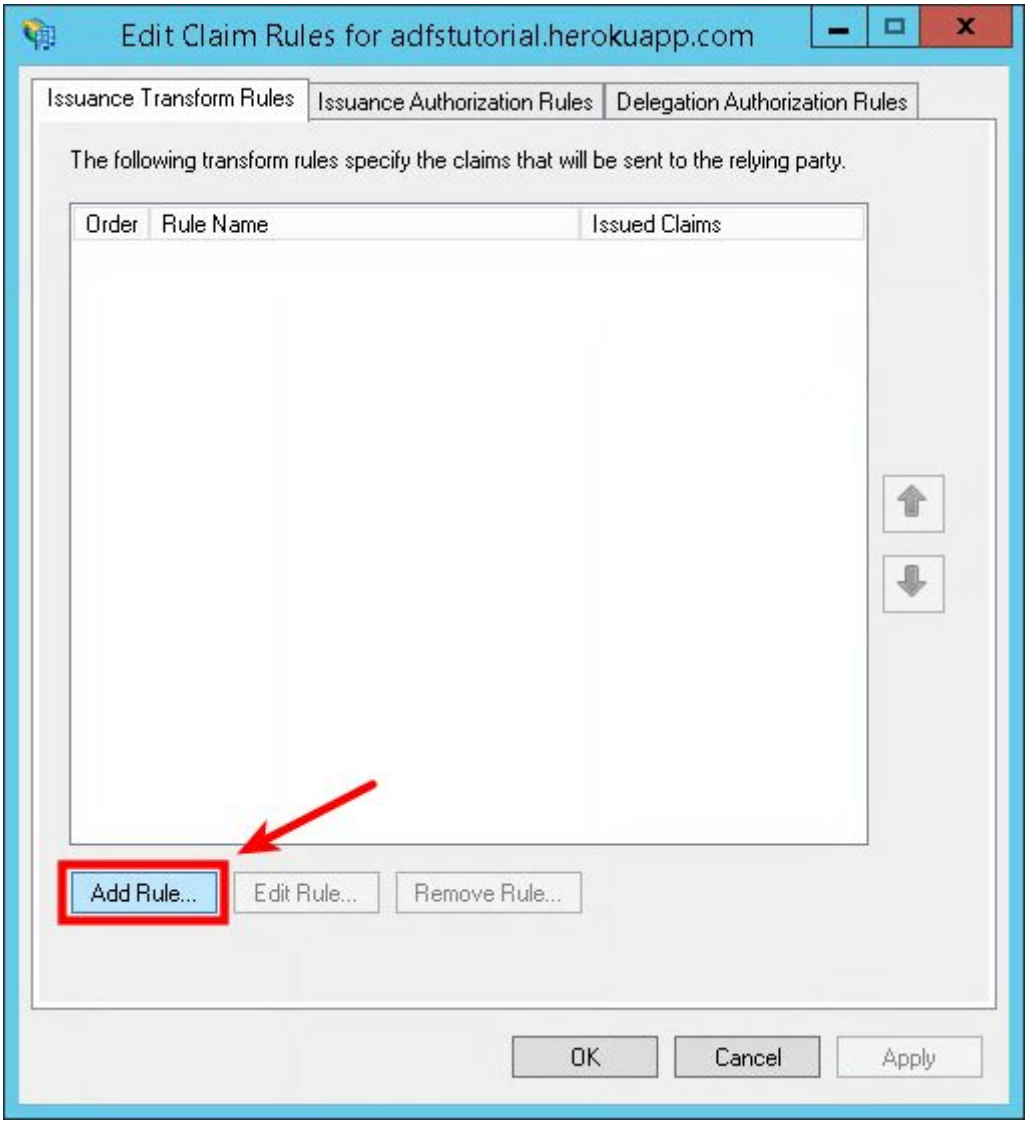
4.6. Create Claim Rules

Claim Rules are required to tell AD FS which data about the Active Directory account should be passed to our Rails application during authentication process (ex. e-mail address, first name, last name, etc)

1) In **“AD FS Management”**, please navigate to **“Relying Party Trusts” (1)**. Select your **“Relying Party Trust” (2)** and from context menu choose **“Edit Claim Rules …” (3)**

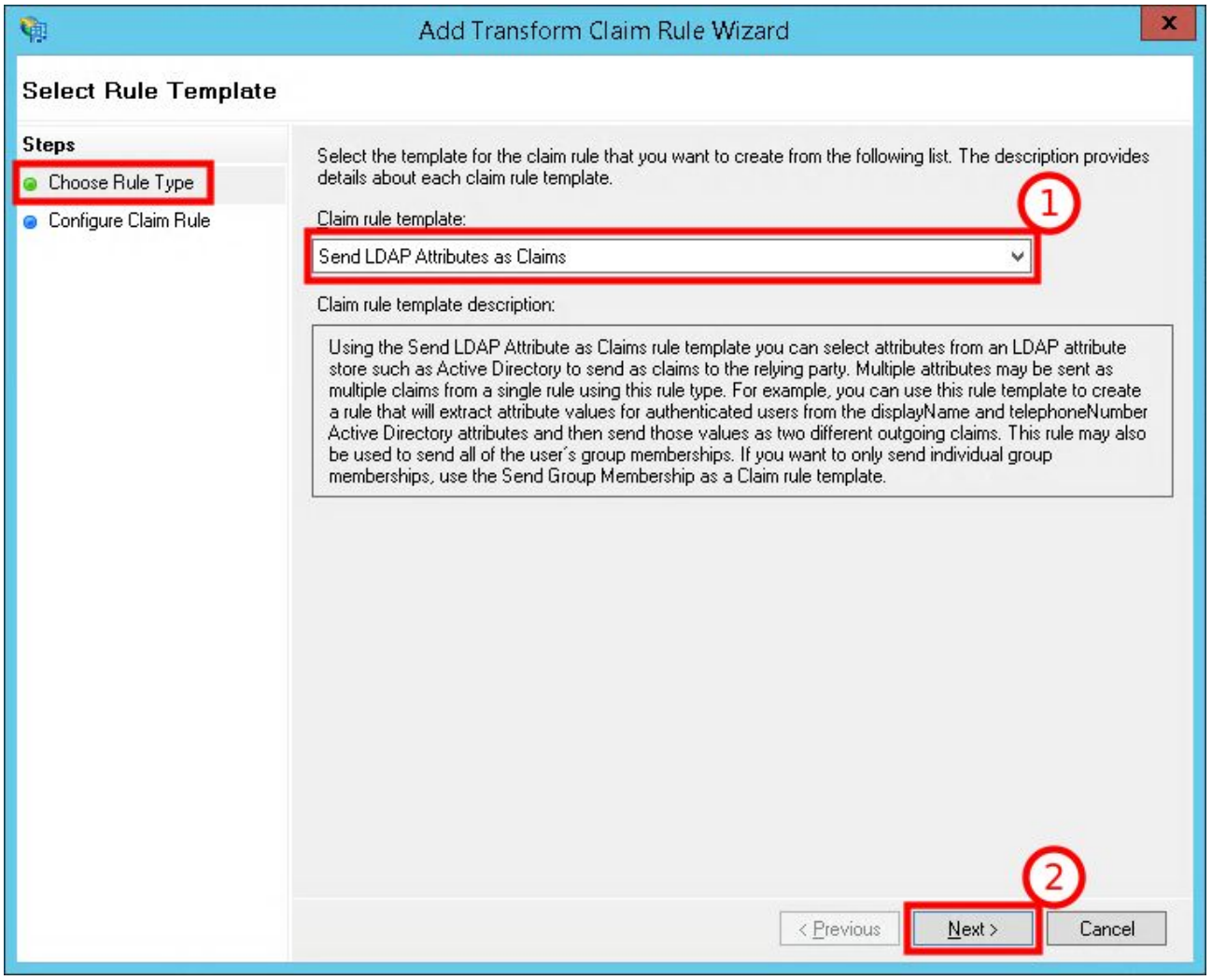


2/ On “Edit Claim Rules” dialog, please click “Add Rule” to add new rule.

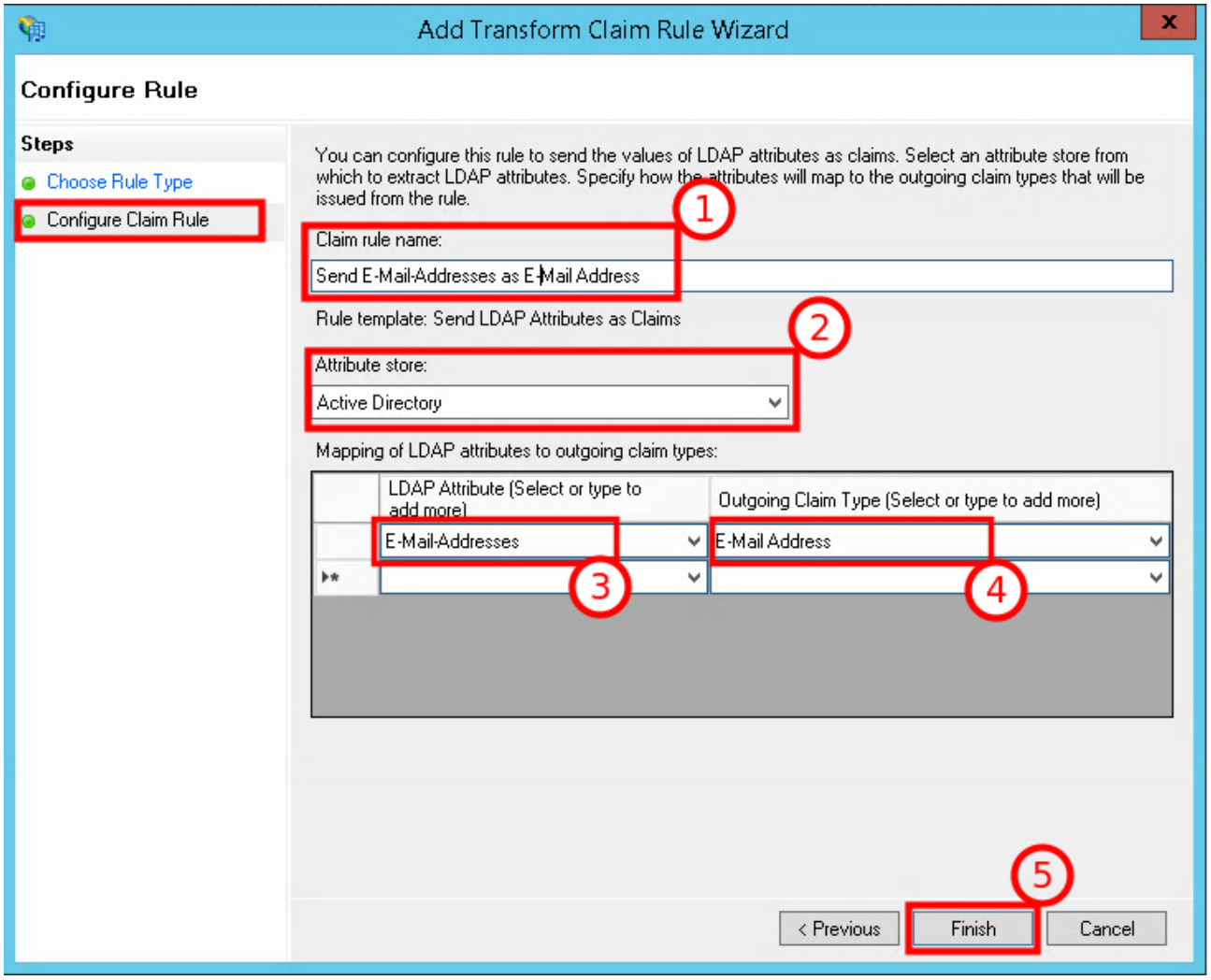


3/ Add **“Send LDAP Attribute as Claims”** rule for E-Mail-Addresses as follows:

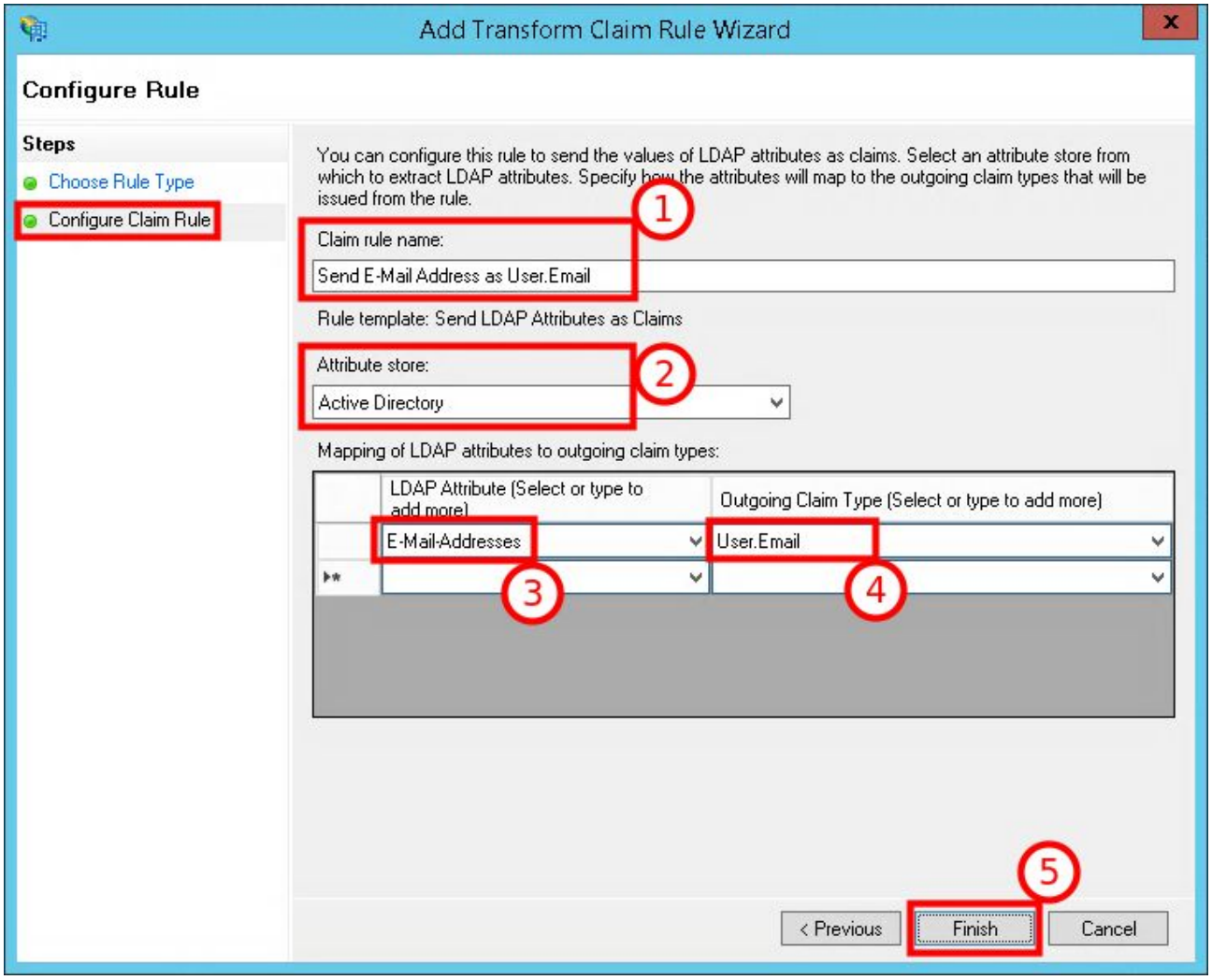
On **“Choose Rule Type”** page, please select **“Claim rule template”** as **“Send LDAP Attributes as Claims” (1)** and then click **“Next” (2)**



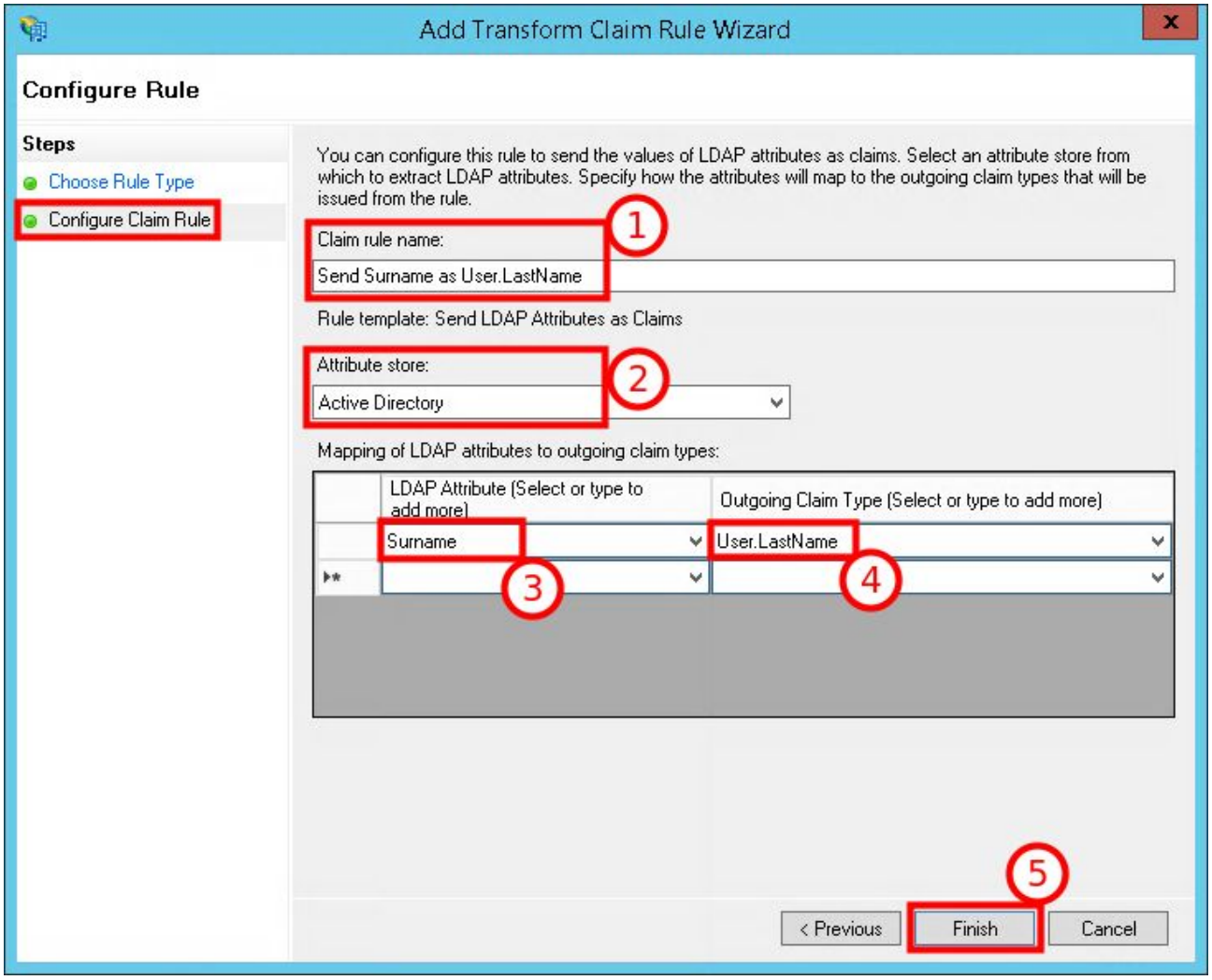
On page **“Configure Claim Rule”**, provide “Claim rule name” (1), then set “Attribute store” (2) to “Active Directory”.
In **“LDAP Attribute (Select or type to add more)”** select **“E-Mail-Addresses” (3)**.
In **“Outgoing Claim Type (Select or type to add more)”** select **“E-Mail Address” (4)** and click **“Next” (5)**.



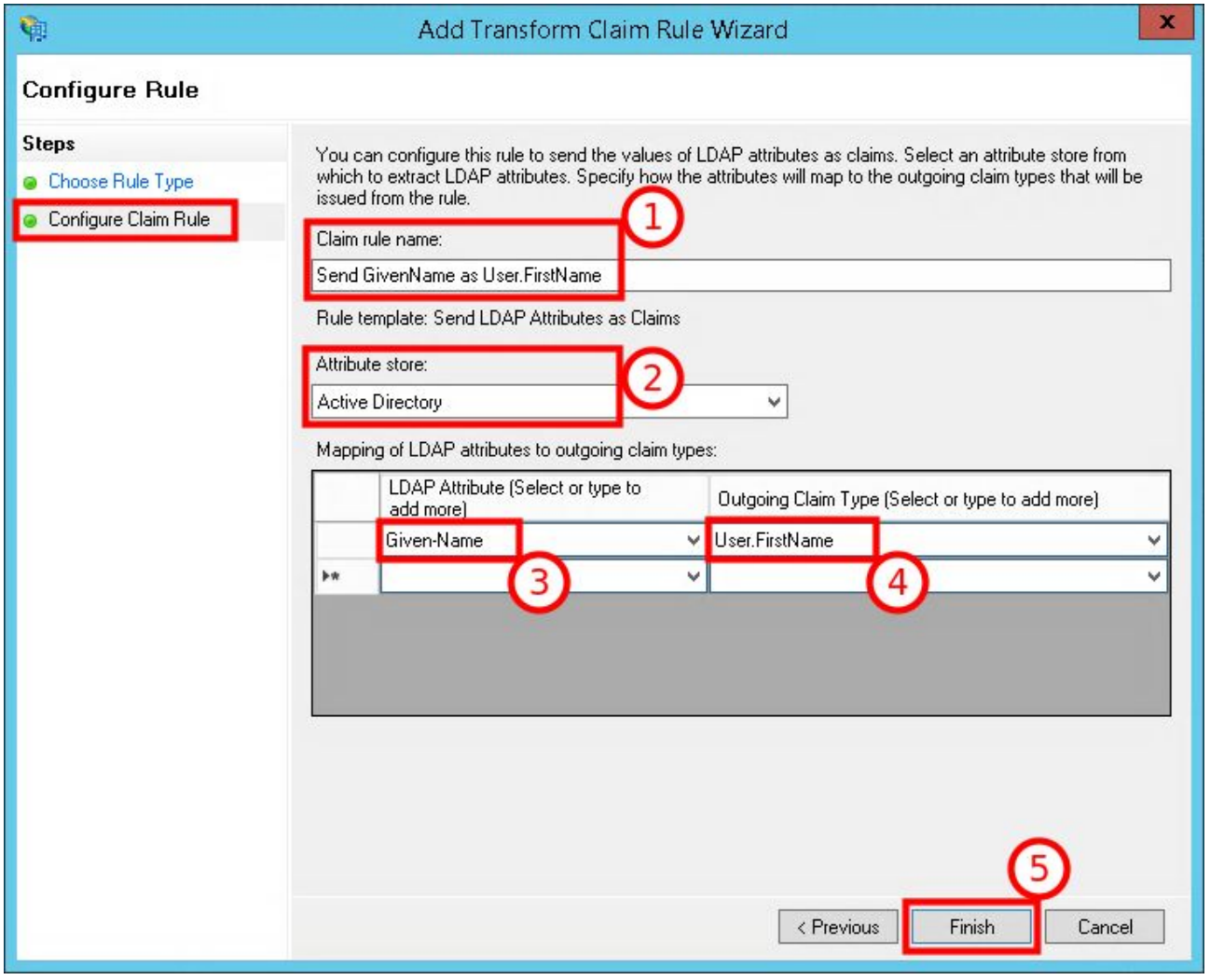
4/ Add rule to send “E-Mail Address” as “User.Email”. Follow analogical steps as described for previous rule. Please use **“Send LDAP Attribute as Claims”** template.
Then on **“Configure Claim Rule”** please provide **“Claim rule name” (1)**, set **“Attribute store”** to **“Active Directory” (2)**.
In **“LDAP Attribute (Select or type to add more)”** select **“E-Mail-Addresses” (3)**.
In **“Outgoing Claim Type (Select or type to add more)”** select **“E-Mail Address” (4)** and click **“Finish” (5)**.



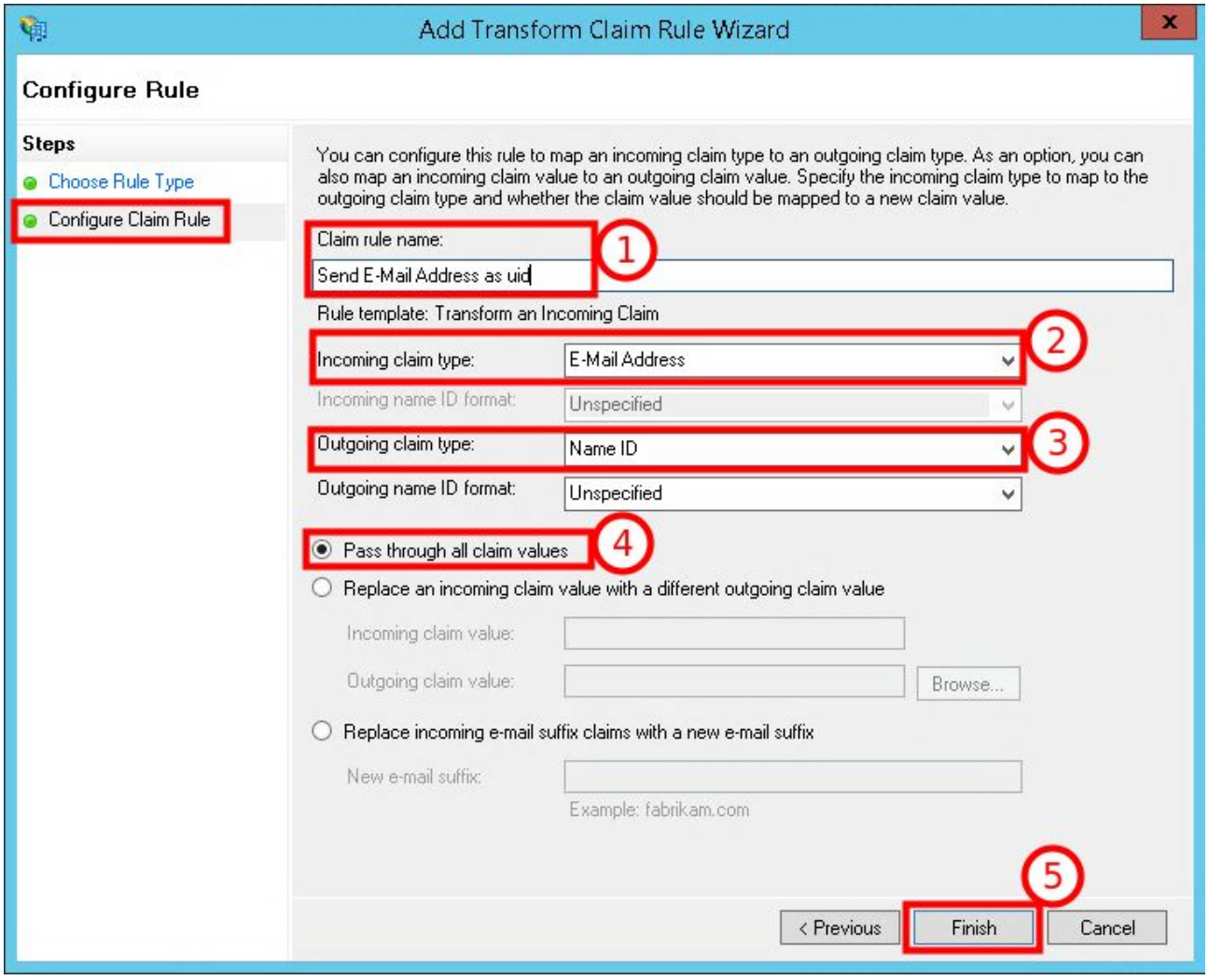
5/ Add rule to send “Surname” as “User.LastName”. Please use “Send LDAP Attribute as Claims” template.
On **“Configure Claim Rule”** please provide **“Claim rule name” (1)**, set **“Attribute store”** to **“Active Directory” (2)**.
In **“LDAP Attribute (Select or type to add more)”** select **“Surname” (3)**.
In **“Outgoing Claim Type (Select or type to add more)”** select **“User.LastName” (4)** and click **“Finish” (5)**.



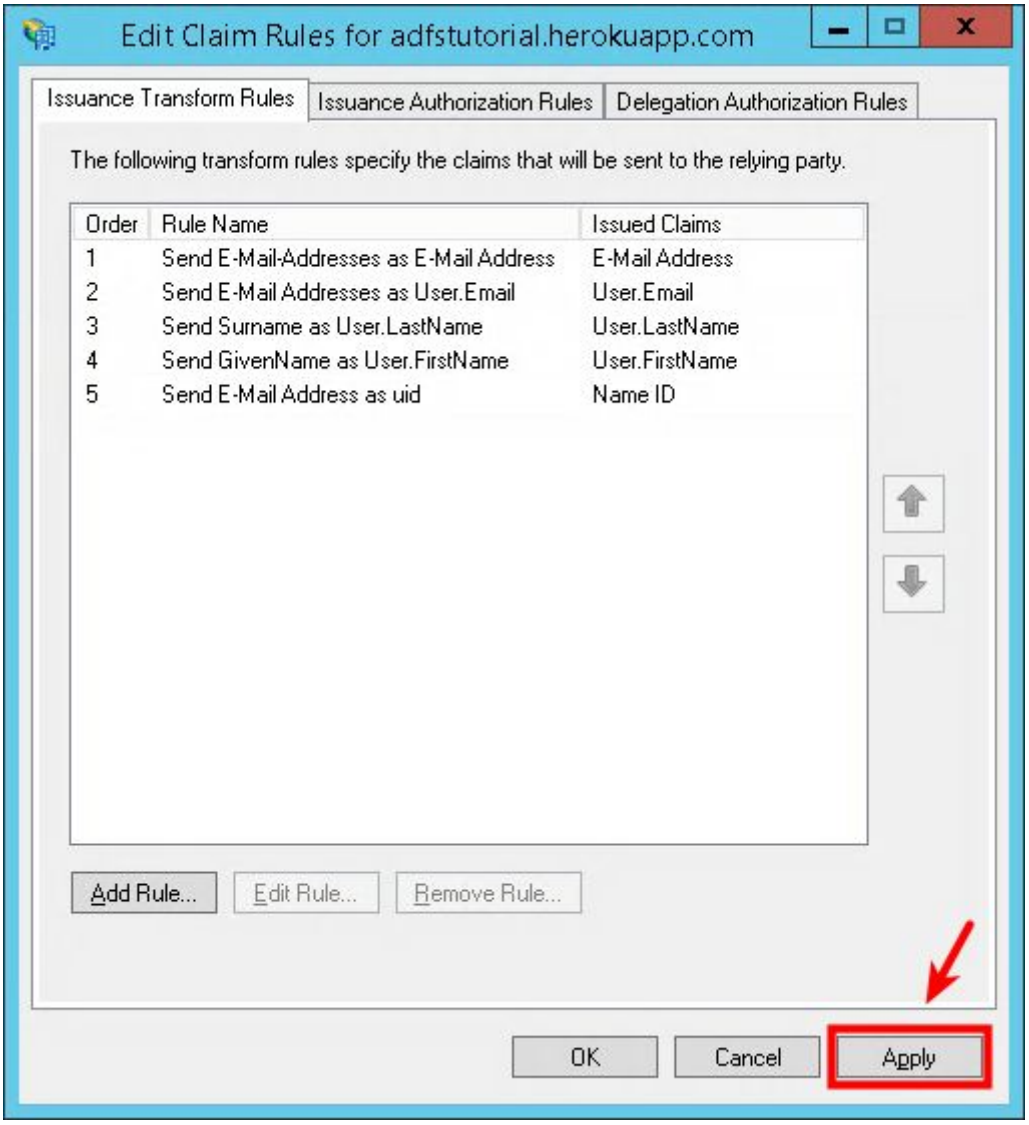
6/ Add rule to send “Given-Name” as “User.FirstName”. Please use **“Send LDAP Attribute as Claims”** template.
On **“Configure Claim Rule”** please provide **“Claim rule name” (1)**, set **“Attribute store”** to **“Active Directory” (2)**.
In **“LDAP Attribute (Select or type to add more)”** select **“Given-Name” (3)**.
In **“Outgoing Claim Type (Select or type to add more)”** select **“User.FirstName” (4)** and click **“Finish” (5)**.



7/ Finally, add rule to send “E-Mail Address” as “uid”. Please use **“Transform an Incomming Claim”** template.
On **“Configure Claim Rule”** please provide **“Claim rule name” (1)**.
Select **“Incoming claim type”** to **“E-Mail Address” (2)**.
Select **“Outgoing claim type”** to **“Name ID” (3)**.
Select **“Pass through all claim values” (4)** and click **“Finish” (5)**.



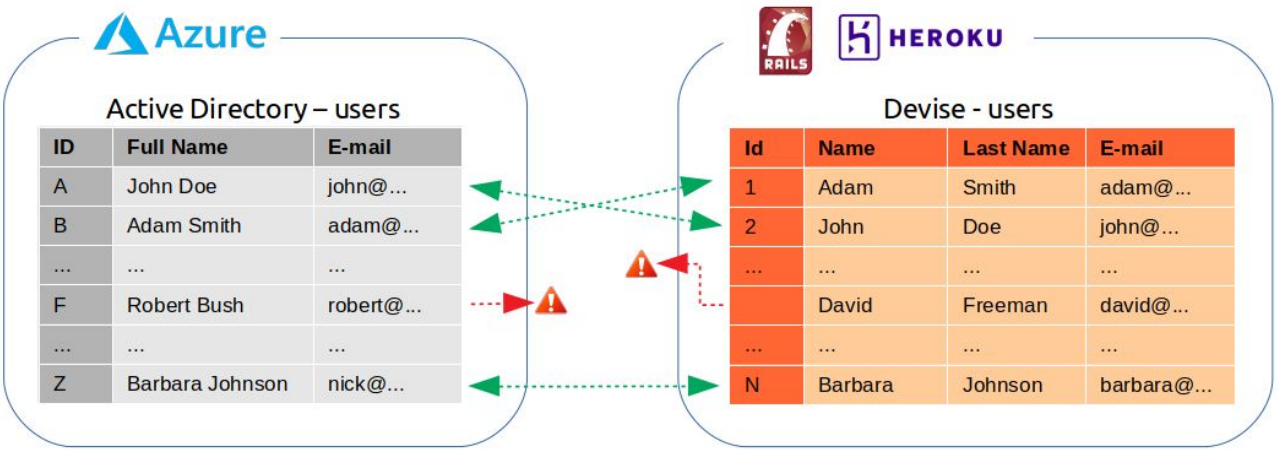
After this step list of claim rules should look as presented below:



Please click “Apply” to finish claim rules configuration.

4.7. Matching AD FS (Active Directory) accounts with RoR/Devise accounts

Single Sign On mechanism requires a link between users/accounts provided by Identity Provider and account hosted in the application (Service Provider).
Whenever user “U” from organization “O” tries to login to an application “A”, application receives SAML request with attributes which identify this user. Rails application has to find matching user in database and provide access/create session for that user.
Image below shows user records stored in Active Directory and in Ruby application.
Users can be matched based on certain properties which are same in both systems.
Not every record in Active Directory has to match another record in Rails application and vice versa.



If user does not exist yet in the Rails application (ex. “Robert Bush” in the example, exists only in Active Directory), one of following strategies can be used:

A) MOST COMMON: Application automatically creates corresponding user, assigns attributes from SAML payload (ex. first name, last name, e-mail address etc.), and then grants an access

B) Application rejects such a request. Access is denied.
In such scenario usually only application administrator can add new users to application database

Fragment below shows how users are matched in sample application: ***auth.uid*** field from SSO payload, which holds actually user email from ActiveDirectory, is used to find user in Rails database.

```
class User < ApplicationRecord
  # Include default devise modules. Others available are:
  # :confirmable, :lockable, :timeoutable and :omniauthable
  devise :database_authenticatable, :registerable,
         :recoverable, :rememberable, :trackable, :validatable,
         :omniauthable, omniauth_providers: [:saml]

  # Lookup for user matching authentication data from ADFS
  # More info: https://github.com/plataformatec/devise/wiki/OmniAuth:-Overview
  def self.from_omniauth(auth)
    Rails.logger.info("Trying to find user: uid #{auth.uid}, provider: #{auth.provider}")
    where(provider: auth.provider, uid: auth.uid).
        first_or_create(email: auth.uid, provider: auth.provider, uid: auth.uid, password: generate_password)
  end

  private

  def self.generate_password
    Devise.friendly_token.first(10)
  end
end
```

5. Test SSO login and logout

Many systems must undergo strict compliance procedures to ensure high security standards.
To ensure that your system follows high security standards, you should incorporate SSO tests into your standard tests which should be executed periodically - it can be before each release or only after changes which involves SSO code, depending on system requirements and/or compliance procedures.
Couple test scenarios are listed here.

TEST CASE
SP initiated login
SP initiated logout
IdP initiated login
IdP initiated logout
Disabled user account in ADFS / ActiveDirectory
Removed user account in ADFS / ActiveDirectory

6. Troubleshooting and Additional Setup for AD FS

6.1. Rails application rejects SSO login request due to time difference between AD FS server and Heroku server

Issue:
When time on AD FS machine is earlier than on Heroku server (ex. AD FS: 14:20:22, Heroku 14:20:24), Rails application rejects login request, and user can not login to application via SSO.

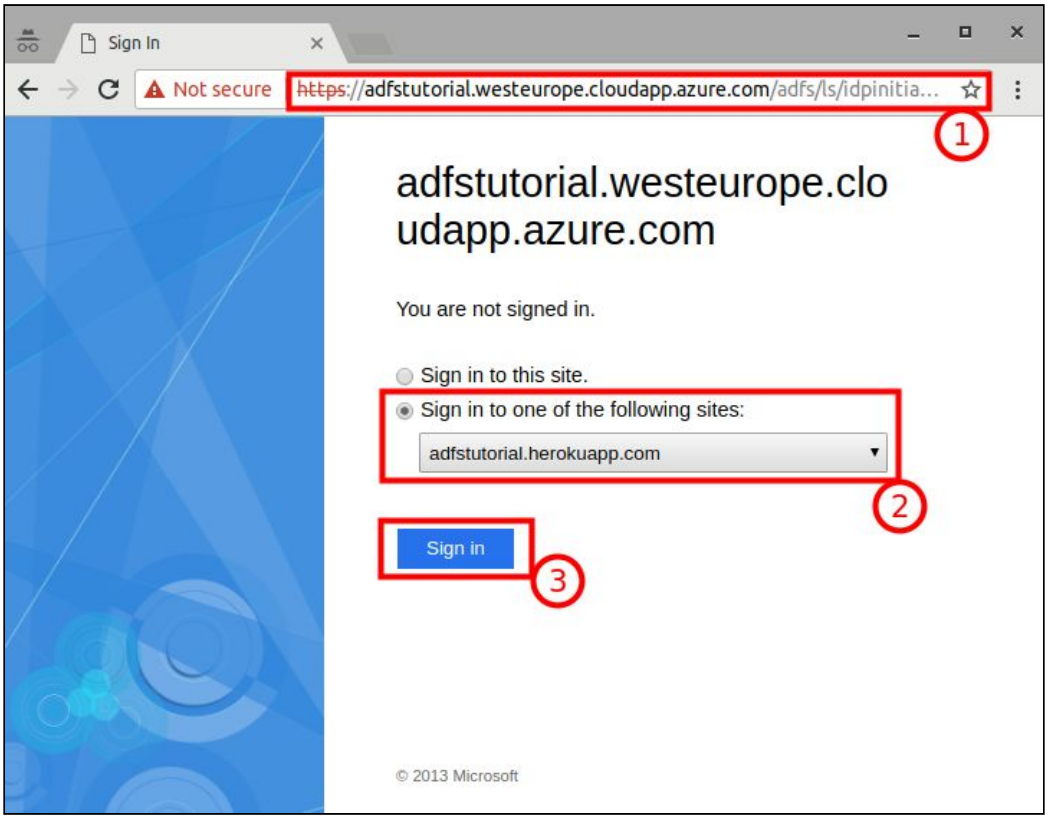
Fix:
Configure AD FS server to allow time drift, ex. 1minute.

```
Set-ADFSRelyingPartyTrust -TargetIdentifier "<replying party identifier>"
-NotBeforeSkew 1
```

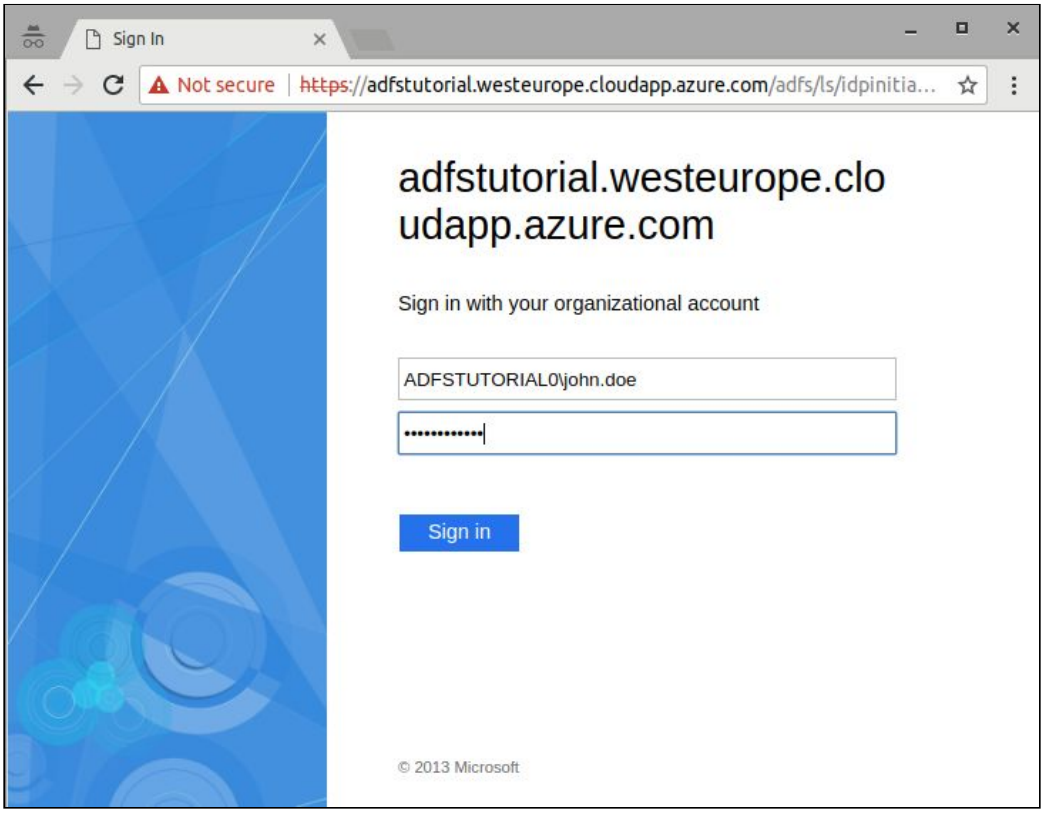
6.2. Fix issue with circular redirect for IdP initiated login

Issue:

1. Go to https://adfstutorial.westeurope.cloudapp.azure.com/adfs/ls/idpinitiatedsignon.aspx
2. Select your site from “Sign in to one of the following sites”
3. Click “Sign In”



On the following screen, please provide domain\username and password (ex. “ADFSTUTORIAL\john.doe” + password) and click “Sign In”.



Expected: User should be logged in to SSO and you should be redirected to your application.

Current: User is logged in, redirected to application and then redirected back to https://adfstutorial.westeurope.cloudapp.azure.com/adfs/ls/idpinitiatedsignon.aspx

Fix:
For IdP initiated login, default OmniAuth behavior redirects back to IdP login form (AD FS login form).
To avoid this situation `app/controllers/users/omniauth_callbacks_controller.rb` should be updated as follows (yellow highlight):

```
# frozen_string_literal: true
class Users::OmniauthCallbacksController < Devise::OmniauthCallbacksController
 skip_before_action :verify_authenticity_token, only: [:saml, :failure]

 def saml
  # You need to implement the method below in your model (e.g. app/models/user.rb)
  @user = User.from_omniauth(request.env["omniauth.auth"])

  if @user
   # Fix for IdP initiated login:
   # default implementation provided on Omniauth wiki page was causing redirect back to IdP login form.
   sign_in(@user)
   redirect_path = stored_location_for(resource) || signed_in_root_path(resource)
   redirect_to redirect_path
   set_flash_message(:notice, :success, kind: 'ADFS SSO') if is_navigational_format?
  else
   failure
  end
 end

 def failure
  redirect_to root_path
 end
end
```

6.3. Configure IdP initiated logout

To ensure that user will be logged out after certain amount of time, AD FS server should be configured properly.

```
Set-ADFSProperties -EnableOAuthLogout $true
```

Additional docs:
https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/development/ad-fs-logout-openid-connect

6.4. Ensure that you are accessing your application via HTTPS

When you test SSO login, ensure that you access your application url via HTTPS ex. https://adfstutorial.herokuapp.com, otherwise AD FS server will reject SSO request and will raise an error.
This can be tricky to determine because error is not shown - you have to check it in Windows AD FS logs:

```
Microsoft.IdentityServer.Service.Policy.PolicyServer.Engine.AssertionConsumerServiceUrlDoesNotMatchPolicyException: MSIS3200: No AssertionConsumerService is configured on the relying party
```

```
trust 'microsoft:identityserver:adfstutorial.herokuapp.com' that is a prefix match of the
AssertionConsumerService URL 'http://adfstutorial.herokuapp.com/users/auth/saml/callback'
specified by the request. at
Microsoft.IdentityServer.Service.SamlProtocol.EndpointResolver.LookupAssertionConsumerServiceBy
Url(Collection`1 assertionConsumerServices, Uri requestedAssertionConsumerServiceUrl, String
scopeIdentity) at [...]
```

6.5. Viewing logs

If anything goes wrong, please first view logs - both on Windows AD FS instance, and on application deployment server (Heroku):
on Windows Machine (AD FS):
Go to “Server Manager” > “Local Server” > “EVENTS”

on Heroku (Rails application):

```
heroku logs -t -a <application-name>
```

6.6 Listing AD FS properties

Listing all ADFS properties from PowerShell can be helpful during troubleshooting.
To list all properties, please run following command from Power Shell:

```
C:\> Get-ADFSProperties
```

6.7. AD FS service does not start after account password change

Please remember that AD FS service uses user credentials.
If you use your local account for AD FS service logon, please notice that changing password for that account will also require to update logon settings for AD FS service.
If you experience issue with AD FS service not starting due to login error, please go to “Services”, find “AD FS” service, click “Properties” and update password on “Logon” tab.

7. References

Promote Server to Domain Controller

https://social.technet.microsoft.com/wiki/contents/articles/12370.windows-server-2012-set-up-your-first-domain-controller-step-by-step.aspx

Setting up AD FS

https://doc.arcgis.com/en/arcgis-online/reference/configure-adfs.htm
https://mizitechinfo.wordpress.com/2015/01/08/simple-step-install-configure-adfs-in-windows-server-2012-r2/

Configure Devise and OmniAuth

Configure Devise: https://github.com/plataformatec/devise
Configure Omniauth with Devise: https://github.com/plataformatec/devise/wiki/OmniAuth:-Overview
Configure SAML authentication: https://github.com/omniauth/omniauth-saml

Configure Self Signed Certificate:

http://www.itprotoday.com/management-mobility/creating-self-signed-certificates-powershell
https://gallery.technet.microsoft.com/Self-signed-certificate-5920a7c6/view/Discussions/2
https://dimitri.janczak.net/2015/07/29/adfs-3-0-in-windows-2012-r2-self-signed-certificate