LDAP ve AD Kimlik Doğrulama

Docker üzerinde LDAP server'i oluşturmak için;

Gerekli olan Docker Pull Command "docker pull osixia/openldap"

ile gerekli dosyaları çekeriz. Ardından server oluşturmak için

"docker run --detach osixia/openldap:1.1.8"

ile server oluşturabiliriz.

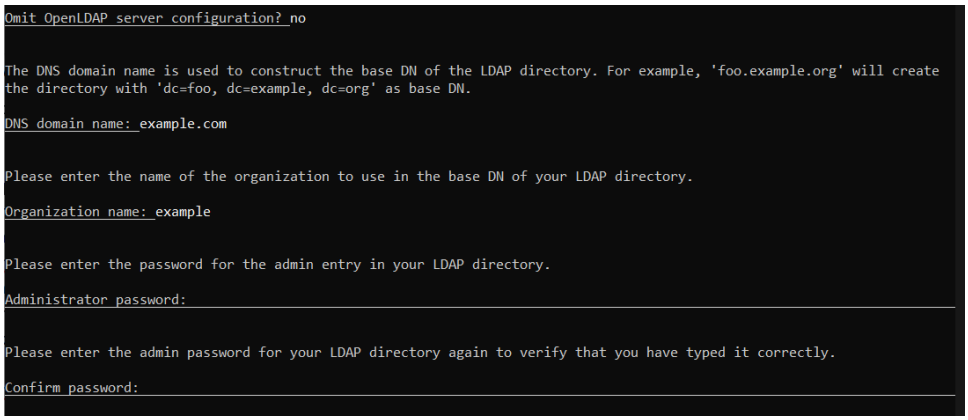
- Port kullanmak için "-p 10389:389" komutunu server oluştururken run edersek serverimiz o port üzerinde çalışmış olacaktır.
- Server'imize isim vermek için "--name openldap-container" komutunu yazarsak serverimizin ismi openldap-container olacaktır.
- Server'e Domain Name eklemek için "--env LDAP_DOMAIN=example.com" komutunu kullanabiliriz.
- Organization name eklemek için "--env LDAP_ORGANISATION=example" komutunu kullanabiliriz.
- Server'imizin Password'unu ise "--env LDAP_ADMIN_PASSWORD=password" komutunu kullanarak oluşturabiliriz.

Server bilgilerini Server'i oluşturduktan sonra ayarlamak veya değiştirmek istersek eğer;

Gerekli Container CLI 'ına girip

- apt-get update
- apt-get install slapd ldap-utils
- dpkg-reconfigure slapd

Komutlarını kullandıktan sonra karşımıza şöyle bir ekran gelmekte



Gerekli ayarlamaları böyle yapabiliriz.

Docker üzerinde phpldapadmin arayüzü oluşturmak için;

Gerekli olan Docker Pull Command “docker pull osixia/phpldapadmin”

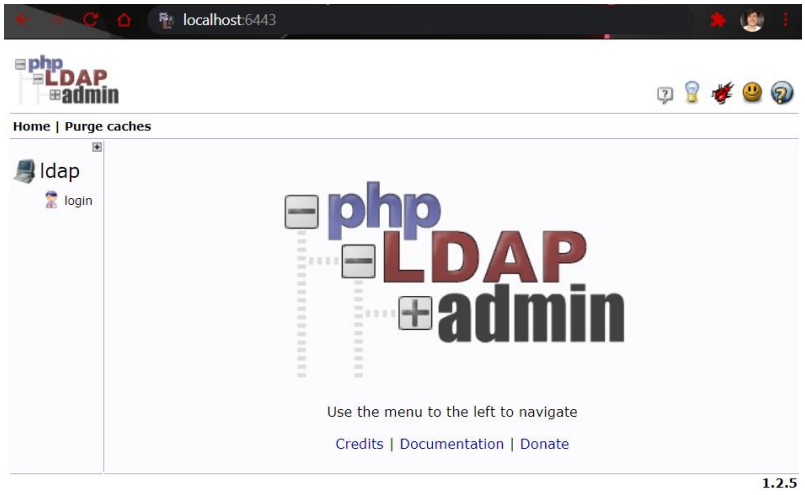
ile gerekli dosyaları çekeriz. Ardından arayüz oluşturmak için

“docker run -p 6443:443 \

--env PHPLDAPADMIN_LDAP_HOSTS=ldap.example.com \

--detach osixia/phpldapadmin:0.9.0”

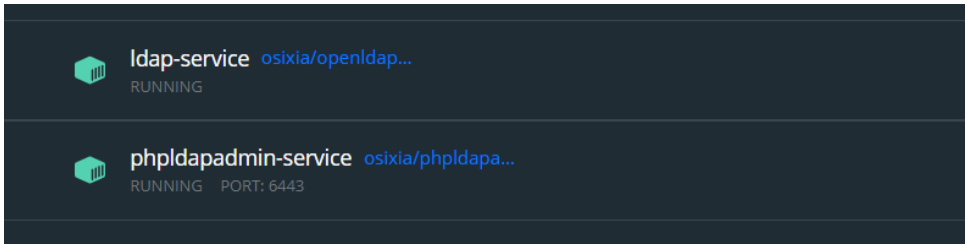
ile arayüz oluşturabiliriz. Bu komutu kullandıktan sonra https://localhost:6443 adresinde phpldapadmin arayüzümüz bizi karşılayacaktır.



Login işlemi için bu arayüzümüz ile ldap serverimiz arasındaki bağlantıyı sağlamamız gerekmektedir. Bu işlemi Containerleri oluştururken yapmak istersek gerekli komut satırı:

“ docker run --name ldap-service --hostname ldap-service --detach osixia/openldap:1.1.8 docker run -p 6443:443 --name phpldapadmin-service --hostname phpldapadmin-service --link ldap-service:ldap-host --env PHPLDAPADMIN_LDAP_HOSTS=ldap-host --detach osixia/phpldapadmin:0.9.0 “

Bu komut satırını kullandıktan sonra docker üzerinde Containerlerimiz şu şekilde gözükecektir:



https://localhost:6443 adresine gittiğimiz zaman login işlemini gerçekleştirebiliriz.

Ldap serverimizi oluştururken veya oluşturduktan sonra server ayarlarını yapmadığımız zaman default olarak o ayarlamaları şu şekilde yapmaktadır:

```
# slapcat
dn: dc=example,dc=org
objectClass: top
objectClass: dcObject
objectClass: organization
o: Example Inc.
dc: example
structuralObjectClass: organization
entryUUID: 9e15be50-a0d9-103b-9d80-db4bc77fc27b
creatorsName: cn=admin,dc=example,dc=org
createTimestamp: 20210903080642Z
entryCSN: 20210903080642.150114Z#000000#000#000000
modifiersName: cn=admin,dc=example,dc=org
modifyTimestamp: 20210903080642Z

dn: cn=admin,dc=example,dc=org
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9R1JZckhkTGVqQlNEeXVrakdPY3ZkNUhBUVliMGsvZmg=
structuralObjectClass: organizationalRole
entryUUID: 9e179b6c-a0d9-103b-9d81-db4bc77fc27b
creatorsName: cn=admin,dc=example,dc=org
createTimestamp: 20210903080642Z
entryCSN: 20210903080642.162323Z#000000#000#000000
modifiersName: cn=admin,dc=example,dc=org
modifyTimestamp: 20210903080642Z
```

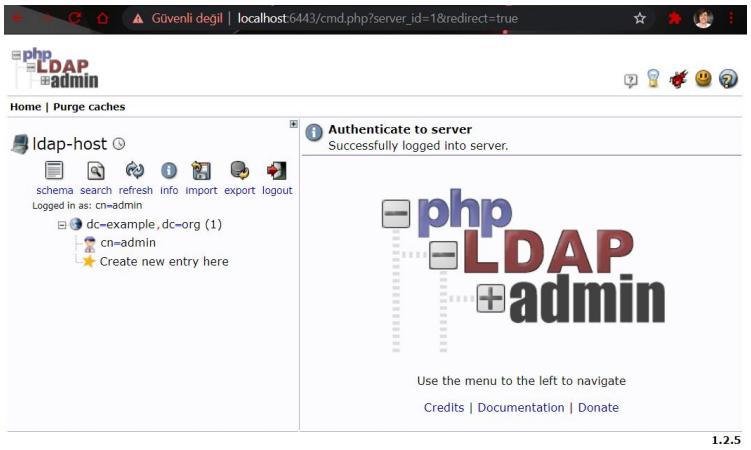
Login olurken

Login Dn = “cn=admin,dc=example,dc=org”

Password = “admin”

Şeklinde giriş yapabiliriz. Böyle bir ekran ile karşılaşacağız:

(Default yerine dpkg-reconfigure slapd ile verileri değiştirip o bilgilerle de giriş yapabiliriz.)



Spring Boot + OpenLDAP ile LDAP kimlik doğrulaması

Sıfırdan ldap server ve arayüzü oluşturup bağlantı ve ayarlarını yaparsak:

“docker run -p 10389:389 --name openldap --hostname openldap --env LDAP_ORGANISATION=example --env LDAP_DOMAIN=example.com --env LDAP_ADMIN_PASSWORD=password --detach osixia/openldap:1.1.8

docker run -p 18080:80 --name phpldapadmin --hostname phpldapadmin --link openldap:ldap --env PHPLDAPADMIN_LDAP_HOSTS=ldap --env PHPLDAPADMIN_HTTPS=false --detach osixia/phpldapadmin:0.9.0 ”

komutunu kullanırız. Buna bağlı Containerlerimiz şu şekilde olmakta

Login DN: “cn=admin,dc=example,dc=com”

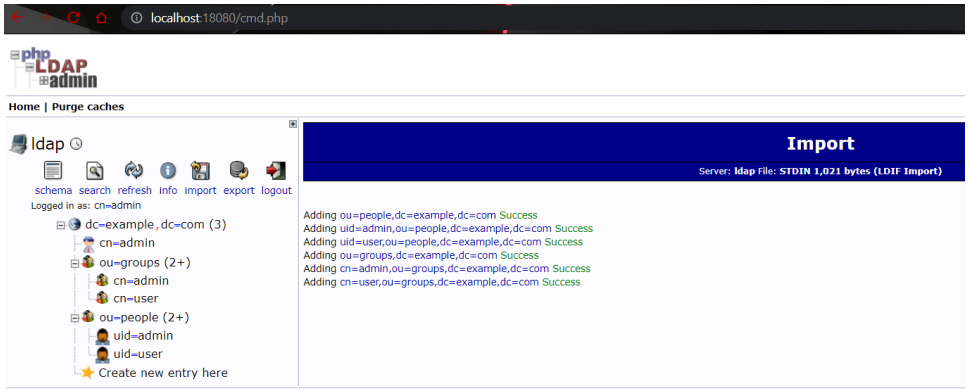
Password: “password”

Olarak giriş yaptığımızda import ile birkaç kullanıcı ekleyelim.

(Ldif dosyam gereksiz yer kaplamasın diye bir drive linki içinde tutuyorum.

https://drive.google.com/drive/folders/1Fs-bNEEgWp8A7eRAWUcQprpPOTc25y_e?usp=sharing)

İmport ettiğim bu ldif dosyasının ardından karşımıza şöyle bir ekran gelmekte:

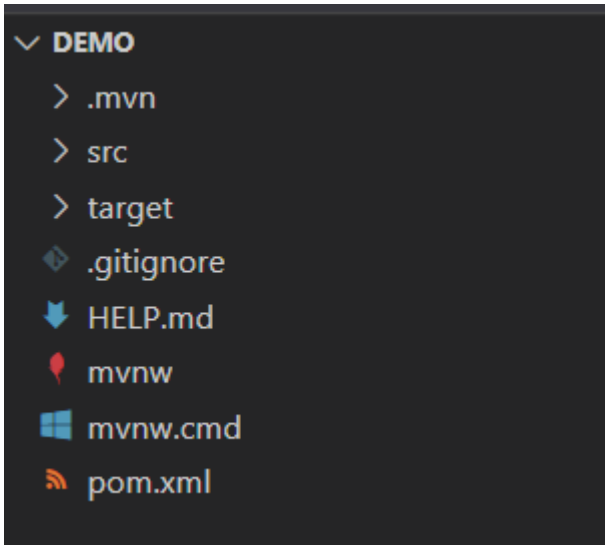


Grupları ve kişileri bu şekilde ayarladık. Ardından kullanıcı kontolünü sağladığımız kısma geçersek:

- İlk olarak start.spring.io adresine gidiyoruz. Karşımıza çıkan ekranda projemiz için gerekli olan verileri giriyoruz.
- Dependencies kısmından ise birkaç eklememiz gereken şey bulunmakta.
- Bunlar “Spring Web”,”Spring Security” ve “Thymeleaf”
- Ekleme işlemlerini yaptıktan sonra GENERATE ile bir dosya oluştururuz.

Rardan çıkardığımız bu dosyayı masaüstüne alıp PowerShell ile “cd /desktop/demo” ve “code .” ile dosyanın kodlarını karşımıza çıkartırız.

Dosyalarımız şu şekilde olmaktadır:



Pom.xml ‘e girip yoksa eğer

```
<dependency>
        <groupId>org.springframework.security</groupId>
        <artifactId>spring-security-ldap</artifactId>
</dependency>
<dependency>
        <groupId>org.thymeleaf.extras</groupId>
        <artifactId>thymeleaf-extras-springsecurity5</artifactId>
</dependency>
```

Bunları depencies blokları arasına ekliyoruz.

Src/main/java/com/example/demo yolunda bir HomeController.java isimli bir class ekleyelim.

src > main > java > com > example > demo > HomeController.java > HomeController

```java
package com.example.demo;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;

@RequestMapping("/")
@Controller
public class HomeController {
    @GetMapping
    String home() {
        return "home";
    }
}
```

Giriş ekranından sonra bizi karşılayacak html sayfasına ulaşmak için böyle bir class tanımlıyoruz.

Src/main/resources/templates package içerisine home.html oluşturuyoruz.

Çıkan ekranda kullanıcının ismini yazdıracak bir label ve logout yapabileceği bir buton oluşturuyoruz.

src > main > resources > templates > home.html > html

```html
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org"
      xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity4">
<head>
  <title>Home</title>
</head>
<body>
<div>
  <h1>Login: <span sec:authentication="principal.cn[0]">User</span> !</h1>
</div>

<form th:action="@{/logout}" method="post">
  <button>Logout</button>
</form>
</body>
</html>
```

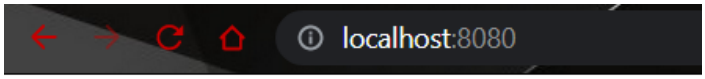
Son olarak ekranımızın phpldapadmin ile bağlantısını sağlamak,güvenliği sağlamak için demo package içerisine SecurityConfig.java adında bir class oluşturuyoruz.

```java
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.ldap.userdetails.PersonContextMapper;

@Configuration
public class SecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests()
                .antMatchers("/**").authenticated()
                .and()
                .logout().permitAll()
                .and()
                .formLogin().permitAll();
    }

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        auth.ldapAuthentication()

                .userDnPatterns("uid={0},ou=people")
                .groupSearchBase("ou=groups")
                .contextSource()
                .url("ldap://localhost:10389/dc=example,dc=com")
                .managerDn("cn=admin,dc=example,dc=com")
                .managerPassword("password")
                .and()
                .userDetailsContextMapper(new PersonContextMapper());
    }
}
```

O class içine de gerekli adapter'i extend edip fonksiyonlarımızı yazıyoruz.

Yaptığımız işlem bu kadar projeyi run ediyoruz.

Default olarak localhost:8080 e atmakta. Bu sayfada ldif ile oluşturduğumuz kullanıcı adı ve şifreyi girdiğimiz zaman;



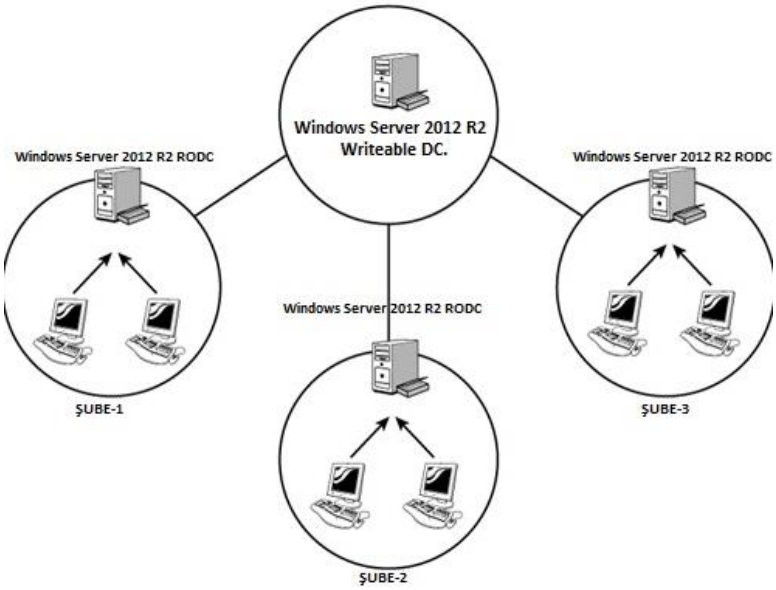
Hello Andasis !

Logout

Kullanıcı girişimizi yapmış bulunmaktayız.

Active Directory ile Kimlik Doğrulama

Domain, mevcut ağı yönetmek için kurulan etki alanıdır. Oluşturmak için server işletim sistemine ihtiyaç duyulur. Projemizde Windows Server'i kullanacağız. İşletim sisteminde oluşturduğumuz bu makine bizim domainin yöneticisi olacak yani bizim Domain Controller (DC) denilen yetkilendirilmiş makinemiz olacak.



Şekildeki şubeler birer "Child domain"dir. Root "Parent Domain"dir ve child domainlerin kontrolünü sağlar. Bu yapının tamamı 'Tree' olarak isimlendirilir. Birden fazla tree olursa o yapıyı 'Forest' olarak adlandırırız.

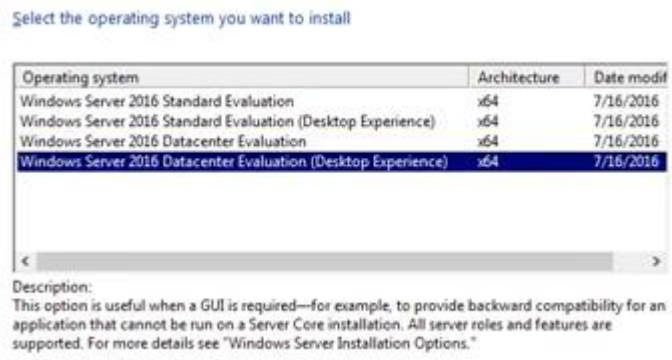
Active Directory ise bu domain'i kurabilmek için ve daha sonra yönetebilmek için kullanacağımız araca verilen isimdir.

Windows Server Microsoft tarafından geliştirilmiş bir işletim sistemidir. Normal Windows işletim sistemlerinden farkı ise yönetim için geliştirilmiş olmasıdır.

Active Directory Kurulumu

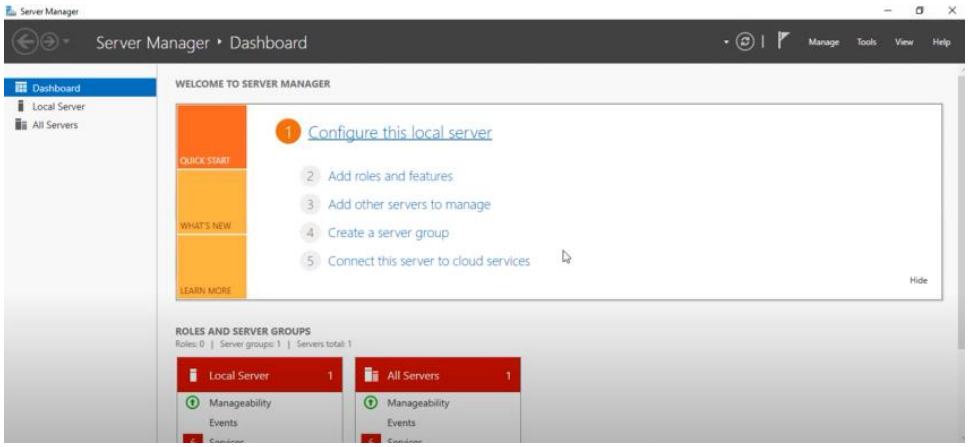
https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2022

Adresinden ISO dosyasını indirip bilgisayarımıza Windows Serverimizi kuruyoruz.



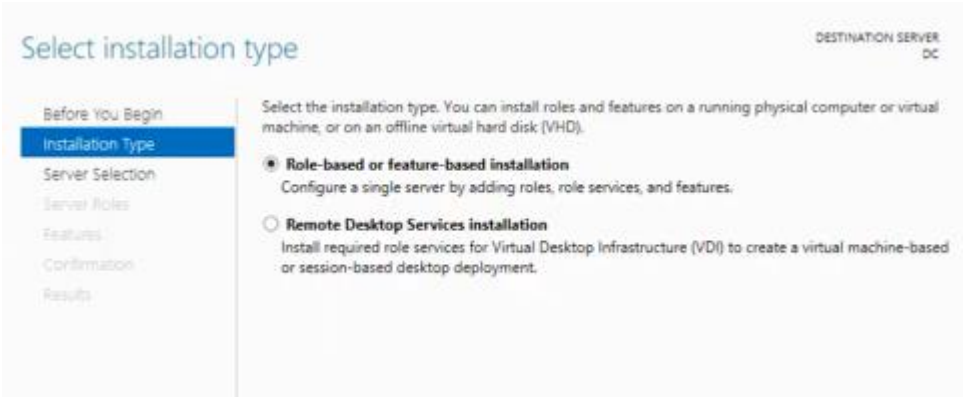
!! Standard sürüm 50 makineye kadar destek verirken Datacenter daha fazla makineye destek sağlamaktadır.

!! Desktop Experience seçeneğini işaretlersek windows arayüzü karşımıza çıkacaktır diğer seçenekte sadece komutlarla yönetebileceğimiz bir server oluşacaktır.

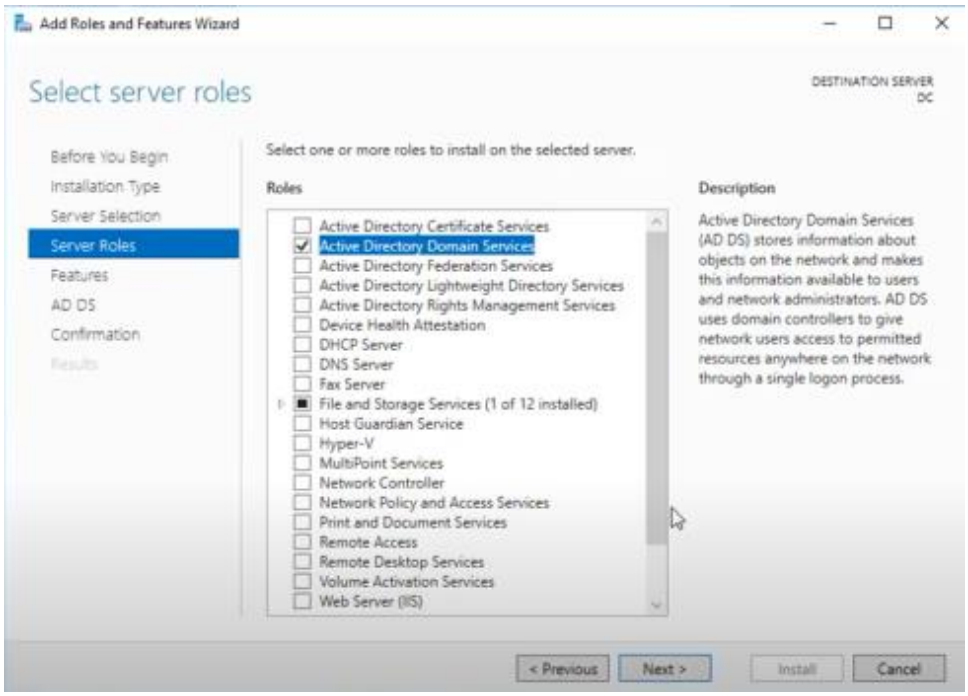


Windows Server kurulumun ardından Server Manager üzerinden Active Directory kurulumunu yapmak için;

- Add roles and fetures seçeneğine tıklayalım

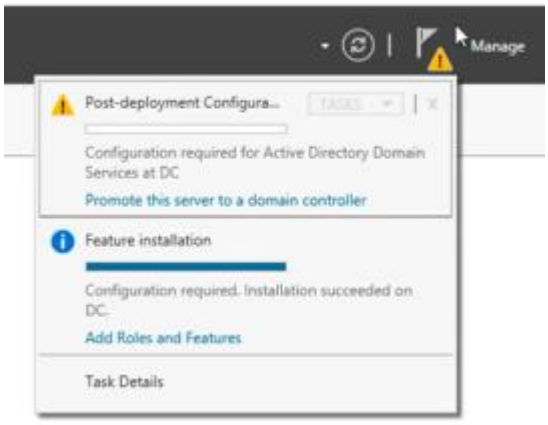


- Uzaktan yükleme yerine rol ve özellik kuracağımız için onun seçimini yapıyoruz.
- Ardından çıkan ekrandan gerekli DC'mizi seçiyoruz.



- Burada kuracağımız rol Active Directory Domain Services olduğu için onun seçimini yapıyoruz.
- Ardından gerekli ilerlemeleri yaptıktan sonra ilgili servisimiz kurulmuş olacaktır.

Kurulumu yaptık yapılandırmasını gerçekleştirmek için



Promote this server to a domain controller seçeneğine tıklıyoruz. Bu sayede oluşturduğumuz servisten bir domain oluşturup makinemizi bu domainin DC si olarak yetkilendirmiş oluyoruz.

!! Kurulumun ardından kullanıcı ve bilgisayarları görmek için

Control Panel > System and Security > Administrative Tools > Active Directory Users and Computers

Şeklinde ilerlememiz gerekmektedir.

Oluşturduğumuz Active Directory üzerinde kimlik doğrulama yapmak için 5-7 sayfa aralığındaki adımları yerine getirmemiz yeterli olacaktır. Tek fark SecurityConfig classımızın içeriğini şu şekilde değiştiriyoruz :

https://drive.google.com/drive/folders/1Fs-bNEEgWp8A7eRAWUcQprpPOTc25y_e?usp=sharing