domain

- 컴퓨터 계정이나 사용자 계정 같은 객체들을 지역적으로 모아 놓은 개념

> 논리적으로 인증과 관리가 동일한 영역

> 윈도우 환경이 파일/폴더를 모아 놓은 환경이라면, 도메인 환경은 계정(객체)들을 모아

놓은 환경

- 도메인을 확장하여 포리스트를 만들거나, 일반 컴퓨터를 도메인에 가입시켜서 사용 및 관리가 가능

OU (조직구조; Organizational Unit)

- 윈도우 환경에서 폴더를 이용하여 파일을 묶어준다면(폴더에 파일을 담아 놓는다면)

도메인 환경에서는 OU를 이용하여 계정들을 묶어준다(OU에 계정을 담아 놓는다).

- 관리 위임의 단위

> 관리 위임을 할 때 최소 단위가 OU이기 때문

forest

- domain tree 집합

> domain은 하위 domain 생성이 가능 >> tree 구조로 구성이 된다  
 domain tree들 간의 신뢰관계를 맺어 확장되는 개념이 forest

> tree 구조를 구별하는 방법 : domain 이름 확인  
 하위 domain은 상위 domain 이름 앞에 자신의 이름을 붙인 형태로 주소가 정해진다.

- 신뢰관계에 놓여있는 domain들의 집합

> 신뢰관계 : 서로 믿을 수 있다 >> 서로 접근했을 때 인증을 통하여 자원을 이용할 수 있는

관계

- forest 내부에는 GC (global catalog) 라는 forest 의 database가 설치

> GC는 forest의 root domain에는 무조건 설치, 그 외의 domain들은 선택하여 설치

> GC에는 forest의 모든 객체의 정보가 저장

- forest의 이름은 root domain의 이름을 따라간다.

domain controller

- AD 정보의 복사본을 가지고 있는 컴퓨터

> AD의 정보 요청에 대한 응답이 가능

- 네트워크를 통한 사용자 인증

> 모든 정보를 가지고 있기 때문에 AD에서 사용자 인증이 가능

- domain controller는 DNS서버와 반드시 통합되어 있다

> domain 환경에서는 모든 서비스의 진행을 이름으로 진행하기 때문에 DNS서버가 반드시   
 필요

> domain controller를 설치하면 DNS서버가 함께 설치

도메인에 가입 후 로그인 방법

- 로컬 로그온, 도메인 로그온 두 가지 방법으로 분류

- 로컬 로그온

> 각 컴퓨터에 존재하는 계정으로 로그온

> 컴퓨터의 SAM 파일에서 인증을 받아 로그온 진행

(c:\windows\system32\config\SAM)

> a 컴퓨터의 로컬 계정을 b 컴퓨터에서 로그온 x

(b 컴퓨터 SAM파일에는 해당 계정의 정보가 없기 떄문)

> 네트워크를 통하여 다른 컴퓨터에 접근하면 인증창이 출력

(해당 컴퓨터의 계정으로 로컬 로그온 하라는 의미)

>워크 그룹 환경에서 사용

(워크 그룹 : workgroup; MS에서 만든 물리적인 네트워크 망)

> DC로 승격을 시키게 되면 로컬 계정은 전부 사라진다

(로컬 계정이 도메인 계정으로 변경)

- 도메인 로그온

> 도메인에 존재하는 계정으로 로그온

> DC의 NTDS.dit 파일에서 인증을 받아 로그온 진행

( C:\windows\ntds\ntds.dit )

> 통합 인증을 받기 때문에 도메인에 가입한 컴퓨터들은 어디서든 도메인의 계정으로

로그온 가능 ( 도메인 망에서 다른 컴퓨터에 접근하면 인증 X >> 이미 DC에서 인증을

받았기 때문)

- 로그온 방법

> 로컬 로그온

계정명

컴퓨터 이름\계정명

> 도메인 로그온

계정명 (단, 같은 이름의 계정이 로컬에 없을 때)

NetBIOS\계정명  
 계정명@도메인주소