

Codis de Automatització de la Configuració

INDEX:

ANSIBLE CODIS:.....	3
Automatització de serveis i seguretat bàsica amb Ansible.....	4
Desplegament automàtic d'una web estàtica amb Ansible	5
Automatització de polítiques de seguretat amb Ansible.....	6
PowerShell DSC.....	8
Aprovisionament automàtic amb PowerShell DSC.....	9
Gestió de configuració amb PowerShell DSC.....	10
Desplegament d'una aplicació web amb DSC.....	11
Definició i aplicació de polítiques de seguretat amb PowerShell DSC.....	13

ANSIBLE CODIS:

Automatització de serveis i seguretat bàsica amb Ansible

Playbook d'aprovisionament bàsic amb Ansible

En aquesta fase hem creat un **playbook bàsic d'aprovisionament** per instal·lar paquets essencials i crear un usuari nou de forma automatitzada. Aquesta tasca és habitual en qualsevol empresa quan es despleguen nous servidors, i garanteix que tots tinguin les mateixes eines bàsiques i usuaris inicials per a tasques de manteniment o desenvolupament.

Aquest tipus d'automatització és útil per establir una base comuna de configuració en entorns de desenvolupament, proves o producció, evitant errors manuals i estalviant temps.

A continuació es mostra el contingut complet del fitxer aprovisionament.ym

- hosts: linux_clients

become: yes

tasks:

- name: Actualitzar tots els paquets (apt update)

apt:

update_cache: yes

- name: Instal·lar paquets essencials (htop, curl, vim)

apt:

name:

- htop

- curl

- vim

state: present

- name: Crear usuari nou automàticament

user:

name: usuari1

state: present

create_home: yes

shell: /bin/bash

Desplegament automàtic d'una web estàtica amb Ansible

En aquesta fase hem creat un **playbook amb Ansible** que automatitza la publicació d'una pàgina web estàtica en els servidors Linux clients. L'objectiu és copiar de manera eficient fitxers HTML i CSS al directori web `/var/www/html/`, assegurant els permisos i propietaris adequats perquè el servidor web (Apache o Nginx) pugui servir-los correctament.

Aquest procés és útil a qualsevol empresa que vulgui desplegar webs internes, portals informatius o recursos corporatius de forma ràpida i estandarditzada.

A continuació mostrem el contingut complet del playbook `desplegament_web.yml`:

- hosts: linux_clients

become: yes

tasks:

- name: Copiar web estàtica (HTML i CSS) al servidor client

copy:

src: web_estatica/

dest: /var/www/html/

owner: www-data

group: www-data

mode: '0755'

Automatització de polítiques de seguretat amb Ansible

En aquesta fase hem implementat un playbook per aplicar una sèrie de polítiques bàsiques de seguretat de forma automatitzada sobre els clients Linux. Aquestes mesures permeten reforçar la seguretat del sistema i assegurar un comportament coherent a través de diversos servidors.

L'objectiu principal és garantir una configuració segura per defecte, evitant accés root remot, assegurant actualitzacions automàtiques, i establint un usuari administrador amb accés sudo sense contrasenya.

A continuació mostrem el contingut complet del playbook seguretat.yml:

```
---

- hosts: linux_clients

  become: yes

  tasks:

    - name: Instal·lar actualitzacions automàtiques

      apt:

        name: unattended-upgrades

        state: present

    - name: Activar servei d'actualitzacions automàtiques

      systemd:

        name: unattended-upgrades

        enabled: yes

        state: started

    - name: Afegir usuari administrador 'adminpaumiro'

      user:

        name: adminpaumiro

        shell: /bin/bash

        groups: sudo

        append: yes

        create_home: yes

        state: present
```

- name: Assegurar que l'usuari 'adminpaumiro' té accés sudo sense contrasenya

copy:

dest: /etc/sudoers.d/adminpaumiro

content: "adminpaumiro ALL=(ALL) NOPASSWD:ALL"

mode: '0440'

- name: Desactivar l'accés SSH per a root

lineinfile:

path: /etc/ssh/sshd_config

regexp: '^PermitRootLogin'

line: 'PermitRootLogin no'

state: present

- name: Reiniciar servei SSH

service:

name: ssh

state: restarted

PowerShell DSC

Aprovisionament automàtic amb PowerShell DSC

Aprovisionament automàtic amb PowerShell DSC

En aquesta fase hem implementat un script de configuració DSC per desplegar una aplicació web bàsica de forma automàtica en un servidor Windows. L'objectiu és copiar fitxers HTML i CSS a la ruta predeterminada del servei IIS (C:\inetpub\wwwroot) i assegurar que el contingut es manté sempre present.

Aquest tipus d'aprovisionament és molt útil en entorns empresarials on cal garantir que cada servidor disposi dels mateixos fitxers i serveis configurats automàticament, evitant errors manuals i facilitant una gestió més eficient.

Aquesta fase inclou:

- La creació dels fitxers index.html i style.css localment.
- L'escriptura d'un script PowerShell DSC que copia els fitxers i directoris a la ubicació final.
- L'aplicació de la configuració mitjançant Start-DscConfiguration.

```
Configuration WebCompleta {
```

```
    Import-DscResource -ModuleName PSDesiredStateConfiguration
```

```
    Node localhost {
```

```
        File Index {
```

```
            SourcePath = "C:\DSC\web\index.html"
```

```
            DestinationPath = "C:\inetpub\wwwroot\index.html"
```

```
            Ensure = "Present"
```

```
            Type = "File"
```

```
        }
```

```
        File Estils {
```

```
            SourcePath = "C:\DSC\web\css"
```

```
            DestinationPath = "C:\inetpub\wwwroot\css"
```

```
            Ensure = "Present"
```

```
            Recurse = $true
```

```
            Type = "Directory"
```

```
        }
```

```
    }
```

```
}
```

Gestió de configuració amb PowerShell DSC

Gestió de configuració amb PowerShell DSC

En aquesta fase hem aprofundit en un dels punts més potents de PowerShell DSC: **la capacitat de mantenir l'estat desitjat al llarg del temps**. Aquest enfocament no només permet desplegar configuracions, sinó també detectar i corregir canvis manuals que puguin posar en risc la coherència del sistema.

Aquesta gestió contínua és molt útil en entorns empresarials per garantir que cap administració manual o error humà alteri la configuració establerta. Això ajuda a mantenir una infraestructura segura, estable i predictable.

Objectius principals:

- Detectar canvis manuals al sistema (per exemple, modificació d'un fitxer web).
- Comprovar si l'estat actual coincideix amb la configuració DSC.
- Reaplicar la configuració per restaurar l'estat original automàticament.
- Consultar l'estat actual aplicat per DSC.

1. Modificar manualment el fitxer index.html

```
Set-Content "C:\inetpub\wwwroot\index.html" -Value "<h1>Pàgina modificada manualment!</h1>"
```

2. Comprovar si el sistema ha canviat

```
Test-DscConfiguration
```

Si retorna False, vol dir que l'estat actual **no coincideix** amb la configuració DSC.

3. Reaplicar la configuració DSC per restaurar l'estat original

```
Start-DscConfiguration -Path .\WebCompleta -Wait -Verbose -Force
```

4. Verificar que s'ha restaurat

```
Get-Content "C:\inetpub\wwwroot\index.html"
```

Hauries de veure de nou:

```
<h1>Servidor configurat amb PowerShell DSC!</h1>
```

5. Consultar l'estat actual aplicat

```
Get-DscConfiguration
```

Aquesta comanda ens mostra totes les configuracions actuals en vigor: serveis actius, fitxers gestionats, i característiques instal·lades.

Desplegament d'una aplicació web amb DSC

Desplegament d'una aplicació web amb PowerShell DSC

En aquesta fase hem aprofitat la potència de PowerShell Desired State Configuration (DSC) per automatitzar el desplegament d'una aplicació web estàtica composta per HTML i CSS. Aquesta pràctica simula un entorn real on, per exemple, una empresa pot desplegar el portal corporatiu o una aplicació interna des de fitxers generats pel departament de disseny o màrqueting.

Hem preparat localment els fitxers web (un index.html i una carpeta css/style.css), i hem dissenyat una configuració DSC que s'encarrega de copiar automàticament aquests fitxers cap al directori de publicació d'IIS (C:\inetpub\wwwroot). Això assegura que sempre disposem d'una versió concreta del lloc web, evitant errors manuals i facilitant el control de canvis.

Aquesta metodologia és altament aplicable a entorns empresarials on es requereix coherència i rapidesa en el desplegament web, especialment quan hi ha múltiples servidors que han de mostrar la mateixa informació.

index.html:

```
Set-Content -Path "C:\DSC\web\index.html" -Value @"
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Web desplegada amb DSC</title>
```

```
  <link rel="stylesheet" href="css/style.css">
```

```
</head>
```

```
<body>
```

```
  <h1>Benvingut a la web de l'empresa!</h1>
```

```
  <p>Aquesta pàgina ha estat desplegada automàticament amb PowerShell DSC.</p>
```

```
</body>
```

```
</html>
```

```
"@
```

style.css:

```
Set-Content -Path "C:\DSC\web\css\style.css" -Value @"
```

```
body {
```

```
  background-color: #f2f2f2;
```

```
  font-family: Arial, sans-serif;
```

```
text-align: center;
}
h1 {
color: #007acc;
}
"@
```

Script DSC per desplegar els fitxers web

Desplegament-Web.ps1:

```
Configuration WebCompleta {
    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node localhost {
        File Index {
            SourcePath = "C:\DSC\web\index.html"
            DestinationPath = "C:\inetpub\wwwroot\index.html"
            Ensure = "Present"
            Type = "File"
        }

        File Estils {
            SourcePath = "C:\DSC\web\css"
            DestinationPath = "C:\inetpub\wwwroot\css"
            Ensure = "Present"
            Recurse = $true
            Type = "Directory"
        }
    }
}
```

Definició i aplicació de polítiques de seguretat amb PowerShell DSC

Definició i aplicació de polítiques de seguretat amb PowerShell DSC

En aquesta darrera fase del projecte hem implementat mesures de seguretat bàsiques a nivell de sistema operatiu, utilitzant PowerShell DSC per assegurar que les configuracions es mantinguin al llarg del temps i es puguin aplicar de manera coherent a diversos servidors.

Aquesta fase té un enfocament clarament orientat a l'àmbit empresarial, ja que permet definir i garantir el compliment de polítiques de seguretat com ara la creació d'usuaris administradors, la gestió d'accés remot i la configuració de serveis clau. Aquest tipus d'automatització és clau per mantenir un entorn segur, estandarditzat i fàcilment escalable.

Objectius principals:

- Crear un usuari administrador (adminpaumiro) automàticament.
- Assegurar que aquest usuari és membre del grup Administrators.
- Desactivar el servei d'Escriptori Remot (RDP) per evitar accés no autoritzat.
- Activar i mantenir en execució el servei WinRM per suportar gestió remota amb PowerShell o Ansible.

```
Configuration PoliticaSeguretat {
```

```
    Import-DscResource -ModuleName PSDesiredStateConfiguration
```

```
    Node localhost {
```

```
        # Crear usuari sense definir contrasenya (es farà manualment després)
```

```
        User UsuariDSC {
```

```
            UserName = "adminpaumiro"
```

```
            Ensure = "Present"
```

```
            FullName = "Administrador Pau DSC"
```

```
            Description = "Usuari creat automàticament per DSC"
```

```
            PasswordNeverExpires = $true
```

```
        }
```

```
        Group AfegirAdmin {
```

```
            GroupName = "Administrators"
```

```
            MembersToInclude = "adminpaumiro"
```

```
            DependsOn = "[User]UsuariDSC"
```

```
}
```

```
Service RDP {
```

```
    Name = "TermService"
```

```
    StartupType = "Disabled"
```

```
    State = "Stopped"
```

```
}
```

```
Service WinRM {
```

```
    Name = "WinRM"
```

```
    StartupType = "Automatic"
```

```
    State = "Running"
```

```
}
```

```
}
```

```
}
```