Gestión infraestructuras procesamiento datos

Resum dels tres documents sobre la gestió d'infraestructures per al processament de dades

1. Gestió de l'emmagatzematge (T3.1)

El document analitza les tecnologies i arquitectures principals d'emmagatzematge de dades: **DAS**, **NAS** i **SAN**, destacant-ne les diferències i avantatges. S'aprofundeix en tecnologies com **RAID**, que millora el rendiment i la fiabilitat distribuint dades entre diversos discos. També es detallen sistemes d'arxius distribuïts com **GlusterFS**, així com protocols com **SCSI** i les seves evolucions modernes, incloent **iSCSI** i **NVMe**. Finalment, inclou passos pràctics per implementar RAID a Linux amb `mdadm`.

---

2. Gestió de xarxes (T4.1.1)

Aquest document tracta la importància de les xarxes en les infraestructures de processament de dades. Es presenten conceptes com el **rendiment (throughput)** i l'**amplada de banda (bandwidth)**, explicant com afecten la velocitat i l'eficiència de les aplicacions. També introdueix les **SDN (Software-Defined Networking)**, un paradigma que separa el control de la xarxa del maquinari, proporcionant flexibilitat i programabilitat. Finalment, es llisten eines per mesurar i optimitzar el rendiment de la xarxa, com **iperf**, **Netdata** i **Nagios**, i s'analitzen algorismes de routing per maximitzar l'eficiència.

---

3. Infraestructura com a Codi (IaC) (T5.1.2)

Aquest document introdueix el concepte d'**Infraestructura com a Codi (IaC)**, que automatitza la creació i gestió d'infraestructures mitjançant codi. Es destaquen dos enfocaments: declaratiu i imperatiu, i s'exposen eines com **Terraform**, **Ansible** i **Docker Compose**. Es ressalten avantatges com la rapidesa, la consistència i la reutilització dels recursos. També es detallen exemples pràctics de configuració i desplegament automatitzat, posant èmfasi en com aquestes tècniques milloren l'agilitat i la documentació en la gestió d'infraestructures.

Si vols, puc aprofundir en algun dels temes o apartats!

Esquema del document T3.1: Gestió de l'emmagatzematge

1. Introducció a l'emmagatzematge

- **Importància de l'emmagatzematge:**
  - Factor crític per ser un coll d'ampolla i la principal causa de fallades (55%).
  - Tipus de tecnologies: **magnètic**, **òptic** i **estat sòlid (SSD)**.

---

2. Tecnologies d'emmagatzematge

- **DAS (Direct Attached Storage):**
  Emmagatzematge connectat directament a l'ordinador amb peticions a nivell de blocs.
- **NAS (Network Attached Storage):**
  - Emmagatzematge accessible a través de la xarxa.
  - Peticions a nivell d'arxius (**file I/O**) en lloc de blocs.
  - Pros: fàcil instal·lació i ús compartit.
  - Contres: consumeix ample de banda de la xarxa.
- **SAN (Storage Area Network):**
  - Xarxa d'emmagatzematge especialitzada amb peticions a nivell de blocs.
  - Ofereix accés ràpid i fiable.

---

3. RAID (Redundant Array of Independent Disks)

- **Objectiu:** Millorar el rendiment i la fiabilitat.
  - Distribueix dades entre discos i afegeix redundància.
- **Tipus principals de RAID:**
  - **RAID 0:** Divisió de dades sense paritat (més velocitat, menys fiabilitat).
  - **RAID 1:** Mirall de dades (redundància total).
  - **RAID 5 i 6:** Paritat distribuïda per recuperar dades en cas de fallada.
  - **RAID 0+1 i 1+0:** Combinació de velocitat i redundància.
- **Implementació a Linux:**
  Comandes bàsiques amb `` `mdadm` `` per crear i gestionar arrays RAID.

---

4. Sistemes de fitxers distribuïts (DFS)

- **GlusterFS:**
  - Sistema d'emmagatzematge distribuït i escalable.
  - Agrupa múltiples servidors en un sistema únic.
  - Suporta rèpliques i geo-replicació per garantir la disponibilitat de dades.

  Comandes bàsiques:
  - Crear volums: `` `gluster volume create` ``.
  - Muntar volums: `` `mount -t glusterfs` ``.

---

5. Protocols i estàndards de connexió

- **SCSI (Small Computer System Interface):**
  - Protocol d'interfície per a dispositius d'emmagatzematge.
  - Evolucions modernes:
    - **iSCSI:** Implementació de SCSI sobre TCP/IP, una alternativa econòmica a SAN.
    - **SAS (Serial Attached SCSI):** Connexió més ràpida i fiable.
    - **UAS (USB Attached SCSI):** Per a dispositius USB amb altes velocitats.

---

6. Resum de tendències modernes

- **NVMe (Non-Volatile Memory Express):**

  - Especificació moderna per a SSDs connectats a través de PCIe.
  - Ofereix velocitats altes i baixa latència.
- **Avantatges del modern emmagatzematge:**
  - Millora de la compatibilitat amb sistemes antics i nous.
  - Integració amb infraestructures al núvol i entorns distribuïts.

Si necessites detalls d'algun apartat o comandes específiques, m'ho pots indicar!

Esquema del document T4.1: Gestió de xarxes

1. Introducció a les xarxes

- **Paper de les xarxes en infraestructures de dades:**
  - Determinen el rendiment de les aplicacions.
  - Paràmetres clau: **Rendiment (throughput)** i **Amplada de banda (bandwidth)**.

---

2. Rendiment i amplada de banda

- **Rendiment (throughput):**
  - Quantitat de dades transferides amb èxit en un període de temps.
  - Factors que el perjudiquen:
    - **Pèrdua de paquets.**
    - **Latència:** Temps que triga un paquet a arribar al destí.
    - **Jitter:** Variabilitat en la latència.
- **Amplada de banda:**
  - Capacitat màxima de transferència de dades per segon.
  - Relació amb el rendiment:
    - Més ample de banda permet més dades alhora, però no sempre millora la velocitat.
    - Ex.: Comparació amb una autopista (més carrils = més capacitat, no més velocitat).

---

3. Optimització del rendiment

- **Mesures per millorar la xarxa:**
  - Actualitzar maquinari (routers, switches).
  - Reduir nodes i prioritzar connexions cablejades.
  - Utilitzar **QoS (Quality of Service)** per prioritzar trànsit crític.
  - Redirigir trànsit no essencial a xarxes secundàries.
  - Analitzar i reconfigurar tallafocs.

---

4. Software-Defined Networking (SDN)

- **Definició:**
  Paradigma que separa el control de la xarxa del maquinari subjacent, permetent una gestió centralitzada.
- **Components:**

    - **Plànol de control:** Decideix com es comporta la xarxa.
    - **Plànol de dades:** Reenvia paquets segons les decisions del plànol de control.
- **Beneficis:**
    - Major flexibilitat i programabilitat.
    - Control centralitzat amb visió global de la xarxa.
    - Reducció de la complexitat en entorns moderns.
- **Comparativa SDN vs. Xarxes tradicionals:**
    - **SDN:** Control centralitzat, més eficient i innovador.
    - **Tradicionals:** Control distribuït, més complex i propens a errors.

---

5. Eines i monitorització

- **Mesurament de xarxes:**
    - **Iperf:** Mesura de rendiment entre client i servidor.
    - **Netdata:** Visualització del rendiment local.
    - **Ntopng:** Anàlisi de trànsit en temps real.
    - **Ping/hping3:** Estimació de temps de resposta i latència.
- **Plataformes de monitorització:**
    - **Nagios, Zabbix, Cacti, Munin:** Supervisen xarxes, dispositius i serveis.

---

6. Algorismes de routing

- **Problema:**
    - Trobar el camí òptim per enviar un paquet entre nodes.
- **Algorismes destacats:**
    - **Dijkstra:** Càlcul de rutes mínimes.
    - **Distance-Vector i BGP:** Solucions per a grans xarxes.

---

7. Conclusió

- **Les xarxes modernes:**
    - Són vitals per a la gestió d'infraestructures.
    - SDN ofereix solucions innovadores per a entorns complexos.
    - Optimitzar el rendiment de xarxa és essencial per garantir velocitat i eficiència.

Pots demanar més detalls d'alguna secció o exemples concrets si ho necessites!

Esquema del document T5.1.2: Infraestructura com a Codi (IaC)

1. Introducció a IaC

- **Definició:**
  La Infraestructura com a Codi (IaC) gestiona i aprovisiona infraestructures informàtiques mitjançant fitxers de codi llegibles per màquina.

  - Elimina la configuració manual.
  - Permet desplegaments ràpids i consistents.
- **Motivació:**
  - Abans, la configuració manual provocava inconsistències i dificultats col·laboratives.
  - Les tecnologies modernes (virtualització, DevOps) han accelerat el ritme de desplegaments.
- **Beneficis:**
  - **Velocitat:** Automatització de configuracions.
  - **Consistència:** Evita errors manuals.
  - **Traçabilitat:** Control de versions amb eines com Git.
  - **Reutilització:** Mòduls que permeten replicar entorns fàcilment.

---

2. Enfocaments i mètodes de IaC

- **Enfocaments principals:**
  - **Declaratiu:** Especifica l'estat desitjat de la infraestructura. Exemple: `**kubectl apply -f config.yaml**`.
  - **Imperatiu:** Detalla els passos necessaris per aconseguir l'estat desitjat. Exemple: `**kubectl run nginx --image=nginx**`.
- **Mètodes de desplegament:**
  - **Push:** El servidor de control envia configuracions als sistemes destí.
  - **Pull:** Els sistemes destí extreuen configuracions del servidor de control.

---

3. Eines principals d'IaC

- **Tipologies:**
  - **Scripting:** Scripts ad-hoc per a tasques simples (bash, Python).
  - **Gestió de configuracions:** Instal·lació i configuració de servidors (Ansible, Puppet, Chef).
  - **Aprovisionament:** Creació d'infraestructures (Terraform, AWS CloudFormation).
  - **Contenidors i plantilles:** Entorns preconfigurats (Docker, Vagrant).
- **Exemples destacats:**
  - **Ansible:** Configuració basada en SSH; fàcil d'utilitzar sense necessitat d'agents.
  - **Terraform:** Aprovisionament d'infraestructures complexes mitjançant fitxers reutilitzables.
  - **Docker Compose:** Orquestració d'aplicacions multi-contenidor amb YAML.
  - **Kubernetes:** Gestió d'escala i alta disponibilitat en entorns de contenidors.

---

4. Implementació de IaC

- **Fluxe de treball típic:**
  1. **Especificació:** Configuracions escrites en un llenguatge específic.
  2. **Gestió:** Les configuracions s'emmagatzemen en un repositori o API.
  3. **Execució:** Una plataforma desplega i configura els recursos automàticament.
- **Exemple amb Ansible:**

```yaml


  - hosts: all
    tasks:
```

```
    - name: Instal·lar Ruby
      apt:
        name: ruby-full
        state: present
```

---

5. Avantatges de IaC

- **Velocitat:** Redueix el temps de desplegament.
- **Consistència:** Garanteix que tots els entorns segueixen el mateix model.
- **Documentació viva:** El codi reflecteix l'estat real de la infraestructura.
- **Traçabilitat:** Permet auditar canvis i millorar la col·laboració.
- **Reutilització:** Facilita replicar entorns amb mòduls idempotents.

---

6. Casos d'ús i exemples

- **Terraform:**
  Configura infraestructures al núvol amb fitxers declaratius. Exemple:

```hcl
resource "aws_instance" "example" {
  ami           = "ami-123456"
  instance_type = "t2.micro"
}
```

- **Docker Compose:**
  Defineix aplicacions multi-contenidor amb YAML:

```yaml
version: "3"
services:
  web:
    image: nginx
    ports:
      - "8080:80"
  db:
    image: mysql
    environment:
      MYSQL_ROOT_PASSWORD: example
```

- **Ansible:**
  Permet gestionar servidors amb playbooks reutilitzables i basats en YAML.

---

7. Conclusió

- **IaC és essencial per a les infraestructures modernes:**
  - Garanteix escalabilitat, agilitat i eficiència.
  - Eines com Ansible, Terraform i Docker permeten transformar la gestió d'infraestructures.

Si vols detalls addicionals sobre un apartat o eina, m'ho pots demanar!