Emmagatzematge

Diferents tecnologies de xarxa:

- DAS (Direct Attached Storage): connectats per cable, peticions en blocs/sectors.
- NAS (Network Attached Storage): xarxa compartida amb altres trànsits, utilitza file I / O en lloc de block I / O (NAS SO el tradueix a blocs).
- NAS Gateways: NAS però sense disc integrat.
- SAN (Storage Area Networks): Emmagatzematge resideix en xarxa dedicada.

Diferents tecnologies d'emmagatzematge:

- Magnètic: codificar dades en polaritat magnètica positiva/negativa. IDE, ATA, SCSI, SATA (600MB/s), SAS.
- Òptic: làser per llegir o escriure dades. CD, DVD, BD.
- Estat sòlid: Memòria no volàtil que reté les dades quan s'apaga l'alimentació. Més car pero lleuger, ràpid, silenciós i eficient. Flash USB, targetes memòria Flash, Memory Stick i unitats d'estat sòlid (SSD).

Interfícies SSD:

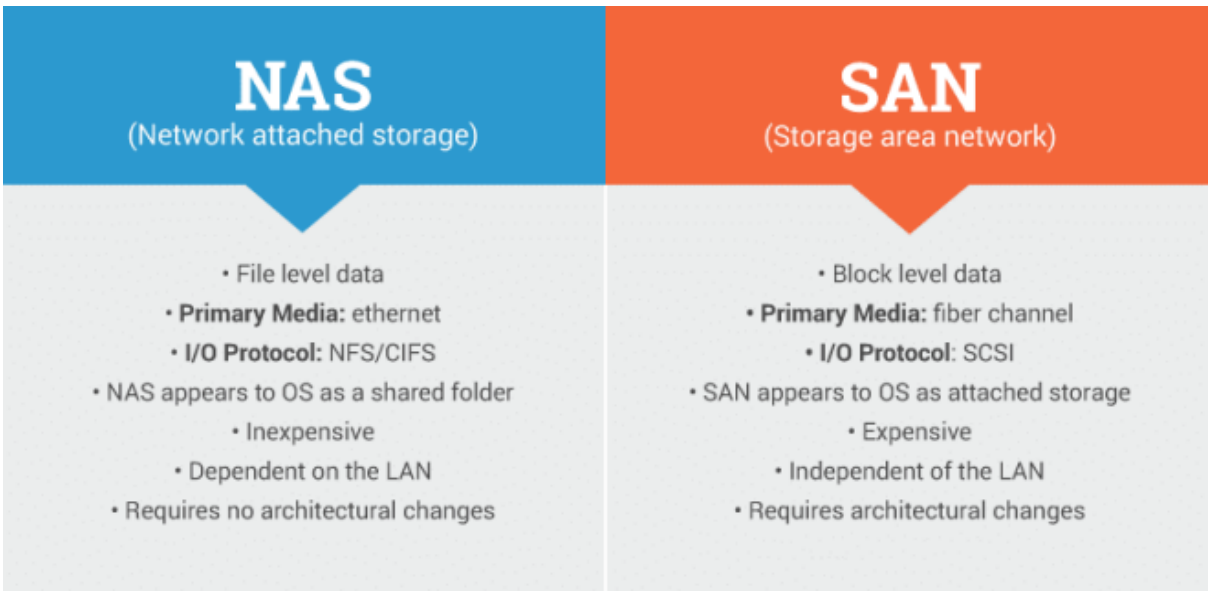
- SAS-3
- SATA 3.0
- PCI Express 3
- M.2
- U.2
- Fibre channel
- USB

NAS

Accés compartit a fitxers des d'un servidor optimitzat.
Inconvenients: ús compartit de la xarxa de producció.

SAN

Entorn emmagatzematge amb xarxa especialitzada (generalment fibra). Accés a nivell de bloc (dispositiu emmagatzematge apareix localment connectat al SO).
Inconvenients: només operacions a nivell de blocs



SCSI (Small Computer System Interface)
Estàndards de connectar i transferir dades
Tipus de connexió:

- Interfície paral·lela SCSI.
- Canal de fibra.
- Serial attached SCSI.
- iSCSI (Internet Small Computer System Interface).

En lloc de SCSI, els ordinadors i ordinadors portatils moderns solen utilitzar interfícies SATA per a les unitats de disc dur interns, amb NVMe sobre PCIe guanyant popularitat, ja que SATA pot ser un coll d'ampolla les modernes unitats d'estat sòlid.

RAID (Redundant Array of Independent Disks)
Lectura/escritura en conjunt de discos al mateix temps.
Tipus:

- Maquinari amb controlador específic.
- Programari de ost.

Tipus 2:

- (estàndard) 0
- (Data Striping), 1,2,3,4,5,6, 5e i 6e
- (nested) 0 + 1, 1 + 0, 30, 100, 50

Ordres Linux RAID:

- **Instal·lar:** apt-get install mdadm
- **Preparar els discos:** fdisk con una partició tipus 0xfd-, **i després** mkfs.ext4 /dev/sdi1
- **Per crear un raid 5 anomenat /dev/md0 con 3 dispositius i chunk size of 16384: (típicament un gran chunk és millor per gran arxius grans, default es 512)** mdadm --create --level=5 [--chunk=16384] --raid-devices=3 /dev/md0 /dev/sdb1 /dev/sdc1 /dev/sdd1 **I crear el filesystem abans de muntar-ho.**
- **Per muntar un array que no està a l'arxiu de configuració:** mdadm --assemble /dev/md0 /dev/sdb1 /dev/sdc1 /dev/sdd1
- **Per crear l'arxiu de configuració per a que arranqui al iniciar l'ordinador:** mdadm --detail --scan >> /etc/mdadm/mdadm.conf (Crear el filesystem en /dev/md0 antes de montarlo)
- **Per afegir un nou disc i fer créixer l'array (després s'haurà de fer créixer el filesystem amb resize2fs /dev/md0)**: mdadm --add /dev/md0 /dev/sde1 (aquest disc quedarà com spare) mdadm --grow /dev/md0 --raid-devices=4 (amb això veurem que passa d'spare a actiu)
- **Marcar un disc amb errors i treure'l de l'array (també és necessari quan volem canviar un disc)** mdadm --fail /dev/md0 /dev/sde1 mdadm --remove /dev/md0 /dev/sde1
- **Aturar l'array : (haurem de assegurar que primer el filesystem estigui desmuntat)** mdadm --stop /dev/md0
- **Iniciar l'array (després es podrà muntar):** mdadm --run /dev/md0
- **Obtenir info de l'array:** mdadm --detail /dev/md0
- **Monitoritzar l'arrays: (i obtenir mails de quan té errors)** mdadm --monitor --scan --mail=[email address] --delay=1800 &

DFS: Gluster FS
Sistema fitxers d'emmagatzematge distribuït. Agrupa servidors emmagatzematge amb connexions Ethernet o Infiniband. Funciona com client i servidor.

LVM (Logical Volume Manager)
framework que proporciona una gestió de volum lògica per al nucli Linux separan la gestió física del disc de la lógica.

- Creació de volums lògics individuals.
- Gestionar granges de disc durs permetent afegir i substituir discos sense temps d'inactivitat ni interrupció del servei.
- Fer còpies de seguretat consistents.
- Xifrar diverses particions físiques amb una sola contrasenya.

Inicialització en 4 passos:

- Definir i inicialitzar els discos creant una partició en cada disc tipus 8e (amb el fdisk)
- **Definir i inicialitzar cada Physical Volumes (PV):** pvcreate /dev/device [/dev/device]
- **Definir el Volume Groups agrupant els PVs (VG):** vgcreate vg-name /dev/device [dev/device]
- **Inicialitzar els Logical Volumes sobre cada VG (LV):** lvcreate –L size –n lv-name vg-name

Ceph
Ceph és una plataforma software open-source de emmagatzemament que implementa object storage sobre clúster i proveeix interfícies 3 in 1 per: object, block, file-level storage. Replica les dades i les fa tolerants a fallades. Sistema auto-reparable i auto-gestionable.
Nivell de fitxer: habitual amb NAS.
Nivell de bloc: utilitzats a les SAN. Un bloc és un volum d'emmagatzematge en brut ple de fitxers dividits en trossos de dades d'igual mida.
Nivell objecte: emmagatzema dades en contenidors aïllats coneguts com a objectes.

Samba
Suite de codi obert, que proporciona servei d'arxius i d'impressió a clients SMB/CIFS i pot compartir un sistema d'arxius de Linux amb Windows i viceversa i també impressores connectades a Linux o un sistema amb Windows.

Xarxes

Throughput (rendiment)
Es mesura en bits per segon
Quantitat de dades que es poden transferir de la font a la destinació en un període de temps determinat. Mesura quants paquets arriben a les seves destinacions amb èxit.
Rendiment baix indica problemes com la pèrdua de paquets.

Velocitat de producció lenta produïda per:

- Pèrdua paquets.
- Latència: temps que triga el paquet en arribar a la destinació. Minimitzar-la és el més important.
- Fluctuació: variacions en la latència.

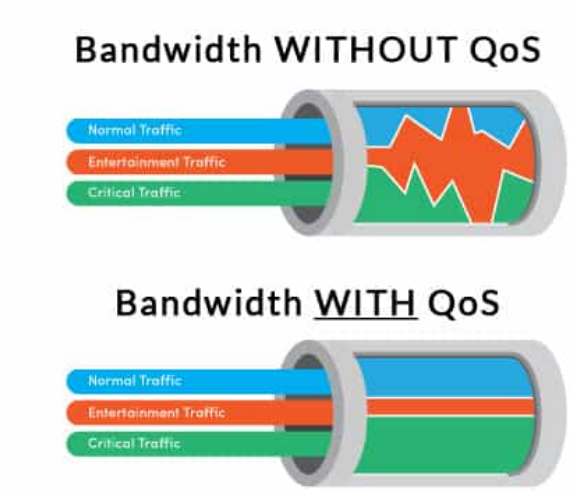
Problema principal que afecta al throughput: Massa aplicacions utilitzant la xarxa.
Solucions:

- Actualitzar routers, reduir nodes xarxa.
- Bonding (més sortides del node).
- Wired connections (Fibra òptica és la millor opció).
- Analitzar apps i discriminar trànsit.
- Analitzar tallafocs i la seva utilitat per busca reduir el seu ús.
- Analitzar maquinari xarxa en busca de defectes.
- Analitzar rendiment real amb SLA i redissenyar.

Bandwidth (amplada de banda)
Es defineix com la capacitat màxima de transferència d'una xarxa. És una mesura de la quantitat de dades que es poden enviar i rebre alhora. L'amplada de banda es mesura en bits, megabits o gigabits per segon.
Important: l'amplada de banda no canvia que tan ràpid es mouen els paquets.

Optimitzar amb:

- Configuració QoS.
- Externalitzar part del trànsit a xarxes de núvol públiques.
- Analitzar si estan ben planificades les operacions no essencials.



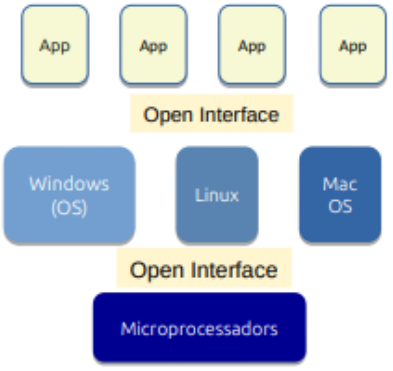
Bandwidth vs Throughput
Es pot considerar l'amplada de banda com un tub i el rendiment de dades com l'aigua que passa per el tub. Si teniu un tub gran, podeu recollir aigua a un ritme més ràpid. Per contra, si el tub és petit, la quantitat d'aigua que recollireu serà menor i el ritme serà més lent. El rendiment és sovint un indicador més important del rendiment de la xarxa que l'amplada de banda perquè us indicarà si la vostra xarxa és literalment lenta o simplement hipotèticament lenta.

Throughput vs Connection Speed
La velocitat de connexió és el suposat màxim de velocitat dels paquets, mentre que el Throughput (rendiment) és la velocitat real després de tenir en compte els problemes com la pèrdua de paquets.

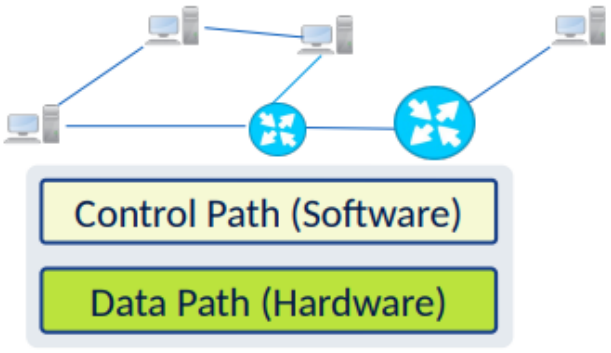
Tools
- Iperf: eina simple que permet mesurar el rendiment de la xarxa. Requereix un client i un servidor.
- Netdata: mostra el rendiment de la màquina, no de la xarxa.
- Ntopng: eina de mesura de trànsit de xarxa que controla l'ús de la xarxa. Proporciona una interfície d'usuari web encriptada i intuïtiva per explorar informació de trànsit en temps real i històrica

Software Defined Networking (SDN)
- Representen una evolució significativa en la manera com es gestionen i operen les xarxes modernes.
- És un enfocament de xarxa que permet la gestió per software i centralitzada del comportament de la xarxa mitjançant interfícies obertes i abstraccions ben definides.
- Les SDN són una tecnologia disruptiva i essencial a l'era de les xarxes modernes.
- Desafia les arquitectures de xarxa tradicionals, oferint més flexibilitat i control.
- La importància de SDN rau en la seva capacitat per desacoplar el pla de control del pla de dades, cosa que permet una gestió centralitzada i una configuració de xarxa dinàmica.
- Les SDN poden transformar les operacions de xarxa, millorar l'eficiència i oferir noves oportunitats per a la innovació.

Arquitectura:
- Plànol de Dades: dispositius de xarxa (switches, routers). Encaminament i reenviament de paquets.
- Plànol de Control: controladors SDN (cervells xarxa). Visió completa de la xarxa i programent dinàmicament els dispositius.
- Plànol d'Aplicació: capa superior, aplicacions que interactuen amb el controlador SDN. Optimitzar trànsit, seguretat i gestió polítiques.



Estructura (Separació entre el control plane i el data plane):
- Plànol de control d'una xarxa: funcions que controlen comportament xarxa.
- Plànol de dades d'una xarxa: funcions que realment reenvien o deixen paquets.
- Plànols integrat verticalment en xarxes tradicionals.
- Visió de SDN/OpenFlow.



Funcionament (Les aplicacions de xarxa especifiquen les funcions de xarxa):

- Control d'accés.
- Aïllament.
- Encaminament.
- Funcionalitat.

Beneficis:

- Flexibilitat i agilitat.
- Gestió centralitzada.
- Reducció de costos.
- Millora la seguretat de la xarxa.

Casos d'ús destacats

- Centres de dades: gestió dinàmica i eficient del trànsit.
- Xarxes empresarials: facilita polítiques de seguretat i qualitat de servei centralitzat.
- Àmbit telecomunicacions: virtualització funcions de xarxa (NFV).
- Internet de les Coses (IoT).
- Xarxes de campus i àrees metropolitanes: necessitat d'escalabilitat i gestió eficient del tràfic.

SDN vs Convencional

SDN	Conventional
El controlador pot no estar en el mateix dispositiu que el hardware de xarxa	Hardware de xarxa y su control estan en el mateix dispositiu físic
Algoritme de routing centralitzats amb una visió global	Algoritme de routing distribuït
Funciones de xarxa realitzades con una vista global	Les funcions de xarxa hauran de ser realitzades de forma distribuïda I seran propenses a errors
S'haurà de fer una nova 'abstracció' de la xarxa amb aquesta visió centralitzada	L'abstracció de la xarxa està integrada als algoritmes distribuïts

OpenFlow

Protocol de comunicacions que permet als controladors de la xarxa determinar la ruta dels paquets a través d'una xarxa de switches/routers. Gestiona millor trànsit sofisticat en comparació a les llistes de control d'accés (ACL).
Permet l'administració remota de taules de reenviament de paquets d'un switch/router de capa 3, afegint, modificant i eliminant regles i accions de concordança de paquets.
Treballa a sobre del protocol de control de transmissió (TCP) i recomana l'ús de Transport Layer Security (TLS). Els controladors fan servir el port TCP 6653 on arribaran les peticions dels dispositius de xarxa que vulguin configurar una connexió.

Open vSwitch
Implementació de codi obert d'un switch multicapa virtual distribuït amb amb l'objectiu de proveir un dispositiu de comunicació per a entorns de virtualització de maquinari amb múltiples protocols i estàndards utilitzats a xarxes.

Bridging Linux
Si no és necessari la funcionalitat (i complexitat) d'Open V Switch sobre Linux es pot fer servir un Bridge Virtual a través del paquet bridge-utils ens permet crear Bridges Virtuals.

Infrastructure as Code

La Infraestructura com a codi (IaC) és una combinació d'estàndards, pràctiques, eines i processos per subministrar, configurar i gestionar la infraestructura informàtica mitjançant codi i altres fitxers llegibles per una màquina.
És una manera d'elevar l'estàndard de la gestió de la infraestructura i el temps de desplegament. Mitjançant l'ús d'una combinació d'eines, llenguatges, protocols i processos, IaC pot crear i configurar elements d'infraestructura de manera segura en qüestió de segons.

En resum: IaC és el procés de gestió i subministrament dels recursos del centre de dades informàtics mitjançant fitxers de definició llegibles per màquina, en lloc de la configuració del maquinari físic o les eines de configuració interactives.

Mètodes:
- Push: servidor extreu configuració del servidor de control.
- Pull: servidor de control envia configuració a servidor.

Avantatges:
- Velocitat: evitar interacció manual.
- Control font.
- Documentació.
- Coherència: una sola estcutura per tot el projecte.
- Agilitat.
- Reutilització.
- Generalització.

Funcionament:
1. Desenvolupadors escriuen les especificacions d'infraestructura.
2. Fitxers s'envien a servidor màster, API de gestió o dipòsit de codi.
3. La plataforma fa tots els passos necessaris per crear i configurar els recursos informàtics.

Exemple (playbook):

```
- name: Configure Raddit App Instance
    hosts: all
    become: true
    tasks:
        - name: Install Ruby
            apt: "name={{ item }} state=present"
            with_items:
                - ruby-full
                - build-essential
```

Tipus:

- Scripting.
- Eines de gestió de configuracions: gestionar programari (instal·lar i configurar servidors).
- Eines de subministrament: creació d'infraestructures.
- Contenidors i eines de plantilles. generació de plantilles o imatges precarregades amb les biblioteques i components necessaris per executar una aplicació.

Comparació Eines

Eina	Millor ús	Tipus de configuració	Proveïdor Cloud	Dificultat
Terraform	Multi-cloud	Declarativa (HCL)	Multi-cloud	Mitjana
AWS CloudFormation	AWS	Declarativa (YAML/JSON)	AWS exclusiu	Baixa
Ansible	Configuració de servidors	Procedimental (YAML)	Multi-cloud/on-prem	Baixa
Kubernetes	Orquestració de contenidors	Declarativa (YAML)	Multi-cloud/on-prem	Alta
Pulumi	Desplegament modern	Llenguatges de programació	Multi-cloud	Mitjana
Docker Compose	Entorns multi-contenidor	Declarativa (YAML)	Local/on-prem	Baixa

Característica	Vagrant	Terraform	Ansible	Docker Compose
Objectiu principal	Entorns locals de desenvolupament	Multi-cloud i infraestructures	Configuració de sistemes	Contenidors locals
Provisionament	Manual o integrat amb Ansible	Declaratiu	Procedimental	Declaratiu
Multiplataforma	VirtualBox, VMware, Hyper-V	AWS, Azure, GCP	Multi-cloud i on-prem	Local
Facilitat d'ús	Alta (fitxers senzills)	Mitjana	Baixa	Alta
Escalabilitat	Limitada a màquines locals	Alta (infraestructura global)	Alta	Baixa
Usos destacats	Simulació de servidors locals	Desplegaments en producció	Gestió de configuracions	Desenvolupament amb contenidors

- AWS CloudFormation: : Eina específica d'AWS per gestionar recursos mitjançant plantilles JSON o YAML.
- Docker Compose: Eina per definir i executar aplicacions multi-contenidor amb fitxers YAML.
- Kubernetes: Plataforma d'orquestració per gestionar contenidors en producció, com Docker. Gestiona la disponibilitat, escalabilitat i desplegaments d'aplicacions.
- Pulumi: Alternativa moderna a Terraform, que permet utilitzar llenguatges de programació com Python, TypeScript o Go. Ideal per equips de desenvolupament amb experiència en aquests llenguatges.

Ansible (Red Hat actualment)
Plataforma codi obert per configurar i administrar molt fàcilment sistemes IT, i desplegar aplicacions i programari en els elements de infraestructura. I tot això utilitzant només SSH. Permet el software provisioning, gestió de configuracions i desplegament d'aplicacions en el que es coneix infraestructura com a codi. Inclou el seu propi llenguatge declaratiu per descriure la configuració del sistema. Té una arquitectura que distingeix entre controlador i nodes.

Definicions:
- Node de control: Màquina amb Ansible instal·lat. Executa ordres i playbooks. Es pot tenir més d'un.
- Nodes gestionats: Dispositius de la xarxa gestionats (també dits amfitrions)
- Inventari: Llista de nodes gestionats (fitxer d'inventari es sol anomenar "fitxer host".
- Col·leccions: Format de distribució d'Ansible que inclou playbooks, rols, mòduls i connectors.
- Mòduls: Unitats de codi executades per Ansible.
- Tasques: Unitats d'acció d'Ansible.
- Playbooks: Llistes ordenades de tasques desades per executar en ordre determinat repetidament.

Vagrant
Proporciona entorns fàcils de configurar, reproduïbles i portàtils amb un únic flux de treball que ajudarà a maximitzar la productivitat i la flexibilitat en el desenvolupament d'aplicacions/serveis.
Pot proveir de màquines de diferents proveïdors (VirtualBox, Vmware, AWS, o altres) utilitzant scripts, Chef o Puppet per a instal·lar i configurar automàticament el programari de la VM.
En l'àmbit IT permet tenir entorns d'un sol ús amb un flux de treball coherent per a desenvolupar i provar scripts d'administració de la infraestructura, ja que ràpidament es poden fer proves en un entorn de virtualització local, com VirtualBox o VMware.
RECOMANO LLEGIR SECCIÓ DE VAGRANT AL POWER

Terraform

Eina desenvolupada por Hashicorp per IaC amb una filosofia de Write, Plan, Apply. Es una eina open source per desenvolupar IaC a través de una CLI i gestionar centenes de serveis cloud. Terraform codifica les API dels cloud en arxius de configuració declaratius que permeten un descripció fàcil i simple de la infraestructura a desplegar.

Permet construir, canviar i gestionar la infraestructura d'una manera segura i repetible. Els operadors i els equips d'infraestructures poden utilitzar Terraform per gestionar entorns amb un llenguatge de configuració anomenat HashiCorp Configuration Language (HCL) per a desplegaments automatitzats i llegibles

Treballa amb 5 ordres principals:

- init: preparar entorn.
- plan: mostra canvis requerits per la configuració.
- apply: crea o actualitza la infraestructura.
- destroy: elimina la infraestructura.
- validate: verifica sintaxis.

Avantatges:

- En un CPD modern existeixen diversos entorns diferents per donar suport a les diverses aplicacions però això no afecta a Terraform ja que es pot gestionar un entorn heterogeni amb el mateix flux de treball creant un fitxer de configuració.
- Terraform crea un arxiu d'estat al iniciar. Aquest es pot comparar amb arxius d'estat posteriors per observar la creació de nous recursos o modificar els ja existents.
- El flux de treball integrat té com a objectiu infondre confiança als usuaris promovent operacions fàcilment repetibles i una fase de planificació per permetre als usuaris assegurar-se que les accions realitzades no causin interrupcions al seu entorn.

Raons per utilitzar Terraform amb Docker:

- Gestionar fàcilment la infraestructura de Docker com a codi, facilitant la col·laboració i escalada en diversos entorn.
- Configuracions de Docker més reutilitzables i coherents.
- Actualitzar fàcilment la infraestructura de Docker modificant els fitxers de configuració i aplicar automàticament els canvis als contenidors en execució.

Terraform vs Ansible

Terraform	Ansible
Més adequat per aprovisionament d'infraestructura	Automatització d'IT
Excel·lent per a configurar infraestructures cloud	Millor eina per configurar servidors
Permet desplegar Load Balancers, VPC, ...	Permet desplegar apps sobre la infraestructura
Llenguatge Declaratiu	Llenguatge procedural
Definit un estat final, pot desenvolupar tot els passos per arribar	Se necessita indicar cada pas per tenir el resultat final
És considerat ideal per conservar un estat estable	Manté tots els components en execució reparant els errors en lloc de reemplaçar la infraestructura