



“Alta disponibilitat a AWS amb IaC”

Marc Teòric

Assignatura: M17 - Seguretat en sistemes, xarxes i serveis

UF4: Seguretat en Xarxes

Curs: 2024-25

Índex

1. Introducció a la Computació al Núvol.....	3
1.1. Què és la computació al núvol?	3
1.2. Per què és tan utilitzada actualment? (Avantatges clau).....	4
1.3. Diferències entre núvol públic, privat i híbrid	4
1.4. Models de servei: IaaS, PaaS i SaaS.....	4
1.5. Conclusió Apartat 1	5
1.6. Recursos Apartat 1	6
2. AWS com a Plataforma al Núvol	7
2.1. Què és AWS i per què es fa servir?	7
2.2. Eines i serveis essencials d'AWS per a la gestió de la infraestructura	7
2.3. AWS vs. Altres proveïdors: Una breu comparativa	9
2.4. Conclusió Apartat 2	9
2.5. Recursos Apartat 2	10
3. Infraestructura com a Codi (IaC).....	11
3.1. Què és IaC i per què és important?.....	11
3.2. Problemes que resol la IaC (Automatització i escalabilitat)	12
3.4. Instal·lació de Terraform i eines complementaries.	13
3.4.1. Terraform.	13
3.4.2. Tfswitch.....	14
3.4.3. Tfsec.	14
3.4.4. Terraform docs.	15
3.5. Proveïdor de Terraform i selecció de versions.....	15
3.6. Versions de Terraform i selecció de versions.	16
3.7. Desenvolupament de mòduls de Terraform.....	16
3.8. Conclusió Apartat 3	17
3.9. Recursos Apartat 3	18
4. Alta Disponibilitat i Desplegament d'una Web Estàtica amb AWS S3 i CloudFront	19
4.1. Què entenem per alta disponibilitat a AWS?	19
4.2. Amazon S3 com a hosting estàtic	19
4.3. Amazon CloudFront: Optimització i seguretat	20
4.4. AWS Certificate Manager (ACM) i HTTPS	20
4.5. Infraestructura com a Codi (IaC) amb Terraform	20
4.6. Seguretat i permisos en S3 i CloudFront.....	21
4.7. Validació i proves del desplegament Un cop finalitzat el desplegament de la infraestructura, és imprescindible verificar que tot funciona correctament.	21
4.8. Conclusió Apartat 4	21
4.9. Recursos Apartat 4	22

- **Escalabilitat ràpida:** Es poden augmentar o reduir els recursos de forma automàtica segons la demanda.
- **Supervisió i mesurament de l'ús:** Els sistemes de núvol controlen i optimitzen els recursos, oferint transparència als usuaris.

1.2. Per què és tan utilitzada actualment? (Avantatges clau)

Principalment la computació al núvol ha guanyat popularitat perquè facilita l'accés a tecnologia avançada sense els costos elevats d'una infraestructura pròpia. Alguns dels principals avantatges són:

- **Costos reduïts:** No cal comprar maquinari ni pagar pel manteniment físic.
- **Flexibilitat i escalabilitat:** Es poden ampliar o reduir recursos fàcilment segons les necessitats.
- **Accessibilitat global:** Permet accedir a dades i aplicacions des de qualsevol lloc.
- **Automatització i gestió eficient:** Els sistemes poden adaptar-se automàticament a les necessitats dels usuaris.
- **Seguretat millorada:** Els proveïdors de núvol implementen mecanismes de seguretat avançats, tot i que la responsabilitat també recau en l'usuari.

1.3. Diferències entre núvol públic, privat i híbrid

Existeixen diversos models de desplegament de la computació al núvol, que determinen com es gestionen els recursos i qui en té accés:

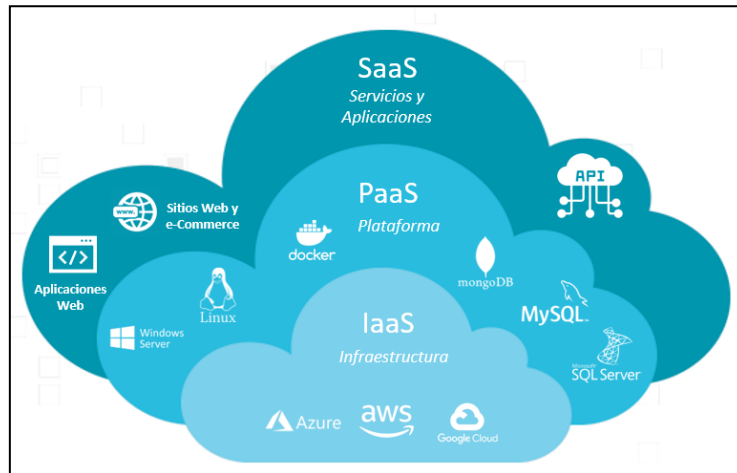
- **Núvol públic:** La infraestructura és gestionada per un proveïdor extern (com AWS, Google Cloud o Microsoft Azure) i és accessible per múltiples organitzacions. És una opció econòmica i flexible.
- **Núvol privat:** L'empresa o la organització gestiona la seva pròpia infraestructura en el núvol, cosa que proporciona més control i seguretat. Es fa servir en sectors amb altes exigències de privacitat, com la banca o la sanitat.
- **Núvol híbrid:** Combina núvols públics i privats, permetent una major flexibilitat. Per exemple, una empresa pot mantenir les seves dades més sensibles en un núvol privat i utilitzar serveis públics per aplicacions menys crítiques.

1.4. Models de servei: IaaS, PaaS i SaaS

Els serveis de computació al núvol es divideixen en tres models principals:

- **Infraestructura com a Servei (IaaS):** Proporciona recursos bàsics com servidors virtuals, emmagatzematge i xarxes. L'usuari té el control sobre el sistema operatiu i les aplicacions. Exemple: Amazon EC2.

- **Plataforma com a Servei (PaaS):** Ofereix eines per al desenvolupament i desplegament d'aplicacions sense haver de gestionar la infraestructura subjacent. Exemple: Google App Engine.
- **Programari com a Servei (SaaS):** Proporciona aplicacions llestes per utilitzar directament des d'Internet. Exemple: Google Drive o Microsoft 365.



Font [Qualitapps](#): IaaS, PaaS i SaaS: Els models de serveis a la núvol.

1.5. Conclusió Apartat 1

La computació al núvol ha transformat la gestió de la infraestructura informàtica, permetent a les empreses i organitzacions accedir a recursos tecnològics sense la necessitat de mantenir infraestructures físiques pròpies. Aquest model ofereix grans avantatges, com la flexibilitat, l'escalabilitat i la reducció de costos, fent-lo especialment atractiu per a empreses de totes les mides.

Els diferents models de desplegament (públic, privat i híbrid) permeten adaptar-se a les necessitats específiques de cada organització, oferint diferents nivells de control i seguretat. Així mateix, els models de servei (IaaS, PaaS i SaaS) proporcionen solucions adaptades tant per a usuaris finals com per a desenvolupadors i administradors de sistemes.

Tot i els seus beneficis, la computació al núvol també introdueix nous reptes en seguretat i privacitat, ja que les dades no es troben en servidors propis, sinó en infraestructures gestionades per tercers. Per això, és essencial comprendre com funcionen aquests serveis i implementar bones pràctiques per garantir la seguretat de la informació.

1.6. Recursos Apartat 1

- Amazon Web Services. (s.f.). *Computación en la nube con AWS*. AWS.
<https://aws.amazon.com/es/>
- Wikipedia. (2024). *Computación en la nube*. Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube
- Qualitapps. (2023). *IaaS, PaaS i SaaS: Els models de serveis al núvol*.
<https://qualitapps.com/ca/que-son-els-models-iaas-paas-i-saas-del-nuvol/>
- Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing*. National Institute of Standards and Technology.
<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
- Buyya, R., Broberg, J., & Goscinski, A. (2010). *Cloud computing: Principles and paradigms*. Wiley. <https://www.amazon.es/Cloud-Computing-Principles-Paradigms-Distributed-ebook/dp/B0052O7CR8>
- IBM Cloud. (s.f.). What is Cloud Computing?
<https://www.ibm.com/cloud/what-is-cloud-computing>
- Google Cloud. (s.f.). What is cloud computing?
<https://cloud.google.com/learn/what-is-cloud-computing>
- Microsoft Azure. (s.f.). Cloud computing basics. Microsoft.
<https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary>

2. AWS com a Plataforma al Núvol

Amazon Web Services (AWS) és una de les plataformes de computació al núvol més utilitzades a nivell mundial. Llançada per Amazon el 2006, ofereix una àmplia gamma de serveis d'infraestructura i aplicacions per a empreses de totes les mides, des de startups fins a grans corporacions i institucions governamentals. AWS permet desplegar, gestionar i escalar aplicacions i serveis de manera flexible i eficient, sense la necessitat de mantenir infraestructura pròpia.

Els serveis d'AWS es distribueixen a través d'una xarxa global de centres de dades situats en diferents regions. Aquesta infraestructura ofereix alts nivells de disponibilitat, escalabilitat i seguretat, permetent a les empreses operar en un entorn fiable i optimitzat segons les seves necessitats.

2.1. Què és AWS i per què es fa servir?

AWS proporciona diverses eines per gestionar servidors, bases de dades, emmagatzematge, xarxes, seguretat i altres serveis sense la necessitat de tenir maquinari físic. L'ús d'AWS permet a les empreses centrar-se en el desenvolupament i l'optimització dels seus serveis, sense preocupar-se per la infraestructura subjacent. AWS es fa servir principalment per:

- **Habilitar infraestructura escalable:** Es poden afegir o eliminar recursos segons la demanda.
- **Reduir costos:** Es paga només pels recursos utilitzats, evitant inversions en hardware.
- **Millorar la seguretat:** Disposa de mecanismes avançats per protegir dades i sistemes.
- **Facilitar la innovació:** Proporciona eines per desplegar ràpidament aplicacions i provar noves tecnologies.
- **Augmentar la disponibilitat:** La seva infraestructura global garanteix redundància i alta disponibilitat.

Gràcies a aquests avantatges, AWS és la plataforma triada per moltes empreses per allotjar els seus serveis en el núvol, sent una peça clau en la transformació digital i l'adopció de models basats en núvol.

2.2. Eines i serveis essencials d'AWS per a la gestió de la infraestructura

Entrant en matèria, a continuació farem un recorregut per els serveis i eines més populars del proveïdor AWS que permeten la gestió eficient de la infraestructura de núvol i l'execució d'aplicacions a escala.

- **Amazon Elastic Compute Cloud (EC2):** Ofereix serveis d'infraestructura per a la creació de màquines virtuals escalables i flexibles. És un dels serveis més utilitzats

d'AWS, permetent als usuaris executar aplicacions en màquines virtuals amb diferents configuracions i sistemes operatius.

- **Amazon Simple Storage Service (S3):** És un servei d'emmagatzematge en línia que permet l'emmagatzematge segur i altament escalable d'arxius i dades. És àmpliament utilitzat per emmagatzemar i gestionar dades, realitzant còpies de seguretat i distribució de continguts.
- **Amazon Relational Database Service (RDS):** RDS proporciona serveis de bases de dades gestionades, incloent MySQL, PostgreSQL, Oracle i altres. Això permet als usuaris gestionar bases de dades sense haver de preocupar-se de la infraestructura subjacent.
- **Elastic Load Balancing (ELB):** Servei que distribueix automàticament el trànsit de xarxa entre múltiples instàncies per garantir alta disponibilitat i escalabilitat. Permet evitar sobrecàrregues en els servidors i millorar el rendiment de les aplicacions. AWS ofereix tres tipus de balancejadors de càrrega: **Application Load Balancer (ALB)** per aplicacions web, **Network Load Balancer (NLB)** per trànsit de xarxa d'alt rendiment, i **Classic Load Balancer (CLB)** per aplicacions més antigues.
- **Amazon Lambda:** AWS Lambda és un servei de computació sense servidors que permet als desenvolupadors executar codi en resposta a esdeveniments sense la necessitat de gestionar servidors. És ideal per a aplicacions amb càrregues de treball variables.
- **Amazon Virtual Private Cloud (VPC):** Permet als usuaris crear una xarxa privada virtual aïllada a AWS. Això ofereix un control granular sobre la xarxa i permet la connexió segura de recursos en núvol amb xarxes on-premises.
- **Amazon Route 53:** Aquest servei ofereix serveis de DNS escalables i fiables, juntament amb equilibri de càrrega i rutes optimitzades per a aplicacions.
- **Amazon CloudWatch:** És una eina de monitoratge i generació de registres que permet als usuaris seguir les mètriques i registres dels seus recursos en AWS. És essencial per garantir un rendiment òptim i detectar problemes de manera proactiva.



Font [Aallcode.com](https://aallcode.com): Top AWS services.

2.3. AWS vs. Altres proveïdors: Una breu comparativa.

AWS no és l'únic proveïdor de computació al núvol. Hi ha altres plataformes destacades com Microsoft Azure i Google Cloud Platform (GCP). A continuació, es fa una comparació entre aquests tres principals proveïdors:

	Amazon Web Services	Microsoft Azure	Google Cloud Platform
Gama de serveis	Més de 200 serveis per a aplicacions, IA, big data, etc.	Integració forta amb productes Microsoft.	Especialitzat en anàlisi de dades i IA.
Ubicacions de Centres de Dades	Present a més regions que qualsevol altre proveïdor.	Amplia presència global.	Menys regions, però creixement ràpid.
Model de preus	Pagament per ús, amb opcions de descomptes per ús prolongat.	Plans flexibles i avantatges per a empreses amb Windows.	Model de pagament per ús altament flexible.
Integració	Ecosistema ampli amb compatibilitat per a diversos sistemes operatius.	Integració completa amb Windows i Active Directory.	Especialitzat en AI i machine learning.
Casos d'ús més comuns	Infraestructura empresarial, hosting, aplicacions escalables.	Empreses que ja utilitzen solucions Microsoft.	Anàlisi de dades, AI i aplicacions en el núvol.

2.4. Conclusió Apartat 2

AWS s'ha consolidat com la plataforma líder en computació al núvol gràcies a la seva fiabilitat, escalabilitat i varietat de serveis. Permet a les empreses desplegar i gestionar la seva infraestructura de manera eficient, evitant la necessitat de mantenir hardware físic.

Els serveis bàsics com EC2, S3, RDS i VPC són fonamentals per gestionar aplicacions i recursos en el núvol, oferint una solució flexible i segura per a qualsevol tipus de projecte.

Tot i que Microsoft Azure i Google Cloud ofereixen alternatives sòlides, AWS és el proveïdor més estès a nivell global, fet que el converteix en una de les millors opcions per a l'administració de sistemes informàtics.

2.5. Recursos Apartat 2

- Allcode. (s.f.). *Top AWS services*.
<https://allcode.com/top-aws-services/>
- Amazon Web Services. (s.f.). *Amazon EC2*.
<https://aws.amazon.com/es/ec2/>
- Amazon Web Services. (s.f.). *Amazon S3*.
<https://aws.amazon.com/es/s3/>
- Amazon Web Services. (s.f.). *Amazon RDS*.
<https://aws.amazon.com/es/rds/>
- Amazon Web Services. (s.f.). *Elastic Load Balancing (ELB)*.
<https://aws.amazon.com/es/elasticloadbalancing/>
- Amazon Web Services. (s.f.). *AWS Lambda*.
<https://aws.amazon.com/es/lambda/>
- Amazon Web Services. (s.f.). *Amazon VPC (Virtual Private Cloud)*.
https://docs.aws.amazon.com/es_es/vpc/latest/userguide/what-is-amazon-vpc.html
- Amazon Web Services. (s.f.). *Amazon Route 53*. <https://aws.amazon.com/es/route53/>
- Amazon Web Services. (s.f.). *Amazon CloudWatch*.
<https://aws.amazon.com/es/cloudwatch/>
- Glowf, J. C. (2023). *Comparación de las principales plataformas en la nube: AWS, Azure y Google Cloud*. LinkedIn. <https://www.linkedin.com/pulse/comparaci%C3%B3n-de-las-principales-plataformas-en-la-nube-juan-carlos-glowf/>

3. Infraestructura com a Codi (IaC)

La Infraestructura com a Codi és una metodologia que permet gestionar i desplegar infraestructures tecnològiques mitjançant codi, en comptes de realitzar configuracions manuals. Aquest enfocament permet automatitzar el provisionament de servidors, xarxes, bases de dades i altres recursos de manera eficient, repetible i escalable.

Aquesta pràctica s'ha tornat essencial en l'administració moderna de sistemes informàtics, especialment en entorns de computació al núvol, on l'escalabilitat i la consistència són crucials. A través de IaC, les organitzacions poden garantir que la seva infraestructura es desplega sempre de la mateixa manera, evitant errors humans i reduint el temps de configuració.

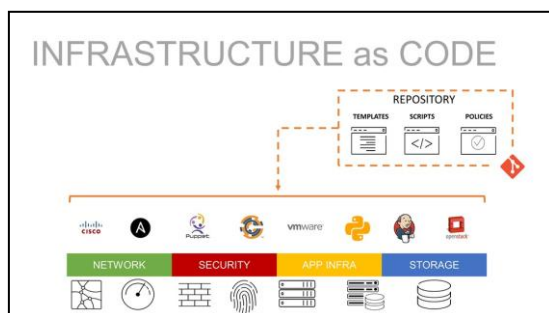
3.1. Què és IaC i per què és important?

La IaC és un conjunt de pràctiques que permeten definir, desplegar i gestionar recursos informàtics mitjançant arxius de codi en lloc de configuracions manuals. Això implica que qualsevol infraestructura, com servidors, bases de dades o xarxes, es pot configurar de manera automatitzada mitjançant scripts o plantilles declaratives.

Aquest enfocament és important perquè:

- **Automatitza processos:** Redueix la necessitat de configuracions manuals, disminuint errors humans.
- **Millora la consistència:** Garanteix que la infraestructura es desplega exactament igual en qualsevol entorn.
- **Facilita la col·laboració:** Les configuracions es poden compartir i versionar amb eines de control de versions com Git.
- **Permet escalar ràpidament:** És ideal per entorns de núvol, on es poden aprovisionar i eliminar recursos segons la demanda.
- **Agilitza el desplegament:** Redueix el temps necessari per posar en marxa nous serveis o recuperar sistemes en cas de fallada.

En el context d'AWS, IaC és fonamental per gestionar recursos de manera eficient, permetent desplegar i escalar aplicacions amb facilitat.



Font f5.com: ¿Qué es la infraestructura como código?

3.2. Problemes que resol la IaC (Automatització i escalabilitat)

Abans de la Infraestructura com a Codi, la gestió de recursos informàtics es feia manualment, cosa que generava diversos problemes:

- **Configuracions inconsistentes:** Quan es feien configuracions manuals, era difícil replicar exactament la mateixa configuració en diferents entorns.
- **Temps de desplegament lent:** La creació de servidors, xarxes i altres recursos requeria temps i coneixement tècnic.
- **Manca d'escalabilitat:** A mesura que les aplicacions creixien, afegir nous servidors o recursos podia ser un procés complicat i propens a errors.
- **Dificultats en la recuperació de desastres:** Si un sistema fallava, recuperar-lo podia ser un procés llarg i costós si no existia un registre clar de la configuració.

La IaC resol aquests problemes perquè permet definir tota la infraestructura com a codi, la qual cosa facilita la seva replicació, escalabilitat i gestió de manera eficient i automatitzada.

3.3. Eines principals: Terraform i CloudFormation.

Existeixen diverses eines per implementar IaC, però dues de les més utilitzades dins del context de AWS trobem:

Terraform:



Font: Pàgina oficial <https://www.terraform.io/>

- Desenvolupada per HashiCorp, Terraform és una eina d'IaC independent del proveïdor de núvol.
- Permet desplegar recursos en múltiples plataformes com AWS, Azure i Google Cloud.
- Utilitza fitxers en format **HCL (HashiCorp Configuration Language)** per definir la infraestructura.
- Ofereix **infraestructura immutable**, és a dir, qualsevol canvi es realitza creant nous recursos en lloc de modificar els existents.

AWS CloudFormation:



AWS CLOUDFORMATION

Font: Pàgina oficial <https://aws.amazon.com/es/cloudformation/>

- Eina específica per a AWS, permet desplegar recursos utilitzant **YAML o JSON**.
- Està totalment integrada amb l'ecosistema d'AWS, oferint una configuració nativa per als serveis del proveïdor.
- Utilitza el concepte de **stacks**, que permet desplegar conjuntament múltiples recursos relacionats.
- Ideal per a equips que treballen exclusivament amb AWS, però menys flexible si es vol operar en múltiples núvols.

Ambdues eines permeten automatitzar el desplegament d'infraestructura i garantir que els recursos es configuren de manera consistent, però Terraform és més versàtil per a entorns multi-núvol, mentre que CloudFormation és la millor opció per als usuaris d'AWS que volen una integració nativa.

3.4. Instal·lació de Terraform i eines complementaries.

En aquesta secció, veurem com preparar l'entorn de treball per començar a treballar amb Terraform i altres eines complementàries que ens ajudaran a gestionar millor la infraestructura, garantir la seguretat i mantenir la documentació actualitzada.

3.4.1. Terraform.

Es distribueix com un únic binari que es pot descarregar des de la pàgina de descàrregues del propi Terraform. Utilitzarem la guia d'instal·lació recomanada que s'ajusti al nostre sistema operatiu.

<https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli>

```
sudo apt-get install terraform
```

Comprovem que la instal·lació s'ha produït correctament:

```
$ terraform -help
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.
```

3.4.2. Tfswitch



Font: Pàgina oficial <https://tfswitch.warrensbox.com/>

És una eina de línia de comandes que facilita canviar entre diferents versions de Terraform. La instal·lació és simple i mínima. Després de la instal·lació, només cal seleccionar la versió desitjada des del menú desplegable i començar a utilitzar Terraform. <https://tfswitch.warrensbox.com/Install/>

```
curl -L https://raw.githubusercontent.com/warrensbox/terraform-switcher/release/install.sh |  
bash
```

Comprovem que la instal·lació s'ha produït correctament:

```
$ tfswitch --version  
Version: 0.13.1218
```

3.4.3. Tfsec.



Font: Pàgina oficial <https://aquasecurity.github.io/tfsec>

És una eina d'anàlisi estàtica de seguretat per al codi de Terraform. Ofereix una sortida amigable per als desenvolupadors i comprovacions documentades, facilitant la detecció i correcció ràpida i eficient dels problemes de seguretat. <https://aquasecurity.github.io/tfsec/v1.28.1/guides/installation/>

```
curl -s https://raw.githubusercontent.com/aquasecurity/tfsec/master/scripts/install_linux.sh |  
bash
```

Comprovem que la instal·lació s'ha produït correctament:

```
$ tfsec --version  
=====  
v1.28.4
```

3.4.4. Terraform docs.



Font: Pàgina oficial: <https://terraform-docs.io/>

És una utilitat que genera documentació a partir de mòduls de Terraform en diversos formats de sortida. Per configurar-lo de manera consistent, es pot utilitzar un fitxer `.terraform-docs.yml`.

Un cop configurat, cada vegada que vulguem regenerar la documentació, simplement s'ha d'executar `terraform-docs` amb la ruta corresponent. Aquesta eina és útil per mantenir la documentació actualitzada i coherent amb el codi dels mòduls de Terraform.

<https://terraform-docs.io/user-guide/installation/>

```
curl -Lo terraform-docs.tar.gz https://github.com/terraform-docs/terraform-docs/releases/download/v0.16.0/terraform-docs-v0.16.0-linux-amd64.tar.gz
```

Comprovem que la instal·lació s'ha produït correctament:

```
$ terraform-docs --version
terraform-docs version v0.16.0 1f686b1 linux/amd64
```

3.5. Proveïdor de Terraform i selecció de versions.

En Terraform, els proveïdors són connectors que permeten gestionar recursos en diferents plataformes, com AWS, Azure o Google Cloud. Són els encarregats d'interpretar el codi de Terraform i convertir-lo en accions reals sobre la infraestructura.

Per què és important la selecció de versions?

Cada proveïdor té diferents versions que poden incloure noves funcionalitats, correccions de seguretat o canvis en la sintaxi. Controlar la versió ens assegura que el nostre codi es comporta sempre de la mateixa manera, evitant incompatibilitats.

<https://developer.hashicorp.com/terraform/language/providers>

Per indicar quina versió del proveïdor volem utilitzar, afegim aquesta configuració al nostre fitxer `versions.tf`:

```
terraform {

  required_providers {
    aws = {
```

```
    source = "hashicorp/aws"
    version = ">= 4"
  }
}
```

3.6. Versions de Terraform i selecció de versions.

Igual que amb els proveïdors, és important controlar la versió de Terraform per evitar errors en l'execució del codi.

<https://developer.hashicorp.com/terraform/tutorials/configuration-language/versions>

Això es fa afegint la següent configuració al mateix fitxer versions.tf:

```
terraform {

  required_version = "~> 1.5.0"

  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = ">= 4"
    }
  }
}
```

3.7. Desenvolupament de mòduls de Terraform.

Quan treballem amb infraestructura complexa, reutilitzar codi ens ajuda a mantenir una estructura clara i fàcil de gestionar. Terraform permet crear mòduls, que són blocs de configuració reutilitzables.

Per què utilitzar mòduls?

- Eviten repetir codi en diferents parts del projecte.
- Faciliten la gestió i l'actualització de recursos.
- Milloren l'organització del codi i el manteniment.

Estructura bàsica d'un mòdul de Terraform

Per norma general un mòdul de Terraform segueix una estructura bàsica que facilita la seva comprensió i ús. A continuació, es descriu breument la disposició típica d'un mòdul:

- **main.tf:** Aquest és el fitxer principal que conté la configuració principal del mòdul. Aquí es defineixen els recursos i paràmetres que formaran part de la infraestructura.
- **variables.tf:** En aquest fitxer, es declaren les variables que es faran servir al mòdul. Això permet la flexibilitat en la configuració, ja que les variables poden ser personalitzades en cada ús del mòdul.
- **outputs.tf:** Defineix les sortides del mòdul, és a dir, els valors que es volen exposar perquè altres configuracions o mòduls els puguin utilitzar.
- **README.md:** Pot contenir documentació sobre l'ús i la configuració del mòdul. Això és especialment útil quan altres persones han de comprendre i utilitzar el mòdul.

Nom mòdul
___main.tf
___outputs.tf
___README.md
___variables.tf

3.8. Conclusió Apartat 3

La Infraestructura com a Codi ha transformat la manera com es gestiona la infraestructura informàtica, fent-la més automàtica, escalable i segura. En comptes de configurar servidors i recursos manualment, amb IaC es pot definir tota la infraestructura com a codi, garantint una gestió més eficient i repetible.

Els avantatges clau d'aquest enfocament són:

- **Automatització:** Redueix errors humans i accelera el desplegament.
- **Escalabilitat:** Permet afegir o eliminar recursos de manera dinàmica.
- **Consistència:** Garanteix que la infraestructura es desplega sempre igual.
- **Gestió eficient:** Millora la col·laboració i integració amb eines de control de versions com Git.

Per implementar IaC, hi ha eines com Terraform i AWS CloudFormation, que permeten definir i gestionar infraestructura en AWS i altres plataformes de núvol. Terraform destaca per la seva flexibilitat multi-núvol, mentre que CloudFormation és ideal per a una integració nativa en AWS.

A més, s'han explorat eines complementàries com Tfswitch (gestió de versions), Tfsec (seguretat en Terraform) i Terraform Docs (documentació automàtica), que milloren la gestió i seguretat dels desplegaments.

En definitiva, la Infraestructura com a Codi no només optimitza el desplegament d'infraestructura, sinó que també facilita la seva gestió a llarg termini.

3.9. Recursos Apartat 3

- HashiCorp. (s.f.). *Terraform by HashiCorp*. <https://www.terraform.io/>
- Amazon Web Services. (s.f.). *AWS CloudFormation*. <https://aws.amazon.com/es/cloudformation/>
- F5 Networks. (s.f.). *¿Qué es la infraestructura como código?*. <https://www.f5.com/labs/articles/education/what-is-infrastructure-as-code>
- HashiCorp. (s.f.). *Terraform Provider for AWS*. <https://registry.terraform.io/providers/hashicorp/aws/latest/docs>
- HashiCorp. (s.f.). *Managing Terraform Versions*. <https://developer.hashicorp.com/terraform/tutorials/configuration-language/versions>
- Warrensbox. (s.f.). *Terraform Switcher (Tfswitch)*. <https://tfswitch.warrensbox.com/>
- Aqua Security. (s.f.). *tfsec - Terraform Security Scanner*. <https://aquasecurity.github.io/tfsec/>
- Terraform Docs. (s.f.). *Generate documentation for Terraform modules*. <https://terraform-docs.io/>

4. Alta Disponibilitat i Desplegament d'una Web Estàtica amb AWS S3 i CloudFront

En aquest apartat s'exploraran els conceptes d'alta disponibilitat, el funcionament de S3 com a hosting estàtic, la utilització de CloudFront per optimitzar el lliurament de contingut i la gestió de la infraestructura mitjançant Terraform.

4.1. Què entenem per alta disponibilitat a AWS?

L'alta disponibilitat és la capacitat d'un sistema per continuar operatiu malgrat errors o fallades, garantint el mínim temps d'inactivitat possible. Aquest concepte és essencial en entorns de computació al núvol, on es requereix escalabilitat i resiliència.

Els components clau per aconseguir alta disponibilitat en AWS són:

- **Redundància:** Disposar de còpies dels recursos en múltiples regions o zones de disponibilitat.
- **Escalabilitat:** Capacitat d'augmentar o reduir recursos segons la càrrega.
- **Tolerància a errors:** Implementació de mecanismes de recuperació automàtica.
- **Distribució del trànsit:** Utilització de balancejadors de càrrega o CDNs per repartir les sol·licituds.

4.2. Amazon S3 com a hosting estàtic

Amazon Simple Storage Service (S3) és un servei d'emmagatzematge d'objectes d'alta disponibilitat i durabilitat. AWS S3 permet:

- Allotjar una web estàtica sense necessitat de servidors.
- Distribuir contingut d'alta disponibilitat a escala global.
- Emmagatzemar dades de manera segura amb control d'accés i xifratge.

Com funciona una web estàtica a S3?

Una web estàtica està formada per fitxers HTML, CSS i JavaScript que no requereixen execució de backend. AWS S3 permet publicar aquests fitxers, fent que siguin accessibles via HTTP/HTTPS.

Configuració bàsica per servir una web estàtica

- Crear un bucket S3 i activar l'opció de "Website Hosting".
- Pujar els fitxers HTML, CSS i JavaScript.
- Configurar polítiques d'accés per permetre l'accés públic o restringit.
- Definir la pàgina d'inici i la pàgina d'error.

Seguretat a S3

No és recomanable fer públic el bucket. Per aquest motiu, en aquesta pràctica farem servir CloudFront per servir el contingut de forma segura.

4.3. Amazon CloudFront: Optimització i seguretat

Amazon CloudFront és un servei de Content Delivery Network (CDN) que millora la velocitat i seguretat d'una aplicació web.

Per què utilitzar CloudFront en una web estàtica?

- Millora del rendiment: Serveix el contingut des de servidors més pròxims als usuaris.
- Suport per HTTPS: Permet habilitar TLS/SSL amb un certificat d'AWS ACM.
- Protecció del bucket S3: Només CloudFront pot accedir-hi, evitant exposició pública.
- Caché i reducció de càrrega: Minimitza peticions repetides al servidor origen.

Configuració bàsica d'una distribució CloudFront:

- Crear una distribució CloudFront i definir com a origen el bucket S3.
- Configurar la caché per optimitzar la distribució de contingut.
- Afegir un certificat SSL/TLS per protegir la connexió.
- Configurar restriccions d'accés perquè només CloudFront pugui accedir a S3.

Diferència clau:

Un bucket S3 pot exposar-se directament a Internet, però això és insegur. CloudFront actua com un intermediari, protegint l'origen i millorant el rendiment.

4.4. AWS Certificate Manager (ACM) i HTTPS

AWS Certificate Manager (ACM) permet gestionar certificats SSL/TLS per protegir les connexions a AWS.

Passos per habilitar HTTPS a CloudFront:

- Crear un certificat SSL/TLS amb ACM.
- Assignar el certificat a CloudFront per servir el contingut amb HTTPS.
- Redirigir tot el trànsit HTTP cap a HTTPS per evitar comunicacions no segures.

4.5. Infraestructura com a Codi (IaC) amb Terraform

Terraform és una eina que permet definir infraestructura en codi. En aquesta pràctica farem servir Terraform per crear:

- Un bucket S3 amb les configuracions necessàries.
- Una distribució CloudFront per optimitzar el lliurament.
- Polítiques de seguretat IAM per controlar l'accés.

Beneficis d'utilitzar Terraform en AWS

- Automatització: Evita configuracions manuals repetitives.
- Consistència: La infraestructura sempre es desplega de la mateixa manera.
- Facilitat de manteniment: Es poden fer modificacions i aplicar-les fàcilment.

4.6. Seguretat i permisos en S3 i CloudFront

Quan es desplega una web estàtica a AWS, és fonamental garantir que el contingut es distribueixi de manera segura i controlada. Sense una configuració adequada, un bucket S3 podria estar exposat a Internet, permetent a qualsevol persona accedir-hi directament.

Per evitar aquest risc, es recomana bloquejar l'accés públic al bucket S3 i configurar CloudFront com a únic punt d'accés, millorant així la seguretat i el rendiment. A més, l'ús de xifratge de dades en repòs i en trànsit i l'aplicació de polítiques IAM restrictives són pràctiques clau per protegir la infraestructura.

Bones pràctiques de seguretat

- Bloquejar l'accés públic al bucket S3.
- Permetre només CloudFront per accedir al bucket.
- Activar encriptació de dades en repòs i trànsit.
- Restringir accés per IP si és necessari.

Per què bloquegem l'accés públic?

Si un bucket S3 és públic, qualsevol persona pot veure el contingut. En lloc d'això, fem que només CloudFront pugui accedir-hi, millorant la seguretat.

4.7. Validació i proves del desplegament

Un cop finalitzat el desplegament de la infraestructura, és imprescindible verificar que tot funciona correctament.

Com validar que el desplegament funciona?

- Accedir a l'URL de CloudFront i comprovar que la web es carrega correctament.
- Validar que HTTPS està activat i funciona correctament.
- Verificar que S3 no és públic i que només CloudFront pot accedir-hi.
- Analitzar temps de resposta i caché per optimitzar el rendiment.

4.8. Conclusió Apartat 4

El desplegament d'una web estàtica amb Amazon S3 i CloudFront permet aconseguir una arquitectura d'alta disponibilitat, segura i escalable sense necessitat de gestionar servidors. Aquest model és àmpliament utilitzat en empreses i serveis en línia per la seva eficiència i baix cost. A través d'aquesta implementació, hem après:

- Com utilitzar Amazon S3 per allotjar contingut estàtic de manera eficient.
- La importància d'un Content Delivery Network (CDN) com CloudFront per reduir la latència i millorar el rendiment global.
- Com aplicar bones pràctiques de seguretat, incloent l'ús de IAM, HTTPS i restriccions d'accés per protegir la infraestructura.
- L'avantatge d'automatitzar la gestió d'infraestructura mitjançant Terraform, seguint el model d'Infrastructure as Code (IaC).

Aquest enfocament proporciona una base sòlida per entendre els conceptes fonamentals de computació al núvol, seguretat en xarxes i administració d'infraestructura moderna. Amb aquests coneixements, els alumnes podran aplicar pràctiques avançades de gestió de serveis en AWS i adaptar-se a entorns professionals amb altes exigències de disponibilitat i seguretat.

4.9. Recursos Apartat 4

- Amazon Web Services. (s.f.). Hosting a static website using Amazon S3. <https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html>
- Amazon Web Services. (s.f.). Amazon CloudFront: Global Content Delivery Network (CDN). <https://aws.amazon.com/cloudfront/>
- Amazon Web Services. (s.f.). AWS Certificate Manager (ACM). <https://aws.amazon.com/certificate-manager/>
- Amazon Web Services. (s.f.). Securing AWS S3 buckets: Best Practices. <https://aws.amazon.com/premiumsupport/knowledge-center/secure-s3-resources/>