

T1.-Introduccio.pdf



dadeess



Computació en Entorns al Núvol



3º Grado en Ingeniería de Datos



Escuela de Ingeniería
Universidad Autónoma de Barcelona



Pack portátil + monitor portátil

Más posibilidades. más productividad.
más portátil. Más de todo.

msi



Si una sola pantalla te sabe a poco, MSI tiene el dúo dinámico que necesitas. Un combo portátil + monitor extra para que tu setup de trabajo te siga a todos lados como tu ex stalkéandote. Más espacio, más productividad, menos excusas.



Más posibilidades.
más productividad.
más portátil.
Más de todo.

msi

Cloud Computing

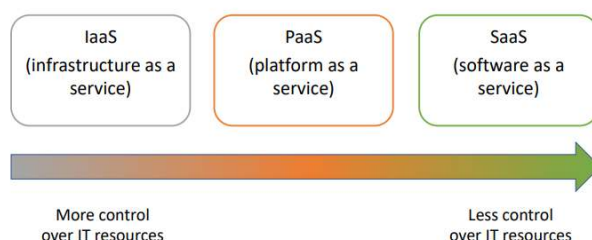
T1. INTRODUCCIÓ

Cloud computing

Computació al Núvol (Cloud Computing): disponibilitat sota demanda dels recursos del sistema informàtic especialment les dades d'emmagatzematge i potencia informàtica, sense accés directe per l'usuari.

- Centres de dades disponibles per molts usuaris en Internet.
- Permet assignar i implementar dinàmicament els serveis necessaris en cada moment.
- Els grans clouds tenen funcions distribuïdes en múltiples ubicacions des de servidors locals.

Models de Cloud service



IaaS: major control sobre els recursos IT.

PaaS: equilibri entre control i facilitat d'ús

SaaS: menys control, però més accessible pels usuaris finals.

Avantatges del Cloud Computing

- Reducció de costos
- **Elasticitat, escalabilitat i eficiència.**
- Disponibilitat global en minuts
- Major agilitat i velocitat de desplegament
- Seguretat avançada i compliment normatiu.
- Cloud
 - Virtualization
 - Virtual Machines
 - Elasticity
 - Dynamic dimensioning
 - Dynamic provisioning

Públic Cloud / Private Cloud

- **Públic:** compartit entre múltiples usuaris (AWS, Azure, Google Cloud)
- **Privat:** exclusiu per una organització (OpenStack, OpenNebula, Cloud Stack)
- **Híbrid:** combinació de núvols privats i públics per optimitzar costos i seguretat.
- **MultiCloud:** ús de múltiples proveïdors per evitar dependència d'una sola plataforma.

Hipervisor

Sistema de software que agrupa recursos informàtics (processament, memòria i emmagatzemament) i els assigna entre les màquines virtuals. Es a dir, podem executar varies màquines virtuals en una única màquina física.

- Els hipervisors són la tecnologia darrera la **virtualització**, es a dir, l'eina que separa el hardware del software.
- Han permès emprar els serveis de computació en el núvol en aplicacions empresarials.
- Permeten escalar els serveis de manera eficient.

WUOLAH

Virtualització i Elasticitat

La virtualització es clau en el núvol, permet que un mateix hardware executa múltiples màquines virtuals mitjançant un hipervisor.

- **Elasticitat:** ajust dinàmic de recursos en funció de la demanda.
- **Eficiència:** optimització de l'ús de hardware.
- **Escalabilitat:** capacitat d'afegir més instàncies segons sigui necessari.

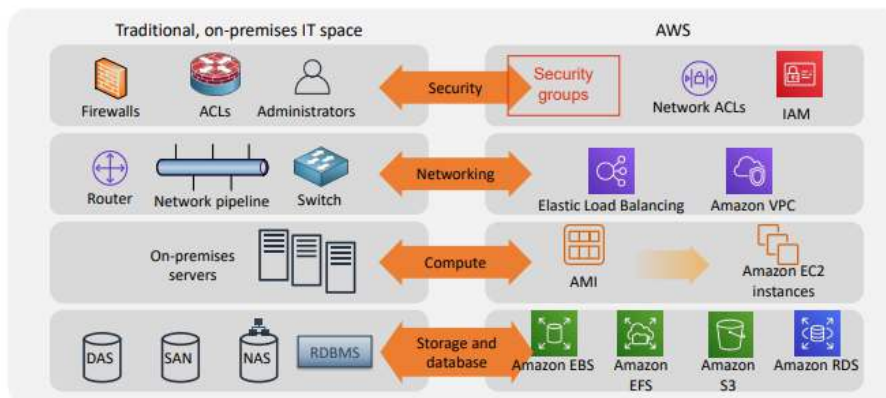
AWS (Amazon Web Services)

Principals proveïdors del cloud i ofereix múltiples serveis. S'ha de basar en 5 pilars:

- Excel·lència d'operacions: tenir a la disposició del client la cantat de servis necessaris sense que et faltin però tampoc sense que sobrin ja que si no, esta malgastant energia.
- Seguretat
- Optimització dels costos
- Rendiment: que la arquitectura tingui un rendiment bo.
- Tolerància: l'arquitectura ha de ser tolerant a falles de sistema i tot i així poder seguir funcionant.

Per tenir un bon servei de cloud necessitem tenir certs recursos, els més importants son les màquines virtuals, els servidors lambda i els contenidor.

Similaritat entre AWS i IT tradicional



Beneficis del Cloud Computing

- **Reducció de costos:** pagament per ús en lloc de grans inversions en hardware.
- **Economies d'escala massives:** a causa de l'ús agregat de tots els clients, AWS pot aconseguir majors economies d'escala i transferir l'estalvi als clients.
- **Escalabilitat i elasticitat:** ajust dinàmic de recursos segons la demanda. (stop guessing capacity)
- **Augmenta agilitat i rapidesa en el desplegament:** implementació de serveis en minuts.
- **Alta disponibilitat i fiabilitat:** redundància en múltiples ubicacions.
- **Stop gastar diners en mantenir centres de dades**
- **Seguretat millorada:** compliment de normatives internacionals i xifrat de dades.

IoT Platforms on Cloud

Algunes plataformes destacades son:

- **AWS IoT:** Plataforma d'Amazon per administrar dispositius connectats.
- **Azure IoT Hub:** Solució de Microsoft per integrar IoT amb serveis al núvol.
- **IBM Watson IoT:** Plataforma per anàlisis avançat de dades IoT.

Edge Computing y Fog Computing

Per reduir la latència en aplicacions IoT, s'utilitzen models de processament com:

- **Edge Computing:** processament de dades a la vora de la xarxa, propers dels dispositius IoT.
- **Fog Computing:** extensió del cloud amb processament intermig abans d'enviar dades al cloud central.

Dell technologies, IBM i Microsoft

- Desenvolupen estratègies de cloud perifèrics i 5G
- Els operadors han reconegut la importància de l'edge computing per oferir 5G. També hi ha hagut una sèrie de proves en el mercat telefònic.
- Però donada la trajectòria de la inversió en xarxa i el desenvolupament del producte, encara no hi som del tot, estem a la etapa de la idea/estratègia.

Bosch and IBM clouds

- Bosch connecta 6 milions de dispositius amb *cloud foundry*
- **IBM cloud foundry:** es una plataforma que no només crea noves apps o emigra les existents o ofereix IaaS/PaaS. Esta dissenyada per donar tots aquests aspectes conjuntament per ajudar a resoldre un problema real, problemes complexes en el cloud.

Protocolos y Tecnologías IoT

- **Peer-to-Peer Technologies:** Bluetooth, WiFi Direct, NFC
- **Short-range: Bluetooth Low-Energy:** Zigbee, Z-Wave, and 6LoWPAN
- **Local Area Network (LAN):** WiFi
- **Long-distance Cellular:** GSM / GPRS, LTE
- **Low-power Long-distance:** LoRa/ LoRaWan (permet que els dispositius de baixa potencia utilitzin long range gateways per connectar-se a un codi obert per intercanviar dades amb aplicacions), NB-IoT, LTE-M

Aplicacions de Cloud Computing en la Industria

- **Automoció i vehicles definits per Software:** Sistemes en el núvol permeten actualitzacions OTA (over the air). Connectivitat amb sensors per conducció autònoma.
- **Industria i manufactura intel·ligent:** "lights out manufacturing" fabriques autònomes sense intervenció humana. Empreses com FANUC utilitzen robots en fabricació sense supervisió.
- **Banca digital i microserveis:** Monzo (banc digital de regne unit) utilitza AWS i arquitectura de microserveis.



Más posibilidades.
más productividad.
más portátil.
Más de todo.

msi®

Cloud Computing

Serverless Computing & AWS Lambda

El model **serverless** permet executar codi sense gestionar servidors (sense preocupar-se per la infraestructura).

Beneficis:

- Escalabilitat automàtica
- Sense costos d'infraestructura
- Execució eficient.

→ **AWS Lambda**: servei de computació serverless (sense servidors) que permet executar codi en resposta a events sense necessitat d'aprovisionar o administrar servidors. Característiques principals:

- **Execució sense administració de servidors**: no es necessari aprovisionar instàncies ni preocupar-se per la infraestructura.
- **Activació per events**: s'executa automàticament en resposta a event de serveis com API Gateway...
- **Escalabilitat automàtica**: augmenta o disminueix els recursos assignats segons la demanda
- **Pagar per ús**: es cobra únicament pel temps d'ús
- **Monitoreig integrat**: utilitza amazon cloudwatch per registrar mètriques.

Casos d'ús: processament de dades en temps real, automatització de tasques, backends sense servidors...

DIFERENCIÈS AMB EL MODEL TRADICIONAL:

En una infraestructura tradicional per executar aplicacions es necessari:

- Configurar una instància
- Actualitzar el sistema operatiu
- Instal·lar la plataforma d'aplicacions (frameworks, bases de dades...)
- **Desplegar i mantenir aplicacions**
- Configurar escalabilitat autonòmica i balanceig de carrega
- **Monitoritzar i mantenir es aplicacions**

Els dos en verd son els que també s'han de fer en Serverless (la resta NO).

Desenvolupament WEB en el cloud

Cloud computing facilita la implementació d'aplicacions web amb frameworks moderns.

Web App VS. Web Page

Las web apps solen requerir major infraestructura i gestió, mentre que les pàgines web poden ser estatiques i emmagatzemades en serveis més senzills com Amazon S3 o Github.

Web app exemples: gmail, facebook, google docs...

Web page exemples: wikipedia, blogs...

[Back-end](#)

WUOLAH

- **Flask** (Python) – Framework lleuger per aplicacions web.
- **Django** (Python) – Framework robust amb seguretat integrada.
- **Node.js** (JavaScript) – Plataforma per aplicacions en temps real.
- **HTML** (estructura de la pagina web)
- **CSS**: disseny i estils visuals.

Front-end

- **Vue.js, React i Angular** – Frameworks moderns per interfases interactives.

Desplegament en el Cloud

- Us d' **Azure CLI** per desplegar aplicacions en **Azure App Services**.
- **NginX y Gunicorn** per optimitzar el rendiment d'aplicacions Flask.

Communication Ports

Els servidors web i aplicacions en el cloud utilitzen ports de comunicació per intercanviar dades. Alguns del més utilitzats son 80 (http – tràfic no segur), 443 (https – tràfic segur), 22 (ssh – accés remot segur)...

Model-Vista-Controlador (MVC)

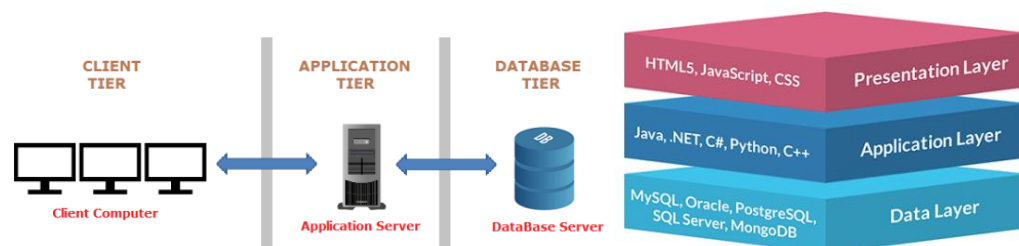
Es un patró de disseny que separa la lògica d'una aplicació en tres components principals:

- **Model**: representa les dades i la lògica. Pot ser una base de dades.
- **Vista**: interfase d'usuari que mostra les dades a l'usuari final.
- **Controlador**: gestiona la comunicació entre la vista i el model, interpretant accions de l'usuari i actualitzant dades.

Three-Tier Arquitectura

Disseny estructural comú en aplicacions web, divideix el sistema en tres nivells independents:

1. **Capa de presentació**: interfase de l'usuari que interactua amb l'usuari final, s'implementa amb HTML, CSS i JavaScript.
2. **Capa lògica o aplicació**: conté la lògica i el processament de dades, s'executa en servidors cloud. Utilitza frameworks com Flask.
3. **Capa de dades**: emmagatzeman i gestiona la informació de l'aplicació. Pot ser una base de dades relacional (MySQL) o NoSQL.



msi[®]

Más posibilidades,
más productividad,
más portátil.
Más de todo.



Pack portátil + monitor portátil



Si una sola pantalla te sabe a poco, MSI tiene el dúo dinámico que necesitas. Un combo portátil + monitor extra para que tu setup de trabajo te siga a todos lados como tu ex stalkeándote. Más espacio, más productividad, menos excusas.

Azure CLI

CLI – Command Line Interface: conjunt de comandes utilitzades per crear i gestionar recursos Azure. S'utilitza CLI en el cloud o localment.

Azure utilitza web accés, cloud CLI, local CLI i visual Studio code.

ACTIVITAT – Desplegar Web Python En Azure App Service

Pas 1: instal·lar les eines necessàries.

- ♦ **Python (3.6 o superior)** → Descarrega desde python.org
- ♦ **Azure CLI** → Descarrega desde [Azure CLI](https://azurecli.azure.microsoft.com/cli)
- ♦ **Git** → Descarrega desde [Git SCM](https://git-scm.com)
- ♦ **Visual Studio Code** → Descarrega desde [VS Code](https://code.visualstudio.com)

Pas 2: iniciar sessió en Azure

Obre la terminal i executa:

az login → això obrirà una finestra on has d'iniciar sessió

Pas 3: clonar i configurar l'app Flask

```
git clone https://github.com/Azure-Samples/python-docs-hello-world
cd python-docs-hello-world
```

```
py -m venv .venv
.venv\scripts\activate # En Windows
pip install -r requirements.txt
```

flask run → executar l'aplicació flask localment i obrir el navegador que surt

```
(.venv) PS C:\Users\adria\OneDrive - UAB\3 ENGINE\2 SEMESTRE\Computació en Entorns al Nuvol\Practicas\python-docs-hello-world> flask run
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Pas 4: desplegar l'aplicació en Azure

```
az webapp up --sku F1 --name <nombre-de-tu-app>
```

- ♦ **<nombre-de-tu-app>** → tria un nom únic per la teva app

Pas 5: modificar l'app i ver logs

Si vols modificar l'aplicació, edita l'arxiu app.py i després observa els registres en temps real amb:

az webapp log tail → també pots veure els registres en Azure Portal en l'opció Log Stream



Más posibilidades.
más productividad.
más portátil.
Más de todo.

msi®

Cloud Computing

Crear una aplicació desde 0

Crear una nova carpeta en Visual Studio Code, afegir un arxiu anomenat **hello.py** amb el següent codi:

```
from flask import Flask

myapp = Flask(__name__)

@app.route("/")
def hello():
    return "¡Mi primera aplicación Flask en la nube!"

if __name__ == "__main__":
    myapp.run(host="0.0.0.0", port=5000)
```

Crear l'arxiu **requirements.txt**:

```
Flask==1.1.1
```

Executa l'aplicació localment:

```
set FLASK_APP=hello:myapp # windows
flask run
```

AzureCLI.sh desde Flask App in Visual Studio Code

```
#!/bin/bash

# Configuración de variables
SUBSCRIPTION="Azure para estudiantes"
RESOURCEGROUP="appsvc_linux_centralus"
LOCATION="centralus"
PLANNAME="appsvc_linux_centralus"
PLANSKU="F1"
SITENAME="test4dani"
RUNTIME="PYTHON|3.8"

# 1. Iniciar sesión en Azure
az login

# 2. Listar suscripciones disponibles
az account list -o table

# 3. Seleccionar la suscripción deseada
az account set --subscription "$SUBSCRIPTION"

# 4. Crear un grupo de recursos
az group create --name "$RESOURCEGROUP" --location "$LOCATION"

# 5. Crear un plan de servicio de aplicaciones (máquina donde correrá la app)
az appservice plan create --name "$PLANNAME" --location "$LOCATION" --is-linux --sku "$PLANSKU" --resource-group "$RESOURCEGROUP"

# 6. Crear la aplicación web en Azure
az webapp create --name "$SITENAME" --plan "$PLANNAME" --runtime "$RUNTIME" --resource-group "$RESOURCEGROUP"

# 7. Configurar despliegue desde repositorio Git local (opcional)
# az webapp deployment source config-local-git --name "$SITENAME" --resource-group "$RESOURCEGROUP"
# git remote add azure "https://$USERNAME@$SITENAME.scm.azurewebsites.net/$SITENAME.git"
# git push azure master

# 8. Abrir la aplicación desplegada en el navegador
az webapp browse --name "$SITENAME" --resource-group "$RESOURCEGROUP"
```

WUOLAH