SECURE CLOUD ARCHITECTURE

OGC-Team Expertise for Cool Delivery

SECURITY BLUE TEAM

TEAM

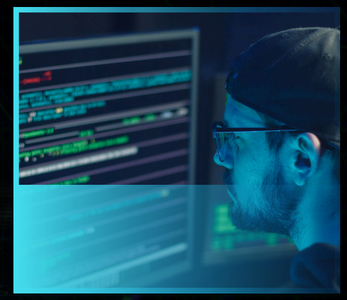
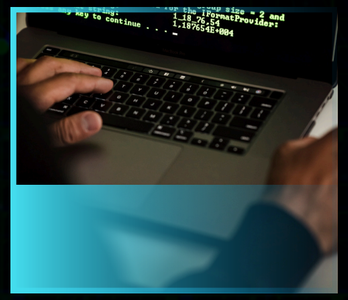
Bilal Benlahcene

Silviu Zaino

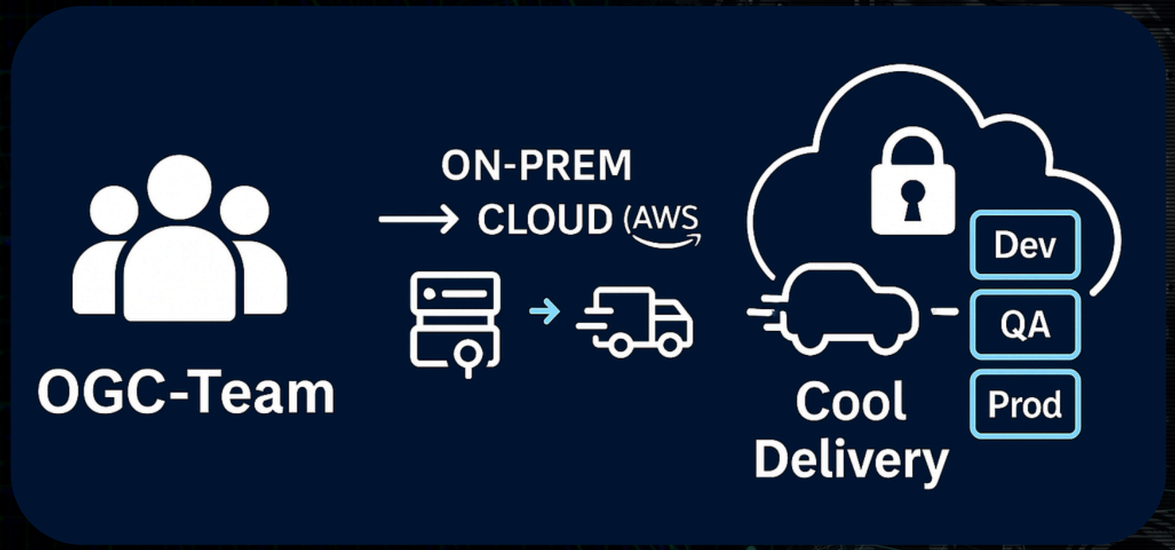
Raphael de Monchy

Louis Outrey

SUMMARY

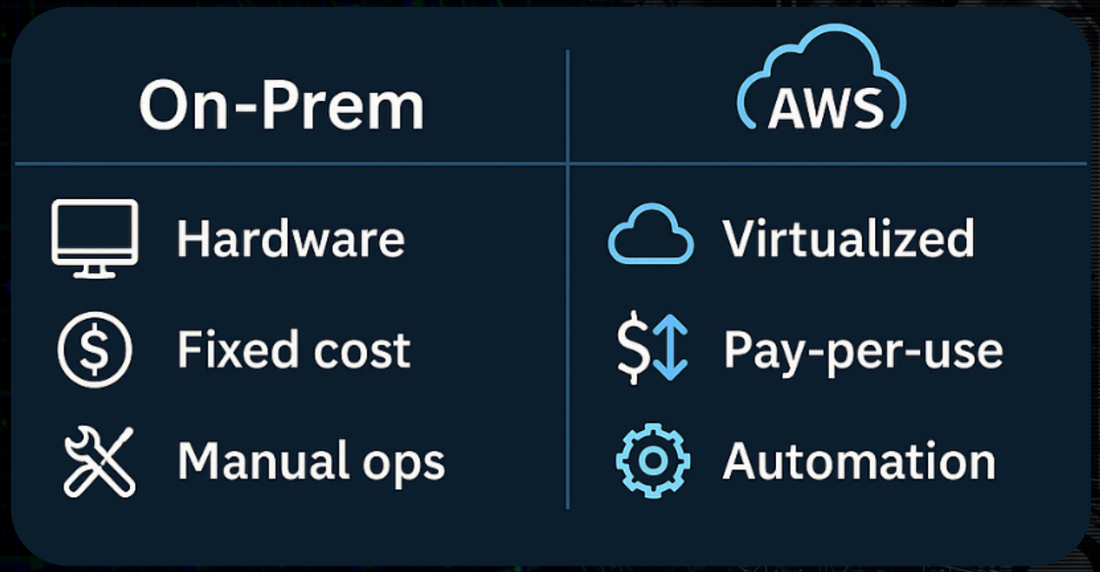



OUR MISSION



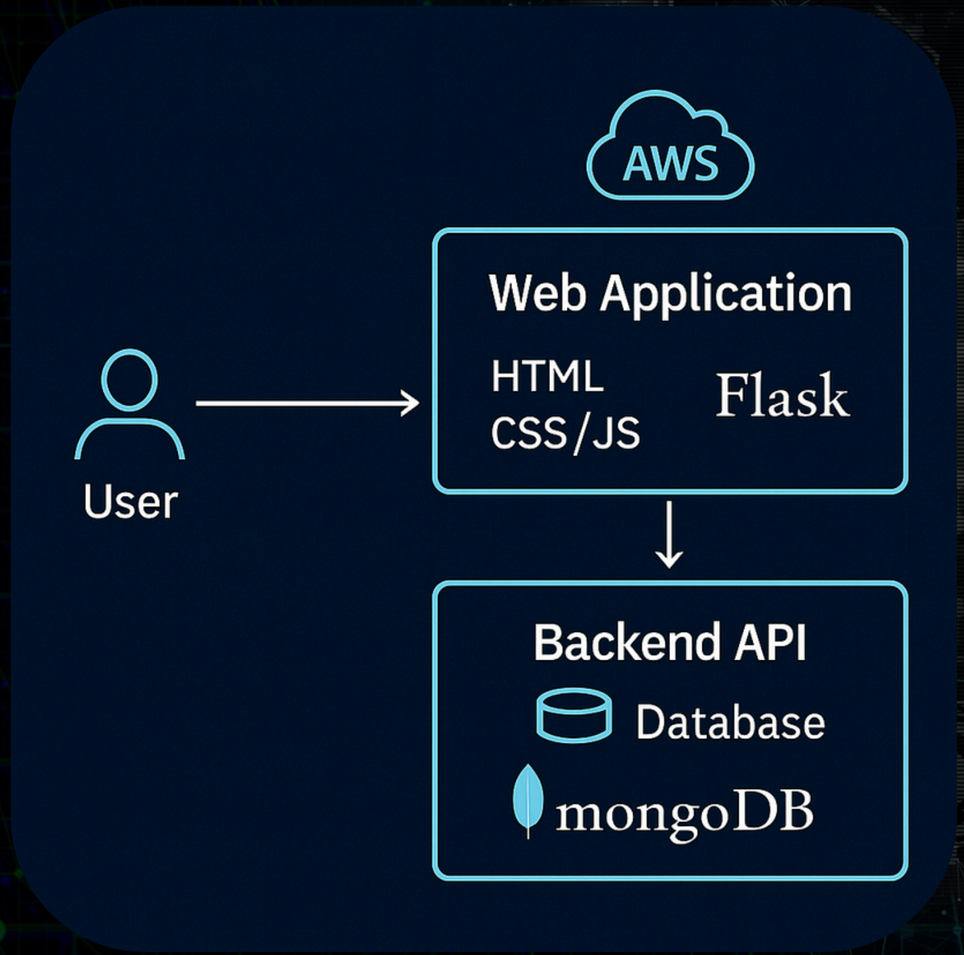
- OG-Team was assigned to secure and modernize Cool Delivery's infrastructure.
- We designed a scalable, secure AWS-based architecture.

WHY MOVE TO THE CLOUD ?



- Scalability
- Flexibility
- Cost efficiency

PROJECT SCOPE

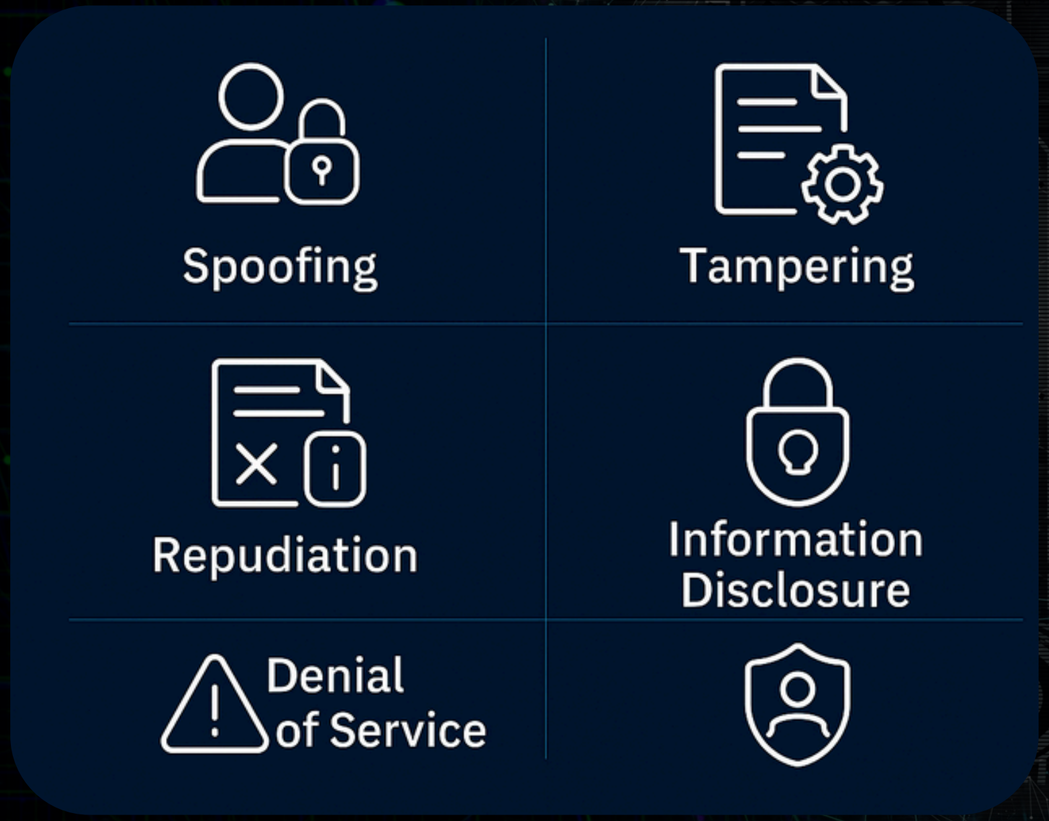


- 3 Environments
- Full stack: Front(s3), api(flask), db (mongodb)
- Goal: secure & scalable setup

FUNDAMENTALS RECAP

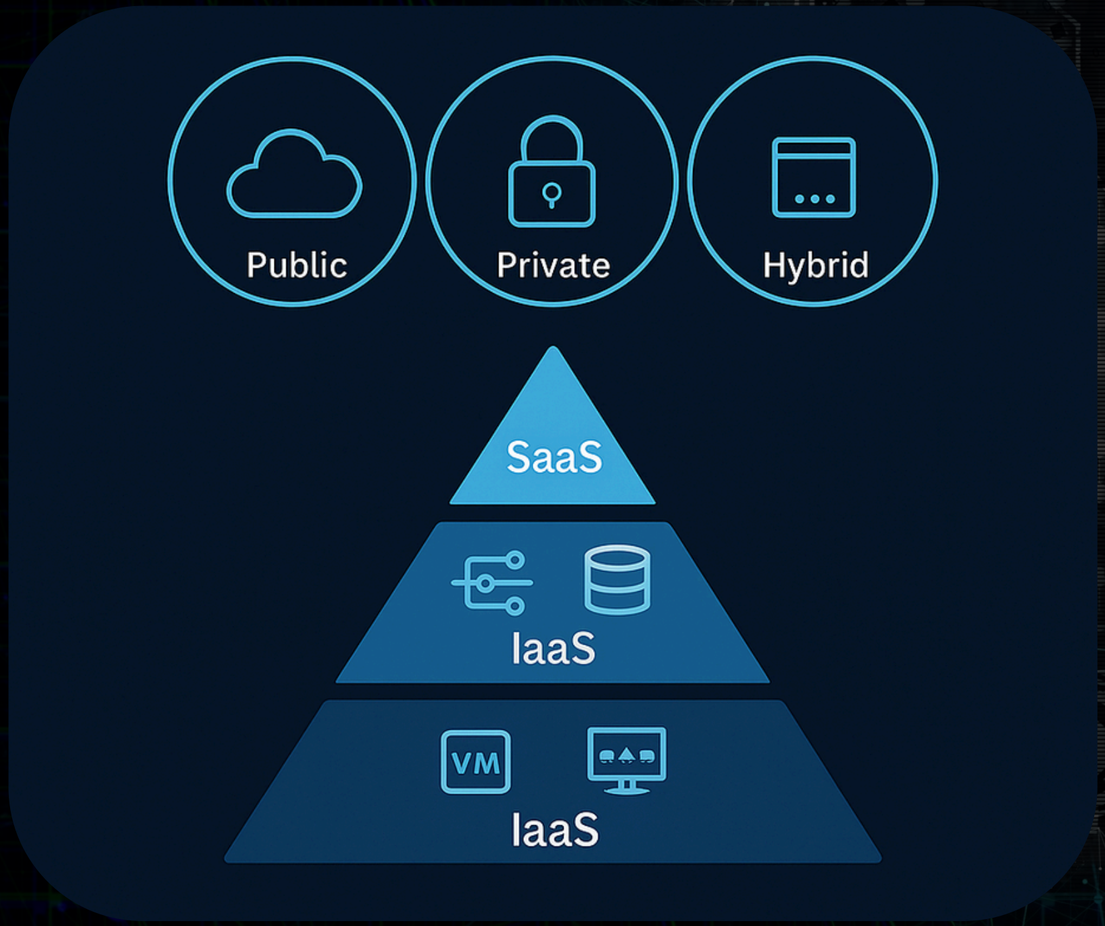
- Cloud: shared computing resources
- Security = protection
- Architecture = structured design

STRIDE THREAT MODEL SUMMARY



- MFA blocks spoofing
- IAM least privilege prevents privilege escalation
- Cloudtrail logs actions

CLOUD TYPE



- Cloud types: public, private, hybrid
- IaaS: infra / vm
- PaaS: platform / db / app hosting
- SaaS: ready-to-use apps

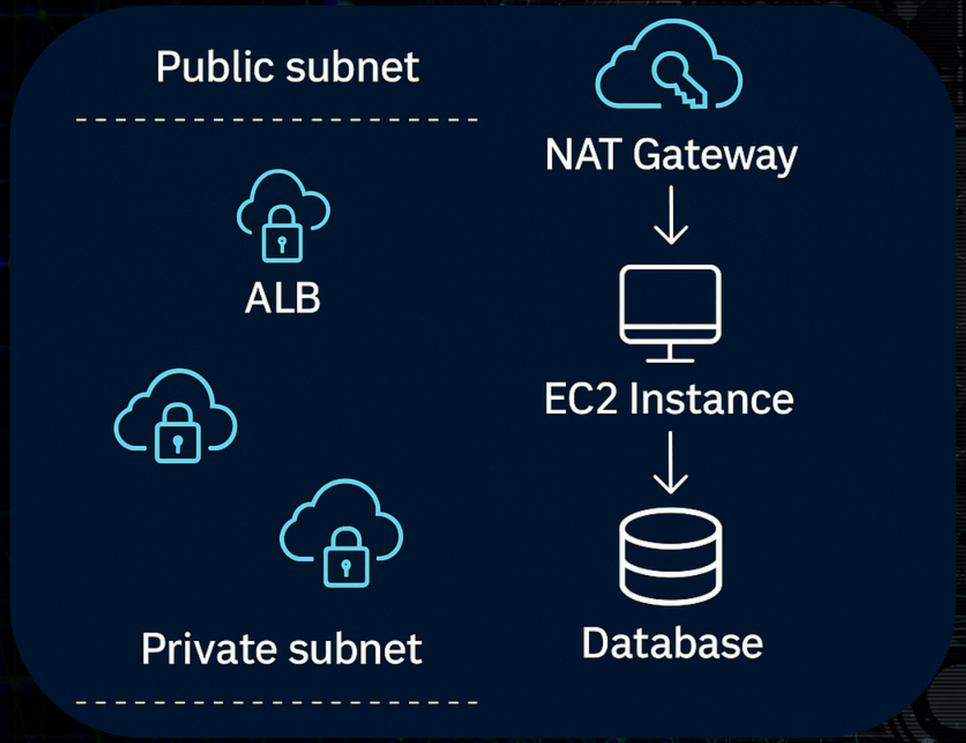
CLOUD SECURITY THREATS

- Top attack vectors in cloud environments
- Often enabled by human error or poor configuration
- Threat modeling is key to mitigation

CLOUD SECURITY THREATS

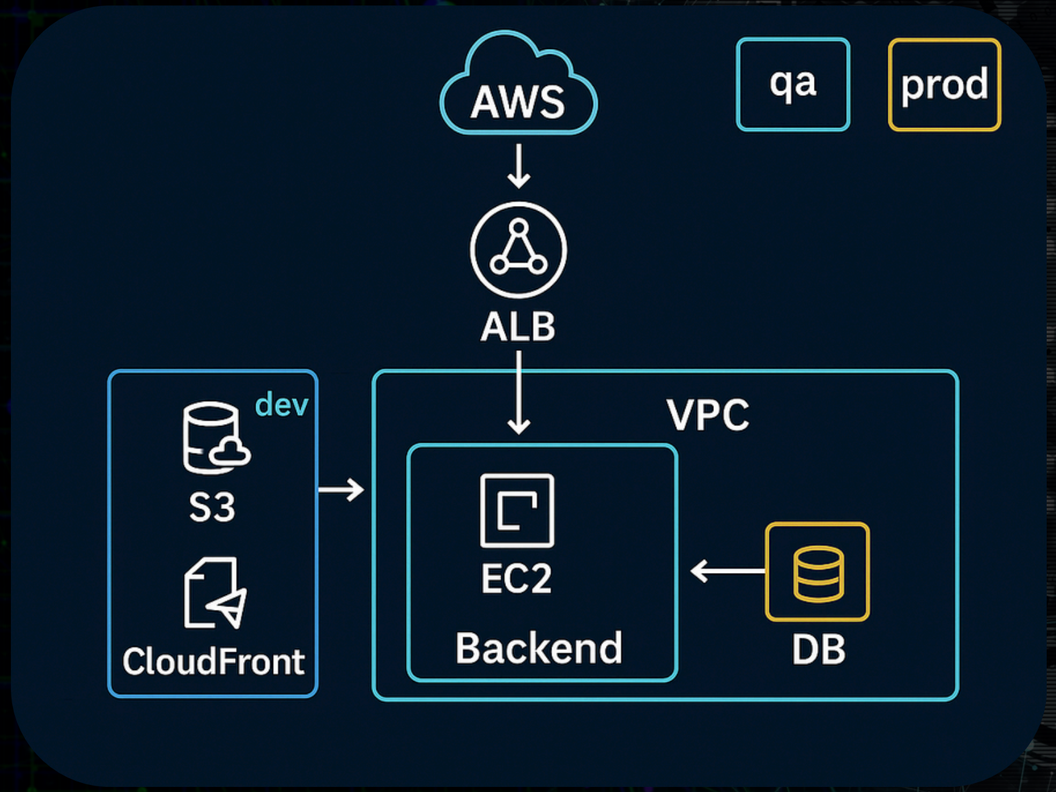
- Breaches = Legal, Financial & Trust Damage
- Misconfigurations = Most common entry point
- Cloud incidents = Higher recovery time and costs

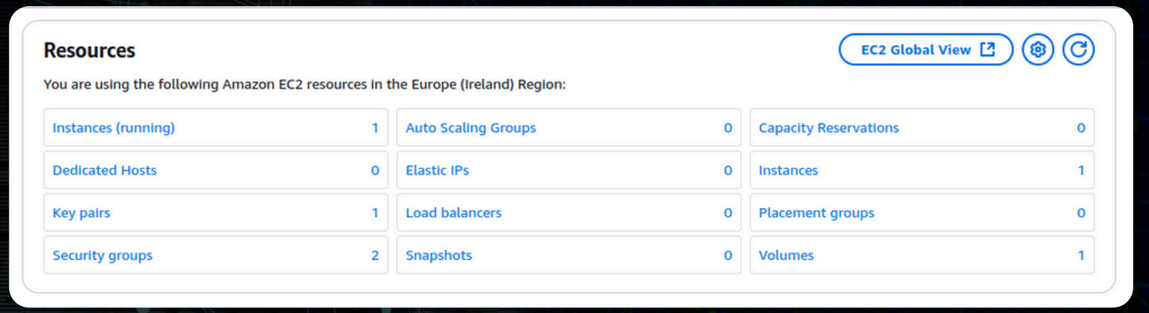
NETWORK SEGMENTATION & SECURITY



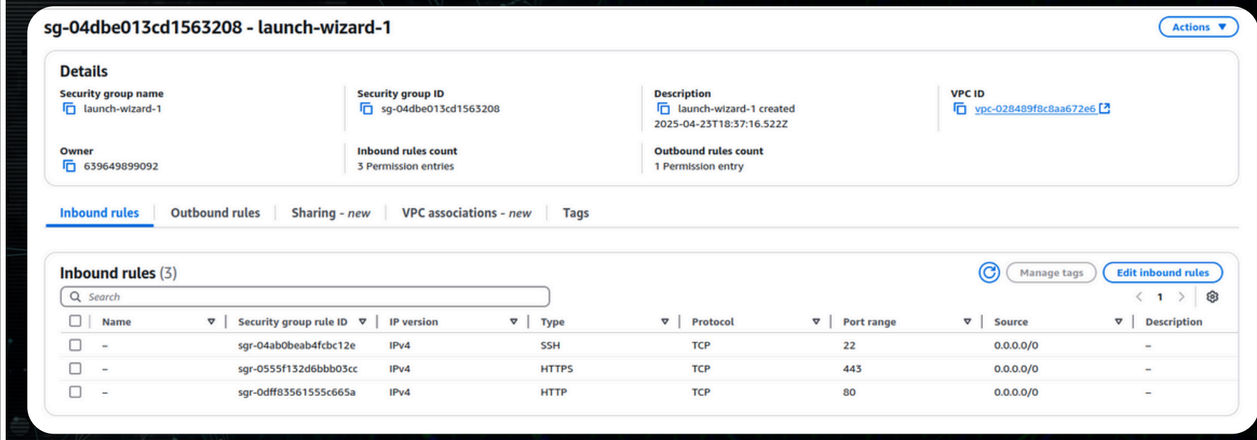
- Public : loader balancer, s3 (statique)
- Private: ec2 (backend)
- Secured: logs & monitoring

OVERVIEW ARCHITECTURE



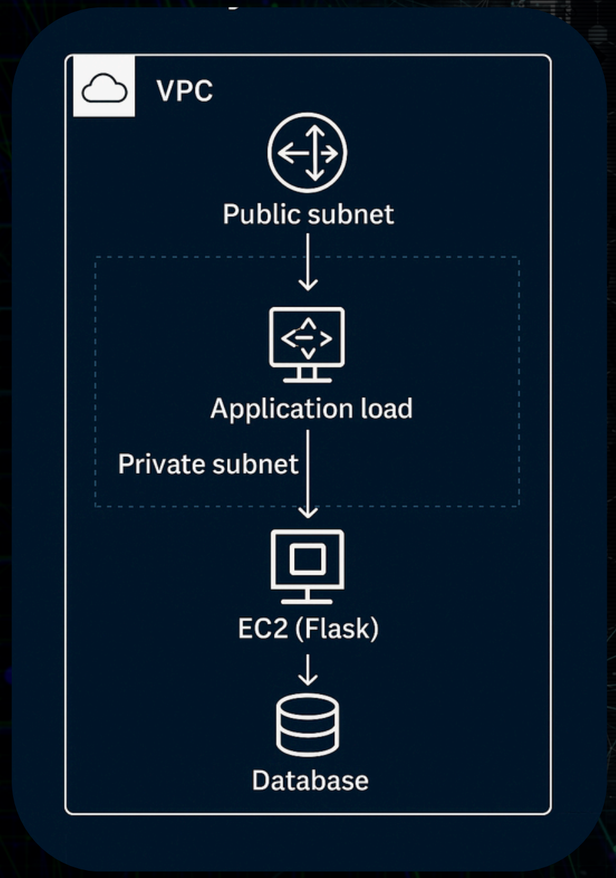


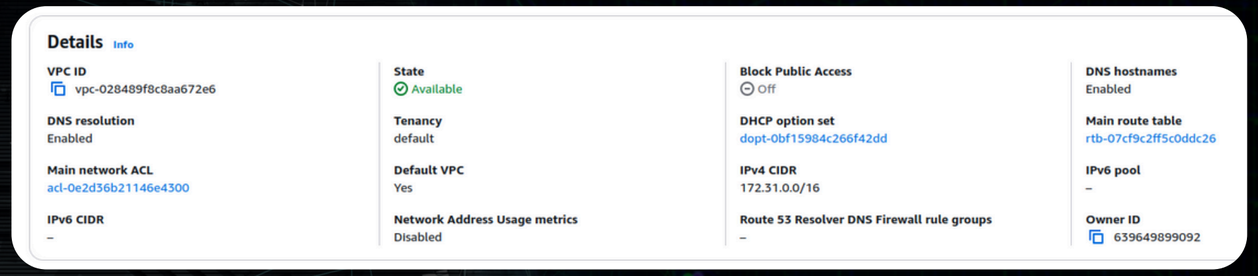
Backend ec2



Frontend s3

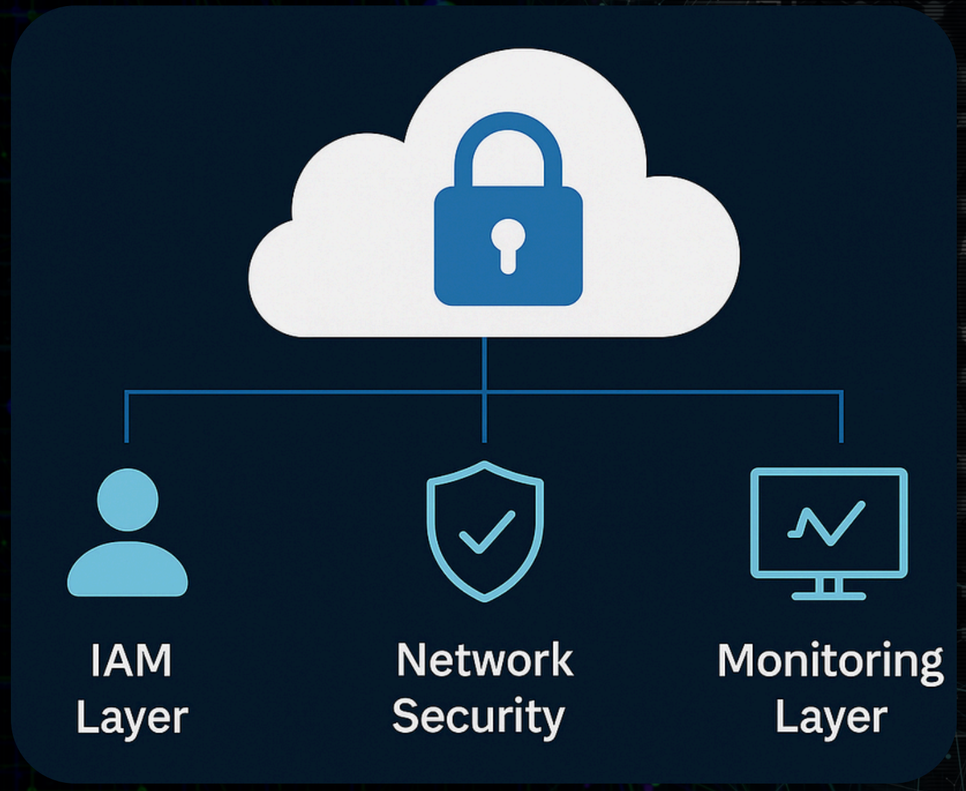
NETWORK AND SEGMENTATION

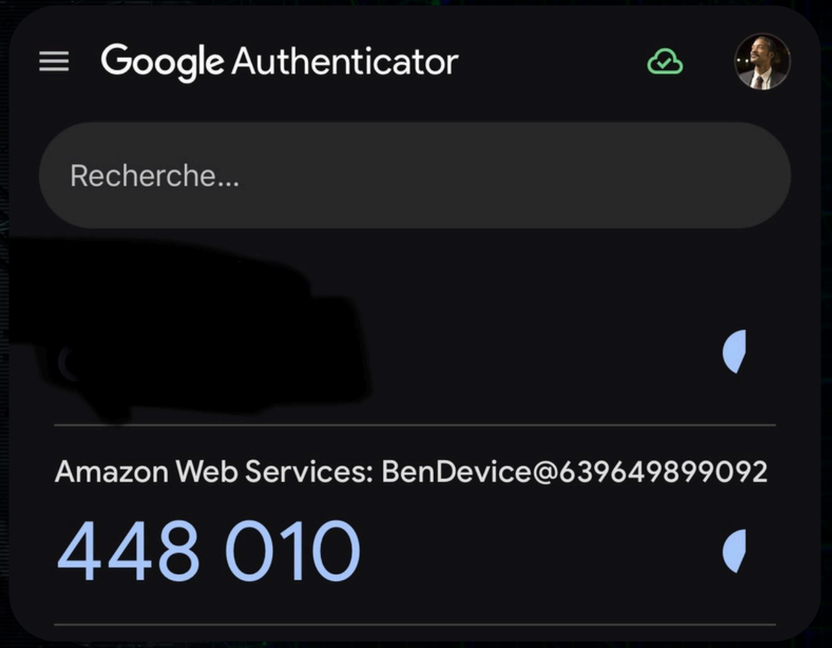




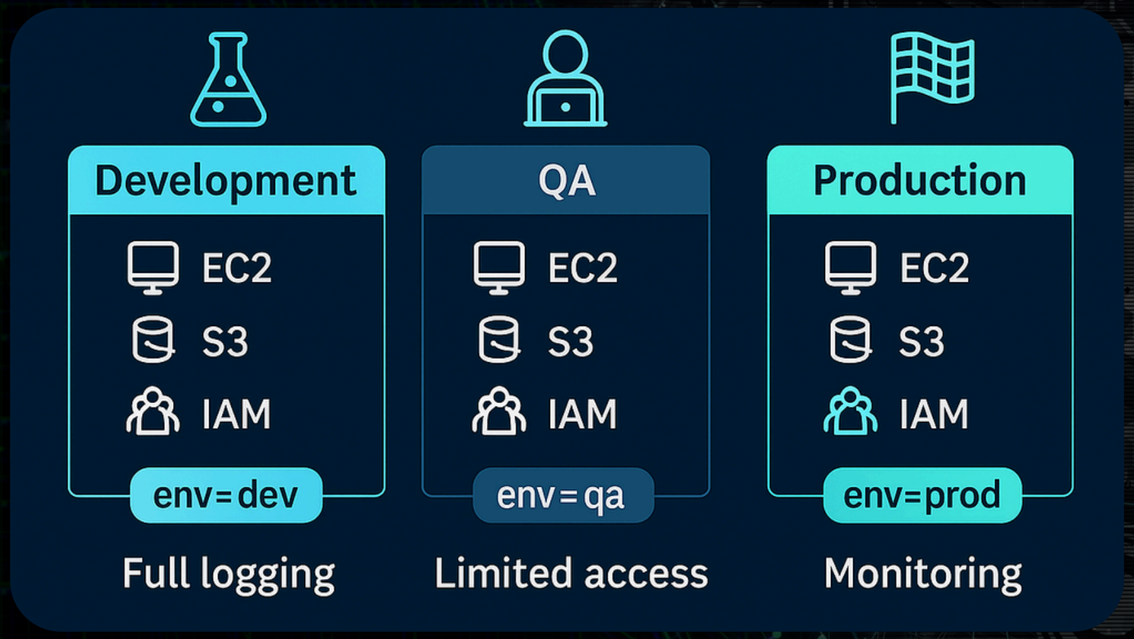
Subnets (vpc)

IAM & SECURITY





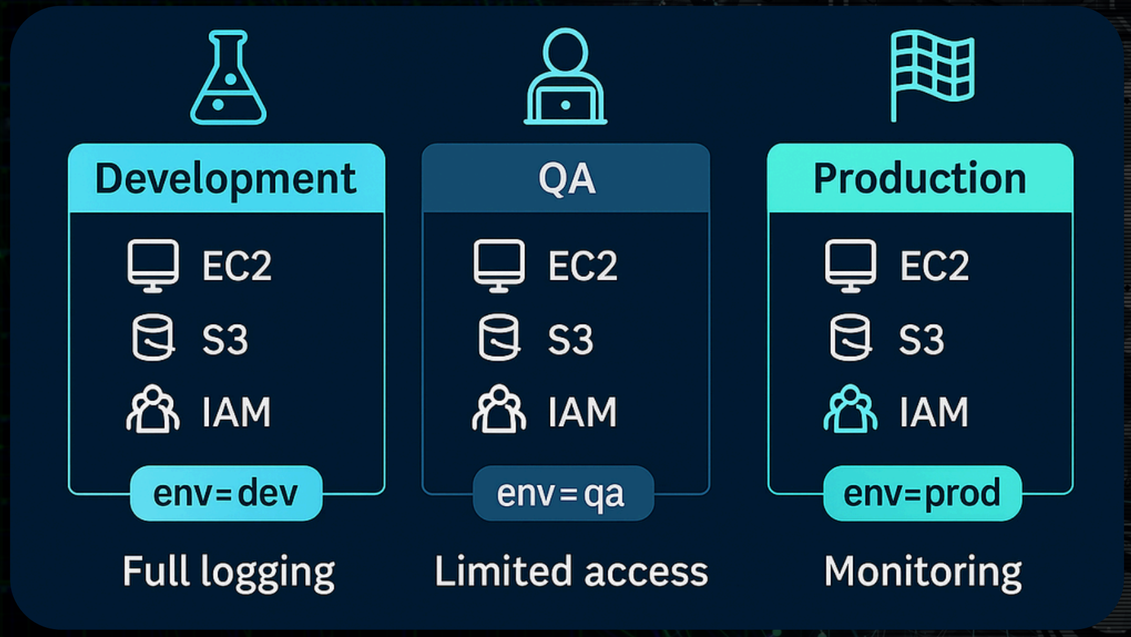
SEPARATION DEV / QA / PROD

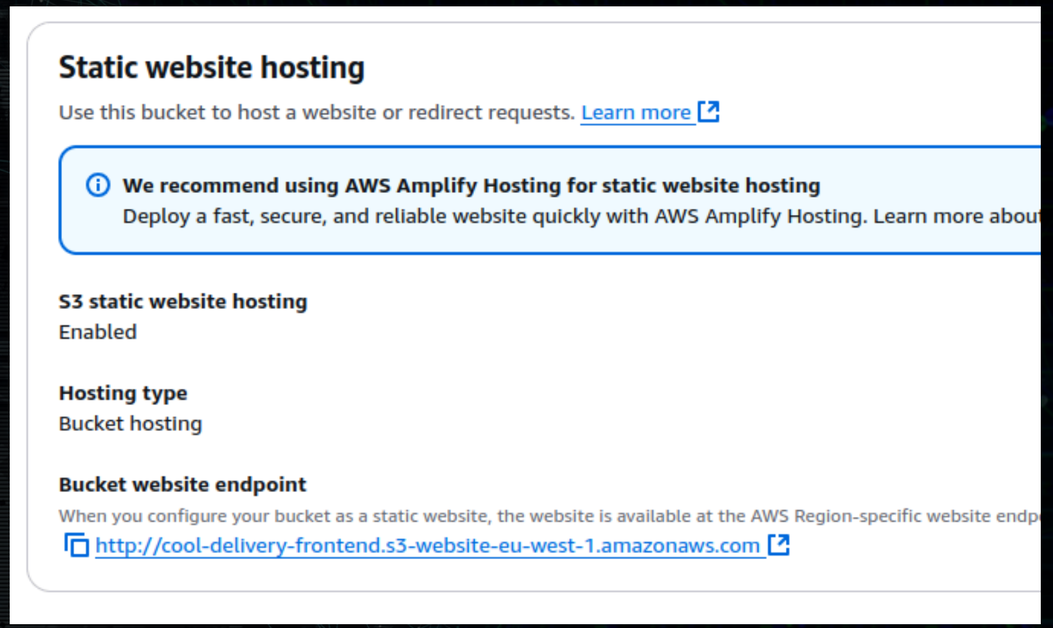


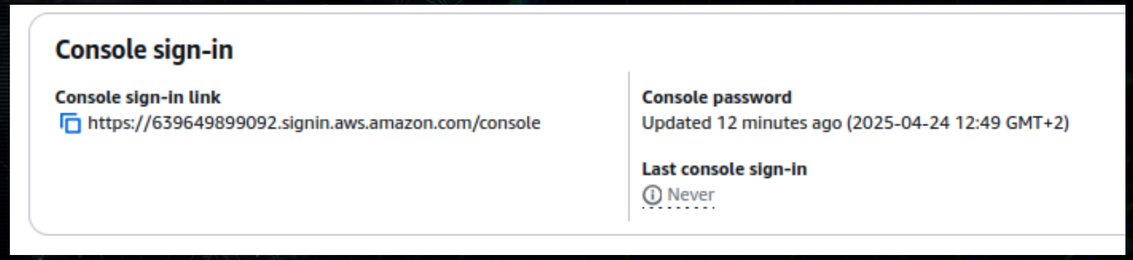

```
[ec2-user@ip-172-31-44-223 ~]$ sudo systemctl enable auditd
[ec2-user@ip-172-31-44-223 ~]$ sudo systemctl start auditd
[ec2-user@ip-172-31-44-223 ~]$ sudo systemctl status auditd
● auditd.service - Security Auditing Service
   Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled; vendor preset: enabled)
   Active: active (running) since Mit 2025-04-23 18:44:03 UTC; 17h ago
     Docs: man:auditd(8)
           https://github.com/linux-audit/audit-documentation
 Main PID: 2609 (auditd)
   CGroup: /system.slice/auditd.service
           └─2609 /sbin/auditd

Apr 23 18:44:03 localhost augenrules[2613]: backlog_wait_time 15000
Apr 23 18:44:03 localhost augenrules[2613]: enabled 1
Apr 23 18:44:03 localhost augenrules[2613]: failure 1
Apr 23 18:44:03 localhost augenrules[2613]: pid 2609
Apr 23 18:44:03 localhost augenrules[2613]: rate_limit 0
Apr 23 18:44:03 localhost augenrules[2613]: backlog_limit 8192
Apr 23 18:44:03 localhost augenrules[2613]: lost 0
Apr 23 18:44:03 localhost augenrules[2613]: backlog 1
Apr 23 18:44:03 localhost augenrules[2613]: backlog_wait_time 15000
Apr 23 18:44:03 localhost systemd[1]: Started Security Auditing Service.
[ec2-user@ip-172-31-44-223 ~]$
```

```
[ec2-user@ip-172-31-44-223 ~]$ sudo lsof -i -P -n | grep LISTEN
rpcbind  2645      rpc    8u  IPv4  15942      0t0  TCP *:111 (LISTEN)
rpcbind  2645      rpc   11u  IPv6  15945      0t0  TCP *:111 (LISTEN)
master   3092     root   13u  IPv4  18079      0t0  TCP 127.0.0.1:25 (LISTEN)
sshd     3522     root    3u  IPv4  77621      0t0  TCP *:22 (LISTEN)
sshd     3522     root    4u  IPv6  77630      0t0  TCP *:22 (LISTEN)
[ec2-user@ip-172-31-44-223 ~]$ sudo systemctl list-units --type=service --state=running
UNIT                            LOAD   ACTIVE SUB     DESCRIPTION
acpid.service                   loaded active running ACPI Event Daemon
amazon-ssm-agent.service        loaded active running amazon-ssm-agent
atd.service                     loaded active running Job spooling tools
auditd.service                  loaded active running Security Auditing Service
chronyd.service                 loaded active running NTP client/server
crond.service                   loaded active running Command Scheduler
dbus.service                    loaded active running D-Bus System Message Bus
fail2ban.service                loaded active running Fail2Ban Service
firewalld.service               loaded active running firewalld - dynamic firewall daemon
getty@tty1.service              loaded active running Getty on tty1
gssproxy.service                loaded active running GSSAPI Proxy Daemon
libstoragemgmt.service          loaded active running libstoragemgmt plug-in server daemon
lvm2-lvmetad.service            loaded active running LVM2 metadata daemon
network.service                 loaded active running LSB: Bring up/down networking
postfix.service                 loaded active running Postfix Mail Transport Agent
rngd.service                    loaded active running Hardware RNG Entropy Gatherer Daemon
rpcbind.service                 loaded active running RPC bind service
rsyslog.service                 loaded active running System Logging Service
serial-getty@ttyS0.service      loaded active running Serial Getty on ttyS0
sshd.service                    loaded active running OpenSSH server daemon
systemd-journald.service        loaded active running Journal Service
systemd-logind.service          loaded active running Login Service
systemd-udevd.service           loaded active running udev Kernel Device Manager

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

23 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

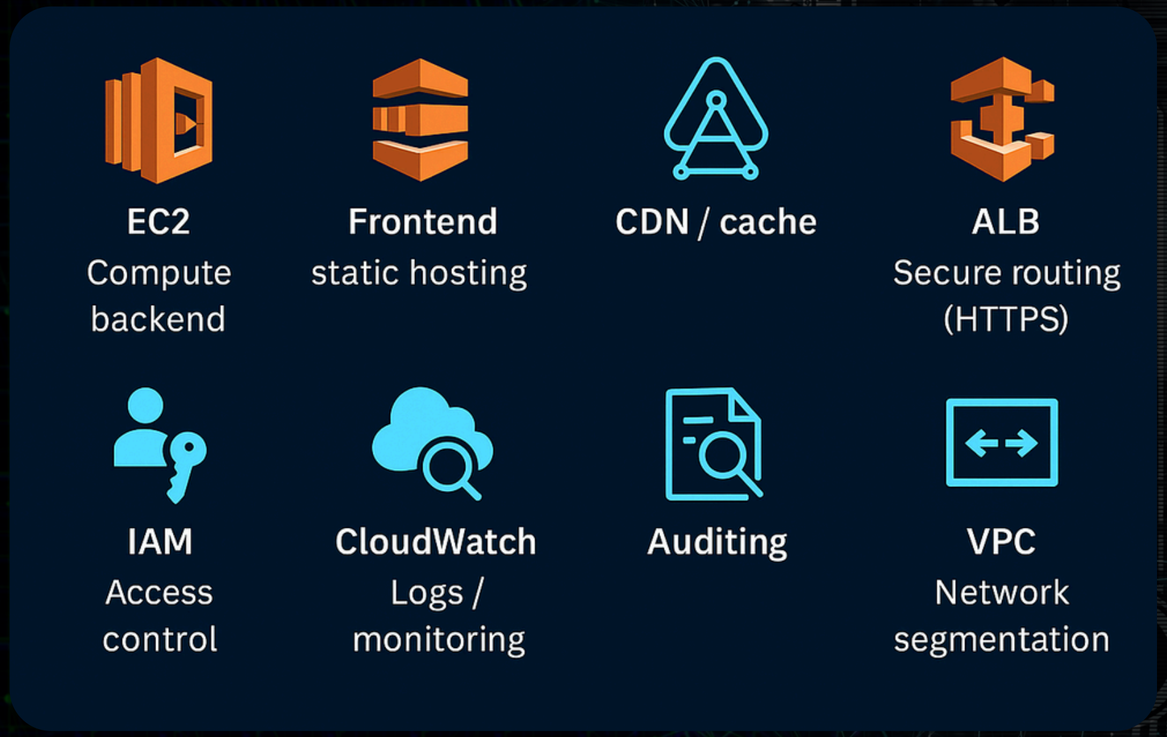
MONITORING & ALERTING







AWS TOOLS & SERVICES USED



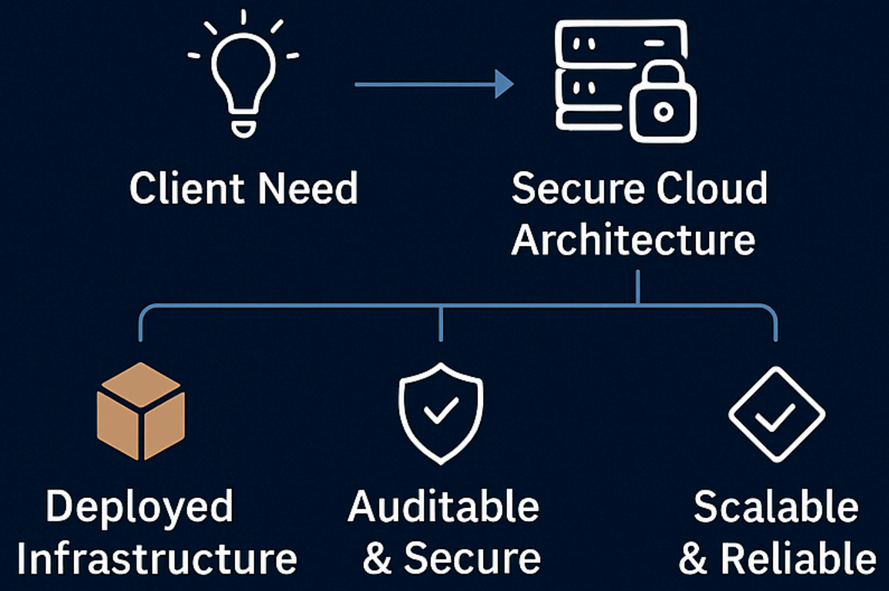
- Each service was selected and integrated according to its role on our architecture
- This ensures performance, security, and maintainability across all environments and workloads

BEST PRACTICES APPLIED

- Least Privilege IAM
- MFA enabled for all users
- Subnet isolation: public / private
- Centralized logging (CloudWatch)
- Full audit trail (CloudTrail)
- Environment tagging enforced
- Public access disabled by default

- Our architecture implements key aws best practices to ensure a secure, scalable, and auditable environment
- Resources are isolated by design, access is tightly controlled and every action is monitored and traceable

CONCLUSION



- ✓ Deployed Inrastructure
- ✓ Auditable & Secure
- ✓ Scalable & Reliable

- Cool Delivery now runs on a secure, scalable cloud platform.
- Architecture is isolated, monitored, and built to evolve.
- Our work enables safer deployments and faster iterations.