

420-414 - Infonuagique

Épreuve finale

Énoncé

Étudiant.e :

Objectif général

L'objectif de cette épreuve est de montrer votre capacité à déployer une infrastructure infonuagique multi-services, sécurisée et hautement disponible sur le Cloud AWS, à l'aide de **Docker**, **Docker Compose** et **AWS EC2**, en respectant les principes de modularité, de sécurité et de haute disponibilité. L'infrastructure devra être accessible via un nom de domaine configuré avec un proxy inverse (*Traefik*) assurant le chiffrement SSL/TLS (HTTPS).

L'épreuve est composée de deux parties, décrites dans les prochaines sections :

- **Partie I - Rapport d'implémentation**
- **Partie II - Démonstration de l'implémentation**

Partie I - Rapport d'implémentation

Mandat

Vous êtes mandaté pour concevoir et documenter une architecture infonuagique répondant aux besoins d'une infrastructure multi-services. Vous devez fournir un rapport technique complet, incluant :

- Les décisions de conception de votre infrastructure réseau sur AWS (VPC, sous-réseaux, sécurité...) ;
- La configuration de/des instances EC2 ;
- Le déploiement *Docker Compose* de vos services ;
- La configuration de *Traefik* comme proxy inverse avec HTTPS ;
- La stratégie de haute disponibilité ;
- (Optionnel) L'utilisation de Terraform pour déployer l'infrastructure.

Le rapport doit être accompagné de schémas (topologie réseau et architecture Docker). Un gabarit du rapport (*gabarit-rapport.docx*) ainsi que des exemples de schémas (*exemple-schemas.drawio.xml*) vous sont fournis avec cet énoncé.

Contraintes et exigences techniques

Infrastructure AWS

- Création d'un VPC personnalisé ;
- Configuration d'au moins deux sous-réseaux (privés et publics) ;
- Déploiement d'une instance EC2 ;
- Gestion des groupes de sécurité pour restreindre le trafic.

Docker / Docker Compose

- Déploiement d'au moins 3 services (**excluant *Traefik***) choisis dans une liste fournie par l'enseignant.
- Tous les services doivent être accessibles via un nom de domaine configuré (Duck DNS ou autre).

Traefik

- Configuration comme reverse proxy avec prise en charge de certificats SSL/TLS (HTTPS) ;
- Routage des services selon leur nom de domaine ou chemin.

Haute disponibilité

Mise en place d'une stratégie de tolérance aux pannes :

- Redémarrage automatique des conteneurs Docker en cas de crash ;
- (Optionnel) Réplication de l'instance EC2 sur plusieurs zones de disponibilité.

Documentation

Le rapport devra contenir :

- Schéma du VPC (zones, sous-réseaux, passerelles...)
- Schéma de l'architecture Docker / services
- Explications techniques pour :
 - Région et plage d'adresses IP du VPC ;
 - Zone de disponibilité et plage d'adresses IP des sous-réseaux ;
 - Configuration réseau (Passerelle internet, tables de routage...) ;
 - Groupes de sécurité (règles entrantes/sortantes justifiées) ;
 - Instance EC2 (type, AML, script d'initialisation ou méthode de configuration manuelle).
 - (Optionnel) Automation du déploiement de l'infrastructure avec Terraform et explication des avantages de l'utilisation d'un tel outil.

Partie II - Démonstration de l'implémentation

Format

Chaque étudiant devra présenter l'implémentation de son infrastructure lors d'une séance dédiée de 15 minutes. Cette démonstration pourra se faire :

- En classe lors de la dernière séance de cours, ou
- Sur rendez-vous individuel pendant la semaine ESO.

Dans les deux cas, la date et l'heure de la démonstration devra être fixée avec votre enseignant **avant le mardi 13 mai 2025**.

Objectifs de la démonstration

Le but de cette présentation est de valider la cohérence entre le rapport d'implémentation (partie I) et le déploiement réel de la solution, ainsi que votre compréhension des outils infonuagiques utilisés.

Durant la démonstration, vous devez être en mesure de :

- Expliquer vos choix d'implémentation (architecture réseau, services choisis, configuration Docker, etc.) ;
- Montrer le bon fonctionnement de votre infrastructure en direct :
 - Connexion à l'instance EC2 (SSH) ;
 - Exécution de commandes Docker (ex. `docker ps`, `docker logs`, etc.) ;
 - Affichage de la réponse des services déployés via leur URL (HTTPS).
- Expliquer la méthode utilisée pour configurer votre instance EC2 : script d'initialisation (*user data*) ou configuration manuelle ;
- Parcourir et commenter votre fichier `docker-compose.yml`
- Expliquer brièvement vos groupes de sécurité, votre stratégie de haute disponibilité, votre schéma du VPC et de l'architecture Docker.

Critères d'évaluation

- Fonctionnalité de l'infrastructure (accès aux services, proxy fonctionnel, HTTPS) ;
- Qualité de l'explication (clarté, maîtrise du sujet) ;
- Cohérence entre la démonstration et le rapport ;
- Capacité à résoudre ou commenter les éventuels problèmes rencontrés.

Grilles d'évaluation

Partie I - Rapport

Critères	Éléments évalués	Points
Structure et clarté du rapport	<i>Respect du gabarit fourni (titres, sections, organisation claire).</i>	1.0
	<i>Langage technique approprié, orthographe, lisibilité.</i>	3.0
Architecture réseau (VPC)	<i>Présence d'un schéma clair du VPC (zones, sous-réseaux, passerelle, etc.).</i>	2.0
	<i>Plages d'adresses, région, sous-réseau publics/privés, zones de disponibilité, tables de routage bien définis et justifiés.</i>	3.0
	<i>Groupes de sécurité bien expliqués (règles entrantes/sortantes).</i>	2.0
Instance EC2	<i>Description des caractéristiques techniques et de la stratégie de configuration de l'instance EC2.</i>	3.0
Déploiement des services	<i>Choix d'au moins 3 services et description claire de leur fonction.</i>	1.5
	<i>Présence d'un schéma clair de l'architecture Docker</i>	2.0
	<i>Présentation claire du fichier <code>docker-compose.yml</code> (volumes, réseaux, ports etc.).</i>	3.0
Proxy inverse et HTTPS	<i>Présentation du proxy inverse (routes, certificats SSL/TLS).</i>	1.0
	<i>Présentation de la méthode de configuration du nom de domaine (Duck DNS ou autre)</i>	0.5
Haute disponibilité	<i>Présentation d'une stratégie de tolérance aux pannes (restarts, AZs etc.)</i>	1.0
Cohérence / réalisme de la solution	<i>Cohérence globale, faisabilité sur AWS, intégration des composants</i>	2.0
laC (optionnel)	<i>Utilisation de Terraform et explication des avantages de l'outil (max 2 points supplémentaires)</i>	
Total des points		25.0

Partie II - Démonstration de l'implémentation

Critères	Éléments évalués	Points
Fonctionnalité de l'infrastructure	Le déploiement Docker est fonctionnel (<i>docker ps</i> , logs, etc.) et au moins 3 services sont déployés (excluant Traefik)	3.0
	Traefik est bien configuré comme reverse proxy avec SSL/TLS	1.5
	Tous les services sont accessibles via HTTPS avec noms de domaine.	0.5
Explication technique	Schéma VPC expliqué clairement (CIDR, sous-réseaux, IGW, etc.)	1.0
	Justification des groupes de sécurité (ports, trafic autorisé)	0.5
	Explication du fichier <i>docker-compose.yml</i> (services, volumes, réseaux, etc.)	1.5
	Présentation de la stratégie de configuration de l'instance (user-data/configuration manuelle)	1.5
	Stratégie de haute disponibilité expliquée (restarts, AZs...)	0.5
Cohérence avec le rapport	Le déploiement est conforme à ce qui est documenté	1.0
Maîtrise globale et présentation	Réponses claires aux questions, bonne compréhension technique, capacité à commenter les problèmes rencontrés et solutions apportées	4.0
Total des points		15