

---

# Projets

## Modalités projets pour le module “Infrastructure Cloud”

Damien Montmoulinex - Toulouse Ynov Campus 2024/2025  
Master Cloud, Sécurité & Infrastructure / DevOps

Module Infrastructure Cloud

---

# Modalités générales

# Calendrier et organisation

---

- Groupes de 2 à 3 personnes (évaluations par compétences, sur les compétences mise en oeuvre dans le projet choisi)
- 3 journées complètes + temps personnel
- Calendrier
  - **16 juin** - Présentation des sujets
  - **19 juin** - Choix et validation des sujets + constitutions des groupes terminés
  - **4 juillet** - Présentation du sujet, des résultats et de vos apprentissages au reste du groupe (20 à 30 minutes + questions)
  - **7 juillet au plus tard** - Dépôt des projets finaux sous Moodle
- Comme pour les TPs le rendu de votre code source se fait au format Git avec des fichiers Markdown pour la documentation et les explications

# Démarche

---

- Les sujets proposés sont donnés à titre indicatif, il vous est possible de vous en inspirer pour réaliser d'autres projets en lien avec le module
- L'objectif est de tester et apprendre sur de nouveaux concepts / nouvelles technologies. L'apprentissage et le travail régulier est aussi important que le résultat final
- Utiliser git et versionner votre travail régulièrement. Le fait d'échouer et recommencer fait partie intégrante des projets. Votre repository Git et vos notes dans les fichiers Markdown devront être représentatif des différentes tentatives que vous avez mises en oeuvre

# Démarche

---

- Vous utilisez les outils et fournisseurs de Cloud présentés en cours ou les équivalents de votre choix
- Avant de démarrer vous mettez en place un plan d'action que vous suivrez ensuite. Ce plan d'action initial doit être clairement identifiable dans un fichier PLAN.md
- Le plan d'action peut être amené à modification au fur et à mesure de l'avancement du projet

---

# Propositions de sujets

# Sujet n°1 - Scalabilité

---

## Objectifs :

- Mettre en place un système qui se met à l'échelle automatiquement (création et suppression de serveurs) en fonction de la charge que doit traiter le système
- Simuler de la charge sur ce système
- Observer (à l'aide d'un outil de monitoring - CloudWatch, Grafana, ELK) la montée en charge et l'absorption de cette dernière
- Vous pouvez également utiliser une métrique “métier” plutôt que la charge du système gérer le scaling
- Votre infrastructure se déploie entièrement via de l'Infrastructure as Code (Terraform, Ansible, ...)

# Sujet n°2 - OpenStack

---

## Objectifs :

- Réaliser une maquette permettant l'utilisation de services cloud basiques (Compute, Network, Object Storage, Identity & Access) avec OpenStack
- A l'aide du provider OpenStack de Terraform, créer une infrastructure simple (Serveur, Equilibreur de charge, Base de données) dans votre Cloud OpenStack

# Sujet n°3 - CozyCloud (ou équivalent)

---

## Objectifs :

- Héberger un Cozy Cloud avec le module OnlyOffice
- Utiliser Terraform (pour automatiser la création de l'infrastructure) et Ansible (pour automatiser l'installation et la configuration de Cozy Cloud)
- Être capable de se connecter à son Cozy Cloud avec une application mobile

# Sujet n°4 - Mise en place d'un workflow GitOps

---

## Objectifs :

- Mettre en place un workflow GitOps comprenant :
  - De l'analyse statique et des tests sur le code
  - De l'analyse et des prédictions tarifaires (FinOps)
  - Un système de release de l'infrastructure avec une stratégie de semversion
  - Des automatisations pour déployer et rollback vos versions d'infrastructures
- Créer un scénario avec une infrastructure minimale et évolutive afin de tester et pouvoir présenter votre workflow

# Sujet n°5 - Multi-cloud

---

## Objectifs :

- Développer une Command Line Interface permettant la gestion et la manipulation de différents services (Compute, Stockage objet, Network, Base de données managées) sur au moins deux Cloud Providers parmi AWS, GCP, Scaleway et Microsoft Azure
- ou
- Développer des modules Terraform permettant de créer des infrastructures identiques (Load Balancer, Serveurs, Bases de données) sur plusieurs Cloud Providers en même temps

# Sujet n°6 - Usine de création d'AMI

---

- Automatiser la création d'AMI sur le Cloud AWS
- Utiliser Ansible, Packer et Github Actions par exemple
- Le principe :
  - Vous concevez deux besoins de serveur différents (ex: un serveur Web + Applicatif et un serveur base de données + gestion des backups)
  - Grâce à Ansible, le contenu d'une machine type (web ou bdd) est défini avec du code. Certaines variables, ou rôles Ansible sont utilisée dans les deux type de machine.
  - Lors d'une modification du code dans git (via Github ou Gitlab), les AMIs sont automatiquement créé à l'aide des playbook Ansible et de Packer

# Sujet n°7 - Dashboard as Code

---

- Déployer Grafana et Prometheus
- Créer une infrastructure de type Stack LAMP
- Utiliser grafonnet pour industrialiser la création de Dashboard de monitoring dans grafana

# Sujet n°8 - IAM

---

- Créer une ou des intégrations IAM, exemples :
  - Intégration IAM entre Hyper Planning et AWS
  - Gestion des droits IAM depuis Kubernetes (en lien avec le module conteneur)
  - Imaginer une organisation et une gestion des droits
  - Lier un LDAP avec AWS IAM. Automatiser la création des Users, la gestion des groupes, l'ajout de droits en lien entre LDAP et IAM ...

# Questions / Remarques

---