

# Djiby Oumar SALL

## Ingénieur DevOps

5 ans d'expérience professionnelle

Certifié Azure / Terraform / CCNA 1&2

### COMPETENCES TECHNIQUES

<b>CI/CD:</b>	Jenkins, GitHub Actions, GitLab, Kubernetes, GitOps (FluxCD), Helm
<b>Conteneur &amp; Orchestrateur :</b>	Kubernetes, Docker, Docker Compose, Helm Charts, FluxCD
<b>Infrastructure as Code :</b>	Terraform, CloudFormation, Ansible
<b>Cloud :</b>	Azure (ACR, AKS, Azure Storage Account, Azure Static Web Apps, Azure Functions), AWS (EC2, ECS, S3, Amplify, Lambda, VPC, EKS, ECR, CodePipeline ...), VMware vRealize (VRA, VRO, VRops)
<b>Versioning :</b>	GitLab, GitHub
<b>Monitoring :</b>	Checkly, Prometheus, Grafana, LogInsight, vROPS, Centreon, playwright
<b>Système :</b>	Windows, Linux
<b>Méthologies :</b>	Agile SCRUM, Kanban, SAFe
<b>Outils de Gestion de projet :</b>	Jira, Confluence, Trello
<b>Langages de programmation :</b>	Python, JavaScript/TypeScript, C#, HTML/CSS, Bash, SQL, YAML, JSON

### CERTIFICATIONS, DIPLÔMES ET LANGUES

#### Certifications :

<b>2023</b>	HashiCorp Certified : Terraform Associate (003)
<b>2023</b>	Microsoft Certified : Azure Fundamentals (AZ- 900)
<b>2023</b>	CCNA R&S : INTRODUCTION TO NETWORKS (CCNA 1)
<b>2023</b>	CCNA Routing and Switching : Routing and Switching Essentials (CCNA 2)
<b>2017</b>	Certificat of Completion IT Essentials CISCO

#### Diplômes :

<b>2023</b>	Master – Expert en Architecture des systèmes d'informations à INTECH <i>Mémoire : l'automatisation des tests de qualité des applications</i>
<b>2021</b>	Bachelor – Systèmes et Réseaux Informatiques à INTECH
<b>2018</b>	Licence professionnelle en réseau informatique – Systèmes et Réseaux à ISI
<b>2017</b>	DTS – Maintenance Informatique à ISI

#### Anglais Technique

## EXPERIENCES PROFESSIONNELLES

SACEM

novembre 2025 à février 2026

**Projet :** Refonte et modernisation de la plateforme Météo DSI - Migration d'Excel VBA vers application web centralisée

**Rôle :** Ingénieur DevOps / Développeur Full Stack

**Contexte :** Transformation complète d'un outil de supervision critique basé sur Excel avec macros VBA vers une plateforme web moderne et scalable. L'ancienne solution, dépendante d'un poste Windows, collectait manuellement les données depuis Dollar U (ordonnancement), Prometheus (disponibilité applications), Centreon (supervision système) et NetBackup (sauvegardes) pour générer des rapports quotidiens. Le nouveau système centralise ces données dans une architecture haute disponibilité sur infrastructure on-premise vSphere/RHEL9, avec authentification LDAP, interface web responsive, API REST, et base de données PostgreSQL pour l'historisation.

**Réalisations :**

- Conception d'une architecture microservices haute disponibilité sur 2 VMs RHEL 9.4 (Backend : 8 vCPU, 16GB RAM / Frontend : 4 vCPU, 8GB RAM) avec séparation frontend/backend et sécurisation réseau (VLAN DMZ, firewalld).
- Développement d'une API REST Node.js/Express avec architecture modulaire (routes, controllers, models, services) et middleware d'authentification JWT/LDAP.
- Conception et implémentation du modèle de données PostgreSQL 15 avec gestion d'historique, relations complexes et migrations automatisées.
- Création de collecteurs de données automatisés pour intégration multi-sources (Prometheus, Dollar U Oracle, Centreon, NetBackup API, EasyVista ITSM) avec gestion erreurs et retry logic.
- Implémentation de schedulers pour collectes automatiques, génération de rapports quotidiens à 7h00, et envoi d'emails via SMTP Exchange.
- Développement d'une interface web moderne en React 18 + TypeScript avec Vite, Tailwind CSS, et composants réutilisables (React Query, Chart.js).
- Implémentation d'un système d'authentification LDAP/Active Directory avec gestion des rôles (Admin, Editor, Viewer) et protection des routes.
- Création de tableaux de bord dynamiques avec visualisation temps réel des statuts de 59 sondes applicatives, indicateurs métiers, et graphiques interactifs.
- Mise en place de Nginx 1.30.1 comme reverse proxy avec configuration TLS 1.3, HTTP/2, et résolution de problèmes critiques de certificats SSL (BOM UTF-8) sur 6 serveurs (Prometheus, Grafana).
- Déploiement automatisé avec pipelines CI/CD, scripts PowerShell/SSH, services systemd pour gestion des processus, et stratégies zero-downtime avec health checks.
- Intégration monitoring avec Prometheus Node Exporter et Grafana pour métriques système, applicatives, et alerting proactif.
- Sécurisation complète de la chaîne applicative : HTTPS obligatoire, tokens JWT, protection CSRF/XSS, Fail2Ban, rate limiting API, et logs centralisés.
- Documentation technique complète : architecture système, guides d'installation, procédures de déploiement, API documentation (OpenAPI/Swagger), et runbooks opérationnels.

**Environnement Technique :** Node.js, Express, TypeScript, React 18, Vite, Tailwind CSS, PostgreSQL 15, Redis, Nginx, RHEL 9.4, vSphere, Git, GitLab CI/CD, Prometheus, Grafana, Checkly, EasyVista API, Oracle, Centreon, NetBackup API, LDAP/Active Directory, JWT, TLS/SSL, SELinux, Firewalld, Systemd

**Projet :** Modernisation et déploiement d'une plateforme e-gouvernement multi-services sur AWS

**Rôle :** Ingénieur DevOps / Infrastructure

**Contexte :** Conception et déploiement d'une infrastructure cloud AWS complète pour héberger une suite d'applications e-gouvernement (e-visa, e-timbre, e-vignette, e-enregistrement, e-TVA) destinées à la digitalisation des services publics. Responsable de l'ensemble de l'infrastructure, du déploiement automatisé et de la gestion des environnements de développement et production.

**Réalisations :**

- **Architecture cloud AWS :** Conception d'une infrastructure multi-services avec ECS Fargate, RDS PostgreSQL, S3, et ALB pour supporter des applications React/Node.js à haute disponibilité
- **Infrastructure as Code :** Développement de modules Terraform réutilisables pour le provisioning automatisé des ressources AWS (VPC, subnets, security groups, ECS, RDS, S3)
- **Gestion des secrets et sécurité :** Implémentation d'AWS Secrets Manager et Parameter Store pour la gestion sécurisée des credentials et variables d'environnement
- **Pipelines CI/CD :** Configuration de pipelines CodePipeline/CodeBuild avec déploiement automatisé via GitHub, incluant build Docker et déploiement ECS
- **Gestion multi-environnements :** Déploiement et maintenance des environnements UAT et PROD avec isolation réseau et gestion différenciée des ressources
- **Stockage documentaire :** Intégration S3 pour la gestion des documents administratifs avec configuration CORS et politiques d'accès granulaires
- **Monitoring et troubleshooting :** Résolution de problèmes de connectivité réseau, optimisation des security groups et configuration des accès inter-services
- **Base de données :** Gestion d'instances RDS PostgreSQL multi-bases avec backup automatisé et gestion des accès sécurisés
- **Application de gestion documentaire :** Déploiement d'une solution complète de gestion documentaire et de scénarios métier pour optimiser les processus administratifs
- **Support et maintenance :** Assurance de la disponibilité 24/7 des services, gestion des incidents et optimisation continue des performances

**Environnement Technique :** AWS (ECS Fargate, RDS PostgreSQL, S3, Secrets Manager, CodePipeline, CodeBuild, ALB, VPC), Terraform, Docker, React.js, Node.js, GitHub Actions, Linux, Bash

**Projet :** Mise en place d'une plateforme Kubernetes automatisée avec GitOps (vSphere)

**Rôle :** Ingénieur DevOps

**Contexte :** Déploiement d'une plateforme Kubernetes clé-en-main, entièrement automatisée et industrialisée, sur une infrastructure on-premise vSphere. Ce projet a pour objectif de simplifier la mise à disposition d'environnements de développement et de production, en adoptant les pratiques DevOps et GitOps avec des outils modernes d'Infrastructure as Code (IaC), le tout orchestré via GitLab CI/CD.

**Réalisations :**

- Conception d'une chaîne CI/CD GitLab pour le provisioning infra, le déploiement RKE2, et la livraison applicative via FluxCD.

- Conception de modules Terraform pour le provisionnement de l'infrastructure (VMs, réseaux, stockage) sur vSphere.
- Déploiement automatisé de clusters Kubernetes RKE2 à l'aide de playbooks Ansible, incluant installation, configuration réseau, et démarrage du service RKE2.
- Mise en place du bootstrap GitOps avec FluxCD pour gérer le déploiement des applications et composants d'infrastructure depuis un dépôt Git.
- Utilisation de Kustomize pour la personnalisation des manifestes Kubernetes selon les environnements (DEV, STAGING, PROD).
- Déploiement d'applications complexes avec Helm et gestion via HelmRelease dans FluxCD.
- Intégration de pipelines GitLab CI pour automatiser l'ensemble du cycle de vie (lint, plan, apply, destroy).
- Contrôle qualité et sécurité du code IaC avec TFLint et Checkov intégrés dans les pipelines CI.
- Structuration du dépôt GitOps multi-clusters (apps/, infrastructure/, clusters/) pour assurer la répliquabilité et la scalabilité du modèle.
- Mise en place de la supervision des clusters avec Prometheus et Grafana.
- Documentation technique complète des modules, flux CI/CD, procédures d'installation, gestion des erreurs et bonnes pratiques partagées avec les équipes.

**Environnement Technique :** Jira, Terraform, Ansible, GitLab CI/CD, Kubernetes (RKE2), vSphere, FluxCD, Kustomize, Helm, Prometheus, Grafana, Kyverno, Checkov, TFLint, Docker, Linux

**SACEM**

**Mai 2024 à Aujourd'hui**

**Rôle :** Ingénieur DevOps – Consultant en Exploitation et Supervision IT

**Contexte :** Mission en régie au sein de l'équipe Exploitation & Supervision en collaboration avec l'équipe Middleware. Interventions à la fois sur la partie RUN (exploitation quotidienne) et BUILD (projets d'automatisation et d'évolution de la supervision). Objectif : garantir la continuité des services IT, refondre la plateforme de supervision, industrialiser les processus d'exploitation, et fiabiliser la supervision applicative interne et externe.

**Réalisations :**

- **RUN**
  - Réalisation quotidienne de la **météo des services** informatiques à 7h15 avec détection proactive des incidents.
  - **Supervision via Grafana** (tableaux de bord Prometheus), confirmation des alertes par tests manuels, redémarrage applicatif ou création de ticket selon les plages horaires.
  - **Gestion des tickets** d'exploitation (ISupport) : sauvegardes, restaurations, certificats SSL (GlobalSign), snapshots, extension disque/RAM, traitement Fin de Vie des VMs.
  - Création et maintenance de **VMs via vCenter** (Centreon, agents Checkly).
- **BUILD :**
  - Utilisation de Terraform pour préparer le provisionnement de l'infrastructure sur Azure.
  - **Migration des sondes HpOvis vers Checkly** : développement de scénarios Playwright pour la supervision applicative (authentification, navigation, API), automatisation du déploiement via GitLab CI/CD, et configuration des fenêtres de maintenance.

- **Ateliers de formation** organisés auprès des équipes métiers pour faciliter la prise en main de Checkly et la création autonome de scénarios E2E.
- Développement d'un **Playwright Exporter Prometheus** interne pour simuler des tests E2E et exposer les résultats via un endpoint HTTP.
- **Automatisation de la configuration Prometheus** via Ansible et GitLab CI, avec structuration des `scrape_configs` par environnement et équipe.
- Mise en place de la **fédération Prometheus** pour répliquer les données Checkly de production vers les environnements de test et recette.
- Industrialisation du **provisionnement des dashboards Grafana** via Terraform et GitLab CI (multi-environnements), avec versioning Git et scripts Python pour gestion via l'API Grafana.
- Rédaction de **guides d'exploitation** (API Checkly, déploiement GitLab, LDAP/Grafana) et maintien du **patrimoine documentaire**.

**Environnement Technique :** JavaScript, TypeScript, Python, Prometheus, Grafana, Checkly, Playwright, GitLab CI/CD, Terraform, Ansible, Linux (RHEL/Debian), vCenter, Keeper, ISupport, GlobalSign, Docker, Kubernetes, Infrastructure as Code (IaC), Jira

**TEOLIA**

**Janvier 2024 à Avril 2024**

**Rôle : Ingénieur DevOps**

**Contexte :** Prise en charge d'un projet de **platform engineering** visant à simplifier et automatiser le déploiement et la gestion des ressources d'infrastructure Cloud. L'objectif est de permettre aux équipes de développement et aux consultants de faire des demandes automatisées de ressources d'infrastructure. Cette plateforme facilite le déploiement des applications pour les développeurs et fournit une ressource dédiée pour les consultants qui passent leur certification. De plus, elle offre une meilleure visibilité sur l'utilisation des ressources Cloud, ce qui simplifie la gestion des coûts.

**Réalisations :**

- **Développement Frontend :**
  - Utilisation des langages HTML, CSS et JavaScript pour concevoir une interface utilisateur dynamique et conviviale.
  - Structuration des pages frontend avec HTML pour une organisation optimale.
  - Application de styles visuels avec CSS pour améliorer l'expérience utilisateur.
  - Utilisation de JavaScript pour manipuler le DOM et permettre des interactions côté client.
- **Provisionnement avec Terraform :**
  - Utilisation de Terraform pour préparer le provisionnement de l'infrastructure sur Azure.
  - Description détaillée de l'infrastructure à déployer sur Azure à l'aide de Terraform.
  - Configuration du Remote Backend de Terraform sur Azure Blob Storage pour assurer une gestion centralisée et sécurisée des états.
- **Déploiement Frontend sur Azure Static Web Apps :**
  - Mise en œuvre du déploiement automatisé du frontend sur Azure Static Web Apps pour garantir des performances optimales et une haute disponibilité.

- **Gestion de version et workflows avec GitHub Actions :**
  - Utilisation de Git pour le versioning des fichiers du projet, assurant ainsi un suivi précis des modifications apportées au code source.
  - Utilisation de GitHub Actions pour la création des pipelines permettant d'implémenter une CI/CD de bout en bout du projet
- **Authentification SSO :**
  - Implémentation de protocoles d'authentification, notamment OpenID Connect, pour garantir la sécurité des accès à la plateforme.
- **Azure Function Backend :**
  - Développement et déploiement de différentes fonctions Azure :
    - **deploy\_ressource** : Déploiement automatisé des ressources sur Azure déclenché par un workflow GitHub Action.
    - **postmessage** : Envoi de messages de demande de ressources sur Microsoft Teams via un Incoming Webhook.
    - **sendmail** : Envoi de notifications par e-mail en cas de refus de demande de ressources.
- **Migration Azure vers AWS :**
  - Déploiement Frontend sur **AWS Amplify**
    - Mise en œuvre du déploiement continue du frontend en utilisant GitHub.
  - Développement et déploiement du fonction **AWS Lambda**
    - **post\_message** : Envoi de message de demande de ressources sur Microsoft Teams via un Incoming Webhook.
  - Utilisation d'**AWS API Gateway** pour créer des endpoints HTTP sécurisés permettant de déclencher les fonctions Lambda.
  - Utilisation de Terraform pour provisionner de l'infrastructure sur AWS (**un VPC, des sous-réseaux, une passerelle Internet, une instance EC2 et un groupe de sécurité**) et la configuration des routes, des groupes de sécurité et des paramètres de sous-réseau

**Environnement Technique :** Jira, HTML, CSS, JavaScript, Terraform, Azure Static Web Apps, GitHub Actions, Azure Functions, AWS Amplify, AWS Lambda, AWS API Gateway, CloudFormation

**TEOLIA**

**Septembre 2023 à Novembre 2023**

**Rôle :** Ingénieur DevOps au sein de la Teo School

**Contexte :** La Practice DevOps de Teolia met en place un **programme de formation personnalisé sur deux mois**, la Teo School, cette school a pour objectif d'accompagner ses membres à **monter en compétences et résoudre des problématiques techniques diverses**. Le projet permet aux apprenants de réaliser une usine logicielle complète permettant à chaque ingénieur DevOps de toucher l'ensemble des outils et technologies DevOps, de la **planification au monitoring en (et hors) production**.

**Projet :** Mise en place d'une **chaîne CI/CD** complète opérant sur un projet au sein de Teolia. Déployé de bout en bout, celle-ci permet le **build, testing, delivery** et **déploiement** d'une application dans un environnement de test ainsi qu'en production.

## Réalisations :

- **Gestion de projets et documentation :**

- Utilisation de Jira pour la gestion de projet & Confluence pour la documentation.
- Choix de Kanban comme méthodologie agile.
- Utilisation de GitHub comme gestionnaire de repository

- **Terraform :**

- Provisionnement de l'infrastructure Azure avec Terraform :
  - Configuration du Remote Backend de Terraform sur Azure BLOB Storage.
  - Provisionnement d'un Cluster Kubernetes AKS sur Azure, de Azure Container Registry (ACR) et Azure Storage Account avec les modules de Terraform.

- **Azure :**

- Azure kubernetes Services (AKS)
  - Déploiement des micro-services de l'application avec les manifests kubernetes.
  - Gestion du cluster, des nœuds, des pods, des services et des ingress
  - Implémentation de l'outil ArgoCD
  - Implémentation de l'autoscaling (via des HPA)
- Gestion Stockage, de la haute Disponibilité et de la Résilience (ACR)
- Stockage des Images sur Azure Container Registry

- **Docker :**

- Création des Dockerfiles : Conception et écriture des fichiers Dockerfiles pour permettre la création d'images Docker optimisées et fonctionnelles.
- Optimisation des Dockerfiles : Révision et amélioration des Dockerfiles existants afin d'optimiser les performances, réduire la taille des images et garantir la sécurité.
- Analyse avec Trivy : Utilisation de l'outil Trivy pour analyser les images Docker et détecter les vulnérabilités de sécurité potentielles.

- **GitHub :**

- Branche « main » : déployer l'infrastructure, images docker et micro-services en prod.
- Gestion des versions et du déploiement
- GitHub Actions :
  - Utilisation de GitHub Actions pour mettre en place des pipelines de CI/CD complets, couvrant l'ensemble du cycle de vie du projet, depuis le développement jusqu'à la mise en production. Ces pipelines automatisent les processus de build, de test, de déploiement et de surveillance, garantissant ainsi une intégration et un déploiement continu de bout en bout.

- **Logging & Monitoring :**

- Déploiement et configuration d'outils de monitoring (Prometheus, Grafana) avec Helm.

**Environnement Technique :** Confluence, Jira, Kanban, GitHub, GitHub Actions, Docker, Kubernetes, Terraform, ACR, AKS, Prometheus, Grafana, Python, Trivy, ArgoCD.

**EDF SA (Électricité de France)**

**Septembre 2020 à septembre 2023**

**Rôle : Ingénieur DevOps**

**Entité :** DSIT / IT-O/ ICS / myCloud

**Contexte et enjeux :** MyCloud est un portail Cloud privée de EDF mis en place par la DSIT afin de mettre à disposition des utilisateurs un large panel de Services Cloud XaaS. Ses offres sont les suivantes : VM, Load Balancer, DNS, CaaS, DBaaS. Pour garantir une haute disponibilité et améliorer la qualité du service, j'ai mis en place des tests automatisés pour valider la fiabilité des workflows, la conception et la publication des catalogues items, ainsi que du monitoring dans la chaîne de production pour avoir une vue d'ensemble du fonctionnement des différents services du portail myCloud.

**Taille et composition de l'équipe :** 9 personnes en interne et une équipe d'infogérance de 8 personnes à Rennes, soit un total de 17 personnes composées d'experts, d'ingénieurs DevOps, de développeurs, d'un pilote de service et Scrum master.

**Méthodologie de l'équipe :** SAFe

**Réalisations :**

- Gestion de projets et documentation :
  - Utilisation de **Jira** pour la gestion du projet & **Confluence** pour la documentation.
  - Utilisation de **GitLab** comme gestionnaire de repository
  - Transfert de connaissances.
- Mise en place et Automatisation des tests du control de la qualité des services provisionner :
  - Implémentation de tests unitaires au niveau de vRealize Orchestrator (**vRO**) pour la validation des workflows.
    - Création de workflows de test vRO avec **JavaScript**.
    - Écriture de fonctions de test en Python à l'aide de la bibliothèque **pytest**
    - Mise en œuvre de **pipelines Jenkins** pour automatiser le lancement des tests
  - Développement d'un processus d'automatisation des tests pour vérifier la conformité des machines virtuelles (VMs).
    - Écriture de modèles de déploiement Terraform décrivant les configurations des VMs à déployer.
    - Utilisation de paramètres Terraform pour récupérer les informations sur les VMs déployées, telles que les adresses IP, les noms d'hôtes pour les utiliser dans les tests automatisés.
    - Changement de mots de passe linux avec ansible
    - Mise en place de tests automatisés à l'aide de **Testinfra** pour vérifier la configuration des VMs déployées, y compris les paramètres réseau, les fichiers de configuration système.
    - Configuration d'un pipeline Jenkins pour déclencher le déploiement et les tests automatisés.
  - Automatisation des tests de l'interface utilisateur (IHM) pour valider les catalogues de service dans VMware vRealize Automation (vRA).
    - Création de deux classes Workspace et ServiceBroker pour gérer l'authentification et la navigation dans l'interface de VMware vRealize Automation (vRA).
    - Utilisation de Selenium WebDriver pour interagir avec les éléments de l'IHM et effectuer des vérifications fonctionnelles.
  - Mise en place d'un pipeline Code Stream permettant de relancer tous les tests unitaires, d'intégration et end-to-end lors d'une mise à jour de la plateforme vRealize Automation (vRA).
- Conception et publication de catalog item dans **vRA** pour fournir des services Cloud aux utilisateurs finaux.
  - Création de blueprints et de catalogues items pour des déploiements de machines virtuelles.

- Logging & Monitoring :
  - Mise en place des indicateurs de pilotage sur **LogInsight**
  - Automatisation de reporting hebdomadaire sur les déploiements via le portail MyCloud (Selenium, Jenkins)
  - Mise en place des indicateurs de supervision via **vROPS**
- Maintien en état fonctionnel des pipelines **Jenkins** existants : mise à jour des agents, mise à jour du code, test de pipelines sur les différents environnements

**Environnement Technique** : Confluence, JIRA, Terraform, Testinfra, Jenkins, GitLab, VRA, VRO, vROPS, LogInsight, Python, Pytest/Selenium

## Carlili

**Novembre 2019 à Juillet 2023**

**Rôle** : Chargé des opérations

**Entité** : Team Opérationnel

**Contexte** : Carlili est une startup qui offre des services de location de voitures clé en main et promeut la mobilité individuelle à faible émission de carbone. L'objectif est de faire livrer et récupérer des véhicules directement à domicile ou adresse du client.

**Projet** : En tant qu'agent OPS (Opérationnel), j'ai la charge d'interagir avec les Carsitters sur le terrain et de travailler en étroite collaboration avec l'équipe du service client pour résoudre rapidement toutes les problématiques clients (incluant les réservations de véhicules). Cela implique la capacité à gérer des situations complexes avec diplomatie, de travailler sous pression et d'être à l'écoute des Carsitters et des clients.

**Taille et composition de l'équipe** : 8 personnes, un manager et un adjoint

**Méthodologie de l'équipe** : Ticketing

**Réalisations** :

- Coordonner les missions quotidiennes de la communauté des carsitters
- Gérer la flotte automobile de Carlili et des agences partenaires
- Gestion des appels Carsitters et clients
- Comptabiliser les problèmes

**Environnement Technique** : Slack, Tookan, Hippo Chat, Opérations back office, Aircall, record360

## Expresso / Sénégal

**Novembre 2017 - Décembre 2017**

**Rôle** : Ingénieure réseau et systèmes

**Entité** : Service IT

**Contexte** : Expresso Sénégal est le 3ème opérateur Télécom au Sénégal qui est une filiale du groupe Expresso Telecom (Expresso Telecom Group) et dont le siège est à Dubai.

**Projet** : Maintenance et entretien annuel du matériel informatique d'Expresso Sénégal

**Taille et composition de l'équipe :** 6 personnes dont 2 responsables

**Méthodologie de l'équipe :** Ticketing

**Réalisations :**

- Maintenance du matériel Informatique
- Entretien des équipements
- Installation et configuration des applications de contrôle et de configuration à distance

**Environnement Technique :** Windows, MacOS, Linux, TeamViewer, mise à jour de logiciel, installations antivirus, débogueur de protocole, CCleaner

**ISI (Institut Supérieur d'Informatique)**

**Août 2017 - Octobre 2017**

**Rôle :** Ingénieur systèmes réseaux

**Entité :** Service Informatique

**Contexte :** ISI (Institut Supérieur d'Informatique), est sous la tutelle du ministère de l'Enseignement Supérieur privé qui est signataire de tous les diplômes délivrés, sous le contrôle de l'ANAQ-Sup.

**Projet :** Au sein de l'ISI (Institut Supérieur d'Informatique), des besoins informatiques émergent et il est nécessaire d'accompagner l'institut dans la mise en place de nouvelles salles informatiques et maintenance préventive de tous les équipements. Aussi, d'accompagner les étudiants sur leurs projets/outils informatiques ainsi qu'intervenir sur la résolution de problèmes logiciels et matériels en cas de pannes de réseau

**Taille et composition de l'équipe :** 3 personnes dont 1 responsable de service

**Méthodologie de l'équipe :** Ticketing

**Réalisations :**

- Assister et accompagner les étudiants dans leurs projets
- Installation, configuration de différents systèmes sur des VMs Linux.
- Vérification quotidienne des équipements du réseau dans les salles informatiques
- Dépannage des ordinateurs
- Mise en réseaux de nouvelles salles informatiques
- Assurer la maintenance préventive

**Environnement Technique :** Trello, Systèmes d'exploitation Windows, MacOS et Linux, Outils de virtualisation : VirtualBox, VMware, multimètre numérique