

Mis à jour le 25/08/2026

S'inscrire

Formation ROS 2

3 jours (21 heures)

Présentation

ROS 2 est un framework logiciel dédié à la robotique, permettant de concevoir des applications distribuées, modulaires et interopérables. Grâce aux nœuds, aux mécanismes de communication et à l'intégration native de Python, ROS 2 facilite le développement de systèmes robotiques fiables et évolutifs. Notre formation ROS 2 vous permettra d'acquérir les compétences nécessaires pour développer, exécuter et superviser des applications robotiques reposant sur ROS 2. Vous découvrirez son architecture, son outillage et les principes essentiels de communication entre composants. Vous apprendrez à créer des packages Python, à développer des nœuds ROS 2, à publier et souscrire à des topics, ainsi qu'à concevoir des services et des actions adaptés aux besoins d'un système robotique. La formation aborde également la configuration des paramètres, les fichiers de lancement, la gestion de la QoS, la transformation de repères avec TF2 et l'analyse des échanges à l'aide des outils de diagnostic ROS 2. À l'issue de la formation, vous serez en mesure de structurer une application ROS 2 intermédiaire, de tester ses communications, d'automatiser son exécution et d'identifier les principaux problèmes de configuration ou de performance.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la technologie et ses nouveautés.

Objectifs

- Comprendre l'architecture distribuée et les concepts fondamentaux de ROS 2
- Créer et organiser des packages Python dans un workspace ROS 2
- Développer des nœuds utilisant les topics, services et actions
- Configurer les paramètres, les fichiers de lancement et les profils de QoS
- Diagnostiquer les communications et enregistrer des données avec les outils ROS 2
- Construire une application robotique modulaire et exécutable

Public visé

- Développeur Python
- Développeur robotique
- Ingénieur en systèmes embarqués
- Ingénieur automatisme
- Architecte logiciel robotique

Pré-requis

- Maîtrise des fondamentaux de la programmation Python
- Connaissance de l'utilisation d'un terminal Linux
- Notions de programmation orientée objet
- Compréhension des principes de communication client serveur et publication souscription

Pré-requis techniques

- Poste sous Ubuntu 24.04 ou machine virtuelle compatible
- Installation de ROS 2 Jazzy Jalisco et de Python 3
- Environnement de développement Visual Studio Code ou éditeur équivalent
- Configuration recommandée : processeur 4 cœurs, 8 Go de RAM et 20 Go d'espace disque disponible
- Connexion Internet stable permettant l'installation de dépendances et de packages

Programme de notre formation ROS 2

[Jour 1 - Matin]

Fondamentaux et environnement ROS 2

- Positionnement de ROS 2 dans un système robotique et différences avec une architecture applicative classique
- Architecture distribuée : nodes, graph ROS, namespaces, interfaces et découverte des composants
- Organisation d'un environnement de travail : workspace, packages, overlays et gestion des dépendances
- Prise en main de la ligne de commande : `ros2 run`, `ros2 node`, `ros2 topic` et introspection du graphe
- Création et compilation d'un package Python avec `colcon`
- Atelier pratique : préparer un workspace ROS 2, créer un premier package Python et explorer les nœuds d'exemple

[Jour 1 - Après-midi]

Nœuds Python et communication par topics

- Structure d'un nœud rclpy : initialisation, cycle d'exécution, logger et gestion propre de l'arrêt
- Conception d'un publisher et d'un subscriber avec les messages standard ROS 2
- Gestion des timers, callbacks, files de messages et traitements asynchrones simples
- Types de messages, conventions de nommage et création d'interfaces personnalisées .msg
- Inspection des flux avec ros2 topic, affichage des types et contrôle de la fréquence des publications
- Atelier pratique : développer en Python un couple publisher subscriber pour simuler et superviser les données d'un capteur

[Jour 2 - Matin]

Services, actions et paramètres

- Choisir entre topic, service et action selon le type d'interaction attendu
- Développer un serveur et un client de service pour les commandes courtes avec réponse immédiate
- Mettre en œuvre un serveur et un client d'action avec objectif, feedback, résultat et annulation
- Définir des interfaces personnalisées .srv et .action
- Déclarer, lire et modifier des paramètres afin de rendre un nœud configurable
- Atelier pratique : créer un service de commande, une action de déplacement simulé et des paramètres de configuration

[Jour 2 - Après-midi]

Configuration, lancement et qualité de service

- Centraliser la configuration applicative dans des fichiers YAML de paramètres
- Construire des fichiers de lancement Python pour démarrer plusieurs nœuds de façon cohérente
- Utiliser les arguments de lancement, les remappings, les namespaces et les variables d'environnement
- Comprendre les politiques de QoS : fiabilité, durabilité, historique, profondeur et compatibilité
- Adapter les profils QoS aux flux de capteurs, aux commandes et aux échanges critiques
- Atelier pratique : lancer une application multi-nœuds configurable et corriger une incompatibilité QoS simulée

[Jour 3 - Matin]

Repères, données et diagnostic

- Principes des repères de coordonnées et rôle de TF2 dans une application robotique
- Publier, écouter et visualiser des transformations statiques et dynamiques
- Découvrir les outils de visualisation et d'inspection : RViz2, rqt et ligne de commande

- Enregistrer et rejouer des données avec ros2 bag pour reproduire un scénario
- Analyser les nœuds, topics, services, actions, paramètres et journaux pour isoler une anomalie
- Atelier pratique : visualiser un arbre TF2, enregistrer un flux de données et diagnostiquer une communication défailante

[Jour 3 - Après-midi]

Conception d'une application robotique intégrée

- Structurer une application ROS 2 en packages cohérents, interfaces explicites et composants réutilisables
- Organiser les responsabilités entre acquisition, décision, commande, supervision et visualisation
- Appliquer les bonnes pratiques de robustesse : gestion des erreurs, logs, paramètres et démarrage contrôlé
- Préparer la maintenance d'une application : documentation, reproductibilité du build et validation des dépendances
- Synthétiser les choix d'architecture, de communication et de QoS pour un cas d'usage robotique
- Atelier pratique : réaliser une mini application ROS 2 intégrant capteur simulé, traitement, commande, lancement et diagnostic

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.