

# Infrastructure (déploiement)

## Docker Compose

### Détails de certaines variable d'environnement défini sur nodecontroller

Variable	Explication	Valeur intial
MQ_SERVICE_NAME	Il s'agit du nom de la queue de message auquel le nodecontroller doit se connecter. La valeur doit être celle définit sous "labels:" -> "ARTIFACT_NAME" dans la configuration du service "event-store".	EventStream
CUSTOM_CONTAINER_PORTS_DISCOVERY	Par défaut, le nodecontrolleur cherche le port 80 pour le routing de requetes. Cependant rabbitmq et son protocol AMQP utilisent un port different. Ceci permet d'ajouter d'autres valeurs de ports. être ajouter comme dans une list. Ex: 443, 5672, etc...	5672
BRIDGE_PORT	Port du rabbitmq portainer. Ne pas modifier	32674

### Balises labels

### Scénarios communs

Variable	Explication	Valeurs Possibles ou commentaire
ARTIFACT_NAME	Le nom donné au service. Doit être unique. Sert au routing de trafic vers ce service par nom au lieu d'adresses. Essentiel sinon vous êtes automatiquement disqualifié de l'évalutation d'implémentation 2,3,4	-
ARTIFACT_CATEGORY	Le type de composant. Essentiel pour les itérations 2, 3, 4	Valeurs possibles: Computation, Connector, Database
REPLICAS	Le nombre d'instances de ce services que l'orchestrateur doit vous garantir	Aucune valeur n'est pas comme 1, avec aucune valeur si le service est détruit, une nouvelle instance ne sera pas créé.

### Scénarios avancés

Variable	Explication	Valeurs Possibles ou commentaire
POD_NAME	Kubernetes gère des ensembles de conteneurs appelés pods et permet de déployer un pod contenant plusieurs conteneurs. Cette fonctionnalité est également disponible ici. Si plusieurs services partagent le même POD_NAME, leur durée de vie est liée (s'il y en a un qui est supprimé, tous le seront). Cela permet également de créer un réseau privé de découverte de services entre les membres d'un même pod. Par défaut, le système crée un pod virtuel pour chaque service, mais une abstraction est utilisée pour simplifier le processus tout en conservant la fonctionnalité. Il est possible d'utiliser un format d'adressage tel que : POD_NAME.ARTIFACT_NAME. Ce cas permet d'avoir un même ARTIFACT_NAME dans deux déclaration de services differente si leur POD_NAME est différent. Cette fonctionnalité permet l'utilisation de Sidecar et Ambassador patterns.	<b>Bonne pratique:</b> Spécifier REPLICAS dans un seul des services partageant un même POD_NAME. Définir un POD_ID identique pour tout les services d'un même Pod (pas essentiel mais évite une reconstruction du pod au déploiement). Mettre un service avec "DNS: Public" sinon le networking va être décevant si le service mesh est utilisé.
POD_ID	Si tout les membres d'un Pod (Partagent le même POD_NAME) n'ont pas le même POD_ID, ils seront détruits instantanément et recréé ensemble.	-
DNS	Permet de rendre invisible un service d'un Pod au accès externe par le service mesh. Les DNS sont en mode privé par défaut, mais si un service à le même nom que son pod, il est traité comme Public.	Peut toujours être atteint par son adresse si elle est déjà connu

### Balise environment

Variable	Explication	Valeurs Possibles ou commentaire
ID	Permet l'utilisation du service discovery, permet de savoir qui fait la demande d'adressage	Je conseil de mettre le même nom que container_name
SERVICES_ADDRESS	Nous voulions pouvoir avoir l'option de déployer sur une autre plateforme que Docker Desktop, mais avons changé d'avis, donc ce champs doit être host.docker.internal	<b>host.docker.internal</b>
NODE_CONTROLLER_PORT	Le port du nodecontroller pour pouvoir appelé son Api (utilisé par le ServiceMeshHelper nuget)	-

## Variables d'environnement

### Fichier .env

Variable	Explication	Exemple
GROUP	L'année, la session et votre numéro de groupe cours; format: Anné Session Groupe => AAAASSGG	20230200
TEAM_NAME	Votre numéro d'équipe	Equipe00

Variable	Explication	Exemple
USER	Votre code permanent (différent à l'intérieur d'une même équipe)	VERD18019700
SECRET	Votre mot de passe. Vous pouvez enlever le .env du git pour ne pas push votre mot de passe. Le mot de passe est encrypté et stocké sur les serveurs de l'école, mais dans l'éventualité d'une brèche, l'école et son personnel n'est pas responsable de tout dommage. De plus, avec un peu d'effort n'importe qui ayant accès à la base de données en temps qu'administrateur pourrait décrypter le hash et voir votre mot de passe.	SUPER_SECRET
INGRESS_PORT	Le port du Ingress déployé sur les serveurs de l'école, la valeur sera donnée à l'itération 2.	-
NODE_CONTROLLER_PORT	Le port du nodecontroller	-
INGRESS_ADDRESS	Cette Adresse va vous être donnée par votre chargé de laboratoire lors de la 2e itération	-
SERVICES_ADDRESS	Nous voulions pouvoir avoir l'option de déployer sur une autre plateforme que Docker Desktop, donc ce champs doit être host.docker.internal	host.docker.internal
STM	Votre API Key de Stm développeur, vous <b>devez</b> vous créer un compte par étudiant: <a href="#">API Hub (stm.info)</a>	-
TOMTOM	Votre API Key de TOMTOM, je recommande de créer un compte par étudiant: <a href="#">Register</a>	-