

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence de l'ouvrage **EIDOC** dans la zone de recherche
et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

Chapitre 1 Principes fondamentaux

- 1. Docker comme alternative légère à la virtualisation 15
 - 1.1 L'approche par virtualisation 15
 - 1.2 Tentatives de réduction de la consommation de ressources 17
 - 1.3 Comment Docker règle radicalement le problème 18
 - 1.4 Positionnement de Docker par rapport à la virtualisation 20
- 2. Principe des conteneurs 22
 - 2.1 Les apports de Docker 22
 - 2.2 Principe des conteneurs industriels 23
 - 2.3 Docker et l'approche normalisée 24
- 3. Les fondements de Docker 25
 - 3.1 Les technologies Linux clés pour Docker 26
 - 3.1.1 Namespaces 26
 - 3.1.2 Mise en œuvre d'un namespace 27
 - 3.1.3 cgroups 29
 - 3.2 Autres dépendances du système 29
 - 3.2.1 netfilter et iptables 29
 - 3.2.2 capabilities 30
 - 3.2.3 AppArmor et SELinux 30
 - 3.3 Architecture du moteur Docker 30
 - 3.3.1 LXC 30
 - 3.3.2 libcontainer 31
 - 3.3.3 containerd 32
 - 3.4 Architectures complémentaires 34
 - 3.4.1 rkt 34
 - 3.4.2 Container Runtime Interface 34

2 --- Docker

Concepts fondamentaux et déploiement d'applications distribuées

3.4.3	Un foisonnement d'alternatives	35
3.5	Système de fichiers en couches	35
3.5.1	Principe d'isolation des fichiers	35
3.5.2	Approche par virtualisation	35
3.5.3	Utilité des systèmes de fichiers en couches	36
3.5.4	Gestion des modifications de fichiers	38
3.5.5	Dernière couche en écriture	39
3.5.6	Technologies utilisées	41
4.	Les plus de Docker	41
4.1	Au-delà du cloisonnement Linux	41
4.2	L'approche un conteneur = un processus	42
4.3	L'écosystème Docker	43
5.	Architectures de services	44
5.1	Historique des architectures de services	45
5.1.1	Principes	45
5.1.2	Approche EAI	45
5.1.3	Approche SOA	45
5.1.4	Micro-services	46
5.1.5	Lien à l'urbanisation des SI	47
5.2	Architecture de microservices	47
5.2.1	Principe	47
5.2.2	Avantages	49
5.2.3	Inconvénients	51
5.3	Apport de Docker	52
5.4	Fil conducteur	53

Chapitre 2 Installation

1.	Éditions et canaux	55
1.1	Un standard sous Linux	55
1.2	Les canaux	56
1.3	Les éditions	56

- 2. Utiliser des machines dans le cloud 57
 - 2.1 Amazon AWS 57
 - 2.1.1 Offre 57
 - 2.1.2 Création de la machine virtuelle 60
 - 2.1.3 Accès à la machine virtuelle 67
 - 2.1.4 Gestion de la consommation 71
 - 2.1.5 Suppression de la machine virtuelle 72
 - 2.2 Microsoft Azure 74
 - 2.2.1 Offre 74
 - 2.2.2 Création de la machine virtuelle 75
 - 2.2.3 Accès à la machine virtuelle 81
 - 2.2.4 Suppression de la machine virtuelle 89
 - 2.3 Google Cloud Platform 89
 - 2.3.1 Offre 89
 - 2.3.2 Création de la machine virtuelle 92
 - 2.3.3 Accès à la machine virtuelle 98
 - 2.3.4 Suppression de la machine virtuelle 99
- 3. Installation de Docker 101
 - 3.1 Installation de Docker sur Linux 101
 - 3.1.1 Prérequis 101
 - 3.1.2 Gestionnaires de paquets 102
 - 3.1.3 Installation par script 102
 - 3.1.4 Cas particulier de l'installation sur Clear Linux 103
 - 3.2 Le paradoxe Docker sous Windows 106
 - 3.2.1 Des technologies différentes 106
 - 3.2.2 Docker et Hyper-V 108
 - 3.2.3 L'âge de la maturité 108
 - 3.3 L'outil Docker pour Windows 109
 - 3.3.1 Outil d'installation 109
 - 3.3.2 Choix du canal 111
 - 3.3.3 Mises à jour 113
 - 3.3.4 Assistant d'installation 113
 - 3.3.5 Enfin du natif avec WSL 2 116
 - 3.3.6 Désinstallation 123
 - 3.4 Docker pour Windows Server 124
 - 3.4.1 Installation 124
 - 3.4.2 Utilisation de machines préinstallées 129

3.4.3	Exécution de conteneurs Linux	130
3.4.4	Paramétrage système	134
3.5	Utilisation de Docker avec Vagrant	135
3.5.1	Principe	135
3.5.2	Création d'une machine	136
3.5.3	Provisionnement de Docker	141
3.5.4	Aménagement de l'accès à la machine	141
3.5.5	Connexion à la machine provisionnée	144
3.5.6	Poursuite des opérations Vagrant	149

Chapitre 3

Premiers pas avec Docker

1.	Hello World, Docker	151
1.1	Démarrage d'un conteneur simple	151
1.2	Détails des opérations effectuées	152
1.2.1	Récupération d'une image	152
1.2.2	Identité de l'image	154
1.2.3	Taille des images	156
1.2.4	Anatomie de l'image Linux	158
1.2.5	Anatomie de l'image Windows	163
1.2.6	Lancement du processus	164
1.2.7	Exécution du processus dans un conteneur	165
1.2.8	Opérations suivant l'exécution	166
2.	Utiliser des images Docker préexistantes	169
2.1	Le registre Docker Hub	169
2.1.1	Le principe	169
2.1.2	Recherche et qualification d'images	170
2.1.3	Exemple de recherche	173
2.1.4	Cas des images communautaires	177
2.1.5	Compléments sur les images officielles	180
2.1.6	Recherche par la ligne de commande	181
2.1.7	Précautions sur une image non officielle	183
2.2	Gestion du compte Docker Hub et dépôts privés	188
2.2.1	Création d'un compte	189
2.2.2	Caractéristiques du compte	190
2.2.3	Automated build et compte GitHub	192

- 2.2.4 Connexion au compte en ligne de commande 203
- 2.2.5 Webhook sur événement de push dans Docker Hub 204
- 2.2.6 Déconnexion des comptes Docker Hub et GitHub 205
- 3. Un second conteneur 207
 - 3.1 Récupération de l'image 207
 - 3.2 Explication des tags 208
 - 3.3 Premier lancement 210
 - 3.4 Lancement en mode interactif 212
 - 3.5 Persistance des modifications sous forme d'une image 214
 - 3.6 Prise en main du client Docker 217
 - 3.6.1 Ménage dans les conteneurs 217
 - 3.6.2 Ménage dans les images 218
 - 3.6.3 Le grand ménage 218
 - 3.6.4 Suppression automatique à la sortie 219
 - 3.6.5 Affectation d'un nom de conteneur 220
 - 3.6.6 Modification du point d'entrée par défaut 221
 - 3.6.7 Envoi de variables d'environnement 222
 - 3.6.8 Modification du hostname 223
 - 3.7 Manipulation des conteneurs 224
 - 3.7.1 Lancement en mode bloquant 224
 - 3.7.2 Lancement en arrière-plan 226
 - 3.7.3 Gestion correcte du cycle de vie des conteneurs 230
 - 3.7.4 Exposition de fichiers 233
 - 3.7.5 Supervision des conteneurs 235
- 4. Retours sur les premiers pas 237

Chapitre 4
Création et gestion d'images Docker

- 1. Création manuelle d'une nouvelle image 239
 - 1.1 Objectif 239
 - 1.2 Approche 239
 - 1.3 Difficultés 240
 - 1.4 Conclusion 240
- 2. Utilisation d'un Dockerfile 241
 - 2.1 Intérêt des fichiers Dockerfile 241
 - 2.2 Utilisation d'un fichier Dockerfile 245

6 **Docker**

Concepts fondamentaux et déploiement d'applications distribuées

2.3	Anatomie d'un fichier Dockerfile	247
2.3.1	FROM	247
2.3.2	RUN	248
2.3.3	ENV	249
2.3.4	VOLUME	250
2.3.5	COPY	257
2.3.6	ENTRYPOINT	257
2.3.7	EXPOSE	259
2.3.8	CMD	259
2.4	Notre premier Dockerfile	260
2.4.1	Fonctionnalité souhaitée	260
2.4.2	Création et test du script.	260
2.4.3	Création du Dockerfile	262
2.4.4	Génération de l'image	264
2.4.5	Lancement du conteneur	265
2.4.6	Arrêt et relance du conteneur	267
2.4.7	Une méthode moins brutale	267
2.4.8	Gestion des paramètres	268
2.4.9	Reconstruction de l'image et cache	270
2.5	Commandes additionnelles	272
2.5.1	Gestion des fichiers	272
2.5.2	Notion de contexte	273
2.5.3	Retours sur l'affectation du processus à démarrer	273
2.5.4	Remarque sur le format ligne de commande ou exécution.	276
2.5.5	Commandes diverses	278
3.	Partage et réutilisation simple des images	281
3.1	Envoi sur votre compte Docker Hub	281
3.2	Export et import sous forme de fichiers	287
4.	Bonnes pratiques	288
4.1	Principe du cache local d'images.	288
4.2	Principe du cache à la compilation.	293
4.2.1	Retour sur les images intermédiaires	293
4.2.2	Anatomie d'une compilation d'image	296
4.2.3	Analyse d'une modification du Dockerfile	298
4.2.4	Gestion correcte des étiquettes	300
4.2.5	Invalidation du cache par modification de l'image de base.	303
4.2.6	Invalidation du cache par modification du contexte.	308

- 4.3 Conséquences sur l'écriture des Dockerfile 310
 - 4.3.1 Le problème sur les opérations non idempotentes 310
 - 4.3.2 Contournement du problème de cache 313
 - 4.3.3 Effets bénéfiques sur le nombre et la taille des images 316
 - 4.3.4 Ordonnancement des commandes dans le Dockerfile 318
- 4.4 Ajout d'une image de cache intermédiaire 318
- 4.5 Mise en œuvre d'un cache de paquets 320
- 4.6 Conséquences sur le choix des images de base 323
 - 4.6.1 La bonne image de base 323
 - 4.6.2 Votre propre image de base 325
 - 4.6.3 Adapter l'image de base avec les compilations multi-stage 327
- 4.7 Arborescence recommandée 330
 - 4.7.1 Avantage d'une arborescence type 330
 - 4.7.2 Intégration des fichiers 331
 - 4.7.3 Limitation du contexte 333
- 4.8 La question du processus unique 334
 - 4.8.1 Principe général 334
 - 4.8.2 Exception au principe général avec Supervisor 335
 - 4.8.3 Critique 336
 - 4.8.4 Approche intermédiaire 337

Chapitre 5
Docker et les microservices

- 1. Présentation de l'application exemple 339
 - 1.1 Principes à l'œuvre 339
 - 1.1.1 Utilité 339
 - 1.1.2 Architecture de microservices 340
 - 1.2 Description de l'application 340
 - 1.2.1 Présentation des fonctionnalités 340
 - 1.2.2 Limites logicielles 341
 - 1.2.3 Où trouver l'application exemple 342
 - 1.2.4 Utilisation des API 344
 - 1.2.5 Scénarios d'utilisation 345
 - 1.3 Mode de développement 349
 - 1.3.1 Code source et paramétrage 349
 - 1.3.2 Intégration continue 352

2.	Déployer automatiquement avec Docker Compose	358
2.1	Docker Compose	358
2.1.1	Principe	358
2.1.2	Installation	359
2.2	Écriture du fichier docker-compose.yml	359
2.2.1	Premier service	359
2.2.2	Un service sans interface graphique	364
2.2.3	Mise en place de Traefik	368
2.2.4	Complément du fichier descriptif	374
2.3	Vérification du déploiement	379
2.4	Paramétrage de l'application	381
2.4.1	Avec Postman	382
2.4.2	Avec curl	390
2.5	Test de l'application	391
2.6	Vérification en base de données	391
2.6.1	Objectif	391
2.6.2	Mise en place du tunnel SSH	392
2.6.3	Lecture par un client MongoDB	395
3.	Exploitation d'une infrastructure Docker	396
3.1	Le réseau dans Docker	396
3.1.1	Mode de fonctionnement standard (bridge)	396
3.1.2	Modes de fonctionnement alternatifs	399
3.1.3	Création dynamique de réseaux dédiés	401
3.1.4	Support des liens entre conteneurs	403
3.1.5	Extension à l'état de santé des conteneurs	405
3.1.6	Approche complètement découplée	407
3.1.7	Autres options	408
3.1.8	Limites de la couche réseau existante	408
3.2	Les volumes Docker	409
3.2.1	Le problème de la persistance	409
3.2.2	Les volumes comme solution simple	409
3.2.3	Lien direct sur un répertoire local	412
3.2.4	Partage de volumes	412
3.2.5	Gestion des volumes orphelins	413
3.2.6	Sophistication de l'approche	414
3.2.7	Application à la gestion des logs	415
3.2.8	L'état de l'art	416

- 4. Exemples de Dockerfile pour d'autres langages 416
 - 4.1 Python 417
 - 4.2 Java 418
 - 4.3 Node.js. 420
 - 4.4 Go 420
- 5. Pour aller plus loin en architecture microservices 421
 - 5.1 Bonnes pratiques. 421
 - 5.2 Lien à l'urbanisation des SI. 421
 - 5.3 Persistance. 422
 - 5.4 Orchestration fonctionnelle. 423

Chapitre 6
Orchestration des conteneurs

- 1. Le besoin d'orchestration. 425
 - 1.1 Objectif 425
 - 1.2 Approche théorique 427
 - 1.2.1 La problématique de montée en charge 427
 - 1.2.2 La solution découplée 430
 - 1.2.3 Conséquences sur l'approche initiale 431
 - 1.3 Lien aux microservices 432
 - 1.3.1 Orchestration des services. 432
 - 1.3.2 Élasticité 433
 - 1.3.3 Robustesse 433
 - 1.4 Fonctionnement pratique 434
 - 1.4.1 Notion de réseau 434
 - 1.4.2 Les différents types de nœuds. 435
 - 1.4.3 Fonctionnalités du cluster. 436
- 2. L'approche Docker Swarm 436
 - 2.1 Gestion du cluster Swarm 437
 - 2.1.1 Prérequis 437
 - 2.1.2 Initialisation. 438
 - 2.1.3 Liaison des agents 440
 - 2.1.4 Ajout d'un manager. 443
 - 2.1.5 Les limites à l'ajout de managers. 445
 - 2.1.6 Promotion d'un nœud 447
 - 2.1.7 Suppression d'un nœud. 448

2.2	Test du cluster avec un cas simple	451
2.3	Déploiement manuel sur le cluster Swarm	455
2.3.1	Application exemple	455
2.3.2	Préparation des images	456
2.3.3	Lancement des services	457
2.3.4	Premier test.	462
2.3.5	Mise en place d'un réseau overlay dédié	463
2.3.6	Validation du fonctionnement	465
2.3.7	Passage à l'échelle	467
2.3.8	Suppression des ressources déployées	468
2.4	Déploiement de l'application microservices	469
2.4.1	Récupérer l'application exemple	469
2.4.2	Le retour de Docker Compose	469
2.4.3	Détail du vocabulaire	471
2.4.4	Compilation éventuelle des images	472
2.4.5	Lancement par Docker Compose	473
2.4.6	Déploiement d'une stack	474
2.4.7	Diagnostic et validation du fonctionnement	479
2.4.8	Utilisation du DNS	489
2.5	Considérations additionnelles	490
2.5.1	Provenance des images	490
2.5.2	Passage à l'échelle	491
2.5.3	Mise à jour des images	491
2.5.4	Gestion des contraintes	496
2.5.5	Arrêt de la stack	497
2.6	Mélanger déploiement Swarm et Traefik	497
2.6.1	Fusion des grammaires précédentes	498
2.6.2	Passer Traefik en mode Swarm	501
2.6.3	Correction du placement de Traefik	501
2.6.4	Correction de la grammaire sur les labels	504
2.6.5	Note sur Traefik V2	507
3.	Kubernetes	507
3.1	Positionnement	507
3.2	Concepts	508
3.2.1	Pods	508
3.2.2	Namespaces	509
3.2.3	Services	509

- 3.2.4 IngressController 510
- 3.2.5 Volumes 510
- 3.2.6 ConfigMap 511
- 3.2.7 Deployment 512
- 3.3 Installation 513
 - 3.3.1 Choix de la méthode 513
 - 3.3.2 Création d'un cluster AKS 514
 - 3.3.3 Paramétrage de l'accès au cluster 516
 - 3.3.4 Accès au cluster 518
- 3.4 Premier déploiement par ligne de commande 519
 - 3.4.1 Contexte 519
 - 3.4.2 Création du déploiement 519
 - 3.4.3 Vérification du déploiement 520
 - 3.4.4 Présence d'un pod 521
 - 3.4.5 Exposition par un service 521
 - 3.4.6 Test de l'application 522
 - 3.4.7 Nettoyage 523
- 3.5 Déploiement de l'application microservices 524
 - 3.5.1 Structure du projet de déploiement 524
 - 3.5.2 Namespace 526
 - 3.5.3 Fichier de paramétrage 526
 - 3.5.4 Fichiers liés aux bases de données 528
 - 3.5.5 Fichiers liés aux services 531
 - 3.5.6 Définition de l'ingress 534
 - 3.5.7 Lancement d'un seul coup 539
 - 3.5.8 Consommation de l'application 540
 - 3.5.9 Déploiement continu 541
 - 3.5.10 Arrêt de la stack 543
- 3.6 Pour aller plus loin avec Kubernetes 543
- 4. Azure Container Instances 543
 - 4.1 Principe 543
 - 4.2 Préparation d'une image 544
 - 4.3 Lancement du conteneur 544
 - 4.4 Correction de l'erreur et relance 548
 - 4.5 Coût et effort 550

Chapitre 7

Docker et Windows

1. Spécificités sous Windows	553
1.1 Images de base	554
1.1.1 Nécessité	554
1.1.2 Windows Server Core	554
1.1.3 Nano Server	555
1.1.4 Compatibilité et disponibilité	557
1.1.5 Autres versions	558
1.1.6 Cas particulier de .NET	558
1.2 Différences avec Linux	559
1.2.1 Généralités	559
1.2.2 Dockerfile	559
1.2.3 Gestion du réseau	560
1.3 Mise en œuvre	562
2. L'outil Docker for Windows	562
2.1 Configuration standard	562
2.2 Les deux modes de conteneurs disponibles	565
2.3 Mode conteneurs Linux (hors mode WSL)	569
2.4 Gestion des images	571
2.5 Cas particulier des lecteurs de disque	572
2.6 Gestion Kubernetes	573

Chapitre 8

Industrialisation

1. Bonnes pratiques de production	575
1.1 Supervision	575
1.1.1 Approche "bétail plutôt qu'animal domestique"	575
1.1.2 Outillage possible	577
1.1.3 Outillage des logs	578
1.2 Bonnes pratiques logicielles	579
1.2.1 Importance de la normalisation des logs	579
1.2.2 API de statut	580
1.2.3 Retry policies et circuit breakers	580
1.2.4 Répartition de la charge	582

- 1.2.5 Inversion de consommation 583
 - 1.2.6 Gestion du multitenant au moyen des conteneurs 585
 - 1.3 Gestion de la performance 586
 - 1.3.1 Passage à l'échelle automatique. 586
 - 1.3.2 Cas particulier du cache 587
 - 1.3.3 Alignement sur le CPU et les threads. 588
 - 1.3.4 Approche "production only" 588
 - 1.4 Sécurité 590
 - 1.4.1 Mangue ou noix de coco ? 590
 - 1.4.2 Évolution de la gestion de la sécurité par Docker 592
 - 1.4.3 Pratique de sécurisation des ports 594
 - 1.4.4 Sécurité sur l'utilisation des images 594
 - 1.4.5 Restriction sur les ressources 596
- 2. Mise en œuvre d'un registre privé 596
 - 2.1 Objectifs 596
 - 2.2 Votre registre en complète autonomie 597
 - 2.2.1 Image à utiliser. 597
 - 2.2.2 Lancement du registre. 598
 - 2.2.3 Utilisation du registre 600
 - 2.3 Les limites du mode autonome 603
 - 2.3.1 Limites en termes d'utilisation 603
 - 2.3.2 Limites en termes de sécurité 604
 - 2.3.3 Limites en termes de performance 606
 - 2.3.4 Limites en termes de stockage 607
 - 2.3.5 Et toutes les autres limitations d'un registre autogéré 609
 - 2.4 Utilisation d'un service cloud déjà préparé 610
 - 2.4.1 Azure Container Registry 611
 - 2.4.2 Notes finales 616
 - 2.5 Approches complémentaires 618
 - 2.5.1 L'API du registre. 618
 - 2.5.2 Mise en place d'un miroir 619
- 3. Plateformes de déploiement de conteneurs 622
 - 3.1 Amazon Web Services 623
 - 3.1.1 Description des services. 623
 - 3.1.2 Amazon ECR 624
 - 3.1.3 Amazon ECS 625
 - 3.1.4 Amazon Elastic Beanstalk. 626

3.2	Microsoft Azure	628
3.2.1	Description des services	628
3.2.2	Azure Kubernetes Service	628
3.2.3	Azure Cluster Service Fabric	633
3.2.4	Azure Container Instances	635
3.3	Google Cloud Platform	637
3.3.1	Description des services	637
3.3.2	Google Kubernetes Engine	638
3.3.3	Google Compute Engine	642
3.3.4	Google Container Registry	643
3.3.5	Google Cloud Run	644
4.	Remarques sur les usages industriels de Docker	646
4.1	Supervision mixte	647
4.2	Intégration de bases de données	648
4.3	Cluster hybride	649
4.4	Docker en usine logicielle	650
4.4.1	Déploiement	652
4.4.2	Compilation	652
4.4.3	Infrastructure de l'usine logicielle	654
4.4.4	Gestion des tests	655
4.4.5	Utilisation pour les machines supports de tests	657
4.4.6	Registre pour l'ALM	658
4.5	Docker comme une commodité	659
	Index	661

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence de l'ouvrage **EPKUBCLU** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Chapitre 1 Fonctionnement de Kubernetes

- 1. Contenu du livre 9
 - 1.1 Objectifs 9
 - 1.2 Prérequis de lecture 10
 - 1.3 Avertissement sur les versions de Kubernetes 10
 - 1.4 Kubernetes et Azure 11
- 2. À quoi sert Kubernetes ? 12
 - 2.1 Les besoins en déploiement applicatif 12
 - 2.1.1 Montée en charge par répartition 12
 - 2.1.2 Mises à jour progressives 13
 - 2.1.3 Composition applicative robuste 13
 - 2.2 Structuration du système d'information 14
 - 2.2.1 Principes 14
 - 2.2.2 Découplage obtenu par Kubernetes 16
- 3. Historique de Kubernetes 18
 - 3.1 Aux origines : Google Borg 18
 - 3.2 Transfert à la Cloud Native Computing Foundation 19
 - 3.3 Et la concurrence ? 19
- 4. Principes d'architecture de Kubernetes 21
 - 4.1 Mise en cluster 21
 - 4.1.1 Principes 21
 - 4.1.2 Les masters et leurs composants 22
 - 4.1.3 Les nodes et leurs composants 25
 - 4.2 Concepts liés à Kubernetes 26
 - 4.2.1 Pods 27
 - 4.2.2 Namespaces 28

2 **Kubernetes**

Mise en œuvre d'un cluster et déploiement de microservices

4.2.3	Services	29
4.2.4	IngressController	32
4.2.5	Volumes	37
4.2.6	PersistentVolume et PersistentVolumeClaim	39
4.2.7	ConfigMap	41
4.2.8	Secrets	45
4.2.9	Deployment	48
4.2.10	ReplicaSet	50
4.2.11	DaemonSet	52
4.2.12	Résumé des concepts	53
4.3	Role Based Access Control	54
4.3.1	Principe	54
4.3.2	Role	56
4.3.3	Cluster Role	56
4.3.4	Role Binding	57
4.3.5	Cluster Role Binding	59
4.4	Notion de réseau	59
4.5	Kubernetes et Docker	60

Chapitre 2

Création et gestion d'un cluster Kubernetes

1.	Méthodes d'installation d'une plateforme Kubernetes	61
1.1	Plusieurs façons de faire	61
1.2	Environnement de développement	62
1.3	Environnement on premise	63
1.4	Environnement cloud	64
2.	Installation d'une plateforme Kubernetes	66
2.1	Généralités et préparation	66
2.1.1	Description du mode stacked	67
2.1.2	Description du mode external	69
2.1.3	Prérequis techniques	71

2.2	Installation avec Kubeadm.	72
2.2.1	Initialisation des machines proxy.	72
2.2.2	Configuration logicielle des machines proxy.	75
2.2.3	Configuration de Keepalived	75
2.2.4	Validation du fonctionnement de Keepalived	79
2.2.5	Configuration de HAProxy	80
2.2.6	Initialisation des machines master	83
2.2.7	Configuration des machines master.	85
2.2.8	Configuration de Docker	86
2.2.9	Installation des paquets pour Kubernetes	89
2.2.10	Préparation de la configuration réseau.	90
2.2.11	Installation du cluster	92
2.2.12	Gestion des certificats	95
2.2.13	Jointure des autres serveurs master	97
2.2.14	Option de récupération automatique des certificats.	98
2.2.15	Vérification de fonctionnement du cluster	99
2.2.16	Déploiement applicatif pour validation	101
2.2.17	Suppression du cluster	110
2.3	Installation avec Kubespray.	110
2.3.1	Description de l'outil	110
2.3.2	Opérations préparatoires.	111
2.3.3	Création éventuelle des machines proxy	112
2.3.4	Création des machines master	114
2.3.5	Paramétrage divers.	116
2.3.6	Installation du cluster	118
2.3.7	Vérification de fonctionnement du cluster	119
3.	Mise en œuvre d'un cluster Kubernetes	123
3.1	Connexion au cluster	123
3.1.1	Les utilisateurs	123
3.1.2	Modes de connexion	124
3.1.3	Le fichier KUBECONFIG	125
3.1.4	Gestion des contextes	125
3.1.5	Ajout du contexte pour les développeurs.	127

4 **Kubernetes**

Mise en œuvre d'un cluster et déploiement de microservices

3.1.6	Test du contexte pour les développeurs	131
3.1.7	Ajout du contexte pour les administrateurs	132
3.1.8	Test du contexte pour les administrateurs	133
3.1.9	Ajustement des autorisations pour les administrateurs	135
3.1.10	Ajustement des autorisations pour les développeurs . .	137
3.2	Exploitation du cluster	141
3.2.1	Binaire kubectl	141
3.2.2	Commandes de base	142
3.2.3	Commandes de déploiement	145
3.2.4	Commandes de gestion	146
3.2.5	Commandes de débogage	147
3.2.6	Commandes de paramétrage	149
3.3	Exposition des applications	150
3.3.1	Installation de l'Ingress Controller Nginx	150
3.3.2	Paramétrage d'une Ingress Rule	153
3.3.3	Validation du fonctionnement	157
3.3.4	Mise en place de la haute disponibilité	157
3.3.5	Passage à l'échelle de l'application	158
3.3.6	Gestion de versions	160
4.	Maintien en condition opérationnelle d'un cluster Kubernetes . . .	164
4.1	Surveillance de l'écosystème Kubernetes	164
4.1.1	Surveillance du cluster	164
4.1.2	Surveillance des pods	165
4.1.3	Présentation des outils de supervision	166
4.1.4	Architecture de supervision proposée	167
4.1.5	Mise en place de l'architecture	168
4.1.6	Paramétrage d'un tableau de bord	172
4.1.7	Utilisation des outils déployés	173
4.2	Filtrage réseau avec les NetworkPolicies	175
4.2.1	Présentation des NetworkPolicies	175
4.2.2	Exemple de NetworkPolicy	176
4.2.3	NetworkPolicies standards	178
4.2.4	Mise en pratique des NetworkPolicies	179

- 4.3 Quelques bonnes pratiques 182
 - 4.3.1 Recommandations sur les créations
de ressources Kubernetes. 182
 - 4.3.2 Recommandations sur les infrastructures cloud 183
 - 4.3.3 Stratégie de téléchargement des images. 184

Chapitre 3
Déploiement d'applications avec Kubernetes

- 1. Contexte 187
 - 1.1 Objectifs généraux 187
 - 1.1.1 Exploiter la plateforme Kubernetes mise en place. 187
 - 1.1.2 Remarque sur l'approche DevOps 188
 - 1.1.3 Principaux enseignements à attendre. 190
 - 1.2 Outillage 191
 - 1.2.1 Cluster Azure Kubernetes Services. 191
 - 1.2.2 Kubectl 207
 - 1.2.3 Azure CLI 207
 - 1.2.4 Paramétrage de kubectl 209
 - 1.2.5 Docker pour Windows 210
- 2. Premier déploiement par ligne de commande 212
 - 2.1 Contexte 212
 - 2.2 Création du déploiement 213
 - 2.3 Vérification du déploiement 213
 - 2.4 Présence d'un pod 214
 - 2.5 Exposition par un service 215
 - 2.6 Test de l'application 216
 - 2.7 Nettoyage 217
- 3. Second déploiement à l'aide d'un fichier de configuration 218
 - 3.1 Objectifs 218
 - 3.2 Application exemple 219
 - 3.2.1 Présentation de l'application 219
 - 3.2.2 Limites logicielles 220

6 **Kubernetes**

Mise en œuvre d'un cluster et déploiement de microservices

3.2.3	Pourquoi des microservices ?	221
3.2.4	API	222
3.2.5	Scénarios d'utilisation	223
3.3	Développement logiciel	228
3.3.1	Code source et paramétrage	228
3.3.2	Intégration continue	232
3.4	Déploiement de l'application	238
3.4.1	Structure du projet de déploiement	239
3.4.2	Namespace	240
3.4.3	Fichier de paramétrage	241
3.4.4	Fichiers liés aux bases de données	243
3.4.5	Fichiers liés aux services	246
3.4.6	Définition de l'Ingress	249
3.4.7	Lancement d'un seul coup	254
3.5	Vérification du déploiement	255
3.6	Paramétrage de l'application	258
3.6.1	Avec Postman	258
3.6.2	Avec curl	267
3.6.3	Modification des variables d'environnement	267
3.7	Test de l'application	269
3.8	Supervision du cluster	270
3.8.1	Azure Monitor Insights	271
3.8.2	Azure Monitor logs	275
3.8.3	Kubernetes dashboard	276
3.8.4	Azure Monitor	284
3.8.5	Autres méthodes de monitoring	286
3.9	Scalabilité	287
3.9.1	Mode statique	287
3.9.2	Mode dynamique	291
3.10	Fonctionnement du déploiement continu	297
3.11	Mise à jour des versions applicatives	306
3.11.1	Commande de mise à jour	306
3.11.2	Fonctionnement en rolling update	308

- 3.11.3 Amélioration du fonctionnement par les sondes 312
- 3.11.4 Historique de mises à jour. 318
- 3.12 Gestion de la persistance 321
 - 3.12.1 Kubernetes et les volumes. 321
 - 3.12.2 Utilisation d'un disque Azure 322
 - 3.12.3 Remarque sur les bases de données en Kubernetes 336
 - 3.12.4 Sauvegarde de bases de données 337
- 3.13 Utilisation de Secrets 349
- 3.14 Suppression de l'application. 356
 - 3.14.1 Suppression dans Kubernetes 356
 - 3.14.2 Cas particulier sur l'ingress 358
 - 3.14.3 Suppression des ressources Azure. 360

- Index 361