



## REFERENTIEL D’EVALUATION ET DE COMPETENCES DE LA CERTIFICATION METTRE EN ŒUVRE DEVOPS POUR LE CLOUD AWS

*(Titre en anglais : AWS Certified DevOps Engineer - Professional)*

### MODALITES D’EVALUATION

Pour obtenir cette certification, il est proposé au candidat d’évaluer leurs compétences à travers un (1) examen en ligne, supervisé par l’organisme Pearson VUE ou PSI Services, délivrés dans un centre d’examen accrédité (ou via de la surveillance à distance).

L’examen dure environ trois heures (3h00) – livres fermés - et comprend une variété de questions appelant différentes formes de réponse\* : Questions à choix multiples, Questions à réponses multiples.

\*Détaillées à cette URL : [https://d1.awsstatic.com/training-and-certification/docs-devops-pro/AWS-Certified-DevOps-Engineer-Professional\\_Exam-Guide.pdf](https://d1.awsstatic.com/training-and-certification/docs-devops-pro/AWS-Certified-DevOps-Engineer-Professional_Exam-Guide.pdf)

Le seuil de réussite est fixé à environ 75% de bonnes réponses, qui correspond à un score de passage de 750 points (score à l'échelle). Le pourcentage réel varie d'un examen à l'autre. La note de passage est basée sur l'apport d'experts en la matière, le niveau de compétence requis pour être considéré comme compétent dans le domaine du contenu, et la difficulté des questions livrées pendant l'examen. Les pourcentages dans le tableau des compétences évaluées indiquent le poids relatif de chaque sujet principal de l'examen. Plus le pourcentage est élevé, plus les candidats devront répondre à des questions sur cette zone de contenu. La liste des tâches évaluées n’est pas exhaustive et peut couvrir d’autres tâches dans le cadre des compétences évaluées.

L’examen compte en tout 65 questions mais seulement 55 sont évaluées. L'examen comprend 10 questions non notées qui n'affectent pas le score du candidat. AWS collecte des informations sur la performance des candidats sur ces questions non notées afin d'évaluer ces questions en vue d'une utilisation future. Ces questions non notées ne sont pas identifiées lors de l'examen. Dans le résumé des compétences ci-dessous le pourcentage représente uniquement le contenu scoré. Il en est de même pour le nombre de questions par domaine évalué.

Le contenu des tests est réévalué régulièrement par les équipes Amazon Web Services pour refléter les dernières évolutions des services et de la plate-forme AWS.



## RESUME DES ACTIVITES PRINCIPALES :

Compétences mobilisables évaluées		Nature des tâches évaluées permettant de valider la compétence	Evaluation		
			% de l'évaluation globale	Modalités d'évaluation	Critères
<b>Automatiser le développement des cycles de vie</b>					
	<b>Automatiser un pipeline Développement continu / Intégration continue (CI/CD)</b>	<ul style="list-style-type: none"> <li>▶ Configurer les référentiels.</li> <li>▶ Configurer les services de génération du pipeline.</li> <li>▶ Intégrer les tests automatisés (par exemple tests unitaires, tests d'intégrité).</li> <li>▶ Configurer les produits et services de déploiement.</li> <li>▶ Orchestrer les différentes étapes du pipeline : Coder / Construire / Intégrer / Tester / Délivrer / Déployer / Opérer.</li> </ul>	<b>22% de l'évaluation globale de l'examen</b>	Examen en ligne avec une variété de questions (cf. détail plus haut) ** <b>Environ 40 minutes</b> (pour 12 à 14 questions évaluées) sont consacrées à cette compétence	Examen compensatoire, le taux de bonnes réponses doit être au <b>global de 75% minimum</b>
	<b>Déterminer les stratégies de contrôle des sources et comment les mettre en œuvre.</b>	<ul style="list-style-type: none"> <li>▶ Déterminer un flux de travail pour intégrer les modifications de code provenant de plusieurs contributeurs.</li> <li>▶ Evaluer les exigences de sécurité et recommander la conception d'accès au référentiel</li> <li>▶ Réconcilier les versions d'application en cours d'exécution avec les versions du référentiels (balises / tags)</li> <li>▶ Différencier les différents types de contrôle de source</li> <li>▶ Utiliser AWS CodeCommit pour gérer le contrôle des sources, conserver et crypter le code, mettre en place les autorisations de collaboration sur les codes.</li> <li>▶ Gérer le répertoire local GIT et les répertoires à distance.</li> <li>▶ Utiliser la fonctionnalité des branches pour le développement avec GIT.</li> <li>▶ Fusionner les branches de fonctionnalité avec la branche de développement.</li> </ul>			



	<b>Automatiser et intégrer les tests</b>	<ul style="list-style-type: none"><li>▶ Exécuter les tests d'intégration dans le cadre du processus de fusion du code.</li><li>▶ Exécuter les tests de charges et de contrainte des applications a grande échelle.</li><li>▶ Mesurer la sante des applications en fonction des codes de sortie (bilan de sante robuste).</li><li>▶ Automatiser les tests unitaires pour verifier la reussite ou l'échec et la couverture du code avec AWS Code Pipeline, AWS CodeBuild.</li><li>▶ Utiliser AWS CodePipeline pour assurer l'automatisation du CI avec l'intégration des tests en suivant le pipeline : Sourcing – Build – Staging – Accord manuel – Production.</li><li>▶ Utiliser le service adéquat en fonction du stage du pipeline : Source (AWS CodeCommit, GitHub), Build (AWS Code Build), Test (AWS Code Build, AWS Device Farm), Deploy (AWS Cloud Formation, AWS Code Deploy), Approval (Amazon SNS), Invoke (AWS Lambda).</li></ul>			
	<b>Construire et gérer des artefacts en toute sécurité</b>	<ul style="list-style-type: none"><li>▶ Déterminer l'option de stockage à utiliser en fonction de la classication de sécurité des artefacts.</li><li>▶ Paramétrer CodePipeline pour produire et stocker dans S3 des artefacts d'entrée et de sortie (input &amp; output artifact) et ainsi pouvoir suivre les API, modifications et tests effectués dans le cadre du pipeline.</li><li>▶ Déterminer les dépendances code/environnement et les ressources requises (Exemple : CodeDeploy AppSpec, spécification de construction CodeBuild)</li></ul>			
	<b>Déterminer les stratégies de déploiement/de prestation (p. ex. A/B, bleu/vert, canari, rouge/noir) et comment les mettre en œuvre à l'aide des services AWS.</b>	<ul style="list-style-type: none"><li>▶ Utiliser les stratégies de déploiement en fonction de la situation et des besoins de l'entreprise : deploy in place, rolling, rolling with additional batch (beanstalk), immutable, blue/green, canary, linéaire et all-at-once.</li><li>▶ Critiquer les stratégies de déploiement existantes et proposer des améliorations</li><li>▶ Palier, pour chacune de ces stratégies de déploiement, a l'impact en cas d'échec de déploiement, le temps de</li></ul>			



		<p>déploiement, le risque de downtime, le processus pour revenir en arrière et la destination du code.</p> <ul style="list-style-type: none"> <li>▶ Comprendre la dynamique de déploiement blue/green sur AWS avec l'utilisation de Amazon Route 53 pour distribuer le trafic a 90% sur l'environnement de production existant et 10% sur le nouvel environnement de production.</li> </ul>			
<p><b>Gérer la configuration et l'infrastructure en tant que code</b></p>			<p><b>19% de l'évaluation globale de l'examen</b></p>	<p>Examen en ligne avec une variété de questions (cf détail plus haut) ** <b>Environ 34 minutes</b> (pour 10 à 12 questions évaluées) sont consacrées à cette compétence</p>	<p>Examen compensatoire, le taux de bonnes réponses doit être au <b>global de 75% minimum</b></p>
	<p><b>Déterminer les services de déploiement en fonction des besoins de déploiement</b></p>	<ul style="list-style-type: none"> <li>▶ Utiliser le service de déploiement en fonction du besoin.</li> <li>▶ Compte tenu d'un modèle de déploiement spécifique, classifiez et implémentez les services AWS pertinents pour répondre aux exigences</li> <li>▶ Utiliser AWS CloudFormation pour créer des ressources AWS en suivant un mode de fonctionnement ordonné et prévisible.</li> <li>▶ Utiliser AWS Elastic Beanstalk pour déployer et mettre à l'échelle des applications web et des services.</li> <li>▶ Utiliser EC2 pour déployer les applications sur une instance avec système d'opération.</li> <li>▶ Utiliser AWS Lambda pour exécuter le code sans utilisation de serveur.</li> </ul>			
	<p><b>Déterminer les modèles de déploiement d'applications et d'infrastructure en fonction des besoins de l'entreprise</b></p>	<ul style="list-style-type: none"> <li>▶ Équilibrer différentes considérations (coût, disponibilité, délai de récupération) en fonction des besoins de l'entreprise afin de choisir le meilleur modèle de déploiement.</li> <li>▶ Déterminer un modèle de déploiement en fonction de services AWS spécifiques</li> <li>▶ Analyser les risques associés aux modèles de déploiement et aux remèdes pertinent</li> <li>▶ Utiliser AWS CloudFormation pour pouvoir déployer une infrastructure automatiquement plutôt que manuellement.</li> <li>▶ Définir un environnement sur AWS CloudFormation et le déployer à plusieurs reprises sans besoin de reconfiguration</li> </ul>			



		<ul style="list-style-type: none"><li>▶ Utiliser AWS Elastic Beanstalk pour ne gérer que le code et laisser le service déployer l'application, provisionner les ressources, gérer les charges de travail et monitorer la santé de l'application.</li><li>▶ Utiliser AWS Step Functions pour orchestrer les différentes fonctions Lambda utilisées.</li></ul>			
	<b>Automatiser le provisionnement des ressources en assurant la sécurité du processus.</b>	<ul style="list-style-type: none"><li>▶ Choisir le meilleur outil d'automatisation en fonction des exigences.</li><li>▶ Utiliser AWS Organizations pour créer des groupes de compte, automatiser la création de comptes et appliquer automatique à chaque compte les accès et restrictions en fonction de l'appartenance aux groupes.</li><li>▶ Examiner les stratégies IAM et évaluer si des privilèges suffisants mais moindres sont accordés pour toutes les étapes du cycle de vie d'un déploiement (par exemple, création, mise à jour, promotion).</li><li>▶ Utiliser AWS Service Catalog pour créer et gérer des catalogues de services / ressources approuvées pour utilisation.</li></ul>			
	<b>Déterminer comment mettre en œuvre des hooks de cycle de vie sur un déploiement</b>	<ul style="list-style-type: none"><li>▶ Déterminer les techniques d'intégration appropriées pour répondre aux exigences du projet</li><li>▶ Ajouter des hooks de cycle de vie sur un déploiement pour mettre en pause certaines instances tandis que le service d'Autoscaling en lance ou en termine certaines.</li><li>▶ Implémenter ces hooks de cycles de vie pour les services AWS le supportant.</li><li>▶ Évaluer les procédures de déploiement pour déterminer les impacts des défaillances et évaluer les processus de restauration et de restauration</li></ul>			



	<p><b>Gérer les systèmes à l'aide des outils et services de gestion de la configuration AWS.</b></p>	<ul style="list-style-type: none"> <li>▶ Identifier les avantages et les inconvénients des outils de gestion de configuration AWS</li> <li>▶ Utiliser AWS Config pour évaluer et auditer les configurations des ressources AWS utilisées.</li> <li>▶ Assurer avec AWS Config un monitoring constant, vérifier la conformité de l'organisation des ressources et gérer les modifications de l'organisation.</li> <li>▶ Utiliser AWS Managed Services pour implémenter les pratiques recommandées pour la maintenance de l'infrastructure AWS et automatiser les tâches communes.</li> <li>▶ Gérer les applications sur AWS et sur les sites physiques via AWS OpsWorks.</li> <li>▶ Montrer la capacité d'exécuter des services de gestion de la configuration de bout en bout sans assistance tout en respectant les meilleures pratiques du secteur</li> </ul>			
<p><b>Surveiller et enregistrer les actions effectuées sur et dans l'infrastructure cloud</b></p>			<p><b>15% de l'évaluation globale de l'examen</b></p>	<p>Examen en ligne avec une variété de questions (cf détail plus haut) ** <b>Environ 27 minutes</b> (pour 8 à 10 questions évaluées) sont consacrées à cette compétence</p>	<p>Examen compensatoire, le taux de bonnes réponses doit être au <b>global de 75% minimum</b></p>
	<p><b>Configurer l'agrégation, le stockage et l'analyse des journaux et des mesures.</b></p>	<ul style="list-style-type: none"> <li>▶ Utiliser CloudWatch pour les tâches de Logging, mesure et monitoring.</li> <li>▶ Maîtriser les concepts de dimensions (filtres / catégories), métriques (timestamps) et namespaces (par exemple AWS/EC2 ou AWS/EBS).</li> <li>▶ Maîtriser les principes statistiques de base : minimum, maximum, somme, moyenne, compte de l'échantillon, pNN.NN)</li> <li>▶ Utiliser la partie métrique de la console pour créer un graphe, filtrer, créer des alarmes sur des seuils, changer la statistique affichée et ajouter le graphique final au Dashboard pour la surveillance.</li> <li>▶ Utiliser la console pour créer des métriques personnalisées à surveiller.</li> </ul>			



	<b>Automatiser la surveillance et la gestion des événements d'un environnement.</b>	<ul style="list-style-type: none"><li>▶ Créer différents contrôles des statuts : pour le système (perte de connectivité au réseau, perte de puissance, problèmes de logiciels sur le serveur physique, ...), pour les instances (contrôle échoué, connexion au réseau ou configuration incorrecte, mémoire épuisée, fichier système corrompu, ...)</li><li>▶ Utiliser les scripts de monitoring existant pour CloudWatch sur les instances EC2 avec Linux.</li><li>▶ Utiliser Trusted Advisor pour suivre les incidents et les problèmes à corriger en fonction de 5 paramètres : optimisation des coûts, performance, sécurité, tolérance et solidité du système, limites du service.</li></ul>			
	<b>Configurer les outils nécessaires à l'audit, à la consignation et à la surveillance des systèmes d'exploitation, des infrastructures et des applications.</b>	<ul style="list-style-type: none"><li>▶ Utiliser CloudWatch Logs pour enregistrer les logs et ainsi surveiller les API sur les instances Amazon EC2 et monitor les différentes actions.</li><li>▶ Configurer Amazon CloudWatch Events pour avoir un suivi en temps réel des événements entraînant un changement des ressources AWS. Programmer le système pour entraîner une réaction (envoi de notification, application d'un changement correctif, activation d'une fonction lambda) en fonction des événements observés.</li><li>▶ Utiliser AWS X-Ray pour collecter les données relatives aux requêtes sur les applications.</li><li>▶ Filtrer ces données pour identifier des problèmes et des opportunités d'optimisation.</li><li>▶ Utiliser CloudTrail pour la surveillance des différents API calls sur le cloud AWS afin d'identifier précisément qui modifie quoi sur l'infrastructure.</li></ul>			
	<b>Etablir les stratégies de balisage et d'établissement des autres métadonnées.</b>	<ul style="list-style-type: none"><li>▶ Procéder au balisage / tagging des éléments en respectant les pratiques recommandées : pas d'information identifiable ou confidentielle, utiliser un format standardisable sensible à la casse, utiliser plutôt trop de tags que pas assez, utiliser des outils automatisés de gestion des tags tels que AWS Resource Groups.</li></ul>			



		<ul style="list-style-type: none"> <li>▶ Adopter la catégorie de tags adéquate en fonction des situations : tags techniques (nom, application ID, rôle de l'application, cluster, version), tags pour l'automatisation, business tags (projet, owner, BI, client), tags de sécurité (confidentialité, conformité).</li> <li>▶ Connaître les limites et les prérequis des tags.</li> <li>▶ Mettre en place les stratégies les plus communes de balisage/tagging : pour l'organisation des ressources, pour l'allocation des coûts, pour l'automatisation, pour le contrôle des accès.</li> </ul>			
<b>Automatiser les politiques de suivi et de respect des normes pour assurer une infrastructure cloud optimisée</b>			<b>10% de l'évaluation globale de l'examen</b>	Examen en ligne avec une variété de questions (cf. détail plus haut) <b>** Environ 18 minutes</b> (pour 4 à 6 questions évaluées) sont consacrées à cette compétence	Examen compensatoire, le taux de bonnes réponses doit être au <b>global de 75% minimum</b>
	<b>Paramétrer les normes de journalisation, de mesure, de surveillance, de test et de sécurité</b>	<ul style="list-style-type: none"> <li>▶ Utiliser le service Identity &amp; Access Management (IAM) d'Amazon.</li> <li>▶ Utiliser AWS Trusted Advisor pour s'assurer que les ressources sont provisionnées et utilisées correctement selon cinq critères : optimisation des coûts, performance, sécurité, tolérance aux pannes, limite des services.</li> <li>▶ Renforcer la sécurité avec AWS Organization pour programmer la création de nouveaux comptes et de nouveaux groupes de comptes.</li> <li>▶ Utiliser AWS Secret Manager pour protéger les secrets nécessaires à l'accès aux applications, services et ressources IT.</li> </ul>			
	<b>Optimiser les coûts grâce à l'automatisation</b>	<ul style="list-style-type: none"> <li>▶ Suivre les recommandations d'AWS Trusted Advisor pour une optimisation des coûts.</li> <li>▶ Configurer Amazon Macie pour découvrir, classifier et protéger automatiquement via le machine learning les données sensibles d'AWS, permettant ainsi une réduction des coûts de surveillance de l'environnement AWS.</li> </ul>			
	<b>Mettre en œuvre les stratégies de gouvernance optimales</b>	<ul style="list-style-type: none"> <li>▶ Utiliser AWS Cloud Formation and AWS Service Catalog pour provisionner correctement les ressources. Utiliser AWS Service Catalog pour proposer des groupes d'application</li> </ul>			





		<p>(portfolio) que les utilisateurs peuvent utiliser sans devoir demander l'accès à la console ou aux services AWS.</p> <ul style="list-style-type: none"> <li>▶ Mener à bien les audits avec AWS CloudTrail et AWS Config.</li> <li>▶ Gérer les instances avec EC2 System Manager (Inventaire, State Manager, Parameter Store, Run Command, Patch Manager).</li> <li>▶ Monitorer, reporter et répondre aux événements avec Amazon CloudWatch.</li> <li>▶ Utiliser AWS Certificate Manager pour manager les certificats sur AWS et auditer leurs utilisations.</li> </ul>			
<b>Répondre aux incidents et aux événements qui affectent le bon fonctionnement de l'infrastructure cloud</b>			<b>18% de l'évaluation globale de l'examen</b>	Examen en ligne avec une variété de questions (cf. détail plus haut) ** <b>Environ 32 minutes</b> (pour 9 à 11 questions évaluées) sont consacrées à cette compétence	Examen compensatoire, le taux de bonnes réponses doit être au <b>global de 75% minimum</b>
	<b>Résoudre les problèmes de l'infrastructure cloud et déterminer comment restaurer les opérations.</b>	<ul style="list-style-type: none"> <li>▶ Utiliser AWS Systems Manager pour grouper les différentes ressources, voir les données opérationnelles agrégées et prendre les mesures adéquates (déployer, configurer, administrer, tracer).</li> <li>▶ Installer et gérer l'agent SSM en fonction des systèmes d'exploitations utilisés sur chaque instance.</li> <li>▶ Utiliser la console SSM pour grouper les ressources, consulter les insights (préconfigurés, dashboard CloudWatch, inventaire, conformité), automatiser certaines actions.</li> <li>▶ Patcher le SSM AWS, programmer des fenêtres de maintenance et sélections des instances à « tagger » pour application du patch.</li> <li>▶ Utiliser le distributeur pour utiliser des packages de logiciel créés ou distribués par des agents AWS.</li> </ul>			
	<b>Automatiser la gestion des événements et les alertes.</b>	<ul style="list-style-type: none"> <li>▶ Utiliser Amazon CloudTrail pour configurer des « trails », créer un espace de stockage pour les enregistrements des logs.</li> <li>▶ Utiliser AWS CloudWatch pour configurer des alarmes.</li> <li>▶ Ajouter des notifications automatiques via les notifications SNS avec CloudWatch via les commandes CreateLogStream et PutLogEvents.</li> </ul>			



	<b>Mettre en œuvre la guérison automatisée de l'infrastructure cloud.</b>	<ul style="list-style-type: none"> <li>▶ Utiliser Auto Scaling pour créer des groupes ou utiliser des templates de lancement, provisionner automatiquement et ajuster les capacités en instaurant un nombre minimum d'instance à maintenir en ligne pour en lancer automatique de nouvelles en cas de sinistre sur certaines.</li> <li>▶ Utiliser l'historique d'activités et filtrer les instances par statut et état de sante pour suivre la guérison automatisée.</li> </ul>			
	<b>Mettre en place des actions automatisées axées sur les événements.</b>	<ul style="list-style-type: none"> <li>▶ Utiliser CloudWatch Events pour répondre automatiquement aux changements d'état des ressources AWS.</li> <li>▶ Créer des règles pour : invoquer automatiquement une fonction AWS Lambda mettant à jour les entrées DNS quand une instance EC2 est lancée, diriger les enregistrements d'APIs de CloudTrail a un stream Kinesis spécifique pour analyse des risques de sécurité et de disponibilité ou encore capture un snapshot d'un volume EBS a un moment donné.</li> <li>▶ Plus généralement, créer des règles en fonction suivant des modèles d'évènements pour mettre en place des actions automatisées axées sur les évènements.</li> </ul>			
<b>Assurer la haute disponibilité, la tolérance aux pannes et la reprise après sinistre de votre infrastructure cloud</b>					
	<b>Déterminer l'utilisation appropriée des architectures multi-AZ par rapport aux architectures multi-régions.</b>	<ul style="list-style-type: none"> <li>▶ Développer un discours technique cohérent sur l'architecture multi-région assure une latence minimum, une reprise rapide après sinistre et une conformité par rapport aux règles locales.</li> <li>▶ Utiliser les services AWS pour paliers aux problèmes de cohérence d'un tel choix (réplication des données synchrone ou asynchrone, déploiement blue green) : réplication interrégional via S3, connexion entre VPC interrégionaux, lecture des répliques de RDS, VPN CloudHub, Database Migration Service, gestion du trafic via Route 53.</li> <li>▶ Appliquer les différents principes de l'AWS Well-Architected Framework (Excellence opérationnelle, sécurité, fiabilité, efficience de la performance, optimisation des couts).</li> </ul>	<b>16% de l'évaluation globale de l'examen</b>	Examen en ligne avec une variété de questions (cf. détail plus haut) ** <b>Environ 29 minutes</b> (pour 8 à 10 questions évaluées) sont consacrées à cette compétence	Examen compensatoire, le taux de bonnes réponses doit être au <b>global de 75% minimum</b>



		<ul style="list-style-type: none"><li>▶ Développer un discours technique cohérent sur déploiement multi-AZ : conçus en prévision des désastres, hautement disponible, basculement automatique, réplication synchrone des données.</li><li>▶ Statuer sur quel mode de déploiement choisir en fonction d'une situation donnée.</li></ul>			
	<b>Mettre en œuvre la haute disponibilité, l'évolutivité et la tolérance aux pannes.</b>	<ul style="list-style-type: none"><li>▶ Utiliser Elastic Load Balancer (Application Load Balancer, Network Load Balancer, Classic Load balancer) pour assurer une haute disponibilité des applications via les groupes de mise à l'échelle automatique.</li><li>▶ Utiliser Amazon CloudFront pour assurer une haute disponibilité au plus près de l'utilisateur : contenu dynamique et statique, service via les edge locations, stockage de domaine personnalisé, utilisation d'AWS Shield pour prévenir les attaques DDos, paramétrage et intégration des pare-feu avec AWS WAF.</li><li>▶ Utiliser les services d'application stateless pour garantir la plus haute disponibilité : utilisation du load balancer en interne dans le cloud avec adresse IP privée, découpler les files avec Amazon SAS.</li></ul>			
	<b>Déterminer les bons services en fonction des besoins de l'entreprise (p. ex., RTO/RPO, coût).</b>	<ul style="list-style-type: none"><li>▶ Déterminer la stratégie de scaling la plus adaptée aux besoins de l'entreprise : scaling verticale ou horizontale.</li><li>▶ Savoir implémenter le scaling vertical dans les services AWS utilisant cette stratégie (par exemple RDS), le scaling horizontal dans les groupes d'auto-scaling</li></ul>			
	<b>Concevoir et automatiser les stratégies de reprise après sinistre.</b>	<ul style="list-style-type: none"><li>▶ Utiliser l'autoscaling pour permettre une reprise après sinistre plus rapide avec l'utilisation de nouvelles instances suite à l'échec de certaines.</li><li>▶ Utiliser Amazon Route53 en tant que DNS pour rerouter le trafic en cas d'avarie sur une ou plusieurs parties du domaine.</li><li>▶ Au niveau des bases de données, paramétrer RDS et les schémas de bases de données utilisées pour tirer profiter des</li></ul>			



		backups, mises à jour et reprises après sinistres automatiques.			
	<b>Évaluer un déploiement pour les points de défaillance.</b>	<ul style="list-style-type: none"><li>▶ Evaluer les « liens faibles » dans la situation.</li><li>▶ Identifier les mauvaises implémentations dans les réponses.</li></ul>			