Déploiement Expertise de SAS Viya 4 TLS 2025.03 sur Red Hat OpenShift

Le présent rapport fournit une explication détaillée et experte des étapes et des considérations architecturales critiques nécessaires au déploiement de la plateforme SAS Viya 4 dans sa version Long-Term Support (LTS) 2025.03 sur un environnement Red Hat OpenShift. Ce processus exige une adhésion stricte aux prérequis de compatibilité, la gestion des contraintes de sécurité spécifiques à OpenShift (Security Context Constraints ou SCC), et la mise en œuvre de configurations personnalisées via Kustomize.

Section 1: Fondations Architecturales et Contexte de la Version TLS 2025.03

Le déploiement de SAS Viya 4 est intrinsèquement lié à la matrice de support du fournisseur de conteneurs sous-jacent. La version TLS 2025.03, qui intègre les correctifs et les fonctionnalités de la version Stable 2025.03 (Mars 2025) [1], introduit des exigences et des modifications critiques qui doivent être traitées avant l'exécution du manifeste de déploiement.

1.1. Matrice de Compatibilité Red Hat OpenShift pour TLS 2025.03

La plateforme SAS Viya 4 LTS 2025.03 a élargi son support pour les plateformes Red Hat.[2] Il est impératif que l'infrastructure OpenShift Container Platform (OCP) ou OpenShift Kubernetes Engine (OKE) respecte les versions validées par SAS.

Les versions prises en charge pour le déploiement de 2025.03 et des versions ultérieures sont **OCP ou OKE 4.16.x jusqu'à 4.18.x**.[2] Cette plage de support est alignée sur les versions sous-jacentes de Kubernetes 1.29.x, 1.30.x, et 1.31.x.[1] La vérification de cette compatibilité est une étape non négociable, car l'exécution sur des versions non supportées (telles que des

versions OCP trop anciennes ou trop récentes non encore validées) pourrait entraîner des défaillances de déploiement ou l'annulation du support technique.

1.2. Modifications Critiques du Déploiement dans la Version 2025.03

Plusieurs changements opérationnels et fonctionnels dans la version 2025.03 requièrent une attention particulière et une planification anticipée.

Mise à Jour Critique du Serveur CAS

La version 2025.03 apporte une mise à jour au serveur CAS (Cloud Analytic Services). Cependant, la mise à jour ne prend effet que lorsque les pods CAS existants sont redémarrés.[5] Après l'application des commandes de déploiement (Kustomize ou sas-orchestration), l'administrateur doit manuellement supprimer les pods CAS existants pour forcer le redémarrage et l'exécution de la nouvelle version.[5]

Il convient de noter que cette action génère des conséquences opérationnelles significatives. La suppression des pods CAS arrête les serveurs CAS en cours d'exécution, ce qui entraîne la perte de toutes les données chargées en mémoire (*in-memory data*). Il en résulte une interruption de service visible par les utilisateurs finaux. Par conséquent, cette étape critique de post-déploiement doit faire l'objet d'une planification rigoureuse pour minimiser l'impact.[5] Les serveurs CAS mis à jour devront être rechargés en données après leur redémarrage.[5]

Modifications d'Ingress et de Réseau

Les notes de déploiement mentionnent explicitement des modifications critiques concernant les exigences d'Ingress.[2] Dans un environnement OpenShift, ceci implique nécessairement de s'éloigner des objets Ingress natifs de Kubernetes et d'utiliser les **Routes OpenShift** pour l'exposition externe.[6] Cela nécessite une modification du manifeste Kustomize de base, comme détaillé dans la Section 5.

Mises à Jour des Bases de Données et des Connecteurs

La version 2025.03 intègre des mises à jour importantes concernant l'accès aux données. Premièrement, elle inclut désormais les pilotes Simba pour les logiciels SAS/ACCESS pour l'accès aux bases de données, en remplacement des anciens pilotes CData pour Hadoop et Spark, ce qui peut nécessiter une configuration supplémentaire.[3]

Deuxièmement, il existe une modification critique concernant l'instance embarquée de SingleStoreDB (appelée maintenant SAS SpeedyStore).[1] L'instance est mise à jour de la version 8.7 à 8.9.[3] Ce changement est **non rétrocompatible**. Une fois la mise à jour vers 2025.03 effectuée, il est impossible de revenir à la version 8.7 ou antérieure de SingleStoreDB. Si une annulation (rollback) vers une version antérieure est nécessaire, l'intégralité de la plateforme SAS Viya doit être désinstallée et redéployée avec la version précédente.[3] Cette non-rétrocompatibilité élève l'opération de mise à jour à un événement de migration de données, justifiant une procédure de sauvegarde/restauration validée avant l'engagement du déploiement.

Limitation du Serveur SMTP

Pour les déploiements effectués uniquement dans des environnements Google Cloud, le support d'un serveur SMTP facultatif utilisant l'Authentification Basique a été temporairement supprimé, car Google a cessé de prendre en charge ce mode d'authentification pour Gmail en Mars 2025. SAS prévoit de restaurer ce support dans une future version qui intégrera OAuth 2.0.[2]

Le Tableau 1.1 résume les impératifs de compatibilité et les risques opérationnels majeurs.

Tableau 1.1: Compatibilité et Exigences Opérationnelles Critiques pour Viya 4 TLS 2025.03

Version SAS Viya 4	Kubernetes Supporté	Versions OCP/OKE Supportées	Note Opérationnelle Critique
TLS 2025.03	1.29.x – 1.31.x	OCP/OKE 4.16.x – 4.18.x	Redémarrage manuel des pods CAS requis après le

			déploiement (cause une panne planifiée).[5]
TLS 2025.03	N/A	N/A	Mise à jour non rétrocompatible de SingleStoreDB à 8.9.x.[3]

Section 2: Prérequis d'Infrastructure OpenShift (Stockage et Calcul)

Une configuration d'infrastructure appropriée est essentielle pour garantir à la fois la performance et la persistance des données SAS Viya.

2.1. Exigences Relatives au Stockage Persistant et aux Modes d'Accès

SAS Viya exige que l'environnement OpenShift dispose de classes de stockage définies qui répondent à des modes d'accès spécifiques, car la plateforme est fondamentalement *stateful* et repose sur des systèmes de fichiers partagés pour la coordination des microservices et la persistance des données.

Une classe de stockage en mode **ReadWriteMany (RWX)** doit être définie et disponible, et il est recommandé qu'elle soit définie comme le défaut de l'environnement.[7] Le mode RWX permet à plusieurs nœuds de travail de monter le même volume simultanément, ce qui est indispensable pour plusieurs composants critiques de Viya.

- **Serveur CAS:** Le CAS, qu'il soit déployé en mode SMP (Symmetric Multi Processing) ou MPP (Massive Parallel Processing), est configuré par défaut pour utiliser le mode RWX pour ses permstores et ses volumes de données.[8] En mode MPP, le serveur CAS est distribué sur plusieurs pods/nœuds et requiert un stockage persistant partagé pour son fonctionnement.[6]
- **PostgreSQL Interne:** L'instance interne par défaut de PostgreSQL nécessite un mode **ReadWriteOnce (RWO)**. Ce stockage doit être de type bloc et de préférence basé sur

SSD pour des performances optimales (128 Go par nœud minimum).[8]

- **Sauvegarde et Surveillance:** Les volumes de sauvegarde (sas-common-backup-data et sas-cas-backup-data) nécessitent explicitement le mode RWX. De même, le SAS Configurator for Open Source nécessite un stockage persistant en mode RWX (20 Go minimum).[8]

Des problèmes de performance fréquents signalés par les clients Viya sont directement attribués à des niveaux de débit d'E/S (Input/Output) insuffisants du stockage.[7] Pour les composants à haute performance comme le CAS, l'utilisation de SSDs ou de stockages Premium spécifiques au cloud est requise pour atteindre les niveaux de débit nécessaires.[7] La capacité minimale combinée des nœuds de travail Kubernetes pour les images et la journalisation doit être de 48 Go.[7]

Tableau 2.1: Volumes de Stockage Persistant Requis et Modes d'Accès

Composant/Volume	**Taille Minimale par Défaut**	**Mode d'Accès Requis**	**Type de Stockage Recommandé**
Serveur CAS (Données/Permstore)	8 Gi / 100 Mi	RWX (Par Défaut)	Stockage partagé à haute performance (NFS/EFS)
PostgreSQL Interne	128 GB (par nœud)	RWO	Stockage Bloc (SSD recommandé)
Sauvegarde Générale (sas-common-backup-data)	25 GB	RWX	Retain Policy recommandée [8]

2.2. Stratégie de Nœuds et Allocation des Ressources

Le serveur CAS est l'un des composants Viya les plus exigeants en ressources CPU et mémoire.[6] Il est vivement recommandé d'utiliser un *node pool* dédié au CAS. Cette configuration permet aux pods CAS de fonctionner sur des nœuds isolés, exploitant plus de 90% des ressources disponibles par défaut.[6]

Si un pool de nœuds dédié n'est pas possible, des *transformer patches* Kustomize doivent être appliqués lors du déploiement afin de limiter l'utilisation des ressources du CAS, permettant ainsi une cohabitation avec d'autres charges de travail sur le même nœud.[6]

Les services de calcul (SAS Compute Services) représentent les capacités de traitement SAS traditionnelles, très parallélisables. Les nœuds de calcul constituent un excellent candidat pour l'utilisation de l'autoscaler de cluster, permettant d'adapter dynamiquement la capacité aux pics de charge (par exemple, les traitements de lots nocturnes).[6]

Section 3: Gestion des Contraintes de Contexte de Sécurité (SCC) OpenShift

Dans un environnement Red Hat OpenShift, la sécurité des conteneurs est gérée par les Security Context Constraints (SCC). La plateforme SAS Viya nécessite des autorisations de sécurité qui vont au-delà des capacités offertes par la SCC restricted par défaut d'OpenShift.[9]

3.1. Nécessité et Procédure d'Application des SCC Personnalisées

Les composants Viya, en particulier ceux qui montent des volumes de fichiers (comme l'exigent les volumes RWX [10]) ou ceux qui accèdent à des dispositifs de stockage spécifiques, ne peuvent pas s'exécuter avec les contraintes par défaut.[9]

L'administration du cluster (nécessitant des privilèges élevés de type cluster-admin) doit effectuer deux tâches principales : l'application de la définition de la SCC et la liaison de cette SCC aux comptes de service des pods Viya. Le processus implique l'utilisation des fichiers YAML fournis dans les ressources de déploiement Viya ($deploy/sas-bases/examples/security/container-security/).

- **Application de la SCC (Création):** Le fichier de définition YAML est appliqué via kubectl apply -f [chemin_scc_yaml].[11]
- **Liaison de la SCC au Compte de Service:** Une fois la SCC créée, elle doit être liée au compte de service du composant Viya concerné dans le *namespace* cible, en utilisant la commande oc adm policy add-scc-to-user fournie par l'outil CLI OpenShift (oc).[12]

Par exemple, pour lier la SCC de détection à son compte de service dans le *namespace* de

déploiement :

```bash
oc -n name-of-namespace adm policy add-scc-to-user sas-detection-definition -z sas-detection-definition
```

3.2. Configuration Spécifique du Serveur CAS

Le serveur CAS nécessite une attention particulière en matière de sécurité. L'administrateur du cluster doit modifier la SCC de CAS pour permettre les montages **hostPath**.[13] Cette autorisation est cruciale, car elle permet à CAS d'utiliser le disque temporaire local du nœud de travail pour des opérations à haute vitesse, ce qui est une optimisation de performance.

De plus, des modifications plus générales des paramètres de sécurité des conteneurs doivent être effectuées. Cela inclut la suppression du profil Secure Computing (seccomp) et la mise à jour du champ fsGroup.[13] Les instructions détaillées pour ces modifications sont disponibles dans le fichier README.md situé dans $deploy/sas-bases/examples/security/container-security/.[13]

Cette phase de gestion des SCC est un point de dépendance organisationnelle majeur. Elle nécessite une collaboration étroite entre l'équipe SAS et l'équipe d'administration OpenShift, car la plupart des opérations SCC requièrent des privilèges au niveau du cluster.

Tableau 3.1: Configuration Obligatoire des SCC OpenShift

Composant SAS	Modification SCC Requise	Niveau de Privilège	Exemple de Liaison (Binding)
Serveur CAS	SCC personnalisée permettant hostPath et ajustements seccomp/fsGroup.[13]	Cluster Administrator	oc adm policy add-scc-to-user sas-cas-server -z sas-cas-server

Services de Détection	Application et liaison de la SCC sas-detection-definition.[11]	Cluster Administrator	oc -n namespace adm policy add-scc-to-user sas-detection-definition -z sas-detection-definition

Section 4: Préparation des Ressources de Déploiement et Mise en Miroir des Images

Une fois l'infrastructure prête et les exigences de sécurité satisfaites, l'étape suivante consiste à récupérer et à préparer les conteneurs d'images logicielles nécessaires au déploiement.

4.1. Récupération des Artefacts de Déploiement

Les artefacts de déploiement SAS Viya 4 (fichiers YAML, certificats, instructions) sont requis pour le déploiement initial et toutes les mises à jour ultérieures.[14] Ils peuvent être obtenus de deux manières principales :

1. Téléchargement direct via le portail My SAS.
2. Utilisation de l'API ou de la CLI SAS Viya Orders pour automatiser le téléchargement des actifs.[14]

4.2. Stratégie du Registre de Conteneurs : Mise en Miroir

Dans les environnements OpenShift sécurisés (souvent déconnectés ou avec des politiques d'accès externes strictes), le déploiement nécessite que les images de conteneurs soient mises en miroir depuis le registre SAS vers un registre privé accessible au cluster OpenShift, généralement le registre interne d'OpenShift Container Registry.[16]

L'outil **SAS Mirror Manager** (mirrormgr) est utilisé pour ce processus. Une exigence technique cruciale unique à l'environnement OpenShift est l'utilisation obligatoire du drapeau **--flatten** lors de la mise en miroir. Ce drapeau garantit que la structure des images est compatible avec la gestion du registre interne d'OpenShift.[16] Omettre cette option entraînerait des échecs généralisés de *pull* d'images à travers le déploiement.

La syntaxe de la commande de mise en miroir doit ressembler à l'exemple suivant:

Bash

```
./mirrormgr mirror registry \
--destination openshift-image-registry-URL/namespace \
--flatten \
--deployment-data SASViyaV4_order-ID_certs.zip \
--latest --username user-name --password $password \
--path /path-to-local-image-cache
```

4.3. Configuration des Secrets d'Accès aux Images (Image Pull Secrets)

Pour que les pods Viya puissent extraire les images du registre privé miroir, les informations d'identification doivent être fournies via un secret Kubernetes de type kubernetes.io/dockercfg.[13]

Après avoir créé le secret d'extraction (par exemple, myregistrykey), il doit être lié aux comptes de service qui lancent les pods dans le *namespace* de déploiement. Sur OpenShift, cela se fait avec la commande oc secrets add.[19]

Exemple de liaison du secret d'extraction au compte de service default :

Bash

```
oc secrets add serviceaccount/default secrets/<pull_secret_name> --for=pull
```

Cette étape est fondamentale pour garantir que tous les pods Viya disposent de l'autorisation nécessaire pour démarrer, sans quoi ils resteront bloqués en statut ImagePullBackOff.

Section 5: Personnalisation du Manifeste de Déploiement via Kustomize

SAS Viya 4 utilise l'outil Kustomize pour gérer la configuration de déploiement. Le processus consiste à appliquer des *overlays* spécifiques au site (placés dans un répertoire tel que site-config) aux modèles de base fournis par SAS (sas-bases).[20] Ces personnalisations doivent intégrer les spécificités de l'environnement OpenShift.

5.1. Intégration de la Configuration de Mise en Miroir

Pour que le déploiement utilise les images mises en miroir dans le registre interne OpenShift, la configuration de mise en miroir doit être appliquée comme un *transformer* Kustomize.[13]

1. Copier le fichier mirror-flattened.yaml (ajusté avec l'URL du registre miroir) dans le répertoire site-config.
2. Modifier ce fichier pour remplacer la variable {{ MIRROR-HOST }} par l'itinéraire interne du service du registre OpenShift. Le format typique est service-name.name-of-registry-namespace.svc:port/platform-namespace.[11]
3. Ajouter le chemin d'accès à mirror-flattened.yaml dans le bloc transformers du fichier kustomization.yaml principal, immédiatement après l'entrée pour sas-bases/overlays/required/transformers.yaml.[13]

5.2. Transformation Réseau pour les Routes OpenShift

OpenShift utilise des objets Route pour l'Ingress, qui ne sont pas pris en compte dans le kustomization.yaml initial conçu pour Kubernetes standard.[22] Une modification du fichier

kustomization.yaml est requise pour pointer vers les ressources OpenShift-natives.

La section resources du fichier de base doit être révisée pour remplacer la référence aux ressources réseau standard de Kubernetes, assurant ainsi que les routes OpenShift sont créées pour exposer les applications Viya.[22]

5.3. Personnalisation du Stockage et des SCC

Les *overlays* Kustomize doivent également contenir les patchs nécessaires pour définir les StorageClasses appropriées (RWX et RWO) conformément aux exigences de la Section 2.1. De plus, bien que les SCC soient des ressources au niveau du cluster, leur utilisation et l'intégration des ajustements de sécurité pour les composants Viya (y compris les transformations pour TLS) sont souvent incluses dans les patchs Kustomize afin d'assurer l'uniformité du déploiement.[21]

Il est essentiel de comprendre que même si l'exécution finale est automatisée (via l'Operator ou sas-orchestration), la création manuelle et la validation de ces fichiers Kustomize restent la partie la plus critique et la plus chronophage du déploiement, car elles définissent l'alignement entre les exigences SAS Viya et les spécificités d'OpenShift.

Section 6: Exécution du Manifeste de Déploiement

Une fois le manifeste de déploiement généré et personnalisé via Kustomize, l'exécution peut être réalisée selon différentes méthodes prises en charge par SAS.[6]

6.1. Options de Déploiement

Trois approches principales sont disponibles pour déployer SAS Viya sur OpenShift [6]:

1. **Déploiement Manuel:** Utilisation directe de l'outil Kustomize pour construire le manifeste final, suivi de l'application des ressources via des commandes kubectl ou oc.[20]
2. **Opérateur de Déploiement SAS Viya:** L'installation d'un opérateur qui surveille une

ressource personnalisée (SASDeployment CR) et gère le cycle de vie de la plateforme.[23]
3. **Utilitaire CLI sas-orchestration:** Un outil conteneurisé qui combine les étapes de récupération des actifs, de construction Kustomize et d'application du manifeste en une seule commande.[6]

6.2. Procédure d'Exécution Manuelle ou via sas-orchestration

Dans de nombreux environnements d'entreprise exigeant une traçabilité et une ségrégation des privilèges, l'approche manuelle ou l'utilisation de sas-orchestration sont privilégiées.

Exécution Manuelle (Kustomize et OC CLI):
Le manifeste final est généré et appliqué directement au cluster OpenShift.

1. Génération du manifeste final : kustomize build -k site-config > site.yaml.[21]
2. Application du manifeste : oc apply -f site.yaml (ou application granulaire par type d'objet Kubernetes).[20]

Exécution via sas-orchestration:
Cet outil simplifie l'exécution en consolidant les étapes de construction et d'application.6

1. Extraction de l'image CLI sas-orchestration.
2. Exécution de la commande de déploiement : sas-orchestration deploy.[20]

Il est important de noter que l'utilisation de l'utilitaire sas-orchestration pour le déploiement nécessite souvent des privilèges élevés (potentiellement cluster-admin) pour gérer les ressources au niveau du cluster, telles que les définitions de ressources personnalisées (CRD) et les SCC.[6] Dans les organisations où la séparation des tâches est stricte, l'approche manuelle permet à l'administrateur OpenShift de gérer les ressources privilégiées (SCC, CRD) tandis que l'équipe SAS gère les ressources limitées au *namespace*.[6]

Section 7: Opérationnalisation et Étapes Critiques Post-Déploiement (TLS 2025.03)

Le déploiement technique de la plateforme se termine par l'application réussie du manifeste, mais l'opérationnalisation complète, en particulier pour la version 2025.03, nécessite des

étapes de validation et d'activation finales.

7.1. Vérification Initiale de l'État du Cluster

Une fois l'application du manifeste terminée, une série de vérifications initiales est nécessaire :

- **Vérification des Pods:** S'assurer que tous les pods des microservices Viya principaux sont en statut Running et prêts.
- **Vérification des Routes:** S'assurer que les Routes OpenShift ont été créées avec succès et qu'elles exposent correctement les applications web Viya à l'accès externe.[6] En l'absence des routes, les services resteront inaccessibles malgré le bon fonctionnement des pods.
- **Vérification de la Base de Données Externe:** En cas d'utilisation d'une instance PostgreSQL externe, il faut surveiller les logs pour éviter les erreurs de connexion de type GenericJDBCException: Unable to acquire JDBC Connection, qui peuvent indiquer des problèmes de dimensionnement, de capacités de connexion ou d'authentification du serveur PostgreSQL.[22]

7.2. Étape CRITIQUE : Finalisation de la Mise à Jour du Serveur CAS

Comme mentionné dans la Section 1.2, l'achèvement de la mise à jour vers la version 2025.03 dépend de l'application manuelle du nouveau serveur CAS. Cette étape est obligatoire et planifiée comme une interruption de service.

Procédure:
L'administrateur, disposant des privilèges de namespace, doit exécuter la commande suivante pour supprimer les pods CAS existants et provoquer leur redémarrage avec la nouvelle version:

Bash

```
kubectl -n name-of-namespace delete pods -l
app.kubernetes.io/managed-by=sas-cas-operator
```

Planification de l'Interruption: La suppression de ces pods entraîne l'arrêt du service CAS et la perte immédiate de toutes les données chargées en mémoire.[5] Le service ne sera restauré qu'après le redémarrage réussi des nouveaux pods CAS mis à jour et le rechargement des données nécessaires.[5] Cette manœuvre opérationnelle nécessite une notification des utilisateurs, l'arrêt des sessions en cours et la pause des pipelines d'ingestion de données pour gérer l'interruption de service.

7.3. Dépannage Général des Problèmes Courants sur OpenShift

Lors du déploiement sur OpenShift, les échecs courants sont souvent liés à des erreurs de configuration de l'infrastructure ou de sécurité :

- **Échec de l'Extraction d'Images (ImagePullBackOff):** Si ce statut persiste, cela indique généralement un problème d'authentification (le secret d'extraction d'image est manquant ou incorrectement lié aux comptes de service) ou une mauvaise référence au registre miroir (vérifier la configuration du mirror-flattened.yaml et l'utilisation du drapeau --flatten).[16]
- **Erreurs de Permission (Security Denials):** Les pods ne parvenant pas à démarrer en raison de restrictions de sécurité nécessitent une vérification que toutes les SCC personnalisées (en particulier pour CAS) ont été correctement appliquées au niveau du cluster et liées aux comptes de service pertinents dans le *namespace* de déploiement via oc adm policy add-scc-to-user.[11]

Tableau 7.1: Actions Opérationnelles Cruciales Post-Déploiement TLS 2025.03

Tâche	Action Requise	Implication Opérationnelle
Achèvement Mise à Jour CAS	Exécuter la suppression des pods CAS via kubectl delete.[5]	Coupure de service et perte de données en mémoire. Nécessite une planification BCP (Business Continuity Plan).[5]
Vérification SingleStore	Confirmer le fonctionnement de SingleStore DB en version	Validation finale d'une mise à jour non rétrocompatible.[3]

	8.9.x.	

Conclusions et Recommandations

Le déploiement de SAS Viya 4 TLS 2025.03 sur Red Hat OpenShift est une entreprise complexe qui exige une approche méthodique de configuration-as-code et une forte dépendance aux outils natifs d'OpenShift. Le processus se distingue des déploiements Kubernetes génériques par trois impératifs majeurs :

1. **Gestion des Privilèges de Sécurité:** La nécessité d'appliquer et de lier des SCC personnalisées, notamment pour autoriser les montages hostPath pour CAS, crée une dépendance critique vis-à-vis des administrateurs du cluster possédant des privilèges élevés. Cette phase doit être traitée en amont pour éviter tout blocage du déploiement.
2. **Configuration d'Ingress et de Registre:** La compatibilité avec OpenShift nécessite des modifications non triviales des manifestes Kustomize pour remplacer l'Ingress par les Routes et pour intégrer le registre d'images en utilisant impérativement l'option --flatten lors de la mise en miroir. Une erreur dans ces configurations empêche l'accès externe ou le démarrage des conteneurs.
3. **Gestion des Changements Critiques (CAS et SingleStore):** La version 2025.03 contient deux changements qui transforment l'opération d'une simple mise à jour logicielle en un événement de gestion de continuité d'activité (BCP). L'administrateur doit planifier et exécuter manuellement la suppression des pods CAS post-déploiement, ce qui implique une coupure de service planifiée et le rechargement subséquent des données en mémoire. Simultanément, la non-rétrocompatibilité de la mise à jour SingleStore exige une stratégie de récupération robuste en cas de besoin de retour arrière.

Il est recommandé que l'équipe de déploiement utilise le modèle d'exécution qui permet la meilleure ségrégation des tâches administratives, souvent le flux Kustomize manuel, permettant aux administrateurs OpenShift de gérer les ressources *cluster-scoped* (SCC, CRD) et à l'équipe SAS de se concentrer sur les ressources au niveau du *namespace*.

Works cited

1. LTS 2025.03 (May 2025) - SAS Help Center, accessed November 4, 2025, https://documentation.sas.com/doc/en/itopswn/latest/p15mdx7we3dm4on1eu5yi0r630b2.htm
2. 2025.03 (2025 年3 月) - SAS Help Center, accessed November 4, 2025, http://documentation.sas.com/doc/en/itopscdc/v_069/itopswn/n09xbra4uccw52n

1vftp3v6agdc9.htm
3. 2025.03 (March 2025) - SAS Help Center, accessed November 4, 2025, https://documentation.sas.com/doc/no/sasadmincdc/v_042/itopswn/n09xbra4uccw52n1vftp3v6agdc9.htm
4. Virtual Infrastructure Requirements - SAS Help Center, accessed November 4, 2025, http://documentation.sas.com/doc/ja/sasadmincdc/v_068/itopssr/n1ika6zxghgsoqn1mq4bck9dx695.htm
5. 2025.03 (March 2025) - SAS Help Center, accessed November 4, 2025, https://documentation.sas.com/doc/en/dplynotes/latest/p0air5hmeo3o7pn1duatauyh28hy.htm
6. SAS Viya on Red Hat OpenShift – Part 1: Reference Architecture and Deployment Considerations, accessed November 4, 2025, https://www.redhat.com/en/blog/sas-viya-on-red-hat-openshift-part-1-reference-architecture-and-deployment-considerations
7. System Requirements for SAS Viya, accessed November 4, 2025, https://documentation.sas.com/api/collections/itopscdc/v_022/docsets/itopssr/content/itopssr.pdf?locale=en
8. Hardware and Resource Requirements - SAS Help Center, accessed November 4, 2025, http://documentation.sas.com/doc/it/itopscdc/v_069/itopssr/n0ampbltwqgkjkn1j3qogztsbbu0.htm
9. Chapter 13. Managing security context constraints | Authentication and authorization | OpenShift Dedicated | 4 | Red Hat Documentation, accessed November 4, 2025, https://docs.redhat.com/en/documentation/openshift_dedicated/4/html/authentication_and_authorization/managing-pod-security-policies
10. System Requirements for the SAS Viya Platform, accessed November 4, 2025, https://documentation.sas.com/api/collections/itopscdc/v_050/docsets/itopssr/content/itopssr.pdf?locale=hi
11. Optional Customizations - SAS Help Center, accessed November 4, 2025, https://documentation.sas.com/doc/de/itopscdc/v_057/dplyml0phy0dkr/n08u2yg8tdkb4jn18u8zsi6yfv3d.htm
12. SAS Viya on Red Hat OpenShift – Part 2: Security and Storage Considerations, accessed November 4, 2025, https://www.redhat.com/en/blog/sas-viya-on-red-hat-openshift-part-2-security-and-storage-considerations
13. Optional Customizations - SAS Help Center, accessed November 4, 2025, https://documentation.sas.com/doc/de/sasadmincdc/v_067/dplyml0phy0dkr/n08u2yg8tdkb4jn18u8zsi6yfv3d.htm
14. Deployment Assets - SAS Help Center, accessed November 4, 2025, http://documentation.sas.com/doc/it/itopscdc/v_043/itopscon/n08bpieatgmfd8n192cnnbqc7m5c.htm
15. sassoftware/viya4-deployment: This project contains Ansible code that creates a baseline in an existing Kubernetes environment for use with the SAS Viya

Platform, generates the manifest for an order, and then can also deploy that order into the Kubernetes environment specified. - GitHub, accessed November 4, 2025, https://github.com/sassoftware/viya4-deployment
16. Create and Populate a Mirror in Red Hat ... - SAS Help Center, accessed November 4, 2025, https://documentation.sas.com/doc/nl/itopscdc/v_050/dplyml0phy0dkr/p1dmjbculxbjvkn1b64o5vxp2rpp.htm
17. Create and Populate a Mirror in Red Hat OpenShift Container Registry - SAS Help Center, accessed November 4, 2025, https://documentation.sas.com/doc/pt-br/sasadmincdc/v_057/dplyml0phy0dkr/p1dmjbculxbjvkn1b64o5vxp2rpp.htm
18. Creating imagePullSecrets for a specific namespace - IBM, accessed November 4, 2025, https://www.ibm.com/docs/en/cloud-private/3.2.x?topic=images-creating-imagepullsecrets-specific-namespace
19. Chapter 15. Image Pull Secrets | Developer Guide | OpenShift Container Platform | 3.1, accessed November 4, 2025, https://docs.redhat.com/fr/documentation/openshift_container_platform/3.1/html/developer_guide/dev-guide-image-pull-secrets
20. New SAS Viya Deployment Methods - SAS Communities, accessed November 4, 2025, https://communities.sas.com/t5/SAS-Communities-Library/New-SAS-Viya-Deployment-Methods/ta-p/856206
21. SAS Viya on Red Hat OpenShift Service for AWS (ROSA), accessed November 4, 2025, https://www.redhat.com/en/blog/sas-viya-red-hat-openshift-service-aws-rosa
22. SAS Viya Platform: Deployment Guide, accessed November 4, 2025, https://documentation.sas.com/api/collections/itopscdc/v_039/docsets/dplyml0phy0dkr/content/dplyml0phy0dkr.pdf?locale=en
23. Introducing the SAS Viya Deployment Operator - SAS Communities, accessed November 4, 2025, https://communities.sas.com/t5/SAS-Communities-Library/Introducing-the-SAS-Viya-Deployment-Operator/ta-p/749770