

# Kubescape for Teenagers

**LA SURVEILLANCE DE VOS  
CLUSTERS KUBERNETES**



# SOMMAIRE

---

|                                     |    |
|-------------------------------------|----|
| Introduction                        | 3  |
| Installation de Kubescape - Local   | 4  |
| Installation de Kubescape - Cluster | 5  |
| Scan d'un cluster                   | 6  |
| Interpréter les résultats           | 7  |
| Frameworks                          | 9  |
| Accepting risk                      | 10 |
| Scan des Images                     | 12 |
| Scan d'un déploiement               | 13 |
| Scan d'un Helm                      | 14 |
| Formats de Résultats                | 15 |
| Surveillance d'un cluster           | 16 |
| Dashboard ARMO                      | 18 |
| Attack Path                         | 19 |
| Repository Scanning                 | 20 |

# Introduction



Kubescape est l'outil de préférence pour tester la sécurité des environnements Kubernetes, offrant une analyse complète pour détecter les configurations à risque selon plusieurs frameworks de sécurité, notamment ceux recommandés par le National Security Agency (NSA) et le Cybersecurity and Infrastructure Security Agency (CISA). Il est conçu pour identifier les vulnérabilités et les mauvaises configurations qui pourraient mettre en péril la sécurité des applications conteneurisées et de l'infrastructure Kubernetes.

Kubescape a été développé par ARMO, une entreprise spécialisée dans la sécurité des applications cloud-native. Au fil du temps, la gestion du projet Kubescape a été transférée à la Cloud Native Computing Foundation (CNCF), une fondation sous l'égide de la Linux Foundation qui soutient des projets open source liés au cloud computing. Ce transfert s'inscrit dans une volonté de placer Kubescape sous une gouvernance ouverte et de favoriser sa croissance au sein de la communauté des technologies cloud-native, en assurant une neutralité et une transparence accrues dans son développement.

L'utilisation de Kubescape est essentielle pour plusieurs raisons :

- Sécurité renforcée : Il aide à détecter et à corriger les configurations à risque et les vulnérabilités dans les clusters Kubernetes, contribuant ainsi à une infrastructure plus sûre.
- Conformité : Kubescape évalue la conformité de vos environnements Kubernetes par rapport aux standards de sécurité reconnus, facilitant l'adoption de meilleures pratiques et la conformité réglementaire.

# Installation de Kubescape - Local



L'installation locale permet d'exécuter Kubescape depuis n'importe quel poste de travail, offrant une flexibilité pour les développeurs et les ingénieurs de sécurité qui analysent plusieurs clusters ou configurations sans avoir besoin d'accéder directement à chaque cluster.

```
● ● ●

# Linux en utilisant Curl & Bash
curl -s
https://raw.githubusercontent.com/armosec/kubescape/master/in
stall.sh | /bin/bash

# Macos en utilisant Homebrew
brew install kubescape

# Windows en utilisant Chocolatay
choco install kubescape
```

# Installation de Kubescape - Cluster



Exécuter Kubescape directement sur un cluster offre une vision précise de l'état de sécurité actuel du cluster, y compris des configurations en direct et des objets Kubernetes, qui pourraient ne pas être disponibles lors d'une analyse locale.

Cela permet d'automatiser les analyses de sécurité dans le cadre des opérations du cluster, par exemple, via des jobs programmés ou des triggers basés sur des événements.



```
# Vérifiez que votre cluster est bien configuré
helm repo update ; helm upgrade --install kubescape
kubescape/kubescape-operator -n kubescape --create-namespace
--set clusterName=`kubectl config current-context` --set
capabilities.continuousScan=enable
```

# Scan d'un cluster



Pour lancer votre première analyse, vous pouvez exécuter Kubescape contre votre cluster Kubernetes actif ou contre des fichiers de configuration spécifiques.

```
# Analyse le cluster Kubernetes actif selon les recommandations de sécurité de la NSA  
kubescape scan framework nsa  
  
# Analyse le cluster Kubernetes avec beaucoup plus de détails  
kubescape scan framework nsa -v  
  
# Analyse des fichiers de configuration Kubernetes spécifiques  
kubescape scan framework nsa -f ./path/to/your/yaml/files/
```

Framework scanned: NSA

|                 |    |
|-----------------|----|
| Controls        | 25 |
| Passed          | 9  |
| Failed          | 14 |
| Action Required | 2  |

Failed resources by severity:

|          |    |
|----------|----|
| Critical | 0  |
| High     | 18 |
| Medium   | 47 |
| Low      | 6  |

# Interpréter les résultats



Après avoir exécuté Kubescape pour analyser la sécurité de votre cluster Kubernetes ou de vos fichiers de configuration, vous obtiendrez un rapport détaillé. Comprendre et interpréter ce rapport est crucial pour améliorer la sécurité de votre environnement.

- Résumé de la Conformité :** Cette section fournit un aperçu global de la conformité de votre environnement aux standards de sécurité analysés. Elle peut inclure un score de conformité général et des statistiques sur les contrôles passés/échoués.
- Détails des Risques Identifiés :** Ici, Kubescape liste tous les risques détectés lors de l'analyse, classés par严重性 (critique, élevée, moyenne, faible). Pour chaque risque, vous trouverez une description du problème, l'impact potentiel, et des recommandations pour la mitigation ou la correction.

| Severity         | Control name                                       | Failed resources | All Resources | Compliance score  |
|------------------|--|------------------|---------------|-------------------|
| Critical         | Disable anonymous access to Kubelet service        | 0                | 0             | Action Required * |
| Critical         | Enforce Kubelet client TLS authentication          | 0                | 0             | Action Required * |
| High             | Ensure CPU limits are set                          | 9                | 29            | 69%               |
| High             | Ensure memory limits are set                       | 9                | 29            | 69%               |
| Medium           | Prevent containers from allowing command execution | 2                | 89            | 98%               |
| Medium           | Non-root containers                                | 11               | 29            | 62%               |
| Medium           | Allow privilege escalation                         | 3                | 29            | 90%               |
| Medium           | Ingress and Egress blocked                         | 10               | 36            | 72%               |
| Medium           | Automatic mapping of service account               | 13               | 87            | 85%               |
| Medium           | Administrative Roles                               | 2                | 89            | 98%               |
| Medium           | Cluster internal networking                        | 1                | 7             | 86%               |
| Medium           | Linux hardening                                    | 3                | 29            | 90%               |
| Medium           | Secret/etcd encryption enabled                     | 1                | 1             | 0%                |
| Medium           | Audit logs enabled                                 | 1                | 1             | 0%                |
| Low              | Immutable container filesystem                     | 5                | 29            | 83%               |
| Low              | PSP enabled  | 1                | 1             | 0%                |
| Resource Summary |  | 26               | 229           | 72.02%            |

# Interpréter les résultats



Commencez par traiter les risques classés comme critiques ou élevés, car ils représentent les menaces les plus sérieuses pour la sécurité de votre environnement. Les vulnérabilités de sévérité moyenne et faible peuvent être abordées ensuite, en fonction de vos ressources et de votre calendrier.

**Planification de la Remédiation :** Pour chaque risque identifié, évaluez les recommandations fournies et planifiez une stratégie de remédiation. Cela peut impliquer la modification de configurations, la mise à jour de politiques de sécurité, ou la révision des pratiques de développement.

```
#####
ApiVersion: v1
Kind: ServiceAccount
Name: default
Namespace: argocd

Controls: 1 (Failed: 1, action required: 0)



| Resources            |                                                                                       |
|----------------------|---------------------------------------------------------------------------------------|
| Severity             | : Medium                                                                              |
| Control Name         | : Automatic mapping of service account                                                |
| Docs                 | : <a href="https://hub.armosec.io/docs/c-0034">https://hub.armosec.io/docs/c-0034</a> |
| Assisted Remediation | : automountServiceAccountToken=false                                                  |


```

# Frameworks



Kubescape supporte plusieurs frameworks de conformité pour évaluer la sécurité de vos environnements Kubernetes. Ces frameworks fournissent des ensembles de règles et de meilleures pratiques pour sécuriser vos clusters. En voici quelques uns :

1. **NSA (National Security Agency) et CISA (Cybersecurity and Infrastructure Security Agency)** : Ce framework fournit des recommandations pour sécuriser les environnements Kubernetes contre les menaces courantes. Il est conçu pour être applicable à une large gamme d'organisations.
2. **CIS Kubernetes Benchmark** : Développé par le Centre for Internet Security, ce benchmark offre des conseils détaillés sur la sécurisation des clusters Kubernetes, en se basant sur des pratiques reconnues de l'industrie.
3. **MITRE ATT&CK®** : Ce framework se concentre sur les tactiques et techniques utilisées par les adversaires pour attaquer les environnements cloud et Kubernetes.

Chaque framework a ses propres forces et se concentre sur différents aspects de la sécurité Kubernetes. Le choix du framework dépend de vos objectifs de sécurité spécifiques, des exigences réglementaires, et du niveau de sécurité souhaité.

```

● ● ●

# Connaître les frameworks disponibles
kubescape list frameworks

```

# Accepting risk



La gestion des exceptions de posture dans Kubescape permet aux équipes de définir des politiques d'exception pour des contrôles de sécurité spécifiques, offrant ainsi une flexibilité dans la gestion des risques de sécurité identifiés. Cette approche est particulièrement utile pour les risques que vous avez choisi d'accepter en raison de contraintes opérationnelles, de nécessités fonctionnelles ou lorsque des mesures compensatoires sont en place.

```
[  
  {  
    "name": "accept-non-root-containers",  
    "policyType": "postureExceptionPolicy",  
    "actions": ["alertOnly"],  
    "resources": [{"designatorType": "Attributes",  
"attributes": {"kind": ".*"}},  
     "posturePolicies": [{"controlID": "C-0013"}]  
  },  
  {  
    "name": "accept-allow-privilege-escalation",  
    "policyType": "postureExceptionPolicy",  
    "actions": ["alertOnly"],  
    "resources": [{"designatorType": "Attributes",  
"attributes": {"kind": ".*"}},  
     "posturePolicies": [{"controlID": "C-0016"}]  
  }  
]
```

# Accepting risk



L'utilisation se fait ensuite de cette manière.

```

● ● ●

kubescape scan --exceptions exceptions.json
  
```

L'utilisation du Dashboard ARMO pour la gestion des risques passe directement par l'onglet "Vulnerabilities".

Il est ensuite possible de mettre en place des règles en fonction de celles-ci.

|               |          |            |      |       |    |     |                            |      |   |                    |
|---------------|----------|------------|------|-------|----|-----|----------------------------|------|---|--------------------|
| CVE-2011-1939 | Critical | <b>9.8</b> | v3.1 | 1.33% | No | Yes | php-checks<br>3.33.0.11274 | Link | 1 | ⋮                  |
| CVE-2011-1939 | Critical | <b>9.8</b> | v3.1 | 1.33% | No | Yes | php-frontend<br>[REDACTED] | Link | 1 | <b>Accept Risk</b> |
| CVE-2014-9912 | Critical | <b>9.8</b> | v3.0 | 1.09% | No | Yes | php-checks<br>3.33.0.11274 | Link | 1 | ⋮                  |

# Scan des Images



Le scan d'image par Kubescape permet d'identifier les vulnérabilités connues dans les images de conteneurs, en se basant sur des bases de données de vulnérabilités à jour. Cette fonctionnalité aide à détecter et à corriger les vulnérabilités avant que les images ne soient déployées dans un environnement de production.

```
● ● ●  
# Scan une image de conteneur locale pour les vulnérabilités  
kubescape scan image yourimage:tag  
  
# Scan une image de conteneur dans un registre distant  
kubescape scan image registrypath/yourimage:tag
```

# Scan d'un déploiement



Kubescape peut être utilisé sur l'analyse de sécurité des déploiements Kubernetes eux-mêmes. Cela en mettant l'accent sur la détection des configurations à risque et des vulnérabilités potentielles

```
● ● ●

# Liste tous les déploiements dans le namespace par défaut
kubectl get deployments --namespace=default

# Exporte la configuration du déploiement en fichier YAML
kubectl get deployment <nom-du-déploiement> --namespace=
<namespace> -o yaml > deployment.yaml

# Analyse la configuration du déploiement avec Kubescape
kubescape scan framework nsa deployment.yaml
```

Framework scanned: NSA

|                 |    |
|-----------------|----|
| Controls        | 19 |
| Passed          | 15 |
| Failed          | 4  |
| Action Required | 0  |

Failed resources by severity:

|          |   |
|----------|---|
| Critical | 0 |
| High     | 2 |
| Medium   | 2 |
| Low      | 0 |

Run with '--verbose'/'-v' to see control failures for each resource.

| Severity         | Control name                 | Failed resources | All Resources | Compliance score |
|------------------|------------------------------|------------------|---------------|------------------|
| High             | Ensure CPU limits are set    | 1                | 1             | 0%               |
| High             | Ensure memory limits are set | 1                | 1             | 0%               |
| Medium           | Non-root containers          | 1                | 1             | 0%               |
| Medium           | Ingress and Egress blocked   | 1                | 1             | 0%               |
| Resource Summary |                              | 1                | 1             | 78.95%           |

# Scan d'un Helm



L'analyse directe des charts Helm avec Kubescape est une fonctionnalité précieuse qui permet d'évaluer la sécurité des packages Helm avant leur déploiement dans un cluster Kubernetes. Cette approche facilite l'identification et la correction des vulnérabilités ou des configurations à risque au sein des charts Helm eux-mêmes, renforçant ainsi la posture de sécurité dès le début du cycle de déploiement.

```
● ● ●  
# Analyse du dossier courant où se trouve le Chart Helm  
kubescape scan ./
```

# Formats de Résultats



Ces commandes offrent une grande flexibilité pour l'analyse et la présentation des résultats, vous permettant de choisir le format qui convient le mieux à vos besoins, que ce soit pour une revue directe, une intégration dans des pipelines d'intégration continue/déploiement continu (CI/CD), ou pour la génération de rapports pour les parties prenantes.

```
● ● ●

# Format JSON - Génère les résultats au format JSON
kubescape scan framework nsa -f <fichier ou dossier> --format json

# Format JUnit - Produit les résultats au format JUnit XML,
# utile pour les intégrations CI/CD
kubescape scan framework nsa -f <fichier ou dossier> --format junit

# Format Prometheus - Expose les résultats sous forme de
# métriques Prometheus
kubescape scan framework nsa -f <fichier ou dossier> --format prometheus

# Format PDF - Crée un rapport PDF des résultats
kubescape scan framework nsa -f <fichier ou dossier> --format pdf

# Utilisez --output filename.extension pour enregistrer le
# résultat dans un fichier
```

# Surveillance d'un cluster



La surveillance continue de la sécurité d'un cluster Kubernetes est essentielle pour détecter et atténuer rapidement les vulnérabilités et les configurations à risque qui pourraient apparaître au fil du temps. La surveillance continue avec Kubescape peut être configurée pour s'exécuter à intervalles réguliers, par exemple en tant que jobs programmés dans un cluster Kubernetes et en envoyant les données à un service Prometheus.

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: kubescape-scan
spec:
  schedule: "0 0 * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - name: kubescape
              image: <votre-image-contenant-kubescape>
              command: ["/bin/sh"]
              args:
                - "-c"
                - "kubescape scan framework nsa --format prometheus > /data/kubescape-results.prom"
            volumeMounts:
              - name: data-volume
                mountPath: /data
          restartPolicy: OnFailure
          volumes:
            - name: data-volume
              persistentVolumeClaim:
                claimName: kubescape-results-pvc
```

# Surveillance d'un cluster



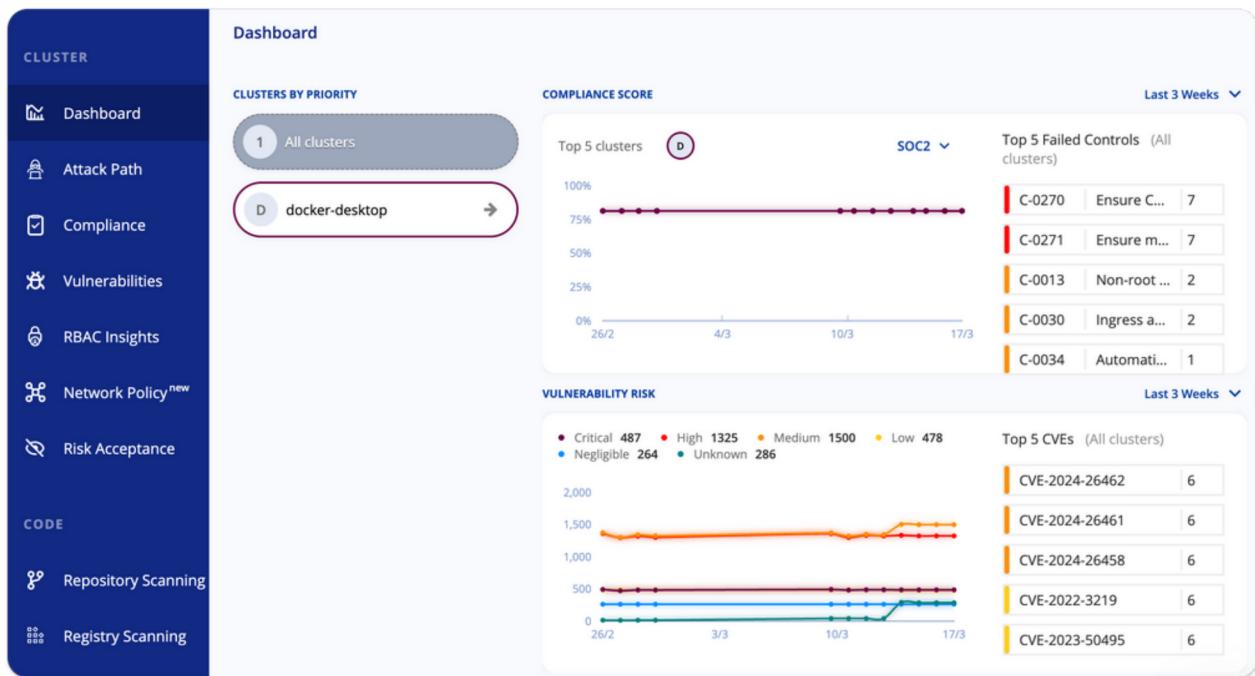
La surveillance continue de la sécurité d'un cluster Kubernetes est essentielle pour détecter et atténuer rapidement les vulnérabilités et les configurations à risque qui pourraient apparaître au fil du temps. La surveillance continue avec Kubescape peut être configurée pour s'exécuter à intervalles réguliers, par exemple en tant que jobs programmés dans un cluster Kubernetes et en envoyant les données à un service Prometheus.

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: kubescape-scan
spec:
  schedule: "0 0 * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - name: kubescape
              image: <votre-image-contenant-kubescape>
              command: ["/bin/sh"]
              args:
                - "-c"
                - "kubescape scan framework nsa --format prometheus > /data/kubescape-results.prom"
            volumeMounts:
              - name: data-volume
                mountPath: /data
          restartPolicy: OnFailure
          volumes:
            - name: data-volume
              persistentVolumeClaim:
                claimName: kubescape-results-pvc
```

# Dashboard ARMO



Le Dashboard ARMO, développé par les créateurs de Kubescape, est un outil puissant pour la visualisation, la surveillance et l'analyse approfondie des résultats des analyses de sécurité effectuées par Kubescape sur vos clusters Kubernetes.



Le Dashboard ARMO amplifie les capacités de Kubescape en offrant une plateforme riche pour la visualisation, la surveillance et l'analyse des risques de sécurité dans vos environnements Kubernetes. En intégrant Kubescape avec le Dashboard ARMO, les équipes de sécurité et de DevOps peuvent travailler de manière plus proactive pour identifier, comprendre et corriger les vulnérabilités, renforçant ainsi la sécurité de leurs déploiements.

# Attack Path



L'Attack Path du Dashboard ARMO aide à visualiser les faiblesses exploitables par des acteurs malveillants dans les clusters Kubernetes. En identifiant ces chemins, ARMO met en lumière les étapes spécifiques où les attaques peuvent être bloquées et guide les ingénieurs dans les étapes de remédiation. Cela facilite la compréhension des risques et la mise en œuvre de contre-mesures efficaces pour renforcer la sécurité des déploiements Kubernetes.

Attack Path
Scan all

⌚ Last scan: Mar 19, 2024

+ Add filter

First seen: Feb 23, 2024 [REDACTED]

**Exposed argocd-server with critical vulnerabilities and 3 severe impacts**

Cluster: docker-desktop  
Namespace: argocd  
Workload: argocd-server



+ Add filter

First seen: Feb 23, 2024 [REDACTED]

**Potential denial of service via argocd-server with no limits**

Cluster: docker-desktop  
Namespace: argocd  
Workload: argocd-server



19

# Repository Scanning



Le scanning de répertoires de code et de registres d'images de conteneurs par ARMO permet d'identifier et de remédier aux vulnérabilités avant le déploiement. Cette approche intègre la sécurité dès les premières étapes du développement, analysant à la fois le code source et les images de conteneurs pour les failles de sécurité. En se concentrant sur ces deux éléments cruciaux, ARMO aide à créer une ligne de défense solide contre les attaques potentielles, assurant ainsi une sécurité robuste dès le début du cycle de vie de l'application.

| Repositories Scan / charts                    |   |                 |            |                       |                      |                        |
|---|---|-----------------|------------|-----------------------|----------------------|------------------------|
| Repositories Scan                             |   | REPOSITORY NAME | FILES      | TIME                  |                      |                        |
|   |   | charts          | 810        | Aug 10, 2022 05:51:46 |                      |                        |
| All Severities                                | v   |                 |            |                       | Search by file names | 🔍                      |
| FOLDER  | FILE NAME                                     | FILE TYPE       | FRAMEWORKS | CONTROLS              | FAILED CONTROLS      | COMMIT BY              |
| stable/prometheus-blackbox-exporter/templates | deployment.yaml                               | Helm Chart      | All        | 20 passed: 10 failed: | ✖ 0 • 2 • 6 • 2      | LucasBoisserie ⌂       |
| stable/prometheus-blackbox-exporter/templates | configmap.yaml                                | Helm Chart      | All        | 1 passed:             | ✖ 0 ✖ 0 ✖ 0 ✖ 0      | Cédric de Saint Martin |
| stable/prometheus-adapter/templates           | custom-metrics-configmap.yaml                 | Helm Chart      | All        | 1 passed:             | ✖ 0 ✖ 0 ✖ 0 ✖ 0      | Dudko Alexey           |
| stable/prometheus-adapter/templates           | custom-metrics-apiserver-service-account.yaml | Helm Chart      | All        | 2 passed:             | ✖ 0 ✖ 0 ✖ 0 ✖ 0      | Mattias Gees           |
| stable/prometheus-adapter/templates           | custom-metrics-apiserver-deployment.yaml      | Helm Chart      | All        | 23 passed: 6 failed:  | ✖ 0 • 2 • 2 • 2      | Eric Herot ⌂           |
| stable/prisma/templates                       | serviceaccount.yaml                           | Helm Chart      | All        | 2 passed:             | ✖ 0 ✖ 0 ✖ 0 ✖ 0      | Giacomo Guiulfo        |
| stable/prisma/templates                       | deployment.yaml                               | Helm Chart      | All        | 17 passed: 13 failed: | ✖ 0 • 2 • 7 • 4      | Peter Belko ⌂          |
| stable/prisma/templates                       | configmap.yaml                                | Helm Chart      | All        | 1 passed:             | ✖ 0 ✖ 0 ✖ 0 ✖ 0      | Peter Belko            |

# Dans la même collection

## Orchestration et Gestion de Conteneurs



## Infrastructure as Code



## Sécurité & Gestion des secrets



## Développement & CI/CD



↓ FOLLOW ME ↓



[ANTONYCANUT](#)



[ANTONY KERVAZO-CANUT](#)