



# Configuration du Système d'Alerting



## Vue d'Ensemble

Vous avez maintenant 4 Watchers configurés :

1. **watcher-mark-resolved.json** - Marque automatiquement les vulnérabilités comme resolved (calcule MTTR)
2. **watcher-code-duplication.json** - Alerte sur code dupliqué (Email + Slack)
3. **watcher-critical-cve.json** - Alerte sur CVE critique (Email + Slack)
4. **watcher-exploitable-vulnerability.json** - Alerte sur vulnérabilités exploitables (Email + Slack)



## Configuration Email dans Elasticsearch

### Étape 1 : Configurer le serveur SMTP

Éditez le fichier `elasticsearch.yml` :



yaml

```
# Configuration Email (SMTP)
```

```
xpack.notification.email.account:
```

```
work:
```

```
  profile: standard
```

```
  smtp:
```

```
    auth: true
```

```
    starttls.enable: true
```

```
    host: smtp.gmail.com
```

```
    port: 587
```

```
    user: votre-email@example.com
```

```
    password: votre-mot-de-passe-app
```

### Étape 2 : Redémarrer Elasticsearch



bash

```
docker-compose restart elasticsearch
```

### Étape 3 : Tester la Configuration Email

Dans Kibana → Dev Tools :



json

```
POST _watcher/watch/_execute
{
  "watch": {
    "trigger": {"schedule": {"interval": "1m"}},
    "input": {"simple": {}},
    "actions": {
      "send_email": {
        "email": {
          "to": "votre-email@example.com",
          "subject": "Test Email Watcher",
          "body": {
            "text": "Si vous recevez cet email, la configuration fonctionne !"
          }
        }
      }
    }
  }
}
```

## Configuration Slack

### Étape 1 : Créer un Webhook Slack

1. Aller sur <https://api.slack.com/apps>
2. Cliquer sur "Create New App"
3. Choisir "From scratch"
4. Nom : Security Alerts
5. Workspace : Sélectionner votre workspace

### Étape 2 : Activer Incoming Webhooks

1. Dans votre app → Incoming Webhooks
2. Activer : ON
3. Cliquer sur "Add New Webhook to Workspace"
4. Choisir le canal : #security-alerts (ou créer un nouveau canal)
5. Autoriser

### Étape 3 : Copier l'URL du Webhook

Vous obtiendrez une URL comme :



https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXXXXXX

## Étape 4 : Mettre à Jour les Watchers

Dans chaque watcher (watcher-code-duplication.json, watcher-critical-cve.json, etc.), remplacez :



json

"path": "/services/YOUR\_SLACK\_WEBHOOK\_PATH"

Par votre vraie URL (seulement la partie après https://hooks.slack.com) :



json


"path": "/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXXXXXX"

## Étape 5 : Tester Slack

Testez avec curl :



bash

```
curl -X POST https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXXXXXX \
-H 'Content-Type: application/json' \
-d '{
  "text": "Test de notification Slack depuis Elasticsearch Watcher  "
}'
```

Si ça fonctionne, vous verrez le message dans votre canal Slack.

## Installation des Watchers dans Kibana

### Méthode 1 : Via l'Interface Kibana (RECOMMANDÉ)

1. Aller dans Kibana → Stack Management → Watcher
2. Cliquer sur "Create" → "Create advanced watch"
3. Copier-coller le contenu JSON d'un watcher
4. Watch ID : Donner un nom (ex: code-duplication-alert)
5. Cliquer sur "Create watch"
6. Répéter pour les 4 watchers

## Méthode 2 : Via API (Alternative)



bash

```
# Watcher 1 : Mark Resolved
curl -X PUT "http://localhost:9200/_watcher/watch/mark-resolved" \
-u elastic:changeme \
-H 'Content-Type: application/json' \
-d @watcher-mark-resolved.json

# Watcher 2 : Code Duplication
curl -X PUT "http://localhost:9200/_watcher/watch/code-duplication-alert" \
-u elastic:changeme \
-H 'Content-Type: application/json' \
-d @watcher-code-duplication.json

# Watcher 3 : Critical CVE
curl -X PUT "http://localhost:9200/_watcher/watch/critical-cve-alert" \
-u elastic:changeme \
-H 'Content-Type: application/json' \
-d @watcher-critical-cve.json

# Watcher 4 : Exploitable Vulnerability
curl -X PUT "http://localhost:9200/_watcher/watch/exploitable-vuln-alert" \
-u elastic:changeme \
-H 'Content-Type: application/json' \
-d @watcher-exploitable-vulnerability.json
```

---

## Vérification des Watchers

**Voir tous les watchers actifs**



bash

```
curl -u elastic:changeme "http://localhost:9200/_watcher/watch/_query" | jq .'
```

**Voir l'état d'un watcher spécifique**



bash

```
curl -u elastic:changeme "http://localhost:9200/_watcher/watch/code-duplication-alert" | jq .'
```

## Voir l'historique d'exécution

Dans Kibana → Stack Management → Watcher → Cliquer sur un watcher → Onglet "Execution history"

## Tester les Watchers

### Test 1 : Exécution Manuelle

Dans Kibana → Stack Management → Watcher → Sélectionner un watcher → "Execute"

### Test 2 : Déclencher une Vraie Alerte

- 1. Lancer un build Jenkins qui contient du code dupliqué ou une CVE
- 2. Attendre 5-10 minutes (intervalle du watcher)
- 3. Vérifier votre email et canal Slack

## Personnalisation des Alertes

### Modifier les Destinataires Email

Dans le JSON du watcher, section `actions.send_email.email.to` :



```
json

"to": [
  "{ctx.payload.hits.hits.0._source.git.commit.author.email}",
  "security-team@example.com",
  "lead-dev@example.com"
]
```

### Modifier le Canal Slack

Créez un nouveau webhook pour un autre canal et remplacez le path dans le watcher.

### Modifier les Conditions de Déclenchement

Exemple : Alerter seulement si `severity >= HIGH` et `service = backend` :



```
json
```

```
"condition": {  
  "script": {  
    "source": "return ctx.payload.hits.total.value > 0 && ctx.payload.hits.hits[0]._source.service.name == 'backend' &&  
  }  
}
```



## Modifier la Fréquence

Dans `trigger.schedule.interval` :







json

```
"interval": "5m" // Toutes les 5 minutes  
"interval": "1h" // Toutes les heures  
"interval": "30s" // Toutes les 30 secondes
```

## Templates HTML Email Personnalisés

Les emails sont déjà en HTML avec :

-  Couleurs selon la severity
-  Tableaux formatés
-  Boutons d'action
-  Logo/branding (vous pouvez ajouter)

Pour ajouter votre logo :



html

```
<div style='text-align:center;margin-bottom:20px;'>  
  <img src='https://votre-domaine.com/logo.png' alt='Logo' style='width:150px;' />  
</div>
```

## Monitoring des Alertes

### Dashboard Kibana pour les Alertes

Créez un dashboard avec :

- Nombre d'alertes envoyées par jour
- Types d'alertes (code\_smell, CVE, exploit)

- Temps de résolution moyen (MTTR)
- Top développeurs avec le plus d'alertes

## Métriques Watchers

Dans Dev Tools :



json

```
GET .watcher-history-*/_search
{
  "size": 0,
  "query": {
    "range": {
      "@timestamp": {"gte": "now-7d"}
    }
  },
  "aggs": {
    "alerts_per_day": {
      "date_histogram": {
        "field": "@timestamp",
        "calendar_interval": "day"
      },
      "aggs": {
        "by_watch": {
          "terms": {
            "field": "watch_id.keyword"
          }
        }
      }
    }
  }
}
```

## Sécurité

### Masquer les Credentials

**IMPORTANT :** Ne committez JAMAIS les watchers avec des vraies credentials dans Git !

**Solution :** Utilisez des variables d'environnement ou Elasticsearch Keystore :



bash

*# Ajouter le webhook Slack au keystore*

elasticsearch-keystore [add](#) xpack.notification.slack.account.monitoring.secure\_url

*# Dans le watcher, référencer :*

"{{#toJson}}ctx.metadata.slack\_webhook{{/toJson}}"

---

## Ressources

- [Documentation Elasticsearch Watcher](#)
  - [Slack Incoming Webhooks](#)
  - [Email Configuration Elasticsearch](#)
- 

## Checklist de Configuration

- ☐ Elasticsearch configuré avec SMTP
- ☐ Email de test envoyé avec succès
- ☐ Webhook Slack créé
- ☐ Notification Slack de test reçue
- ☐ 4 Watchers créés dans Kibana
- ☐ Watchers mis à jour avec vraies URLs Slack
- ☐ Test d'exécution manuelle réussi
- ☐ Test avec vraie alerte (build Jenkins)
- ☐ Email reçu par le développeur
- ☐ Notification Slack reçue dans le canal
- ☐ Dashboard monitoring des alertes créé