

SecurityContext, NetworkPolicy



Сергей
Андрюнин



Сергей Андрюнин

DevOps-инженер

RTLabs




Сергей Андрюнин



План занятия

1. [Общие сведения о SecurityContext](#)
2. [Общие сведения о NetworkPolicy](#)
3. [Итоги](#)
4. [Домашнее задание](#)



Общие сведения о SecurityContext

Общие сведения о SecurityContext

- предназначены для настройки политики безопасности;
- любое изменение требует полного понимания последствий;
- все изменения опасны и могут привести к взлому и получению полного контроля над узлом кластера злоумышленниками.

Список базовых политик

- привилегированный статус контейнера;
- гранулированные права;
- SELinux;
- маскировка файловой системы /proc;
- параметры ядра.

Привилегированный статус контейнера

Поля:

- `spec.containers[*].securityContext.privileged`
- `spec.initContainers[*].securityContext.privileged`

Описание:

- используется как простой способ повышения привилегий контейнера;
- очень опасен, использовать не рекомендуется.

Гранулированные права

Поля:

- `spec.containers[*].securityContext.capabilities`
- `spec.initContainers[*].securityContext.capabilities`

Описание:

- управляет некоторыми разрешениями на вызовы ядра capabilities;
- очень опасен, использовать не рекомендуется;
- более подробное описание [по ссылке](#).

SELinux

Поля:

- `spec.securityContext.seLinuxOptions.type`
- `spec.containers[*].securityContext.seLinuxOptions.type`
- `spec.initContainers[*].securityContext.seLinuxOptions.type`
- `spec.securityContext.seLinuxOptions.user`
- `spec.containers[*].securityContext.seLinuxOptions.user`
- `spec.initContainers[*].securityContext.seLinuxOptions.user`
- `spec.securityContext.seLinuxOptions.role`
- `spec.containers[*].securityContext.seLinuxOptions.role`
- `spec.initContainers[*].securityContext.seLinuxOptions.role`

SELinux

Описание:

- управляет политикой SELinux;
- очень опасен, использовать не рекомендуется.

Маскировка файловой системы /proc

Поля:

- `spec.containers[*].securityContext.procMount`
- `spec.initContainers[*].securityContext.procMount`

Описание:

- снимает маскировку некоторых веток для точки монтирования /proc;
- очень опасен, использовать не рекомендуется.

Параметры ядра

Поля:

- `spec.securityContext.sysctls`

Описание:

- отключает механизм безопасности для всех контейнеров узла;
- позволяет изменить некоторые параметры ядра;
- может вызвать нестабильную работу операционной системы узла;
- опасен, использовать не рекомендуется.

Список служебных политик

- привилегированный статус контейнера;
- запуск от имени другого пользователя (не root).

Привилегированный статус контейнера

Поля:

- `spec.containers[*].securityContext.allowPrivilegeEscalation`
- `spec.initContainers[*].securityContext.allowPrivilegeEscalation`

Описание:

- используется как простой способ повышения привилегий контейнера;
- очень опасен, использовать не рекомендуется.

Запуск от имени другого пользователя

Поля:

- `spec.securityContext.runAsNonRoot`
- `spec.containers[*].securityContext.runAsNonRoot`
- `spec.initContainers[*].securityContext.runAsNonRoot`
- `spec.securityContext.runAsUser`
- `spec.containers[*].securityContext.runAsUser`
- `spec.initContainers[*].securityContext.runAsUser`
- `spec.securityContext.runAsUser`
- `spec.containers[*].securityContext.runAs`
- `spec.initContainers[*].securityContext.runAsGroup`

Запуск от имени другого пользователя

Описание:

- используется для явной установки пользователя UID и группы GID;
- возможно, в образе уже имеется пользователь с необходимыми правами, но его использование остается на усмотрение разработчиков и администраторов;
- по мере необходимости.

Запуск от имени другого пользователя

Поля:

- `spec.securityContext.supplementalGroups[*]`.

Описание:


- указание списка дополнительных групп GID.

Поля:

- `spec.securityContext.fsGroup`.

Описание:

- указание группы GID для монтируемых файлов в томах;
- для больших и медленных файловых систем может привести к существенному замедлению при запуске.



Общие сведения о NetworkPolicy

Общие сведения о NetworkPolicy

- позволяет настроить политику сетевого доступа к ресурсам;
- имеет два типа: входящий ingress и исходящий egress;
- определяется набором правил в селекторах.

Пример определения

```
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: default-deny-ingress
spec:
  podSelector: {}
  policyTypes:
    - Ingress
```

Селекторы

Участвуют как в выборке групп модулей, к которым будет применена политика, так и в выборке групп модулей в качестве поведенческого фильтра для источника входящего трафика или получателя исходящего трафика:

- podSelector – выбирает модули;
- namespaceSelector – выбирает пространства.

Отдельно имеется селектор ipBlock для указания IP адресов.

Политики по умолчанию

- запретить весь входящий трафик (default-deny-ingress);
- разрешить весь входящий трафик (allow-all-ingress);
- запретить весь исходящий трафик (default-deny-egress);
- разрешить весь исходящий трафик (allow-all-egress);
- запретить весь входящий и исходящий трафик (default-deny-all).

Пример ingress

```
ingress:
- from:
- ipBlock:
    cidr: 172.17.0.0/16
    except:
      - 172.17.1.0/24
- namespaceSelector:
    matchLabels:
      project: myproject
- podSelector:
    matchLabels:
      role: frontend
ports:
- protocol: TCP
  port: 6379
```

Пример egress

```
egress:
- to:
- ipBlock:
    cidr: 10.0.0.0/24
ports:
- protocol: TCP
    port: 5978
```

Итоги

Сегодня мы изучили:

- что такое SecurityContext и NetworkPolicy;
- как их создавать и использовать в kubernetes.

Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Сергей Андрюнин