

Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет «Высшая школа экономики»

Московский институт электроники и математики им. А.Н. Тихонова

Компьютерная безопасность

(Название ОП)

Специалитет

(уровень образования)

О Т Ч Е Т
по проектной работе

Создание кластерной инфраструктуры для разработки и развертывания программных
проектов

(Название проекта)

Выполнили студенты гр. СКБ181

Петров Артем Эдуардович

Рымкулова Диана

(ФИО)

(подпись)

Руководитель проекта:

Минченков Виктор Олегович

(должность, ФИО руководителя проекта)

(оценка)

(подпись)

(дата)

г. Москва 2022

СОДЕРЖАНИЕ

Техническое задание	3
Актуальность проекта	3
Цель проекта	3
Цели проекта на текущий год	3
Задачи	4
Планируемый результат	4
Пользовательский опыт	4
Обеспечение безопасности	4
Требуемые и приобретаемые навыки	4
Требуемые навыки:	4
Приобретаемые навыки:	5
Форма и способы промежуточного контроля	5
Форма представления результатов	5
Реализация и внедрение результатов проекта (опыт или планы)	6
Методика и результаты испытаний	7
Тестирование политик Gatekeeper	7
Тестирование RBAC (Role-based access control)	8
Тестирование API сервиса	9
Тестирование веб-сайтов	9
Информация о составе проектной команды, контакты.	9
Подробное описание разработанной системы	10
API сервис	10
Веб-сайт для взаимодействия с API сервисом	11
Веб-сайт с пользовательской документацией	12
Экономическая эффективность	13
Перечень основных технических и научных результатов	14
Примеры работы программного обеспечения	14
“Мои проекты”	14
“Доступ”	19
“Управление”	20
“Документация”	20
Ход работ	22
Роли участников команды	23
Новизна/преимущества решений, полученных по результатам проекта	24

Техническое задание

Актуальность проекта

Kubernetes - это открытое программное обеспечение для автоматизации развёртывания, масштабирования и управления контейнеризированными приложениями.

Kubernetes за небольшой срок существования зарекомендовал себя как стандарт де-факто для использования в облачных системах и сервисах. В частности, является основой контейнеризационных сервисов ряда крупнейших публично-облачных провайдеров — Amazon, Microsoft, Google и помимо этого множества отечественных компаний, включая и из быстроразвивающейся финтех среды.

Тем самым благодаря созданию и интегрированию кластера Kubernetes в архитектуру систем поддержания проектной деятельности, мы можем гарантировать погружение учащихся в актуальную систему и получение важных практических навыков, схожих по своей сути с непосредственной коммерческой разработкой.

Цель проекта

Разработка, создание и интегрирование кластера Kubernetes в архитектуру систем поддержания проектной деятельности.

Цели проекта на текущий год

- Реализовать политику безопасности
- Составить методические материалы для пользователей
- Интеграция возможности быстрого развертывания и публикации проектов с WEB интерфейсами
- Реализовать грейдинг проектов в панели управления

Задачи

- Разработать и интегрировать кластер Kubernetes в архитектуру систем поддержания проектной деятельности
- Настроить автоматическое выделение ресурсов и поддержание работоспособности приложений
- Автоматизировать и стандартизировать процесс разработки и публикации сервисов
- Улучшить User Experience
- Обеспечить безопасность рабочей нагрузки
- Составить документацию по работе с необходимыми ресурсами

Планируемый результат

Пользовательский опыт

Одной из целей в данном году является упрощение процесса взаимодействия пользователя с кластером. Разработанный интерфейс станет удобным дополнением утилиты kubectl. Он позволит упростить административную работу участникам и руководителям проектов.

Обеспечение безопасности

Поскольку в Kubernetes по умолчанию нет требуемого уровня защиты, будет обеспечена дополнительная безопасность. Так наш кластер будет защищен как от неосторожных действий пользователей, так и от намеренных атак и угроз.

Требуемые и приобретаемые навыки

Требуемые навыки:

- Навыки программирования (python, go, nodejs)
- Базовые навыки веб разработки

- Базовые навыки по администрировании сетей
- Расширенные навыки администрированию DEVOPS сервисов
- Владение английский язык на уровне, необходимом для чтения технической литературы

Приобретаемые навыки:

- Улучшение навыков программирования
- Навык по составлению и публикации документации
- Навык по верстке сайтов
- Навык проведения тестирований конечного продукта
- Навык по проектированию и публикации сервисов

Форма и способы промежуточного контроля

1. Постерная сессия в МИЭМ
2. Презентация проекта

Форма представления результатов

1. Отчет о выполнении проекта с описанием проделанной работы и полученных результатов.
2. API сервис для авторизации и получения основной информации о кластере
3. Веб-сайт с пользовательской документацией

Реализация и внедрение результатов проекта (опыт или планы)

На данном этапе результатом проекта является продукт, предназначенный для использования проектным офисом, а именно - развертыванию проектов в кластере Kubernetes. Продукт создавался для студентов в виде помощи по изучению современных средств разработки и с учетом возможного сокращения использования внешних ресурсов, по типу виртуальных машин.

В дальнейшем проект возможно доработать в сторону расширения функционала, например, пополнение базы проверок на уязвимостей и ошибок в конфигурации. Помимо этого, возможно расширить функционал взаимодействия с кластером через веб-интерфейс, а также дорабатывать и дополнять документацию по мере изменения версий и стандартов.

Методика и результаты испытаний

Тестирование политик Gatekeeper

Для проверки работоспособности введенных ограничений была совершена попытка публикации манифеста приложения в Kubernetes.

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: nginx
5 spec:
6   containers:
7     - name: nginx
8       image: nginx:1.14.2
9       ports:
10      - containerPort: 80
```

Данное приложение не было опубликовано, была выведена ошибка, которая содержит список непройденных ограничений с поясняющим сообщением, что подтверждает корректное поведение.

```
crio@crio-virtual-machine:~/Desktop/423/gatekeeper-library$ kubectl apply -f pod.yml
Error from server ([psp-volume-types] The volume type projected is not allowed, pod: nginx. Allowed volume types: ["configMap", "emptyDir", "secret", "persistentVolumeClaim"]
[])
[psp-allow-privilege-escalation-container] Privilege escalation container is not allowed: nginx
[psp-pods-allowed-user-ranges] Container nginx is attempting to run without a required securityContext/runAsUser
[psp-pods-allowed-user-ranges] Container nginx is attempting to run without a required securityContext/runAsGroup. Allowed runAsGroup: {"ranges": [{"max": 200, "min": 100}]}
["rule": "MustRunAs"}
[psp-pods-allowed-user-ranges] Container nginx is attempting to run without a required securityContext/supplementalGroups. Allowed supplementalGroups: {"ranges": [{"max": 2
00, "min": 100}], "rule": "MustRunAs"}
[psp-pods-allowed-user-ranges] Container nginx is attempting to run without a required securityContext/fsGroup. Allowed fsGroup: {"ranges": [{"max": 200, "min": 100}], "rul
e": "MustRunAs"}
[psp-readonlyrootfilesystem] only read-only root filesystem container is allowed: nginx
[repo-is-openpolicyagent] container <nginx> has an invalid image repo <nginx:1.14.2>, allowed repos are ["registry.miem.hse.ru/"]
[psp-autounmount-serviceaccount-token-pod] Autounmounting service account token is disallowed, pod: nginx
[container-must-meet-memory-and-cpu-ratio] container <nginx> has no resource limits
[container-must-meet-memory-and-cpu-ratio] container <nginx> has no resource requests
[container-must-have-limits] container <nginx> has no resource limits: error when creating "pod.yml": admission webhook "validation.gatekeeper.sh" denied the request: [psp
-volume-types] The volume type projected is not allowed, pod: nginx. Allowed volume types: ["configMap", "emptyDir", "secret", "persistentVolumeClaim"]
[psp-allow-privilege-escalation-container] Privilege escalation container is not allowed: nginx
[psp-pods-allowed-user-ranges] Container nginx is attempting to run without a required securityContext/runAsUser
[psp-pods-allowed-user-ranges] Container nginx is attempting to run without a required securityContext/runAsGroup. Allowed runAsGroup: {"ranges": [{"max": 200, "min": 100}]}
["rule": "MustRunAs"}
[psp-pods-allowed-user-ranges] Container nginx is attempting to run without a required securityContext/supplementalGroups. Allowed supplementalGroups: {"ranges": [{"max": 2
00, "min": 100}], "rule": "MustRunAs"}
[psp-pods-allowed-user-ranges] Container nginx is attempting to run without a required securityContext/fsGroup. Allowed fsGroup: {"ranges": [{"max": 200, "min": 100}], "rul
e": "MustRunAs"}
[psp-readonlyrootfilesystem] only read-only root filesystem container is allowed: nginx
[repo-is-openpolicyagent] container <nginx> has an invalid image repo <nginx:1.14.2>, allowed repos are ["registry.miem.hse.ru/"]
[psp-autounmount-serviceaccount-token-pod] Autounmounting service account token is disallowed, pod: nginx
[container-must-meet-memory-and-cpu-ratio] container <nginx> has no resource limits
[container-must-meet-memory-and-cpu-ratio] container <nginx> has no resource requests
[container-must-have-limits] container <nginx> has no resource limits
```

Чтобы исключить случай, когда наши политики блокируют все публикуемые приложения, было решено написать манифест, который удовлетворяет всем политикам безопасности.

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: nginx
5 spec:
6   containers:
7     - name: nginx
8       image: "registry.miem.hse.ru/inf20/k8s-cluster-auth/docs:master.83755"
9       resources:
10         limits:
11           cpu: 200m
12           memory: 256Mi
13         requests:
14           cpu: 200m
15           memory: 256Mi
16       ports:
17         - containerPort: 80
18       securityContext:
19         allowPrivilegeEscalation: false
```

Данный манифест успешно прошел все проверки и был опубликован в кластере.

В ходе тестирования системы было обнаружено, что полный перечень ограничений затрудняет работу начинающих пользователей в кластере, так как некоторые ограничения требуют настройку параметров исполняемой среды, которую трудно выполнить не обладая опытом работы с контейнеризацией. Например, параметры “runAsUser”, “runAsGroup”, “fsGroup”, определяющие пользователя и его группу, под которыми выполняются действия в среде исполнения.

Тестирование RBAC (Role-based access control)

По умолчанию в Kubernetes реализована система управления доступом, основанная на ролях пользователей. Для выполнения принципа минимальных привилегий было принято решение ограничить возможности пользователей в системе. Были определены три основных роли: администратор кластера, администратор проекта и разработчик.

Тестирование заключалось в попытке выполнить всевозможные комбинации действий под каждой ролью. Если под какой-либо ролью обнаруживалось нелегитимное действие, то оно исключалось из прав этой роли. Также корректировки вносились исходя из пожеланий пользователей.

Тестирование API сервиса

Тестирование производилось путем перебора входных данных на каждый метод программы. Все подходы к тестированию можно разделить на категории:

- Корректные входные данные
- Входные данные другого типа данных
- Попытка доступа без авторизации
- Отправка пустого тела запроса

Тестирование веб-сайтов

Тестирование производилось путем перебора возможных действий пользователя в интерфейсе, а также на разных устройствах. В процессе происходила корректировка итогового функционала и внешнего вида сайтов, исходя из пользовательского опыта.

Информация о составе проектной команды, контакты.

Руководитель проекта Минченков Виктор Олегович

DevOps-инженер Петров Артем Эдуардович СКБ181

Контакт для связи: aepetrov_1@edu.hse.ru

DevOps-инженер Рымкулова Диана СКБ 181

Контакт для связи: drymkulova@edu.hse.ru

Подробное описание разработанной системы

Разработанный проект включает в себя три основных компонента:

1. API сервис для авторизации и получения основной информации о кластере
2. Веб-сайт для взаимодействия с API сервисом
3. Веб-сайт с пользовательской документацией

API сервис

Данный компонент создан с целью отображения информации о кластере: содержащаяся рабочая нагрузка, отчеты о проверках безопасности, найденных уязвимостей и получения доступа в кластер.

Помимо этого, он является промежуточным звеном между кластером, личным проектным кабинетом и интерфейсом пользователя, а именно выполняет получение, редактирование и создание содержимого сущностей в кластере.

Перечень методов программы вместе с входными данными

Action	Nodes	State	Cached	Params
\$node.actions \$node.events \$node.health \$node.list \$node.metrics \$node.options \$node.services	(*) 1 (*) 1 (*) 1 (*) 1 (*) 1 (*) 1 (*) 1	OK OK OK OK OK OK OK	No No No No No No No	onlyLocal, skipInternal, withEndpoints, onlyAvailable onlyLocal, skipInternal, withEndpoints, onlyAvailable withServices, onlyAvailable types, includes, excludes onlyLocal, skipInternal, withActions, withEvents, onlyAvailable, grouping
api.addRoute api.listAliases api.removeRoute	(*) 1 (*) 1 (*) 1	OK OK OK	No No No	route, toBottom grouping, withActionSchema name, path
auth.can auth.logout auth.me auth.oidcCallback auth.refresh auth.resolveToken	(*) 1 (*) 1 (*) 1 (*) 1 (*) 1 (*) 1	OK OK OK OK OK OK	No No No No No No	token
v1.cluster.info.getInfo	(*) 1	OK	Yes	
v1.cluster.namespace.create v1.cluster.namespace.list	(*) 1 (*) 1	OK OK	No Yes	name, projectId
v1.cluster.rbac.createRole v1.cluster.rbac.createRoleBinding v1.cluster.rbac.createServiceAccount v1.cluster.rbac.createServiceAccountWithPermissions v1.cluster.rbac.getServiceAccountSecrets	(*) 1 (*) 1 (*) 1 (*) 1 (*) 1	OK OK OK OK OK	No No No No No	namespace, name, rules namespace, name, roleName, serviceAccounts namespace, name namespace, projectId namespace, name
v1.cluster.security.getConfigAudit v1.cluster.security.getVulnerability v1.cluster.security.list	(*) 1 (*) 1 (*) 1	OK OK OK	No No No	namespace, report namespace, report namespace
v1.cluster.workload.list	(*) 1	OK	No	namespace
v1.lk.getProject v1.lk.getUserByEmail v1.lk.getUserProjects	(*) 1 (*) 1 (*) 1	OK OK OK	Yes Yes Yes	id email userId
v1.projects.boilerplateProjectNamespace v1.projects.getMyProjects v1.projects.getNamespaceAccess v1.projects.getNamespaceAdmin v1.projects.getProjectsSecurityConfigAuditReports v1.projects.getProject v1.projects.getProjectSecurityReports v1.projects.getProjectSecurityVulnerabilityReports v1.projects.getProjectWorkload	(*) 1 (*) 1 (*) 1 (*) 1 (*) 1 (*) 1 (*) 1 (*) 1 (*) 1	OK OK OK OK OK OK OK OK OK	No Yes No No No Yes No No No	id projectId namespace, report id id namespace, report id
v1.users.getMyInfo v1.users.getMyProjects	(*) 1 (*) 1	OK OK	No Yes	

Фреймворк moleculer предоставляет возможность посмотреть перечень методов программы. Представлены функции по авторизации; просмотру и созданию namespace, сервис-аккаунтов, рабочей нагрузки, отчетов о безопасности; получению списка проектов и их участников вместе с входными данными.

Технологический стек: docker образ, содержащий nginx сервер и программу, написанную на node.js, typescript, moleculer.

Веб-сайт для взаимодействия с API сервисом

Веб-сервер являющийся графическим пользовательским интерфейсом для взаимодействия с API сервером. Содержит методы по отправке запросов данных с API сервера и визуализация полученных данных

Технологический стек: docker образ, содержащий nginx сервер и программу, написанную на node.js, typescript, vue.



#423. Создание кластерной инфраструктуры для разработки и развертывания программных проектов

Рабочий Программный 2020 - 2022

Управление Пользователи Приложения Отчёты

Рабочая нагрузка

4	1	1	0	2	0	0
pods	replicaSets	deployments	daemonSets	statefulSets	jobs	cronJobs

Deployments

nginx-deployment	Helm Chart:	G.1 поколение	2 обновлено	2 готово	2 цель
nginx-deployment-66b6c48dd5		r.1	2 / 2		
● nginx-deployment-66b6c48dd5-7fl45	nginx:1.14.2			Running	
● nginx-deployment-66b6c48dd5-vftfj	nginx:1.14.2			Running	

Pods

my-release-redis-master-0	docker.io/bitnami/redis:7.0.5-debian-11-r7	Pending
my-release-redis-replicas-0	docker.io/bitnami/redis:7.0.5-debian-11-r7	Pending

Веб-сайт с пользовательской документацией

Данный компонент создан с целью объединить и описать максимальное количество полезной информации для пользователей, которая может понадобится в рамках работы с системой.

Помимо основных страниц добавлен поиск для упрощения пользования и ссылки на API сервис и Gitlab.

Сайт создан на VuePress: <https://vuepress.vuejs.org>

Инструкция по работе >

Gitlab >

Docker >

Helm >

Kubernetes >

Работа с Kubernetes

Объекты Kubernetes

Kubernetes vs Docker compose

Лучшие практики Kubernetes

Kubectl

Установка и настройка kubectl
Синтаксис

Gatekeeper >

Мониторинг >

Установка и настройка kubectl

Kubectl – консольная утилита, без которой даже нельзя представить работу с кластером Kubernetes. Это инструмент командной строки для управления кластерами Kubernetes.

Существует множество способов установки консольной утилиты (в том числе и в зависимости от используемой операционной системы). Например, для "классической" установки kubectl в Ubuntu или Debian, следует воспользоваться следующими командами:

```
sudo apt-get update && sudo apt-get install -y apt-transport-https
curl -s https://packages.cloud.google.com/doc/apt/doc/apt-key.gpg | sudo apt-key add -
sudo touch /etc/apt/sources.list.d/kubernetes.list
echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a /etc/apt/sources.list
sudo apt-get update
sudo apt-get install -y kubectl
```

В операционных системах CentOS, RHEL или Fedora установка будет выглядеть так:

```
cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
```

Экономическая эффективность

Для развертывания, аprobирования тестирования программных проектов для студентов требуется выделение специальных ресурсов с возможностью доступа к ним из внешней сети. В большинстве случаев студенты за свой счет арендуют ресурсы на внешних хостингах, либо администраторы серверов МИЭМ выдают доступ к ресурсам поименно.

Благодаря внедрению кластерной инфраструктуры мы сможем добиться того, что не возникнет необходимости затрачивать средства на дополнительные виртуальные машины и выделять отдельное пространство каждой команде, так как все необходимые возможности будут предоставлены внутри кластера, и несколько проектов будут находиться вместе на эквивалентных узлах Kubernetes.

Помимо этого, учащимся будет предоставлена возможность уже в рамках учебных проектов подробнее разобраться с актуальными инструментами разработки и обеспечению безопасности систем. Тем самым приобретая опыт, который будет учтен при приеме на будущую работу, с возможностью в самых ближайших сроках принести выгоду, как рынку труда, так и стране, в целом, если смотреть в разрезе импортозамещения.

Перечень основных технических и научных результатов

- 1) API сервис для авторизации и получения основной информации о кластере
- 2) Веб-сайт для взаимодействия с API сервисом
- 3) Веб-сайт с пользовательской документацией
- 4) Gatekeeper, развернутый в рамках кластера
- 5) Сервис для выписывания SSL LetsEncrypt сертификатов
- 6) Сервис для проведения проверок безопасности рабочей нагрузки

Примеры работы программного обеспечения

“Мои проекты”

Для того, чтобы ознакомиться со всей информацией по проекту необходимо перейти во вкладку “Мои проекты”.

Мои проекты

Далее необходимо провалиться в интересующий проект, нажав на него.

The screenshot shows the K8S Кластер web interface. At the top, there is a navigation bar with icons for cluster status, user profile, and documentation. Below the navigation bar, the main area is titled 'Мои проекты'. It displays a list of projects, with one project highlighted. The highlighted project card has the following details:

- #423
- Создание кластерной инфраструктуры для разработки и развертывания программных проектов
- Студент
- Развернут на кластере >

A red arrow points from the bottom left towards the 'Развернут на кластере' button.

На текущей вкладке можно увидеть в каком состоянии находится проект (1), к какому типу относится проект (2) и тайминг проекта (3).

Внутри проекта доступны четыре вкладки: “Управление”, “Пользователи”, “Приложения” и “Отчёты” (4).

The screenshot shows the K8S Cluster interface. At the top, there is a header with a gear icon, the text "K8S Кластер", "Мои проекты", "Доступ", and a "Документация" button. On the right side of the header is a user profile icon.

The main content area has a title "#423. Создание кластерной инфраструктуры для разработки и развертывания программных проектов". Below the title is a navigation bar with four tabs: "Рабочий" (1), "Пользователи" (2), "Приложения" (3), and "Отчёты" (4). Red arrows point from the numbers 1, 2, 3, and 4 to their respective tab labels.

Under the "Приложения" tab, there is a section titled "Рабочая нагрузка" showing metrics: 4 pods, 1 replicaSets, 1 deployments, 0 daemonSets, 2 statefulSets, 0 jobs, and 0 cronJobs.

Below that is a "Deployments" section for the "nginx-deployment" Helm Chart. It shows two replicas: "nginx-deployment-66b6c48dd5-7fl45" and "nginx-deployment-66b6c48dd5-vftfj", both running on "nginx:1.14.2". The status is "Running".

At the bottom, there is a "Pods" section listing two pods: "my-release-redis-master-0" and "my-release-redis-replicas-0", both in a "Pending" state.

В рамках “Приложения” можно увидеть текущую рабочую нагрузку: количество различных объектов Kubernetes (pods, replicaSets, deployments,...) (1).

В рамках Deployments (2) можно увидеть информацию о обновлениях, поколении, статусе реплик приложения.

В рамках Pods (3) можно увидеть информацию о используемом базовом образе и статусе Pod.

#423. Создание кластерной инфраструктуры для разработки и развертывания программных проектов

 Рабочий  Программный  2020 - 2022

[Управление](#) [Пользователи](#) [Приложения](#) [Отчёты](#)

1
Рабочая нагрузка

4	1	1	0	2	0	0
pods	replicaSets	deployments	daemonSets	statefulSets	jobs	cronJobs

2
Deployments

nginx-deployment		G.1	поколение	обновлено	готово	цель
Helm Chart:		r.1	2 / 2			
nginx-deployment-66b6c48dd5	nginx:1.14.2			Running		
nginx-deployment-66b6c48dd5-7fl45	nginx:1.14.2			Running		
nginx-deployment-66b6c48dd5-vftfj	nginx:1.14.2					
Pods		3				
my-release-redis-master-0	docker.io/bitnami/redis:7.0.5-debian-11-r7			Pending		
my-release-redis-replicas-0	docker.io/bitnami/redis:7.0.5-debian-11-r7				Pending	

На вкладке “Пользователи” содержится информация об участниках проектах, их должности, уровне доступа и контактной информации.

#423. Создание кластерной инфраструктуры для разработки и развертывания программных проектов

 Рабочий  Программный  2020 - 2022

[Управление](#) [Пользователи](#) [Приложения](#) [Отчёты](#)

УЧАСТНИК И ДОЛЖНОСТЬ	EMAIL	УРОВЕНЬ ДОСТУПА
 Минченков Виктор Олегович Руководитель проекта, Инициатор проекта	vminchenkov@miem.hse.ru vminchenkov@hse.ru	Администратор
 Сергеев Антон Валерьевич Руководитель направления	avsergeev@miem.hse.ru avsergeev@hse.ru	Гость
 Левинсон Матвей Максимович Консультант	mmlevinson@miem.hse.ru mmlevinson@edu.hse.ru	Гость
 Петров Артём Эдуардович DevOps-инженер	aepetrov.1@edu.hse.ru	Разработчик
 Рымкулова Диана - DevOps-инженер	drymkulova@edu.hse.ru	Разработчик
 Безель Максим Алексеевич DevOps-инженер	mabezels@miem.hse.ru mabezels@edu.hse.ru	Разработчик
 Левинсон Матвей Максимович DevOps-инженер	mmlevinson@miem.hse.ru mmlevinson@edu.hse.ru	Разработчик

В рамках страницы “Отчёты” представлены отчёты о проверках безопасности: уязвимости в приложениях и ошибки в конфигурации (1).

Указано количество уязвимостей и ошибок согласно уровню критичности (2).

Для того, чтобы подробнее ознакомиться с ошибками необходимо нажать на специальную кнопку (3).

The screenshot shows the K8S Cluster security report interface. At the top, there are navigation links: K8S Кластер, Мои проекты, Доступ, and Документация. Below the header, a section titled "#423. Создание кластерной инфраструктуры для разработки и развертывания программных проектов" is displayed, along with a date range from 2020 - 2022. The main content area has tabs: Управление, Пользователи, Приложения, and Отчёты. The Отчёты tab is selected. A red arrow labeled '1' points to the "Отчёты о проверках безопасности" section, which contains two categories: "Уязвимости в приложении" and "Ошибки в конфигурации". Red arrows labeled '2' point to the numerical counts of vulnerabilities and errors for each category. Red arrows labeled '3' point to the individual items listed under each category, which are hyperlinks to detailed reports.

Уязвимости в приложении

3 vulnreports 3 configreports

replicaset-nginx-deployment-66b6c48dd5-nginx
library/nginx:1.14.2

statefulset-my-release-redis-master-redis
bitnami/redis:7.0.5-debian-11-r7

statefulset-my-release-redis-replicas-redis
bitnami/redis:7.0.5-debian-11-r7

Ошибки в конфигурации

replicaset-nginx-deployment-66b6c48dd5

statefulset-my-release-redis-master

statefulset-my-release-redis-replicas

В окне “Репорт об уязвимостях” содержится информация о сканировании системы (1) и информация об уязвимостях (2):

- Критичность: насколько критична существующая уязвимость
- Значимость: число от 1 до 10 показывающее насколько уязвимость значима для системы, этот параметр определен в рамках CVE (база данных общезвестных уязвимостей информационной безопасности)
- Идентификатор: Каждой уязвимости присваивается идентификационный номер вида CVE-год-номер, описание и ряд общедоступных ссылок с описанием. В случае нажатия на идентификатор можно перейти на страницу <https://avd.aquasec.com/> с общей информацией об уязвимости
- Компонент: в какой части приложения находится найденная уязвимость
- Описание: описание уязвимости

- Установленная версия: текущая версия компонента, в которой найдена уязвимость
- Версия с исправлением: до какой версии необходимо обновиться, чтобы устранить уязвимость

Репорт об уязвимостях

Обновлено: 2022-09-07T13:57:02Z

App info:							Scanner info:	
criticalCount	highCount	lowCount	mediumCount	noneCount	unknownCount		Scanner name: Trivy	Scanner version: 0.25.2
Image: library/nginx Tag: 1.14.2 Registry: index.docker.io	28	48	4	46	0	7	Scanner vendor: Aqua Security	
КРИТИЧНОСТЬ ЗНАЧИМОСТЬ ИДЕНТИФИКАТОР КОМПОНЕНТ ОПИСАНИЕ УСТАНОВЛЕННАЯ ВЕРСИЯ ВЕРСИЯ С ИСПРАВЛЕНИЕМ								
MEDIUM	5.7	CVE-2020-27350	apt	apt: integer overflows and underflows while parsing .deb packages	1.4.9	1.4.11		
MEDIUM	5.5	CVE-2020-3810	apt	Missing input validation in the ar/tar implementations of APT before v ...	1.4.9	1.4.10		
UNKNOWN		DLA-2948-1	debian-archive-keyring	debian-archive-keyring - security update	2017.5	2017.5+deb9u2		
CRITICAL	9.8	CVE-2022-1664	dpkg	Dpkg::Source::Archive in dpkg, the Debian package management system, b ...	1.18.25	1.18.26		
MEDIUM	6.4	CVE-2019-5094	e2fslibs	e2fsprogs: Crafted ext4 partition leads to out-of-bounds write	1.43.4-2	1.43.4-2+deb9u1		
MEDIUM	7.5	CVE-2019-5188	e2fslibs	e2fsprogs: Out-of-bounds write in e2fsck/rehash.c	1.43.4-2	1.43.4-2+deb9u2		
MEDIUM	6.4	CVE-2019-5094	e2fsprogs	e2fsprogs: Crafted ext4 partition leads to out-of-bounds write	1.43.4-2	1.43.4-2+deb9u1		

В окне “Репорт об ошибках в конфигурации” содержится информация о сканировании системы (1) и информация о проведенных проверках (2):

- Критичность: насколько критична проводимая проверка
- Идентификатор проверки согласно Aqua security
- Название проверки
- Описание проверки
- Сообщение об ошибке, если проверка не пройдена
- Информация о прохождении проверки, в случае, если проверка пройдена успешно строка останется белого цвета и в соответствующей графе появится галочка, если проверка не пройдена, то строка подсвечивается и в графике будет стоять крестик. Такая логика приложения заложена специально, чтобы пользователи видели все проводимые проверки, а не только неудавшиеся. Это позволит накопить больше знаний о популярных ошибках конфигурации.

Репорт об ошибках в конфигурации						
				Scanner info:		
criticalCount 0				Scanner name: Starboard Scanner version: 0.15.6 Scanner vendor: Aqua Security		
КРИТИЧНОСТЬ	КАТЕГОРИЯ	ИДЕНТИФИКАТОР	НАЗВАНИЕ	ОПИСАНИЕ	СООБЩЕНИЕ	SUCCESS
MEDIUM	Kubernetes Security Check	KSV027	Non-default /proc masks set	The default /proc masks are set up to reduce attack surface, and should be required.	✓	
LOW	Kubernetes Security Check	KSV015	CPU requests not specified	When containers have resource requests specified, the scheduler can make better decisions about which nodes to place pods on, and how to deal with resource contention.	["Container 'nginx' of ReplicaSet 'nginx-deployment-66b6c48dd5' should set 'resources.requests.cpu'"]	✗
HIGH	Kubernetes Security Check	KSV005	SYS_ADMIN capability added	SYS_ADMIN gives the processes running inside the container privileges that are equivalent to root.	✓	
LOW	Kubernetes Security Check	KSV013	Image tag ':latest' used	It is best to avoid using the ':latest' image tag when deploying containers in production. Doing so makes it hard to track which version of the image is running, and hard to roll back the version.	✓	

“Доступ”

На данной странице описана инструкция по получению доступа к кластеру:
 получение токена от сервисного аккаунта Kubernetes (1) и инструкция по установке (2).

Получение доступа

Доступ к кластеру осуществляется при помощи клиента `kubectl`. В кластере у Вас будет доступ к пространствам Ваших проектов на основании данных по проектной деятельности в Личном Кабинете.

Получение токена

Сгенерировать токен

Установка токена

Для установки токена и получения доступа к кластеру при помощи утилиты `kubectl` вам необходимо выполнить следующие команды:

- Добавление параметров подключения к кластеру

```
kubectl config set-cluster "miem-k8s" --server="https://94.79.55.21:6443" --certificate-authority-data="LS0tLS1CRUdJTiBDRVJUSUZJQ0Fj
```

- Добавление учетной записи

```
kubectl config set-credentials "<username>" --token="<token>"
```

- Создание контекста из кластера и учетной записи

```
kubectl config set-context "miem-k8s" --cluster="miem-k8s" --user="<username>"
```

- Смена текущего контекста

```
kubectl config use-context "miem-k8s"
```

Работа данных команд добавляет соответствующие настройки в файл конфигурации (по умолчанию `~/.kube/config`). Он отвечает за все параметры доступа к кластеру, авторизацию и контексты.

При необходимости вы можете его отредактировать вручную.

“Управление”

На текущей странице можно создать и удалить namespace.
А также выдать или удалить доступ у участника команды по необходимости.
Данная страница доступна только администратору проекта.

The screenshot shows the 'Management' tab selected in the top navigation bar. Below it, a sub-section titled '#423. Создание кластерной инфраструктуры для разработки и развертывания программных проектов' is displayed. Under this, there are buttons for creating ('Создать namespace 423') and deleting ('Удалить namespace 423'). A table lists project members with their roles and access levels. The table includes columns for 'УЧАСТНИК И ДОЛЖНОСТЬ', 'EMAIL', 'УРОВЕНЬ ДОСТУПА', and 'УПРАВЛЕНИЕ ДОСТУПОМ'. Buttons for 'Выдать' (Grant) and 'Удалить' (Delete) are present in the last two columns.

УЧАСТНИК И ДОЛЖНОСТЬ	EMAIL	УРОВЕНЬ ДОСТУПА	УПРАВЛЕНИЕ ДОСТУПОМ
Минченков Виктор Олегович Руководитель проекта, Инициатор проекта	vminchenkov@miem.hse.ru vminchenkov@hse.ru	Администратор	
Сергеев Антон Валерьевич Руководитель направления	avsergeev@miem.hse.ru avsergeev@hse.ru	Гость	
Левинсон Матвей Максимович Консультант	mmlevinson@miem.hse.ru mmlevinson@edu.hse.ru	Гость	
Петров Артём Эдуардович DevOps-инженер	aepetrov_1@edu.hse.ru	Разработчик	
Рымкулова Диана - DevOps-инженер	drymkulova@edu.hse.ru	Разработчик	+ Выдать
Безель Максим Алексеевич DevOps-инженер	mabezel@miem.hse.ru mabezel@edu.hse.ru	Разработчик	- Удалить
Левинсон Матвей Максимович DevOps-инженер	mmlevinson@miem.hse.ru mmlevinson@edu.hse.ru	Разработчик	+ Выдать

“Документация”

Для того, чтобы ознакомиться с пользовательской документацией необходимо перейти на соответствующую вкладку.

Документация

После чего откроется страница с существующей документацией. Для того, чтобы ознакомиться с конкретной темой можно открыть соответствующий раздел в меню(1), либо воспользоваться поиском по ключевым словам (2). С пользовательской инструкцией по взаимодействию с системой также можно ознакомиться в рамках документации (3). Помимо этого есть возможность перейти напрямую из документации в Gitlab (4) и проектный офис (5).

K8S Кластер

Инструкция по работе > 3

Gitlab

Docker 1

- Работа с Docker
- Установка и работа с Docker
- Основные понятия
- Основные команды Docker
- Загрузка образов Docker
- Лучшие практики Docker
- Docker swarm
- Docker compose

Helm

Kubernetes

Gatekeeper

Мониторинг

Установка и работа с Docker

Установка Docker происходит по разному в зависимости от операционной системы. Один из способов: воспользоваться менеджером пакетов YUM (Yellowdog Updater Modified) для Linux - систем.

```
yum install docker
```

После установки необходимо проверить корректность работы сервиса:

```
systemctl start docker
```

```
systemctl enable docker
```

```
systemctl status docker
```

Далее необходимо запустить тестовый образ, чтобы проверить правильность работы Docker:

```
docker run hello-world
```

MIEM Gitlab 2 4

MIEM Projects 5

Ход работ

1. На первом этапе были изучены итоги и результаты работы над проектом прошлого года для того, чтобы спроектировать план действий на текущий год
2. После того, как были распределены задачи, перешли к настраиванию RBAC (Role-based access control)
3. После завершения настройки было проведено тестирование ролей для пользователей
4. Была осуществлена установка и настройка Gatekeeper (реализация Open Policy Agent (OPA) для Kubernetes, которая работает в качестве Webhook для валидации манифестов)
5. По итогу настройки проведено тестирование политик Gatekeeper
6. Была осуществлена настройка LetsEncrypt
7. Был добавлен сервис по генерации отчетов тестирований
8. Реализовано отображение результатов сканирования Starboard в front-end приложении
9. Была добавлена авторизация на обращения к методам API сервиса, содержащая конфиденциальные данные
10. Помимо этого добавлен процесс онбординга новых пользователей
11. На протяжении всего процесса разработки была написана пользовательская документация
12. Изучен функционал VuePress для публикации документации и осуществлен переезд документации на эту платформу
13. Осуществлена интеграция Starboard для проверки безопасности приложений
14. Реализована адаптация чарта MongoDB для запуска в кластере
15. Осуществлена адаптация чарта PostgreSQL для запуска в кластере
16. Создан автоматизированный pipeline для удаления ресурсов в namespace

Роли участников команды

Петров Артем: конфигурация RBAC, разработка backend и frontend приложений, написание политик Gatekeeper, создание Halm-чартов, настройка автоматического сканера уязвимостей рабочей нагрузки.

Рымкулова Диана: ведение документации по проекту, написание приложения для публикации пользовательской документации, осуществление тестирования, реализация графической составляющей проекта (постер, презентация, ux/ui frontend-сервиса и документации).

Новизна/преимущества решений, полученных по результатам проекта

Проект был реализован с целью погрузить студентов в формат коммерческой разработки продукта, что до этого в рамках образовательных программ реализовано не было. Помимо этого, данное решение поможет существенно сократить экономические затраты на дополнительные ресурсы в рамках проектной деятельности университета. Сервис поможет в изучении и получении практических навыков разработки безопасных систем, а также познакомит студентов с актуальными инструментами работы.