УДК 004.7 https://doi.org/10.33099/2304-2745/2024-3-83/31-39

Капілевич В. О. (0000-0001-9025-7608)
Ліпко І. О. (0009-0001-6663-5899)
Звір В. Б. (0000-0002-6823-7552)

Центр воєнно-стратегічних досліджень Національного університету оборони України, Київ

Аналіз використання технології керування контейнерами під час побудови інформаційної інфраструктури Міністерства оборони України

Резюме. Розглянуто програмні рішення використання технології керування контейнерами (оркестрування) у сфері оборони, яке полягає у поєднанні сфер розробки проєктів цифровізації та безпосередньої операційної діяльності. Це дає змогу автоматизувати розгортання контейнерів, організувати їх масштабування і керування ними в різнорідних хмарних середовищах та забезпечити своєчасне планування, створення, безперервну інтеграцію та постачання необхідних застосунків (програмних засобів) для надання ІТ-сервісів.

Ключові слова: оркестрування (керування контейнерами); контейнеризація; кластерне хмарне середовище.

Постановка проблеми. В умовах широкомасштабної збройної агресії, направленої проти України, набуває важливого значення забезпечення стійкості обчислювальних ресурсів, за допомогою яких надаються критично важливі ІТ-сервіси як для цивільних, так і для військових потреб. Отже, у загальному, в частині що стосується воєнної сфери, таку *стійкість* можливо охарактеризувати двома чинниками: здатністю протистояти зовнішнім атакам або внутрішнім збоям та залишатися спроможними надавати необхідні ІТ-сервіси в підтримку забезпечення бойових дій. Вказане можливо досягти шляхом швидкого розгортання застосунків (програмних засобів), за допомогою яких надаються необхідні ІТ-сервіси, у хмарному середовищі (центрі обробки даних) із використанням такої складової технології віртуалізації, як контейнеризація. Технологія контейнеризації дає змогу "упаковувати" застосунки у контейнери, які є єдиним автономним функціонуючим блоком та які можливо масштабувати (розгортати, дублювати) і здійснювати керування ними у кількох хмарних середовищах.

Довідка. Під *контейнеризацією* розуміється розгортання застосунків (програмних засобів) разом з необхідними компонентами (наприклад, програмний код (або визначена версія програмного коду) разом із необхідними бібліотеками, каталогами, фреймворками тощо), таким чином, щоб вони були ізольовані у власному контейнері.

Для автоматизації розгортання, масштабування та керування контейнерами у хмарному середовищі використовують технологію керування контейнерами або оркестрування контейнерів (у подальшому використовується термін "оркестрування"), що включає планування та запуск контейнерів в різних типах (приватних, публічних, гібридних, мультихмарах) хмар (або центрах обробки даних). До того ж на кількох обчислювальних вузлах *(en: nodes)*, розподіл обчислювальних ресурсів між контейнерами здійснюється таким чином, що внаслідок застосування вказаних технологій ІТ-сервіси залишаються доступними, навіть якщо вузли виходять з ладу.

Довідка. *Оркестрування контейнерів (en: container orchestration)* являє собою процес керування набором контейнерів або подами, які функціонують разом як частина застосунка.

Поштовхом із використання хмарних обчислень у державному секторі стало прийняття 17 лютого 2022 року Закону України "Про хмарні послуги" [1] та набрання ним чинності з липня 2022 року. У зв'язку із військовою агресією Російської Федерації проти України та введенням в Україні воєнного стану Кабінетом Міністрів України уточнені питання [2] щодо можливості розміщувати державні інформаційні ресурси, у т.ч. створення їх додаткових резервних копій, у хмарних середовищах, що розташовані поза межами України. Після внесення у червні 2022 року до цього Закону [1] змін, Міністерству оборони України (Міноборони) та Збройним Силам України було надано дозвіл у період дії воєнного стану на розміщення інформаційних ресурсів воєнної та оборонної сфери у приватних хмарах (центрах обробки даних), у т.ч. розміщених за кордоном.

Зі свого боку, основною складовою єдиної інформаційної системи Міноборони, порядок функціонування якої визначено наказом Міноборони від 21.03.2024 № 188 [3], є центри обробки даних. До того ж для забезпечення функціонування складових єдиної інформаційної системи Міноборони можуть використовуватися зовнішні (українські та міжнародні) хмарні інфраструктури. Отже, інформаційна інфраструктура Міноборони буде складатися із власних центрів обробки даних та орендованих хмарних середовищ (інфраструктур), які розташовані як на території України, так і поза межами України.

Також, відповідно до положень вказаного Закону [1] особливості порядку застосування технології хмарних обчислень / послуг вирішуються Міноборони із урахуванням посилення національної безпеки у воєнній та оборонній сферах. На цей час, такі особливості у Міноборони не визначено, робота з цього приводу продовжується. Указані особливості мають ураховувати запровадження кращих практик реалізації новітніх інформаційних технологій у сфері хмарних обчислень, прикладом такої реалізації є технологія керування контейнерами (оркестрування), основою якої є технологія контейнеризації.

Таким чином, потребує необхідність (доцільність) проведення аналізу використання технології керування контейнерами (оркестрування), як можливої технології захисту даних та інформації, для більш широкого та масштабованого застосування цієї технології під час нарощення спроможностей інформаційної інфраструктури Міноборони.

Аналіз останніх досліджень та публікацій. Протягом останніх років було проведено низку досліджень щодо можливості запровадження хмарних технологій (технологій хмарних обчислень) у Міноборони. Так, у [4–6] розглянуті питання запровадження хмарних технологій для потреб оборони з використанням центрів обробки даних в приватних хмарах. Дослідники звернули увагу на аспекти створення кластерного середовища за допомогою центрів обробки даних, орієнтованого на зберігання та обробку великих масивів даних, проаналізовано актуальні підходи до побудови інформаційної інфраструктури на основі хмарних технологій.

У НАТО також приділяється значна увага щодо використання технології хмарних обчислень. Активне запровадження хмарних технологій в НАТО розпочалось із прийняттям Політики НАТО із хмарних обчислень у 2016 році. Так, у моделі стеку хмарних обчислень, описаною у вказаній Політиці, визначено використання такого компоненту як оркестрування, як складової моделі "Платформа як сервіс" *(en: Platform as a service, PaaS)*.

Довідка. Політика НАТО із хмарних обчислень є однією із політик НАТО щодо запровадження інформаційних технологій та викладеній у загальній політиці НАТО – C3 політика Альянсу *(en: Alliance C3 Policy)* [7].

У подальшому питання щодо запровадження сервісів оркестрування та контейнеризації були відображені у C3 Таксономії НАТО [8]. Згідно цієї класифікації сервісів оркестрування і контейнеризація є базовими сервісами та входять до складу сервісів платформ, які надають засоби для координації виконання декількох технічних сервісів таким чином, що вся сукупність технічних сервісів виглядає як єдиний, агрегований (об'єднаний) технічний сервіс, який відповідає на один індивідуальний запит.

Примітка. Розгортання контейнерних застосунків є достатньо новим поняттям у технології віртуальної обробки даних у хмарному середовищі. На відміну від віртуальних машин, для роботи контейнера відсутня необхідність у встановленні окремої операційної системи для забезпечення функціонування застосунка, декілька контейнерів можуть працювати у межах однієї операційної системи та використовувати обчислювальні ресурси цієї операційної системи.

Однак у 4 спіралі ініціативи НАТО FMN, яку затвердили у лютому 2021 року та якою описано вимоги до технічних сервісів, що мають бути реалізовані в комунікаційних та інформаційних системах НАТО, визначені вимоги до віртуальної обробки із використанням лише віртуальних машин, а саме:

сервісною інструкцією з віртуальної обробки визначено вимоги щодо підтримки сервісів віртуальної обробки для обміну віртуальними пристроями між різними гостьовими платформами;

визначено використання комерційних пропрієтарних форматів файлу, розроблених компаніями VmWare (формат файлу .vmdk) та Microsoft (формати файлів .vhd та .vhdx), для використання у вигляді віртуального образу жорсткого диску у віртуальних машинах.

Довідка. Більш детальніше про зміст зазначених документів НАТО, пропозиції щодо їх запровадження у Міноборони та ініціативу НАТО FMN наведено у [9, 10].

Отже, як показує зміст вказаних документів, в НАТО продовжуються дослідження щодо запровадження технологій керування контейнерами та їх технічної реалізації в комунікаційних та інформаційних системах НАТО.

Також слід відмітити, що стратегічним документом міністерства оборони США [11], який було презентовано міжнародній спільноті у 2020 році, передбачено запровадження таких новітніх технологій як мобільних 5G-мереж, хмарних обчислень, штучного інтелекту тощо.

Ураховуючи вказане, у державах – членах НАТО проводяться дослідження щодо застосування хмарних обчислень у військовій сфері. Наприклад, дослідження [12–14] були направлені на використання контейнерних програмних рішень в об'єднаному хмарному середовищі, огляд питань взаємосумісності кластерів на базі платформи Kubernetes. У той же час, у 2021 році НАТО за результатами міжнародного конкурсу було вибрано компанію Thales з метою розроблення рішення щодо розгортання військової хмари на театрі операцій за менш ніж 24 години [15]. А міністерством оборони США у 2022 році було укладено контракти із компаніями Google, Amazon, Microsoft та Oracle на суму 9 мільярдів доларів на створення спільної хмарної системи ведення бойових дій *(en: Joint Warfighting Cloud Capability, JWCC)* [16].

Таким чином, використання хмарних обчислень у військовій сфері набуває все більш широкого застосування. Тому, аналіз програмних рішень, за допомогою яких реалізовані технології керування контейнерами (оркестрування), для розуміння шляхів їх подальшого застосування в інформаційній інфраструктурі Міноборони в умовах збройної агресії є актуальним завданням.

Метою статті є проведення аналізу використання програмних рішень технології керування контейнерами (оркестрування) для можливої реалізації цих рішень під час побудови інформаційної інфраструктури Міноборони.

Викладення основного матеріалу. На сьогодні програмними рішеннями (інструментами), які найбільш широко використовуються під час створення та керування контейнерів, є:

Docker: середовище контейнеризації з відкритим кодом, функціоналом якого є створення, розгортання та тестування контейнерних застосунків на різних операційних системах (як-то, Windows, Linux або MacOS);

Kubernetes: програмна платформа з відкритим кодом, розроблена компанією Google, яка використовується для розгортання, масштабування мікросервісів і керування ними, автоматизації керування контейнерами (оркестрування);

VMware Tanzu: програмна платформа, розроблена компанією VMware, яка використовується для керування хмарним середовищем, побудованому на базі програмного забезпечення VMware, на основі контейнерів застосунків Kubernetes;

Linux Containers: програмні засоби, які використовуються для розгортання контейнерних застосунків на базі операційної системи Linux.

У цій статті розглядаються особливості застосування програмних рішень (інструментаріїв) Docker та Kubernetes, які, на думку авторів, є найбільш доцільними при розгортанні сервісів контейнеризації та оркестрування в інформаційній інфраструктурі Міноборони.

Віртуалізацію інформаційної інфраструктури та подальшу контейнеризацію застосунків (програмних засобів) *(en: application)* можливо розглядати як сучасний та ефективний підхід до керування та використання ІТ-ресурсів комунікаційних та інформаційних ресурсів у хмарному середовищі. У цьому випадку, такий підхід керування контейнерами забезпечує мережевий доступ на вимогу до спільного пулу конфігурованих обчислювальних ресурсів, які можливо швидко надати та вивільнити з мінімальними зусиллями адміністратора або організації за чіткої взаємодії з постачальником сервісів.

Основу хмарного середовища мають складати розподілені кластери із контейнерами, що забезпечують функціональні спроможності для розгортання служб (сервісів) на окремих вузлах, кінцевих пристроях та інших мобільних чи роботизованих платформах (прикладами таких пристроїв або платформ є зразки озброєння, військової та спеціальної техніки, шолом солдата, БпЛА). Зі свого боку, це дає змогу керувати бажаним станом у

різноманітних (гетерогенних) кластерних середовищах за допомогою оркестрування контейнерів, розгорнутих та запущених на багатьох платформах одночасно.

Сервіси контейнеризації забезпечують логічні виконувані модулі програмних засобів, за допомогою яких застосунки можливо абстрагувати від середовища, в якому вони фактично працюють. Таке відокремлення дає змогу легко та узгоджено розгортати застосунки на основі контейнерів на різних ІТ-платформах та в різних ІТ-середовищах, тобто, здійснювати їх швидке масштабування без зайвих затрат на узгодження на програмному рівні. Контейнеризація забезпечує чітке розмежування між етапом розробки застосунка з визначеною логікою, бібліотеками і залежностями та етапом розгортання застосунка, який зосереджено на управлінні якістю обслуговування, робочим навантаженням і масштабом.

Разом з тим, сервіси контейнеризації можуть забезпечити контрольований ієрархічний доступ до іменованих контейнерів зберігання даних як до окремого об’єкту у вигляді частини інформації, що міститься в документі, на веб-сайті чи будь-якому іншому контейнері, здатному зберігати та передавати (поширювати) інформацію. Це дає змогу прозоро обробляти, здійснювати фрагментацію, кешування та цілісність зберігання даних, що, у свою чергу, забезпечує логічний доступ до них абстрагуючись від топології фізичного зберігання.

Для забезпечення вказаного функціоналу архітектура сервісів контейнеризації має декілька рівнів:

інфраструктурний (апаратний) рівень: апаратне забезпечення обчислювального вузла;

рівень операційної системи: можливе використання локальної та віртуальної операційної системи;

рівень образів контейнера: файли, які містять інформацію щодо розгортання контейнера із застосунком;

контейнерний рушій: програмний засіб, який являється посередником між операційною системою та контейнером і за допомогою якого здійснюється створення контейнерів відповідно до інформації образів контейнерів. При цьому, такі програмні засоби можуть здійснювати керування одним або декількома контейнерами в одній операційній системі. До того ж забезпечується відповідна ізоляція між контейнерами;

застосунок та визначення залежностей: код застосунку та інші пов’язані файлі, необхідні для роботи застосунку: пов’язані бібліотеки, файли конфігурації, бази даних тощо.

Ураховуючи описану архітектуру, до основних етапів конфігурування контейнерів відносяться:

створення образу програмного застосунку із необхідними бібліотеками для подальшого тиражування зразка цього застосунку у вигляді контейнера;

розгортання та запуск контейнера зі створеного образу;

налаштування (оновлення) контейнера;

зупинка чи видалення контейнера;

видалення створеного образу.

Наприклад, у середовищі Docker використовується декілька способів встановлення необхідних програмних засобів або швидкого відновлення їх функціонування на інших потужностях у разі відмови обладнання за допомогою таких компонентів:

середовище виконання (рушій) Docker Engine;

пакетний менеджер Docker Compose;

інструмент Docker swarm mode.

Можливість тиражування (масштабування) контейнерів з програмними застосунками на базі створеного образу надає змогу переналаштовувати такі застосунки без суттєвих дій в адмініструванні, зі збереженням усіх налаштувань для користувачів будь-яких платформ, що підтримують роботу з контейнерами. Це забезпечує швидке перенесення програмних засобів між різними хмарними середовищами. Разом з тим, звертається увага на механізми для збереження даних, які використовуються контейнерами [17]:

монтування (приєднання) елементів файлової системи (Рис. 1);

використання томів (Рис. 2).

Якщо використовувати перший механізм збереження даних (Рис. 1), то каталог чи файл вузла, на якому розгорнуто контейнер, буде змонтовано (приєднано) безпосередньо в сам контейнер. Такий механізм залежить від файлової системи вузла, яка має певну структуру каталогів [17].

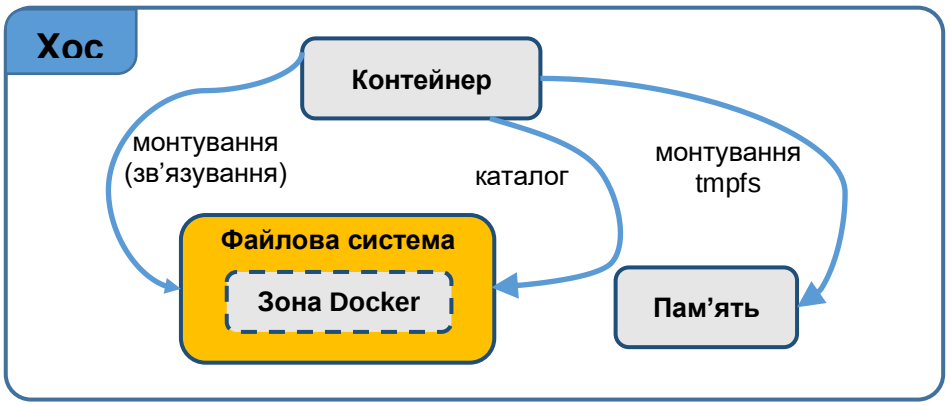

Рис. 1. Механізм збереження даних за допомогою

Другий механізм збереження даних (Рис. 2) передбачає створення томів, вмістом яких повністю керує Docker, а не операційна система вузла. Це забезпечує зберігання даних поза межами контейнеру і запобігає збільшенню його розміру, оскільки томи існують поза життєвим циклом контейнера. Такі томи можна розподіляти між кількома контейнерами, що дає змогу зберігати їх на віддалених вузлах або інших хмарних середовищах, шифрувати їх вміст або додавати інші необхідні функції, при чому вміст нового тому може бути попередньо заповнений контейнером [17].

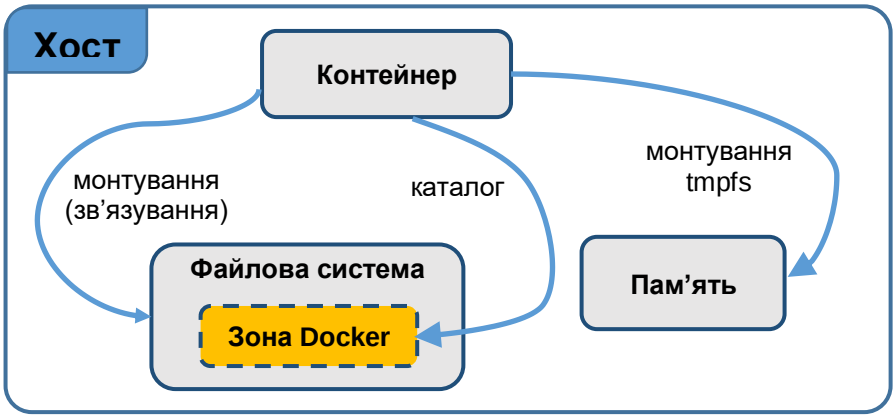

Рис. 2. Механізм збереження даних за допомогою

У кластерному середовищі, організованому за допомогою Kubernetes, у якому об'єднано декілька вузлів, зазвичай, вони поділяються та головний вузол *(en: master node)* і декілька робочих вузлів *(en: worker nodes)* [18]. Робочі вузли функціонують як віртуальні машини та контролюють ресурси, які використовуються різними застосунками та сервісами, що згруповані у набори контейнерів або поди *(en: pods)*. Головний вузол відповідає за підтримку та керування станом кластера із робочими вузлами, на яких розгорнуто різноманітні програмні засоби та сервіси, а також розподіляє навантаження на робочі вузли з наборами контейнерів. Хоча набір контейнерів функціонує як окреме програмне середовище та, зазвичай, ізольований від решти кластера, він зберігає здатність комунікувати та обмінюватися даними з іншими подібними наборами контейнерів. Він може складатися з одного або кількох контейнерів, але кожен такий набір обмежений розміром вузла та не може бути фрагментований між кількома вузлами.

Можливий склад головного та робочих вузлів та призначення їх компонентів наведено у Табл. 1, 2.

Таблиця 1

Компоненти головного вузла

Назва компонента	Призначення компонента
Сервіс ***etcd***	Використовується як сховище даних для конфігурації контейнерів із програмними застосунками
Менеджер	Запускає процеси контролера (вузла, реплікації, кінцевих пристроїв, облікових записів

Назва компонента	Призначення компонента
kube-controller-manager	та токенів), відповідального за прив'язку сервісів та наборів контейнерів для кінцевих пристроїв, а також за створення облікових записів та відповідних токенів доступу, ведення необхідної кількості наборів контейнерів та забезпечення їх реплікації на інші вузли у випадку збоїв
Сервер ***kube-apiserver***	Обробляє команди для наборів контейнерів у кластері, тобто забезпечує розгортання декількох екземплярів. За допомогою HTTP / REST API (API з кінцевим пристроєм HTTP / REST) забезпечується взаємодія з іншими необхідними компонентами як головного, так і робочих вузлів, що надає змогу горизонтально масштабувати контейнери із програмними застосунками
Сервер доменних імен ***kube-dns***	Є доповненням, яке можливо включити до набору контейнерів. Даний компонент веде актуальні DNS-записи для IP-адрес сервісів, що надає можливість розпізнати ім'я сервісу, зареєстрованого після запуску контейнера із програмним застосунком
Сервіс ***kube-proxy***	Реалізовується на головному та робочих вузлах кластера. За допомогою сервісу можливо сформувати мережеві правила, щоб дозволити чи заборонити мережеве підключення до набору контейнерів для балансування навантаження в середині чи ззовні кластера
Планувальник завдань ***kube-scheduler***	Визначає які вузли відповідають вимогам та обмеженням для розгортання набору контейнерів шляхом надання рейтингу цим вузлам. При цьому, планувальник перевіряє доступність та спорідненість обраних вузлів за критерієм живучості

Таблиця 2

Компоненти робочого вузла

Назва компонента	Призначення компонента
Агент ***kubelet***	Підключається до головного вузла і відслідковує актуальність та працездатність всіх наборів контейнерів на робочому вузлі кластера
Сервіс ***kube-proxy***	Реалізовується на головному та робочих вузлах кластера. За допомогою сервісу можливо сформувати мережеві правила, щоб дозволити чи заборонити мережеве підключення до набору контейнерів для балансування навантаження в середині чи ззовні кластера.
Набори контейнерів:	Комунікація з іншими наборами контейнерів, що знаходяться у спільному кластері.
службові набори контейнерів;	Відслідковують доступність наборів контейнерів при балансуванні навантаження в кластері, а також забезпечують роботу мережевих та інших системних сервісів.
пакетні набори контейнерів	Відслідковують розгортання та фактичне виконання та завершення завдань

У Kubernetes сервіс розглядається як абстрактний спосіб уявлення програмного засобу у вигляді логічного набору контейнерів та відповідних політик доступу. Типи сервісів, які використовуються у Kubernetes, наведено у Табл. 3.

Таблиця 3

Типи сервісів у Kubernetes

Назва типу сервісу	Опис сервісу
ClusterIP	Надає сервіс внутрішній IP-адресі кластера, що робить сервіс доступним лише з кластера. Такий тип сервісу визначається по замовчуванню
NodePort	Розкриває сервіс на IP-адресі кожного вузла за статичним номером порту. Такий тип сервісу направляє запити до сервісу ***ClusterIP***, який створюється автоматично. Сервіс ***NodePort*** доступний за межами кластера і може бути використаний на етапі розробки та за умови наявності достатньої кількості портів
LoadBalancer	Надає зовнішній доступ за допомогою сервісу балансування навантаження від постачальника хмарного середовища. Такий тип сервісу направляє запити до сервісів ***NodePort*** та ***ClusterIP***, які створюються автоматично

Під час формування набору контейнерів, виділяють основний та додаткові контейнери. Додаткові контейнери допомагають основному покращити його функціонал та адаптувати його до середовища розгортання. У свою чергу це створює абстракцію окремого контексту з використанням спільної файлової системи та простору імен мереж.

Для роботи з наборами контейнерів в Kubernetes виділяють декілька шаблонів конфігурацій, інформація про які наведена у Табл. 4.

Таблиця 4

Шаблони конфігурацій контейнерів Kubernetes

Назва шаблону конфігурації контейнера	Опис шаблону конфігурації контейнера
Контейнери (ініціалізації) ***Init***	Запуск та завершення роботи всіх додаткових контейнерів (тобто, контейнерів ініціалізації) відбувається до початку роботи основного контейнера в наборі контейнерів. Як правило, ініціалізація охоплює окремий підготовчий життєвий цикл, при якому виконується послідовна інсталяція програмного застосунку (налаштування баз даних чи надання дозволу на необхідні елементи файлової системи), необхідного для запуску та подальшої роботи основного контейнера
Контейнери ***Sidecar***	У такому наборі контейнерів, додатковий контейнер запускається разом із основним та використовується для пересилання файлів журналів, відстеження за оновленням конфігураційних змінних або реалізації додаткових мережевих завдань. Додатковий контейнер включає нестандартні або службові функції, що забезпечують розширення та покращення роботи основного контейнера. Це дає змогу додати бажані функції до основного контейнера, не змінюючи його
Контейнери ***Adapter***	Додатковий контейнер забезпечує уніфікований інтерфейс для роботи основного контейнера з іншими контейнерами в зовнішньому середовищі. Такі контейнерні адаптери використовують для перетворення даних та протоколів, за якими функціонує основний контейнер, щоб працювати за сумісними стандартами з іншими сторонами. Це дає змогу інтегрувати зацікавлені сторони до основного контейнера, який працює за власним форматом, що забезпечує єдиний доступ до централізованих послуг, реалізованих сервісами основного контейнеру
Контейнери ***Ambassador***	Використання такого набору контейнерів обумовлене потребою приховати складність зовнішніх ресурсів від основного контейнера, підключеного безпосередньо до додаткового контейнера, який з'єднує та абстрагує визначений діапазон потенційно складних віддалених ресурсів. Додатковий контейнер спрощує доступ до важкодоступних сервісів за межами набору контейнерів, що забезпечує єдиний інтерфейс доступу до потрібних сервісів. Таким чином, це надає змогу основному контейнеру сприймати відповіді віддалених сервісів, конвертовані додатковим контейнером, та не звертати уваги на реальне середовище розгортання для підключення до віддалених сервісів

У зв'язку з тим, що основний та додатковий контейнери у зазначених у Табл. 4 шаблонах ***Sidecar***, ***Adapter*** та ***Ambassador*** працюють паралельно, необхідно враховувати налаштування для їх сумарних обмежень ресурсів.

Сучасним прикладом використання підходів керування контейнерами в корпоративному середовищі є розгорнуті в контейнерах програмні застосунки розподіленої системи керування версіями файлів та спільної роботи Git *(https://git-scm.com/)* та репозиторію GitLab *(https://gitlab.com)*, що забезпечують роботу з ІТ-проєктами. Указані програмні засоби є безкоштовними та можуть бути застосовані під час удосконалення інформаційної інфраструктури Міноборони за напрямом супроводження ІТ-проєктів цифрової трансформації нашого відомства. Розгортання ІТ-проєктів у програмних контейнерах для спільної роботи зацікавлених сторін (замовників та виконавців) надає можливості виконувати наступні функції:

зберігання програмного коду та історії його змін;

зберігання інформації про користувачів, які змінюють код;

повернення до попереднього коду будь-якої версії;

об'єднання різних версій, зміни версій;

підготовки кінцевого коду до випуску чергового релізу.

Таким чином, використання підходів керування контейнерами (оркестрування) у сфері оборони полягає у поєднанні сфер розробки проєктів цифровізації та безпосередньої операційної діяльності. Це дає змогу замовникам і командам виконавців проєктів цифровізації (таким проєктом у сфері оборони, наприклад, може бути обробка та зберігання потокового відео від БпЛА: створення контейнерів, масштабування їх образів в інтересах підрозділів різних видів та родів військ (сил), у т.ч. інших складових сил оборони, керування цими кнтейнерами) покращити співпрацю та забезпечити своєчасне планування, створення, безперервну інтеграцію та безперервне постачання *(en: Continuous Integration / Continuous Delivery)* вкрай необхідних програмних рішень. У цьому випадку, інтеграція досягається завдяки об'єднанню програмного коду у централізованому сховищі, а безперервне постачання спрямоване на автоматизоване розгортання нових версій програмного рішення у середовище їх функціонування, будь-то для підтримки

військових операцій чи управління повсякденною діяльністю. Завдяки використанню підходів керування контейнерами скорочується час розгортання програмних рішень, і, відповідно, час відновлення їх працездатності після можливих збоїв. За рахунок ізоляції програмного рішення в окреме контейнерне середовище, унеможливлюються конфліктні випадки функціонування між версіями цього програмного рішення та пов'язаними бібліотеками.

У сучасних умовах військової агресії проти України, сили оборони України прагнуть мати інформаційну перевагу над ворогом, користуючись даними від нових джерел інформації, які постійно змінюються (прикладом такого є стримінгові відео-потоки від БпЛА). В багатьох випадках, це призводить до зміни моделей даних, процедур чи сервісів роботи з цими даними, що в свою чергу може призвести до високих операційних витрат та збільшення часу на розгортання оновленого функціоналу програмних рішень. Підходи щодо керування контейнерами націлені на вирішення цих проблем, які доцільно використовувати для потреб сил оборони України.

Необхідність постійного використання сенсорів та платформ військового призначення, які генерують дані, призводить до збільшення кількості даних для обробки. Пов'язані з цим ресурси та сервіси повинні бути постійно доступними, щоб швидко приносити користь тим, хто приймає рішення. Для того, щоб більше інформації стало доступним якомога швидше, необхідно забезпечити стійкість роботи цих ресурсів та сервісів, а також можливість їх відновлення (чи оновлення) в умовах динамічної зміни обстановки.

Автоматична реєстрація в кластері програмного контейнера, що реалізує сервіс, забезпечує їх видимість та доступність для будь-якої іншої військової платформи, яка функціонує в спільному кластері об'єднаного хмарного середовища. Тобто, коли відсутні потужності для обробки даних поблизу місця їх походження, необхідні ресурси та сервіси для цього можна знайти у кластері завдяки тому, що програмні контейнери з інших військових платформ діляться ними для спільного користування у об'єднаному хмарному середовищі.

Висновки. Технологія керування контейнерами (оркестрування) являє собою не лише сучасні інформаційні технології, а й новий підхід до обробки даних та забезпечення їх захисту, запровадження яких у Міноборони дозволить пришвидшити розгортання (масштабування) ресурсів та сервісів в різнорідних обчислювальних середовищах (хмарних платформах) в умовах постійно змінюваної обстановки на полі бою, організувати ефективне застосування різного роду інформаційних ресурсів (даних, інформації, процесів) оптимізуючи використання апаратного забезпечення, підвищити стійкість до несанкціонованим діям зловмисників та вірусів, забезпечити безпеку розгортання застосунків.

Серед різних програмних рішень (інструментів) найбільш розповсюдженими та популярними є інструменти, які розроблені з відкритим вихідним кодом, Kubernetes та Docker, опис яких є у вільному доступі та які постійно удосконалюються міжнародною IT-спільнотою.

У свою чергу, запровадження технології керування контейнерами (оркестрування) під час побудови інформаційної інфраструктури Міноборони дозволить підвищити ефективність запровадження сучасних інформаційних технологій у сфері оборони та здійснити наступні кроки з цифровізації Міноборони.

Подальші дослідження доцільно зосередити на дослідженні розгортання технології керування контейнерами (оркестрування) під час побудови гібридної хмари на базі розгорнутих у Міноборони центрів обробки даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Про хмарні послуги : Закон України від 17.02.2022 р. № 2075-IX. Дата оновлення: 05.06.2024. URL: https://zakon.rada.gov.ua/laws/show/2075-20#Text (дата звернення: 25.09.2024).
2. Деякі питання забезпечення функціонування інформаційно-комунікаційних систем, електронних комунікаційних систем, публічних електронних реєстрів в умовах воєнного стану : постанова Кабінету Міністрів України від 12.03.2022 р. № 263. Дата оновлення: 09.05.2023. URL: https://zakon.rada.gov.ua/laws/show/263-2022-%D0%BF#Text (дата звернення: 25.09.2024).
3. Порядок функціонування єдиної інформаційної системи Міністерства оборони України : наказ Міністерства оборони України від 21.03.2024 р. № 188. Дата оновлення: 17.04.2024. URL: https://www.mil.gov.ua/ministry/normativno-pravova-baza/nakazi-ministra-oboroni-ukraini/nakazi-ministerstva-oboroni-ukraini-za-2024-rik.html (дата звернення: 25.09.2024).
4. Андрощук О. В., Голобородько М. Ю., Головченко О. В., Миронюк А. Б. Теоретичні особливості використання центрів обробки даних в приватних хмарах: вимоги побудови, види, переваги та недоліки, надійність // Молодий вчений. 2021. № 7 (95). С. 1–4. DOI: https://doi.org/10.32839/2304-5809/2021-7-95-1 (дата звернення: 05.09.2024).
5. Головченко О.В. Аналіз сучасних підходів до створення центру обробки даних // Молодий вчений. 2020. № 4 (80). С. 221–227. DOI: https://doi.org/10.32839/2304-5809/2020-4-80-47 (дата звернення: 05.09.2024).

6. Андрощук О.В., Черевко Р.М., Петрушен М.В., Голобородько М. Ю. Актуальні підходи до побудови інформаційної інфраструктури на основі хмарних технологій з використанням референсної архітектури // Сучасні інформаційні технології у сфері безпеки та оборони. 2023. № 1 (46). DOI: https://doi.org/10.33099/2311-7249/2023-46-1-89-94 (дата звернення: 05.09.2024).
7. C-M(2015)0041-REV1, ALLIANCE C3 POLICY, 25 April 2016. North Atlantic Council, 50 p.
8. ACT/DIR/DIV/TT-6492/SER:NU:1394, C3 TAXONOMY BASELINE 6, 9 December 2022. HQ Supreme Allied Commander Transformation, 275 p.
9. Ліпко І. О., Звір В. Б., Миколенко Ю. М. Модель досягнення взаємосумісності комунікаційних та інформаційних систем: запровадження досвіду НАТО в інтересах сил оборони держави // Збірник наукових праць Центру воєнно-стратегічних досліджень Національного університету оборони України. 2023. № 2 (78). С. 108–120.
10. Ліпко І., Звір В., Капілевич В., Сугак С. Запровадження класифікації інформаційних технологій у сфері оборони: досвід НАТО для України // Збірник наукових праць Центру воєнно-стратегічних досліджень Національного університету оборони України. 2024. № 2 (81). С. 74–85.
11. DOD COMMAND, CONTROL, AND COMMUNICATIONS (C3) MODERNIZATION STRATEGY, September 2020, 40 p.
12. Fogli M. et al. Performance Evaluation of Kubernetes Distributions (K8s, K3s, KubeEdge) in an Adaptive and Federated Cloud Infrastructure for Disadvantaged Tactical Networks. *2021 International Conference on Military Communication and Information Systems (ICMCIS)*. The Hague, Netherlands, 04-05 May 2021, P. 1–7. DOI: 10.1109/ICMCIS52405.2021.9486396 (дата звернення: 05.09.2024).
13. Andersen H. et al. NATO Federated Coalition Cloud with Kubernetes: A National Prototype Perspective. *26th International Command and Control Research and Technology Symposium (ICCRTS)*. October 18-22 and 25-28. 2021. Washington, DC, 2021. URL: https://www.researchgate.net/publication/355511979_NATO_Federated_Coalition_Cloud_with_Kubernetes_A_National_Prototype_Perspective (дата звернення: 05.09.2024).
14. Pingen G., van der Geest J., Beaujon M., Voogd J., and Pieneman R. Data Centric C2-Services Deployment: an Experiment on Fleets of Military Vehicles // EasyChair Preprint. 2021. № 6312. С. 3–5. URL: https://easychair.org/publications/preprint/5kcS (дата звернення: 05.09.2024).
15. NATO Selects Thales to Supply Its First Defence Cloud for the Armed Forces : офіційний веб-сайт компанії Thales Group, January 25, 2021. URL: https://www.thalesgroup.com/en/group/journalist/press-release/nato-selects-thales-supply-its-first-defence-cloud-armed-forces (дата звернення: 05.09.2024).
16. Pentagon splits $9 billion cloud contract among Google, Amazon, Oracle and Microsoft : веб-сайт агентства Reuters, December 8, 2022. URL: https://www.reuters.com/technology/pentagon-awards-9-bln-cloud-contracts-each-google-amazon-oracle-microsoft-2022-12-07/ (дата звернення: 05.09.2024).
17. Manage data in Docker. URL: https://docs.docker.com/storage (дата звернення: 05.09.2024).
18. Kubernetes Components. URL: https://kubernetes.io/docs/concepts/overview/components (дата звернення: 05.09.2024).

Стаття надійшла до редакційної колегії 09.09.2024

Analysis of the features of the use of container management technology in the construction of the information infrastructure of the Ministry of Defense of Ukraine

Annotation

In the context of large-scale armed aggression directed against Ukraine, ensuring the resilience of national computing resources, through which critically important IT services are provided for both civilian and military needs, becomes of significant importance. Generally, such resilience can be characterized by two factors: the ability to withstand external attacks or internal failures and remain capable of providing the necessary IT services in support of military operations. This can be achieved through the rapid deployment of applications that provide the necessary IT services in a cloud environment using such a virtualization technology component as containerization.

The purpose of the article is to determine the features of using application management technologies in *Kubernetes* containers and container management using *Docker* software solutions, and to provide substantiated proposals based on the analysis results for their possible use in the interests of Ukraine's defense forces.

The use of container management (orchestration) approaches in the defense sector involves combining the areas of digitalization project development and direct operational activities. This allows customers and digitalization project teams to improve collaboration and ensure timely planning, creation, continuous integration, and continuous delivery of critically needed software solutions. In this case, integration is achieved by combining software code in a centralized repository, and continuous delivery is aimed at the automated deployment of new versions of the software solution into their operating environment.

Keywords: orchestration; containerization; aggregated technical service; application management technologies in containers; cluster environment.