



Силабус освітнього компонента

Програма навчальної дисципліни

Основи DevOps та хмарні технології



Шифр та назва спеціальності

F4 – Системний аналіз та наука про дані

Спеціалізація

-

Освітня програма

Системний аналіз і управління

Рівень освіти

Бакалавр

Семестр

6

Інститут

Навчально-науковий інститут комп'ютерних наук та інформаційних технологій

Кафедра

Кафедра системного аналізу та інформаційно-аналітичних технологій

Тип дисципліни

Вибіркова

Форма навчання

Денна

Мова викладання

Українська

Викладачі, розробники



Соболь Максим Олегович

Maksym.Sobol@khnpi.edu.ua

Кандидат технічних наук, доцент, доцент кафедри

Досвід роботи – понад 20 років. Автор понад 60 наукових та навчально-методичних праць. Провідний лектор з дисциплін: «Інженерія програмного забезпечення», «Якість та тестування програмного забезпечення».

[Детальніше про викладача на сайті кафедри](#)

Загальна інформація

Анотація

Навчальна дисципліна «Основи DevOps та хмарні технології» зосереджена на вивченні сучасних практик розробки, розгортання та експлуатації програмного забезпечення. Курс охоплює основні принципи DevOps культури, автоматизацію процесів CI/CD, контейнеризацію, оркестрацію, Infrastructure as Code, моніторинг та спостережуваність систем. Особлива увага приділяється хмарним платформам (AWS, Azure, Google Cloud), їх сервісам та архітектурним патернам для побудови масштабованих, надійних та економічно ефективних рішень.

Мета та цілі дисципліни

Метою навчальної дисципліни «Основи DevOps та хмарні технології» є формування у студентів комплексного розуміння сучасних практик DevOps та хмарних обчислень, необхідних для ефективної розробки, розгортання та підтримки програмних систем. Основні завдання дисципліни передбачають опанування принципів DevOps культури та автоматизації, вивчення систем контейнеризації (Docker) та оркестрації (Kubernetes), оволодіння методами Infrastructure

as Code (Terraform, Ansible), освоєння CI/CD пайплайнів (GitHub Actions, GitLab CI), вивчення хмарних платформ та їх сервісів (обчислення, зберігання, мережі, бази даних), а також опанування методів моніторингу, логування та забезпечення безпеки в хмарних середовищах.

Формат занять

Лекції, лабораторні заняття, самостійна робота, консультації. Підсумковий контроль – залік.

Компетентності

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Здатність планувати і управляти часом.

ЗК4. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК9. Здатність до адаптації та дії в новій ситуації.

ЗК10. Здатність працювати автономно.

ЗК12. Здатність працювати в команді.

ЗК13. Здатність працювати в міжнародному контексті.

ЗК14. Здатність оцінювати та забезпечувати якість виконуваних робіт.

СК6. Здатність до комп'ютерної реалізації математичних моделей реальних систем і процесів; проектувати, застосовувати і супроводжувати програмні засоби моделювання, прийняття рішень, оптимізації, обробки інформації, інтелектуального аналізу даних.

СК7. Здатність використовувати сучасні інформаційні технології для комп'ютерної реалізації математичних моделей та прогнозування поведінки конкретних систем а саме: об'єктно-орієнтований підхід при проектуванні складних систем різної природи, прикладні математичні пакети, застосування баз даних і знань.

СК8. Здатність організовувати роботу з аналізу та проектування складних систем, створення відповідних інформаційних технологій та програмного забезпечення.

СК10. Здатність розробляти експериментальні та спостережувальні дослідження і аналізувати дані, отримані в них.

Результати навчання

РН8. Володіти сучасними методами розробки програм і програмних комплексів та прийняття оптимальних рішень щодо складу програмного забезпечення, алгоритмів процедур і операцій.

РН9. Вміти створювати ефективні алгоритми для обчислювальних задач системного аналізу та систем підтримки прийняття рішень.

РН10. Знати архітектуру сучасних обчислювальних систем і комп'ютерних мереж.

РН11. Знати і вміти застосовувати на практиці системи управління базами даних і знань та інформаційні системи.

РН12. Застосовувати методи і засоби роботи з даними і знаннями, методи математичного, логіко-семантичного, об'єктного та імітаційного моделювання, технології системного і статистичного аналізу.

РН13. Проектувати, реалізовувати, тестувати, впроваджувати, супроводжувати, експлуатувати програмні засоби роботи з даними і знаннями в комп'ютерних системах і мережах.

Обсяг дисципліни

Загальний обсяг дисципліни 120 год. (4 кредити ECTS): лекції – 32 год., лабораторні роботи – 16 год., самостійна робота – 72 год.

Передумови вивчення дисципліни (пререквізити)

СП 3. Алгоритмізація та програмування – базові навички програмування для автоматизації та написання скриптів.

СП 6. Об'єктно-орієнтоване програмування – розуміння принципів розробки застосунків для їх контейнеризації та розгортання.

СП 4. Архітектура комп'ютерних систем та мереж – знання основ мережевих технологій та архітектури систем для роботи з хмарними платформами.

СП 11. Організація баз даних – розуміння принципів роботи БД для їх розгортання та управління в хмарі.

Особливості дисципліни, методи та технології навчання

Дисципліна «Основи DevOps та хмарні технології» поєднує теоретичну підготовку (лекції) та практичну діяльність (лабораторні заняття, індивідуальні завдання, самостійна робота), що спрямовані на розвиток як професійних, так і загальних компетентностей студентів.

Особливості дисципліни:

- практична орієнтованість – основний акцент зроблено на відпрацювання навичок роботи з реальними DevOps інструментами та хмарними платформами;
- інтеграція з реальними процесами – завдання наближені до практики побудови промислових CI/CD пайплайнів та хмарної інфраструктури;
- використання індустріальних інструментів (Git, Docker, Kubernetes, Terraform, GitHub Actions, AWS/Azure/GCP), що забезпечує можливість відпрацювання навичок з технологіями, затребуваними ринком;
- поєднання індивідуальної та проектної роботи – виконання лабораторних завдань і розробка власного проєкту з розгортання застосунку в хмарі;

Методи навчання:

- лекції з мультимедійним супроводом для представлення ключових концепцій DevOps та хмарних архітектур;
- лабораторні заняття з виконанням практичних завдань – від базової контейнеризації до повного CI/CD пайплайну з розгортанням у хмарі;
- самостійна робота студентів, що включає опрацювання технічної документації, додаткових джерел та виконання практичних завдань;
- проектно-орієнтоване навчання – виконання підсумкового проєкту з розгортання мікросервісного застосунку в хмарі;
- дискусії та обговорення кейсів з практики DevOps інженерії та хмарних міграцій;

Технології навчання:

- використання систем контролю версій (Git, GitHub, GitLab) для організації командної роботи та CI/CD.
- робота з контейнерами (Docker, Docker Compose) та оркестрацією (Kubernetes, K3s).
- застосування Infrastructure as Code (Terraform, Ansible, CloudFormation).
- налаштування CI/CD пайплайнів (GitHub Actions, GitLab CI/CD, Jenkins).
- використання хмарних платформ (AWS, Azure, Google Cloud Platform) та їх сервісів.
- інструменти моніторингу та спостережуваності (Prometheus, Grafana, ELK Stack, CloudWatch).

Програма навчальної дисципліни

Навчальні заняття

Лекції

Теми лекцій	Кількість годин
Тема 1. Вступ до DevOps: культура, практики, інструменти. Еволюція від традиційної розробки до DevOps. Принципи DevOps культури. Огляд інструментального екосистеми.	4

Тема 2. Системи контролю версій та колаборація. Git workflows. Branching strategies. Code review. GitOps підходи.	4
Тема 3. Контейнеризація та Docker. Основи контейнерів. Docker архітектура. Dockerfile. Docker Compose. Best practices.	6
Тема 4. CI/CD: безперервна інтеграція та доставка. Принципи CI/CD. GitHub Actions, GitLab CI. Стратегії deployment. Automated testing.	6
Тема 5. Оркестрація контейнерів: Kubernetes. Архітектура Kubernetes. Pods, Services, Deployments. Scaling та self-healing.	4
Тема 6. Infrastructure as Code (IaC). Terraform основи. Declarative vs Imperative. State management. Модулі та best practices.	4
Тема 7. Хмарні платформи: AWS, Azure, Google Cloud. Огляд хмарних провайдерів. Compute, Storage, Networking сервіси. Serverless архітектури.	6
Тема 8. Моніторинг, логування та спостережуваність. Metrics, Logs, Traces. Prometheus, Grafana. Alerting. Cost monitoring.	4
Тема 9. Безпека в DevOps (DevSecOps). Security as Code. Secrets management. Vulnerability scanning. Compliance.	2
Тема 10. Масштабування та надійність систем. High Availability. Load balancing. Auto-scaling. Disaster recovery.	2
Загальна кількість годин	32

Лабораторні заняття

Теми лабораторних занять	Кількість годин	Вагові коефіцієнти <i>a</i>
Тема 1. Git та GitHub workflows. Налаштування Git. Основні операції. Pull requests та code review процес.	2	1
Тема 2. Контейнеризація застосунку з Docker. Створення Dockerfile. Multi-stage builds. Docker Compose для мікросервісів.	3	1
Тема 3. Побудова CI/CD пайплайну. GitHub Actions workflow. Automated tests. Build та push Docker images.	3	1
Тема 4. Розгортання в Kubernetes. Створення Kubernetes manifests. Deployment стратегії. Services та Ingress.	3	1
Тема 5. Infrastructure as Code з Terraform. Створення хмарної інфраструктури. VPC, Compute instances. State management.	3	1
Тема 6. Моніторинг системи (Prometheus + Grafana). Налаштування метрик. Створення dashboards. Alerting rules.	2	1
Загальна кількість годин	16	6

Самостійна робота

До самостійної роботи відноситься самостійне опрацювання теоретичного матеріалу та виконання індивідуального завдання.

Опрацювання теоретичного матеріалу

Кількість годин

Теми для самостійного вивчення

Тема 1. DevOps культура та практики. Вивчення DevOps manifesto. Agile та DevOps зв'язок. Дослідження кейсів успішних DevOps трансформацій.

6

Тема 2. Поглиблене вивчення Git. Advanced Git операції (rebase, cherry-pick, bisect). Git hooks. Monorepo vs multirepo підходи.	6
Тема 3. Docker екосистема. Docker networking. Volumes та storage. Security best practices. Docker registry alternatives.	7
Тема 4. CI/CD платформи та інструменти. Порівняння CI/CD рішень. Jenkins. GitLab CI. CircleCI. Testing strategies в pipeline.	7
Тема 5. Kubernetes поглиблено. StatefulSets, DaemonSets. ConfigMaps та Secrets. RBAC. Helm charts. Service mesh (Istio).	7
Тема 6. Terraform та IaC практики. Terraform modules. Workspaces. Remote state. TerraformCloud. Ansible альтернатива.	7
Тема 7. Хмарні архітектури. Well-Architected Framework. Serverless patterns. Multi-cloud стратегії. Cost optimization.	7
Тема 8. Observability та Site Reliability Engineering. SRE principles. SLI, SLO, SLA. Distributed tracing. ELK Stack. OpenTelemetry.	7
Тема 9. DevSecOps практики. SAST, DAST, SCA. Secrets scanning. Container security. Compliance as Code.	6
Тема 10. Chaos Engineering та Disaster Recovery. Chaos Monkey. Failure injection. Backup strategies. Multi-region deployments.	6
Загальна кількість годин	40

Тематика індивідуальних завдань

Розрахункове завдання

Тематика: Розрахункові завдання передбачають індивідуальне проектування та реалізацію повного життєвого циклу застосунку з використанням DevOps практик та хмарних технологій. Тематика завдань охоплює контейнеризацію застосунків, побудову CI/CD пайплайнів, розгортання в Kubernetes, створення інфраструктури через Terraform, налаштування моніторингу та логування, впровадження практик безпеки, а також оптимізацію витрат та забезпечення високої доступності. Кожен студент обирає тему відповідно до власного застосунку або сервісу, що дозволяє поєднати набуті знання з індивідуальними інтересами та майбутньою професійною діяльністю.

Обсяг: 15-20 сторінок пояснювальної записки, архітектурні діаграми, робочий CI/CD пайплайн, інфраструктурний код

Теми індивідуального завдання (приклад)

- Тема 1.** Розгортання веб-застосунку з CI/CD. Контейнеризація (Docker), GitHub Actions pipeline, deployment в Kubernetes, автоматичне тестування.
- Тема 2.** Infrastructure as Code проєкт. Terraform конфігурація для AWS/Azure/GCP, VPC, compute instances, RDS/CloudSQL, monitoring setup.
- Тема 3.** Мікросервісна архітектура в Kubernetes. Розробка 3-4 мікросервісів, service mesh, distributed tracing, centralized logging.
- Тема 4.** Serverless застосунок з CI/CD. AWS Lambda/Azure Functions, API Gateway, DynamoDB/CosmosDB, automated deployment.
- Тема 5.** Система моніторингу та alerting. Prometheus, Grafana, проектування dashboards, alert rules, integration з Slack/PagerDuty.
- Тема 6.** Multi-environment deployment strategy. Dev/Staging/Prod середовища, GitOps з ArgoCD/FluxCD, secrets management з Vault.
- Тема 7.** Cost optimization проєкт. Аналіз витрат хмарної інфраструктури, впровадження auto-scaling, spot instances, reserved capacity.
- Тема 8.** DevSecOps pipeline. Integration SAST/DAST tools, container scanning, secrets detection, compliance checking в CI/CD.

Неформальна освіта

Здобувач має можливість перезарахувати окремі теми або курс шляхом проходження професійних курсів чи тренінгів у сфері архітектури ПЗ.

Рекомендовані курси, тренінги, стажування

1. AWS Certified Cloud Practitioner Training – вступний курс від Amazon для розуміння основ хмарних обчислень та сервісів AWS. <https://aws.amazon.com/certification/certified-cloud-practitioner/>
2. Microsoft Azure Fundamentals (AZ-900) – базовий курс від Microsoft про концепції хмарних обчислень та сервіси Azure. <https://learn.microsoft.com/en-us/certifications/azure-fundamentals/>
3. Docker for Beginners (Docker.com) – офіційні навчальні матеріали від Docker Inc. про контейнеризацію застосунків. <https://www.docker.com/101-tutorial/>
4. Kubernetes Basics (CNCF) – інтерактивний курс від Cloud Native Computing Foundation про основи Kubernetes. <https://kubernetes.io/docs/tutorials/kubernetes-basics/>
5. Terraform Associate Certification (HashiCorp) – офіційний курс від HashiCorp про Infrastructure as Code з Terraform. <https://developer.hashicorp.com/terraform/tutorials>
6. GitHub Actions Learning Path (GitHub Learning Lab) – практичні вправи з побудови CI/CD workflows на GitHub Actions. <https://skills.github.com/>
7. Google Cloud Skills Boost – безкоштовні лабораторні роботи та курси з Google Cloud Platform. <https://www.cloudskillsboost.google/>
8. Site Reliability Engineering (Google) – книга та курс від Google про SRE практики. <https://sre.google/books/>

Література, навчальні матеріали та інформаційні ресурси

Список джерел інформації та матеріалів, оформлений згідно зі стандартом. Можна виділити розділи списку. Наприклад, «Основна література», «Додаткова література» тощо.

Основна література

1. Міністерство цифрової трансформації України. Посібник з DevOps практик для державних установ (2023). https://thedigital.gov.ua/storage/uploads/files/news_post/2023/11/Posibnyk_DevOps.pdf
2. Kubernetes Documentation. Production Best Practices Guide (2024). <https://kubernetes.io/docs/setup/best-practices/>
3. Microsoft Azure. DevOps Resource Center - Cloud Adoption Framework (2024). <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/>
4. HashiCorp. Terraform Best Practices Guide (2024). <https://developer.hashicorp.com/well-architected-framework/operational-excellence/operational-excellence-terraform-maturity>
5. CNCF. Cloud Native Security Whitepaper (2022). https://github.com/cncf/tag-security/blob/main/security-whitepaper/v2/CNCF_cloud-native-security-whitepaper-May2022-v2.pdf
6. AWS Well-Architected Framework Whitepaper (2023). <https://docs.aws.amazon.com/pdfs/wellarchitected/latest/framework/wellarchitected-framework.pdf>

Додаткова література

1. Docker Documentation. Official Docker docs and best practices. <https://docs.docker.com/>
2. Kubernetes Documentation. Official K8s documentation. <https://kubernetes.io/docs/>
3. Terraform Documentation. HashiCorp Terraform guides and reference. <https://developer.hashicorp.com/terraform/docs>
4. GitHub Actions Documentation. CI/CD workflows and marketplace actions. <https://docs.github.com/en/actions>

5. AWS Documentation. Comprehensive AWS services documentation. <https://docs.aws.amazon.com/>

Інформаційні ресурси

1. <https://github.com/dockerexamples> – офіційні приклади Docker застосунків для різних стеків технологій
2. <https://github.com/kubernetes/examples> – колекція прикладів Kubernetes manifests та patterns
3. <https://github.com/terraform-aws-modules> – community-maintained Terraform modules для AWS
4. <https://github.com/awesome-selfhosted/awesome-selfhosted> – список self-hosted сервісів з Docker підтримкою
5. <https://12factor.net/> – методологія побудови SaaS застосунків (обов'язкове read для DevOps)
6. <https://github.com/cncf/landscape> – Cloud Native landscape з усіма інструментами екосистеми
7. <https://roadmap.sh/devops> – інтерактивна roadmap для вивчення DevOps
8. <https://github.com/bregman-arie/devops-exercises> – велика колекція практичних завдань та питань DevOps

Система оцінювання

Підсумкова оцінка з освітнього компонента визначається відповідальним лектором за темами, видами занять, тощо у відповідності до силабусу і є інтегральною оцінкою результатів усіх вид навчальної діяльності здобувача вищої освіти. Підсумкова оцінка повинна відображати всі оцінки за складовими навчального процесу з урахуванням їх вагових показників k :

Поточний контроль (тести), k_1	Середньозважена оцінка за лабораторні роботи, k_2	Індивідуальне завдання (розрахункове завдання), k_3	Підсумковий контроль (залік), k_4
0,3	0,3	0,3	0,1

Сума коефіцієнтів повинна складати одиницю: $k_1 + k_2 + k_3 + k_4 = 1$. Підбір вагових коефіцієнтів підсумкової оцінки здійснює розробник курсу.

Розрахунок підсумкової оцінки проводиться за формулою:

$$O = П \cdot k_1 + K \cdot k_2 + I \cdot k_3 + Пк \cdot k_4$$

де: $П$ – середньозважена середня оцінка за поточний контроль
 I – оцінка за виконання індивідуального завдання
 K – середньозважена оцінка за лабораторні роботи
 $Пк$ – оцінка за підсумковий контроль (залік)

$$П = \frac{П_1 \cdot b_1 + П_2 \cdot b_2 + \dots + П_n \cdot b_n}{\sum_{i=1}^n b_i}$$

де: b_i - ваговий коефіцієнт за кожен тест.

$$K = \frac{K_1 \cdot a_1 + K_2 \cdot a_2 + \dots + K_m \cdot a_m}{\sum_{i=1}^m a_i}$$

де: a_i - ваговий коефіцієнт за кожне лабораторне заняття.

Поточні оцінки за кожен складову ($П, K, I, \dots$) виставляються за 100-бальною шкалою згідно з [положенням «Про критерії та систему оцінювання знань та вмінь і про рейтинг здобувачів вищої освіти» НТУ «ХПІ»](#).

Шкала оцінювання

Сума балів	Національна оцінка	ECTS
90–100	Відмінно	A
82–89	Добре	B
75–81	Добре	C
64–74	Задовільно	D

Підсумкова оцінка виставляється відповідно до розрахованої O з округленням до найближчого цілого числа в більшу сторону.

60–63	Задовільно	E
35–59	Незадовільно (потрібне додаткове вивчення)	FX
1–34	Незадовільно (потрібне повторне вивчення)	F

Норми академічної етики і політика курсу

Здобувач вищої освіти повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність. Конфліктні ситуації повинні відкрито обговорюватися в навчальних групах з викладачем, а при неможливості вирішення конфлікту – доводитися до відома співробітників дирекції інституту.

Нормативно-правове забезпечення впровадження принципів академічної доброчесності НТУ «ХПІ» розміщено на сайті: <http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/>

Погодження

Силабус погоджено

30.08.2025

Завідувач кафедри

Тетяна АЛЕКСАНДРОВА

30.08.2025

Гарант ОП

Юрій ДОРОФЄЄВ