Agenda kursu Docker Masters

📌 Kurs **Docker Masters** to rewolucyjny na skalę Polski program szkoleniowy, dzięki któremu znacząco zoptymalizujesz proces wytwarzania i wdrażania oprogramowania.

Szkolenie zostało podzielone na bardzo obszerną, **przekrojową część główną oraz cztery dodatkowe ścieżki tematyczne** dla Frontendowca, Backendowca, DevOpsa oraz inżyniera QA.

Jeżeli masz pytania dotyczące kursu - pisz śmiało: online@infoshareacademy.com

Podsumowanie kursu w pigułce

Kurs	Liczba lekcji	Czas trwania
Docker Masters - Ścieżka Zasadnicza	63	19 godzin
Docker Masters - Backend Path	39	12 godzin
Docker Masters - DevOps Path	15	4,5 godziny
Docker Masters - Frontend Path	8	3 godziny
Docker Masters - QA Path	7	2 godziny
SUMA:	**132**	**40,5 godziny**

Spis treści

Technologie omawiane w kursie

Docker Engine, Docker Desktop, Docker Client, BuildKit, BuildX, Docker Compose, Docker Hub, GitHub Packages, Google Container Registry, Harbor, Docker Registry, PHPStorm, WebStorm, Microsoft VS Code, Desktop Application in Docker, Ubuntu Desktop, X Server, noVNC, Sass, SCSS, Prettier, ESLint, NPM, WebPack, React.JS, Next.JS, Angular, React Native, Python, Ruby, Node.JS, GoLang, PHP, nginx, Apache, Tomcat, Memcached, MySQL, PostgreSQL, MariaDB, MongoDB, ElasticSearch, Apache Solr, Redis, RabbitMQ, Kafka, HaProxy, Varnish Cache, ZSH, AppArmor, SELinux, Aqua Microscanner, Anchore, Clair, Trivy, Docker Swarm, Kubernetes, Debian Linux, Fedora, Ubuntu Linux, CentOS, Alpine Linux, PowerShell, Prometheus, Grafana, Kibana, AlertManager, GitLab, GitLab Runner, GitHub Actions, Jenkins, Selenium, Chrome Headless, Firefox Headless, PlantUML

Szczegółowa Agenda

▼ Docker Masters - Ścieżka Zasadnicza

1. Wstęp
2. Struktura szkolenia oraz wykorzystane rozwiązania i technologie
3. Czym jest Docker, Docker Engine oraz Docker Client?
4. Wirtualizacja, emulacja i konteneryzacja
5. Powody powstania Dockera oraz zalety konteneryzacji
6. Uruchamianie rozwiązań Linuxowych na systemach Windows i Mac
7. System operacyjny oraz kernel hosta z perspektywy Dockera
8. Instalacja Docker Engine na systemie Linux

9. Instalacja Docker Desktop na systemach Windows i Mac
10. Konfiguracja Docker Engine
11. Dostęp do plików i logów Docker Engine
12. Sprawdzanie zużycia zasobów i dysku twardego przez Docker Engine
13. Czyszczenie stanu Docker Engine
14. Wykrywanie i analizowanie problemów z Docker Engine
15. Różnice w pracy z Dockerem na systemach Linux, Windows i Mac
16. Uruchomienie pierwszego kontenera Docker
17. Cykl życia kontenera
18. Uzyskiwanie lokalnej kopii publicznego obrazu Dockera
19. Container name
20. Uruchamianie wielu kontenerów jednocześnie
21. Komunikacja pomiędzy kontenerami
22. Wykorzystanie klienta Dockera do monitorowania kontenerów
23. Uruchamianie kontenerów w trybie daemona
24. Główny proces działający w kontenerze (PID 1) i obsługa sygnałów
25. Restartowanie oraz przyłączenie się do kontenera
26. Uzyskiwanie dostępu do wnętrza kontenera i uruchamianie komend
27. Bezpieczne zatrzymywanie kontenera (graceful shutdown)
28. Zabijanie kontenera i jego procesów (forceful shutdown)
29. Jedna technologia - 100 języków programowania - Docker [Masterclass]
30. Budowanie własnego obrazu kontenera - wstęp
31. Różnice pomiędzy kontenerem i obrazem
32. Budowanie obrazu w trybie interaktywnym
33. Budowanie obrazu z pliku Dockerfile
34. Struktura pliku Dockerfile i dostępne instrukcje
35. Kopiowanie lokalnych plików projektu do obrazu Dockerowego
36. Strategia budowanie obrazu "multi-stage”
37. Różnice i możliwości wykorzystania BuildKit oraz BuildX do budowania obrazu i mogącego działać na kilku architekturach

38. Dobre praktyki pisania plików Dockerfile
39. Docker Volumes - wstęp
40. Udostępnianie lokalnego katalogu i plików do kontenera
41. Docker Networks - wstęp
42. Podstawy budowy i działania sieci pomiędzy kontenerami a hostem
43. Dostępne sterowniki sieci kontenerów
44. Zapewnianie komunikacji pomiędzy większą liczbą kontenerów (service discovery)
45. Tryb sieciowy "host”
46. Docker Compose - wstęp
47. Struktura i części składowe pliku docker-compose.yml
48. Definiowanie całego stosu aplikacji w pliku Docker Compose
49. Definicje serwisów za pomocą zmiennych środowiskowych i pliku .env
50. Zaawansowane możliwości pliku docker-compose.yml [Masterclass]
51. Przechowywanie obrazów Dockera - wstęp
52. Zapisywanie całości obrazu do pojedynczego pliku na dysku twardym
53. Czym jest rejestr obrazów
54. GitHub Packages - masterclass
55. Uruchamianie lokalnego rejestru obrazów
56. Konfiguracja i zabezpieczanie dostępu do rejestru obrazów
57. Uruchomienie lokalnego mirrora rejestru Docker Hub [Masterclass]
58. Komunikacja pomiędzy klientem Dockera a Container Registry [Masterclass]
59. Budowa obrazu od wewnątrz - manifest obrazu oraz jego warstwy [Masterclass]
60. Modyfikowanie obrazu bez przebudowywania za pomocą "docker build" [Masterclass]
61. Narzędzia pomocnicze i autouzupełnianie komend Dockera [Masterclass]
62. Integracja Dockera z Microsoft VS Code [Masterclass]

63. Aplikacje desktopowe w kontenerze - X Server, Tor browser, DOSBox, Python GUI (PyGObject), Wireshark, PHPStorm, Firefox, Wine - [LIVE Masterclass]

▼ Docker Masters - Backend Path

1. Docker dla programistów backend - wstęp
2. Development, budowanie, testowanie i uruchamianie aplikacji (Python)
3. Development, budowanie, testowanie i uruchamianie aplikacji (Ruby)
4. Development, budowanie, testowanie i uruchamianie aplikacji (Node.JS)
5. Development, budowanie, testowanie i uruchamianie aplikacji (GoLang)
6. Development, budowanie, testowanie i uruchamianie aplikacji (PHP)
7. Uruchamianie całego stosu aplikacji w Dockerze
8. Zapisywanie danych tymczasowych z wykorzystaniem Memcached
9. Uruchamianie i praca z relacyjną bazą danych (MySQL)
10. Uruchamianie i praca z relacyjną bazą danych (PostgreSQL)
11. Uruchamianie i praca ze skonteneryzowaną relacyjną bazą danych (MariaDB)
12. Uruchamianie i praca ze skonteneryzowaną bazą danych dokumentów (MongoDB)
13. Uruchamianie i praca z bazą danych dokumentów (ElasticSearch)
14. Uruchamianie i praca ze skonteneryzowaną bazą danych dokumentów (Apache Solr)
15. Uruchamianie i praca z bazą danych NoSQL (Redis)
16. Uruchamianie i praca ze skonteneryzowanym brokerem wiadomości (RabbitMQ)
17. Kwestie wydajnościowe i bezpieczeństwa
18. Czym jest reverse-proxy, load-balancer i jego zadania
19. Uruchamianie i praca z load-balancerem (nginx)

20. Cachowanie odpowiedzi HTTP (Varnish Cache)
21. Uzyskiwanie dostępu do powłoki kontenera
22. Testowanie aplikacji w oparciu o różne technologie
23. Bezpieczeństwo kontenerów - wstęp
24. Zarządzanie zużyciem zasobów (CPU, RAM) kontenera
25. Definiowanie zasady automatycznego uruchomienia kontenera
26. Uruchamianie kontenera z systemem plików tylko do odczytu
27. Uruchamianie kontenera w trybie "privileged”
28. Uruchamianie kontenera w trybie sieciowym "host”
29. Zabezpieczanie kontenerów z pomocą AppArmora i SELinux
30. Uruchamianie aplikacji z dedykowanego użytkownika i grupy
31. Nadmiarowe pliki pozostałe w obrazie - konsekwencje i wykrywanie
32. Podpisywanie obrazów a ataki man-in-the-middle [Masterclass]
33. Skanowanie bezpieczeństwa obrazu z Anchore
34. Skanowanie bezpieczeństwa obrazu z Trivy i Aqua Microscanner [Masterclass]
35. Orkiestracja - wstęp
36. Uruchamianie lokalnego klastra Docker Swarm i wykorzystanie Play-with-Docker
37. Wydanie skonteneryzowanego stosu aplikacji na klaster Docker Swarm
38. Uruchomienie lokalnego klastra Kubernetes
39. Konwersja pliku Docker Compose na definicje YAML Kubernetesa

▼ Docker Masters - Ścieżka DevOps

1. Docker dla DevOpsa - wstęp
2. Praca z różnymi systemami operacyjnymi (Debian, Fedora, Ubuntu, CentOS, AlpineLinux)
3. Uruchamianie i praca z różnymi powłokami w kontenerze (Sh, Bash, Zsh, PowerShell)
4. Praca z narzędziami powłoki - manipulacja tekstem

5. Praca z narzędziami powłoki - monitorowanie i debugowanie procesów
6. Praca z narzędziami powłoki - monitorowanie i debugowanie sieci
7. Różne koncepcje systemów operacyjnych (zarządzanie procesami)
8. Różne koncepcje systemów operacyjnych (zarządzanie I/O)
9. Monitorowanie i debugowanie systemu operacyjnego hosta
10. Metryki definiujące klaster, aplikacje działające w kontenerach oraz Docker Engine
11. Zbieranie metryk za pomocą oprogramowania Prometheus
12. Korzyści z Dockera w procesie CI/CD
13. Konfigurowanie procesu CI w celu statycznej analizy, testowania i skanowania bezpieczeństwa CI/CD (GitLab.com)
14. Docker w procesie CI/CD (lokalna instancja GitLab oraz GitLab Runnera i Jenkins)
15. Integracja Docker Engine z AppArmor

▼ Docker Masters - Ścieżka Frontend

1. Docker dla programistów frontend - wstęp
2. Konteneryzacja statycznego projektu składającego się z plików HTML, CSS, JavaScript oraz plików graficznych
3. Praca z rozszerzonymi plikami CSS - Sass i SCSS
4. Statyczna analiza i formatowanie plików CSS z użyciem kontenerów - Prettier i ESLint
5. Uruchamianie projektów Node.JS w Dockerze
6. Konteneryzacja projektu, uzyskiwanie dostępu do technologii i budowanie obrazu aplikacji (React.JS)
7. Konteneryzacja projektu, uzyskiwanie dostępu do technologii i budowanie obrazu aplikacji (Angular)
8. Uruchamianie przeglądarek Chrome i Firefox wewnątrz kontenera [Masterclass]

▼ Docker Masters - Ścieżka QA

1. Docker w środowisku Quality Assurance (QA) - wstęp
2. Jakie są korzyści Dockera podczas kontroli jakości aplikacji?
3. Selenium w Dockerze - wstęp
4. Uruchamianie lokalnego klastra Selenium w Dockerze i wykorzystanie przeglądarek w trybie headless
5. Zapewnienie komunikacji pomiędzy skonteneryzową aplikacją a klastrem Selenium i uruchomienie testów w języku Python
6. Stworzenie i uruchomienie testów Selenium w języku Node.JS
7. Praca z diagramami UML i uruchomienie PlantUML Server