Program Akdemii DevOps

PODSTAWY LINUX

ARCHITEKTURA SYSTEMU LINUX

- Hierarchia katalogów Linuxa (FHS, Filesystem Hierarchy Standard)
- Jądro systemu Linux
- Proces uruchomienia systemu i jego nadzorowanie
- Opcje uruchamiania systemu
- Restart i wyłączenie systemu

INSTALACJA SYSTEMU LINUX

- Partycjonowanie dysku
- Tworzenie systemu plików
- Konfiguracja bootloadera
- Instalacja i usuwanie oprogramowania

PRACA W KONSOLI

- Działanie powłoki i wydawanie poleceń
- Konfiguracja powłoki
- Korzystanie z historii
- Uzyskiwanie pomocy i korzystanie z dokumentacji
- Przeglądanie katalogów i plików
- Katalog domowy
- Poruszanie się w systemie plików
- Tworzenie, kopiowanie i usuwanie plików
- Uprawnienia do plików, właściciele plików
- Pliki i atrybuty specjalne
- Tworzenie i działanie łączy twardych i miękkich (symbolicznych)
- Strumienie standardowe, przekierowania i potoki
- Tworzenie, monitorowanie i wysyłanie sygnałów do procesów
- Priorytety procesów

NARZĘDZIA DOSTĘPNE W SYSTEMIE

- Kompresja i archiwizacja plików
- Wyrażenia regularne
- Wyszukiwanie i przeszukiwanie zawartości plików
- Edycja plików
- Przetwarzanie danych tekstowych
- Wyszukiwanie użytecznych narzędzi

TWORZENIE I URUCHAMIANIE SKRYPTÓW

- Uruchamianie interpretera skryptów, znak shebang
- Wykorzystanie zmiennych
- Komunikacja z użytkownikiem
- Instrukcja warunkowa
- Pętle

KONFIGURACJA I ADMINISTRACJA SYSTEMEM

- Konfiguracja konta użytkownika
- Zarządzanie uprawnieniami użytkownika
- Zmiana hasła
- Konto i uprawnienia administratora
- Konfiguracja systemu, katalog etc
- Tworzenie i usuwanie użytkowników oraz grup
- Uruchamianie i zatrzymywanie usług
- Przeglądanie i interpretacja dzienników systemu
- Przeglądanie i zarządzanie logami aplikacji
- Współpraca systemu ze sprzętem komputerowym
- Monitorowanie i diagnostyka działania urządzeń
- Pamięć operacyjna, obszar swap
- Wykorzystanie pamięci masowej

USTAWIENIA I USŁUGI SYSTEMOWE

- Tworzenie i konfiguracja usług
- Planowanie uruchamiania zadań
- Obsługa czasu systemowego
- Lokalizacja i ustawienia międzynarodowe

LINUX W SIECI

- Działanie i dostęp do sieci komputerowej
- Konfiguracja sieci, IPv4, IPv6Najważniejsze protokoły i narzędzia sieciowe
- Konfiguracja klienta DNS (Domain Name Server)Bezpieczeństwo usług sieciowych
- Monitorowanie działania sieci
- Diagnostyka i rozwiązywanie typowych problemów

PRACA ZDALNA

- Dostęp do systemu za pomocą haseł i kluczy
- Bezpiecznie kopiowanie plików

WPROWADZENIE DO DEVOPS ORAZ CI/CD

CZYM JEST DEVOPS?

- Pipeline DevOps
- Narzędzia DevOps
- SRE jako ulepszona wersja DevOps
- Automatyzacja i narzędzia CI/CD na przykładzie Jenkins
- Zarządzanie wieloma środowiskami np, test, prod, qa

ZARZĄDZANIE ORAZ KONFIGURACJA MASZYN WIRTUALNYCH ZA POMOCĄ ANSIBLE

- Czym jest Ansible?
- Automatyzacja Pull vs Push
- Koncepty: idempotentność, moduły, host inventory, task'i, polecenia ad-hoc
- Podstawy Ansible'a: Playbook'i, YAML, moduły, zmienne, role

SYSTEM KONTROLI WERSJI GIT

WPROWADZENIE

- Jak działają systemy kontroli wersji
- Charakterystyka pracy z rozproszonym systemem kontroli wersji
- Instalacja i podstawowa konfiguracja Gita
- Narzędzia dostarczane z Gitem

LOKALNA PRACA Z GITEM

- Inicjalizacja i konfiguracja repozytorium
- Katalog .git
- Zatwierdzanie zmian (commit), narzędzie Git GUI
- Dobre praktyki tworzenia opisów zmian
- Przechowywanie zmian (stash)
- Ignorowanie i modyfikacja zmian
- Przeglądanie zmian (diff), narzędzia do przeglądania zmian
- Przegląd historii (log), narzędzie gitk
- Tworzenie etykiet, rodzaje etykiet (tag)
- Cofanie zmian (reset, reflog)
- Praca z gałęziami (branch)
- Scalanie zmian i zmiana bazy (merge, rebase, cherry pick)
- Rozwiązywanie konfliktów
- Tworzenie aliasów

ZDALNE REPOZYTORIA

- Metody dostępu do repozytoriów, używane protokoły (SSH, HTTP)
- Pobieranie repozytorium (clone)
- Wypychanie i pobieranie zmian (push, fetch, pull)
- Powiadamianie o zmianach (pull request)
- Praca ze zdalnymi gałęziami
- Praca z podprojektami (submodule, subtree)

PRACA GRUPOWA Z GIT

- Metody współdzielenia zmian
- Praca w środowisku użytkowników o różnych rolach (programista, tester, wdrożeniowiec)
- Zasady zarządzania wydaniami i historią projektu
- Rozszerzenie GitFlow

KONTENERYZACJA NA PRZYKŁADZIE DOCKER'A

WPROWADZENIE DO KONTENERYZACJI I JEJ PODSTAWY TECHNOLOGICZNE

- Zalety i ograniczenia rozwiązań kontenerowych
- Architektura i sposób działania Docker'a
- Podstawy pracy z Docker CLI
- Budowanie obrazów z użyciem Dockerfile
- Konfiguracja sieci
- Stan kontenera oraz praca z wolumenami
- Przydzielanie i kontrolowanie wykorzystywanych zasobów
- Administracja kontenerami - gromadzenie logów, metryki wydajności, polityki restartowania aplikacji
- Deklaratywne zarządzanie usługami w oparciu o Docker Compose
- Repozytoria i dystrybucja obrazów
- Bezpieczeństwo
- Orkiestracja usług z Docker Swarm
- Alternatywy dla Dockera na przykładzie;: rkt, lxd, podman

KUBERNETES

ARCHITEKTURA KUBERNETES

- Komponenty klastra (masters oraz workers)
- Zarządzanie obiektami Kubernetes (imperatywne oraz deklaratywne)
- Manifesty obiektów - struktura oraz format (YAML)
- Docker w orkiestratorze Kubernetes

PODSTAWY KONFIGURACJI

- Minikube jako najłatwiejszy sposób na instalację lokalnego klastra
- Dashboard czyli podgląd stanu klastra w przeglądarce internetowej
- Podział klastra na wirtualne przestrzenie (Namespaces)
- Pod jako podstawowa jednostka alokacji procesów w klastrze
- Labels and Selectors - grupowanie obiektów za pomocą etykiet
- Services - konfiguracja komunikacji pomiędzy aplikacjami w klastrze jak i dostęp do aplikacji ze świata zewnętrznego
- Zarządzanie liczbą replik aplikacji oraz sposobem ich aktualizacji za pomocą Deployments
- Jobs/CronJobs czyli sposób na uruchamianie zadań skryptowych w klastrze Kubernetes
- Konfiguracja centralnego monitoringu oraz logowania przy użyciu DaemonSets
- StatefulSets jako sposób na uruchamianie aplikacji stanowych w klastrze

UWIERZYTELNIANIE ORAZ AUTORYZACJA

- Organizacja informacji o klastrach i użytkownikach w pliku kubeconfig
- Przedstawienie typów użytkowników w Kubernetes: używanych przez administratorów (użytkownicy) oraz aplikacje (konta serwisowe)
- Omówienie strategii uwierzytelniania w klastrze, od użytkownika z hasłem, przez certyfikaty x509, po tokeny OpenID
- Kontrola dostępu oparta na rolach jako sposób na przypisywanie uprawnień do użytkowników
- Walidacja lub modyfikacja żądań za pomocą Admission Controllers
- Sieci
- Porównanie architektury sieciowej: Docker vs. Kubernetes
- CNI jako interfejs służący do konfiguracji kart sieciowych kontenerów
- Omówienie różnych sposobów na publikacje aplikacji za pomocą Services (ClusterIP, NodePort, LoadBalancer, ExternalIP, ExternalName)
- Ingress czyli przekierowanie przychodzących zapytań HTTP do aplikacji uruchomionych w Kubernetes (na przykładzie kontrolera nginx)
- Blokada komunikacji sieciowej w klastrze za pomocą Network Policies
- Konfiguracja serwera rozwiązywania nazw w klastrze* Storage
- Zarządzanie zmiennymi środowiskowymi i plikami konfiguracyjnymi aplikacji z użyciem ConfigMaps
- Przechowywanie danych wrażliwych takich jak: hasła, klucze czy tokeny za pomocą Secrets (generic, docker-registry, tls)
- Utrwalanie danych z użyciem wolumenów różnego typu
- Dynamiczne oraz statyczne zarządzanie wolumenami w klastrze

DODATKOWE FUNKCJONALNOŚCI

- Kontrolowanie poprawnego funkcjonowania aplikacji za pomocą różnego rodzaju próbówek (Liveness, Readiness oraz Startup probes)
- Konfiguracja automatycznego skalowania aplikacji ze względu na obciążenie przy użyciu Horizontal Pod Autoscalers
- Uruchamianie kontenerów/zadań przed startem głównej aplikacji
- Startowanie aplikacji na określonych maszynach oraz konfiguracja zależności pomiędzy uruchomionymi kontenerami (nodeSelector, affinity/antiAffinity, taints/tolerations)
- Zarządzanie zasobami klastra: minimalnymi oraz maksymalnymi limitami przypisanymi do kontenerów
- Priorytety aplikacji w klastrze oraz wywłaszczanie kontenerów z niskim priorytetem
- Utrzymanie maszyn klastra wraz z przygotowaniem okien obsługi
- Omówienie polityk aktualizacji aplikacji (recreate, ramped, blue/green, canary, a/b testing, shadow)

LOGOWANIE ORAZ MONITORING

- Omówienie różnych architektur zbierania logów w klastrze Kubernetes wraz z przeglądem najpopularniejszych narzędzi m.in Elastic Stack, Jaeger, Logback
- Centralny monitoring zasobów w klastrze na przykładzie: Prometheus, AlertManager, Grafana

KUBERNETES CZ.2

BEZPIECZEŃSTWO

- Testy penetracyjne klastra Kubernetes
- Ograniczenie uprawnień oraz kontrola dostępu aplikacji do komponentów systemu operacyjnego z użyciem SecurityContext
- Wymuszanie globalnych standardów specyfikacji aplikacji wykorzystując Pod Security Policy

DYSTRYBUCJE ORAZ UŻYTECZNE NARZĘDZIA

- Kubernetes jako usługa na przykładzie najpopularniejszych dostawców chmur publicznych (GKE, EKS, AKS)
- Zautomatyzowana instalacja klastra przy użyciu Kubernetes Operations (KOPS)
- Instalacja klastra Kubernetes "on premise" (Kubespray)
- Kubeless, Kubeapps, kubectx, kubens i inne narzędzia użyteczne w codziennej pracy z klastrem Kubernetes
- Dobre praktyki / 12 factor appy

HELM JAKO NATYWNY MENEDŻER PAKIETÓW APLIKACJI W KUBERNETES

WPROWADZENIE DO ISTIO SERVICE MESH

- Wprowadzenie
- Service proxy (Envoy)
- Zarządzanie ruchem
- Monitorowanie i tracing
- Bezpieczeństwo