Agenda kursu Kubernetes Masters

📌 Kurs **Kubernetes Masters** to rewolucyjny na skalę Polski program szkoleniowy, dzięki któremu zoptymalizujesz proces wytwarzania i wdrażania oprogramowania. Szkolenie zostało podzielone na bardzo obszerną, **przekrojową część główną oraz trzy dodatkowe ścieżki tematyczne** dla DevOpsa, SecOpsa i Developera.

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

Podsumowanie kursu w pigułce

Kurs	Liczba lekcji	Czas trwania
Entrypoint - Ścieżka Zasadnicza	40 + 56 labów	5,5 godziny
Ścieżka DevOps	28 + 43 labów	4,5 godziny

Kurs	Liczba lekcji	Czas trwania
Ścieżka SecOps	26 + 45 labów	4,5 godziny
Ścieżka Developera	16 + 42 labów	4,5 godziny
SUMA:	110 lekcji teoretycznych + 186 labów	**19 godzin**

Spis treści

Entrypoint - Ścieżka Zasadnicza

Wprowadzenie

- Czym jest Kubernetes?
  - Co znaczy Kubernetes? Kto jest autorem? Kiedy został udostępniony kod?
  - Jakie są alternatywne rozwiązania do K8s?
  - Jak wyglądają alternatywne / przestarzałe metody zarządzania infrastrukturą?
  - Jak Kubernetes plasuje się względem alternatywnych metod wdrożenia i zarządzania infrastrukturą?
  - Co K8s potrafi od razu, zaraz po instalacji?
  - Czym K8s nie jest? Czego K8s nie potrafi?
  - Jakie są zalety / wady K8s względem innych rozwiązań?
  - Czym jest stan?
- Struktura klastra Kubernetes

    - Jak wygląda struktura klastra K8s? Na jakie logiczne części się dzieli?
    - Czym jest node?
    - Czym jest Control Plane?
    - Czym jest Worker poola?
    - Jak wysokopoziomowo wygląda struktura klastra?
    - Jakie są komponenty (szczegółowe) klastra?
    - Czym jest kube-apiserver? Jaka jest odpowiedzialność kube-apiserver?
    - Czym jest etcd? Jaka jest jego odpowiedzialność?
    - Czym jest kube-scheduler? Jaka jest jego odpowiedzialność?
    - Czym jest kube-controller-manager? Jaka jest jego odpowiedzialność?
    - Czym jest cloud-controller-manager? Jaka jest jego odpowiedzialność?
- Komponenty workerpooli
    - Jakie usługi uruchamia maszyna worker-pooli?
    - Czym jest kubelet? Jaka jest jego odpowiedzialność?
    - Czym jest kube-proxy? Jaka jest jego odpowiedzialność?
    - Czym jest container-runtime? Czy musi być to Docker?
- Czym są obiekty?
    - Czym jest obiekt? Co reprezentuje?
    - Jak wygląda interakcja z obiektami K8s? Co rozumiemy przez interakcję imperatywną i deklaratywną? Jakie są metody interakcji? Jak wygląda zarządzanie obiektami od strony plików?
    - Co to jest specyfika obiektu?
    - Co to jest status obiektu?

- Czym są przestrzenie?
  - Co to jest przestrzeń nazw w kontekście K8s?
  - Jaka jest odpowiedzialność przestrzeni nazw?
  - Jakie obiekty są ograniczone do przestrzeni nazw?
  - Jakie przestrzenie są domyślne?
- Czym są Pod-y?
  - Co to jest Pod? Czym Pod różni się od kontenera?
  - Jakie dodatkowe funkcjonalności dostarcza?
  - Jak rozumiemy Pod w kontekście K8s? Czy jest coś mniejszego co można wdrożyć? Jakie zasoby Pody współdzielą?
  - Jak przestrzenie sieciowe łączą się z Podami?
  - Jakie są sposoby zarządzania Podami?
  - Jak wygląda cykl życia Poda?
  - Jakie są stany Podów? Jakie są stany kontenerów?
- Sondy diagnostyczne
  - Czym są sondy diagnostyczne? Do czego służą?
  - Jakie są typy sond diagnostycznych? Jaka jest ich odpowiedzialność?
  - Jakie są mechanizmy sprawdzające?
  - Jaki może być wynik sondy? Jak zareaguje na odpowiedni wynik klaster?
  - Kiedy użyć której sondy diagnostycznej?
  - Jakie akcje możemy wykonać przy określonym cyklu życia?
- Limity zasobów
  - Jakie są limity zasobów? Czym różnią się requests i limits?

    - Co możemy limitować?
- Tryb uprzywilejowany
    - Czym jest tryb uprzywilejowany? Dlaczego nie powinien być wykorzystywany?
- Inne elementy Pod-a
    - Jakie są inne elementy Poda?
    - Czym są kontenery typu init? Jakie muszą spełnić warunki, żeby Pod się uruchomił? Jakie są możliwe zastosowania? Czym są reguły affinity?
    - Czym jest PDB? Czym są Pody statyczne?
- Co to jest etcd?
    - Czym dokładnie jest baza danych etcd? Czym jest algorytm Raft?
    - Jak osiągane jest kworum? Jak osiągana jest wysoka dostępność?
    - Jaką rolę odgrywa etcd w K8s?
- Czym jest kontroler?
    - Jaka jest odpowiedzialność kontrolerów w klastrze?
    - Jakie są podstawowe kontrolery działające w klastrze?
    - Jakie dodatkowe kontrolery może zainstalować administrator?
    - W oparciu o co działa kontroler?
    - Jak wygląda kontrola bezpośrednia?
- Pod kontra kontener
    - Czym różni się Pod od kontenera?
    - Dlaczego wprowadzono wyższą warstwę abstrakcji?
    - Co gwarantuje taka warstwa?

    - Czym jest kontener typu sidecar?

Podstawy pracy z klastrem

- Jak funkcjonuje Kubernetes API?
    - Jak funkcjonuje K8s API? Czym jest OpenAPI?
    - Co globalnie umożliwia K8s API?
    - Jakie są metody interakcji z API?
    - Jak wygląda podział obiektów na grupy?
    - Jakie wersjonowanie przyjęto w K8s? Jak wygląda kompatybilność wsteczna?
    - Jakie są reguły wygaszania API? Jak wygląda globalny okres wsparcia?
    - Jak wygląda utrzymywanie obiektów niestandardowych?
- Feature gates
    - Czym jest feature-gate? Co umożliwiają feature-gates?
    - Jakie są domyślne konfiguracje dla alpha, beta, GA?
- Czym jest kubectl?
    - Co to jest kubectl? Gdzie utrzymywana jest konfiguracja?
    - Jak wygląda struktura poleceń kubectl?
    - Do czego służy kubectl? Przykłady poleceń, które możemy wydać.
    - Przykłady typów, o które możemy zapytać. Jaki jest wynik polecenia kubectl?
    - Jak filtrować obiekty? Co to znaczy, że K8s jest agnostyczny względem formatu?
- Czym jest kubeconfig?
    - Czym jest plik kubeconfig?
    - Gdzie jest zlokalizowany? Czym jest kontekst?

- Zarządzanie uprawnieniami
  - Jak wygląda autoryzacja i uwierzytelnianie?
  - Jakie są podstawowe tryby autoryzacji? Czym różni się tryb Node, RBAC, ABAC, Webhook? Czym są obiekty Role, ClusterRole? Czym są obiekty RoleBinding, ClusterRoleBinding?
  - Jak wygląda agregacja obiektów typu ClusterRole?
  - Czym są podmioty? Jak wyglądają odnośniki do podmiotów?
  - Jacy są domyślni użytkownicy w K8s? Jakie są domyślne role w K8s?
- Jak funkcjonują etykiety?
  - Czym jest etykieta? Jak wygląda unikalność etykiet?
  - Kto nadaje etykiety? Jakie są ograniczenia etykiet?
  - Jakie są preferowane prefiksy?
  - Jakie typy selektorów wyróżniamy? Czym są selektory względem etykiet?
  - Przykład selektora.
- Etykiety kontra adnotacje
  - Czym jest adnotacje? Jak wygląda unikalność adnotacji?
  - Jakie są ograniczenia co do kluczy i wartości adnotacji?
  - Kiedy używamy etykiet, a kiedy adnotacji?
- ReplicaSet, DaemonSet, StatefulSet, Deployment
  - Czym są obiekty typu StatefulSet, DaemonSet, Deployment oraz ReplicaSet?
  - Czym różnią się te obiekty między sobą? Jaka jest odpowiedzialność każdego z nich? Kiedy użyć którego obiektu?
  - Czym jest etykieta pod-template-hash?

    - Jakie są metody aktualizacji Podów? Czym jest metoda Recreate oraz RollingUpdate?
    - Przykłady zastosowania DaemonSet-u; alternatywy dla DaemonSetu.
    - Które aplikacje należy hostować na StatefulSetcie? Jakie są ograniczenia StatefulSetu? Jak wygląda konfiguracja sieci w StatefulSetcie? Czym jest VolumeClaimTemplate? Jak wygląda proces kasacji Podów w Statefulsetcie?

Przechowywanie danych

- Czym są wolumeny?
    - Czym jest przestrzeń dyskowa w kontekście K8s? Jakie są jej predykaty?
    - Czym jest emptyDir? Co to znaczy, że wolumen jest efemeryczny?
    - Co to jest wolumen? Czy wolumeny są zależne od działania Poda?
    - Jakie są tryby wolumenu?
    - Co to jest CSI? Przykłady CSI.
    - Jakie są ograniczenia wolumenów?
    - Jakie są typy wolumenów?
    - Czym są wolumeny dostawców sieciowych?
    - Jaki jest cykl życia wolumenu? Jakie są tryby dostępu do wolumenu?
    - Jak wygląda montowanie wolumenów?
    - Co się dzieje z wolumenem po odmontowaniu?
    - Czym jest obiekt PersistentVolume (PV) oraz PersistentVolumeClaim (PVC)?
    - Jak wygląda rozszerzane przestrzeni dyskowej?

    - Czy wolumeny wspierają migawki / snapshoty?
    - Czy wolumeny wspierają klonowanie?
- Czym są StorageClasses?
    - Czym jest StorageClass? Co umożliwia StorageClass?
- Automatyczne przydzielanie wolumenów
    - Jak automatycznie i dynamicznie przydzielać wolumeny?
    - Jak zależne od topologii infrastruktury jest tworzenie wolumenów?
    - Jak wygląda użycie obiektu PV oraz PVC?
- Wolumen typu HostPath
    - Jakie są zastosowania dla wolumenu hostPath?
    - Czy bezpieczne jest wykorzystywanie tego wolumenu?
    - Jakie są zagrożenia związane z hostPath?
- Współdzielenie przestrzeni między kontenerami
    - Jak wygląda współdzielenie przestrzeni między kontenerami?
    - Kiedy współdzielenie przestrzeni między kontenerami jest przydatne?
    - Jakie są zagrożenia związane z tą przestrzenią?

Zarządzanie sekretami

- Zarządzanie sekretami
    - Jak wygląda zarządzanie sekretami w K8s?
    - Co zawiera obiekt Secret? Jak jest kodowany?
    - Jakie ograniczenia ma obiekt Secret?
    - Jakie są typy obiektów Secret?
    - Jakie są różnice między typami Opaque, service-account-token, dockercfg, basic-auth, ssh-auth, tls?

    - Jak wygląda dostarczanie sekretów do kontenera?
    - Czym są sekrety niezmiennicze?
    - Dlaczego obiekt Secret nie jest sekretny?
- ConfigMap vs Secret
    - Do czego służy obiekt typu ConfigMap?
    - Jakie ograniczeina ma obiekt typu ConfigMap?
    - Jakie są sposoby przekazania obiektu ConfigMap do kontenera?
    - Czy są niezmiennicze ConfigMapy?
    - Jaka jest różnica między ConfigMap a Secret?
    - Kiedy używać obiektów typu ConfigMap?
    - Kiedy używać obiektów typu Secret?
- Integracja ze źródłami zewnętrznymi
    - Jak wygląda integracja z zewnętrznymi źródłami sekretów?
    - Jakie są zalety i wady zewnętrznych źródeł?
    - Jakie są inne metody przetrzymywania sekretów?
- Rotacja sekretów
    - Dlaczego sekrety należy rotować?
    - Jakie są wyzwania przy utrzymywaniu sekretów w infrastrukturze?
    - Jakie są best-practices rotowania sekretów?

Podsumowania graficzne

- Komponenty K8s
- Obiekty K8s
- Pod
- Sondy diagnostyczne

- Limity zasobów
- Etykiety i adnotacje
- Wolumeny
- StorageClass

Laboratoria

Konfiguracja środowiska laboratoryjnego

- Opis środowiska laboratoryjnego
- Opis dostępnych platform wirtualizacyjnych
- Wprowadzenie do VMware Fusion
- Instalacja Kali cz. I
- Instalacja Kali cz. II (utworzenie maszyny)
- Instalacja Kali cz. III (autologin)
- Instalacja toolboxa cz. I
- Instalacja toolboxa cz. II
- Weryfikacja sieci
- Instalacja toolboxa cz. III
- Instalacja toolboxa cz. IV
- Instalacja toolboxa cz. V
- Dostęp po SSH
- Instalacja toolboxa cz. VI (omówienie narzędzi)
- Instalacja Docker
- Instalacja zsh
- Instalacja kubectl
- Instalacja fzf
- Instalacja Helm

- Instalacja krew
- Instalacja kubectx oraz kubens
- Instalacja minikube
- Final, final, poprawiony snapshot
- Instalacja kind
- Instalacja kctx, kns

Laboratoria ścieżki

- Utworzenie klastra K8s
- Opis konfiguracji w kubectl
- Skoki między klastrami
- Eksport obiektów z kind - logi
- Usunięcie klastrów - kind
- Utworzenie klastra z pliku - kind
- Dodanie roli workera - kind
- Uruchomienie klastra w trybie high-availability - kind
- Pokazanie procesów master-worker
- Klaster bez high-availability
- Klaster z high-availability
- Failover - etcd
- Kind - procesy
- Opisanie maszyn w obrębie klastra
- Praca z przestrzeniami nazw
- Listowanie obiektów w klastrze
- Oskryptowane listowanie obiektów
- Utworzenie pierwszego Deployment - nginx

- Tworzenie obiektu z różnych źródeł
- Różne zachowania klastra, dla różnych obiektów
- Detaliczny opis obiektów - explain
- Przekazywanie portów - kubectl port-forward
- Jeden Pod - wiele kontenerów
- Utworzenie `Daemonsetu`
- Znacznik pod-template-hash
- Bazowe cofanie wersji deploymentu
- Strategia aktualizacji - RollingStrategy
- Wolumen - emptyDir
- Wolumen - hostPath
- Wolumen - readOnly
- Łączenie wielu wolumenów

Ścieżka DevOps

Sieci w Kubernetes

- Realizacja sieci w klastrze
  - Czym jest sieć?
  - Wymagania dotyczące sieci w klastrze K8s
  - Przestrzenie sieciowe w klastrze K8s
  - Netfliter; ipvs oraz iptables
  - Jak wygląda topologia sieci?
  - Jak Pody komunikują się ze sobą?
  - Jak wygląda topologia klastra?
  - Czym jest Service w klastrze K8s?

    - Jak działają Service różnych typów? NodePort, ClusterIP, ExternalName oraz LoadBalancer.
    - Jak wygląda flow requestu?
- Wtyczki sieciowe i ich zastosowanie
    - Czym jest LPAM? Czym jest CNI?
    - Jakie są dostępne wtyczki CNI?
    - Calico, WaveNet, Flannel, Canal, AWS CNI.
    - Który CNI należy wybrać?
    - Jeden klaster - wiele CNI.
- Złożone stosy sieciowe
    - Jak wyglądają złożone stosy sieciowe?
    - Jak wygląda dual-stack -- IPv4 / IPv6?
- DNSser
    - kube-dns vs CoreDNS
    - Jakie są schematy nazewnictwa serwisów w CoreDNS?
    - Co jest ważne przy monitorowaniu CoreDNS?
- Polityki sieci
    - Na jakiej warstwie działają polityki NetworkPolicies?
    - Z jakich elementów składa się obiekt NetworkPolicies?
    - Jakie są domyślne ustawienia takiego obiektu?
    - Jakie są zastosowania tego obiektu?
    - Czego polityki NetworkPolicies nie potrafią?

Komunikacja aplikacji ze światem zewnętrznym

- Ingress
    - Jakie odpowiedzialności ma obiekt typu Ingress?

    - Czym jest ingressClassName?
    - Co oznacza termin “to satisfy an Ingress”?
    - Jakie są przykłady ingress-controllerów?
    - Jak wybrać ingress-controller?
    - Czym ingress-controllery różnią się między sobą?
    - Co jest ważne w monitoringu ingress-controllera?
    - Jak wystawić aplikację z klastra na ruch zewnetrzny?
- Load-balancing
    - Parametry load-balancingu różnych ingress-controllerów
    - Jakie są ograniczenia takiego load-balancingu?
    - Które elementy wysokiej dostępności powinne zostać wyniesione poza ingress-controller?

Wdrożenia i zarządzanie aplikacjami

- Helm
    - Co składa się na pojedyncze wdrożenie?
    - Jaką odpowiedzialność ma Helm?
    - Czym jest Helm Chart?
    - Czym jest repozytorium Helm Chartów?
    - Jakie są alternatywy do Helm Chartów?
    - Jak renderowane są szablony manifestów?
    - Czym są helpery?
    - Jaką odpowiedzialność mają pliki NOTES.txt, values.yaml oraz Chart.yaml?
    - Jaka jest architektura Helma?
    - Jak działają wtyczki Helmowe?
- Release, wersjonowanie i funkcje pomocnicze

    - Co rozumiemy przez wydanie w kontekście Helma?
    - Jak wygląda wersjonowanie, rollback oraz upgrade?
    - Czym jest backend wydania?
- Helm2 vs. Helm3
    - Jakie są różnice między Helm2 a Helm3?
    - Jakie problemy generował Helm2? Jak te problemy rozwiązuje Helm3?
- Skalowanie aplikacji
    - Co wpływa na skalowanie aplikacji?
    - Jakie są dwie główne metody skalowania aplikacji?
    - Względem jakich metryk możemy skalować aplikację?
    - Jakie są wady i zalety skalowania wertykalnego i horyzontalnego?
    - Na jakich warstwach możemy te skalowanie realizować?
    - Jak fizycznie przebiega skalowanie wertykalne oraz horyzontalne?
    - Jak automatycznie skalować infrastrukturę? Jakie obiekty do tego służą?
    - Jak działa jest HPA / VPA? Czym jest cluster-autoscaler oraz Karpenter?
- Zadania cykliczne i jednorazowe
    - Jak realizowane są zadania cykliczne w K8s?
    - Czym różnią się od at oraz crontab?
    - Co jest efektem zadań cyklicznych w K8s?
    - Czym są polityki zrównoleglenia? Jakie problemy generują polityki zrównoleglenia?
    - Jaka jest wartość dodana zadań cyklicznych w K8s względem zadań standardowych?

Monitoring

- Monitoring i capacity planning
  - Jak realizować monitoring w klastrze K8s?
  - Dlaczego nie istnieje dobry klaster K8s bez monitoringu?
  - Jakie metryki możemy uznać za minimalne, optymalne oraz wystarczające dla monitoringu klastra K8s?
  - Jak wprowadzić automatyzację względem monitoringu klastra?
  - Jak określać potrzeby oraz capacity-planning?
  - Jakie są dobre praktyki związane z monitorowaniem klastra?
- Narzędzia
  - Jak wyglądają natywne narzędzia monitorowania klastra?
    - node-problem-detector
    - Metrics-server
    - kubectl top
  - Jakie są narzędzia komercyjne?
  - Jakie są narzędzia open-source?

Disaster recovery

- Zakłócenia w pracy Pod-a
  - Czym spowodowane są niedostępności w pracy Poda?
  - Jak dzielimy zakłócenia? Czym są zakłócenia zamierzone i niezamierzone?
  - Jak wygląda eksmisja Poda? W jakich sytuacjach się dzieje?
  - Czym jest okres łaski? Jak wygląda twarda eksmisja?
  - Jak można przeciwdziałać zakłóceniom?
- Backupowanie klastra K8s
  - Co należy backupować?

    - Jakie są techniki backupowania klastra?
    - Jakie narzędzia są pomocne przy backupowaniu?
    - Dlaczego warto rozważyć wykonywanie kopii całego klastra w kontekście Infrastructure as Code?
    - Jakie są dobre praktyki związane z kopiami zapasowymi?
- Self healing
    - Jaka jest definicja self-healingu?
    - Jak w tym kontekście wyglądają sondy livenessProbe oraz readinessProbe?
    - Dlaczego współpraca między aplikacją a klastrem jest ważna?
    - Jakie są warunki dla poprawnego działania self-healingu?
- Scenariusze reagowania na awarie i disaster-recovery
    - Co należy do podstaw reagowania na awarie?
    - Jak wygląda pętla reagowania na awarię?
    - Co się dzieje gdy node utraci sprawność?
    - Scenariusze reagowania:
        - Disaster response -- Awaria jednej instancji
        - Disaster response -- Awaria trzech i więcej instancji
        - Disaster response -- Out of memory kills
        - Disaster response -- Nowe Pody są terminowane ze względu na przekroczony czas oczekiwania
- Stan aplikacji
    - Jak wygląda stan aplikacji?
    - Gdzie należy utrzymywać stan aplikacji?
    - Jak wygląda stanowość przy bazach danych?
    - Jak wygląda stanowość przy rozproszonych systemach plików?

    - Jak wygląda wzorcowy przykład aplikacji bezstanowej?
    - Kontrprzykład aplikacji, która utrzymuje swój stan źle i niespójnie.
    - Jakie są best-practices utrzymywania stanu aplikacji?
- Sposoby na rozszerzenie klastra
    - Jakie są sposoby na automatyzację klastra?
    - Po czym poznać, ż klaster wymaga rozszerzenia?
    - Skalowanie horyzontalne i wertykalne
    - Metody rozszerzania i skalowania Control Plane
    - Problem z możliwościami szybkiego skalowania infrastruktury
    - Best practices rozszerzania klastra
- Przerwania - PodDisruptionBudget
    - Które obiekty chcemy ochronić przed przerwaniami?
    - Przed czym chcemy te obiekty uchronić?

Podsumowania graficzne

- Sieć w klastrze
- Service
- Ingress
- Helm
- Skalowanie
- Kopie zapasowe

Laboratoria

Konfiguracja środowiska laboratoryjnego

- Opis środowiska laboratoryjnego

- Opis dostępnych platform wirtualizacyjnych
- Wprowadzenie do VMware Fusion
- Instalacja Kali cz. I
- Instalacja Kali cz. II (utworzenie maszyny)
- Instalacja Kali cz. III (autologin)
- Instalacja toolboxa cz. I
- Instalacja toolboxa cz. II
- Weryfikacja sieci
- Instalacja toolboxa cz. III
- Instalacja toolboxa cz. IV
- Instalacja toolboxa cz. V
- Dostęp po SSH
- Instalacja toolboxa cz. VI (omówienie narzędzi)
- Instalacja Docker
- Instalacja zsh
- Instalacja kubectl
- Instalacja fzf
- Instalacja Helm
- Instalacja krew
- Instalacja kubectx oraz kubens
- Instalacja minikube
- Final, final, poprawiony snapshot
- Instalacja kind
- Instalacja kctx, kns

Laboratoria ścieżki

- Wydanie cert-managera
- Praca z wydaniami Chart-u
- Usuwanie wydania - Chart
- Pobranie tara - Helma
- Modyfikacja źródła wydania
- Różnica między wydaniami
- Różnica w wydaniach Helm
- Rollback wydania
- Helm template - helm lint
- Serwis typu ExternalName
- Serwis typu NodePort
- Serwisy typu ClusterIP
  - Konfiguracja połączenia między aplikacjami.
- Realizacja serwisu z wieloma Podami
- Instalacja sterownika CNI
- Utworzenie klastra na Digitalocean
- Inne parametry konfiguracyjne w klastrze chmurowym
- Utworzenie zewnętrznego Loadbalancera; ingress-managed LB
  - Wdrożenie ingress-controllera; umożliwienie dostępu do aplikacji.
- Instalacja metrics-servera
- Wprowadzenie zaawansowanych metryk
- Zarządzanie fizycznymi wolumenami na poziomie K8s

Ścieżka SecOps

Sieci w Kubernetes

- Realizacja sieci w klastrze
  - Czym jest sieć?
  - Wymagania dotyczące sieci w klastrze K8s
  - Przestrzenie sieciowe w klastrze K8s
  - Netfliter; ipvs oraz iptables
  - Jak wygląda topologia sieci?
  - Jak Pody komunikują się ze sobą?
  - Jak wygląda topologia klastra?
  - Czym jest Service w klastrze K8s?
  - Jak działają Service różnych typów? NodePort, ClusterIP, ExternalName oraz LoadBalancer.
  - Jak wygląda flow requestu?
- Wtyczki sieciowe i ich zastosowanie
  - Czym jest LPAM? Czym jest CNI?
  - Jakie są dostępne wtyczki CNI?
  - Calico, WaveNet, Flannel, Canal, AWS CNI.
  - Który CNI należy wybrać?
  - Jeden klaster - wiele CNI.
- Złożone stosy sieciowe
  - Jak wyglądają złożone stosy sieciowe?
  - Jak wygląda dual-stack -- IPv4 / IPv6?
- DNSser
  - kube-dns vs CoreDNS
  - Jakie są schematy nazewnictwa serwisów w CoreDNS?
  - Co jest ważne przy monitorowaniu CoreDNS?
- Polityki sieci

    - Na jakiej warstwie działają polityki NetworkPolicies?
    - Z jakich elementów składa się obiekt NetworkPolicies?
    - Jakie są domyślne ustawienia takiego obiektu?
    - Jakie są zastosowania tego obiektu?
    - Czego polityki NetworkPolicies nie potrafią?

Kubernetes i bezpieczeństwo

- Bezpieczeństwo jest skomplikowane
    - Dlaczego bezpieczeństwo jest skomplikowane?
    - Jak wygląda podział odpowiedzialności i krzywe decyzyjne?
    - O jakie obszary bezpieczeństwa należy zadbać?
    - Czym są zagrożenia wewnętrzne i zewnętrzne?
    - Jak wygląda modelowanie ryzyka? Jakie są standardy?
    - Co rozumiemy przez *Skarb*?
    - Dlaczego w wypadkowej zawsze coś jest kosztem czegoś w netsec?
    - Dlaczego broniący są na straconej pozycji, z definicji?
- Generalne pryncypia budowania bezpieczeństwa
    - Czym jest obrona w głąb?
    - Jak wygląda zasada najmniejszych uprawnień?
    - Dlaczego redundancja i duplikaty w bezpieczeństwie są potrzebne?
    - Co rozumiemy przez ograniczenie efektu fali?
    - Jak wygląda redukcja wektorów ataku oraz odpowiednie dostosowanie poziomu bezpieczeństwa?

Zarządzanie politykami

- Polityki

    - Czy są w szczególności Polityki? Jakie typy polityk wyróżniamy?
    - Czym jest LimitRange oraz Resource Quota?
    - Jak Resource Quota wpływa na bezpieczeństwo?
    - Jaka jest różnica między LimitRange a ResourceQuota?
    - Czym jest PIDs Limit? Czym jest Node Resource Manager?
    - Czym jest PIDs Reservation?
    - Czym jest PodSecurityPolicy / PodSecurityProfile oraz SecurityContext?
    - Jaka jest odpowiedzialność wszystkich wymienionych wyżej obiektów?
- Pod Security Policy
    - Czym jest PodSecurityPolicy? Jakie były z nim problemy?
    - Dlaczego został wycofany?
    - Czym jest PodSecurityAdmission? Jak różni się od PodSecurityPolicy? Czym jest konfigurowany? Na poziomie jakiego obiektu działa?
    - Czym jest PodSecurityStandards?
    - Czy różnią się profile baseline, restricted oraz priviledged?
    - W jakich szczegółowych obszarach różnią się te polityki?
    - Jaka jest różnica między security context a security profile?

Komunikacja aplikacji ze światem zewnętrznym

- Ingress
    - Jakie odpowiedzialności ma obiekt typu Ingress?
    - Czym jest ingressClassName?

    - Co oznacza termin “to satisfy an Ingress”?
    - Jakie są przykłady ingress-controllerów?
    - Jak wybrać ingress-controller?
    - Czym ingress-controllery różnią się między sobą?
    - Co jest ważne w monitoringu ingress-controllera?
    - Jak wystawić aplikację z klastra na ruch zewnetrzny?
- Load-balancing
    - Parametry load-balancingu różnych ingress-controllerów
    - Jakie są ograniczenia takiego load-balancingu?
    - Które elementy wysokiej dostępności powinne zostać wyniesione poza ingress-controller?

Bezpieczeństwo

- Bezpieczeństwo klastra API
    - Co jest ważne w bezpieczeństwie API K8s?
    - Co jest ważne w bezpieczeństwie etcd?
    - Czym jest log audytowy? Dlaczego należy go włączyć?
    - Jak badać integralność klastra? Co należy wyłączyć?
    - Jak zadbać o uprawnienia certyfikatów i kluczy w K8s; szczególnie tych należących do komunikacji mTLS?
    - Jakie techniki atakujący może zastosować atakując klaster?
    - Czym jest shadow-api?
- Atak na Helm
    - Jakie są techniki ofensywne na Helm? Jak realizowane są ataki phishingowe, ataki łańcucha dostaw, ataki na repozytorium?
    - Jak atakować serwer Tillera?

    - Jak wyglądają techniki defensywne na ataki na Helm?
- Bezpieczeństwo aplikacji
    - Co leży w warstwie bezpieczeństwa aplikacji?
    - Jakie elementy bezpieczeństwa / od strony infrastruktury / są istotne?
    - Jakie są techniki defensywne? Jak przenieść bezpieczeństwo w lewo?
    - Jak zminimalizować wektory ataku na obraz?
    - Jak zminimalizować wektory ataku na kontener?
    - Jak wykorzystać Ingress do minimalizacji wektorów ataku?
    - Które elementy Ingress należy wynieść poza controller?
- Bezpieczne przechowywanie sekretów
    - Jak bezpiecznie przechowywać sekrety?
    - Gdzie na pewno nie przetrzymywać sekretów aplikacji i infrastruktury?
    - Jakie wady ma zewnętrzny dostawca?
    - Jake zalety ma zewnętrzny dostawca sekretów?
    - Jakie wady i zalety ma utrzymywanie sekretów w obiekcie Secret?
    - Jakie wady i zalety ma utrzymywanie sekretów w hybrydzie ExternalSecrets + Secrets?
    - Jak wygląda drabina wdrożeniowa?

Eskalacja uprawnień

- Metody eskalacji uprawnień
    - Jakie są metody eskalacji uprawnień?
    - Jak wygląda wykorzystanie technik eskalacji przez wolumen?
    - Które wolumeny mogą zostać do tego wykorzystane?

    - Jak wygląda eskalacja przez docker.socket?
    - Jak wygląda metoda eskalacji przez ServiceAccount?
    - Jak wygląda eskalacja przez Cloud Metadata Server?
    - Czym są ataki boczne?
- OPA
    - Czy jest Open Policy Agent? Z czym współpracuje OPA?
    - Co to jest Rego? Jak pisane są reguły OPA?
    - Jak wygląda ogólny flow w przypadku OPA?
    - Jak działa OPA w kontekście K8s? Czy jest K8s Gatekeeper?
    - Jakie są metody wykorzystywania OPA?
    - Jak wymusić wersję obrazu, rejestr, limity, czy etykiety?

Compliance i mTLS

- AdmissionWebhooks | Compliance
    - Czym są AdmissionWebhooki?
    - Czym są ValidatingWebhooks oraz MutatingWebhooks?
    - Jak w ich kontekście rozumieć OPA?
    - Jak realizować zgodność i compliance z wykorzystaniem Webhooków?
    - Jakie są inne metody realizowania zgodności?
    - Jak zaangażować w to biznes oraz zbudować spójny proces?
    - Jak z powyższym wiążą się testy?
- mTLS
    - Czym jest mTLS? Jak mTLS zwiększa bezpieczeństwo?
    - Czy mTLS może być zrealizowane tylko na poziomie etcd?
    - Jak Linkerd wprowadza mTLS oraz ServiceMesh?

    - Czy można wprowadzić pełny Service Mesh oraz mTLS bez zmian w kodzie?

Podsumowania graficzne

- Sieć w klastrze
- Obrona w głąb
- Ważne elementy obronne
- Ingress
- Limity
- Metody eskalacji uprawnień

Laboratoria

Konfiguracja środowiska laboratoryjnego

- Opis środowiska laboratoryjnego
- Opis dostępnych platform wirtualizacyjnych
- Wprowadzenie do VMware Fusion
- Instalacja Kali cz. I
- Instalacja Kali cz. II (utworzenie maszyny)
- Instalacja Kali cz. III (autologin)
- Instalacja toolboxa cz. I
- Instalacja toolboxa cz. II
- Weryfikacja sieci
- Instalacja toolboxa cz. III
- Instalacja toolboxa cz. IV
- Instalacja toolboxa cz. V
- Dostęp po SSH
- Instalacja toolboxa cz. VI (omówienie narzędzi)

- Instalacja Docker
- Instalacja zsh
- Instalacja kubectl
- Instalacja fzf
- Instalacja Helm
- Instalacja krew
- Instalacja kubectx oraz kubens
- Instalacja minikube
- Final, final, poprawiony snapshot
- Instalacja kind
- Instalacja kctx, kns

Laboratoria ścieżki

- Domyślne Memory - LimitRange
- Domyślne CPU - Memory - LimitRange
- Minimum i maksimum - LimitRange
- Limity na obiektach abstrakcyjnych
- Limity - ResourceQuota - Pod
- Limity - PodProfile - nginx
- Polityki - PSP & PSA
- Hunt - kube-hunter (Zabezpieczenie klastra K8s)
- Hunt - Obrona przed zdobyciem wersji
- Konfiguracja docker-a - docker-bench-security
- Konfiguracja i audytowanie manifestów
- Analiza obrazu - docker (Bezpieczne przechowywanie sekretów)
- OPA - Przykład (Implementacja OPA)

- Docker-socket
- Instalacja linkerd
- Linkerd - instalacja wizualizacji
- Linkerd - prezentacja wizualizacji
- Linkerd - Automatyczne meshowanie aplikacji
- Linkerd - Automatyczny tracing
- Linkerd - Jaeger
- Linkerd - Wykazanie ruchu mTLS

Ścieżka Developera

Konfiguracja aplikacji, logi i metryki

- Kubernetes a aplikacja
  - Jak wyglądają predykaty dobrego projektowania aplikacji pod K8s?
  - Czy jest standard 12FACTOR?
  - Jak wyglądają założenia tego standardu?
  - Co rozumiemy przez termin *Share nothing*?
  - Co to znaczy, że aplikacja posiada konfigurowalny zestaw zależności?
  - Jak aplikacja powinna reagować na odpięcie takiego zasobu?
  - Jak wygląda podział odpowiedzialności przy odpowiednich etapach wdrożenia?
  - Które elementy dotyczą runtime a które buildtime?
  - Z czego składa się konfiguracja całej aplikacji?
- Schematy logowania w aplikacji

    - Jak wygląda agregacja logów w kontekście K8s?
    - Dlaczego *standardowe* mechanizmy logowania w K8s się nie sprawdzają?
    - Jaki format logów należy stosować?
    - Jakie są dobre praktyki takiego logowania?
    - Jakie są cele logowania?
    - Dlaczego cel logowania implikuje to co powinno się znaleźć w logu?
    - Jakie są dobre praktyki generycznego loggera w środowisku K8s?
    - Jakie są alternatywne metody logowania?
    - Jak wygląda bezpośrednia implementacja?
- Metryki i ich ekspozycja
    - Jak możemy eksponować metryki z aplikacji?
    - Czym są metody pull i push?
    - Jakie są zalety i wady każdej z tych technik?
    - Jakich protokołów możemy użyć do ekspozycji?
    - Dlaczego metryki są istotne?
    - Dlaczego metryki biznesowe są szczególnie istotne?
    - Jakie rodzaje metryk możemy eksponować?
- Przygotowanie aplikacji pod K8s
    - Jak wygląda przygotowanie aplikacji pod K8s?
    - Jakie są etapy wdrożenia aplikacji pod K8s?
    - Jakie są dobre praktyki budowania obrazu pod K8s?
    - Jakie są dobre praktyki budowania Chartu aplikacji pod K8s?

    - Jakie znaczenie ma Continuous Integration / Continuous Deployment w tym procesie?
    - Jakie są dobre praktyki CI? Jakie są dobre praktyki CD?
    - Co należy wykonać po wdrożeniu aplikacji?
    - Jak wygląda pętla wdrożenia aplikacji?
    - Jak dzielić odpowiedzialność za wdrożenie między zespołami?

Wdrożenia i zarządzanie aplikacjami

- Helm
    - Co składa się na pojedyncze wdrożenie?
    - Jaką odpowiedzialność ma Helm?
    - Czym jest Helm Chart?
    - Czym jest repozytorium Helm Chartów?
    - Jakie są alternatywy do Helm Chartów?
    - Jak renderowane są szablony manifestów?
    - Czym są helpery?
    - Jaką odpowiedzialność mają pliki NOTES.txt, values.yaml oraz Chart.yaml?
    - Jaka jest architektura Helma?
    - Jak działają wtyczki Helmowe?
- Release, wersjonowanie i funkcje pomocnicze
    - Co rozumiemy przez wydanie w kontekście Helma?
    - Jak wygląda wersjonowanie, rollback oraz upgrade?
    - Czym jest backend wydania?
- Helm2 vs. Helm3
    - Jakie są różnice między Helm2 a Helm3?

    - Jakie problemy generował Helm2? Jak te problemy rozwiązuje Helm3?
- Skalowanie aplikacji
    - Co wpływa na skalowanie aplikacji?
    - Jakie są dwie główne metody skalowania aplikacji?
    - Względem jakich metryk możemy skalować aplikację?
    - Jakie są wady i zalety skalowania wertykalnego i horyzontalnego?
    - Na jakich warstwach możemy te skalowanie realizować?
    - Jak fizycznie przebiega skalowanie wertykalne oraz horyzontalne?
    - Jak automatycznie skalować infrastrukturę? Jakie obiekty do tego służą?
    - Jak działa jest HPA / VPA? Czym jest cluster-autoscaler oraz Karpenter?
- Zadania cykliczne i jednorazowe
    - Jak realizowane są zadania cykliczne w K8s?
    - Czym różnią się od at oraz crontab?
    - Co jest efektem zadań cyklicznych w K8s?
    - Czym są polityki zrównoleglenia? Jakie problemy generują polityki zrównoleglenia?
    - Jaka jest wartość dodana zadań cyklicznych w K8s względem zadań standardowych?

Środowisko lokalne Developera

- Środowisko lokalne
    - Jak wygląda środowisko lokalne dewelopera?
    - Z czego składa się to środowisko?

    - Jak powinien wyglądać workflow środowiska lokalnego?
    - Jakie są bez practices budowania takiego środowiska?
    - Jakie są sposoby odzwierciedlania środowisk produkcyjnych?
    - Jakie są wady i zalety każdego z nich?
    - Szczególne przypadki problemów, które nie pojawiają się przy źle zbudowanym środowisku lokalnym, a które ostatecznie pojawią się w produkcji.
    - Jak wygląda flow pracy z testami?
    - Jakie rodzaje testów powinniśmy uruchamiać?
- Tilt
    - Jakie problemy rozwiązuje Tilt?
    - Jakie są komponenty Tilta? Co to jest Starlark?
    - Jakie etapy budowania środowiska rozwiązuje Tilt?
    - Co dodatkowo dostarcza Tilt?
    - Jakie są zalety tilta względem poprzednio wskazanych metod?

CI/CD

- CI/CD
    - Co składa się na pojedyncze wdrożenie?
    - Jakie są best-practices takiego wdrożenia?
    - Jak pętla DevOps odzwierciedla ten proces?
- Continuous Integration
    - Czym jest z definicji Continuous Integration?
    - Co jest efektem dobrego Continuous Integration?
    - Jaki jest wynik testów?
    - Jakie powinne być testy w kontekście CI?

    - Jak flow testów zdalnych spina się z flow testów lokalnych?
    - Co dają małe zmiany?
    - Co można testować w kontekście statycznej analizy kodu?
    - Czym są dobre testy?
    - Jakie są best practices dobrego Continuous Integration?
- Continuous Deployment
    - Czym jest z definicji Continuous Deployment?
    - Jakie są best practices dobrego Continuous deployment?
    - Jakie są przykłady testów po wdrożeniu?
    - Jakie są techniczne metody wdrożenia?
- Narzędzia
    - Co jest najważniejsze w narzędziach?
    - Jakie są narzędzia do kontroli wersji kodu, narzędzia do Continuous Integration oraz Continuous deployment?
    - Jakie narzędzia i system wybrać?

K8s a CI/CD i aplikacja

- K8s a CI/CD i aplikacja
    - Jakich problemów z aplikacją K8s nie rozwiąże?
    - Jakich błędów K8s nie rozwiążę?
    - Jaki stan abstrakcji podnosi K8s?
    - Jakie problemy z aplikacją K8s rozwiązuje?
    - Jaki koszt od strony aplikacji wprowadza K8s?

Laboratoria

Konfiguracja środowiska laboratoryjnego

- Opis środowiska laboratoryjnego
- Opis dostępnych platform wirtualizacyjnych
- Wprowadzenie do VMware Fusion
- Instalacja Kali cz. I
- Instalacja Kali cz. II (utworzenie maszyny)
- Instalacja Kali cz. III (autologin)
- Instalacja toolboxa cz. I
- Instalacja toolboxa cz. II
- Weryfikacja sieci
- Instalacja toolboxa cz. III
- Instalacja toolboxa cz. IV
- Instalacja toolboxa cz. V
- Dostęp po SSH
- Instalacja toolboxa cz. VI (omówienie narzędzi)
- Instalacja Docker
- Instalacja zsh
- Instalacja kubectl
- Instalacja fzf
- Instalacja Helm
- Instalacja krew
- Instalacja kubectx oraz kubens
- Instalacja minikube
- Final, final, poprawiony snapshot
- Instalacja kind
- Instalacja kctx, kns

Laboratoria ścieżki

- Wprowadzenie do laboratoriów
- Kodu źródłowego projektu
- Zbudowanie obrazu aplikacji — Dockerfile
- Wprowadzenie Helma do aplikacji
- Wprowadzenie prywatnego rejestru do klastra
- Instalacja Tilta
- Pierwsze uruchomienie Tilta
- Interfejs webowy Tilta
- Automatyczne wykrywanie zasobów sieciowych w Tilt
- Automatyczna aktualizacja kodu w Tilt
- W pełni wykorzystanie Tilta w projekcie
- Czyste uruchomienie projektu — Tilt
- Wystawienie port-u w Tilt-cie
- Weryfikacja live_update
- Wprowadzenie testów pull-request na poziomie Github Actions
- Wdrożenie przykładowego Continuous Integration
- Wprowadzenie Continuous Deployment
- Developer - Wprowadzenie CD
- Co jeszcze należałoby wdrożyć do pełnego CI/CD?